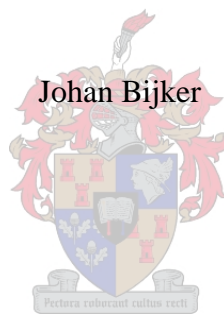


Development of an Attitude Heading Reference System for an Airship



Thesis presented in partial fulfilment of the requirements for the degree of
Master of Science in Electrical and Electronic Engineering at the University of Stellenbosch

Study leader: Prof WH Steyn

December 2006

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:

Date:

Abstract

A real time attitude and heading reference system (AHRS) was successfully implemented for use on an airship. The AHRS was tested on board a small airship (blimp) with real data supplied from the inertial measurement unit and GPS receiver.

The inertial measurement unit was built with lower grade sensors, resulting in significant reductions in component cost. To ensure accurate navigation results, the high rate inertial measurements were complemented with low rate GPS velocity and position updates in an extended Kalman filter configuration.

A study was made of various Kalman filter configurations, especially the possibility of splitting a big Kalman filter into smaller Kalman filters. It was found that the best trade-off between accuracy and processing power was achieved by having two smaller Kalman filters running in sequence. The first extended Kalman filter estimates the attitude of the airship, while the second extended Kalman filter estimates the velocity and position of the airship.

The two smaller Kalman filters were implemented on an onboard computer to provide real time estimates of the attitude, velocity and position of the airship.

Opsomming

'n Intydse oriëntasie-afskatter is suksesvol geïmplementeer vir gebruik op 'n lugskip. Die stelsel is aan boord van 'n klein lugskip getoets met werklike sensordata soos verskaf deur die inersiële meeteenheid en GPS ontvanger.

Die inersiële meeteenheid is gebou met swakker kwaliteit sensors, wat komponentkoste aansienlik verlaag het. Om akkurate navigasie resultate te verseker, is die hoëspoed inersiële metings met laespoed GPS snelheid en posisie opdaterings aangevul met behulp van 'n nie-lineêre Kalman-filter.

Verskillende Kalman-filterkonfigurasies is bestudeer, met spesifieke fokus op die moontlikheid om 'n groot Kalman-filter in kleiner Kalman-filters op te deel. Die beste akkuraatheid opgeweg teenoor verwerkingsvermoë kan verkry word deur twee kleiner Kalman-filters agtermekaar uit te voer. Die eerste nie-lineêre Kalman-filter skat die oriëntasie van die lugskip af, terwyl die tweede nie-lineêre Kalman-filter die snelheid en posisie van die lugskip afskat.

Die twee kleiner Kalman-filters is op 'n rekenaar aan boord van die lugskip geïmplementeer om intydse afskattings van die oriëntasie, snelheid en posisie van die lugskip te gee.

Acknowledgements

I would like to thank the following persons:

- Prof Steyn for all your guidance and suggestions.
- SunSpace for sponsoring the blimp.
- My lab partners for all of your help and discussions: Willem Hough, Jan-Cor Roos, George Thopil and Dolf Bredenkamp.
- Elise Geerts for your love and regular visits.
- Mariana Loots for proof reading this thesis.

Table of Contents

Chapter 1: Introduction and overview	1
1.1 Airships	1
1.2 Attitude and Heading Reference System	3
1.3 Aim.....	5
1.4 Outline.....	6
Chapter 2: Axes definitions and rotations.....	7
2.1 Body axial system	7
2.2 Inertial axial system	8
2.3 Earth axial system	8
2.3.1 ECEF rectangular axial system	8
2.3.2 ECEF geocentric axial system	8
2.3.3 NED axial system.....	9
2.4 Euler rotations	12
2.5 Euler Rotation rates.....	13
2.6 Quaternion.....	14
2.7 Quaternion rotation rates.....	17
Chapter 3: Kalman filter	20
3.1 Linear state space form	20
3.2 Discrete Estimation	21
3.2.1 Predictor estimator	21
3.2.2 Current estimator.....	22
3.2.3 Optimal discrete estimator	23
3.2.4 Summary of Kalman equations.....	25
3.2.5 Floating point operations	26
3.2.6 Evaluation of discrete process noise $Q(k)$	27
3.2.7 Evaluation of discrete measurement noise.....	28
3.3 Extended Kalman filter	28
Chapter 4: Hardware and calibration	30
4.1 Calibration.....	30
4.2 Allan Variance	31
4.2.1 Angle / velocity random walk.....	33
4.2.2 Rate random walk	34

4.2.3 Log-log plot.....	34
4.3 Accelerometer	36
4.3.1 Background	36
4.3.2 Noise characteristics	37
4.3.3 Dynamic and static acceleration	37
4.3.4 Calibration.....	38
4.3.5 Simulink model.....	38
4.3.6 Flight measurements	38
4.4 Gyroscope	40
4.4.1 Background	40
4.4.2 Noise characteristics	41
4.4.3 Calibration.....	41
4.4.4 Flight measurements	42
4.5 Magnetometer	43
4.5.1 Background	43
4.5.2 Reset/set function.....	45
4.5.3 Noise characteristics	45
4.5.4 Disturbances.....	46
4.5.5 Calibration.....	47
4.6 IMU.....	50
4.7 GPS receiver	50
4.7.1 Protocol.....	50
4.7.2 Differential GPS.....	51
4.7.3 Noise characteristics	51
4.7.4 Delay	52
4.7.5 Simulink model.....	52
4.7.6 Flight measurements	52
4.8 Simulations.....	54
4.9 Onboard computer.....	56
4.10 RF Link	56
4.11 Power budget.....	56
4.12 AHRS cost breakdown.....	57
Chapter 5: Attitude determination.....	59
5.1 Gyro integration	60
5.1.1 Definition	60

5.1.2 Simulation	61
5.2 TRIAD	63
5.2.1 Definition	63
5.2.2 Simulation	64
5.3 QUEST.....	65
5.3.1 Definition	65
5.3.2 Simulation	66
5.4 Multiplicative extended Kalman filter	67
5.4.1 Quaternion vector perturbations.....	68
5.4.2 Quaternion covariance matrix	70
5.4.3 Bias drifts	71
5.4.4 State space model.....	72
5.4.5 Measurement updates.....	72
5.4.6 Measurement noise figures	73
5.4.7 Initialisation	74
5.4.8 Simulations.....	74
Chapter 6: Inertial navigation	76
6.1 Gimballed inertial platform.....	77
6.2 Strap down platforms	78
6.3 Simplifications	78
6.4 Position Kalman filter	78
6.4.1 State space model.....	79
6.4.2 Simulation results.....	80
6.5 Attitude and position Kalman filter	81
6.5.1 Quaternion vector perturbations states.....	83
6.5.2 NED velocity perturbation states	83
6.5.3 Position perturbation states	84
6.5.4 State space model.....	85
6.5.5 Measurement matrix	86
6.5.6 Simulation with magnetometer and GPS updates.....	86
6.5.7 Simulation with magnetometer, accelerometer and GPS updates	87
6.5.8 GPS update rates	88
6.6 GPS delay states.....	88
6.7 Smaller attitude and velocity Kalman filter	90
Chapter 7: Solution implementation	93

7.1 Solution A	93
7.2 Solution B	93
7.3 Solution C	94
7.4 Solution D	94
7.5 Comparison	94
7.6 Solution selection	96
7.7 Solution implementation	96
7.8 Solution results.....	98
7.8.1 Attitude estimation.....	98
7.8.2 Velocity and position estimation.....	100
Chapter 8: Summary and recommendations	102
8.1 Summary	102
8.2 Recommendations.....	102
8.2.1 IMU.....	102
8.2.2 RF link.....	104
8.2.3 Differential GPS.....	104
8.2.4 Blimp control	105
Bibliography.....	106
Appendix A: Vector and quaternion operations in matrix format	110
A.1 Vector multiplication	110
A.2 Quaternion multiplication	110
Appendix B: Quaternion operations	111
B.1 Convert a quaternion to Euler angles	111
B.2 Convert Euler angles to a quaternion	111
B.3 Extracting a quaternion out of the direction cosine matrix	112
Appendix C: Matrix trace properties	113
Appendix D: Orthogonal matrix properties	115
Appendix E: Noise transforms	116
E.1 Quaternion noise to roll, pitch & yaw noise.....	117
E.2 ECEF rectangular noise to ECEF geocentric noise.....	118
Appendix F: Hardware.....	119
F.1 CANsense board	119
F.2 CANsenseIMU.....	122
F.2.1 IMU module.....	122
F.2.2 Magnetometer	124

F.2.3 Ultrasonic sensor.....	124
F.3 Sensor module.....	126
F.4 Power and GPS	127
Appendix G: Software	128
G.1 CANsense PIC	128
G.2 OBC.....	131

List of tables

Table 1-1: Comparison of aerial vehicles for environmental research (from: Elfes <i>et al.</i> 1998).	2
Table 3-1: Floating-point operations for matrix calculations.	26
Table 3-2: Number of FLOPS required for the Kalman filter equations.	26
Table 4-1: Noise characteristics of the accelerometers.	37
Table 4-2: Noise characteristics of the gyros.	41
Table 4-3: Magnetic field at Stellenbosch for 2004 (from: Sindle 2005).	44
Table 4-4: Noise characteristics of the magnetometer.	46
Table 4-5: Comparison between the reference and measured magnetic vector.	48
Table 4-6: Comparison between NMEA and UBX protocols.	51
Table 4-7: Power budget analysis of the AHRS.	57
Table 4-8: Cost breakdown of the AHRS.	58
Table 5-1: Attitude error statistics for gyro integration without bias drifts.	61
Table 5-2: Attitude error statistics for gyro integration with bias drifts.	62
Table 5-3: TRIAD attitude estimation error statistics.	65
Table 5-4: QUEST attitude estimation error statistics.	67
Table 5-5: MEKF attitude estimation error statistics.	75
Table 6-1: Position estimation error statistics.	81
Table 6-2: Estimation error statistics with magnetometer and GPS updates.	87
Table 6-3: Estimation error statistics with magnetometer, accelerometer and GPS updates.	88
Table 6-4: Attitude estimation error statistics for the reduced Kalman filter.	92
Table 7-1: Comparison of different Kalman filter solutions.	95
Table G- 1: Channel allocation for the sensor module.	130
Table G- 2: COM port settings of the OBC.	131
Table G- 3: Task priorities in AMX.	133

List of figures

Figure 1-1: Blimp used for test flights.	3
Figure 1-2: Integration of INS and GPS (from: Hjortsmarker 2005).	4
Figure 2-1: Body axes definition.	7
Figure 2-2: Definition of the ECEF geocentric axial system.	9
Figure 2-3: Definition of the NED axial system.	10
Figure 2-4: Transform the NED axial system through latitude λ (a) and longitude ϕ (b).	11
Figure 2-5: Definition of Euler 3-2-1 angle rotations.	12
Figure 2-6: Vector diagrams for the derivation of the transformation matrix (from: Goldstein <i>et al.</i> 2002).	15
Figure 3-1: Predictor estimator (from: Franklin <i>et al.</i> 1998).	21
Figure 3-2: Current estimator (from: Franklin <i>et al.</i> 1998).	22
Figure 4-1: Simulink model for random walk and measurement noise simulations.	35
Figure 4-2: Simulated bias drift and measurement noise (a) with the Allan deviation results (b). ...	36
Figure 4-3: Accelerometer data over time (a) with the Allan variance (b).	37
Figure 4-4: Simulink model for the accelerometers.	39
Figure 4-5: Magnitude of the accelerometer measurements during a test flight.	39
Figure 4-6: Gyro data over time (a) with the Allan variance (b).	41
Figure 4-7: Simulink model for the gyros.	42
Figure 4-8: Magnitude of the gyro measurements during a test flight.	43
Figure 4-9: Magnetometer data over time (a) with the Allan variance (b).	46
Figure 4-10: Illustration of hard iron (a) and soft iron (b) effects on the magnetometer.	47
Figure 4-11: Magnetometer calibration results.	49
Figure 4-12: Simulink model of the magnetometer.	49
Figure 4-13: Simulink model of the GPS.	53
Figure 4-14: GPS velocities during a test flight.	53
Figure 4-15: Simulink model of the motion of the blimp and the associated sensor outputs.	54
Figure 4-16: Simulated body accelerations (a) and resulting NED velocities of the blimp (b).	55
Figure 4-17: Simulated body rotation rates (a) and resulting attitude of the blimp (b).	55
Figure 5-1: Attitude errors for gyro integration without bias drifts.	61
Figure 5-2: Attitude errors for gyro integration with bias drifts.	62
Figure 5-3: TRIAD attitude estimation errors.	64
Figure 5-4: QUEST attitude estimation errors.	67

Figure 5-5: MEKF attitude estimation errors.....	74
Figure 5-6: MEKF bias estimation results	75
Figure 6-1: Basic principle of inertial navigation.	76
Figure 6-2: Attitude Kalman filter in parallel with the position Kalman filter.....	81
Figure 6-3: Velocity estimation errors.	82
Figure 6-4: Position estimation errors.....	82
Figure 6-5: Combined attitude and position Kalman filter.	83
Figure 6-6: RMS errors of the 12 state Kalman filter as a function of GPS update interval.	89
Figure 6-7: Reduced state attitude and position Kalman filters.....	90
Figure 7-1: Schematic layout of the final solution.....	96
Figure 7-2: Ground station display of the Kalman filter results.	97
Figure 7-3: Attitude estimation in terms of roll, pitch and yaw angles during a blimp flight.	98
Figure 7-4: Gyro bias estimates during a blimp flight.....	99
Figure 7-5: Position estimates of the blimp during a test flight.....	100
Figure 7-6: NED displacement estimates of the blimp during a test flight.....	101
Figure 7-7: Velocity estimates of the blimp during a test flight.	101
Figure 8-1: MEMS accelerometers (a) and gyros (b) temperature dependancies.....	103
Figure 8-2: Change in magnetic field over a one-second period compared with the roll rate of the blimp.	104
Figure F- 1: CANsense analogue schematics.	120
Figure F- 2: CANsense digital schematics.....	121
Figure F- 3: CANsenseIMU main schematic.....	122
Figure F- 4: Single axis IMU board.	123
Figure F- 5: Three axes IMU module.	123
Figure F- 6: Magnetometer interface schematics.....	125
Figure F- 7: Sensor axes layout.....	126
Figure F- 8: Power and GPS schematics.....	127
Figure G- 1: CANsenseIMU main loop.....	128
Figure G- 2: CANsenseIMU interrupt service routine.....	129
Figure G- 3: COM port interrupt service routine.	132
Figure G- 4: COM port transmit procedure.	133
Figure G- 5: AMX receiver task.	134

Acronyms

A/D	Analogue to Digital
AHRS	Attitude and Heading Reference System
AMR	Anisotropic Magneto Resistive
ARW	Angle Random Walk
AVAR	Allan Variance
bps	Bits per second
COTS	Commercial Off The Shelf components
CTP	Conventional Terrestrial Pole
DC	Direct Current
DCM	Direction Cosine Matrix
DGPS	Differential GPS
ECEF	Earth Centred Earth Fixed
EKF	Extended Kalman Filter
FLOPS	Floating Point Operations
FOG	Fibre Optic Gyro
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IC	Integrated Circuit
IEEE	Institute for Electrical and Electronics Engineers
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LIPO	Lithium Ion Polymer
LLH	Longitude, Latitude, Height
MEKF	Multiplicative Extended Kalman Filter
MEMS	Micro Electromechanical Sensor
MOEMS	Micro Optical Electromechanical Sensor
NED	North, East, Down
NMEA	National Marine Electronics Association
NTRIP	Networked Transport of RTCM via Internet Protocol
OBC	Onboard Computer

PC	Personal Computer
PCB	Printed Circuit Board
PSD	Power Spectral Density
RF	Radio Frequency
RLG	Ring Laser Gyroscope
RMS	Root Mean Square
RRW	Rate Random Walk
RTCM	Radio Technical Commission for Maritime Services Commission
RTOS	Real Time Operating System
UAV	Unmanned Aerial Vehicle
UBX	uBlox protocol
UKF	Unscented Kalman Filter
VRW	Velocity Random Walk
WGS	World Geodetic System
ZAR	South African Rands

Symbols / Nomenclature

Symbols:

ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
φ	Longitude (E/W)
λ	Latitude (N/S)
h	Height

Mathematical operations:

\otimes	Quaternion multiplication
\times	Vector cross product

State notation:

$\bar{\mathbf{x}}$	Propagated states
$\tilde{\mathbf{x}}'$	Error in propagated states
$\hat{\mathbf{x}}$	Estimated states
$\tilde{\mathbf{x}}$	Error in estimated states
$\bar{\mathbf{x}}$	3D vector

Matrices:

A	Direction cosine matrix
I	Identity matrix
F	State transition matrix in the continuous domain
G	Input matrix in the continuous domain
L	Estimator feedback gain
M	Error covariance matrix of $\bar{\mathbf{x}}$
P	Error covariance matrix of $\hat{\mathbf{x}}$
Q	Equivalent process noise covariance matrix in the discrete domain
R	Equivalent measurement noise covariance matrix in the discrete domain
Φ	State transition matrix in the discrete domain
Γ	Input matrix in the discrete domain

Chapter 1: Introduction and overview

1.1 Airships

Throughout history airships have been used for a number of special roles. During the First World War, airships were used as the first bombers and the first specialist anti-submarine aircraft. Count Zeppelin formed an airline starting intercontinental passenger carriage. Even after the First World War and the rapid development of the aeroplane, the airship still was the logical choice for long-range travel (Khoury *et al.* 1999).

However, public perception of the safety of airships was shattered after the R101 (British airship) crashed in 1930, and even more after the well-known Hindenburg disaster in 1936. In addition, advances in conventional aircraft like the development of jet engines and helicopters, stopped any use of airships.

During the early 1980's advantages of the airship was rediscovered and used for a variety of tasks, including surveillance, advertising, aerial photography, monitoring and research (Hima *et al.* 2002). More recently, there has been a tremendous interest in controllable unmanned airships as aerial inspection platforms due to their stability properties. Elfes *et al.* (1998) and Khoury *et al.* (1999) describe the benefits of using airships for exploration and monitoring applications:

- An airship can fly safely at low speed and low altitude, and is able to hover.
- Airships do not need a lot of energy to float in the air, only to manoeuvre.
- Smaller engines consume less fuel and produce less noise and turbulence, resulting in minimal environmental disturbance.
- Sensor noise is reduced due to low vibration.
- Compared to other aerial vehicles, airships have very long endurance.
- An airship can perform vertical take off and landing, thus it can be used in areas with limited logistic support.
- Airships have low radar and infrared signatures, which is ideal for military applications.

Table 1-1 compares the use of an aeroplane, helicopter and airship in a typical environmental monitoring application. Compliance is shown using marks: high compliance is indicated by three marks and low compliance by one mark.

Requirement	Aeroplane	Helicopter	Airship
Low operation cost	✓✓	✓	✓✓✓
Long endurance	✓✓	✓	✓✓✓
Hovering capability	✓	✓✓✓	✓✓✓
Payload to weight ratio	✓✓	✓	✓✓✓
High manoeuvrability	✓✓	✓✓✓	✓
Low noise and disturbance	✓	✓	✓✓✓
Vertical take-off and landing	✓	✓✓✓	✓✓✓
Low fuel consumption	✓✓	✓	✓✓✓
Low vibration	✓✓	✓	✓✓✓

Table 1-1: Comparison of aerial vehicles for environmental research (from: Elfes *et al.* 1998).

One can infer from Table 1-1 that airships are on the average more suited to environmental monitoring tasks than aeroplanes or helicopters.

The working of an airship depends on the law of buoyancy, which states that the buoyant force is equal to the weight of the displaced fluid, or displaced air in the case of airships. The average molecular mass of air is 28.5 g/mol¹, meaning any gas with a molecular mass lower than 28.5 g/mol displacing a volume of air will cause a buoyant force. Usually helium with a molecular mass of about 4 g/mol is used. Helium is an inert gas, which means it is not reactive under normal circumstances and is therefore safe to use.

Advances in aeroplane and helicopter unmanned aerial vehicles (UAV) led to the development of sensors, techniques, etc. needed for autonomous flight. At the University of Stellenbosch Carstens (2005), Peddle (2005), Venter (2005) and Groenewald (2006) worked on fixed wing and rotary wing vehicles, but relatively little work has been done towards the development of lighter than air vehicles. This thesis starts by developing an attitude heading reference system specifically for use by an airship.

Instead of taking measurements on an expensive airship, a relative small (4m length) blimp was used. A blimp is defined as a small airship that has no metal framework and collapses when

¹ Air consists of 78.1% N₂, 20.9% O₂, 0.9% Ar and other minor components.

deflated. Figure 1-1 shows the blimp used for test flights. It was filled with helium to provide the buoyant force.



Figure 1-1: Blimp used for test flights.

1.2 Attitude and Heading Reference System

An Attitude and Heading Reference System (AHRS) estimates the roll, pitch and yaw angles of a body relative to a reference frame. The AHRS system uses gyroscopes, accelerometers and magnetometers on all three axes to estimate the attitude. Usually a Kalman filter is used to combine all the measurements.

Advances in micro electromechanical sensors (MEMS) have led to significant developments in low cost inertial technology in the last decade (Hide *et al.* 2004). MEMS are sensors with at least some of their dimensions in the micrometer range (Madou 2002). Due to the need of effective, accurate, reliable and low-cost electronics for the automotive market, MEMS underwent enormous developments. However, application of MEMS is not limited to the automotive market and the devices can nowadays be found in a wide range of applications. Because of the ongoing development, these sensors are becoming better (reductions in size and weight) and cheaper due to mass production. However, the drawback is that MEMS are not yet able to reach the types of accuracies that are obtained from traditional technologies (Hide *et al.* 2004).

Development of Inertial Navigation Systems (INS) is also affected by the combination of MEMS and the Global Positioning System (GPS). Without GPS, the sensors used in an INS had to be very

accurate and stable to ensure good navigation results over a long time. The reason is that the navigation errors grow with time due to uncertainty in measurements and bias drifts. With the integration of GPS and INS, the navigation errors are always kept within a certain range, because GPS measurement errors do not grow over time.

Barbour (2001) describes why navigation with GPS alone is not a good solution. The problem is that GPS signals are prone to interference, can easily be obstructed and GPS measurements are usually transmitted at a low rate. INS, on the other hand, is not sensitive to interference or obstruction and offers a high measurement rate. The combination of the two delivers a high bandwidth system with limited errors while GPS is available. Lower grade INS sensors can be used, as the error growth is not a limiting factor any more. The accuracy will be better than both the INS and GPS alone. However, if the system must perform reasonably well during long times of GPS unavailability, good quality sensors must still be used. See Figure 1-2 or an overview of GPS and INS integration.

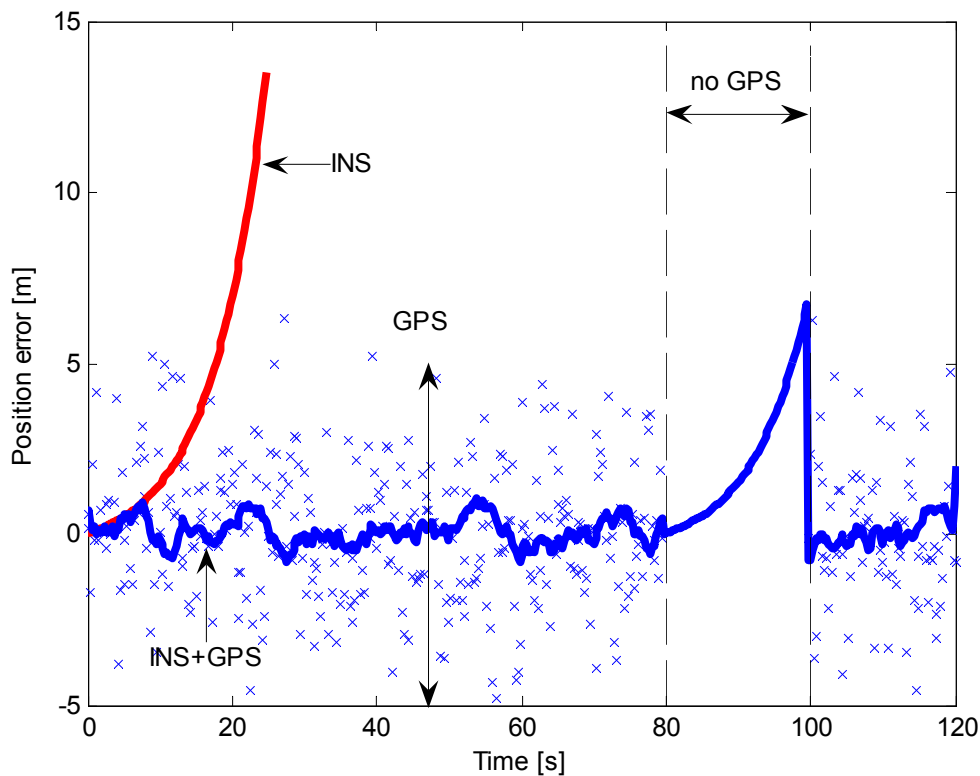


Figure 1-2: Integration of INS and GPS (from: Hjortsmarker 2005).

There are different types of GPS/INS integration methods (Hjortsmarker 2005). Loose integration is the process where the velocity and position updates of the GPS are used. This is the most common method of integration, but does not always deliver the best results. Tight integration is the process where the raw GPS pseudo range measurements are used. However, this creates additional complexity due to the tracking of satellites, but delivers better results. Deep integration is the process where the INS measurements are used to aid the GPS tracking loops.

In this thesis, loose integration was used, as most commercial GPS receivers do not transmit pseudo range measurements but only velocity and position measurements. Due to the simplicity, loose integration is also more stable. The AHRS was implemented on a low power PC104 computer with floating point capabilities for the Kalman filter.

More recent developments in the field of GPS/INS integration points towards using multiple Kalman filters (Hide *et al.* 2004). The idea is to run the different Kalman filters with different noise matrices, because the definition of noise models is usually not very accurate. All information from the Kalman filters is used to estimate the best noise matrix and the best estimates of the states. As this approach requires a lot of computational power, it was not used.

1.3 Aim

The aim of this study was to compare various methods of Kalman filtering for use as an AHRS on an airship. During the study the following factors were kept in mind:

- The AHRS must be a low-cost device. This implies that lower grade sensors had to be used. However, with more processing power and the integration of GPS, the same results could be obtained as when expensive high-grade sensors are used.
- The AHRS must use as little computational power as possible. As the computational power for a Kalman filter is a function of the number of states to the power of three (see section 3.2.5), every additional state implies a significant increase in computational power. In addition, the convergent time of the filter increases and the overall stability of the filter are decreased; reason enough to keep the Kalman filter as small as possible. A study was made to see if it is possible to split the AHRS into two sections: attitude determination and position determination.

1.4 Outline

Chapter 2 discusses the axial systems and rotation representation that was used. Chapter 3 focuses on state estimators and the development of the Kalman filter in general. Chapter 4 discusses all the sensors that are used for the AHRS system of an airship.

With all the background information in chapters 2 to 4, various attitude estimation techniques are discussed and compared in chapter 5. Chapter 6 focuses on the estimation of position and velocity. Chapter 7 compares the different solutions to find the best solution for implementation on a real time operation system, while chapter 8 concludes with a discussion of the results and suggestions for further work.

Chapter 2: Axes definitions and rotations

2.1 Body axial system

The attitude of a body is usually defined in terms of roll, pitch and yaw angles. Although the angles are usually defined in terms of an aircraft body, the same principles apply to a blimp.

Before the angles can be defined, a suitable body right-hand axial system needs to be defined. Let the X-axis point from the tail to the head of the blimp. Looking directly in the direction of the X-axis, the Y-axis will be to the right, and the Z-axis downwards. The origin of the body axial system is the blimp's centre of gravity.

A positive roll angle (ϕ) is defined as a clockwise rotation around the X-axis when looking into the direction of the X-axis, while a positive pitch angle (θ) is a clockwise rotation around the Y-axis when looking into the direction of the Y-axis. Finally, a positive yaw angle (ψ) is described as a clockwise rotation around the Z-axis when looking in the direction of the Z-axis. See Figure 2-1 for a graphical representation of the body axial system.

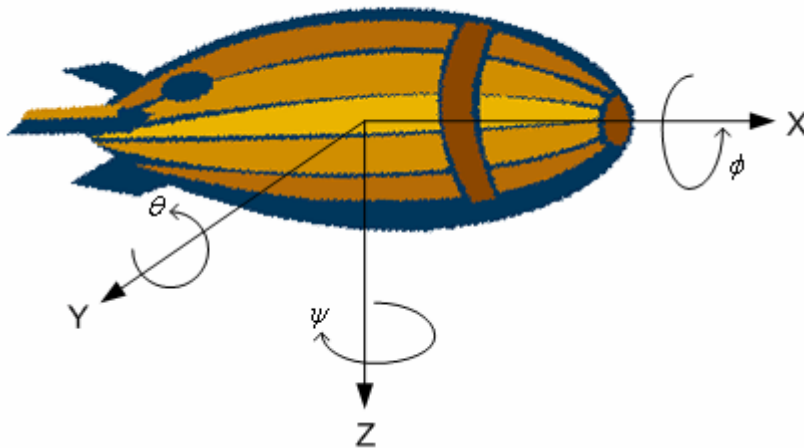


Figure 2-1: Body axes definition.

The body axial system is fixed to the body and is changed with respect to an inertial axial system when the body is moved. As the sensors are fixed to the body (known as a strap down configuration), the sensor outputs will be in terms of the body axial system.

2.2 Inertial axial system

An inertial axial system is defined as an axial system that does not change over time. Because the body axial system changes over time as the body is translating and rotating, the attitude of the body must be described with respect to an inertial (or reference) axial system.

2.3 Earth axial system

The earth axial system that was used in this thesis is called the earth centred, earth fixed (ECEF) axial system. The origin is at the centre of the earth, and the axial system rotates with the earth. The rectangular and geodetic ECEF systems are discussed below.

2.3.1 ECEF rectangular axial system

The ECEF rectangular axial system is defined as follows: The Z-axis points to the CTP (conventional terrestrial pole), which is the average North Pole location between 1900 and 1905 (El-Rabbany 2002). The X-axis passes through the equator at the Greenwich meridian, and the Y-axis is perpendicular to both the X-axis and Z-axis to form a right handed axial system, which passes through the equator at 90° East.

2.3.2 ECEF geocentric axial system

To make navigation easier, the earth's surface is divided into grids that are usually referred to as longitude and latitude. Longitude starts at the Greenwich meridian and is positive in an eastern direction and negative in a western direction. Longitude is given between -180° and $+180^\circ$, resulting in a span of 360° . The symbol for longitude is φ . Latitude is measured from the equator, with positive angles in a northern direction and negative angles in a southern direction. The North Pole is defined to be located at latitude $+90^\circ$ and the South Pole at latitude -90° . The symbol for latitude is λ . For a round earth, this axial system is called the ECEF geocentric axial system. The height h of an object will be defined as the height above the earth's radius R . See Figure 2-2 for the definition of the ECEF geocentric axial system.

To convert a position from ECEF rectangular to ECEF geocentric, the following formulas can be used:

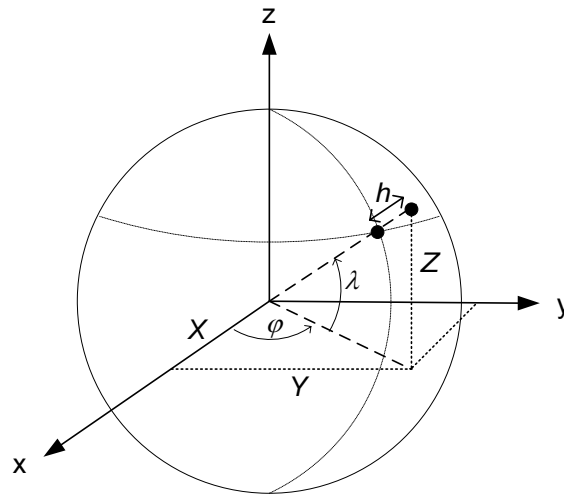


Figure 2-2: Definition of the ECEF geocentric axial system.

$$\varphi = \tan^{-1}\left(\frac{Y}{X}\right) \quad (2.1)$$

$$\lambda = \tan^{-1}\left(\frac{Z}{\sqrt{X^2 + Y^2}}\right) \quad (2.2)$$

$$h = \sqrt{X^2 + Y^2 + Z^2} - R \quad (2.3)$$

To convert from ECEF geocentric to ECEF rectangular, the following formulas can be used:

$$X = (R + h) \cos \lambda \cos \varphi \quad (2.4)$$

$$Y = (R + h) \cos \lambda \sin \varphi \quad (2.5)$$

$$Z = (R + h) \sin \lambda \quad (2.6)$$

The equations above are only valid for a round earth model. However, the earth is not round, and various earth models exist to describe the earth more accurately. The model mostly used is the World Geodetic System 1984 (WGS-84). For the purpose of this thesis, the round earth model is accurate enough, as the blimp will not travel at a high speed where the earth's radius will change significantly over time.

2.3.3 NED axial system

At a specific location on the earth, a convenient axial system to use for local navigation is the North East Down (NED) axial system. The first axis points north, the second axis points to the east, and

the third axis points down to the centre of the earth. The North-East plane is tangent to the geodetic ellipsoid as shown in Figure 2-3.

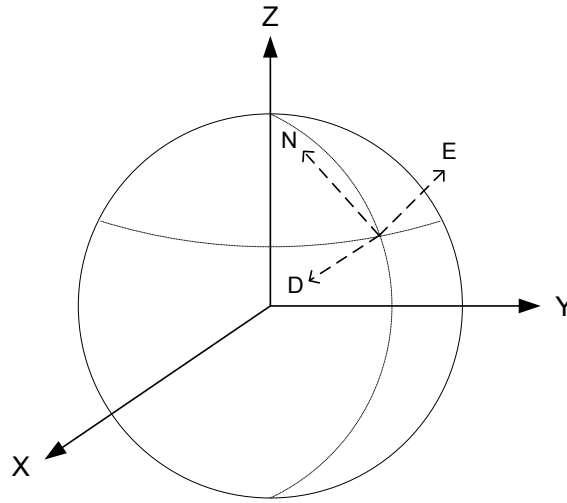


Figure 2-3: Definition of the NED axial system.

The reference axial system used in this thesis is the NED axial system. This means that if the body X-axis is aligned with north, body Y-axis is aligned with east, and body Z-axis is aligned downwards, the roll, pitch and yaw angles will all be 0 degrees.

As the NED axial system rotates with the earth and changes as a function of the position on the earth, it is not a strict inertial axial system. However, as airships usually fly at a low speed and do not travel far, the NED axial system can be assumed to be an inertial axial system.

To align the NED axial system $(\bar{x}_2, \bar{y}_2, \bar{z}_2)$ with the ECEF rectangular $(\bar{x}_0, \bar{y}_0, \bar{z}_0)$ axial system, the following two-step method can be used:

- Rotate the NED axial system through latitude λ to align the downward axis with the Z_0 -axis. See Figure 2-4(a). This intermediate axial system will be called $(\bar{x}_1, \bar{y}_1, \bar{z}_1)$.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} -\cos(90^\circ - \lambda) & 0 & -\cos \lambda \\ 0 & 1 & 0 \\ \sin(90^\circ - \lambda) & 0 & -\sin \lambda \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} -\sin \lambda & 0 & -\cos \lambda \\ 0 & 1 & 0 \\ \cos \lambda & 0 & -\sin \lambda \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (2.7)$$

- Transform the intermediate axial system $(\bar{x}_1, \bar{y}_1, \bar{z}_1)$ through the longitude to align X_1 with X_0 and Y_1 with Y_0 , as indicated by Figure 2-4(b).

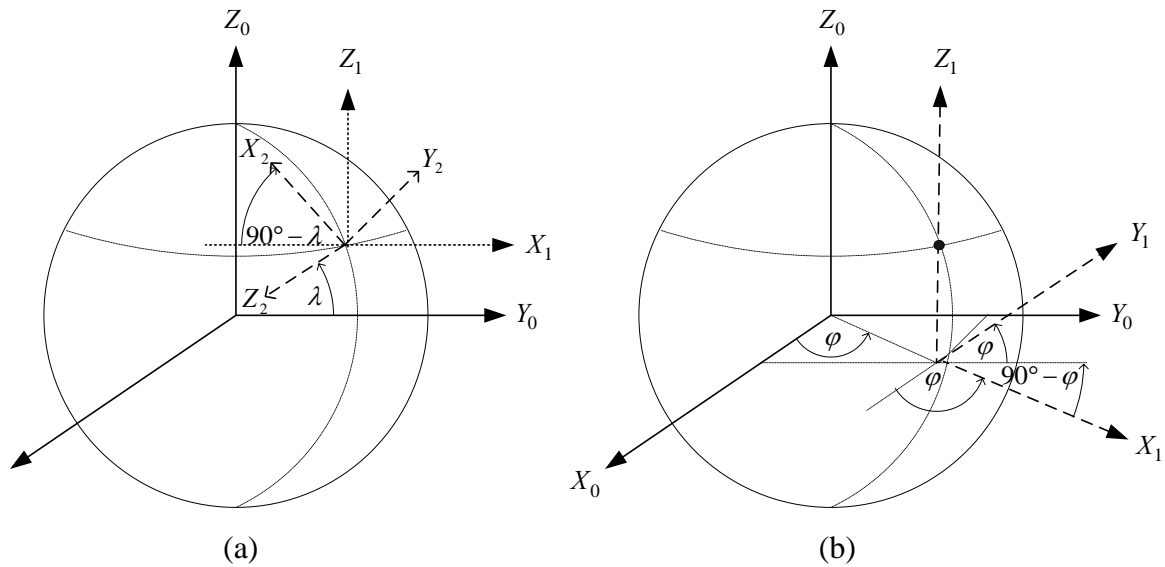


Figure 2-4: Transform the NED axial system through latitude λ (a) and longitude ϕ (b).

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \cos(90^\circ - \phi) & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (2.8)$$

Combining equations (2.7) and (2.8):

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} -\sin \lambda \cos \phi & -\sin \phi & -\cos \lambda \cos \phi \\ -\sin \lambda \sin \phi & \cos \phi & -\cos \lambda \sin \phi \\ \cos \lambda & 0 & -\sin \lambda \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (2.9)$$

Transforming a speed vector in NED coordinates to ECEF rectangular coordinates:

$$\dot{X} = -V_N \sin \lambda \cos \phi - V_E \sin \phi - V_D \cos \lambda \cos \phi \quad (2.10)$$

$$\dot{Y} = -V_N \sin \lambda \sin \phi + V_E \cos \phi - V_D \cos \lambda \sin \phi \quad (2.11)$$

$$\dot{Z} = V_N \cos \lambda - V_D \sin \lambda \quad (2.12)$$

Taking the time derivative of equations (2.1) to (2.3):

$$\dot{\phi} = \frac{X\dot{Y} - Y\dot{X}}{X^2 + Y^2} \quad (2.13)$$

$$\dot{\lambda} = \frac{1}{(R+h)^2} \left(\sqrt{X^2 + Y^2} \dot{Z} - \frac{XZ\dot{X}}{\sqrt{X^2 + Y^2}} - \frac{YZ\dot{Y}}{\sqrt{X^2 + Y^2}} \right) \quad (2.14)$$

$$\dot{h} = \frac{1}{R+h} (X\dot{X} + Y\dot{Y} + Z\dot{Z}) \quad (2.15)$$

Using equations (2.10) to (2.12), the equations above can be simplified:

$$\dot{\phi} = \frac{V_E}{(R+h) \cos \lambda} \quad (2.16)$$

$$\dot{\lambda} = \frac{V_N}{R+h} \quad (2.17)$$

$$\dot{h} = -V_D \quad (2.18)$$

Equations (2.16) to (2.18) are a convenient set of equations to work with as many GPS models give velocities in NED axial system and position in terms of longitude, latitude and height.

2.4 Euler rotations

The attitude of an object can be described in terms of the roll, pitch and yaw angles. In essence, the attitude of an object is the angles through which the inertial axial system must be rotated to align it with the body axial system.

The order of rotations is important (Milne 2001). In this thesis the Euler 3-2-1 system was used, which means that rotations are described first in terms of yaw, then pitch, then roll. See Figure 2-5 for the definition of Euler 3-2-1 angle rotations.

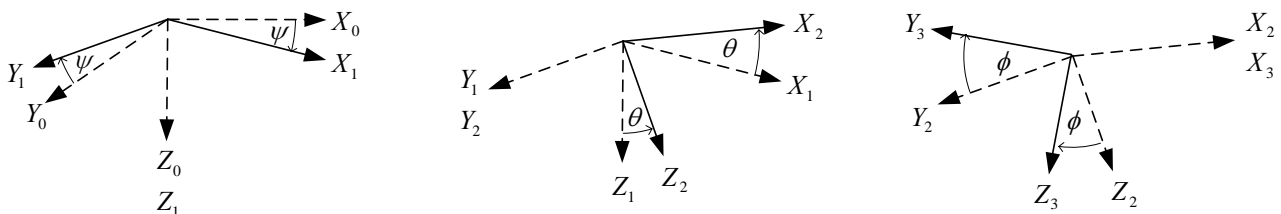


Figure 2-5: Definition of Euler 3-2-1 angle rotations.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (2.19)$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (2.20)$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (2.21)$$

Combining equations (2.19) to (2.21), and noticing that $(\bar{x}_3, \bar{y}_3, \bar{z}_3)$ is the body axial system, the transformation matrix that transforms a vector in inertial axes to body axes can be constructed. This transformation matrix is called the direction cosine matrix (DCM) and is denoted by \mathbf{A} :

$$\mathbf{A} = \mathbf{T}_\psi \mathbf{T}_\theta \mathbf{T}_\phi$$

$$= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.22)$$

2.5 Euler Rotation rates

The transformation matrix in equation (2.22) is only for the transformations of vectors. An alternative method is needed to describe rotation rates in terms of another axial system.

The total body spin of the blimp in terms of its body roll rates may be described as:

$$\bar{\omega} = \omega_x \bar{x}_3 + \omega_y \bar{y}_3 + \omega_z \bar{z}_3 \quad (2.23)$$

where $(\bar{x}_3, \bar{y}_3, \bar{z}_3)$ is the body axes as defined above.

The body spin may also be written as a change in yaw angle around the \bar{z}_0 axis, change in pitch angle around the \bar{y}_1 axis and change in roll angle around the \bar{x}_2 axis.

$$\bar{\omega} = \psi \bar{z}_0 + \dot{\theta} \bar{y}_1 + \dot{\phi} \bar{x}_2 \quad (2.24)$$

Using equations (2.19) to (2.21), the \bar{z}_0 , \bar{y}_1 and \bar{x}_2 axes can be written in terms of the body axes as follows:

$$\bar{z}_0 = -\sin \theta \bar{x}_3 + \cos \theta \sin \phi \bar{y}_3 + \cos \theta \cos \phi \bar{z}_3 \quad (2.25)$$

$$\bar{y}_1 = \cos \phi \bar{y}_3 - \sin \phi \bar{z}_3 \quad (2.26)$$

$$\bar{x}_2 = \bar{x}_3 \quad (2.27)$$

Rewriting equation (2.24) by using the equations above:

$$\bar{\omega} = \dot{\psi} \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}^T + \dot{\theta} \begin{bmatrix} 0 \\ \cos \phi \\ -\sin \phi \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}^T + \dot{\phi} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}^T \quad (2.28)$$

Combining equation (2.23) and (2.28):

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.29)$$

Taking the inverse of equation (2.29) gives:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.30)$$

The matrix in equation (2.30) becomes singular when θ becomes an odd multiple of 90° . This singularity is a major drawback when using Euler angles to represent attitude. However, several techniques that have no singularities have been developed (Friedland 1978). One of them is the quaternion, where the attitude is represented by four elements. Choukron *et al.* (2002) states that four elements is the minimum number of elements needed for a non-singular attitude representation.

2.6 Quaternion

Euler's theorem states that two arbitrarily oriented bases with common origin can be made to coincide with one another by rotating one of them through a certain angle about an axis that is passing through the origin (Wittenburg 1977). Gauss also had unpublished notes on the quaternion, and Hamilton introduced the quaternion as an abstract mathematical object (Markley 2003).

To derive the direction cosine matrix in terms of a quaternion, assume we have two axial systems: inertial and body axes. Assume the inertial and body axes are aligned before a rotation and a vector is described as $\bar{\mathbf{r}}$ in inertial axes. When the body axial system is rotated through an angle θ about a unit vector $\bar{\mathbf{u}}$, vector $\bar{\mathbf{r}}$ will be described as vector $\bar{\mathbf{r}}'$ in the rotated body axial system. See Figure 2-6 for vector diagrams.

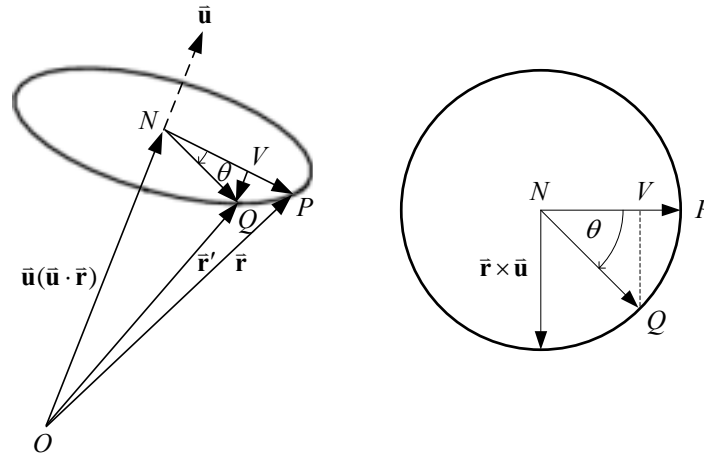


Figure 2-6: Vector diagrams for the derivation of the transformation matrix (from: Goldstein *et al.* 2002).

The vector in body axes can be described in terms of the vector in inertial axes:

$$\begin{aligned}\bar{\mathbf{r}}' &= \bar{\mathbf{u}}(\bar{\mathbf{u}} \cdot \bar{\mathbf{r}}) + [\bar{\mathbf{r}} - \bar{\mathbf{u}}(\bar{\mathbf{u}} \cdot \bar{\mathbf{r}})]\cos\theta + (\bar{\mathbf{r}} \times \bar{\mathbf{u}})\sin\theta \\ &= \bar{\mathbf{r}}\cos\theta + \bar{\mathbf{u}}(\bar{\mathbf{u}} \cdot \bar{\mathbf{r}})(1 - \cos\theta) + (\bar{\mathbf{r}} \times \bar{\mathbf{u}})\sin\theta\end{aligned}\quad (2.31)$$

Using the following trigonometry equities:

$$\cos\theta = 2\cos^2\frac{\theta}{2} - 1 \quad (2.32)$$

$$1 - \cos\theta = 2\sin^2\frac{\theta}{2} \quad (2.33)$$

$$\sin\theta = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2} \quad (2.34)$$

Equation (2.31) can be rewritten as:

$$\bar{\mathbf{r}}' = \left(2\cos^2\frac{\theta}{2} - 1\right)\bar{\mathbf{r}} + 2\sin^2\frac{\theta}{2}\bar{\mathbf{u}}(\bar{\mathbf{u}} \cdot \bar{\mathbf{r}}) + 2\sin\frac{\theta}{2}\cos\frac{\theta}{2}(\bar{\mathbf{r}} \times \bar{\mathbf{u}}) \quad (2.35)$$

Defining the following four parameters, called Euler parameters, or a normalised quaternion:

$$\mathbf{q} = \begin{bmatrix} \bar{\mathbf{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{u}} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.36)$$

Some textbooks use q_0 instead of q_4 , so that the first element in the quaternion will be the angle part and the last three elements will be the vector part. However, in this thesis the quaternion is used as defined in equation (2.36).

The dot product can be written in matrix format as:

$$\bar{\mathbf{u}} \cdot \bar{\mathbf{r}} = \bar{\mathbf{u}}^T \bar{\mathbf{r}} \quad (2.37)$$

Using equations (2.36) and (2.37), equation (2.35) can be rewritten as:

$$\bar{\mathbf{r}}' = (2q_4^2 - 1)\bar{\mathbf{r}} + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^T \bar{\mathbf{r}} + 2q_4(\bar{\mathbf{r}} \times \bar{\mathbf{q}}) \quad (2.38)$$

Rewriting equation (2.38) in matrix format (see Appendix A.1) and simplifying yields:

$$\bar{\mathbf{r}}' = \begin{bmatrix} 2q_4^2 + 2q_1^2 - 1 & 2(q_4q_3 + q_1q_2) & 2(q_1q_3 - q_4q_2) \\ 2(q_1q_2 - q_4q_3) & 2q_4^2 + 2q_2^2 - 1 & 2(q_4q_1 + q_2q_3) \\ 2(q_4q_2 + q_1q_3) & 2(q_2q_3 - q_4q_1) & 2q_4^2 + 2q_3^2 - 1 \end{bmatrix} \bar{\mathbf{r}} \quad (2.39)$$

The matrix in equation (2.39) can be seen as a matrix transforming vectors from inertial axes to body axes (the direction cosine matrix).

The quaternion norm is defined as (note that $\bar{\mathbf{u}}$ is a unit vector):

$$\sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = \sqrt{\bar{\mathbf{q}} \cdot \bar{\mathbf{q}} + q_4^2} = \sqrt{\bar{\mathbf{u}} \cdot \bar{\mathbf{u}} \sin^2 \frac{\theta}{2} + \cos^2 \frac{\theta}{2}} = 1 \quad (2.40)$$

Rewriting the matrix with the help of equation (2.40):

$$\mathbf{A} = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_4q_3 + q_1q_2) & 2(q_1q_3 - q_4q_2) \\ 2(q_1q_2 - q_4q_3) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_4q_1 + q_2q_3) \\ 2(q_4q_2 + q_1q_3) & 2(q_2q_3 - q_4q_1) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.41)$$

Markley (2003) states that a quaternion is not a unique representation of the attitude. It can be shown that \mathbf{q} and $-\mathbf{q}$ represent the same attitude.

2.7 Quaternion rotation rates

A constant vector $\bar{\mathbf{v}}_B$ in body axes is described in inertial axes by:

$$\bar{\mathbf{v}}_I = \mathbf{A}^{-1} \bar{\mathbf{v}}_B \quad (2.42)$$

Taking the time derivative of equation (2.42) and noticing that direction cosine matrix is orthogonal (see Appendix D), the time derivative can be written as:

$$\dot{\bar{\mathbf{v}}}_I = \dot{\mathbf{A}}^T \bar{\mathbf{v}}_B \quad (2.43)$$

However, the time derivative of equation (2.42) can also be written as relative motion in a rotating frame as:

$$\dot{\bar{\mathbf{v}}}_I = \dot{\bar{\mathbf{v}}}_B + \bar{\boldsymbol{\omega}}_B \times \bar{\mathbf{v}}_I = \bar{\boldsymbol{\omega}}_B \times (\mathbf{A}^T \bar{\mathbf{v}}_B) \quad (2.44)$$

Equation (2.43) and (2.44) will be equal, and after simplifying and writing in matrix form (see Appendix A) yields:

$$\begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} = \dot{\mathbf{A}} \mathbf{A}^T \quad (2.45)$$

Using the element in row 2 and column 3 of the matrix on the right side of equation (2.45), an expression for ω_1 could be obtained in terms of a quaternion and the time derivative of that quaternion. Similarly, the matrix element in row 3 and column 1 would give an expression for ω_2 ,

and the matrix element in row 1 and column 2 would give an expression for ω_3 . The results are given below in matrix format:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (2.46)$$

Taking the time derivative of the norm of a quaternion equation (2.40):

$$2q_1\dot{q}_1 + 2q_2\dot{q}_2 + 2q_3\dot{q}_3 + 2q_4\dot{q}_4 = 0 \quad (2.47)$$

Equation (2.47) can be added to equation (2.46) to form a skew symmetric 4x4 matrix:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (2.48)$$

The matrix in equation (2.48) is orthogonal (see Appendix D); therefore the inverse of equation (2.48) can be written as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \quad (2.49)$$

The matrix in equation (2.49) is not singular, as was the case with equation (2.30). Therefore, the quaternion can be seen as a non-singular representation of the attitude.

Defining the body angular rates as a quaternion:

$$\boldsymbol{\omega} = \begin{bmatrix} \vec{\boldsymbol{\omega}} \\ 0 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \quad (2.50)$$

Equation (2.49) can be written in terms of quaternion multiplication (see Appendix A.2) as:

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\omega} \otimes \mathbf{q} \quad (2.51)$$

Lefferts *et al.* (1982) list the following reasons why to use a quaternion:

- 1) The quaternion prediction equations are linear.
- 2) The quaternion representation is free from singularities.
- 3) The direction cosine matrix in terms of a quaternion is algebraic, eliminating the need for transcendental functions.

See Appendix B for more quaternion operations.

Chapter 3: Kalman filter

The Kalman filter was introduced in 1960 (Kalman 1960) and is defined as the time-varying optimal estimation solution (Franklin *et al.* 1998). A mathematical model of a plant is used to predict the outputs of the plant as a function of the inputs to the plant. The difference between the estimated outputs and measured outputs are used to correct the model.

The inputs to the plant and the measurements of the plant are usually corrupted with noise. All the noise sources in the system are assumed independent Gaussian random processes with zero mean. The statistical averages of the errors between the real and estimated states are traced through the model of the plant with the help of an error covariance matrix. The Kalman filter is used to calculate the optimal feedback gain as a function of the various noise figures, so that the errors between estimated states and real states are minimised.

Sorenson (1970) compares the Kalman filter with a least squares fit. He concludes that the Kalman filter is a recursive solution to Gauss' least squares problem. He also points out that if the input noise is zero, which implies a perfect model, the Kalman filter will not use any information from the measurement updates. Likewise, if the measurement noise is zero, the Kalman filter will not use any information from the model predictions.

3.1 Linear state space form

A continuous, linear state space system can be written in the form:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \quad (3.1)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{J}\mathbf{u} \quad (3.2)$$

where \mathbf{x} is the state vector, \mathbf{y} the output vector, \mathbf{F} the state transition matrix, \mathbf{G} the input matrix, \mathbf{H} the output matrix and \mathbf{J} the feed-forward matrix. \mathbf{J} will be assumed zero through the rest of this thesis.

The continuous state space system can be rewritten as discrete difference equations with sampling interval T in the following format (Franklin *et al.* 1998):

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) \quad (3.3)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (3.4)$$

where $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are defined below.

$$\mathbf{\Phi} = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F}^2T^2}{2!} + \frac{\mathbf{F}^3T^3}{3!} + \dots \quad (3.5)$$

$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G} \quad (3.6)$$

3.2 Discrete Estimation

3.2.1 Predictor estimator

When the difference between the measured output and the estimated output is used to correct the model, the divergence between the real and estimated output is minimised. See Figure 3-1 for a schematical layout of the predictor estimator.

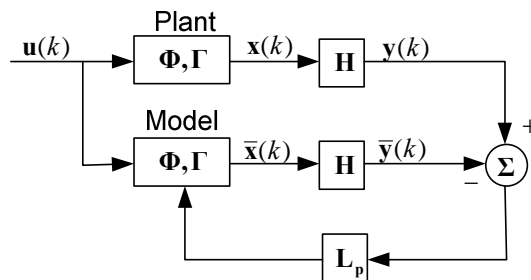


Figure 3-1: Predictor estimator (from: Franklin *et al.* 1998).

The estimator dynamics can be written as follows:

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\bar{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{L}_p[\mathbf{y}(k) - \mathbf{H}\bar{\mathbf{x}}(k)] \quad (3.7)$$

This type of estimator is called a predictor estimator.

3.2.2 Current estimator

Equation (3.7) states that the current state $\bar{\mathbf{x}}(k+1)$ is a function of the previous measurements $\mathbf{y}(k)$, not the current measurements $\mathbf{y}(k+1)$. In order to use the most recent measurements, the estimator must be modified as follows (called a current estimator):

$$\hat{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k+1) + \mathbf{L}[\mathbf{y}(k+1) - \mathbf{H}\bar{\mathbf{x}}(k+1)] \quad (3.8)$$

with

$$\bar{\mathbf{x}}(k+1) = \Phi\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) \quad (3.9)$$

See Figure 3-2 for a schematic layout. This type of estimator forms the base of a Kalman filter.

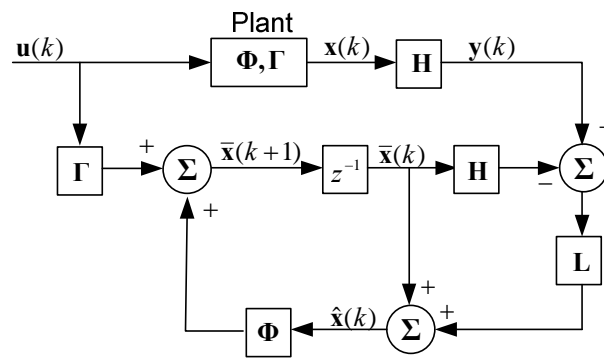


Figure 3-2: Current estimator (from: Franklin *et al.* 1998).

Using equations (3.8) and (3.9), the estimate errors can be written as:

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \\ &= [\Phi - \mathbf{LH}\Phi]\tilde{\mathbf{x}}(k) \end{aligned} \quad (3.10)$$

The estimate error must have zero mean to have an unbiased estimate of the state, or in mathematical terms:

$$E\{\tilde{\mathbf{x}}(k+1)\} = 0 \quad (3.11)$$

The covariance matrix of estimate error is defined as:

$$E\{\tilde{\mathbf{x}}(k+1)\tilde{\mathbf{x}}^T(k+1)\} = \mathbf{P}(k+1) \quad (3.12)$$

$\mathbf{P}(k+1)$ is a symmetric square matrix where the diagonal elements represents the state variances, and the rest is the cross correlation between the states. Equations (3.11) and (3.12) totally define the error characteristics.

3.2.3 Optimal discrete estimator

With the addition of discrete noise the current estimator equations change to:

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{w}(k) \quad (3.13)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (3.14)$$

where $\mathbf{w}(k)$ is the discrete process noise and $\mathbf{v}(k)$ is the discrete measurement noise. The process and measurement noise are assumed independent Gaussian random processes with zero mean.

The goal of optimal discrete estimation is to minimise the variance (noise) of the errors in equation (3.12). Ignoring the cross correlation, the sum of the diagonal entries of $\mathbf{P}(k)$ (called the trace of the matrix) will give the sum of state variances. If the trace of $\mathbf{P}(k)$ is minimised, the estimate error variances will be minimised.

Assume the process and measurement noise sources have the following characteristics:

$$E\{\mathbf{w}(k)\mathbf{w}^T(j)\} = \mathbf{Q}(k)\delta_{kj} \quad (3.15)$$

$$E\{\mathbf{v}(k)\mathbf{v}^T(j)\} = \mathbf{R}(k)\delta_{kj} \quad (3.16)$$

where δ_{kj} is the Kronecker delta, defined as

$$\delta_{kj} \equiv \begin{cases} 0 & \text{if } k \neq j \\ 1 & \text{if } k = j \end{cases} \quad (3.17)$$

When the noise is tracked through equation (3.13), the error between the true states and propagated states are:

$$\begin{aligned}\mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) - \mathbf{\Phi}\hat{\mathbf{x}}(k) - \mathbf{\Gamma}\mathbf{u}(k) - \mathbf{w}(k) \\ \tilde{\mathbf{x}}'(k+1) &= \mathbf{\Phi}\tilde{\mathbf{x}}'(k) - \mathbf{w}(k)\end{aligned}\quad (3.18)$$

The expected covariance of equation (3.18) will define the estimate errors after state propagation, called $\mathbf{M}(k+1)$. Assuming no correlation between \mathbf{w} and state errors it turns out to be:

$$\begin{aligned}\mathbf{M}(k+1) &= E\left\{(\tilde{\mathbf{x}}'(k+1))(\tilde{\mathbf{x}}'(k+1))^T\right\} \\ &= \mathbf{\Phi}\mathbf{P}(k)\mathbf{\Phi}^T + \mathbf{Q}(k)\end{aligned}\quad (3.19)$$

When the noise is traced through the feedback equation (3.8) and using equation (3.14):

$$\begin{aligned}\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) &= \mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1) - \mathbf{L}[\mathbf{H}\mathbf{x}(k+1) + \mathbf{v}(k+1) - \mathbf{H}\bar{\mathbf{x}}(k+1)] \\ \tilde{\mathbf{x}}(k+1) &= [\mathbf{I} - \mathbf{LH}]\tilde{\mathbf{x}}'(k+1) - \mathbf{L}\mathbf{v}(k+1)\end{aligned}\quad (3.20)$$

Taking the expected covariance of equation (3.20), and assuming no correlation between \mathbf{v} and state errors, the state error covariance after an update turns out to be:

$$\begin{aligned}\mathbf{P}(k+1) &= E\left\{(\tilde{\mathbf{x}}(k+1))(\tilde{\mathbf{x}}(k+1))^T\right\} \\ &= [\mathbf{I} - \mathbf{LH}]\mathbf{M}(k+1)[\mathbf{I} - \mathbf{LH}]^T + \mathbf{L}\mathbf{R}(k+1)\mathbf{L}^T\end{aligned}\quad (3.21)$$

It is possible to simplify equation (3.21), but this form is less susceptible to round off errors and keeps $\mathbf{P}(k)$ positive semidefinite. This form is called the Joseph form and the numerical stability justifies the more computational power (Bucy and Joseph 1968).

Taking into account that

$$(\mathbf{I} - \mathbf{LH})^T = (\mathbf{I}^T - \mathbf{H}^T\mathbf{L}^T) = (\mathbf{I} - \mathbf{H}^T\mathbf{L}^T)\quad (3.22)$$

and dropping the $(k+1)$ indices, equation (3.21) can be rewritten as:

$$\mathbf{P} = \mathbf{M} - \mathbf{LHM} - \mathbf{MH}^T\mathbf{L}^T + \mathbf{LHMH}^T\mathbf{L}^T + \mathbf{LRL}^T\quad (3.23)$$

Next, the trace of \mathbf{P} can be found as (see Appendix C for more information on the trace of a matrix):

$$Tr(\mathbf{P}) = Tr(\mathbf{M}) - 2Tr(\mathbf{LHM}) + Tr(\mathbf{LHMH}^T\mathbf{L}^T) + Tr(\mathbf{LRL}^T) \quad (3.24)$$

Any covariance matrix is symmetric, so that the transpose of any covariance matrix is equal to the covariance matrix.

To find the minimum of equation (3.24) with respect to \mathbf{L} (estimator gains), take the partial derivative of equation (3.24) to \mathbf{L} . The following result is obtained:

$$\frac{\partial Tr(\mathbf{P})}{\partial \mathbf{L}} = -2\mathbf{MH}^T + 2\mathbf{L}(\mathbf{HMH}^T) + 2\mathbf{LR} \quad (3.25)$$

where use have been made of equations (C.6) and (C.8).

When setting equation (3.25) to zero to find the value of \mathbf{L} that will result in the minimum value of equation (3.24), \mathbf{L} is solved to be:

$$\mathbf{L} = \mathbf{MH}^T(\mathbf{HMH}^T + \mathbf{R})^{-1} \quad (3.26)$$

Equation (3.26) can be seen as the Kalman filter gain that minimises the state error variances in the presence of process and measurement noise.

3.2.4 Summary of Kalman equations

Between measurements updates, only state propagation can be done. With every propagation, the noise figures increase due to the addition of new process noise.

- 1) State propagation:

$$\bar{\mathbf{x}}(k+1) = \Phi\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) \quad (3.9)$$

- 2) Increase in noise figures of the propagated states:

$$\mathbf{M}(k+1) = \Phi\mathbf{P}(k)\Phi^T + \mathbf{Q}(k) \quad (3.19)$$

If there are measurement updates available, the error between the propagated and measured outputs could be used to correct the model as defined below.

3) Calculation of the optimal feedback gain:

$$\mathbf{L}(k+1) = \mathbf{M}(k+1)\mathbf{H}^T [\mathbf{H}\mathbf{M}(k+1)\mathbf{H}^T + \mathbf{R}(k+1)]^{-1} \quad (3.26)$$

4) State update:

$$\hat{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k+1) + \mathbf{L}(k+1)[\mathbf{y}(k+1) - \mathbf{H}\bar{\mathbf{x}}(k+1)] \quad (3.8)$$

5) Update of noise figures:

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{L}(k+1)\mathbf{H}]\mathbf{M}(k+1)[\mathbf{I} - \mathbf{L}(k+1)\mathbf{H}]^T + \mathbf{L}(k+1)\mathbf{R}(k+1)\mathbf{L}(k+1)^T \quad (3.21)$$

3.2.5 Floating point operations

Defining a floating-point operation as the addition, subtraction, multiplication or division of two floating-point numbers, the number of floating point operations needed for basic matrix calculations can be calculated. Matrix multiplication is done by using the classic method, meaning no special speed-up algorithms are used. Matrix inversion is done by LU decomposition with forward and back substitution (Press 1992). See Table 3-1 for the FLOPS required to do matrix multiplication and inversion.

Matrix multiplication	$(n \times m) \times (m \times p)$	$2nmp$
Matrix inversion	$(n \times n)^{-1}$	$2n^3 + 3n^2 + 6n - 1$

Table 3-1: Floating-point operations for matrix calculations.

Assume a Kalman filter has n states, m inputs and p outputs (measurements). Now the number of floating point operations (FLOPS) for the Kalman filter equations can be calculated. See Table 3-2 for the number of FLOPS required for the different Kalman filter equations.

Equation number	FLOPS
(3.9)	$2n^2 + 2nm + n$
(3.19)	$4n^3 + n^2$
(3.26)	$2p^3 + 4p^2 + 4n^2p + 4np^2 + 6p - 1$
(3.8)	$4np + n + p$
(3.21)	$4n^3 + 4n^2p + 2np^2 + n^2$

Table 3-2: Number of FLOPS required for the Kalman filter equations.

The total number of FLOPS between measurements is:

$$4n^3 + 3n^2 + 2nm + n \quad (3.27)$$

The total number of FLOPS during measurement updates is:

$$4n^3 + n^2(8p + 1) + n(6p^2 + 4p + 1) + 2p^3 + 4p^2 + 7p - 1 \quad (3.28)$$

Knowing the update rate between measurements and the rate of measurement updates, the number of floating-point operations per second can be calculated. The number of FLOPS is a function of the number of states and number of measurements to the power of three, which means a significant increase in computing power for an increase in Kalman filter size.

3.2.6 Evaluation of discrete process noise $\mathbf{Q}(k)$

The continuous process noise must be converted to discrete process noise before it can be used in a discrete system. Using equation (3.6) to couple continuous noise into the discrete system, and recalculating equation (3.19) to use continuous process noise, the following result is obtained (Franklin *et al.* 1998):

$$\mathbf{M}(k+1) = \mathbf{\Phi P}(k)\mathbf{\Phi}^T + \int_0^T \int_0^T \mathbf{\Phi}(\eta)\mathbf{G}E\{\mathbf{w}(\eta)\mathbf{w}^T(\tau)\}\mathbf{G}^T\mathbf{\Phi}^T(\eta).d\tau.d\eta \quad (3.29)$$

If \mathbf{w} is assumed white noise, the autocorrelation of \mathbf{w} will have an impulse at the origin with size equal to the power spectral density of the noise (denoted by \mathbf{Q}_{psd}), and the equation above will simplify to:

$$\mathbf{M}(k+1) = \mathbf{\Phi P}(k)\mathbf{\Phi}^T + \int_0^T \mathbf{\Phi}(\tau)\mathbf{G}\mathbf{Q}_{psd}\mathbf{G}^T\mathbf{\Phi}^T(\tau).d\tau \quad (3.30)$$

Comparing equation (3.30) with equation (3.19), the discrete process noise can be evaluated as follows:

$$\mathbf{Q}(k) = \int_0^T \mathbf{\Phi}(\tau)\mathbf{G}\mathbf{Q}_{psd}\mathbf{G}^T\mathbf{\Phi}^T(\tau).d\tau \quad (3.31)$$

In the special case where T is short compared to system time constants, Φ will be close to the identity matrix and equation (3.31) will simplify to:

$$\mathbf{Q}(k) = T\mathbf{G}\mathbf{Q}_{\text{psd}}\mathbf{G}^T \quad (3.32)$$

3.2.7 Evaluation of discrete measurement noise

After comparing the output equation in a discrete system, equation (3.4), with the output equation in a continuous system, equation (3.2), one can see that there are no transformations from the continuous domain to the discrete domain. Therefore, the continuous measurement noise covariance matrix can be used directly in the discrete domain:

$$\mathbf{R}(k) = E\{\mathbf{v}(k)\mathbf{v}^T(j)\} \quad (3.33)$$

3.3 Extended Kalman filter

An extended Kalman filter is the application of a Kalman filter on a non-linear system. Assume the model of the plant can be written as non-linear equations in the following format:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (3.34)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \quad (3.35)$$

Using a first order Taylor series around the current state estimates, the perturbation plant dynamics can be approximated in state space format by taking partial differentiations to the different states:

$$\mathbf{F} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (3.36)$$

$$\mathbf{G} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (3.37)$$

$$\mathbf{H} = \left. \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (3.38)$$

This linearised system can be converted to the discrete domain as before using equations (3.5) and (3.6). The non-linear implementation of the Kalman filter is called an Extended Kalman Filter (EKF).

However, the error covariance matrices in an EKF would only be a linear approximation of the real error figures. Therefore, the EKF would not perform strictly optimal. Various methods exist to track the errors more carefully through a non-linear system. The unscented Kalman filter (UKF) uses a minimal set of sample points to capture the mean and covariance estimates, but it has been shown (LaViola 2003) that the UKF does not really improve estimates in the case of quaternion motion. The EKF was used throughout this thesis.

Chapter 4: Hardware and calibration

This chapter focuses on the various sensors needed in an AHRS system. A selection of commercial of the shelf (COTS) components was made for a real time implementation of the AHRS. During the selection of sensors, the aim of the thesis was kept in mind. The aim was to develop an AHRS using lower grade components, while ensuring good performance by optimal processing of the data.

As MEMS are becoming cheaper due to mass production (Hide *et al.* 2004), MEMS gyroscopes and accelerometers were chosen.

The Allan variance technique was used to characterise the sensors in terms of noise and bias random walk figures. A Simulink² model was constructed for every sensor to provide realistic sensor output to be used in simulations.

4.1 Calibration

A calibration method for every type of sensor is discussed in this chapter. Calibration is needed to compensate for deviations from the nominal gain, signal conditioning effects, A/D effects, sensor misalignments and sensor bias values for each specific sensor.

By using the theoretical gain from sensor to A/D values, all the above effects can be combined in a 3x3 matrix, also known as the cross coupling matrix, and one optional vector for bias values. If the sensors are perfectly aligned with the body axial system and all gains are exactly equal to the theoretical gains, the cross coupling matrix should be a unity 3x3 matrix. With small misalignments and gain deviations the cross coupling matrix should still be close to a unity 3x3 matrix.

All calibrations are done with respect to an available reference. The calibration will only be accurate if the reference was accurate.

If a reference is available, the calibration process can be described mathematically as:

² Simulink is an extension of MATLAB used to model and simulate dynamic systems. It is developed by The MathWorks (www.mathworks.com)

$$\begin{bmatrix} X_{AD} \\ Y_{AD} \\ Z_{AD} \end{bmatrix} = G_{SAD} \left(\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} X_{ref} \\ Y_{ref} \\ Z_{ref} \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \right) \quad (4.1)$$

where G_{SAD} is the theoretical gain from sensor to A/D value, \mathbf{C} the cross coupling matrix and \mathbf{D} the bias vector.

After the calibration is completed, the calibrated values can be calculated by using the inverse of equation (4.1):

$$\begin{bmatrix} X_{calib} \\ Y_{calib} \\ Z_{calib} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}^{-1} \left(\frac{1}{G_{SAD}} \begin{bmatrix} X_{AD} \\ Y_{AD} \\ Z_{AD} \end{bmatrix} - \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \right) \quad (4.2)$$

4.2 Allan Variance

The Allan Variance (AVAR) is a method of analysing a time sequence to pull out the intrinsic noise in a system as a function of the averaging time. It was originally developed by Dr. David Allan for the time and frequency standards community to characterise phase and frequency instability of precision oscillators (Allan *et al.* 1997). Because of the close analogies to inertial sensors, the method has been adapted to random drift characterisation of a variety of devices, including MEMS devices (Hou 2004).

It will be assumed that the uncertainty in sensor data is generated by noise sources of specific character. Hou (2004) lists the basic five noise terms:

1. Angle / velocity random walk
2. Rate random walk
3. Bias instability
4. Quantisation
5. Drift rate ramp

Only the first two sources will be used in this thesis, as these two noise sources dominate the noise on MEMS devices.

While the sensor is at rest, sample the output for a period. For most sensors, the sampled values will contain measurement noise plus a bias drift. The idea of the Allan variance analysis is to get the variance of the sampled data as a function of averaging time. For short average times, the variance will be dominated by high-frequency measurement noise, while for long average times the variance will be dominated by the low-frequency bias drift.

The principle working of the Allan Variance can be described as follows: Assume $x(k)$ is consecutive data points spaced with a sampling period of T_s . Divide the data into bins with a window time of T seconds, and calculate the average value $y(k)$ of each bin. Take the sum of the squared difference between two successive bins over the whole range and normalise it. The answer will give a quantitative measure of how much the average value changed for a specific window time T . The Allan variance is therefore a function of the window time T , and not a scalar value as in the case of normal deviation calculation.

For a window time of T seconds:

$$T = nT_s \quad (4.3)$$

The average values over the intervals of T seconds are:

$$y(k) = \frac{1}{n} \sum_{i=(k-1)n+1}^{kn} x(i) \quad (4.4)$$

The Allan variance can be described mathematically as

$$\sigma_A^2(T) = \frac{1}{2(N-1)} \sum_{k=1}^{N-1} (y(k+1) - y(k))^2 \quad (4.5)$$

where N is the number of bins in the data.

There is a unique relationship between the Allan variance and the power spectral density (PSD) of the random processes (Hou 2004). According to the IEEE standard, it is defined as:

$$\sigma_A^2(T) = 4 \int_0^{\infty} S(f) \frac{\sin^4(\pi fT)}{(\pi fT)^2} df \quad (4.6)$$

where $S(f)$ is the power spectral density of the random process.

4.2.1 Angle / velocity random walk

Angle / velocity random walk is the measurement noise on the output of the sensor that causes a random walk after the output of the sensor is integrated over time. Angle random walk is the integration of gyro output noise given in rad/s. Velocity random walk is the integration of accelerometer output noise given in m/s². According to the IEEE standard, this noise can be characterised by a white noise spectrum and the PSD is given as:

$$S_{ARW}(f) = S_{VRW}(f) = Q^2 \quad (4.7)$$

where Q is the angle / velocity random walk spectral density coefficient. Equation (4.7) specifies that the power stays the same over the whole frequency range. This is impossible in the real world, because it implies that the noise signal will have infinite energy. However, the white noise model can still be used as a good approximation of real noise when the bandwidth is restricted to a certain range.

Using equation (4.7) in equation (4.6) and simplifying gives (Hou 2004):

$$\sigma_{ARW}(T) = \frac{Q}{\sqrt{T}} \quad (4.8)$$

The SI-units of Q will be $rad/s/\sqrt{Hz}$ for an angle random walk and $m/s^2/\sqrt{Hz}$ for a velocity random walk.

To convert the angle / velocity random walk spectral density figure to a variance that can be used in the Kalman filter, the following equation must be used (Cemenska 2003):

$$\sigma_{ARW}^2 = 2 \int_{f_1}^{f_2} Q^2(f) df \quad (4.9)$$

The PSD of a signal is usually given as a double-sided PSD, where the spectral density is split in half between the positive and negative frequencies. For integration over a specified frequency range, the integration must also be done for negative frequencies. As the PSD is symmetric for real-type signals, the integration over the negative frequencies will be equal to the integration over the positive frequencies.

4.2.2 Rate random walk

Rate random walk is the instability of the output of the sensor. This can be due to temperature dependency of the bias value, or any other reason that causes the bias of the sensor to drift.

According to the IEEE standard, this noise can be characterised by a PSD of:

$$S_{RRW}(f) = \left(\frac{K}{2\pi f} \right)^2 \quad (4.10)$$

where K is the rate random walk spectral density coefficient. The power is frequency dependant with more power in lower frequencies, which makes sense, as bias drifts are usually slow processes. The rate random walk power spectral density can be seen as the integration of white noise.

Using equation (4.10) into equation (4.6) and simplifying gives (Hou 2004):

$$\sigma_{RRW}(T) = K \sqrt{\frac{T}{3}} \quad (4.11)$$

The SI-units of K will be $rad/s^2/\sqrt{Hz}$ for a gyro and $m/s^3/\sqrt{Hz}$ for an accelerometer.

To convert the rate random walk spectral density to a variance for use in the Kalman filter, equation (4.9) can be used, substituting Q with K .

4.2.3 Log-log plot

When the square root of the Allan variance (can also be called the Allan deviation) is plotted as a function of averaging time T on a log-log scale, equation (4.8) will appear as a straight line with a

slope of -0.5 and equation (4.11) will appear as a straight line with a slope of +0.5 on the log-log plot. The value of Q can be obtained where the negative slope crosses $T=1$ and the value of K can be obtained where the positive slope crosses $T=3$.

A bias drift combined with measurement noise was generated in Simulink (see Figure 4-1). The output of both noise sources are filtered through second order Butterworth filters to set the bandwidth.

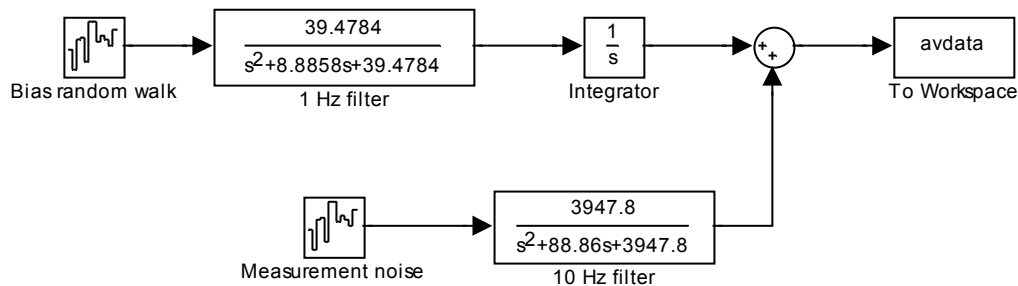


Figure 4-1: Simulink model for random walk and measurement noise simulations.

The simulation results as a function of time is given in Figure 4-2(a). The Allan deviation as a function of window time T is given in Figure 4-2(b). The straight lines with slopes of +0.5 and -0.5 can clearly be distinguished. Values of Q and K can be read off the graph, and by using equations (4.8) and (4.11) the noise densities can be found to be very close to the noise densities simulated in the noise source blocks.

For short window times, the Allan variance will be dominated by the angle / velocity random walk noise. As the window time increases, the Allan variance will decrease due to the averaging of noise. At a certain window time, the Allan variance will start to increase again, due to rate random walk noise. The minimum value of the Allan variance will give an indication what stability can be achieved with a fully modelled sensor and active bias estimation.

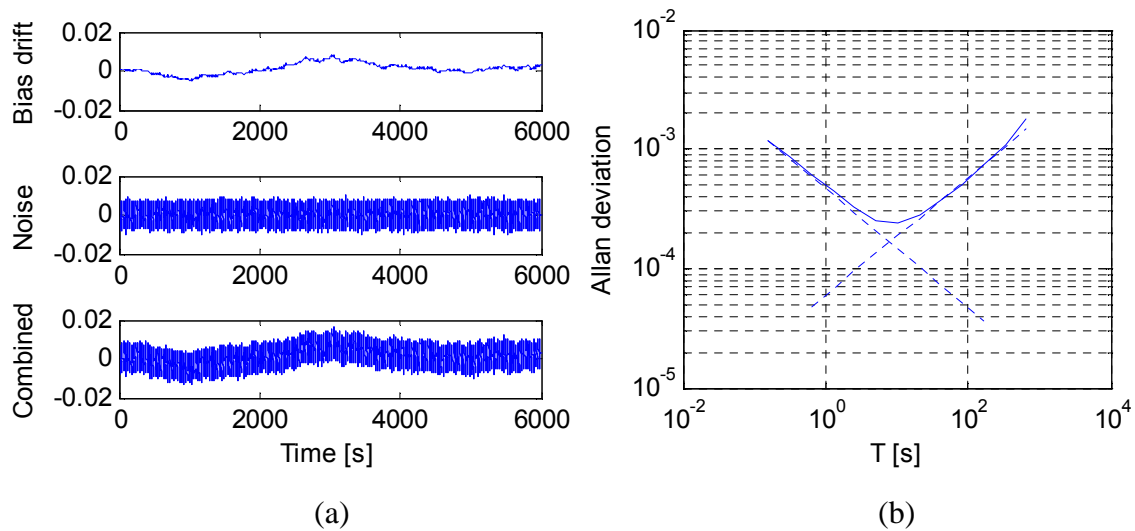


Figure 4-2: Simulated bias drift and measurement noise (a) with the Allan deviation results (b).

4.3 Accelerometer

4.3.1 Background

The first demonstration of a silicone micro machined accelerometer took place in 1979 at Stanford University. Over the years, a variety of accelerometers based on different physical detection schemes have been developed, but the capacitive method proved to be the best in terms of manufacturing. Mass production of the sensors had led to significant price drops. The principle market for accelerometers is crash detection and air bag release systems in the automotive market (Madou 2002). The working of the accelerometer can be described as a proof mass suspended by polysilicon springs that provide resistance against acceleration forces. The motion of the proof mass is detected by a capacitive pickoff, and demodulated to provide a voltage output as a function of the device's acceleration.

Analog Devices' ADXL103 and ADXL203 sensors were chosen due to the low cost and availability (Carstens 2005). It has been used by Carstens (2005), Peddle (2005), Groenewald (2006) and Venter (2006) in the ESL. The ADXL103 measures acceleration only in one axis, while the ADXL203 measures the acceleration in two perpendicular axes. The range of measurement is $\pm 1.7g$.

A triad of accelerometers is needed to measure both dynamic and static acceleration in three axes. The static acceleration is needed to measure the earth's gravitational vector for attitude determination as discussed in Chapter 5. By integrating the dynamic acceleration, the velocity of the

blimp can be obtained, and by integrating the velocity, the position of the blimp can be obtained (see Chapter 6). The schematics of the accelerometers in the IMU are given in Appendix F.2.

4.3.2 Noise characteristics

The output of the triad of accelerometers was recorded for one hour while the sensors were at rest. Figure 4-3(a) shows the accelerometer data over time. The data was processed with the Allan variance technique. See Figure 4-3(b) for the Allan deviation as a function of window time for the three accelerometers. The noise densities of the accelerometers are listed in Table 4-1.

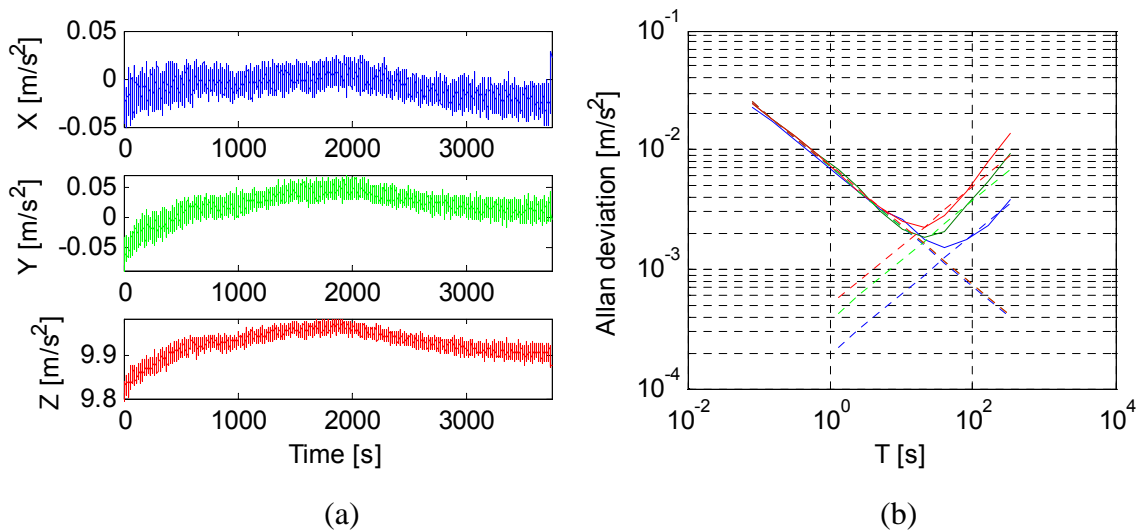


Figure 4-3: Accelerometer data over time (a) with the Allan variance (b).

Accelerometer	X	Y	Z	Simulations
Velocity random walk $[m/s^2/\sqrt{Hz}]$	7.06×10^{-3}	7.29×10^{-3}	7.31×10^{-3}	8.00×10^{-3}
Rate random walk $[m/s^2/s/\sqrt{Hz}]$	3.39×10^{-4}	6.62×10^{-4}	8.70×10^{-4}	8.00×10^{-4}

Table 4-1: Noise characteristics of the accelerometers.

4.3.3 Dynamic and static acceleration

An accelerometer measures specific force, which means it measures dynamic acceleration and gravity (static acceleration). The sensor's working can be viewed as a deflection of a beam due to accelerations exerted on the sensor. Without dynamic acceleration, the sensor will measure only

gravity. When the sensor is in free fall (that is accelerating 9.81m/s^2 towards earth if air resistance is ignored), the sensor will measure 0 m/s^2 due to the balance of static and dynamic acceleration.

If the sensor is calibrated with gravity (sensor measures $+9.81\text{ m/s}^2$ with its sensitive axis pointing to the middle of the earth), dynamic acceleration can be calculated as follows:

$$\mathbf{a}_{\text{dynamic}} = \mathbf{a}_{\text{static}} - \mathbf{a}_{\text{measured}} \quad (4.12)$$

The static acceleration in body axes can be estimated by converting the gravity vector in NED axes to body axes, as discussed in section 6.4.

4.3.4 Calibration

The easiest way to calibrate the triad of accelerometers is to assume the earth's gravity is constant to 9.81 m/s^2 . By rotating the IMU to different attitudes, the direction cosine matrix can be computed using equation (2.22). The DCM is used to transform the theoretical gravity vector $(0; 0; 9.81)\text{ m/s}^2$ in NED axes to body axes. By comparing the theoretical vector with measured vector and using equation (4.1), all parameters can be estimated if enough data is available. The IMU was rotated to 64 different angles and a least squares fit was done on the data to give an accurate calibration.

4.3.5 Simulink model

To generate accurate accelerometer data for use in simulations, a model of the accelerometer was implemented in Simulink. This model incorporates the static and dynamic accelerations. The output was corrupted with measurement noise and a random walk as defined in Table 4-1. It includes the same non-orthogonally as the practical sensors used, as well as A/D effects. See Figure 4-4 for an implementation of a Simulink model for the accelerometers.

4.3.6 Flight measurements

During a test flight of the blimp, data from the accelerometers indicated dynamic accelerations in the order of 2 m/s^2 during launch and recovery, but peaking at about 0.2 m/s^2 during free flight. This is shown in Figure 4-5 where the magnitude of the accelerometers is plotted together with the GPS height. The static acceleration of 9.81 m/s^2 can clearly be seen.

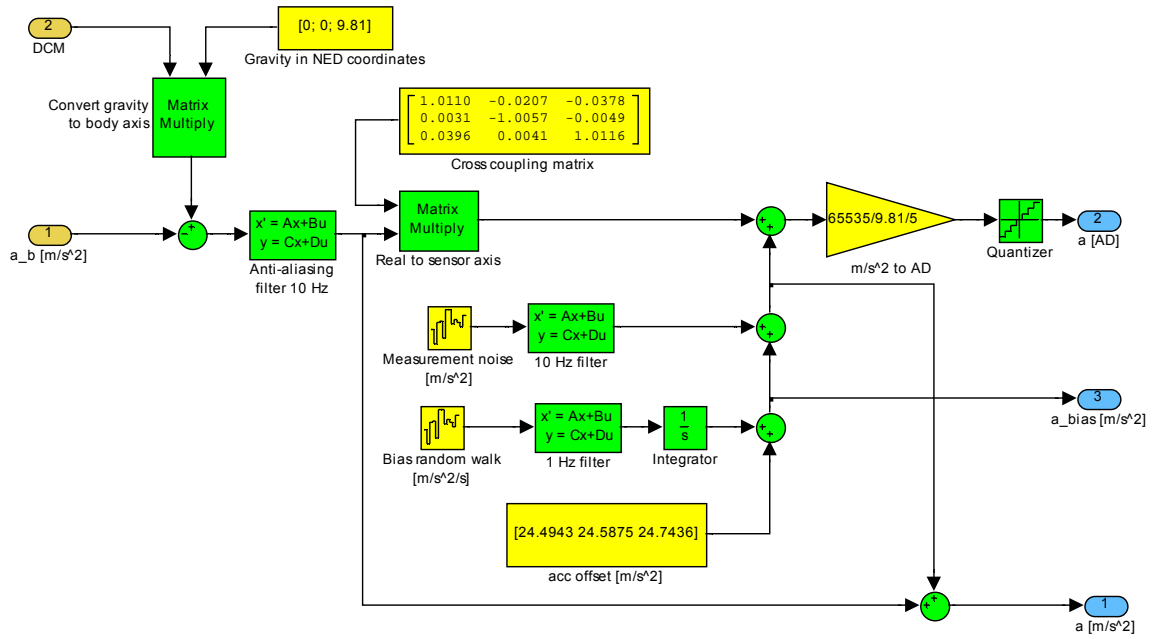


Figure 4-4: Simulink model for the accelerometers.

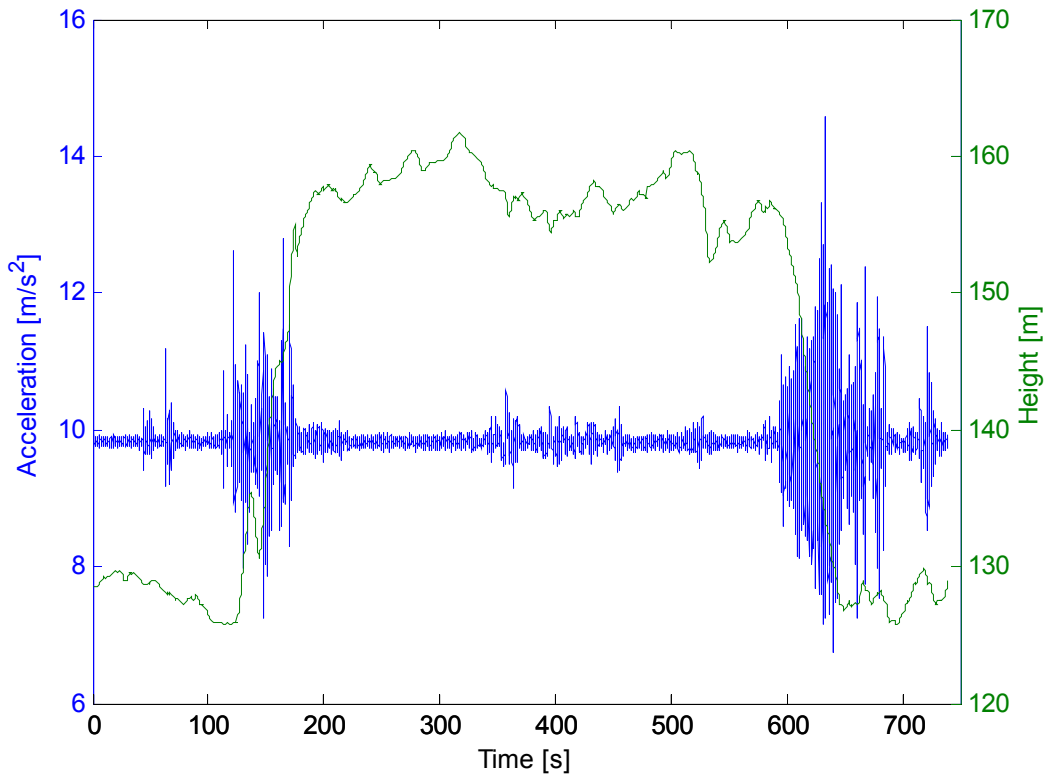


Figure 4-5: Magnitude of the accelerometer measurements during a test flight.

4.4 Gyroscope

4.4.1 Background

Barbour (1992) describes the history of the gyroscope. The spinning mass gyroscope in gimbals was developed around 1920 and used as a basic turn indicator for instrument flying. They were inexpensive, very rugged and reliable. Accuracy was in the order of 10 degrees per hour. However, these devices were heavy, consumed much power, and were restricted to certain operating regions due to mechanical constraints. Later the rate-integrating gyro was developed and was used extensively from the 1950's to 1980's.

A ring laser gyroscope (RLG) was first demonstrated in 1963. The RLG operates by setting up clockwise and counter clockwise resonant light beams reflected around a close cavity by mirrors. By detecting phase shifts between the two beams due to a rotation (called the Sagnac effect), the angular rates can be calculated. However, it was not until the 1980's that the RLG became commercially available and began to dominate the market.

The fibre optic gyro (FOG) works on the same principle as the RLG, but eliminates the need for mirrors by using fibre optics. It is an economic replacement for RLG as no precise machining or alignment is needed.

Micro electromechanical (MEMS) gyros are usually designed as an electronically driven resonator, often fabricated out of a single piece of quartz or silicone. The working of a MEMS gyro relies on the Coriolis force that causes displacements of the resonator when an angular rate is applied to the sensor. These displacements can be detected by a capacitive pickoff and demodulated to form the output of the sensor.

Comparing the FOG and MEMS gyro, the MEMS gyro offers a weaker performance, but is significantly cheaper and smaller. Analog Devices' popular ADXRS401 sensor was chosen due to the low cost, low weight and wide availability (Carstens 2005). It was used by Peddle (2005), Groenewald (2006) and Venter (2006) in the ESL. The ADXRS401 sensor measures up to $\pm 75^\circ/\text{s}$ angular rates. A triad of gyros is needed for a complete angular rate sensing in three axes.

The schematics of the gyros in the IMU are given in Appendix F.2.

4.4.2 Noise characteristics

The outputs of the triad of gyros were recorded for one hour while the sensors were at rest. The gyro data as a function of time is given in Figure 4-6(a). The data was processed with the Allan variance technique. See the Figure 4-6(b) for the Allan deviation as a function of window time for the three gyros. The noise densities of the gyros are listed in Table 4-2.

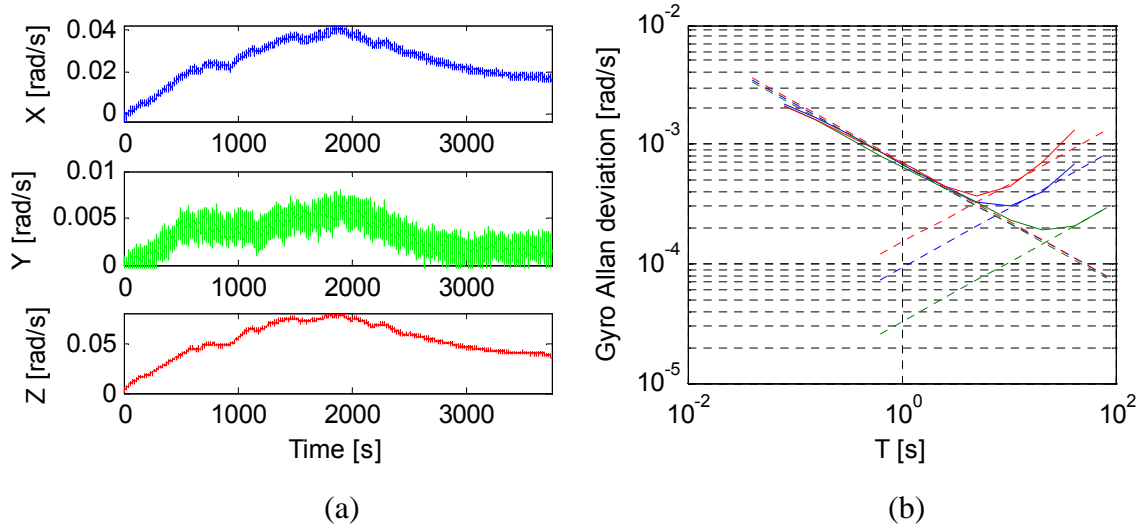


Figure 4-6: Gyro data over time (a) with the Allan variance (b).

Gyro	X	Y	Z	Simulation
Angle random walk $\left[\text{rad} / \text{s} / \sqrt{\text{Hz}} \right]$	7.01×10^{-4}	6.77×10^{-4}	7.12×10^{-4}	7.20×10^{-4}
Rate random walk $\left[\text{rad} / \text{s} / \text{s} / \sqrt{\text{Hz}} \right]$	1.60×10^{-4}	5.50×10^{-5}	2.56×10^{-4}	2.60×10^{-4}

Table 4-2: Noise characteristics of the gyros.

4.4.3 Calibration

There are two methods to calibrate the gyros:

1. Use a rate table and spin the IMU at different rotation rates. Fit the gyro output with the known rotation rates.
2. Rotate the IMU through different angles. By integrating the output of the gyros, the angles can be measured. Fit the measured angles with the known angles.

The integration method was used, as no working rate table was available. A least squares fit on the data was done to get an accurate calibration.

The bias of a MEMS type gyro is sensitive to accelerations, and that includes the earth's gravity. During calibration care must be taken that the sensor will be exposed to constant static accelerations, implying the sensor must be orthogonal to the earth's gravity during the rotation.

Figure 4-7 displays the Simulink model of the gyros to include all the noise, cross coupling and A/D effects.

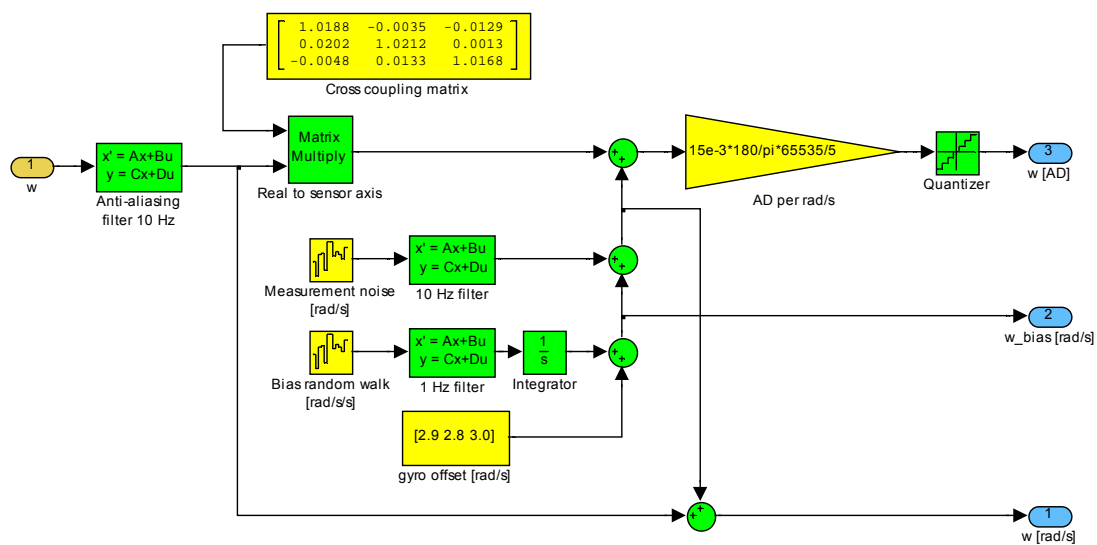


Figure 4-7: Simulink model for the gyros.

4.4.4 Flight measurements

During a test flight of the blimp, data from the gyros indicated turn rates of up to 80 degrees per second during launch and recovery, but staying most of the time below 20 degrees per second during free flight. The magnitude of the gyros together with the GPS height is displayed in Figure 4-8 as a function of time.

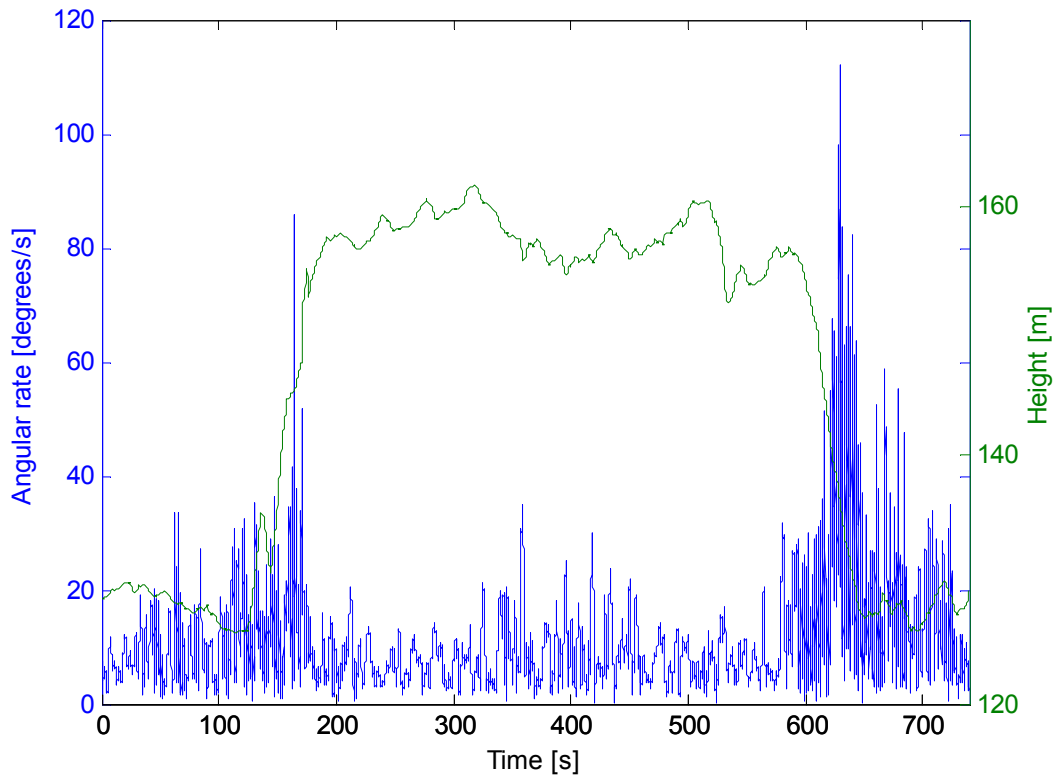


Figure 4-8: Magnitude of the gyro measurements during a test flight.

4.5 Magnetometer

4.5.1 Background

Magnetic sensors are devices that measure the magnitude of a magnetic field. As the magnetic field is a vector, the magnetometer must measure the field in three axes.

The earth has a magnetic field similar to a dipole. At the poles, the field magnitude is about 0.6 Gauss, and in the equatorial region about 0.3 Gauss. However, the dipole is only a crude approximation to the earth's magnetic field. The International Geomagnetic Reference Field (IGRF) is a more accurate model. This model consists of a series of spherical harmonic terms, each of which has a coefficient that determines the strength of that term. Since the geomagnetic field changes over time, these coefficients are updated every five years by a committee of the International Association of Geomagnetism and Aeronomy (Ripka 2001).

The earth's magnetic field is subjected to several variations. There are daily variations in the range of 0.0001 to 0.001 Gauss due to the ionisation of the atmosphere by solar radiation. Magnetic

storms due to solar flares may occur a few times per month with variations of up to 0.01 Gauss, depending on the severity of the flare. The earth's magnetic field is slowly changing in strength and position over the years.

In this thesis, the magnetic field is assumed constant, as the field remains the same for the small distances the blimp travels. See Table 4-3 for the magnetic field at Stellenbosch for 2004.

Direction	North	East	Down	Magnitude
Field [Gauss]	0.0969974	-0.0432305	-0.237753	0.260392

Table 4-3: Magnetic field at Stellenbosch for 2004 (from: Sindle 2005).

Three types of magnetometers can measure the earth's magnetic field:

- **Flux gate:** The flux gate magnetometer was developed in 1928 and is still in use today. The working depends on two coils wrapped around a common high-permeability ferromagnetic core. A drive signal is applied to one coil that causes the core to oscillate between saturation points. The other coil is used to sense the changes in the core's permeability, which can be demodulated to retrieve the magnetic field value. These devices tend to be bulky, not as rugged as smaller integrated sensors, and have a slow response time.
- **Magneto inductive:** The working of a magneto inductive magnetometer depends on a single winding coil on a ferromagnetic core that changes permeability within the earth's magnetic field. The coil is used as an inductor in a simple oscillator configuration. The oscillator's frequency is proportional to the magnetic field and can be used to determine the field values. These magnetometers are simple in design, inexpensive, and have low power requirements.
- **Anisotropic magneto resistive (AMR):** The magneto resistive effect is a change in resistance of a resistive perm alloy strip due to a magnetic field. Usually four of these resistors are connected in a Wheatstone bridge to measure the magnetic field. AMR sensors can be bulk manufactured on silicon wafers and mounted in commercial IC packages, offering high sensitivity, small size and fast response time.

Honeywell's HMC2003 three-axis magnetic sensor hybrid was chosen. It is an AMR sensor that measures the field in all three axes, making implementation very easy. It has been used by Carstens (2005), Groenewald (2006) and Venter (2006) in the ESL.

The HMC2003 can measure fields up to two Gauss. However, the earth's magnetic field is usually less than 0.6 Gauss, and even lower in Stellenbosch (0.26 Gauss). Therefore, the output of the magnetometer is amplified by a factor of four, to increase accuracy in the region of interest. The schematic of the amplifier appears in Appendix F.2.2.

4.5.2 Reset/set function

The HMC2003 has a set/reset function for its perm alloy film. The film is usually polarised in a certain direction, but a strong external magnetic field could flip the polarity, thus changing the sensor's characteristics. To restore the polarisation, a strong magnetic field must be applied.

In addition to the restoration of the polarisation, any offsets due to temperature, signal conditioning, etc. may be eliminated if the polarity is first reset (polarised in the opposite direction) and then set (polarised in the normal direction). This is because in a set/reset cycle, the polarity of the sensor is reversed, but the offsets associated with the sensor are not. Therefore, the offsets can be cancelled, giving an output proportional to the absolute field.

4.5.3 Noise characteristics

The outputs of the triad of magnetometers were recorded for one hour while the sensor was at rest. Only one reset/set cycle was done at the beginning of the period, which implies that there should be a bias drift. If the reset/set cycle were done continuously, any bias drifts would be eliminated. The magnetometer data as a function of time is displayed in Figure 4-9(a). The data was processed with the Allan variance technique. See Figure 4-9(b) for the Allan deviation as a function of window time for the magnetometer.

The noise densities of the magnetometer are listed in Table 4-4.

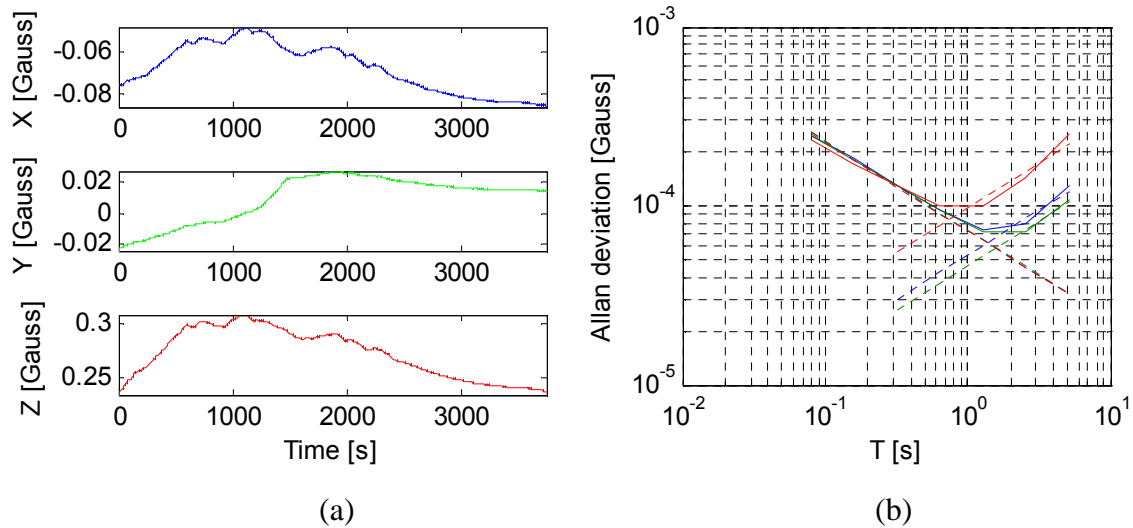


Figure 4-9: Magnetometer data over time (a) with the Allan variance (b).

Magnetometer axis	X	Y	Z	Simulation
Magnetic random noise $\left[Gauss / \sqrt{Hz} \right]$	7.29×10^{-5}	7.30×10^{-5}	7.22×10^{-5}	7.50×10^{-5}
Rate random walk $\left[Gauss / s / \sqrt{Hz} \right]$	9.22×10^{-5}	8.04×10^{-5}	1.68×10^{-4}	1.50×10^{-4}

Table 4-4: Noise characteristics of the magnetometer.

4.5.4 Disturbances

The earth's magnetic field can locally be deformed by three sources:

- **Hard iron:** Hard irons are ferromagnetic materials with permanent magnetic fields. These unwanted fields are superimposed on the magnetometer reading, resulting in offsets in all axes. If these unwanted fields are not time varying, the constant offsets can easily be removed in software. Figure 4-10(a) shows a 2-D representation of hard iron effects. Hard iron offsets are not cancelled by a reset / set action, as the offset is part of the magnetic field.
- **Soft iron:** Soft iron materials are materials that generate magnetic fields in response to the earth's magnetic field. The generated magnetic field is depending on the magnitude and direction of the earth's magnetic field, which means that the soft iron effects will depend on the attitude of the sensor. There can be hysteresis, but usually the hysteresis is small enough to be ignored. To compensate for soft iron effects without hysteresis in 3-D, a 3x3 matrix is used. Figure 4-10(b) illustrates soft iron effects in a 2-D case.

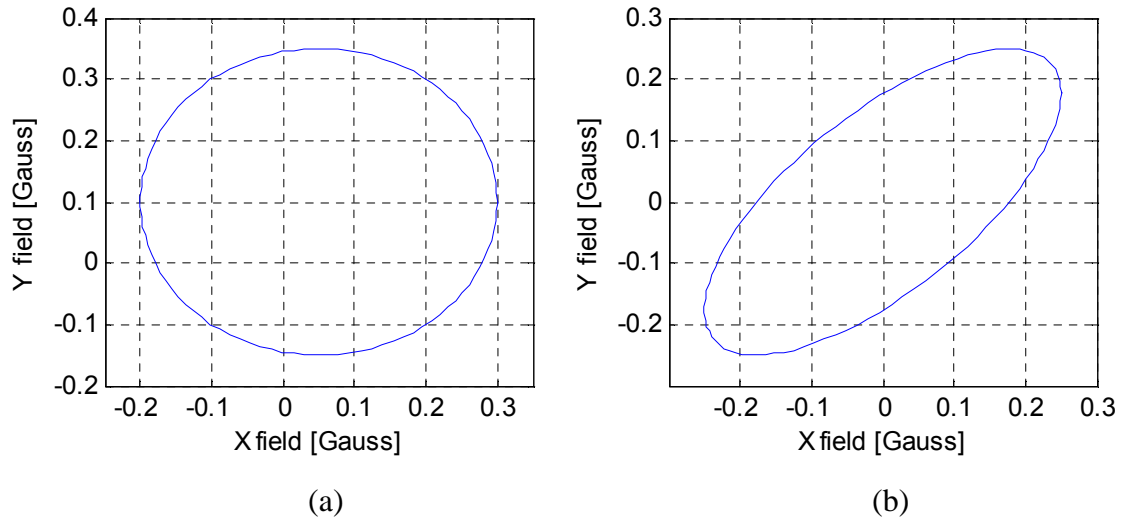


Figure 4-10: Illustration of hard iron (a) and soft iron (b) effects on the magnetometer.

- **Currents flowing:** A magnetic field is set up around a conductor carrying a current. A steady current flow will result in magnetic offsets, which can be compensated for if the magnetometer is calibrated with all electronics running on board.

To keep the disturbance effects as small as possible, the magnetometer must be mounted far away from ferromagnetic materials and electronic circuits.

4.5.5 Calibration

Instead of an absolute calibration, a relative calibration method was used as described by Gebre-Egziabher *et al.* (2001). The problem with an absolute calibration is that the magnetometer must be positioned at known angles with a calibration table. The ferrous effects of the calibration table will disturb the earth's magnetic field, resulting in a calibration only valid near the calibration table. For use on a blimp, the calibration must be done in a ferromagnetic free space.

With a relative calibration method, no accurate attitude information is needed. The working relies on the principle that all the magnetometer readings must lie on a 3D-sphere with a magnitude equal to the earth's magnetic field strength at that location.

Rewriting equation (4.2) in a simpler form:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} X_{sensor} \\ Y_{sensor} \\ Z_{sensor} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad (4.13)$$

The radius of the sphere can then be calculated as follows:

$$\begin{aligned} R^2 = & (a_1 X_{sensor} + b_1 Y_{sensor} + c_1 Z_{sensor} + d_1)^2 + \dots \\ & (a_2 X_{sensor} + b_2 Y_{sensor} + c_2 Z_{sensor} + d_2)^2 + \dots \\ & (a_3 X_{sensor} + b_3 Y_{sensor} + c_3 Z_{sensor} + d_3)^2 \end{aligned} \quad (4.14)$$

Setting the magnitude of the magnetic field as defined in Table 4-3, a radius error could be calculated for each sampling period. If all the batch errors are added, and a minimum seeking algorithm is applied to the batch data by changing the matrix elements and offset vector, all the parameters can be identified if there are enough data available. Note that the calibration must be done with all hardware running onboard to include all the effects of hard and soft iron. For a good calibration, the IMU must be rotated so that the magnetometer samples are distributed all over the sphere.

The minimum seeking algorithm was done with Microsoft Excel's solver add-in. See Figure 4-11 for calibrated magnetometer samples in a 3-D scatter plot. Clearly, the magnetometer samples lie on a 3-D sphere.

To test the calibration, place the IMU with its axes aligned with the NED axial system. The measured magnetic vector can be compared to the reference vector as defined in Table 4-3. The alignment process is usually not very accurate, so relative small differences may occur. Table 4-5 gives a comparison between the reference and measured magnetic vector.

Field [Gauss]	North	East	Down
Reference	0.0969974	-0.0432305	-0.237753
Measured	0.096627	-0.046761	-0.238267

Table 4-5: Comparison between the reference and measured magnetic vector.

A Simulink model was developed to include all the noise, hard iron, soft iron and A/D effects of the magnetometer. The Simulink model appears in Figure 4-12.

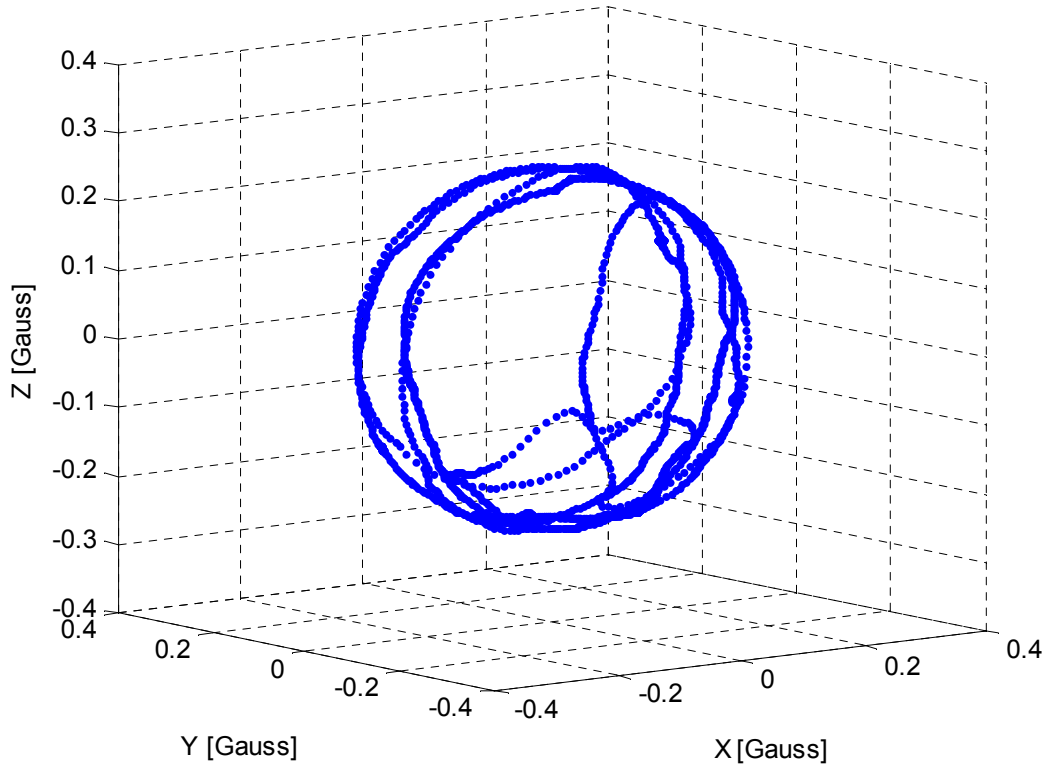


Figure 4-11: Magnetometer calibration results.

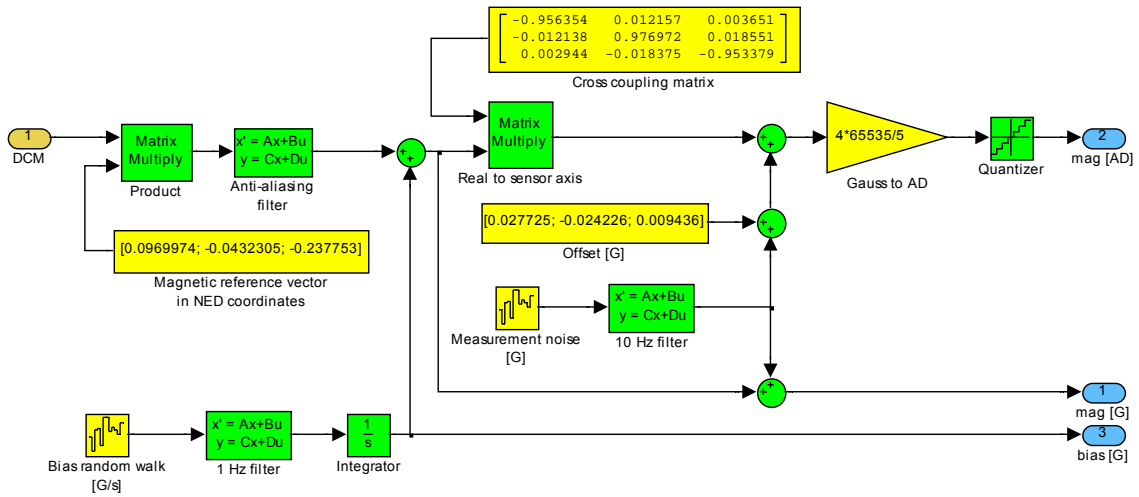


Figure 4-12: Simulink model of the magnetometer.

4.6 IMU

The IMU module was designed to measure gyro, accelerometer and magnetometer readings in all three axes³. All the analogue sensor signals are sampled with the CANsense board as developed by Groenewald (2006). The discrete data is transmitted at a fixed rate over an RS232 port to be used by the OBC for a discrete Kalman filter implementation.

4.7 GPS receiver

A GPS receiver decodes the signals from the Global Navigation Satellite System (GNSS) to determine the receiver's position and velocity. The working depends on the time it takes for a signal to propagate from a satellite to the receiver, which will be proportional to the path length. If at least four satellite signals are received, and the satellite positions are known, the position and time of the GPS receiver can be calculated (El-Rabbany 2002). The velocity of the receiver can be determined by measuring the carrier frequency Doppler shift, or by computing the change in position over time.

There are a wide range of commercial GPS receivers available, offering different accuracies and update rates. However, as accuracies and update rates increase, so does the cost. As low cost was a requirement, a high accuracy GPS receiver could not be used. Instead, a uBlox Antaris receiver was used. This was the only receiver in its price class to offer a 4 Hz update rate. Additionally, it supports differential GPS (DGPS), can track up to 16 satellites, and is small. It has also been used by Carstens (2005), Peddle (2005), Groenewald (2006) and Venter (2006) in the ESL.

4.7.1 Protocol

The uBlox GPS receiver offers two protocols, namely NMEA and UBX. NMEA is an acronym for the National Marine Electronics Association, while UBX is an uBlox proprietary format. Table 4-6 compares the two protocols.

It was decided to use the UBX protocol, because NMEA does not output NED velocities as needed for the Kalman filter as defined in Chapter 6. In addition, UBX gives more accurate sigma values on the position and velocity estimates. The GPS module was set up to send the following UBX navigation packets at 4 Hz: POSLLH (longitude, latitude and height), STATUS (type of GPS fix) and VELNED (velocities in NED axes).

³ See Appendix F.3 for more detail on the hardware that was developed.

NMEA	UBX
ASCII-based industry standard protocol	Propriety protocol optimized for uBlox
Compatible to other GPS receivers	Provides a wider range of parameters
ASCII protocol, easy to use	Binary protocol
Limited information available	Wide range of receiver information
Limited configuration available	Direct access to configuration settings
Low density data	High density data

Table 4-6: Comparison between NMEA and UBX protocols.

4.7.2 Differential GPS

Dedicated GPS reference stations are placed all over the world. They can be seen as GPS receivers working in reverse (El-Rabbany 2002). Usually with GPS receivers the position is unknown, and with the help of pseudo range measurements to different GPS satellites the position is determined. For a GPS reference station the position is known and the pseudo range to GPS satellites are calculated. The difference between the theoretical pseudo range and measured pseudo range is due to a range of unknowns, for instance atmospheric delay effects. The error between the theoretical and measured pseudo range is put in a RTCM (Radio Technical Commission for Maritime Services Commission) format and broadcast. It was originally developed for radio transmission, and any GPS receiver near the reference station could receive and decode this information in order to get a more accurate position fix.

With the advance of the internet, some stations broadcast the same RTCM information stream via the internet, called Networked Transport of RTCM via Internet Protocol (NTRIP). With the correct RTCM receiver software, the RTCM data can be received over the internet and relayed to the GPS receiver for processing.

Software to receive NTRIP data can be downloaded from http://igs.ifag.de/ntrip/ntrip_down.htm. A list of GPS broadcasters can be accessed at <http://www.rtcn-ntrip.org/home>. At the time of writing two GPS reference stations in Cape Town were available providing real time RTCM data.

4.7.3 Noise characteristics

The noise on the position and velocity measurements are a function of a variety of factors, including the number of satellites used for a solution fix, the geometry of the satellite constellation and signal

to noise ratio of the GNSS signals. It means that the accuracy of the measurements may vary over time.

The UBX protocol gives the standard deviation of the estimated error on all states. These values were used to construct the measurement noise covariance matrix for a Kalman filter implementation.

4.7.4 Delay

The uBlox receiver has a varying delay on its position and speed updates (Carstens 2005). This delay is important for high velocity vehicles. However, for the application of a blimp where the velocities are low, this delay can be ignored.

4.7.5 Simulink model

The displacement and velocities of the blimp is simulated in NED axes. However, the GPS position is given in longitude, latitude and height (LLH). To add the movement in NED to LLH, the initial position in LLH is converted to ECEF rectangular format using equations (2.4) to (2.6). The displacement vector in NED coordinates is converted to rectangular format using equation (2.9) and added to the initial position in ECEF rectangular format. The answer in ECEF rectangular format is converted back to LLH using equations (2.1) to (2.3). Constant noise figures were used in the simulation, typical to what the GPS reported in a test flight. The standard deviation for horizontal position was 3m, 4m for vertical position and 0.5 m/s for velocity measurements. The Simulink model of the GPS is given in Figure 4-13. The values are scaled and converted to integer values to simulate the output of the GPS.

4.7.6 Flight measurements

During a test flight of the blimp, data from the GPS indicated that the blimp speed usually stays below 1 m/s. In addition, the blimp oscillates with a period of 20 to 25 seconds. A plot of the North, East and Down velocities versus time is given in Figure 4-14. The height of the blimp is included for reference. Note that the Down velocity gives a good indication when the blimp is gaining or losing altitude.

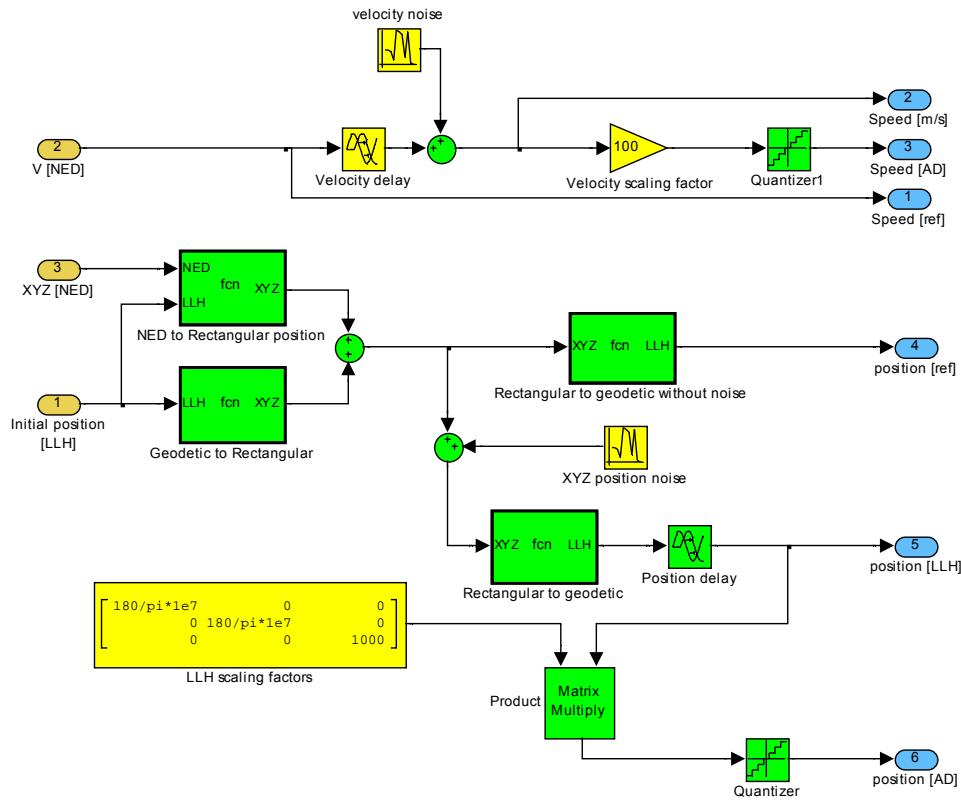


Figure 4-13: Simulink model of the GPS.

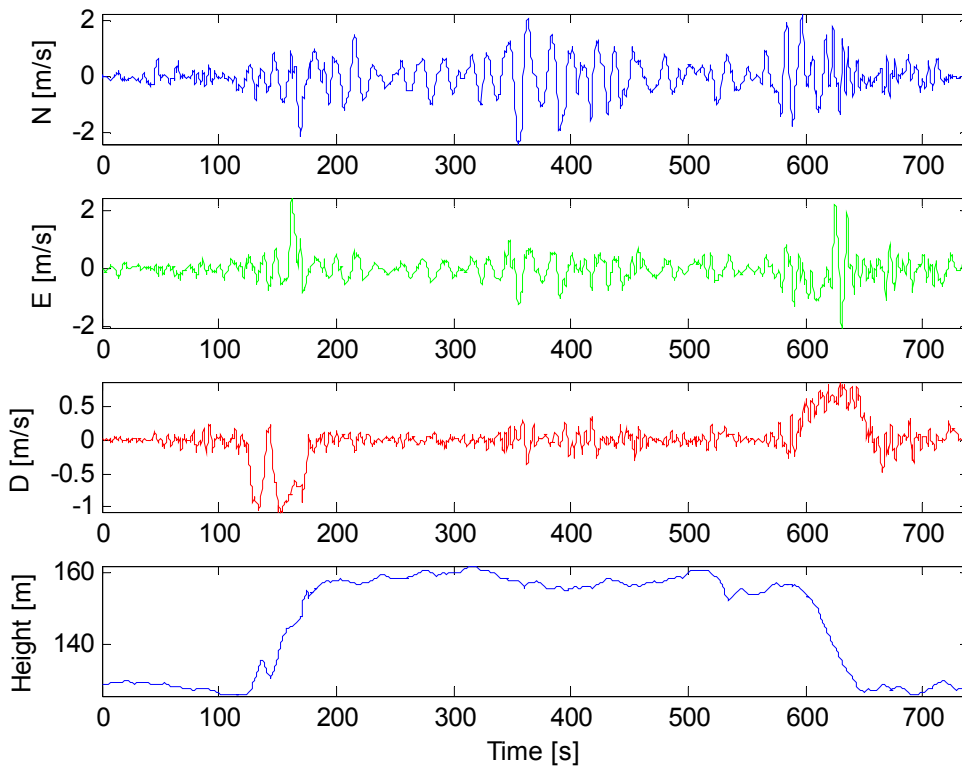


Figure 4-14: GPS velocities during a test flight.

4.8 Simulations

To simulate the various sensor measurements on board the blimp, the blimp was treated as a point mass. The motion of the point mass was simulated with a standard six degrees of freedom block. Inputs to the block are forces and torques, and the block gives accelerations, velocities, angular accelerations, angular rates and angles (attitude) as output. Figure 4-15 gives the Simulink model of all the sensor models combined in one simulation.

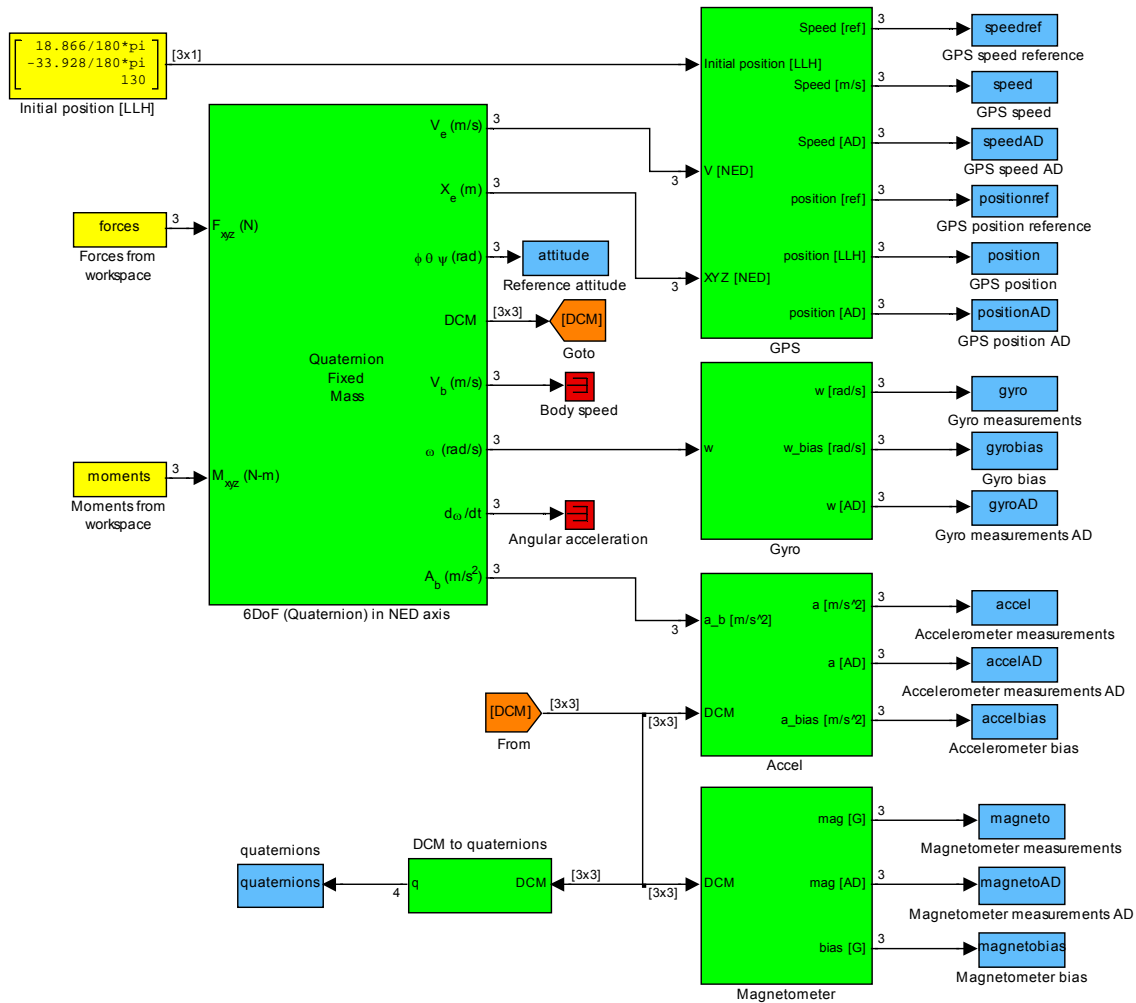


Figure 4-15: Simulink model of the motion of the blimp and the associated sensor outputs.

The inputs to the block were set up to represent typical blimp movement. The forces were set up so that the blimp had NED sinusoidal velocities with a period of 20 to 25 seconds and magnitudes below 1 m/s as shown in Figure 4-16.

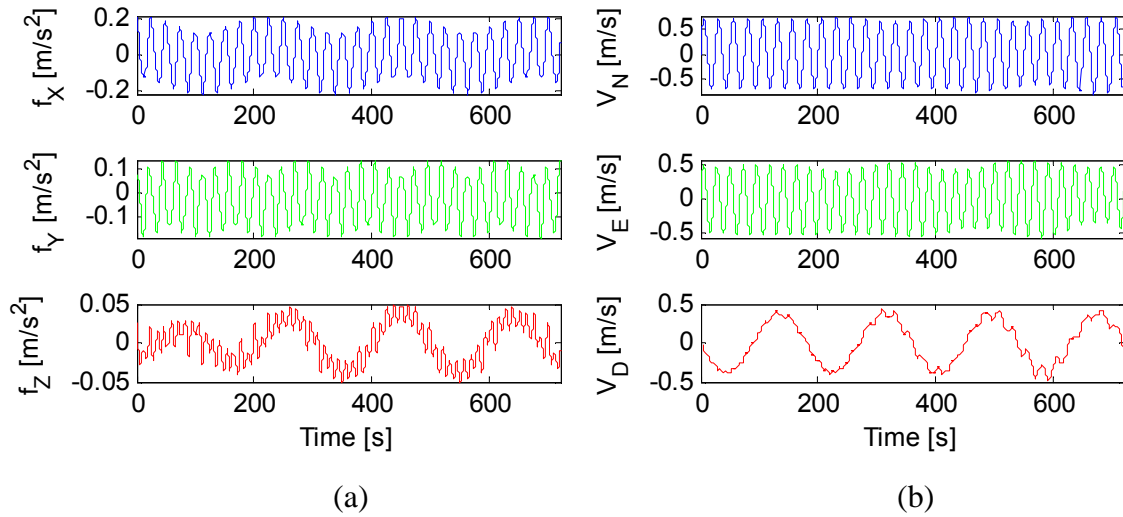


Figure 4-16: Simulated body accelerations (a) and resulting NED velocities of the blimp (b).

The torques were set up so that the blimp rolls and pitches up to 10 degrees, and yaws up to 50 degrees with a period of 20 to 25 seconds. See Figure 4-17 for the simulated attitude of the blimp in terms of roll, pitch and yaw angles.

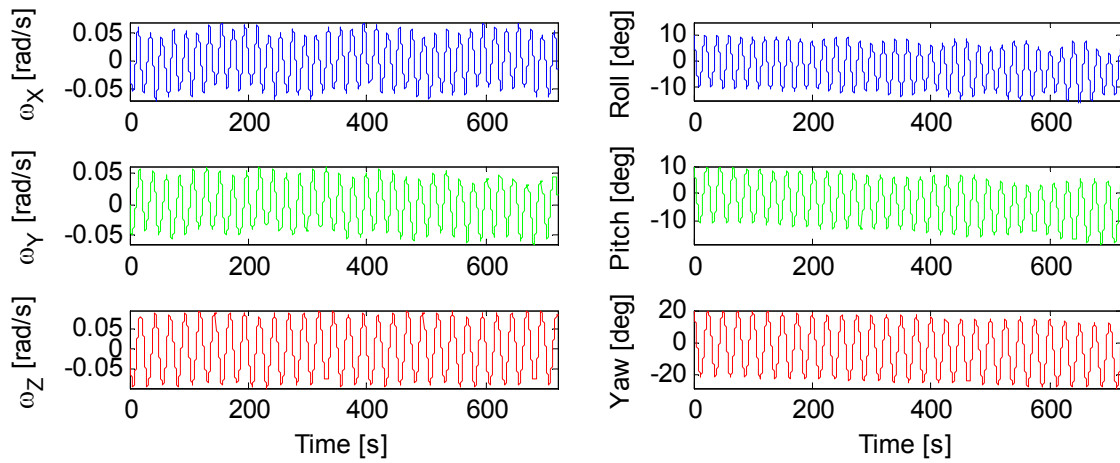


Figure 4-17: Simulated body rotation rates (a) and resulting attitude of the blimp (b).

The simulated data was exported to the Matlab workspace. Various Kalman filter simulations were done with this data as described in Chapters 5 and 6, and the Kalman filter estimates were compared to the reference values in order to evaluate the performance of the different Kalman filters.

4.9 Onboard computer

A standardised form-factor for onboard computers is the PC/104 form-factor. The idea behind PC/104 is to provide standard computer hardware (the 80x86 processor range from IBM) for an embedded system. An ISA bus is provided in an interface of 104 pins, hence the name. Different modules can be stacked together on the bus to expand the system. Usually the PC/104 system is made of low-power components to reduce the power requirements.

As the architecture of a PC/104 computer is the same as desktop computers, normal program development tools can be used to compile a program for use on a PC/104 system. In fact, it is possible to run an operating system usually used on desktop computers on a PC/104 system.

Because Kalman filtering requires a lot of floating point processing power, and at least three RS232 ports are needed to interface with all the hardware, the Prometheus-LC from Diamond Systems was chosen. It is a low power 80486 (which includes a numeric coprocessor) running at a clock frequency of 100 MHz, but consuming only 2.5 Watts of power. It has 16 Mb of onboard RAM, together with 1 Mb flash memory for program storage. It has four RS232 ports where three are used to connect to the RF link, IMU, and GPS. Figure 7-1 shows the final system configuration of the solution.

4.10 RF Link

The radio frequency (RF) link was used to establish a data link between a ground station and the blimp's OBC. The data link is used to command the OBC and receive real time information from the blimp.

HandyWave's HandyPort provides an economic and efficient RF link using RS232 ports. It uses a 2.4 GHz frequency hopping spread spectrum modulation and can reach up to 100 m with the dipole antenna, or up to 500 m with a patch antenna. Baud rates of up to 115.2 kbps can be used.

4.11 Power budget

For an operational AHRS, the system needs about 5 W of power. Table 4-7 gives the power budget analysis.

Device	Voltage [V]	Current [mA]	Power [W]
OBC	5.0	500	2.500
GPS	5.0	120	0.600
IMU	6.5	230	1.495
RF Link	6.5	50	0.325
Total			4.920

Table 4-7: Power budget analysis of the AHRS.

A Lithium ion polymer (Li-Po) battery was chosen because of the high energy density. A battery from Kokam was used, which has three Li-Po batteries in series, giving an output of 11.1 V. It has a capacity of 1200 mAh and weighs only 80 g. When running the system for one hour, 50% of the capacity of the batteries would have been used. It is crucial that the Li-Po battery does not discharge completely, as this would damage the battery.

DC-to-DC converters are used to keep the system as efficient as possible. With the losses in the DC-to-DC converters included, the system needs about 6 W of power when running from an 11.1 V supply. This loss of 1 W compares favourably to a loss of about 5 W if the system is run from linear regulators.

See Appendix F.4 for the schematics of the power regulation circuits.

4.12 AHRS cost breakdown

The cost breakdown of the AHRS is given in Table 4-8. The value is given in South African Rands (ZAR). The cost of the blimp and helium used for test flights are excluded.

Component		Cost [ZAR]
OBC		3493
IMU		3639
Gyros	778	
Accelerometers	345	
Magnetometer	1920	
PCB	596	
RF Link		1508
Battery		400
GPS		595
Module	430	
Antenna	165	
Total		9635

Table 4-8: Cost breakdown of the AHRS.

Chapter 5: Attitude determination

Originally used to estimate the attitude of a satellite, Wahba (1965) posed a problem to find the best transformation matrix between two axial systems, where one axial system is the satellite body axial system, and the other axial system a reference axial system. The transformation matrix can be determined by transforming vectors in the reference axial system to the body axial system and comparing it with the measured vectors in the body axial system. More than one vector is needed, because a vector only contains two parameters when it is constrained by a certain magnitude. As attitude is determined by three independent parameters, at least two vectors are needed (Psiaki 1990). The error between the measured vectors and transformed reference vectors must be minimised to find a least-squares estimate of the transformation matrix.

Shuster and Oh (1981) describe two solutions to Wahba's problem: the deterministic TRIAD algorithm and the Quaternion Estimator (QUEST) algorithm, which is an optimal TRIAD variation in terms of a quaternion. These methods are discussed in section 5.2 and 5.3.

Next, Kalman filters were used to estimate the quaternion (Lefferts *et al.* 1982). Instead of estimating all four quaternion elements, which violates the independence requirement due to the unit norm constraint, they proposed a new method where only the quaternion vector part is estimated. This method was also used in the final solution as described in section 5.4.

Kalman filters have been widely applied to spacecraft attitude determination (Psiaki 1990). On a spacecraft, there are multitudes of sensors that can be used: sun sensors, star sensors, earth sensors, and magnetometers. On a blimp the amount of sensors is limited. However, the blimp experiences small accelerations compared to gravity (see section 4.3) when floating in the air. It means that the accelerometer output can be used as a gravity vector. Together with the magnetometer vector, the attitude of the blimp can be determined.

More recently, it has been shown that the attitude of a body can be estimated by using GPS signals only (Lightsey *et al.* 2003). However, carrier-phase and signal strengths information of the different satellite signals are needed, and therefore an expensive GPS receiver must be used. A standard GPS receiver was used and the attitude was determined by conventional methods only.

This chapter will focus on different methods to determine attitude. The various methods eventually lead to the Multiplicative Extended Kalman Filter (MEKF) used for accurate attitude estimation.

5.1 Gyro integration

5.1.1 Definition

The gyro integration method uses data from the gyros only to calculate the attitude of the body over time. From Chapter 2, the dynamic equation of the quaternion in terms of angular rates is:

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\omega} \otimes \mathbf{q} \quad (2.51)$$

When the initial attitude is known, the dynamics of the quaternion can be propagated by numerical integration as follows:

$$\mathbf{q}(t + T_s) = \mathbf{q}(t) + \frac{T_s}{2} \boldsymbol{\omega} \otimes \mathbf{q} \quad (5.1)$$

Because the numerical integration process does not necessarily keep the unity norm constraint, the updated quaternion must be normalised after each time step:

$$\mathbf{q}(t + T_s) = \frac{\mathbf{q}(t) + \frac{T_s}{2} \boldsymbol{\omega} \otimes \mathbf{q}}{\left| \mathbf{q}(t) + \frac{T_s}{2} \boldsymbol{\omega} \otimes \mathbf{q} \right|} \quad (5.2)$$

Alternatively, a closed-form solution of equation (2.51) can be used. Wertz (1978) gives an approximate closed-form solution in the case where the gyro data is sampled at a fixed rate and the angular velocity vector is constant over the sampling period:

$$\mathbf{q}(t + T_s) = \left(\cos\left(\frac{|\boldsymbol{\omega}| T_s}{2}\right) \mathbf{I}_{4 \times 4} + \frac{\sin\left(\frac{|\boldsymbol{\omega}| T_s}{2}\right)}{|\boldsymbol{\omega}|} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \right) \mathbf{q}(t) \quad (5.3)$$

5.1.2 Simulation

Using the Simulink simulation data (see section 4.8), the attitude of the airship was calculated by using equations (5.2) and (5.3). The two methods differed only by an RMS error of 0.1 degree. For a fast update rate, the more complex calculation of equation (5.3) had no obvious advantages. Only equation (5.2) was used further.

In the first case, gyro data with no bias drifts was used. Figure 5-1 shows the attitude estimation errors over time. The noise on the gyro data results in an angle random walk (ARW). The error statistics are listed in Table 5-1.

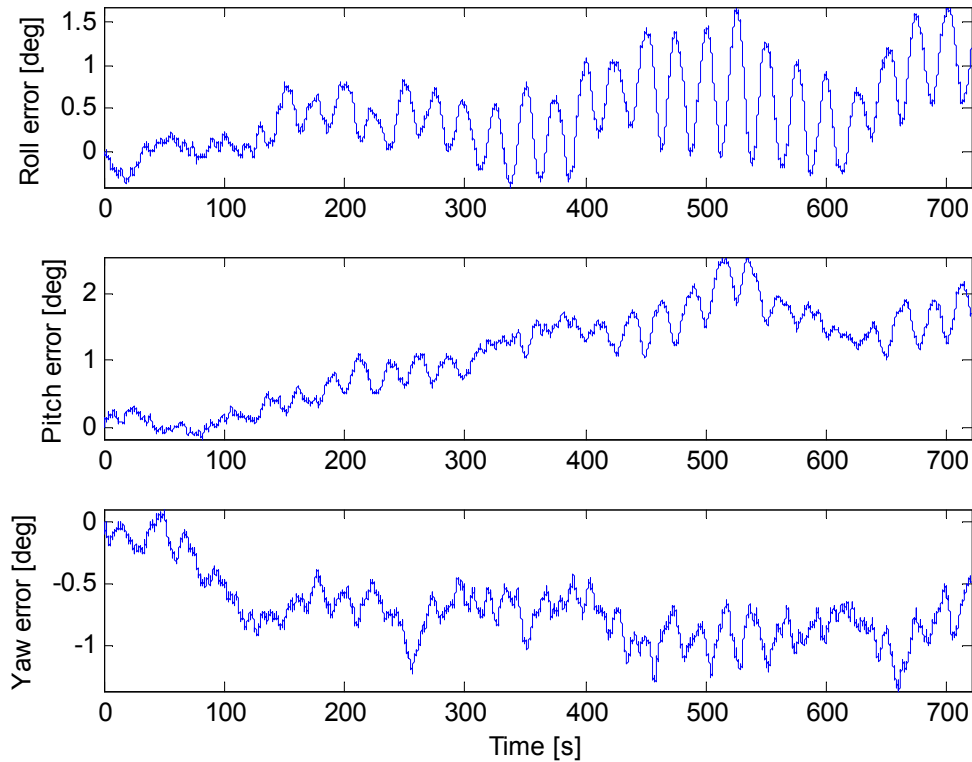


Figure 5-1: Attitude errors for gyro integration without bias drifts.

	Roll [deg]	Pitch [deg]	Yaw [deg]
Max	1.6717	2.5362	1.3666
Standard deviation	0.4615	0.6624	0.2702
RMS error	0.6631	1.2833	0.7606

Table 5-1: Attitude error statistics for gyro integration without bias drifts.

When the gyro data contains noise combined with a random walk (as a typical MEMS gyro has), the attitude errors become much larger. This is shown in Figure 5-2 where the attitude estimation errors are small in the beginning, but reach unacceptable levels after some time when there have been significant bias drifts. The bias drifts result in incorrect angular rates, and subsequently huge errors in attitude estimation. The error statistics are listed in Table 5-2.

A practical implication of the results above is that accurate attitude determination cannot rely on gyro data alone.

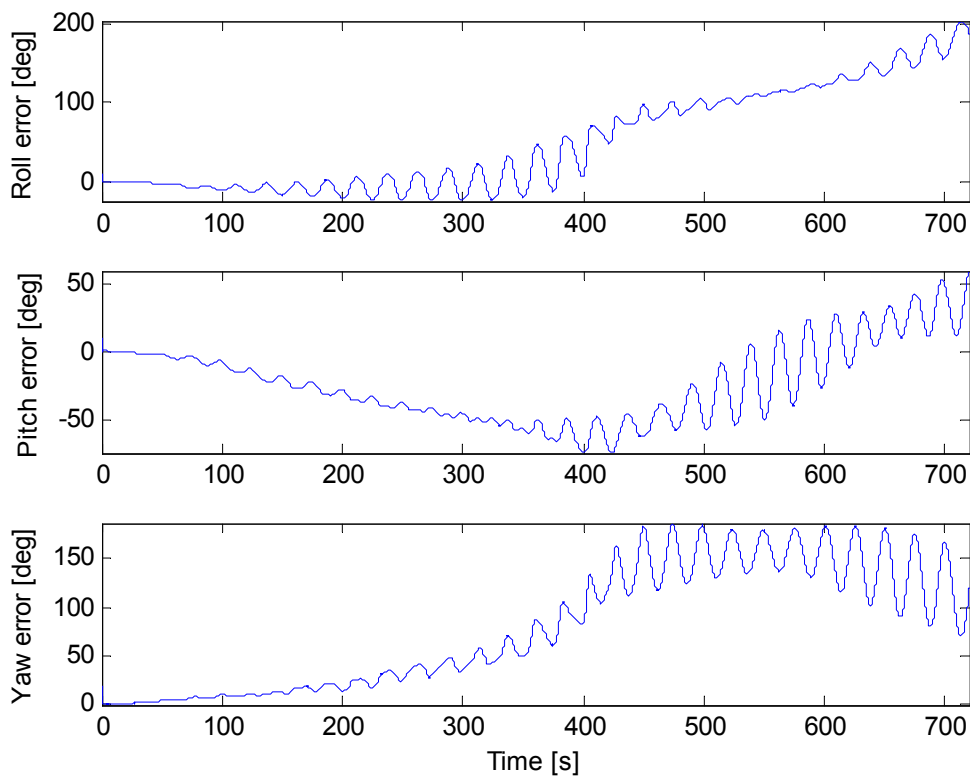


Figure 5-2: Attitude errors for gyro integration with bias drifts.

	Roll [deg]	Pitch [deg]	Yaw [deg]
Max	203.3206	74.9388	186.2271
Standard deviation	64.5810	29.0450	62.4046
RMS error	82.1961	37.4093	100.6837

Table 5-2: Attitude error statistics for gyro integration with bias drifts.

5.2 TRIAD

5.2.1 Definition

The TRIAD algorithm is a deterministic way to find the transformation matrix from a reference axial system to the body axial system. The TRIAD algorithm uses only two vectors, where each vector has a reference vector in reference axes ($\bar{\mathbf{v}}^r$) and a measurement vector in body axes ($\bar{\mathbf{v}}^b$). These two are related through a transformation matrix as follows:

$$\bar{\mathbf{v}}^b = \mathbf{A}\bar{\mathbf{v}}^r \quad (5.4)$$

Because two vectors have four independent parameters, and only three are needed for attitude determination, one parameter must be discarded. This is done by constructing three orthonormal vectors, called a triad. The most accurate vector is taken as the base vector of the triad. The second vector of the triad is constructed orthonormal to the two measurement vectors, and the third vector of the triad is generated orthonormal to the other two vectors of the triad. In effect, one parameter of the second measurement vector is discarded, and therefore the second measurement vector must be the less accurate vector. The triad vectors must be unit vectors. Writing it in mathematical terms:

$$\bar{\mathbf{u}}_1 = \frac{\bar{\mathbf{v}}_1}{|\bar{\mathbf{v}}_1|} \quad (5.5)$$

$$\bar{\mathbf{u}}_2 = \frac{\bar{\mathbf{v}}_1 \times \bar{\mathbf{v}}_2}{|\bar{\mathbf{v}}_1 \times \bar{\mathbf{v}}_2|} \quad (5.6)$$

$$\bar{\mathbf{u}}_3 = \bar{\mathbf{u}}_1 \times \bar{\mathbf{u}}_2 \quad (5.7)$$

The triad vectors must be constructed for both the reference and measurement vectors. Packing the vectors in a matrix, one matrix can be seen as a transformation matrix from triad to reference axes, while the other matrix can be seen as a transformation matrix from triad to body axes:

$$\mathbf{T}_{rt} = \begin{bmatrix} \bar{\mathbf{u}}_1^r & \bar{\mathbf{u}}_2^r & \bar{\mathbf{u}}_3^r \end{bmatrix} \quad (5.8)$$

$$\mathbf{T}_{bt} = \begin{bmatrix} \bar{\mathbf{u}}_1^b & \bar{\mathbf{u}}_2^r & \bar{\mathbf{u}}_3^r \end{bmatrix} \quad (5.9)$$

To calculate the transformation matrix from reference to body axes:

$$\mathbf{A} = \mathbf{T}_{br} = \mathbf{T}_{bt} \mathbf{T}_{tr} = \mathbf{T}_{bt} \mathbf{T}_{rt}^{-1} \quad (5.10)$$

However, due to the orthonormality of the transformation matrices, the inverse of the matrix is equal to the transpose of the matrix. Therefore, equation (5.10) simplifies to:

$$\begin{aligned} \mathbf{A} &= \mathbf{T}_{bt} \mathbf{T}_{rt}^T \\ &= \begin{bmatrix} \bar{\mathbf{u}}_1^b & \bar{\mathbf{u}}_2^b & \bar{\mathbf{u}}_3^b \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_1^r & \bar{\mathbf{u}}_2^r & \bar{\mathbf{u}}_3^r \end{bmatrix}^T \end{aligned} \quad (5.11)$$

5.2.2 Simulation

Using the Simulink data (see section 4.8) the attitude of the airship was calculated using the accelerometer and magnetometer vectors. As the magnetometer has relative big bias drifts, the accelerometer vector was taken as most accurate vector. Figure 5-3 is a representation of the attitude estimation errors in degrees when using the TRIAD method, while Table 5-3 shows the attitude estimation error statistics.

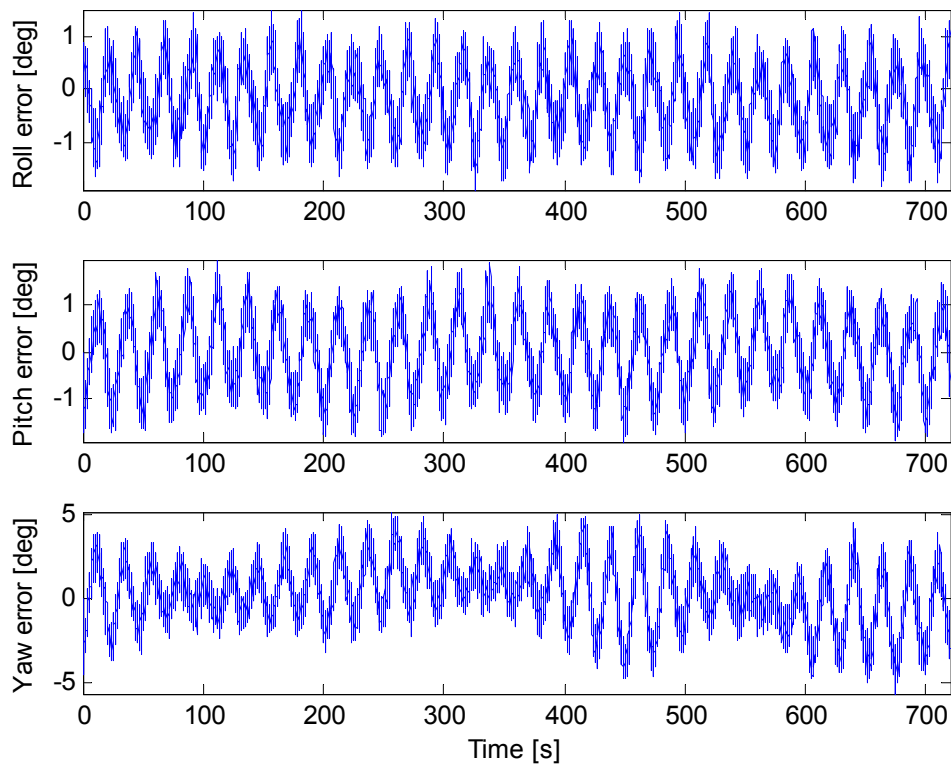


Figure 5-3: TRIAD attitude estimation errors.

	Roll [deg]	Pitch [deg]	Yaw [deg]
Max	1.9041	1.9641	5.6333
Standard deviation	0.6305	0.7502	1.7868
RMS error	0.6582	0.7502	1.8136

Table 5-3: TRIAD attitude estimation error statistics.

When compared to the gyro integration method, the TRIAD method shows a significant increase in accuracy over time. The biggest estimation error is made on the yaw angle, which can be explained by the fact that both the gravity and magnetometer vectors have big downward components. When the blimp is yawing around the downward axis, the measured vectors will not change as much as when the blimp is rolling or pitching, resulting in bigger estimation errors on the yaw angle.

5.3 QUEST

5.3.1 Definition

The Quest algorithm uses a statistical approach to minimise the overall error when estimating the transformation matrix from reference to body axes, as stated in equation (5.4). The loss function is defined as:

$$J(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^N \frac{1}{\sigma_i^2} \left| \bar{\mathbf{v}}_i^{\mathbf{b}} - \mathbf{A} \bar{\mathbf{v}}_i^{\mathbf{r}} \right|^2 \quad (5.12)$$

This approach uses all the information of the N vector measurements to estimate the transformation matrix. Each vector is assumed to have zero mean white Gaussian noise with variance σ^2 . For a perfect transformation matrix, the loss function would be equal to zero. However, due to noise and other imperfections, the loss function must be minimised to get a best estimate of the transformation matrix.

Equation (5.12) can be rewritten as:

$$J(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^N \frac{1}{\sigma_i^2} \left(\bar{\mathbf{v}}_i^{\mathbf{b}} - \mathbf{A} \bar{\mathbf{v}}_i^{\mathbf{r}} \right)^{\mathbf{T}} \left(\bar{\mathbf{v}}_i^{\mathbf{b}} - \mathbf{A} \bar{\mathbf{v}}_i^{\mathbf{r}} \right) = \frac{1}{2} \sum_{i=1}^N \frac{1}{\sigma_i^2} \left(\bar{\mathbf{v}}_i^{\mathbf{b}\mathbf{T}} \bar{\mathbf{v}}_i^{\mathbf{b}} + \bar{\mathbf{v}}_i^{\mathbf{r}\mathbf{T}} \bar{\mathbf{v}}_i^{\mathbf{r}} - 2 \bar{\mathbf{v}}_i^{\mathbf{b}\mathbf{T}} \mathbf{A} \bar{\mathbf{v}}_i^{\mathbf{r}} \right) \quad (5.13)$$

If the vectors are assumed unity vectors, equation (5.13) can be rewritten as:

$$J(\mathbf{A}) = \sum_{i=1}^N \frac{1}{\sigma_i^2} \left(1 - \bar{\mathbf{v}}_i^{\mathbf{b}\mathbf{T}} \mathbf{A} \bar{\mathbf{v}}_i^{\mathbf{r}} \right) \quad (5.14)$$

Shuster and Oh (1981) expanded equation (5.14) in terms of a quaternion. Adding the quaternion magnitude constraint as a Lagrange multiplier, the problem narrows down to finding the largest Eigenvalue and corresponding eigenvector (optimal quaternion estimation) of the following 4x4 matrix:

$$\mathbf{K} = \sum_{i=1}^N \frac{1}{\sigma_i^2} \begin{bmatrix} \left(\bar{\mathbf{v}}_i^{\mathbf{b}} \bar{\mathbf{v}}_i^{\mathbf{r}\mathbf{T}} + \bar{\mathbf{v}}_i^{\mathbf{r}} \bar{\mathbf{v}}_i^{\mathbf{b}\mathbf{T}} \right) - \left(\bar{\mathbf{v}}_i^{\mathbf{b}} \cdot \bar{\mathbf{v}}_i^{\mathbf{r}} \right) \mathbf{I}_{3 \times 3} & \bar{\mathbf{v}}_i^{\mathbf{b}} \times \bar{\mathbf{v}}_i^{\mathbf{r}} \\ \left(\bar{\mathbf{v}}_i^{\mathbf{b}} \times \bar{\mathbf{v}}_i^{\mathbf{r}} \right)^{\mathbf{T}} & \bar{\mathbf{v}}_i^{\mathbf{b}} \cdot \bar{\mathbf{v}}_i^{\mathbf{r}} \end{bmatrix} \quad (5.15)$$

5.3.2 Simulation

Using the Simulink simulation data of section 4.8, the attitude of the airship was calculated with the accelerometer and magnetometer vectors. As each vector has a bias drift combined with measurement noise, all this uncertainties must be combined in a single noise figure for use with the QUEST method.

During the simulations, each magnetometer axis has a bias drift of up to 0.008 Gauss, much larger than the RMS measurement noise of 0.335 mGauss. The magnetometer measurements will therefore be accurate within more or less 0.008 Gauss. Each accelerometer axis has a bias drift of up to 0.04 m/s². However, with dynamic accelerations peaking up to 0.24 m/s² and an RMS measurement noise figure of 0.0358 m/s², the gravity measurements will be accurate within 0.32 m/s².

The QUEST method was simulated with a magnetometer standard deviation of 0.008 Gauss and accelerometer standard deviation of 0.32 m/s². These figures are only used as relative weights between the different vectors, and not as real noise figures. No error bounds are calculated for the final answer.

Figure 5-4 shows the attitude estimation errors in degrees when using the QUEST method. Table 5-4 lists the attitude estimation error statistics. When compared to the TRIAD method, one can see

that there is a slight improvement in the attitude estimation, as expected. However, the slight improvement in estimation does not justify the additional computational complexity. The QUEST method only excels in situations where there are more than two vectors available.

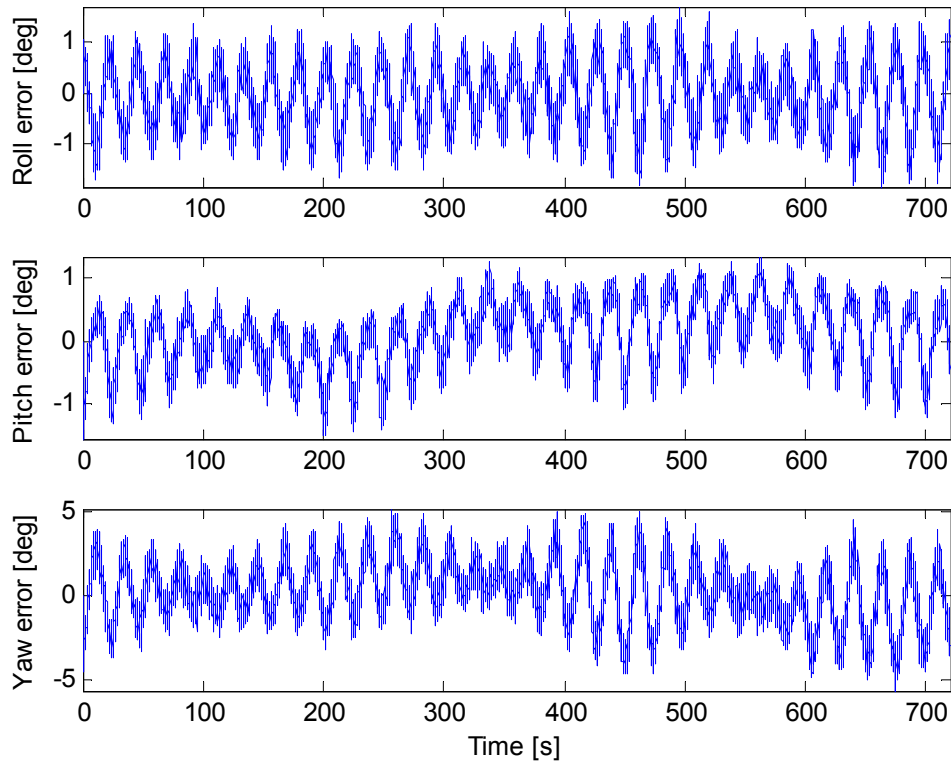


Figure 5-4: QUEST attitude estimation errors.

	Roll [deg]	Pitch [deg]	Yaw [deg]
Max	1.8134	1.5732	5.5963
Standard deviation	0.6148	0.5017	1.7770
RMS error	0.6173	0.5131	1.8023

Table 5-4: QUEST attitude estimation error statistics.

5.4 Multiplicative extended Kalman filter

Using the gyro integration method, but combining it with update measurements to keep the errors from growing calls for the use of a Kalman filter. The dynamics of the quaternion can be described as a function of the gyro inputs, which will be the model of the plant. The quaternion estimate will be used to estimate the magnetometer and accelerometer vectors in body axes. The difference

between the measured and estimated body vectors will be used to keep the quaternion errors within bounds.

5.4.1 Quaternion vector perturbations

In the multiplicative extended Kalman filter (MEKF) the error between the real (\mathbf{q}) and estimated ($\hat{\mathbf{q}}$) quaternion is defined as an error quaternion ($\delta\mathbf{q}$) with which the estimated quaternion must be composed to obtain the true quaternion (Markley 2003). The primary reason for using an error quaternion is that the unity constraint of the quaternion will be met when using quaternion multiplication. In mathematical terms it can be expressed as:

$$\mathbf{q} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \quad (5.16)$$

Taking the time derivative of the equation (5.16):

$$\dot{\mathbf{q}} = \delta\dot{\mathbf{q}} \otimes \hat{\mathbf{q}} + \delta\mathbf{q} \otimes \dot{\hat{\mathbf{q}}} \quad (5.17)$$

Using equation (2.51) in terms of the real quaternion \mathbf{q} and real angular rates $\boldsymbol{\omega}$:

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\omega} \otimes \mathbf{q} \quad (5.18)$$

Equation (5.18) can also be written in terms of the estimated quaternion $\hat{\mathbf{q}}$ and estimated angular rates $\hat{\boldsymbol{\omega}}$. Combining both equations in equation (5.17) and rearranging the terms, the following can be obtained:

$$\delta\dot{\mathbf{q}} \otimes \hat{\mathbf{q}} = \frac{1}{2} \boldsymbol{\omega} \otimes \mathbf{q} - \delta\mathbf{q} \otimes \left(\frac{1}{2} \hat{\boldsymbol{\omega}} \otimes \hat{\mathbf{q}} \right) \quad (5.19)$$

After right multiplying equation (5.19) by the inverse of $\hat{\mathbf{q}}$, the following result can be obtained:

$$\delta\dot{\mathbf{q}} = \frac{1}{2} (\boldsymbol{\omega} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \hat{\boldsymbol{\omega}}) \quad (5.20)$$

The gyro measurements include a bias drift as described in section 4.4. For accurate attitude estimation, the gyro bias must be estimated as well.

Define the real gyro measurements vector $\bar{\omega}$ in terms of the real gyro output vector $\bar{\mathbf{u}}$ and real gyro bias vector $\bar{\mathbf{b}}$:

$$\bar{\omega} = \bar{\mathbf{u}} - \bar{\mathbf{b}} \quad (5.21)$$

The estimated gyro measurements vector $\hat{\omega}$ is defined in terms of the estimated gyro bias vector $\hat{\mathbf{b}}$ and measurement noise vector $\bar{\eta}_{\text{gm}}$:

$$\hat{\omega} = \bar{\mathbf{u}} - \hat{\mathbf{b}} + \bar{\eta}_{\text{gm}} \quad (5.22)$$

Defining the perturbation in angular rates as a quaternion:

$$\delta\omega = \begin{bmatrix} \bar{\omega} - \hat{\omega} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}} - \bar{\mathbf{b}} - \bar{\eta}_{\text{gm}} \\ 0 \end{bmatrix} = \begin{bmatrix} -\delta\bar{\mathbf{b}} - \bar{\eta}_{\text{gm}} \\ 0 \end{bmatrix} \quad (5.23)$$

Using equation (5.23), equation (5.20) can be rewritten as

$$\begin{aligned} \delta\dot{\mathbf{q}} &= \frac{1}{2}((\hat{\omega} + \delta\omega) \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \hat{\omega}) \\ &= \frac{1}{2}(\hat{\omega} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \hat{\omega}) + \frac{1}{2}\delta\omega \otimes \delta\mathbf{q} \end{aligned} \quad (5.24)$$

Using equations (A.3) and (A.4), equation (5.24) can be rewritten in matrix form as:

$$\begin{bmatrix} \delta\dot{q}_1 \\ \delta\dot{q}_2 \\ \delta\dot{q}_3 \\ \delta\dot{q}_4 \end{bmatrix} = \begin{bmatrix} 0 & \hat{\omega}_3 & -\hat{\omega}_2 & 0 \\ -\hat{\omega}_3 & 0 & \hat{\omega}_1 & 0 \\ \hat{\omega}_2 & -\hat{\omega}_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta q_4 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & \delta\omega_3 & -\delta\omega_2 & \delta\omega_1 \\ -\delta\omega_3 & 0 & \delta\omega_1 & \delta\omega_2 \\ \delta\omega_2 & -\delta\omega_1 & 0 & \delta\omega_3 \\ -\delta\omega_1 & -\delta\omega_2 & -\delta\omega_3 & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta q_4 \end{bmatrix} \quad (5.25)$$

If the error quaternion denotes a small rotation, δq_4 will be close to unity, and equation (5.25) can be simplified to:

$$\begin{bmatrix} \delta\dot{q}_1 \\ \delta\dot{q}_2 \\ \delta\dot{q}_3 \\ \delta\dot{q}_4 \end{bmatrix} = \begin{bmatrix} 0 & \hat{\omega}_3 & -\hat{\omega}_2 & 0 \\ -\hat{\omega}_3 & 0 & \hat{\omega}_1 & 0 \\ \hat{\omega}_2 & -\hat{\omega}_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta q_4 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \delta\omega_3 \\ 0 \end{bmatrix} \quad (5.26)$$

Using equation (A.1), the matrix in equation (5.26) can be written as a cross product between the angular rate vector and the perturbation quaternion vector. Rewriting equation (5.26) in terms of the vector and angle part:

$$\delta\dot{\bar{\mathbf{q}}} = -\bar{\boldsymbol{\omega}} \times \delta\bar{\mathbf{q}} + \frac{1}{2}(-\delta\bar{\mathbf{b}} - \bar{\boldsymbol{\eta}}_{\text{gm}}) \quad (5.27)$$

$$\delta\dot{q}_4 = 0 \quad (5.28)$$

As can be seen in the equations above, the only dynamics of the system is contained in the vector part of the perturbation quaternion, and therefore only the vector perturbations need to be used.

5.4.2 Quaternion covariance matrix

Because the Kalman filter will only keep track of the covariance matrix of the quaternion vector perturbations, this covariance matrix must be converted to a full quaternion covariance matrix in order to have noise statistics on all four quaternion elements.

Rewriting equation (5.16) in matrix form using equation (A.4):

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 & \hat{q}_1 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 & \hat{q}_2 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 & \hat{q}_3 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 & \hat{q}_4 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta q_4 \end{bmatrix} \quad (5.29)$$

Assuming that the error quaternion will denote a small rotation, so that δq_4 will be close to unity, equation (5.29) can also be written as:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix} + \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix} \quad (5.30)$$

It can be seen from equation (5.30) that all four quaternion elements can be written as a function of the quaternion vector perturbations. Using the standard covariance transformation method (see Appendix E), the 4x4 covariance matrix of the full quaternion can be calculated from the 3x3 covariance matrix of the quaternion vector perturbations as follows:

$$\mathbf{P}_{4 \times 4} = \begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix} \mathbf{P}_{3 \times 3} \begin{bmatrix} \hat{q}_4 & \hat{q}_3 & -\hat{q}_2 & -\hat{q}_1 \\ -\hat{q}_3 & \hat{q}_4 & \hat{q}_1 & -\hat{q}_2 \\ \hat{q}_2 & -\hat{q}_1 & \hat{q}_4 & -\hat{q}_3 \end{bmatrix} \quad (5.31)$$

5.4.3 Bias drifts

Lefferts *et al.* (1982) showed that the gyro bias dynamics can be modelled as integrated zero mean Gaussian white noise (also see section 4.2.2). In reality, the gyro bias drift rates are the combination of several drift terms, but Gaussian white noise is a good approximation. The real bias vector can be described as:

$$\dot{\mathbf{b}} = \bar{\boldsymbol{\eta}}_{\text{gb}} \quad (5.32)$$

As there is no analytical model for the bias vector in terms of some state or input, the bias vector estimation dynamics are:

$$\hat{\dot{\mathbf{b}}} = 0 \quad (5.33)$$

Taking the perturbation of the bias vector (difference between the real bias and estimated bias), the perturbation bias dynamics can be written as follows:

$$\dot{\delta \mathbf{b}} = \bar{\boldsymbol{\eta}}_{\text{gb}} \quad (5.34)$$

5.4.4 State space model

Combining equations (5.27) and (5.34), the following state space perturbation model for the Kalman filter can be constructed:

$$\begin{bmatrix} \delta \dot{q}_1 \\ \delta \dot{q}_2 \\ \delta \dot{q}_3 \\ \delta \dot{b}_1 \\ \delta \dot{b}_2 \\ \delta \dot{b}_3 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & -0.5 & 0 & 0 \\ -\omega_3 & 0 & \omega_1 & 0 & -0.5 & 0 \\ \omega_2 & -\omega_1 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta b_1 \\ \delta b_2 \\ \delta b_3 \end{bmatrix} + \begin{bmatrix} -0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_{gmx} \\ \eta_{gmy} \\ \eta_{gmz} \\ \eta_{gbx} \\ \eta_{gby} \\ \eta_{gbz} \end{bmatrix} \quad (5.35)$$

There are six states, with three gyro measurement inputs. Noise couples into the system as follows: three gyro measurement noise sources for the gyro inputs ($\bar{\eta}_{gm}$) and three gyro bias noise sources ($\bar{\eta}_{gb}$) for the bias states. This model is given in the continuous domain and must be converted to the discrete domain as described in Chapter 3.

5.4.5 Measurement updates

After state propagation, a method is needed to write the update measurements as a function of the quaternion vector perturbations. Equation (5.16) states that the real quaternion is a function of the error quaternion and the estimated quaternion, and Markley (2003) showed that the direction cosine matrix can be expressed as a function of the perturbation quaternion and estimated quaternion as:

$$\mathbf{A}(\mathbf{q}) = \mathbf{A}(\delta \mathbf{q} \otimes \hat{\mathbf{q}}) = \mathbf{A}(\delta \mathbf{q})\mathbf{A}(\hat{\mathbf{q}}) \quad (5.36)$$

Writing the vector in body axes in terms of the reference vector and equation (5.36):

$$\bar{\mathbf{v}}^b = \mathbf{A}(\mathbf{q})\bar{\mathbf{v}}^r = \mathbf{A}(\delta \mathbf{q})\mathbf{A}(\hat{\mathbf{q}})\bar{\mathbf{v}}^r = \mathbf{A}(\delta \mathbf{q})\hat{\mathbf{v}}^b \quad (5.37)$$

The direction cosine matrix for a small error quaternion can be approximated by:

$$\mathbf{A}(\delta \mathbf{q}) \approx \begin{bmatrix} 1 & 2\delta q_3 & -2\delta q_2 \\ -2\delta q_3 & 1 & 2\delta q_1 \\ 2\delta q_2 & -2\delta q_1 & 1 \end{bmatrix} \quad (5.38)$$

Therefore, the measurement matrix for one vector update in terms of the perturbation states is:

$$\mathbf{H} = 2 \begin{bmatrix} 0 & -\hat{v}_{body}^z & \hat{v}_{body}^y & 0 & 0 & 0 \\ \hat{v}_{body}^z & 0 & -\hat{v}_{body}^x & 0 & 0 & 0 \\ -\hat{v}_{body}^y & \hat{v}_{body}^x & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.39)$$

For a measurement update, the direction cosine matrix in terms of the estimated quaternion must be constructed to convert the vector in reference axes to an estimated vector in body axes. This estimated body vector is used in equation (5.39) to calculate the measurement matrix. If there are two vector updates available, two of the matrices in equation (5.39) can be concatenated to form a 6x6 measurement matrix.

5.4.6 Measurement noise figures

The vectors used for the updates must be unity vectors and the noise figure for each vector must be scaled correspondingly. This way the Kalman filter can use the relative noise figures to determine which vector is the most accurate vector. The measurement vectors were normalised with the magnitude of the reference vectors. The measurement noise must include some modelling errors due to approximations of the error quaternion in terms of the quaternion vector perturbations, and approximations of the direction cosine matrix in terms of the error quaternion.

Due to bias drifts on both the accelerometers and magnetometer, the assumption of zero mean white Gaussian measurement noise is violated. This implies that the MEKF will not be strictly optimal. However, even with this violation, the MEKF still outperforms other attitude determination solutions (see section 5.4.8 for the simulated results).

By trial and error the measurement noise figures in the simulations were fine-tuned so that the diagonal elements of the Kalman filter state error covariance matrix were a reasonable indication of the real errors. As have been discussed in section 5.3.2, the magnetometer measurement will be accurate within 0.008 Gauss and the accelerometer measurement will be accurate within 0.32 m/s². Keeping the relative ratio, a magnetometer deviation of 5×0.008 Gauss and an accelerometer deviation of 5×0.32 m/s² had the best simulation results in terms of accurate estimation error figures in the Kalman filter covariance matrix.

5.4.7 Initialisation

To calculate the initial conditions of the Kalman filter states, it was assumed that the IMU would be at rest. This implies that the gyro outputs will be the bias values and the accelerometers will measure gravity only. Using the accelerometer and magnetometer as two vectors, the TRIAD algorithm is used to estimate the transformation matrix from inertial to body axes. The accelerometer was chosen as the more accurate vector, because the bias drifts on the accelerometers are significantly less than the bias drifts on the magnetometer. From the transformation matrix the quaternion values were calculated as defined in Appendix B.3. It is possible to use the QUEST method to calculate initial conditions, but the additional mathematical complexity does not justify the improvement in initial attitude estimation.

5.4.8 Simulations

Using the Simulink simulation data from section 4.8, the attitude of the airship was calculated with the MEKF using the gyro, accelerometer and magnetometer vectors. Figure 5-5 shows the attitude estimation errors in degrees when using the MEKF method. The dotted lines are the 2σ error bounds, meaning that the error should be 95% between the positive and negative error bounds.

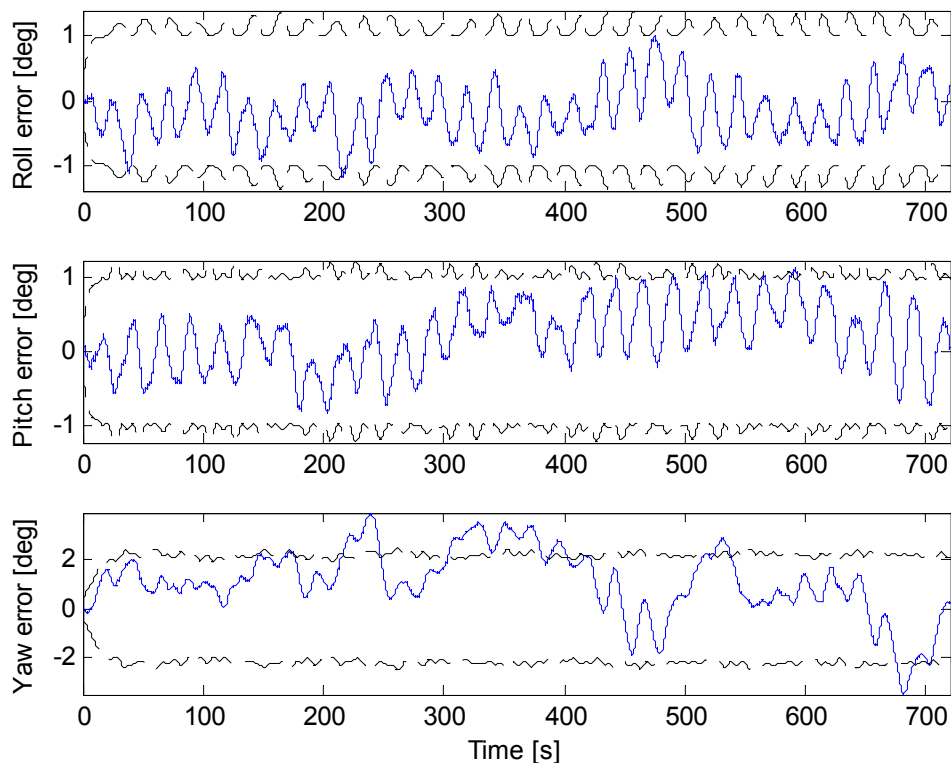


Figure 5-5: MEKF attitude estimation errors.

Table 5-5 lists the attitude estimation error statistics. When compared to all other methods, the MEKF proves to be the most accurate method.

	Roll [deg]	Pitch [deg]	Yaw [deg]
Max	1.1974	1.1489	3.9367
Standard deviation	0.4036	0.4368	1.3836
RMS error	0.4374	0.4822	1.7670

Table 5-5: MEKF attitude estimation error statistics

Figure 5-6 shows the bias estimation results. The dotted lines are the real bias values, while the solid lines are the estimated bias values. It can be seen that the estimated bias values track the real bias values very well, even in the presence of dynamic accelerations.

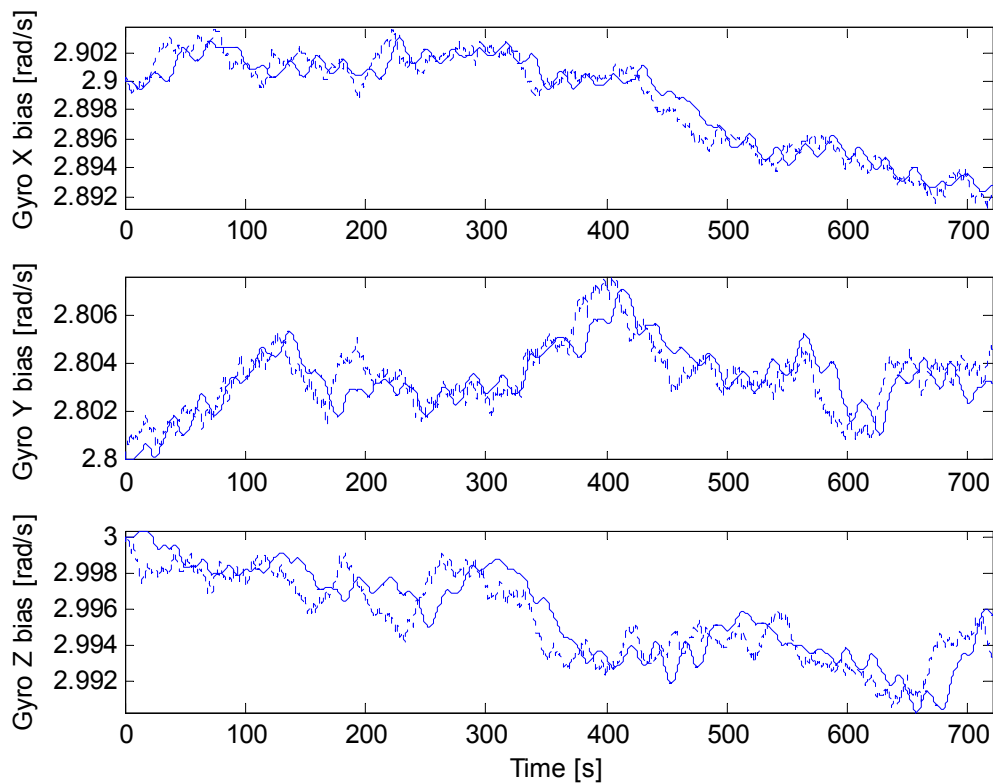


Figure 5-6: MEKF bias estimation results

Chapter 6: Inertial navigation

The operation of inertial navigation systems relies on Newton's laws of classical mechanics (Titterton *et al.* 1997). The first two laws are used, being:

1. *An object at rest or travelling in uniform motion will remain at rest or travelling in uniform motion unless acted upon by a net force.*
2. *The rate of change of the momentum of a body is directly proportional to the net force acting on it, and the direction of the change in momentum takes place in the direction of the net force.*

Newton's second law is used to derive the well-known $F = m\ddot{x}$ equation, assuming the mass of the object stays constant during the motion.

The acceleration of the body can be measured by accelerometers. By integrating the acceleration, speed can be determined, and by integrating the speed, position can be obtained. Figure 6-1 shows a diagram of the basic principle of inertial navigation. The initial speed and position must be known before this system can be used.

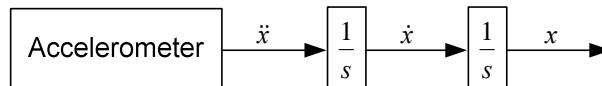


Figure 6-1: Basic principle of inertial navigation.

Performing the integrations when the accelerations are known result in the following equations:

$$\dot{x}(t) = \int \ddot{x}(t) dt = \dot{x}_0 + \ddot{x}(t)t \quad (6.1)$$

$$x(t) = \int \dot{x}(t) dt = x_0 + \dot{x}_0 t + \frac{1}{2} \ddot{x}(t)t^2 \quad (6.2)$$

where \ddot{x} is the acceleration, \dot{x} is the speed, \dot{x}_0 the initial speed, x is the position and x_0 the initial position.

The double integrating system is very susceptible to noise and bias drifts, resulting in growing errors over time. However, when the system is complemented with a GPS receiver that provides accurate estimates of the position and velocity, the whole system will provide better estimates than the two systems alone (Barbour 2001). Firstly, the error will be bounded by the GPS receiver's error. Secondly, the GPS usually gives relatively slow updates (a typical update rate of one second is common), and the double integrator system can be used to provide accurate data between the GPS updates (also called dead reckoning). Even when GPS updates are not available over some time range, the double integrating system can still provide navigation information. Availability of GPS updates eliminates the need for very accurate accelerometers and gyros (Hjortsmarker 2005). This leads to significant costs saving in building an accurate navigation system.

For an inertial navigation system to work, the integration of the accelerations must be done in inertial axes. This can be done with a gimballed inertial platform or strapdown platform. They are discussed subsequently.

6.1 Gimballed inertial platform

The idea of a gimballed inertial platform is to keep the accelerometers in a fixed orientation with respect to the inertial axial system, independent of the body orientation. This is done by placing the accelerometers in a set of three gimbals that are stabilised by electric torque motors. The inertial momentum vector of the platform will tend to keep the platform aligned with the inertial axial system. In addition, the attitude of the vehicle can be measured directly at the bearings of the gimbals. However, due to friction in the bearings and slip rings, gyros on board the platform are used to drive the torque motors to compensate for any rotations.

This system was extremely popular since the beginning of inertial sensing (Barbour *et al.* 1992) and proved reliable and accurate. Simple electronic circuits can be used to do the integrations. However, the system also has several disadvantages (Lawrence 1998):

- The system requires many precision mechanical parts, making it an expensive solution.
- The system is prone to gimbal lock. This can be a situation where two of the three gimbals become aligned and one of the rotation references is lost, or a situation where not all the gimbals can move over the whole 360 degrees due to physical limitations.
- The system consumes a lot of power.
- The accuracy and reliability of the system degrades as the components wear.

6.2 Strap down platforms

Instead of using a mechanically complex platform, the sensors are "strapped down" to the body of the vehicle. By keeping track of the attitude of the vehicle (see Chapter 5), the accelerations in body axes can be transformed to inertial axes. This turns out to be mathematically more complex, but has the following advantages (Lawrence 1998):

- The system is cheaper (no expensive mechanical parts), has low weight and is small.
- The system is reliable as there are no mechanical parts and no gimbal lock situations.

Typically, the sensors needed on a strapdown platform needs to be of a higher grade than the sensors on a gimbal platform due to the increase in dynamic range. However, with the advent of GPS, lower-grade sensors can be used as the total error of the system will be lower than the error on the GPS readings.

With the availability of low cost high performance embedded computers, the mathematical complexity can be justified and this method was chosen for implementation.

6.3 Simplifications

In order to obtain high accuracy inertial navigation, the earth's geodesy, rotation rate and gravity anomalies must be taken into account (Chatfield 1997). However, for application on a slow moving blimp, all the mentioned effects can be lumped together as modelling errors. As the GPS updates include all the above-mentioned effects, the problem can be simplified considerably. The earth was assumed perfectly round and non-rotating.

6.4 Position Kalman filter

From section 2.3.3, the dynamic equations of the position in terms of NED velocities are as follows:

$$\dot{\phi} = \frac{V_E}{(R+h)\cos\lambda} \quad (2.16)$$

$$\dot{\lambda} = \frac{V_N}{R+h} \quad (2.17)$$

$$\dot{h} = -V_D \quad (2.18)$$

Selecting the reference frame as the NED axial system, the transformation matrix of equation (2.41) will transform a vector in NED axes to body axes. Taking the inverse (or transpose) of equation (2.41) will convert a vector in body axes to NED axes. The accelerations in NED axes can therefore be written as:

$$\bar{\mathbf{f}}_{\text{NED}} = \mathbf{A}^T(\mathbf{q})\bar{\mathbf{f}} \quad (6.3)$$

Using equation (4.12) to remove the static acceleration in order to get dynamic acceleration:

$$\bar{\mathbf{a}}_{\text{NED}} = [0 \ 0 \ 9.81]^T - \bar{\mathbf{f}}_{\text{NED}} \quad (6.4)$$

The speed dynamics in NED axes can therefore be written as:

$$\dot{V}_N = -(q_4^2 + q_1^2 - q_2^2 - q_3^2)f_x - 2(q_1q_2 - q_4q_3)f_y - 2(q_4q_2 + q_1q_3)f_z \quad (6.5)$$

$$\dot{V}_E = -2(q_4q_3 + q_1q_2)f_x - (q_4^2 - q_1^2 + q_2^2 - q_3^2)f_y - 2(q_2q_3 - q_4q_1)f_z \quad (6.6)$$

$$\dot{V}_D = -2(q_1q_3 - q_4q_2)f_x - 2(q_4q_1 + q_2q_3)f_y - (q_4^2 - q_1^2 - q_2^2 + q_3^2)f_z + 9.81 \quad (6.7)$$

6.4.1 State space model

Equations (2.16), (2.17), (2.18), (6.5), (6.6) and (6.7) are six non-linear state equations. These non-linear equations can be used in an extended Kalman filter configuration as described in section 3.3. The extended Kalman filter has seven inputs: three for the accelerometers and four for the quaternion (as estimated by the attitude Kalman filter defined in chapter 5). All of the six states can be updated by a GPS receiver to keep the navigation errors from growing.

Although the accelerometers have bias drifts, no bias estimation is done. This is because the velocity and position updates of the GPS are too noisy for accurate bias estimates. Any error due to accelerometer bias drifts will be corrected by GPS updates.

After linearising the equations above, the state perturbations dynamics can be written as a state space model in the continuous domain:

$$\begin{bmatrix} \delta\dot{\phi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \\ \delta\dot{V}_N \\ \delta\dot{V}_E \\ \delta\dot{V}_D \end{bmatrix} = \begin{bmatrix} 0 & \frac{V_E \sin \lambda}{(R+h)\cos^2 \lambda} & \frac{-V_E}{(R+h)^2 \cos \lambda} & 0 & \frac{1}{(R+h)\cos \lambda} & 0 \\ 0 & 0 & \frac{-V_N}{(R+h)^2} & \frac{1}{R+h} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta\lambda \\ \delta h \\ \delta V_N \\ \delta V_E \\ \delta V_D \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\mathbf{A}^T \end{bmatrix} \begin{bmatrix} \eta_{fx} \\ \eta_{fy} \\ \eta_{fz} \end{bmatrix} + \dots$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -2q_1f_x - 2q_2f_y - 2q_3f_z & 2q_2f_x - 2q_1f_y - 2q_4f_z & 2q_3f_x + 2q_4f_y - 2q_1f_z & -2q_4f_x + 2q_3f_y - 2q_2f_z & \\ -2q_2f_x + 2q_1f_y + 2q_4f_z & -2q_1f_x - 2q_2f_y - 2q_3f_z & -2q_4f_x + 2q_3f_y - 2q_2f_z & -2q_3f_x - 2q_4f_y + 2q_1f_z & \\ -2q_3f_x - 2q_4f_y + 2q_1f_z & 2q_4f_x - 2q_3f_y + 2q_2f_z & -2q_1f_x - 2q_2f_y - 2q_3f_z & 2q_2f_x - 2q_1f_y - 2q_4f_z & \end{bmatrix} \begin{bmatrix} \eta_{q1} \\ \eta_{q2} \\ \eta_{q3} \\ \eta_{q4} \end{bmatrix} \quad (6.8)$$

where $\boldsymbol{\eta}_q$ is the noise on the quaternion estimates.

The measurement matrix for the state space model will be a 6x6 identity matrix, meaning there will be six measurements equal to the six different states. The continuous state space model can be converted to the discrete domain as discussed in Chapter 3.

It is possible to use the velocity estimates to estimate the dynamic accelerations of the body. This estimated dynamic accelerations can be used to remove the effect of dynamic accelerations from the accelerometer output, so that only the static accelerations (gravity) is used by the attitude Kalman filter. However, simulations showed that the noise on the velocity estimates transformed to dynamic accelerations would yield a very noisy gravity vector, which in turn resulted in a very poor attitude estimation.

6.4.2 Simulation results

The position Kalman filter was run sequentially with the MEKF attitude Kalman filter using the Simulink data from section 4.8. See Figure 6-2 for a schematic layout of the two Kalman filters. The accelerometer readings are used by both Kalman filters, but with different noise figures. The noise figure for the attitude estimator includes dynamic acceleration uncertainties, whereas the position Kalman filter only uses measurement noise on the accelerometers. With a bias drift of up to 0.04 m/s^2 on each accelerometer axis, combined with 0.036 m/s^2 RMS measurement noise, a standard deviation of $5 \times 0.08 \text{ m/s}^2$ was used for the accelerometers in the position EKF. The position and velocity estimation error statistics are given in Table 6-1. The attitude estimation error statistics were evaluated in Chapter 5 and is given in Table 5-5.

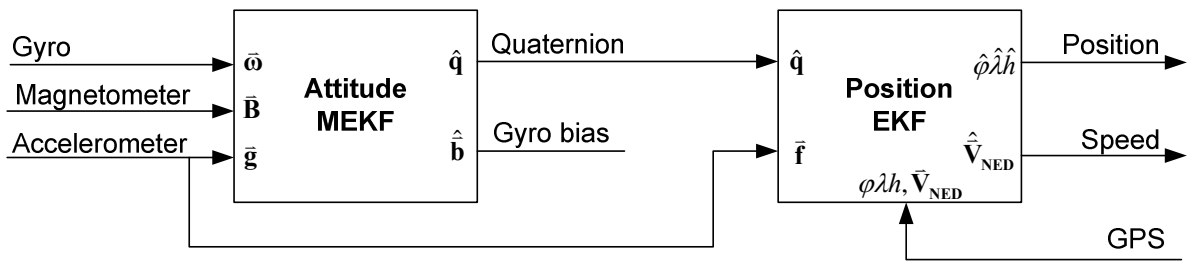


Figure 6-2: Attitude Kalman filter in parallel with the position Kalman filter.

Estimation errors	Max	Standard deviation	RMS error
Longitude [deg]	0.2085×10^{-4}	0.6531×10^{-5}	0.6651×10^{-5}
Latitude [deg]	0.1758×10^{-4}	0.5145×10^{-5}	0.5198×10^{-5}
Height [m]	1.9536	0.5734	0.5828
North velocity [m/s]	0.8378	0.2168	0.2173
East velocity [m/s]	0.8174	0.2230	0.2233
Down velocity [m/s]	0.7516	0.2038	0.2045

Table 6-1: Position estimation error statistics.

Figure 6-3 shows the velocity estimation errors. The noise figures on the estimated velocities are lower than the noise on the GPS velocity measurements, illustrating the advantage of integrating GPS with INS. The position estimation errors are shown in Figure 6-4. Due to the non-linear noise transformations, the 2σ -bounds for the position estimation are not very accurate (the errors are not 95% of the time between the two error bounds). The noise on the position estimates are not white Gaussian noise, and cannot be fully described by a covariance matrix alone. However, the real errors are still close to the theoretical error bounds.

6.5 Attitude and position Kalman filter

Because the position Kalman filter uses the direction cosine matrix in terms of a quaternion to transform body accelerations into inertial accelerations, the GPS velocity and position updates can help to determine the attitude of the body. The attitude Kalman filter of section 5.4.4 and position Kalman filter of section 6.4 can be combined to form a big Kalman filter, estimating the attitude, position and velocity of the blimp. Figure 6-5 is a schematical layout of the combined attitude and position Kalman filter.

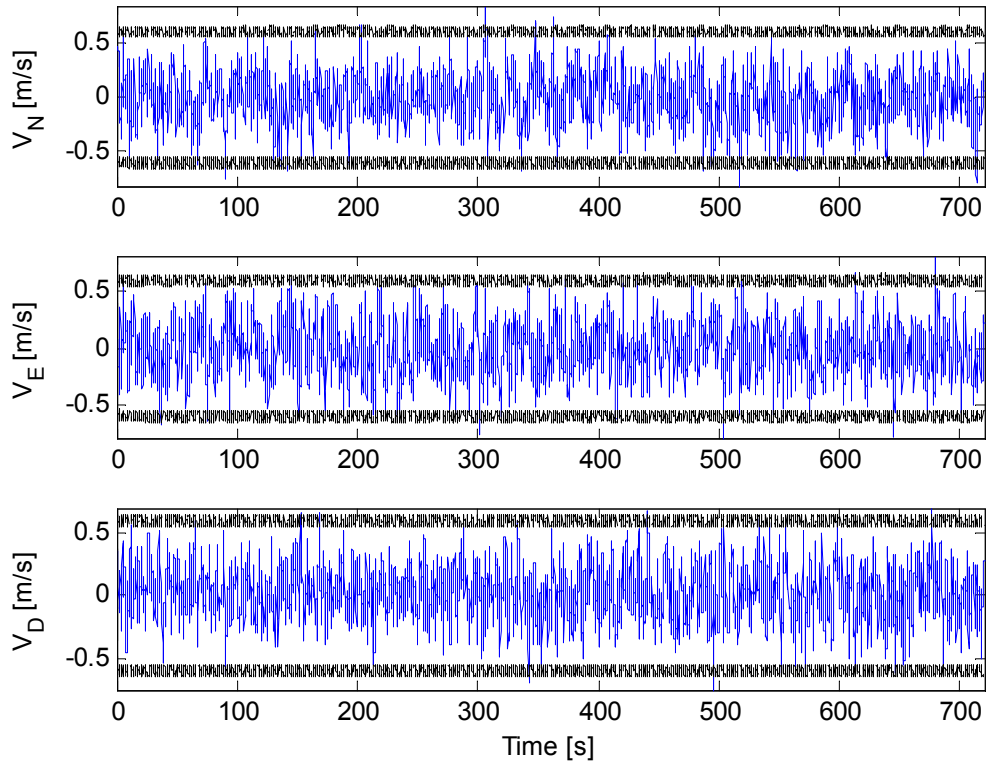


Figure 6-3: Velocity estimation errors.

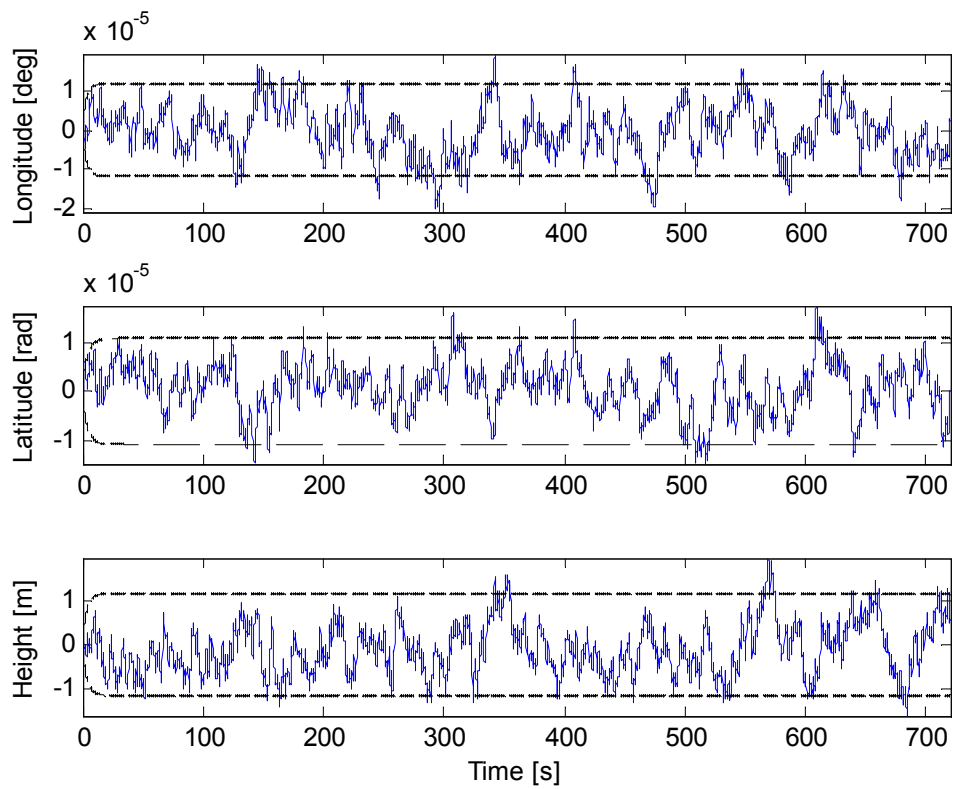


Figure 6-4: Position estimation errors.

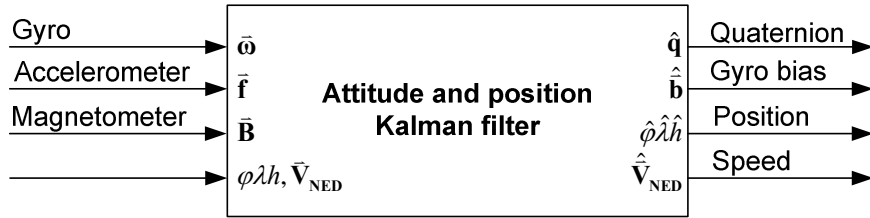


Figure 6-5: Combined attitude and position Kalman filter.

6.5.1 Quaternion vector perturbations states

The quaternion vector perturbation dynamics will be the same as developed in section 5.4. This dynamic equation calculates the quaternion vector perturbation as a function of the gyro inputs. The three gyro biases are also estimated. The final state space system is given in equation (5.35).

6.5.2 NED velocity perturbation states

The NED velocity perturbations must be written as a function of the quaternion vector perturbations to use the same quaternion vector perturbations as defined above.

From equations (6.3) and (6.4) the accelerations in NED axes are:

$$\mathbf{f}_{NED} = [0 \ 0 \ 9.81]^T - \mathbf{A}^T(\mathbf{q})\bar{\mathbf{f}} \quad (6.9)$$

The transpose of equation (5.36) can be expressed as:

$$\mathbf{A}^T(\mathbf{q}) = \mathbf{A}^T(\delta\mathbf{q} \otimes \hat{\mathbf{q}}) = \mathbf{A}^T(\hat{\mathbf{q}})\mathbf{A}^T(\delta\mathbf{q}) \quad (6.10)$$

Therefore, the real accelerations in NED axes can be expressed as a function of the quaternion vector perturbations as follows:

$$\mathbf{f}_{NED} = [0 \ 0 \ 9.81]^T - \mathbf{A}^T(\hat{\mathbf{q}})\mathbf{A}^T(\delta\mathbf{q})\bar{\mathbf{f}} \quad (6.11)$$

The estimated accelerations in NED axes can be written as:

$$\hat{\mathbf{f}}_{NED} = [0 \ 0 \ 9.81]^T - \mathbf{A}^T(\hat{\mathbf{q}})(\bar{\mathbf{f}} + \bar{\boldsymbol{\eta}}_f) \quad (6.12)$$

where $\bar{\boldsymbol{\eta}}_{\mathbf{f}}$ denotes the noise on the accelerometers.

The dynamic NED speed perturbation equations can then be written in state space format:

$$\begin{aligned}\delta\dot{\mathbf{V}}_{\text{NED}} &= \mathbf{f}_{\text{NED}} - \hat{\mathbf{f}}_{\text{NED}} = \mathbf{A}^T(\hat{\mathbf{q}}) \begin{bmatrix} 0 & -2f_z & 2f_y \\ 2f_z & 0 & -2f_x \\ -2f_y & 2f_x & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix} + \mathbf{A}^T(\hat{\mathbf{q}}) \begin{bmatrix} \eta_{f_x} \\ \eta_{f_y} \\ \eta_{f_z} \end{bmatrix} \\ &= 2\mathbf{A}^T(\hat{\mathbf{q}})\bar{\mathbf{f}} \times \delta\bar{\mathbf{q}} + \mathbf{A}^T(\hat{\mathbf{q}})\bar{\boldsymbol{\eta}}_{\mathbf{f}}\end{aligned}\quad (6.13)$$

6.5.3 Position perturbation states

The position perturbation equations can be written as the difference between the true and estimated states as follows:

$$\delta\dot{\varphi} = \frac{V_E}{(R+h)\cos\lambda} - \frac{\hat{V}_E}{(R+\hat{h})\cos\hat{\lambda}} \quad (6.14)$$

$$\delta\dot{\lambda} = \frac{V_N}{R+h} - \frac{\hat{V}_N}{R+\hat{h}} \quad (6.15)$$

$$\delta\dot{h} = -V_D + \hat{V}_D \quad (6.16)$$

Using the superposition principle, equations (6.14) to (6.16) can be expanded in terms of perturbations as follows:

$$\delta\dot{\varphi} = \frac{1}{(R+h)\cos\lambda} \delta V_E + \frac{-V_E}{(R+h)^2 \cos\lambda} \delta h + \frac{V_E \sin\lambda}{(R+h)\cos^2\lambda} \delta\lambda \quad (6.17)$$

$$\delta\dot{\lambda} = \frac{1}{R+h} \delta V_N + \frac{-V_N}{(R+h)^2} \delta h \quad (6.18)$$

$$\delta\dot{h} = -\delta V_D \quad (6.19)$$

Rewriting equations (6.17) to (6.19) in state space format:

$$\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{(R+h)\cos\lambda} & 0 & 0 & \frac{V_E \sin\lambda}{(R+h)\cos^2\lambda} & \frac{-V_E}{(R+h)^2 \cos\lambda} \\ \frac{1}{R+h} & 0 & 0 & 0 & 0 & \frac{-V_N}{(R+h)^2} \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta V_N \\ \delta V_E \\ \delta V_D \\ \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} \quad (6.20)$$

6.5.4 State space model

Combining equations (5.35), (6.13) and (6.20), a 12 state perturbation Kalman filter can be constructed. As defined in section 5.4.2, the quaternion vector perturbations can be used to update the whole quaternion, resulting in 13 estimated states.

$$\begin{bmatrix} \delta\dot{q}_1 \\ \delta\dot{q}_2 \\ \delta\dot{q}_3 \\ \delta\dot{b}_x \\ \delta\dot{b}_y \\ \delta\dot{b}_z \\ \delta\dot{V}_n \\ \delta\dot{V}_e \\ \delta\dot{V}_d \\ \delta\dot{\varphi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \end{bmatrix} = \mathbf{A}^T(\hat{\mathbf{q}}) \begin{bmatrix} 0 & -2f_z & 2f_y \\ 2f_z & 0 & -2f_x \\ -2f_y & 2f_x & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta b_x \\ \delta b_y \\ \delta b_z \\ \delta V_n \\ \delta V_e \\ \delta V_d \\ \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} + \dots \quad (6.21)$$

$$\begin{bmatrix} -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{gmx} \\ \eta_{gmy} \\ \eta_{gmc} \\ \eta_{gbx} \\ \eta_{gby} \\ \eta_{gbz} \\ \eta_{fx} \\ \eta_{fy} \\ \eta_{fz} \end{bmatrix}$$

The gyro and acceleration measurements are inputs to the Kalman filter, while the magnetometer and GPS are used to update the states. The Kalman filter estimates the attitude, gyro bias, velocity and position of the body.

As stated in Chapter 5, at least two vectors are needed for attitude determination. The magnetometer supplies one vector, while the remaining attitude information is supplied by the GPS velocity

updates. The reason why the GPS velocity contains attitude information is that the accelerometer vector in body axes is transformed to an acceleration vector in inertial axes, which is a function of the attitude of the body. The transformed accelerations are used to estimate the NED velocities, and any update in NED velocities can aid in determining the attitude. This attitude information update turns out to be very useful, because this method eliminates the need for using the accelerometer measurements as a gravity vector. Therefore, this Kalman filter can be used in applications where the body experiences large accelerations.

6.5.5 Measurement matrix

As the GPS updates are much slower than the IMU updates, the Kalman filter was updated only with the magnetometer between GPS updates. The measurement matrix given in equation (5.39) can be used for the magnetometer updates. However, the measurement matrix needs to be expanded with zeros to have 12 columns, as there are 12 perturbation states.

When GPS updates are available, six more rows must be added to the measurement matrix of equation (5.39) to update the position and velocity states:

$$\mathbf{H} = \begin{bmatrix} 0 & -2\hat{v}_{body}^z & 2\hat{v}_{body}^y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\hat{v}_{body}^z & 0 & -2\hat{v}_{body}^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2\hat{v}_{body}^y & 2\hat{v}_{body}^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.22)$$

In the case of the blimp, the acceleration vector can still be used to aid in determining attitude. First, the Kalman filter will be simulated with only magnetometer and GPS updates, and then the Kalman filter will be simulated with magnetometer, accelerometer and GPS updates.

6.5.6 Simulation with magnetometer and GPS updates

Using the Simulink simulation data from section 4.8 with magnetometer updates supplied at 20 ms and GPS updates supplied at 250 ms, enough information is given so that the attitude and position

Kalman filter can converge to a solution. Table 6-2 shows the estimation error statistics of the attitude and position Kalman filter with magnetometer and GPS updates.

Estimation errors	Max	Standard deviation	RMS error
Roll [deg]	1.5949	0.6136	0.6187
Pitch [deg]	1.4222	0.4780	0.5327
Yaw [deg]	4.8758	1.9170	1.9512
Longitude [deg]	0.2050×10^{-4}	0.6535×10^{-5}	0.6547×10^{-5}
Latitude [deg]	0.1742×10^{-4}	0.5131×10^{-5}	0.5182×10^{-5}
Height [m]	1.9538	0.5740	0.5833
North velocity [m/s]	0.8439	0.2169	0.2176
East velocity [m/s]	0.8547	0.2300	0.2300
Down velocity [m/s]	0.7492	0.2038	0.2044

Table 6-2: Estimation error statistics with magnetometer and GPS updates.

6.5.7 Simulation with magnetometer, accelerometer and GPS updates

For application on a blimp, the accelerometer measurements can also be used as a noisy gravity vector to aid in attitude determination as was done in the dedicated attitude estimator of Chapter 5. Using the Kalman filter given in equation (6.21), the accelerometer data is both used as an input to the Kalman filter, as well as a feedback measurement of the attitude. Using the Simulink simulation data from section 4.8, the Kalman filter was updated with magnetometer and accelerometer data every 20 ms. In addition, GPS updates were added every 250 ms.

Table 6-3 contains the estimation error statistics for the Kalman filter with magnetometer, accelerometer and GPS updates. As the attitude is updated by three sources, one would expect that the attitude would be estimated better. Comparing Table 5-5 with Table 6-3, one can see that this is indeed the case.

However, in order to reach this increased accuracy, a lot more processing power was needed. The different Kalman filter accuracies as a function of processing power is compared and summarised in Chapter 7.

Estimation errors	Max	Standard deviation	RMS error
Roll [deg]	1.1045	0.3577	0.3782
Pitch [deg]	1.0797	0.3998	0.4359
Yaw [deg]	3.5154	1.2919	1.6044
Longitude [deg]	0.2062×10^{-4}	0.6573×10^{-5}	0.6590×10^{-5}
Latitude [deg]	0.1768×10^{-4}	0.5145×10^{-5}	0.5200×10^{-5}
Height [m]	1.9532	0.5739	0.5832
North velocity [m/s]	0.8424	0.2178	0.2182
East velocity [m/s]	0.8233	0.2246	0.2248
Down velocity [m/s]	0.7497	0.2038	0.2045

Table 6-3: Estimation error statistics with magnetometer, accelerometer and GPS updates.

6.5.8 GPS update rates

The performance of the 12 perturbation state Kalman filter was evaluated at different GPS update rates. During simulation, magnetometer updates were supplied at 20 ms, and the GPS update rate was varied between 20 ms and 10 s. Figure 6-6 shows the RMS error on the attitude, velocity and position estimates as a function of the GPS update rate. Clearly, a faster GPS update rate will result in better estimates. The Kalman filter stays stable for slow GPS update rates, but the estimate errors rise dramatically between two update intervals. This means that the filter will still function when there are no GPS updates available (due to signal blockage or whatever other reason). However, the estimates will be significantly affected during that period.

6.6 GPS delay states

Due to GPS update delays (see section 4.7.3), the filter may not perform optimally when the body is moving at a fast speed. It is therefore advisable to include the update delay dynamics in the Kalman filter. As the blimp is not a fast moving object, this extra dynamics does not need to be implemented, but it is given below for completeness.

A delay of T_d seconds can be modelled in the frequency domain as a phase shift:

$$D(s) = e^{-T_d s} \quad (6.23)$$

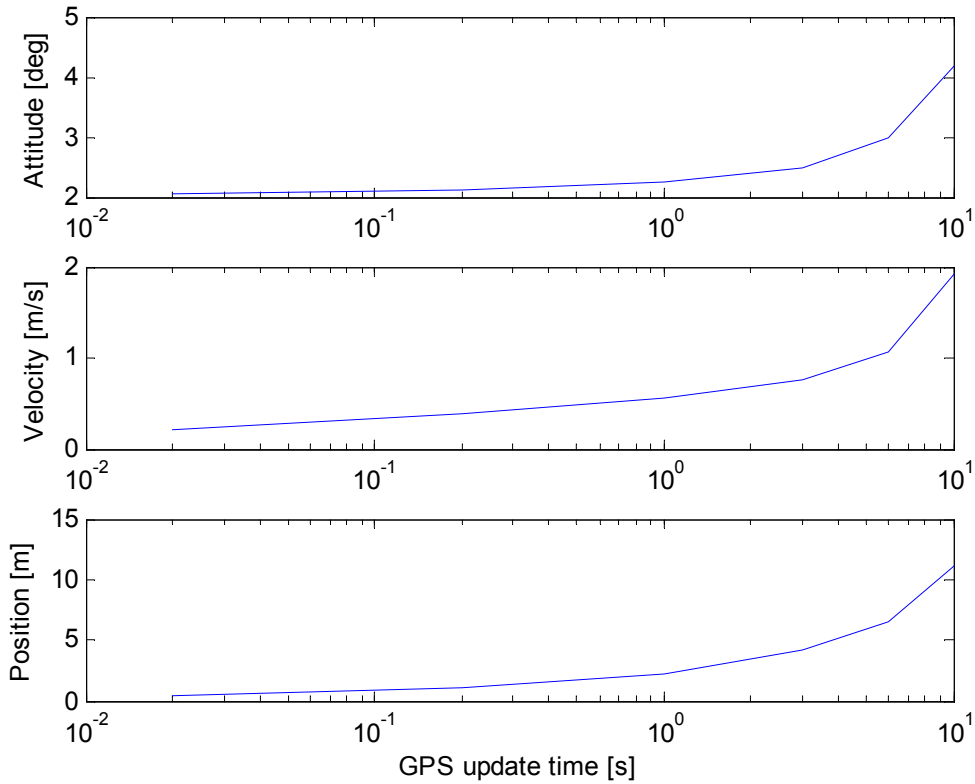


Figure 6-6: RMS errors of the 12 state Kalman filter as a function of GPS update interval.

The first order Padé approximation of equation (6.23) is (Franklin *et al.* 2002):

$$D(s) \approx \frac{-s + \frac{2}{T_d}}{s + \frac{2}{T_d}} \quad (6.24)$$

Writing the delayed state as a function of the normal state and transforming back to the time domain:

$$\dot{X}_d = \frac{2}{T_d} X - \frac{2}{T_d} X_d - \dot{X} \quad (6.25)$$

Using the equation of \dot{X} as a function of other states, the delayed state dynamics can be written as a function of the normal states and the delayed state. Instead of updating state X , the delayed state X_d must be updated for the Kalman filter to use the modelled delay.

For every delay that has to be modelled, the number of states used in the Kalman filter will increase by one. In the case of GPS updates, there are delays on six states: longitude, latitude, height and three NED velocities.

When six states are added to the attitude and position Kalman filter of section 6.5.4, the result is an 18 state Kalman filter. This is quite large for a Kalman filter and usually impractical to implement. Even without GPS delayed states, the 12 state Kalman filter can be impractical to implement. However, it is possible to divide the big Kalman filter in two separate filters, as discussed below.

6.7 Smaller attitude and velocity Kalman filter

The longitude, latitude and height states only depend on the velocity states, so these states can be removed and put it in a smaller dedicated Kalman filter. The attitude is estimated by using the magnetometer and GPS velocity updates, while the position is estimated by using the velocity estimates of the first Kalman filter. Figure 6-7 gives a schematical layout of the two Kalman filters.

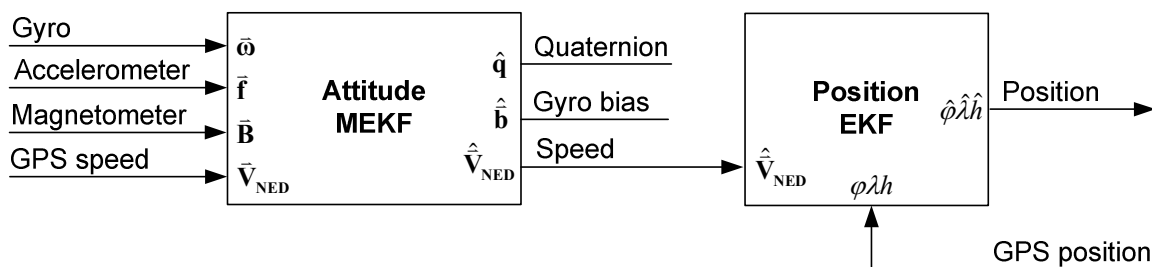


Figure 6-7: Reduced state attitude and position Kalman filters.

No delayed states have been added to the Kalman filters given below, but it can easily be done. The two filters without the delayed states only illustrates the splitting principle.

The state equations for the attitude Kalman filter are given in equation (6.26). When there are only magnetometer updates available at a specific time, the measurement matrix will have three rows for the magnetometer update. In the case of an additional GPS velocity update, the measurement matrix will be expanded to six rows. The magnetometer and GPS velocity measurement matrix is given in equation (6.27).

$$\begin{aligned}
 \begin{bmatrix} \delta \dot{q}_1 \\ \delta \dot{q}_2 \\ \delta \dot{q}_3 \\ \delta \dot{b}_x \\ \delta \dot{b}_y \\ \delta \dot{b}_z \\ \delta \dot{V}_n \\ \delta \dot{V}_e \\ \delta \dot{V}_d \end{bmatrix} &= \begin{bmatrix} 0 & \omega_3 & -\omega_2 & -0.5 & 0 & 0 & 0 & 0 & 0 \\ -\omega_3 & 0 & \omega_1 & 0 & -0.5 & 0 & 0 & 0 & 0 \\ \omega_2 & -\omega_1 & 0 & 0 & 0 & -0.5 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -2f_z & 2f_y & 0 & 0 & 0 & 0 & 0 & 0 \\ 2f_z & 0 & -2f_x & 0 & 0 & 0 & 0 & 0 & 0 \\ -2f_y & 2f_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \delta b_x \\ \delta b_y \\ \delta b_z \\ \delta V_n \\ \delta V_e \\ \delta V_d \end{bmatrix} + \dots \\
 & \begin{bmatrix} -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}^T(\hat{\mathbf{q}}) & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \eta_{fx} & & \\ & & & & & & \eta_{fy} & & \\ & & & & & & \eta_{fz} & & \end{bmatrix} \begin{bmatrix} \eta_{gmx} \\ \eta_{gmy} \\ \eta_{gmz} \\ \eta_{gbx} \\ \eta_{gby} \\ \eta_{gbz} \\ \eta_{fx} \\ \eta_{fy} \\ \eta_{fz} \end{bmatrix} \tag{6.26}
 \end{aligned}$$

$$\mathbf{H} = \begin{bmatrix} 0 & -2\hat{v}_{body}^z & 2\hat{v}_{body}^y & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\hat{v}_{body}^z & 0 & -2\hat{v}_{body}^x & 0 & 0 & 0 & 0 & 0 & 0 \\ -2\hat{v}_{body}^y & 2\hat{v}_{body}^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.27}$$

The state equations for the position Kalman filter are given in equation (6.28). The measurement matrix for the GPS positions will be a 3x3 identity matrix.

$$\begin{bmatrix} \delta \dot{\varphi} \\ \delta \dot{\lambda} \\ \delta \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & \frac{V_e \sin \lambda}{R+h \cos^2 \lambda} & \frac{-V_e}{(R+h)^2 \cos \lambda} \\ 0 & 0 & \frac{-V_n}{(R+h)^2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \varphi \\ \delta \lambda \\ \delta h \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{(R+h) \cos \lambda} & 0 \\ \frac{1}{R+h} & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \eta_{Vn} \\ \eta_{Ve} \\ \eta_{Vd} \end{bmatrix} \tag{6.28}$$

where $\bar{\boldsymbol{\eta}}_V$ is the noise on the NED velocity estimates.

The two filters were run sequentially with the simulation data from section 4.8. The estimation error statistics are given in Table 6-4.

Comparing Table 6-4 with Table 6-2, one can see that the GPS position updates do help in estimating the attitude. Overall, the smaller attitude and velocity Kalman filter performs slightly worse in all the estimations. However, when comparing the number of floating point operations

needed for each Kalman filter, the degradation in estimation can be justified by the decrease in computational power needed.

Estimation errors	Max	Standard deviation	RMS error
Roll [deg]	1.6142	0.6245	0.6294
Pitch [deg]	1.4432	0.4802	0.5353
Yaw [deg]	5.0020	1.9385	1.9739
Longitude [deg]	0.2072×10^{-4}	0.6730×10^{-5}	0.6733×10^{-5}
Latitude [deg]	0.1786×10^{-4}	0.5658×10^{-5}	0.5679×10^{-5}
Height [m]	2.1805	0.5820	0.5828
North velocity [m/s]	0.8420	0.2176	0.2183
East velocity [m/s]	0.8584	0.2289	0.2290
Down velocity [m/s]	0.7425	0.2047	0.2055

Table 6-4: Attitude estimation error statistics for the reduced Kalman filter.

Chapter 7: Solution implementation

To get a good overview of all the attitude and position Kalman filters presented in Chapters 5 and 6, the error statistics as well as the number of floating point operations required for the implementation of the Kalman filter must be compared.

Using equations (3.27) and (3.28), the number of floating point operations per second can be calculated with IMU data supplied at 20 ms and GPS updates at 250 ms. Note that these number of floating point operations are only a raw indication of the real number of floating point operations needed. It does not include any initialisation, linearisation and other miscellaneous operations needed for the whole Kalman filter implementation. However, the Kalman filter equations will usually dominate the total number of floating-point operations needed, especially for a big Kalman filter.

Four different solutions are presented below. Every solution estimates the attitude, velocity and position of the blimp, but different Kalman filter configurations are used.

7.1 Solution A

The attitude is estimated with the MEKF attitude estimator given in equation (5.35). It consists of six perturbation states with six noise inputs (gyro measurement noise and gyro bias drift noise). The measurement matrix has six rows (magnetometer and gravity vectors) every 20 ms. The Kalman filter estimates the attitude and gyro bias drifts.

The velocity and position is estimated by the Kalman filter given in equation (6.8). It uses the attitude estimates of the MEKF attitude estimator. The Kalman filters consist of six states, with seven noise inputs (four quaternion elements and three accelerometer measurements). All six states are updated by the GPS every 250 ms.

7.2 Solution B

Instead of two separate Kalman filters, the attitude, velocity and position are estimated by one big Kalman filter, given in equation (6.21). It has twelve perturbation states, with nine noise inputs (gyro measurement noise, gyro bias drift noise and accelerometer measurement noise). Every 20 ms

the filter is updated with a magnetometer vector, resulting in a measurement matrix of three rows. Every 250 ms the GPS velocity and position updates are included, resulting in a measurement matrix of nine rows. The filter estimates the attitude, gyro bias drifts, velocity and position of the blimp. This solution is a general-purpose solution with no constraints on the accelerations the vehicle may experience.

7.3 Solution C

The same Kalman filter is used as in Solution B, but this time the updates are customised for the application on a blimp where the accelerometers can also be used to measure the earth's gravity vector. Every 20 ms the measurement matrix consists of six rows: three for the magnetometer vector and three for the gravity vector. Every 250 ms the GPS velocity and position updates are included, resulting in a measurement matrix of twelve rows.

7.4 Solution D

In this solution, the big Kalman filter of solution B is split in two: the attitude and velocity are estimated by a Kalman filter given in equation (6.26), and the position is estimated by the Kalman filter given in equation (6.28). This solution is also a general-purpose solution suitable for all types of vehicles.

The attitude and velocity Kalman filter consists of nine perturbation states, with nine noise inputs: gyro measurement noise, gyro bias drift noise and accelerometer measurement noise. Every 20 ms the filter is updated with the magnetometer vector, resulting in a measurement matrix of three rows. Every 250 ms the GPS velocity updates are included, resulting in a measurement matrix of six rows. The filter estimates the attitude, gyro bias drift and velocity of the body.

The position Kalman filter consists of three states with three noise inputs. The velocity estimates of the first Kalman filter are used to estimate the position of the vehicle. GPS position updates are provided every 250 ms.

7.5 Comparison

The four different solutions described above are compared in Table 7-1. The RMS error of the attitude, velocity and position is given as an indication of the solution performance. The number of

floating point operations per second is given as an indication of the processing power needed for implementation on a computer.

Kalman filter solution	States	FLOPS	RMS errors		
			Attitude [deg]	Velocity [m/s]	Position [m]
Solution A	12	354666	1.8831	0.3727	1.1051
Attitude estimator (5.35)	6	287050	1.8831		
Position estimator (6.8)	6	67616		0.3727	1.1051
Solution B (6.21)	12	1006372	2.1151	0.3769	1.0966
Solution C (6.21)	12	1326346	1.7050	0.3741	1.1009
Solution D	12	498480	2.1399	0.3773	1.1400
Attitude estimator (6.26)	9	488104	2.1399	0.3773	
Position estimator (6.28)	3	10376			1.1400

Table 7-1: Comparison of different Kalman filter solutions.

Comparing solution A with solution B, the first significant conclusion is that a big Kalman filter does not always results in better estimates. In fact, the bigger Kalman filter of solution B performed worse than the two smaller Kalman filters of solution A, while solution B required almost three times more floating-point operations per second.

However, solution B is usable for all types of vehicles, whereas solution A is not. When solution B is customised for use on a blimp (as is done in solution C), the bigger Kalman filter of solution C does give better estimates, but requires about a million more floating-point operations per second compared to solution A. This implies that there is a trade-off between accuracy and processing power.

Comparing solution B with solution D, the big Kalman filter of solution B performed slightly better than the two smaller Kalman filters of solution D. However, the increased accuracy comes with a huge increase in the number of floating point operations required. In terms of processing power it is therefore better to split a big Kalman filter into smaller Kalman filters.

7.6 Solution selection

For application on a blimp, solution A gives the best trade-off between accuracy and processing power. The additional accuracy of solution C cannot be justified by the huge increase in processing power needed.

Another factor for consideration is that the attitude estimator of solution A will still operate optimally even when there are no GPS updates available. GPS updates can be temporarily lost due to the weak reception of satellite signals, which can be caused by a variety of reasons. The other solutions will be significantly affected by the absence of GPS updates.

7.7 Solution implementation

Figure 7-1 is a schematic layout of the solution implemented. On board the blimp are the OBC, IMU, GPS receiver and RF link. The OBC communicates via the RF link with a ground station computer.

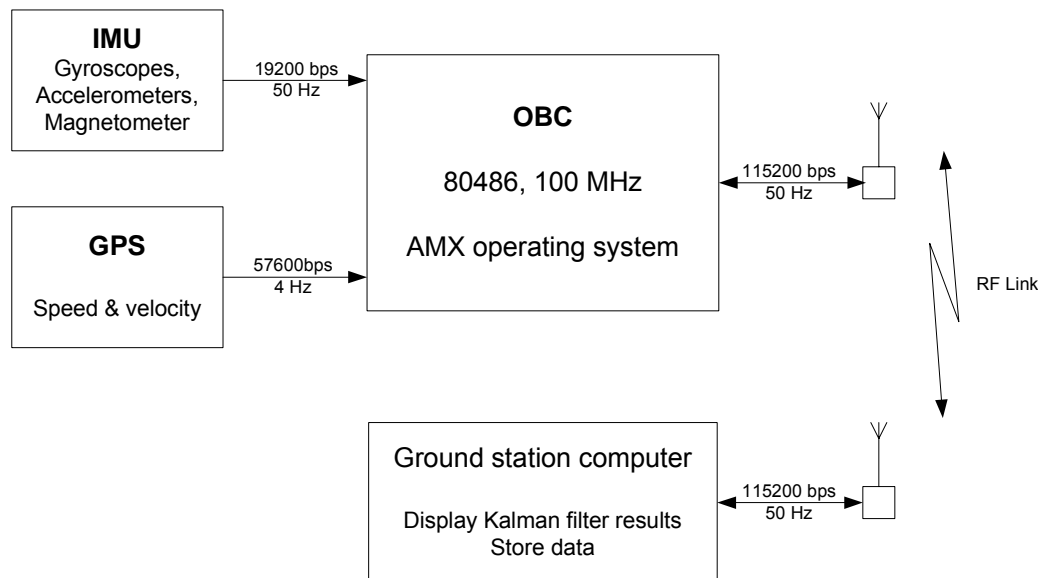


Figure 7-1: Schematic layout of the final solution.

For a real-time implementation of solution A, the two Kalman filters were written in C and linked with the AMX 86 real-time operating system (RTOS) to run on the OBC. An RTOS is needed to ensure that all timing-sensitive tasks are executed on time. With an ordinary operating system, housekeeping tasks may prevent that the time-sensitive tasks are executed on time, which may be critical for control applications. Matrix inversion routines were taken from Press (1992).

The OBC receives the gyroscope, accelerometer and magnetometer outputs every 20 ms from the CANsense board via a RS232 link operating at 19200 bps. GPS updates are received every 250 ms from the GPS module via a RS232 link operating at 57600 bps. The sensor and GPS data is packed, time stamped and sent over the RF-link to the ground station. All the data is processed onboard with the two Kalman filters. The Kalman filter results are sent over the RF-link to the ground station. Appendix G.2 has more information on the software running on the OBC.

The ground station software displays the IMU data and the Kalman filter results on screen in real time (see Figure 7-2). In addition, the ground station is used to command the OBC to perform certain tasks, e.g. Kalman filter initialisation, sensor self test functions, and magnetometer set/reset actions. All the received data can also be saved to a binary file for post processing. Software was developed to convert the binary file to comma separated values (CSV) format for easy access to the data.

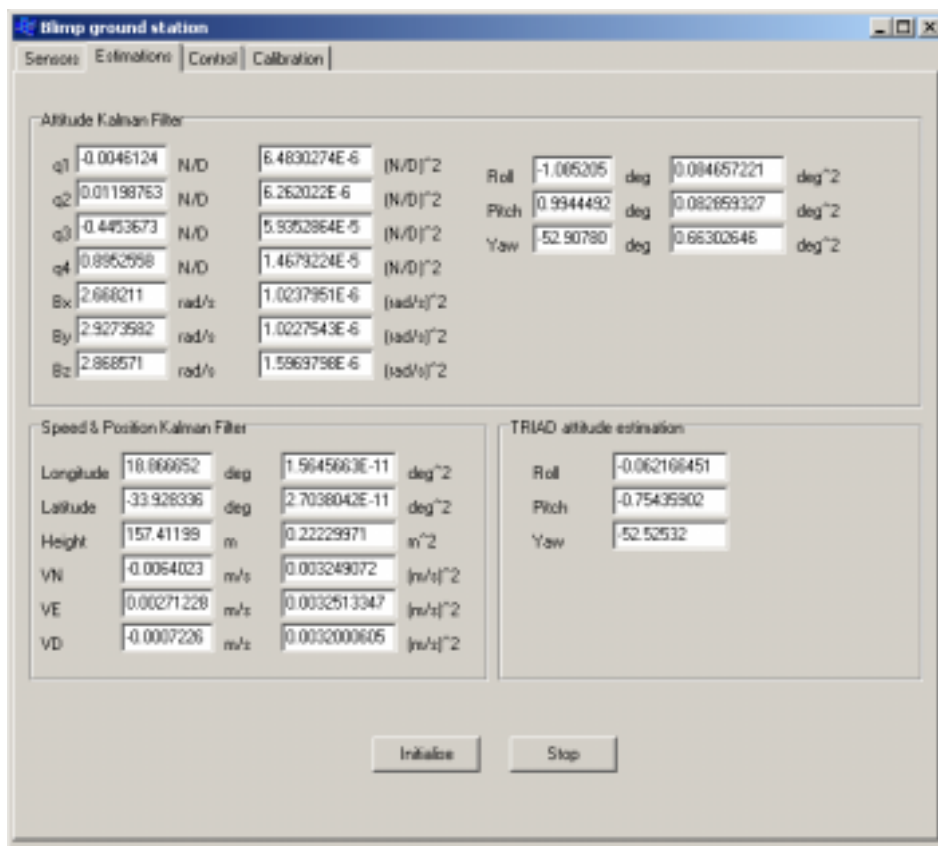


Figure 7-2: Ground station display of the Kalman filter results.

7.8 Solution results

The final solution was implemented on a flight test of about 12 minutes. During the whole flight the blimp was anchored with ropes to prevent the blimp from floating away. The blimp was launched about 25m in the air, where it floated for about 8 minutes, and was then pulled back for retrieval.

7.8.1 Attitude estimation

Figure 7-3 shows the attitude estimation in terms of roll, pitch and yaw angles. It is clear that most of the blimp's movement is in the yaw-angle. The roll and pitch angles rarely exceed 10 degrees to both sides.

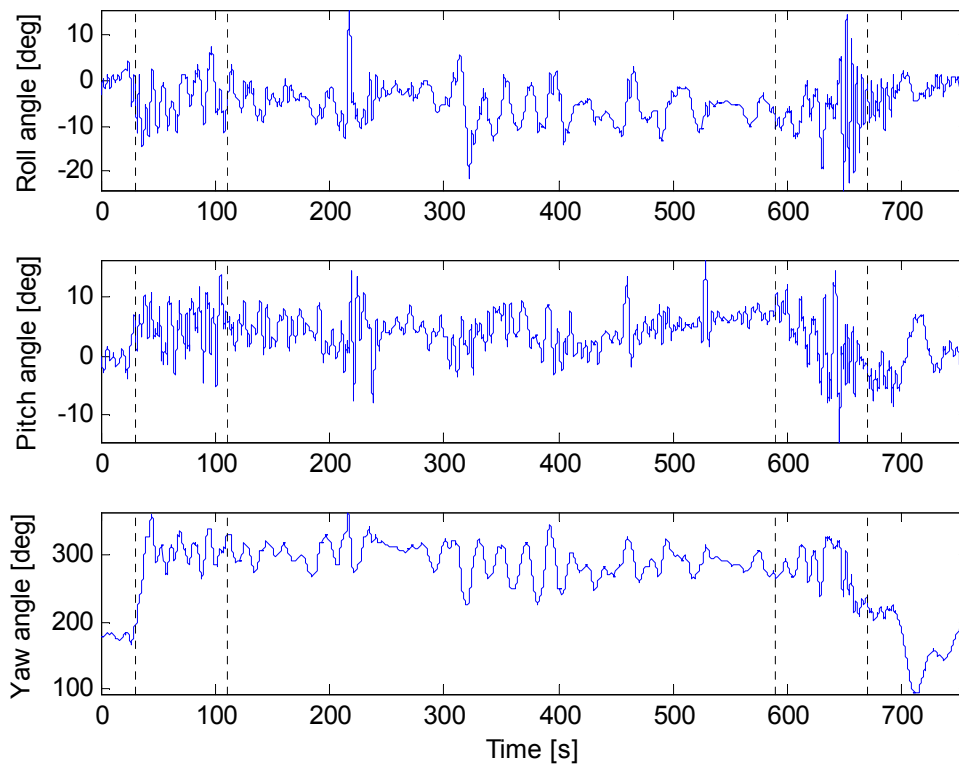


Figure 7-3: Attitude estimation in terms of roll, pitch and yaw angles during a blimp flight.

The attitude estimator performed better as expected during the whole period. As stated in Chapter 5, the dynamic acceleration that the blimp may experience during flight must be limited. Usually the biggest accelerations are experienced during the launch and retrieval processes, and one can expect the attitude estimator to misbehave during these times. However, as there was no reference attitude available, no real error evaluations could be made. The launch and retrieval times are indicated with vertical dotted lines in Figure 7-3 to Figure 7-7.

The only evident misbehaviour can be seen in the gyro bias estimates, as shown in Figure 7-4. Especially the Z-axis gyro has a clear increase in bias estimation during the launch, and decrease in bias estimation during the retrieval process. This misbehaviour in gyro bias estimation can be explained by inaccurate attitude estimations due to the following two reasons:

- Relatively big dynamic accelerations are present during the launch and retrieval process, preventing accurate attitude estimations.
- The magnetic field near the launching site was distorted by a long iron bar. When the magnetometer was near the bar (before launch and after retrieval), the bar caused magnetic offsets on the magnetometer readings, resulting in wrong attitude estimations. When the blimp was floating in the air, the iron bar was too far to have any significant effects.

The Kalman filter attempts to compensate for this wrong attitude estimation by changing the gyro bias values, in effect faking gyro measurements to rotate the blimp to the new attitude. However, the bias estimates quickly settle back after that, converging again to an accurate estimation. While the blimp is floating in the air there are still fluctuations present in the bias estimates, which can be attributed to imperfect calibration and modelling of all sensors, especially the magnetometer. In addition, temperature changes and accelerations have an influence on the gyro biases.

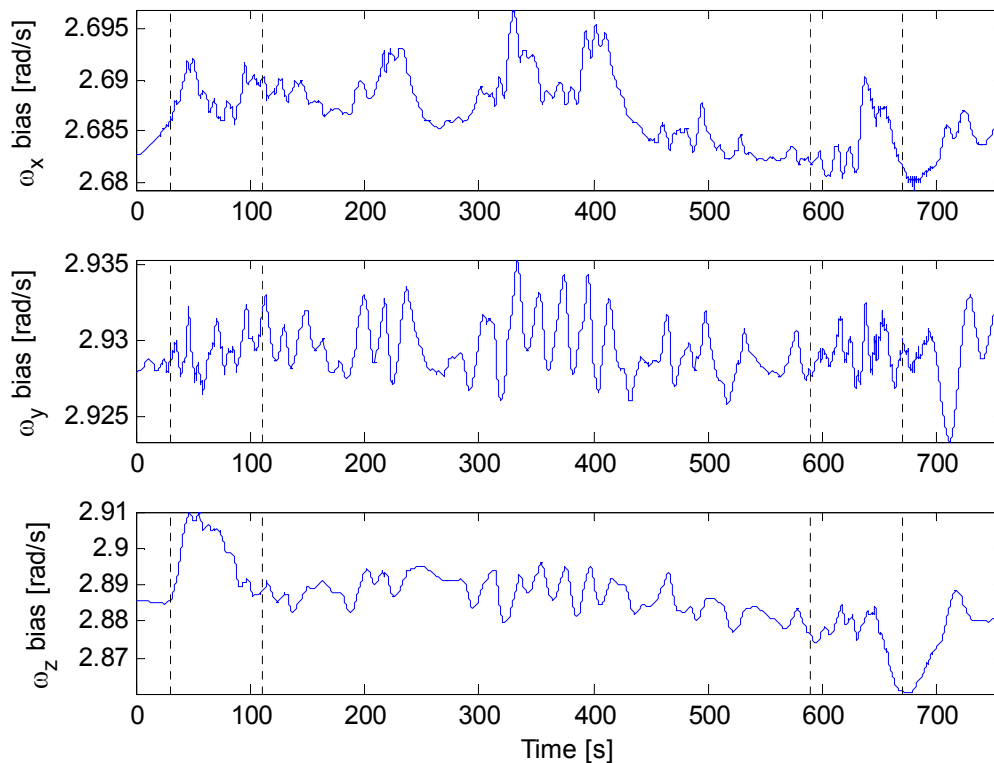


Figure 7-4: Gyro bias estimates during a blimp flight.

7.8.2 Velocity and position estimation

Figure 7-5 provides the position in ECEF geocentric coordinates of the blimp during the test flight. To get a clearer overview of the distance the blimp travelled, Figure 7-6 gives the NED displacements from the initial position. The velocity estimates of the blimp are given in Figure 7-7. Dotted lines indicate the GPS updates, while solid lines indicate the estimated position and velocity. One can see that the GPS updates and Kalman filter estimates fit very well. It signifies that the attitude estimation must be accurate as the attitude is used to convert accelerations in body axes to NED axes for velocity and position estimation.

As can be seen from the figures mentioned above, the estimates contain more high-speed manoeuvres. The GPS has a limited bandwidth due to the low update rate and cannot detect all of the blimp's dynamics. This clearly shows the advantage of using a combination of INS and GPS.

The launch (from 30 to 110 seconds) and retrieval (from 590 to 670 seconds) can clearly be seen in both the position (height) and velocity (down) estimates.

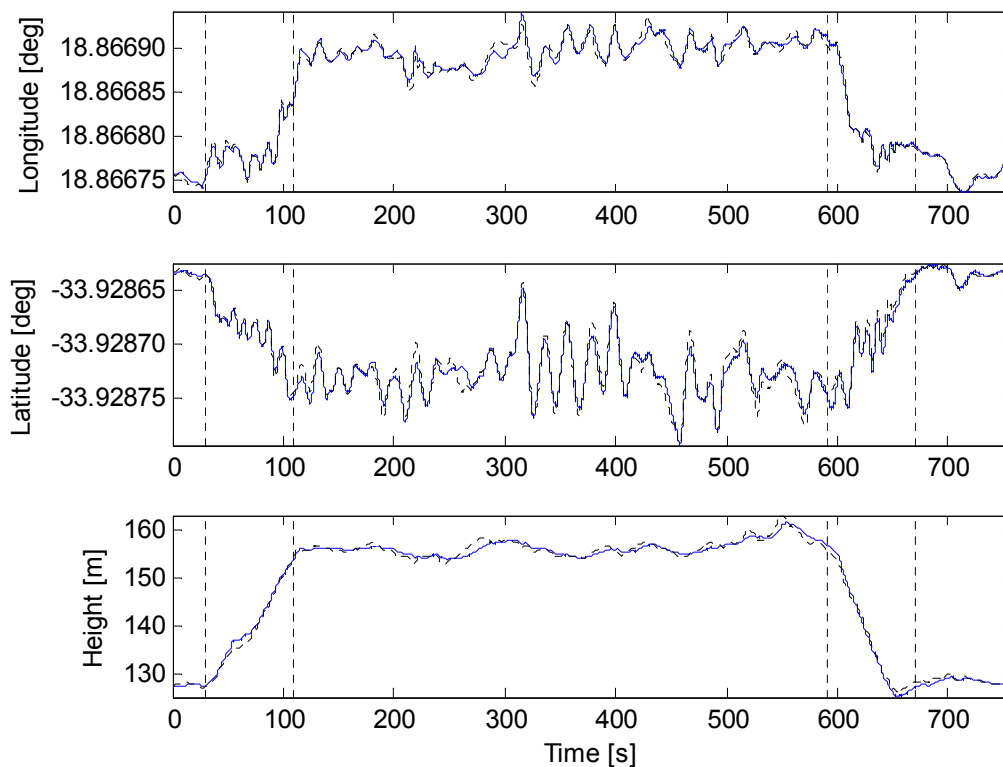


Figure 7-5: Position estimates of the blimp during a test flight.

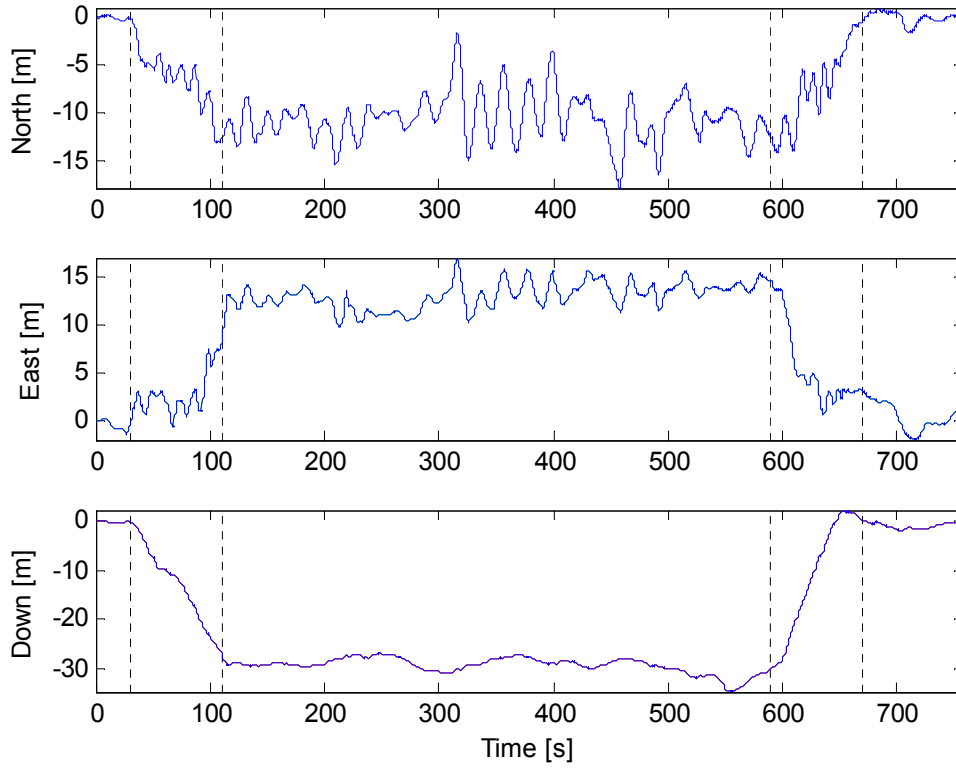


Figure 7-6: NED displacement estimates of the blimp during a test flight.

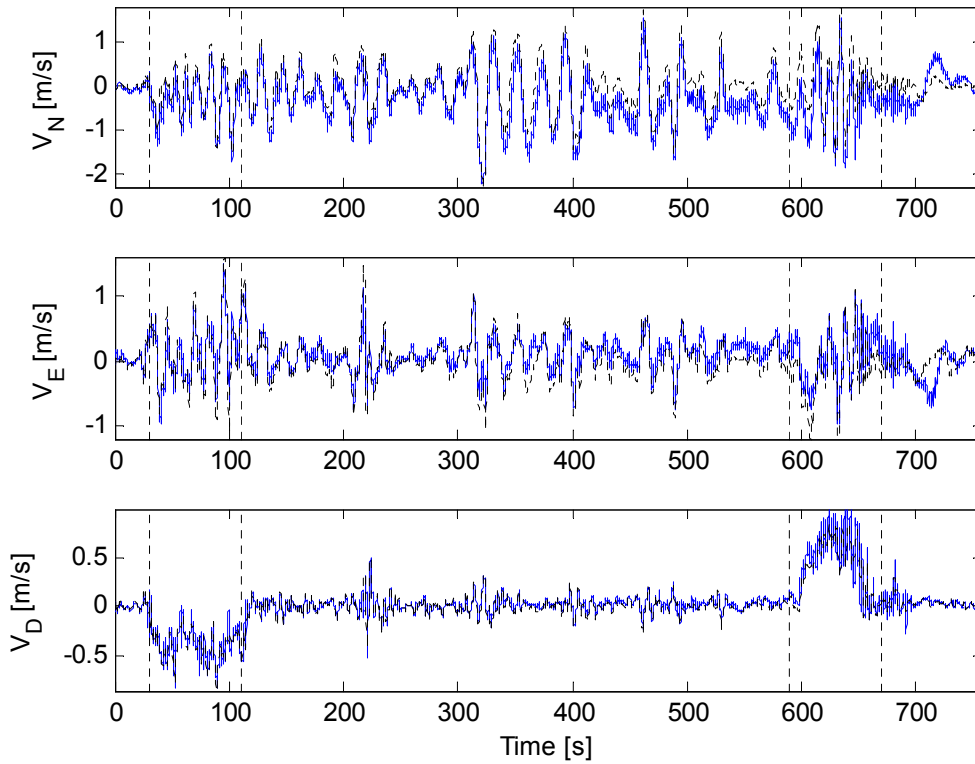


Figure 7-7: Velocity estimates of the blimp during a test flight.

Chapter 8: Summary and recommendations

8.1 Summary

A low cost attitude and heading reference system for an airship was made with the combination of MEMS devices and GPS updates. A study of different Kalman filters strategies was done, illustrating that a bigger Kalman filter does not always guarantee better state estimations. A trade-off between accuracy and processing power can be made by splitting the estimator in two parts: an attitude estimator and a position estimator.

For application on an airship, it was shown that the accelerometers of an IMU might be used as the gravity vector, with relevant noise figures applied to compensate for any dynamic acceleration the airship may experience.

8.2 Recommendations

8.2.1 IMU

While designing the IMU, no characteristics of the airships' dynamics were known. The bandwidth of the IMU was set to 100 Hz to ensure that all the dynamics were captured. In hindsight, the bandwidth could be limited to 10 Hz, which will result in lower noise on all sensors. In addition, the propagation rate of 50 Hz could be lowered to about 20 Hz without losing any significant resolution, resulting in a decrease of computational power needed.

The gyros are ratiometric devices, and therefore the reference output voltage can be used to compensate for any change in the supply voltage. This implies that more analogue signals need to be sampled, but the number of outputs will stay the same. In addition, a 12-bit A/D converter can be used instead of a 16-bit A/D converter to sample the signals. The 16-bit A/D converter was chosen to minimise the quantisation noise on the sampled signals. However, the sensor noise is much higher than the quantisation noise, implying that nothing will be gained by minimising the quantisation noise. A 12-bit A/D converter will be more practical to use.

As all MEMS devices are temperature dependant, temperature calibrations must be done to compensate for any temperature variations. Figure 8-1 shows the temperature together with the

output of the MEMS accelerometers and MEMS gyros during a bias drift test. The MEMS bias values tracks the temperature curve very well.

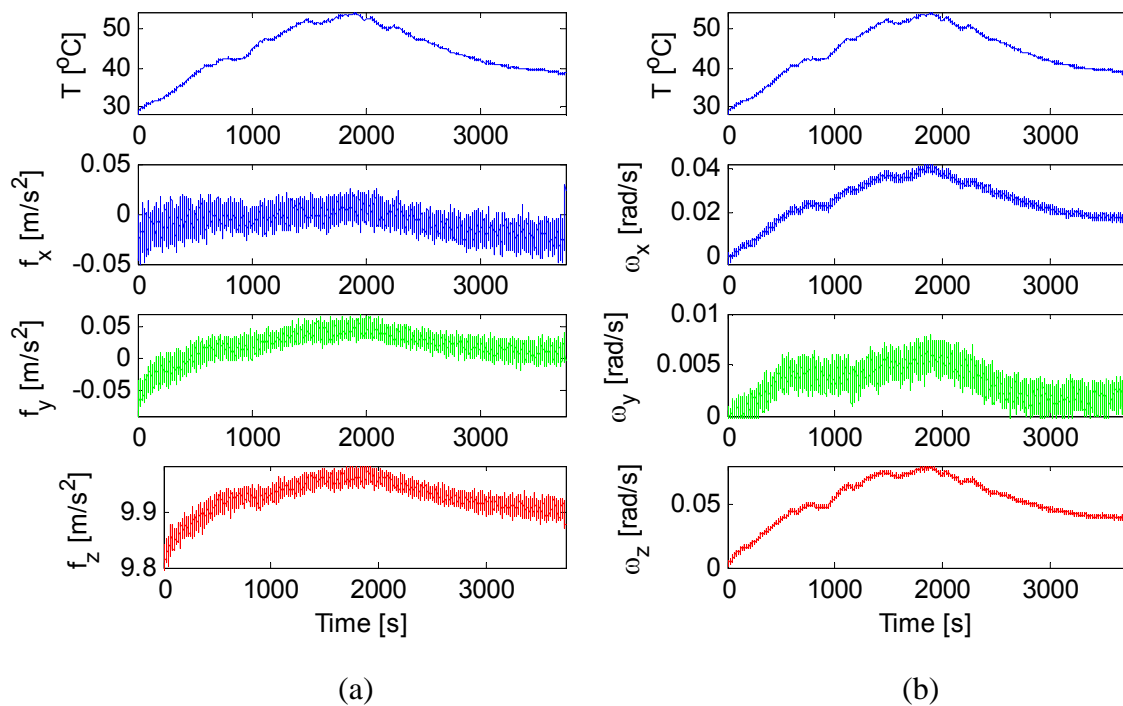


Figure 8-1: MEMS accelerometers (a) and gyros (b) temperature dependencies.

Simulations showed that attitude accuracies of about 1.88 degrees RMS could be obtained by using the magnetic and gravity vector. Magnetometer bias drifts limit the accuracy of the whole solution. The accuracy of the magnetometer can be increased when the magnetometer is periodically given a set/reset cycle to eliminate any bias drifts. However, it is important that the blimp does not experience any turn rates during the set/reset cycle, as this would degrade the magnetometer outputs. Because the set/reset action takes about one second to perform, a set/reset action can only be issued when the magnetic field is not changing at a fast rate. Even with the slow motion of the blimp, a change in magnetic field of up to 0.07 Gauss per second can be experienced. This is clear in Figure 8-2 where the change in magnetic field vector over one second is compared to the roll rate of the blimp. Performing a set/reset cycle periodically while the blimp is floating in the air can therefore cause more harm than benefit. The total roll rate of the blimp must be monitored before any set/reset actions are commanded.

8.2.2 RF link

The RF link is operating at a bit rate of 115.2 kbps. This fast bit rate is needed to send all data to the ground station, but causes transmission errors. Therefore, the bit rate must be lowered to increase reliability. Data could be stored on the OBC and downloaded at a later stage, resulting in less data to be transmitted.

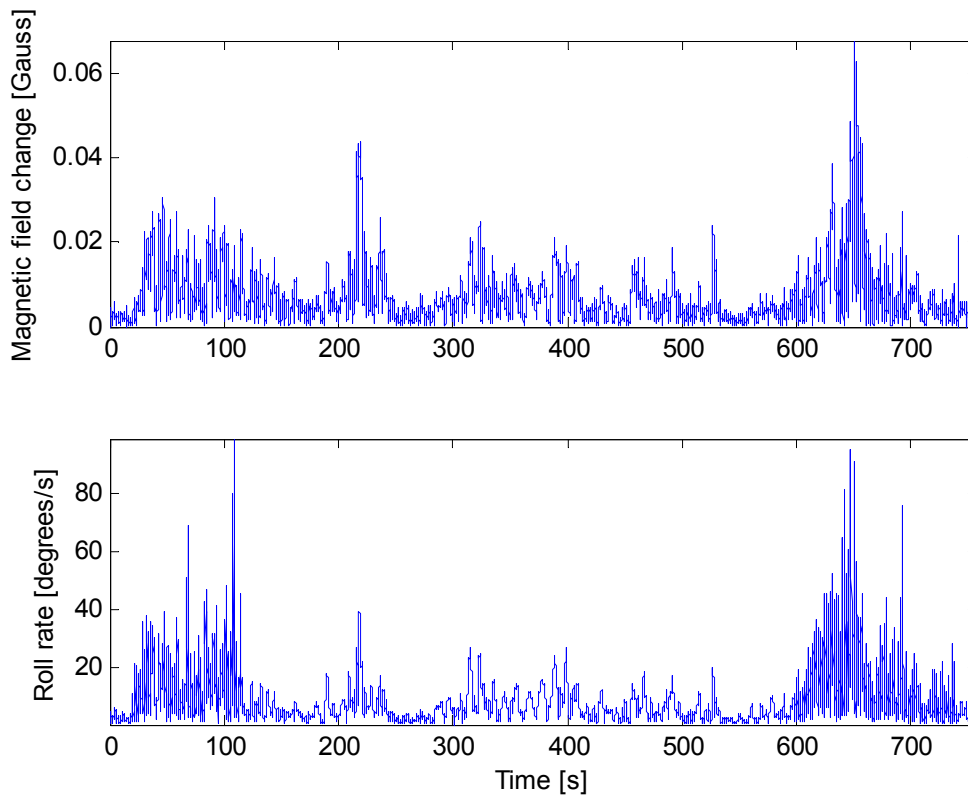


Figure 8-2: Change in magnetic field over a one-second period compared with the roll rate of the blimp.

8.2.3 Differential GPS

The RTCM data received from the internet tends to come through in big bursts of data. If these bursts of data are transmitted to the blimp, the RF-link can get overloaded. In addition, the RTCM data stream contains messages not used by the UBX receiver, wasting valuable bandwidth. A better method would be to decode the RTCM data stream at the ground station and only send the most recent messages to the blimp. El-Rabbany (2002) provides more information on the RTCM data stream.

8.2.4 Blimp control

For future work on the blimp, motors and actuators could be added for closed loop control. The modelling and control of a blimp has been researched quite extensively by others. See Khoury and Gillett (1999) for an example of the dynamic equations of an airship.

Bibliography

Allan, D.W., N. Ashby & C.C. Hodge 1997. *The science of timekeeping*. HP Application Note 1289.

Barbour, N.M., J.M. Elwell & R.H. Setterlund 1992. Inertial instruments: where to now? In: *Proceedings of the AIAA guidance, navigation and control conference, August 1992 Hilton Head*, pp. 566-574.

Barbour, N.M. & G. Schmidt 2001. Inertial sensor technology trends. *IEEE sensors journal*, 1 (4), pp. 332-339.

Bucy, R.S & P.D. Joseph 1968. *Filtering for stochastic processes with applications to guidance*. New York: Wiley-Interscience.

Carstens, N. 2005. *Development of a low-cost, low-weight flight control system for an electrically powered model helicopter*. Thesis (MScEng). University of Stellenbosch.

Cemenska, J. 2003. *Sensor modeling and Kalman filter applied to satellite attitude determination*. Thesis (MSc). University of California.

Chatfield, A.B. 1997. *Fundamentals of high accuracy inertial navigation*. Virginia: AIAA

Choukroun, D., I.Y. Bar-Itzhack & Y. Oshman 2002. A novel quaternion Kalman filter. In: *AIAA guidance, navigation, and control conference, August 2002, Monterey, California*.

El-Rabbany, A. 2002. *Introduction to GPS: the global positioning system*. Boston: Artech house.

Elfes, A., S.S Bueno, M. Bergerman & J.J.G. Ramos 1998. Project Aurora: development of an autonomous unmanned remote monitoring robotic airship. *Journal of the Brazilian Computer society*, 4 (3), pp. 70-78.

Franklin, G.F., J.D. Powell & M.L. Workman 1998. *Digital control of dynamic systems*. 3rd ed. California: Addison-Wesley.

Franklin, G.F., J.D. Powell & A. Emami-Naeini 2002. *Feedback control of dynamic systems*. 4th ed. New Jersey: Prentice Hall.

Friedland, B. 1978. Analysis strapdown navigation using quaternions. *IEEE transactions on aerospace and electronic systems*, 14 (5), pp. 764 – 768.

Gebre-Egziabher, D., G.H. Elkaim, J.D. Powell & B.W. Parkinson 2001. A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In: *Proceedings of the 8th international conference on integrated navigation systems, May 2001, St. Petersburg, Russia*, pp. 290-297.

Goldstein, H., C. Poole & J. Safko 2002. *Classical mechanics*. 3rd edition. San Francisco: Addison Wesley.

Groenewald, S. 2006. *Development of a rotary-wing test bed for autonomous flight*. Thesis (MScEng). University of Stellenbosch.

Hide, C.D. & T. Moore 2004. Low cost sensors, high quality integration. In: *Proceedings of NAV/AIS04, London, Nov 2004*.

Hide, C.D., T. Moore & M.J. Smith 2004. Multiple model Kalman filtering for GPS and low-cost INS integration. In: *Proceedings of ION GNSS 2004, Long Beach, California, September 2004*.

Hima S. & Y. Bestaoui 2002. Motion generation on trim trajectories for an autonomous underactuated airship. In: *Proceedings of the 4th international airship convention & exhibition, Cambridge July 2002*.

Hjortsmarker, N. 2005. *Experimental system for validating GPS/INS integration algorithms*. Thesis (MSc). Luleå University of Technology.

Hou, H. 2004. *Modeling inertial sensors errors using Allan variance*. Thesis (MSc). University of Calgary.

- Kalman, R.E. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of basic engineering*, 82 (series D), pp. 35-45.
- Khoury, G.A. & J.D. Gillett 1999. *Airship technology*. Cambridge: Cambridge University Press.
- King, A.D. 1998. Inertial navigation – forty years of evolution. *GEC review*, 13 (3), pp. 140-149.
- LaViola, J.J. 2003. A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In: *Proceedings of the 2003 American control conference, Denver June 2003*.
- Lawrence, A. 1998. *Modern inertial technology: navigation, guidance and control*. Second edition. New York: Springer.
- Lefferts, E.J., F.L. Markley & M.D. Shuster 1982. Kalman filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 5 (5), pp. 417 – 429.
- Lightsey, E.G. & J. Madsen 2003. Three-axis determination using global positioning system signal strength measurements. *Journal of guidance, control, and dynamics*, 26 (2), pp. 304 – 310.
- Madou, M.J. 2002. *Fundamentals of microfabrication: the science of miniaturization*. 2nd ed. Boca Raton: CRC Press.
- Markley, F.L. 1994. New quaternion attitude estimation method. *Journal of guidance, control, and dynamics*, 17 (2), pp. 407 – 409.
- Markley, F.L. 2003. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26 (2), pp. 311 – 317.
- Markley, F.L. 2004. Multiplicative versus additive filtering for spacecraft attitude determination. In: *Dynamics and control systems and structures in space (DCSSS) 6th conference, Jul 2004, Riomaggiore, Italy*, pp. 467 – 474.
- Milne, G.W. 2001. Simplifying vector rotations and coordination. In: *AIAA Atmospheric Flight Mechanics Conference and Exhibit, Aug 2001, Montreal, Canada*.

Peddle, I.K. 2005. *Autonomous flight of a model aircraft*. Thesis (MScEng). University of Stellenbosch.

Press, W.H. 1992. *Numerical recipes in C: the art of scientific computing*. 2nd edition. Cambridge: Cambridge University Press.

Psiaki, M.L. 1990. Three-axis attitude determination via Kalman filtering of magnetometer data. *Journal of guidance, control and dynamics*, 13 (3), pp. 506 – 514.

Ripka, P. 2001. *Magnetic sensors and magnetometers*. Boston: Artech House.

Sindle, T. (tsindle@sun.ac.za), 25 Jul 2005. RE: *Stellenbosch magnetic field*. E-mail to J. Bijker (jbijker@sun.ac.za).

Sorenson, H.W. 1970. Least-squares estimation: from Gauss to Kalman. *IEEE spectrum*, 7 (7), pp. 63-68.

Titterton, D.H. & J.L. Weston 1997. *Strapdown inertial navigation technology*. London: Peter Peregrinus.

Venter, J. 2005. *Development of an experimental tilt-wing VTOL unmanned aerial vehicle*. Thesis (MScEng). University of Stellenbosch.

Wabha, G. 1965. A Least squares estimate of satellite attitude. *SIAM review*, 7 (3), p. 490.

Wertz, J.R. 1978. *Spacecraft attitude determination and control*. Dordrecht: Kluwer.

Wittenburg, J. 1977. *Dynamics of Systems of Rigid Bodies*. Stuttgart: BG Teubner.

Appendix A: Vector and quaternion operations in matrix format

A.1 Vector multiplication

Vector multiplication can be written in matrix format as:

$$\bar{\mathbf{a}} \times \bar{\mathbf{b}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (\text{A.1})$$

$$\bar{\mathbf{a}} \times \bar{\mathbf{b}} = \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (\text{A.2})$$

A.2 Quaternion multiplication

Quaternion multiplication can be written in matrix format as:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_4 & a_3 & -a_2 & a_1 \\ -a_3 & a_4 & a_1 & a_2 \\ a_2 & -a_1 & a_4 & a_3 \\ -a_1 & -a_2 & -a_3 & a_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (\text{A.3})$$

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} b_4 & -b_3 & b_2 & b_1 \\ b_3 & b_4 & -b_1 & b_2 \\ -b_2 & b_1 & b_4 & b_3 \\ -b_1 & -b_2 & -b_3 & b_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (\text{A.4})$$

The methods above differ from historical multiplication conventions. Markley (2004) shows that by using the multiplication as defined above, the direction cosine matrix in terms of the product of two quaternions can be written as:

$$\mathbf{A}(\mathbf{a} \otimes \mathbf{b}) = \mathbf{A}(\mathbf{a})\mathbf{A}(\mathbf{b}) \quad (\text{A.5})$$

With historical multiplication conventions the order on the right hand side of equation (A.5) must be reversed.

Appendix B: Quaternion operations

B.1 Convert a quaternion to Euler angles

Because a quaternion cannot be easily interpreted, it is necessary to convert the quaternion to Euler angles.

Comparing the direction cosine matrix in terms of Euler angles and a quaternion, the following conversions from a quaternion to Euler angles can be made:

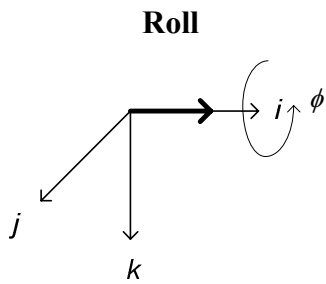
$$\phi = \tan^{-1} \left(\frac{2(q_4 q_1 + q_2 q_3)}{q_4^2 - q_1^2 - q_2^2 + q_3^2} \right) \quad (\text{B.1})$$

$$\theta = \sin^{-1}(-2(q_1 q_3 - q_4 q_2)) \quad (\text{B.2})$$

$$\psi = \tan^{-1} \left(\frac{2(q_4 q_3 + q_1 q_2)}{q_4^2 + q_1^2 - q_2^2 - q_3^2} \right) \quad (\text{B.3})$$

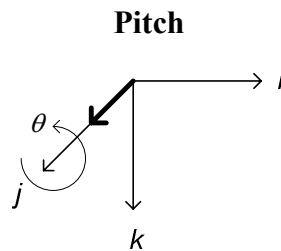
B.2 Convert Euler angles to a quaternion

To convert Euler angles to a quaternion, the whole rotation can be seen as three individual rotations around a specific axis. Each rotation is written in terms of a quaternion:



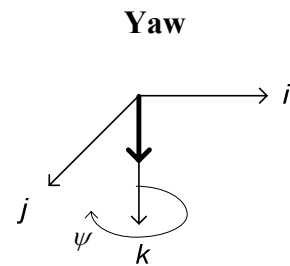
$$\mathbf{q}_{\text{roll}} = \begin{bmatrix} \sin \frac{\phi}{2} \\ 0 \\ 0 \\ \cos \frac{\phi}{2} \end{bmatrix}$$

(B.4)



$$\mathbf{q}_{\text{pitch}} = \begin{bmatrix} 0 \\ \sin \frac{\theta}{2} \\ 0 \\ \cos \frac{\theta}{2} \end{bmatrix}$$

(B.5)



$$\mathbf{q}_{\text{yaw}} = \begin{bmatrix} 0 \\ 0 \\ \sin \frac{\psi}{2} \\ \cos \frac{\psi}{2} \end{bmatrix}$$

(B.6)

Multiplying the quaternions in the order yaw, pitch, roll (Euler 3-2-1), the following result is obtained:

$$\mathbf{q} = \begin{bmatrix} \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \end{bmatrix} \quad (\text{B.7})$$

B.3 Extracting a quaternion out of the direction cosine matrix

The absolute values of the quaternion elements can be found by taking the necessary addition and/or subtracting of the elements on the diagonal of the direction cosine matrix:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2} \sqrt{\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} A_{11} \\ A_{22} \\ A_{33} \end{bmatrix} + 1} = \frac{1}{2} \sqrt{\begin{bmatrix} 4q_1^2 \\ 4q_2^2 \\ 4q_3^2 \\ 4q_4^2 \end{bmatrix}} \quad (\text{B.8})$$

The absolute value is found due to the positive square root. To get the correct sign of each quaternion, one element must be selected as the pivot for calculating the other elements. For best accuracy, usually the largest quaternion element is selected as the pivot.

Pivot	q_1	q_2	q_3	q_4
Equation	$\mathbf{q} = \pm \begin{bmatrix} q_1 \\ \frac{a_{12} + a_{21}}{4q_1} \\ \frac{a_{13} + a_{31}}{4q_1} \\ \frac{a_{23} - a_{32}}{4q_1} \end{bmatrix}$ <p>(B.9)</p>	$\mathbf{q} = \pm \begin{bmatrix} \frac{a_{12} + a_{21}}{4q_2} \\ q_2 \\ \frac{a_{23} + a_{32}}{4q_2} \\ \frac{a_{31} - a_{13}}{4q_2} \end{bmatrix}$ <p>(B.10)</p>	$\mathbf{q} = \pm \begin{bmatrix} \frac{a_{13} + a_{31}}{4q_3} \\ \frac{a_{23} + a_{32}}{4q_3} \\ q_3 \\ \frac{a_{12} - a_{21}}{4q_3} \end{bmatrix}$ <p>(B.11)</p>	$\mathbf{q} = \pm \begin{bmatrix} \frac{a_{23} - a_{32}}{4q_4} \\ \frac{a_{31} - a_{13}}{4q_4} \\ \frac{a_{12} - a_{21}}{4q_4} \\ q_4 \end{bmatrix}$ <p>(B.12)</p>

Note that $-\mathbf{q}$ gives the same rotation as \mathbf{q} . This means that when calculating the quaternion as above, there may be sudden sign difference over time. To keep the quaternion smooth over time, a sudden sign difference may be eliminated by comparing the sign of the current pivot element with the corresponding previous quaternion element, and multiplying the whole quaternion with -1 if needed.

Appendix C: Matrix trace properties

The trace of an $N \times N$ matrix \mathbf{A} is defined as the sum of all elements on the diagonal:

$$Tr(\mathbf{A}) = \sum_{i=1}^N A_{ii} \quad (\text{C.1})$$

As the transpose of a matrix will not change the diagonal elements, the trace of a matrix will be the same as the trace of the transpose of the matrix:

$$Tr(\mathbf{A}^T) = Tr(\mathbf{A}) \quad (\text{C.2})$$

As matrix addition is commutative, the trace of a matrix will also be commutative.

$$Tr(\mathbf{A} + \mathbf{B}) = Tr(\mathbf{A}) + Tr(\mathbf{B}) \quad (\text{C.3})$$

Because the trace of a matrix \mathbf{A} is a scalar, taking the derivative of the trace with respect to another $N \times M$ matrix will result in a new matrix where each element in the matrix will be the derivative of the trace with respect to that element.

$$\frac{\partial Tr(\mathbf{XA})}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial Tr(\mathbf{XA})}{\partial X_{11}} & \frac{\partial Tr(\mathbf{XA})}{\partial X_{12}} & \dots & \frac{\partial Tr(\mathbf{XA})}{\partial X_{1M}} \\ \frac{\partial Tr(\mathbf{XA})}{\partial X_{21}} & \frac{\partial Tr(\mathbf{XA})}{\partial X_{22}} & \dots & \frac{\partial Tr(\mathbf{XA})}{\partial X_{2M}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial Tr(\mathbf{XA})}{\partial X_{N1}} & \frac{\partial Tr(\mathbf{XA})}{\partial X_{N2}} & \dots & \frac{\partial Tr(\mathbf{XA})}{\partial X_{NM}} \end{bmatrix} \quad (\text{C.4})$$

Using matrix multiplication, and assuming \mathbf{X} is an $N \times M$ matrix and \mathbf{A} is an $M \times N$ matrix, the trace of \mathbf{XA} can be written as:

$$Tr(\mathbf{XA}) = \sum_{i=1}^M X_{1i} A_{i1} + \sum_{i=1}^M X_{2i} A_{i2} + \dots + \sum_{i=1}^M X_{Ni} A_{iN} \quad (\text{C.5})$$

Therefore, equation (C.4) can be simplified to:

$$\frac{\partial Tr(\mathbf{XA})}{\partial \mathbf{X}} = \begin{bmatrix} A_{11} & A_{21} & \dots & A_{M1} \\ A_{12} & A_{22} & \dots & A_{M2} \\ \dots & \dots & \dots & \dots \\ A_{1M} & A_{2M} & \dots & A_{MN} \end{bmatrix} = \mathbf{A}^T \quad (\text{C.6})$$

To compute the trace of a matrix \mathbf{XAX}^T , assume \mathbf{X} is an $N \times M$ matrix and \mathbf{A} is a square $M \times M$ matrix, so that \mathbf{XAX}^T will be an $N \times N$ matrix.

The trace of the matrix can be found to be:

$$Tr(\mathbf{XAX}^T) = \sum_{j=1}^M \sum_{i=1}^M X_{1i} A_{ij} X_{1j} + \sum_{j=1}^M \sum_{i=1}^M X_{2i} A_{ij} X_{2j} + \dots + \sum_{j=1}^M \sum_{i=1}^M X_{Ni} A_{ij} X_{Nj} \quad (\text{C.7})$$

Taking the partial derivative of equation (C.7) to \mathbf{X} :

$$\begin{aligned} \frac{\partial Tr(\mathbf{XAX}^T)}{\partial \mathbf{X}} &= \begin{bmatrix} \sum_{i=1}^M X_{1i} A_{i1} + \sum_{j=1}^M A_{1j} X_{1j} & \sum_{i=1}^M X_{1i} A_{i2} + \sum_{j=1}^M A_{2j} X_{1j} & \dots & \sum_{i=1}^M X_{1i} A_{iN} + \sum_{j=1}^M A_{Nj} X_{1j} \\ \sum_{i=1}^M X_{2i} A_{i1} + \sum_{j=1}^M A_{1j} X_{2j} & \sum_{i=1}^M X_{2i} A_{i2} + \sum_{j=1}^M A_{2j} X_{2j} & \dots & \sum_{i=1}^M X_{2i} A_{iN} + \sum_{j=1}^M A_{Nj} X_{2j} \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^M X_{Ni} A_{i1} + \sum_{j=1}^M A_{1j} X_{Nj} & \sum_{i=1}^M X_{Ni} A_{i2} + \sum_{j=1}^M A_{2j} X_{Nj} & \dots & \sum_{i=1}^M X_{Ni} A_{iN} + \sum_{j=1}^M A_{Nj} X_{Nj} \end{bmatrix} \quad (\text{C.8}) \\ &= \mathbf{XA} + \mathbf{XA}^T \end{aligned}$$

Appendix D: Orthogonal matrix properties

Matrix \mathbf{A} is said to be orthogonal if and only if

$$\mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{D.1}$$

The inverse of a matrix is defined as

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \tag{D.2}$$

Comparing equations (D.1) and (D.2), it follows that the transpose of an orthogonal matrix is identical to the inverse of the matrix:

$$\mathbf{A}^T = \mathbf{A}^{-1} \tag{D.3}$$

To test if a matrix is orthogonal, the left hand side of equation (D.1) must be evaluated. If it is equal to the identity matrix, the matrix is orthogonal.

Appendix E: Noise transforms

Suppose we have a set of (non-linear) equations as functions of a set of states (\mathbf{x}):

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad (\text{E.1})$$

Assume the noise on the states \mathbf{x} are white Gaussian noise with known covariance matrix of \mathbf{P}_x . The noise figures must be transformed via equation (E.1) to give a reasonable estimate of the noise on the output \mathbf{y} . However, the transformed noise figures are only estimates for non-linear systems.

If the noise errors are sufficiently small and $\mathbf{f}(\mathbf{x})$ is differentiable, the noise errors in \mathbf{y} can be estimated by using first-order Taylor series. First construct the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_j} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_j} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial y_i}{\partial x_1} & \frac{\partial y_i}{\partial x_2} & \cdots & \frac{\partial y_i}{\partial x_j} \end{bmatrix} \quad (\text{E.2})$$

The noise errors in \mathbf{y} can then be estimated by:

$$\delta \mathbf{y} = \mathbf{J} \delta \mathbf{x} \quad (\text{E.3})$$

The noise covariance matrix \mathbf{P}_y can be calculated as follows:

$$\begin{aligned} \mathbf{P}_y &= E \{ \delta \mathbf{y} \delta \mathbf{y}^T \} \\ &= E \{ \mathbf{J} \delta \mathbf{x} \delta \mathbf{x}^T \mathbf{J}^T \} \\ &= \mathbf{J} E \{ \delta \mathbf{x} \delta \mathbf{x}^T \} \mathbf{J}^T \\ &= \mathbf{J} \mathbf{P}_x \mathbf{J}^T \end{aligned} \quad (\text{E.4})$$

E.1 Quaternion noise to roll, pitch & yaw noise

The roll, pitch and yaw angles in terms of a quaternion is given in equations (B.1) to (B.3). Taking the partial derivative of each equation to each of the quaternion elements gives:

$$\frac{\partial \phi}{\partial q_1} = \frac{2q_4(q_4^2 - q_1^2 - q_2^2 + q_3^2) + 4q_1(q_4q_1 + q_2q_3)}{(q_4^2 - q_1^2 - q_2^2 + q_3^2)^2 + 4(q_4q_1 + q_2q_3)^2} \quad (\text{E.5})$$

$$\frac{\partial \phi}{\partial q_2} = \frac{2q_3(q_4^2 - q_1^2 - q_2^2 + q_3^2) + 4q_2(q_4q_1 + q_2q_3)}{(q_4^2 - q_1^2 - q_2^2 + q_3^2)^2 + 4(q_4q_1 + q_2q_3)^2} \quad (\text{E.6})$$

$$\frac{\partial \phi}{\partial q_3} = \frac{2q_2(q_4^2 - q_1^2 - q_2^2 + q_3^2) - 4q_3(q_4q_1 + q_2q_3)}{(q_4^2 - q_1^2 - q_2^2 + q_3^2)^2 + 4(q_4q_1 + q_2q_3)^2} \quad (\text{E.7})$$

$$\frac{\partial \phi}{\partial q_4} = \frac{2q_1(q_4^2 - q_1^2 - q_2^2 + q_3^2) - 4q_4(q_4q_1 + q_2q_3)}{(q_4^2 - q_1^2 - q_2^2 + q_3^2)^2 + 4(q_4q_1 + q_2q_3)^2} \quad (\text{E.8})$$

$$\frac{\partial \theta}{\partial q_1} = \frac{-2q_3}{\sqrt{1 - 4(q_1q_3 - q_4q_2)^2}} \quad (\text{E.9})$$

$$\frac{\partial \theta}{\partial q_2} = \frac{2q_4}{\sqrt{1 - 4(q_1q_3 - q_4q_2)^2}} \quad (\text{E.10})$$

$$\frac{\partial \theta}{\partial q_3} = \frac{-2q_1}{\sqrt{1 - 4(q_1q_3 - q_4q_2)^2}} \quad (\text{E.11})$$

$$\frac{\partial \theta}{\partial q_4} = \frac{2q_2}{\sqrt{1 - 4(q_1q_3 - q_4q_2)^2}} \quad (\text{E.12})$$

$$\frac{\partial \psi}{\partial q_1} = \frac{2q_2(q_4^2 + q_1^2 - q_2^2 - q_3^2) - 4q_1(q_4q_3 + q_1q_2)}{(q_4^2 + q_1^2 - q_2^2 - q_3^2)^2 + 4(q_4q_3 + q_1q_2)^2} \quad (\text{E.13})$$

$$\frac{\partial \psi}{\partial q_2} = \frac{2q_1(q_4^2 + q_1^2 - q_2^2 - q_3^2) + 4q_2(q_4q_3 + q_1q_2)}{(q_4^2 + q_1^2 - q_2^2 - q_3^2)^2 + 4(q_4q_3 + q_1q_2)^2} \quad (\text{E.14})$$

$$\frac{\partial \psi}{\partial q_3} = \frac{2q_4(q_4^2 + q_1^2 - q_2^2 - q_3^2) + 4q_3(q_4q_3 + q_1q_2)}{(q_4^2 + q_1^2 - q_2^2 - q_3^2)^2 + 4(q_4q_3 + q_1q_2)^2} \quad (\text{E.15})$$

$$\frac{\partial \psi}{\partial q_4} = \frac{2q_3(q_4^2 + q_1^2 - q_2^2 - q_3^2) - 4q_4(q_4q_3 + q_1q_2)}{(q_4^2 + q_1^2 - q_2^2 - q_3^2)^2 + 4(q_4q_3 + q_1q_2)^2} \quad (\text{E.16})$$

E.2 ECEF rectangular noise to ECEF geocentric noise

The measurement noise or inaccuracies of the GPS position are given in ECEF rectangular format, units in meters. This noise must be converted to ECEF geodetic noise in order to use it in the Kalman filter.

Taking the partial derivative of equations (2.1) to (2.3) to X, Y and Z, and simplifying using equations (2.4) to (2.6) results as follows:

$$\frac{\partial \varphi}{\partial X} = \frac{-Y}{X^2 + Y^2} = \frac{-\sin \varphi}{(R+h) \cos \lambda} \quad (\text{E.17})$$

$$\frac{\partial \varphi}{\partial Y} = \frac{X}{X^2 + Y^2} = \frac{\cos \varphi}{(R+h) \cos \lambda} \quad (\text{E.18})$$

$$\frac{\partial \varphi}{\partial Z} = 0 \quad (\text{E.19})$$

$$\frac{\partial \lambda}{\partial X} = \frac{-XZ}{(X^2 + Y^2 + Z^2)\sqrt{X^2 + Y^2}} = \frac{-\cos \varphi \sin \lambda}{R+h} \quad (\text{E.20})$$

$$\frac{\partial \lambda}{\partial Y} = \frac{-YZ}{(X^2 + Y^2 + Z^2)\sqrt{X^2 + Y^2}} = \frac{-\cos \lambda \sin \varphi}{R+h} \quad (\text{E.21})$$

$$\frac{\partial \lambda}{\partial Z} = \frac{X^2 + Y^2}{(X^2 + Y^2 + Z^2)\sqrt{X^2 + Y^2}} = \frac{\cos \lambda}{R+h} \quad (\text{E.22})$$

$$\frac{\partial h}{\partial X} = \frac{X}{\sqrt{X^2 + Y^2 + Z^2}} = \cos \lambda \cos \varphi \quad (\text{E.23})$$

$$\frac{\partial h}{\partial Y} = \frac{Y}{\sqrt{X^2 + Y^2 + Z^2}} = \cos \lambda \sin \varphi \quad (\text{E.24})$$

$$\frac{\partial h}{\partial Z} = \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} = \sin \lambda \quad (\text{E.25})$$

Appendix F: Hardware

F.1 CANsense board

The CANsense board was developed by Groenewald (2006). It consists of a PIC microprocessor, A/D converter and anti-aliasing filters. The data can be transmitted via the CAN-bus or RS232 port. As the OBC had no CAN-bus, the RS232 port was used. The code was adapted to sample at 800 Hz, filter the samples digitally and output the filtered values at 50 Hz over the RS232 port. The digital filter is a second order Butterworth filter with a -3dB cut off frequency at 10 Hz. The transfer function of the digital filter is:

$$H(z) = \frac{0.00146 + 0.002921z^{-1} + 0.00146z^{-2}}{1 - 1.889z^{-1} + 0.8949z^{-2}} \quad (\text{F.1})$$

Equation (F.1) can be scaled and implemented as a difference equation:

$$32768y(n) = 48u(n) + 96u(n-1) + 48u(n-2) + 61900y(n-1) - 29323y(n-2) \quad (\text{F.2})$$

In total twelve analogue channels can be sampled:

- Eight 16-bit channels. These signals are sampled by the external A/D converter. The reference voltage for the A/D converter is 4.096V, and as most sensors can have values up to 5V, some signal scaling must be done (see section F.2 for more information). Eight second-order Butterworth anti-aliasing filters are provided on board. The transfer function of the analogue filters are:

$$H(s) = \frac{\left(\frac{1}{RC}\right)^2}{\left(s + \frac{1}{RC}\right)^2} \quad (\text{F.3})$$

With R as 33k Ω and C as 22nF, the -3dB cut off frequency is at 100 Hz.

- Four 10-bit channels. These signals are sampled by the PIC microprocessor. The reference voltage for the PIC A/D converter is 5.0V, and therefore no signal scaling has to be done. No anti-aliasing filters are provided for these channels and must be done elsewhere.

The schematics are divided in two parts: the analogue part (Figure F- 1) and the digital part (Figure F- 2).

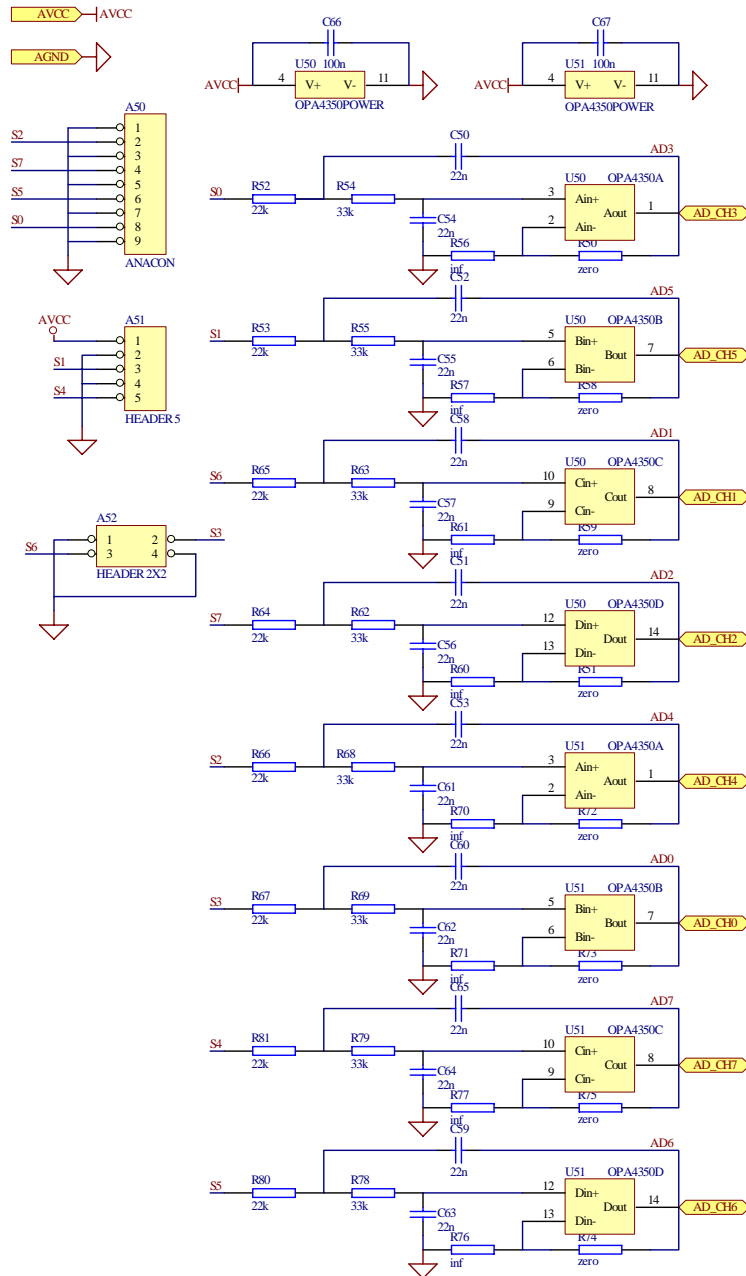


Figure F- 1: CANsense analogue schematics.

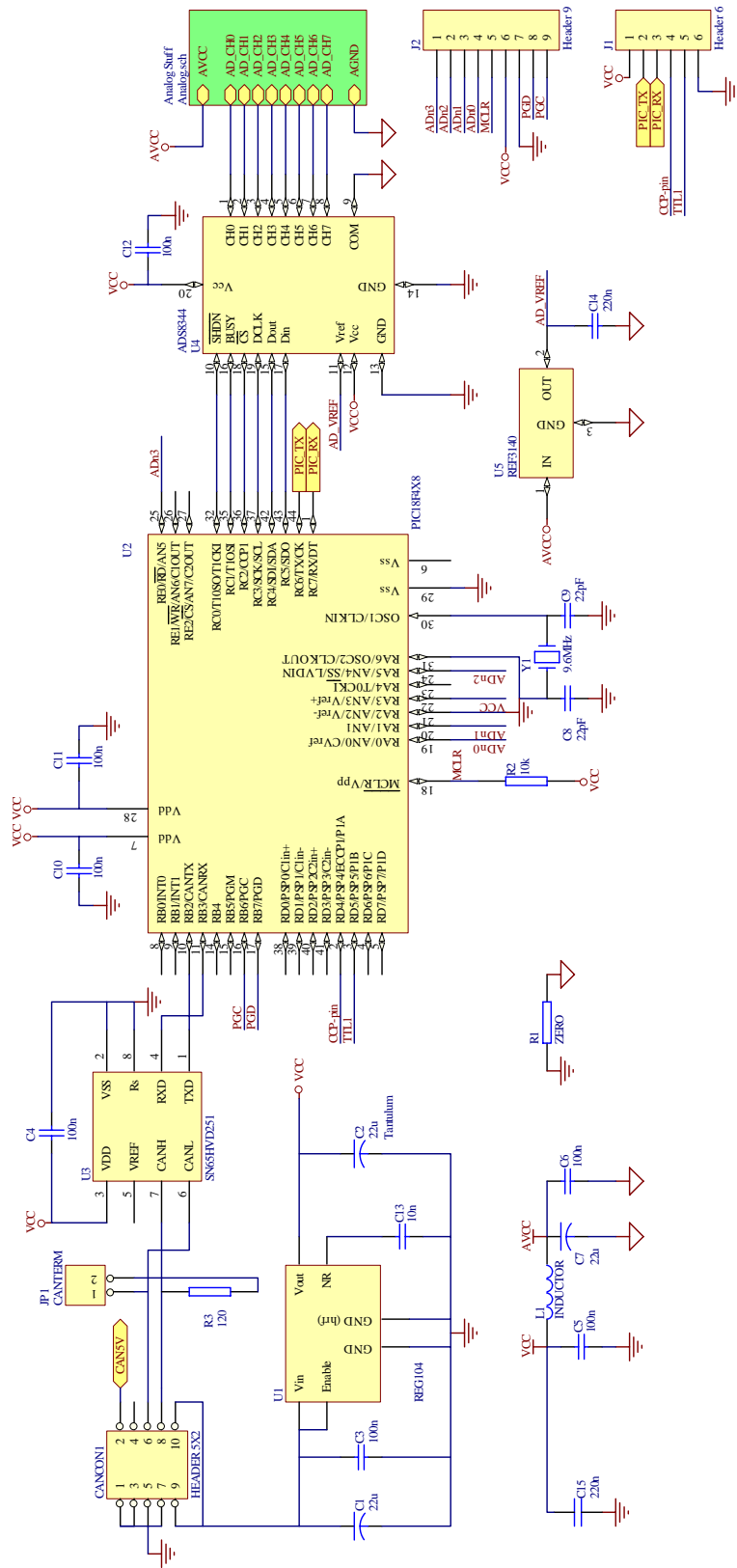


Figure F- 2: CANsense digital schematics.

F.2 CANsenseIMU

A PCB was designed to interface an IMU to the CANsense board. It consists of three gyros, three accelerometers and a three-axis magnetometer. See Figure F- 3 for the main CANsenseIMU schematic.

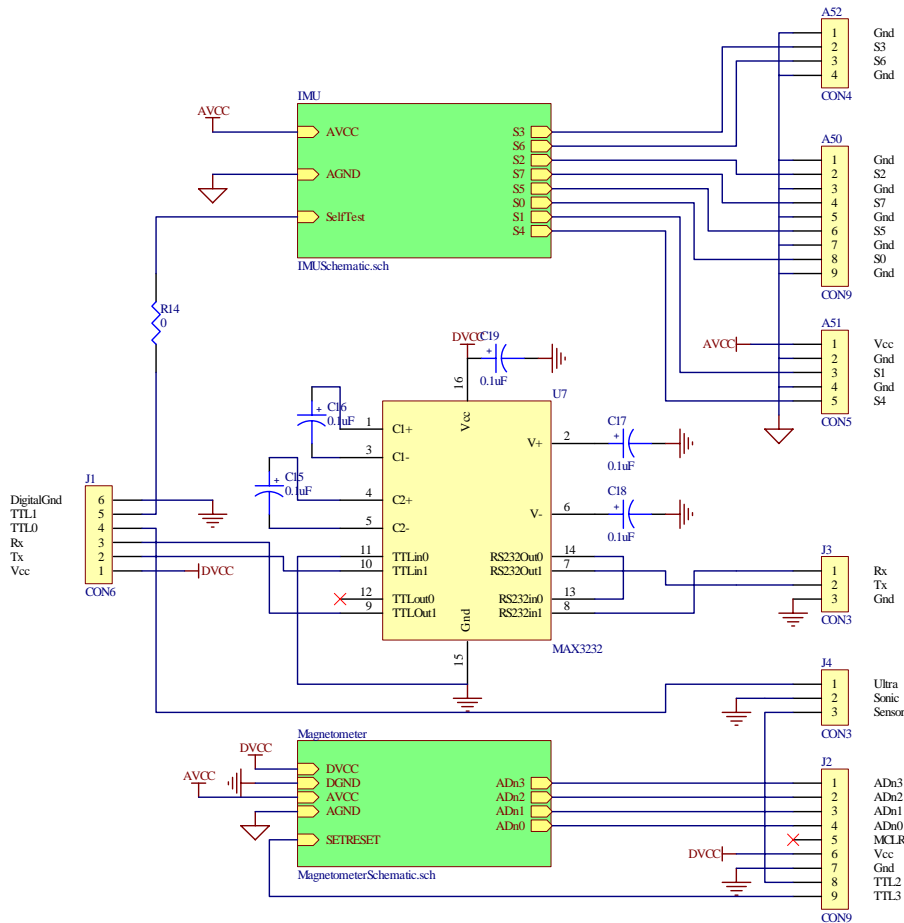


Figure F- 3: CANsenseIMU main schematic.

F.2.1 IMU module

The IMU consists of three perpendicular boards (called single axis IMU boards) which hosts a gyro and an accelerometer. The gyro measures turn rates in one axis, while the accelerometer can be either a single axis or dual axes accelerometer. Three single axis IMU boards are combined to form the IMU module.

Figure F- 4 contains the schematics of a single axis IMU board and Figure F- 5 the schematics of the complete IMU module.

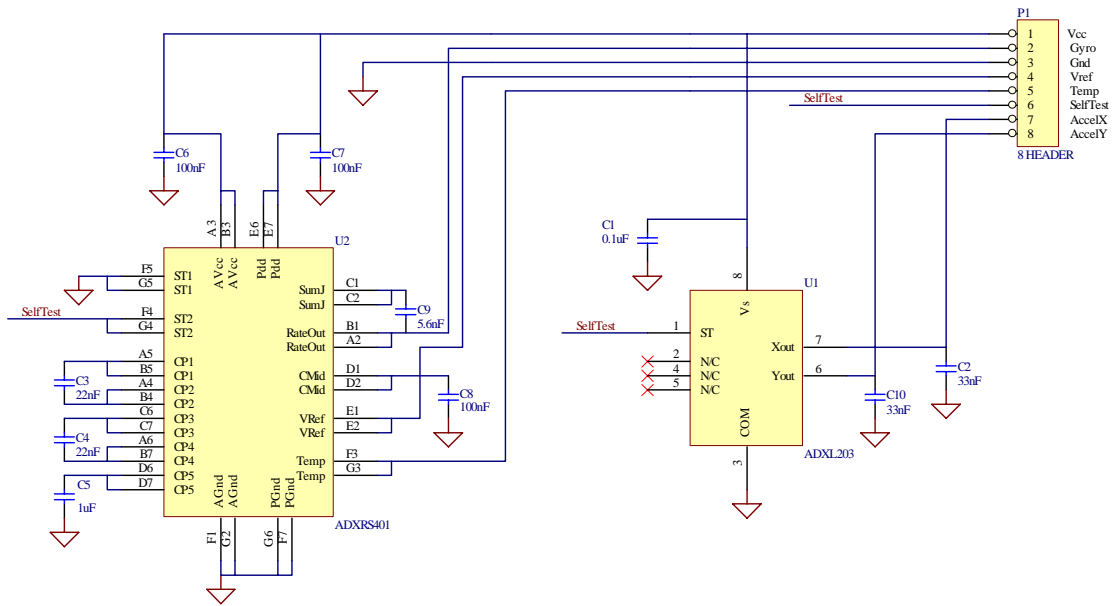


Figure F- 4: Single axis IMU board.

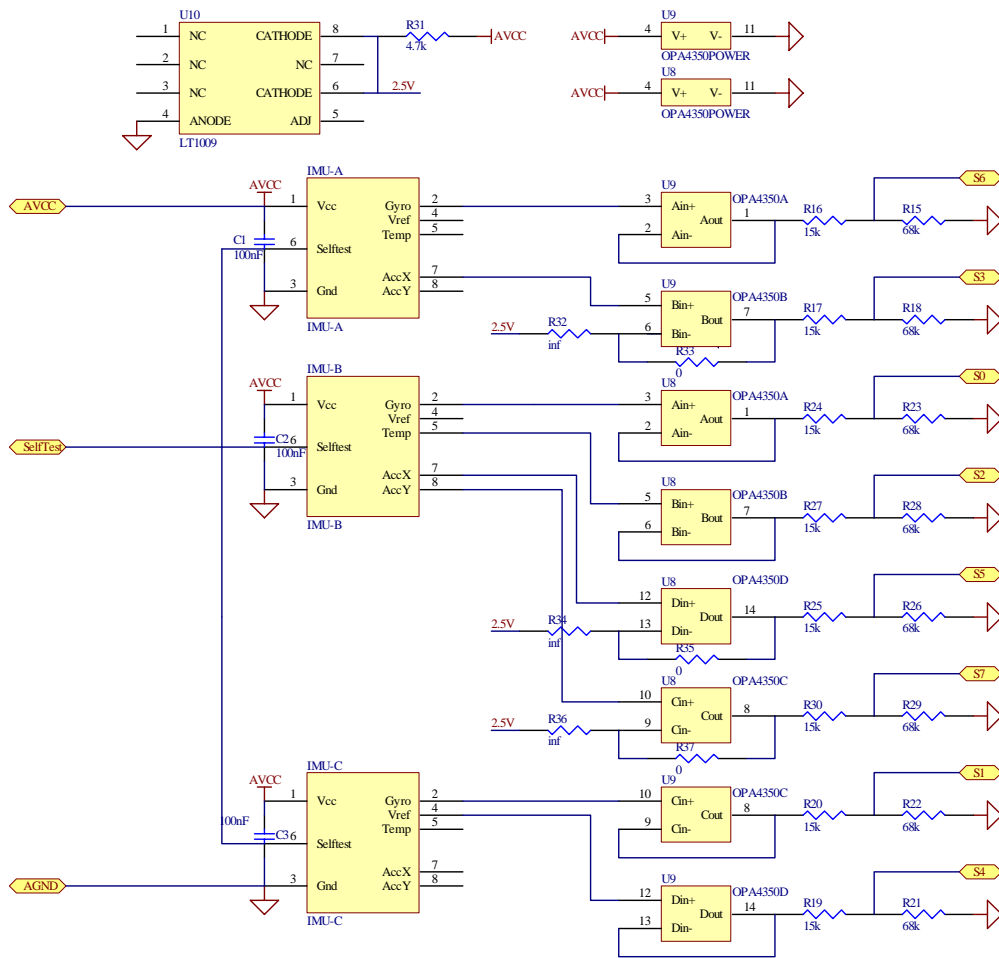


Figure F- 5: Three axes IMU module.

All the outputs of the IMU module are routed to the 16-bit A/D channels of the CANsense board. As there are anti-aliasing filters onboard the CANsense board, no anti-aliasing filtering is needed on the CANsenseIMU board.

The signals are scaled so that the whole range of values can be sampled by the A/D converter. This scaling also affects the anti-aliasing filter because of the output impedance of the voltage divider. To compensate for the output impedance of the voltage divider, the first resistor in the anti-aliasing filter must be changed from 33kΩ to 22kΩ.

Provision has been made for the amplification of the accelerometer signals. However, this option has never been used.

F.2.2 Magnetometer

Figure F- 6 shows the magnetometer interface schematics. This includes the set/reset circuitry, output amplification (see section 4.5) and anti-aliasing filters. The outputs of the magnetometer are sampled by the 10-bit A/D converters of the CANsense board.

The transfer function of the amplification and anti-aliasing filter is:

$$H(s) = \frac{4}{s + \frac{1}{RC}} \quad (\text{F.4})$$

Equation (F.4) is a first order Butterworth filter with a DC gain of 4. With R as 47kΩ and C as 33nF, the -3dB cut off frequency is at 100 Hz.

F.2.3 Ultrasonic sensor

The CANsenseIMU has a connector (J4) to interface with an ultrasonic sensor. However, this feature was not used.

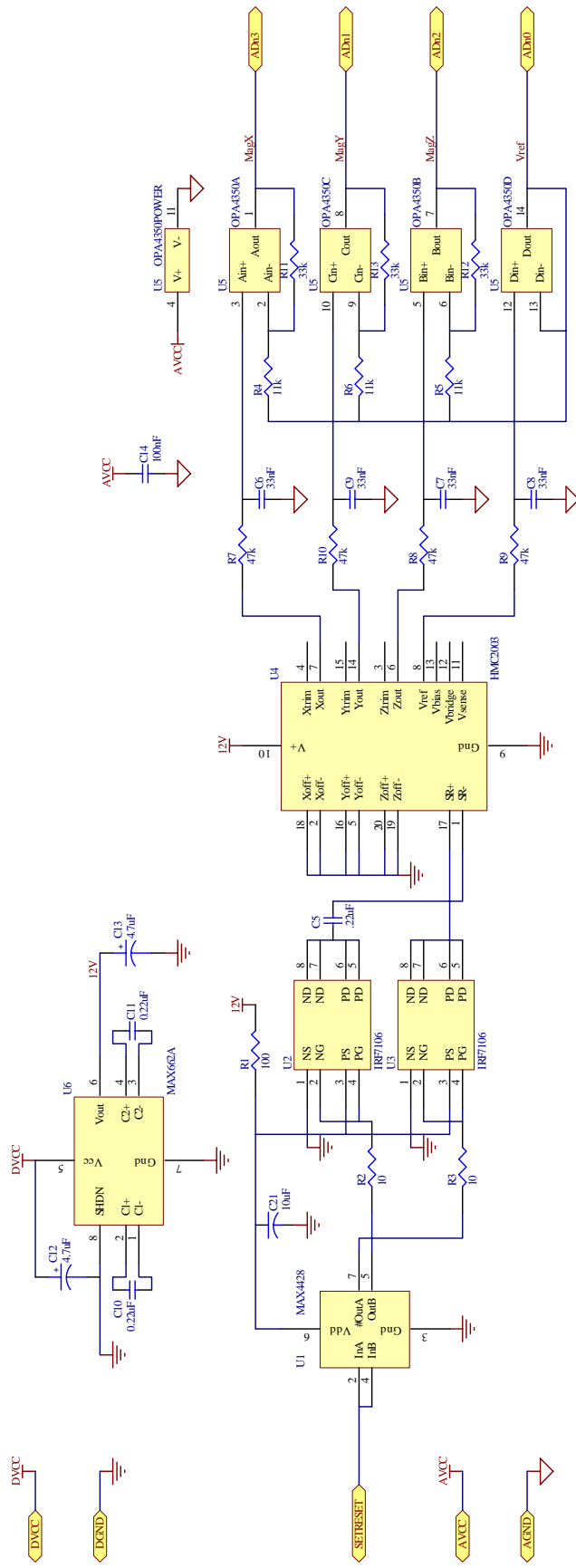


Figure F- 6: Magnetometer interface schematics.

F.3 Sensor module

The sensor module consists of a CANSense board connected to a CANSenseIMU board, hosting the gyros, accelerometers and magnetometer.

When the CANSenseIMU board is rotated so that the three single axis IMU boards are on top and at the right, the X-axis is the short side (from the horizontal IMU board towards the vertical IMU board), the Y-axis is the long side (from the magnetometer to the IMU side), and the Z-axis is downwards. Figure F- 7 provides a schematical layout.

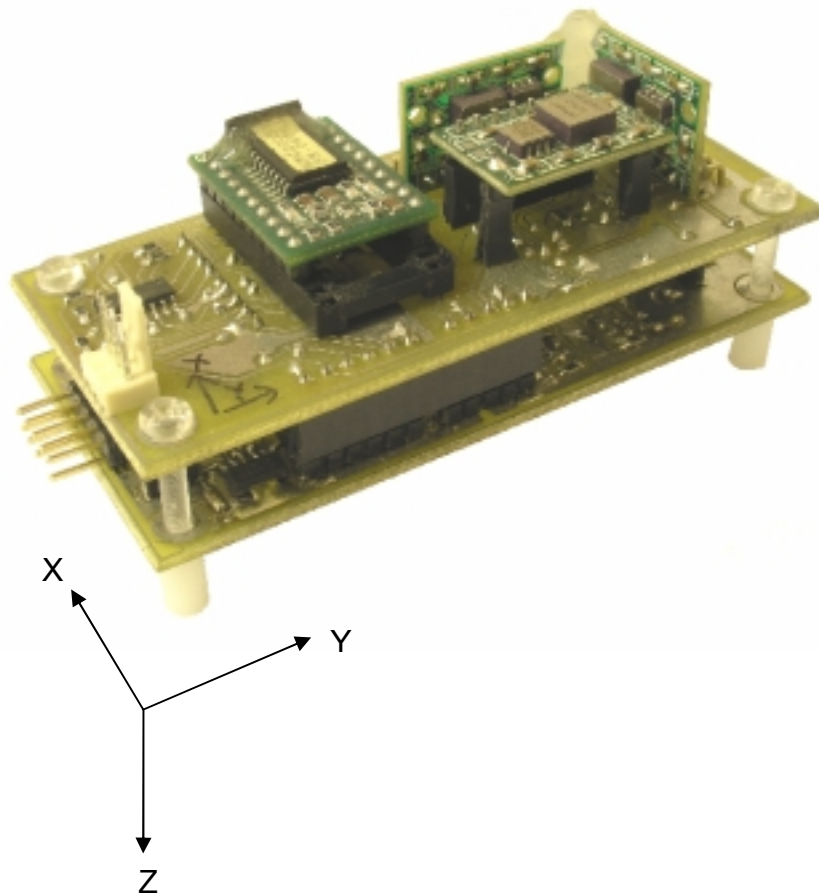


Figure F- 7: Sensor axes layout.

F.4 Power and GPS

The power and GPS module is the same form factor as the PC/104 OBC. It has two DC-DC converters for voltage regulation. One DC-DC converter's output is at 5.0V, which is used to power the PC/104 OBC and GPS module. The PC/104 OBC is powered through the PC/104 bus. The other DC-DC convert's output is at 6.5V, which is used to power the IMU module and RF Link.

The uBlox GPS interface consists only of power connections and a RS232 line driver / receiver. Note that the uBlox GPS receiver must have at least firmware 5.00 loaded to decode the RTCM data correctly.

Figure F- 8 contains the schematics of the power and GPS board.

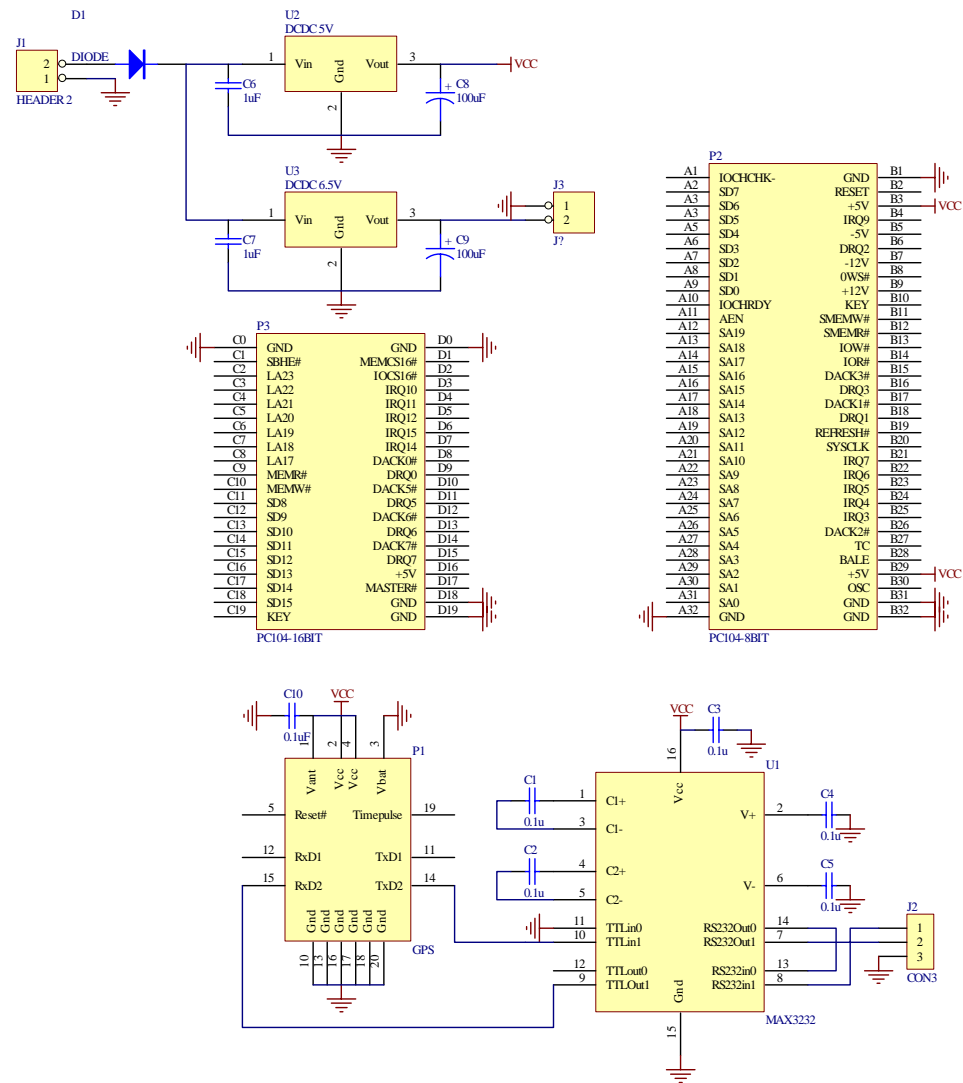


Figure F- 8: Power and GPS schematics.

Appendix G: Software

G.1 CANsense PIC

The code for the CANsense PIC microcontroller can be divided in two parts: main loop (see Figure G- 1) and the interrupt service routine (see Figure G- 2). The interrupt service routine handles all A/D conversions and filtering actions. When it is time to transmit data (every 20ms), the interrupt loop sets a flag so that the main loop can do the transmissions while the interrupt loop continues in the background. The source code can be found on the accompanying CD.

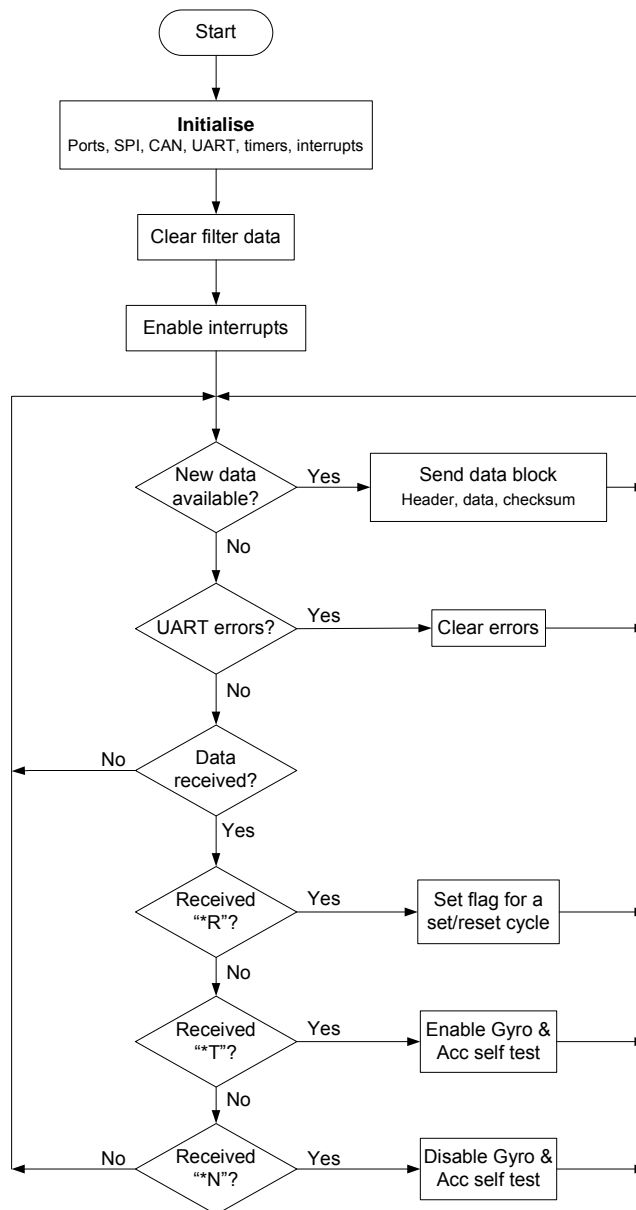


Figure G- 1: CANsenseIMU main loop.

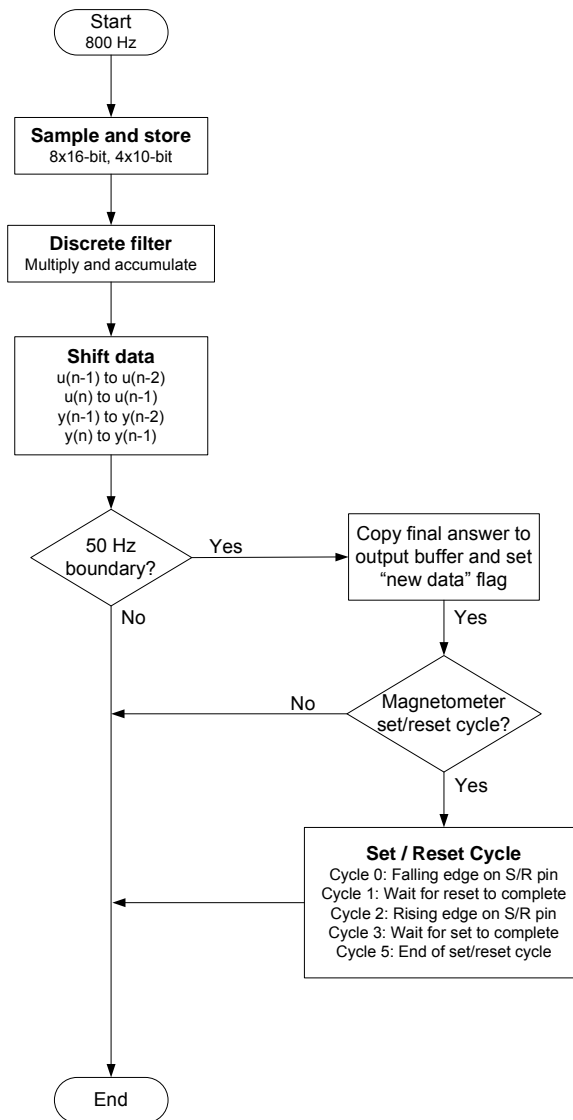


Figure G- 2: CANsenseIMU interrupt service routine.

The RS232 port is set up at 19200 bits per second, 8 data bits, 1 stop bit and no parity. No hardware or software handshaking is implemented.

The sensors module accepts the following inputs:

Input	Function
*R	Initiates a set / reset cycle for the magnetometer on the next sample
*T	Self test mode for the gyros and accelerometers.
*N	Normal mode for the gyros and accelerometers.

The sensor module starts to output sensor data at intervals of 20ms (50 Hz) in the following format:

<Header><Ch00><Ch01> . . . <Ch11><Checksum>

The header consists of the following two hexadecimal characters: 0xe1 0x4d

There are 2 bytes per channel, least significant byte first. This unsigned 16-bit word will give 0 for a sensor output of 0V and 65535 for a sensor output of 5V. A checksum is run over the data part, consisting of 2 bytes. The checksum algorithm used is the 8-Bit Fletcher Algorithm, which is being used in the TCP standard (RFC 1145) and the UBX binary protocol.

A sensor packet consists of 28 bytes. With a sampling frequency of 50 Hz and 10 bits per byte (start and stop bit included), the data rate is 14000 bps.

The channel allocation is given in Table G- 1.

Channel	Function
0	Gyro +X
1	Accelerometer -Y
2	Magnetometer Vref
3	Accelerometer +Z
4	Gyro +Y
5	Magnetometer +Y
6	Temperature
7	Gyro +Z
8	Magnetometer -Z
9	Accelerometer +X
10	Gyro Vref
11	Magnetometer -X

Table G- 1: Channel allocation for the sensor module.

G.2 OBC

The OBC runs a real time operating system called AMX on top of MS-DOS. MS-DOS is only used as a boot loader and file system handler for AMX. Custom software routines written in C are compiled with the AMX operating system, resulting in a single MS-DOS-executable file. When launching this program from MS-DOS, AMX bypasses MS-DOS and takes control over all the hardware.

The custom routines were written in Borland Turbo C++, but only C-style programming was done for compatibility with the AMX operating system. Standard C libraries can be used, which includes mathematical libraries and support for hardware floating-point operations.

The AMX PC Supervisor routines provides access to the input and output devices of a standard IBM PC. This includes the keyboard, display adapter, COM ports, parallel ports, disk adapters, clock and speaker. Only the clock interface and keyboard support was used. Custom interrupt driven routines for the COM ports was developed, as the AMX PC Supervisor routines required hardware handshaking signals to be present on all COM ports. In addition, support for the FIFO buffers was added for better performance. See Figure G- 3 for the COM ports interrupt service routine flow diagram and Figure G- 4 for the flow diagram of the COM port transmit procedure.

In order to have all four COM ports driven by interrupt service routines, every COM port must have its own IRQ number, as the ISA bus has no support for IRQ-sharing. The base addresses and IRQ numbers used for the four COM ports are listed in Table G- 2.

Port	Base address	IRQ number
COM1	0x3f8	4
COM2	0x2f8	3
COM3	0x3e8	9
COM4	0x2e8	15

Table G- 2: COM port settings of the OBC.

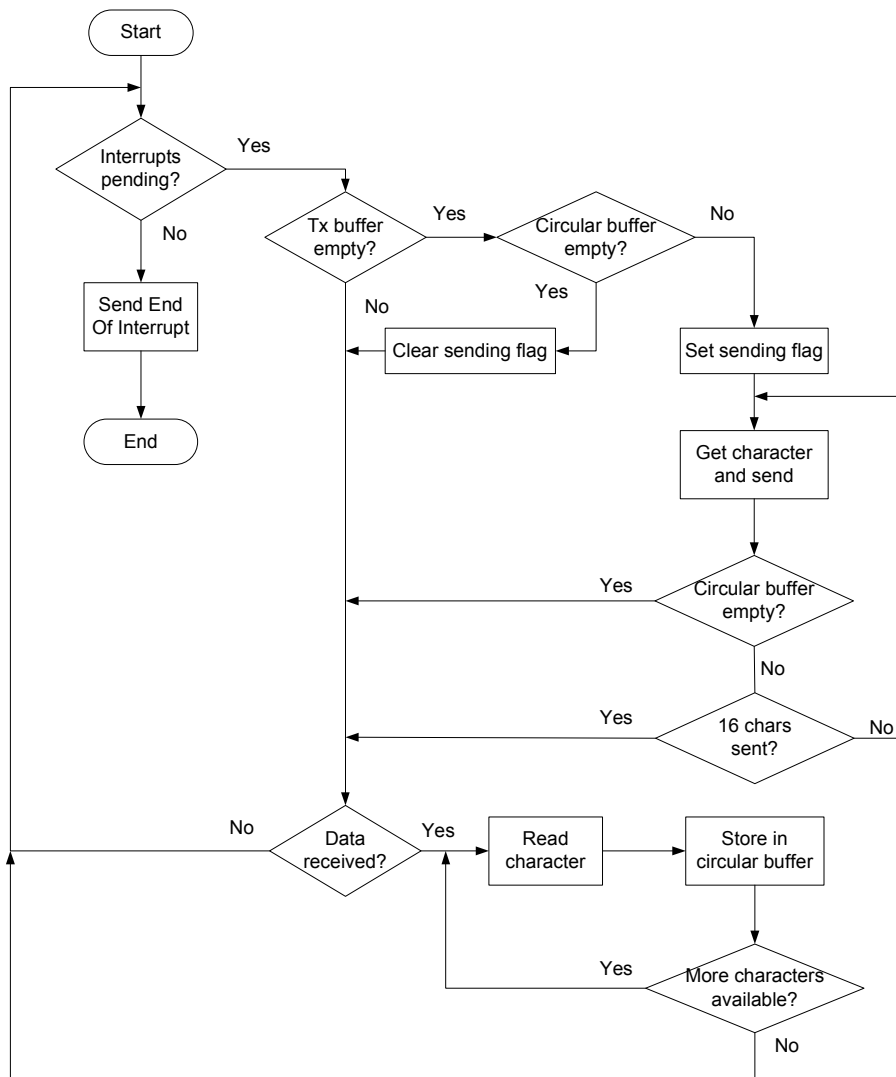


Figure G- 3: COM port interrupt service routine.

The heart of the AMX operating system is a task scheduler deciding when which task must be executed. Every task has a certain priority set to it, to indicate if it is a high priority (small number) or a low priority (big number). The different tasks with their respective priority settings are listed in Table G- 3.

The receiver task is set at lowest priority, because it is an endless loop. Whenever data is received and no other task needs to be executed, the received data is processed. As the CANSenseIMU board sends data at 50 Hz to the OBC, the CANSenseIMU board can be seen as the clock source for the propagation of the Kalman filters. When a complete IMU packet is received, the Kalman filters are propagated. Figure G- 5 represents the flow diagram of the AMX receiver task.

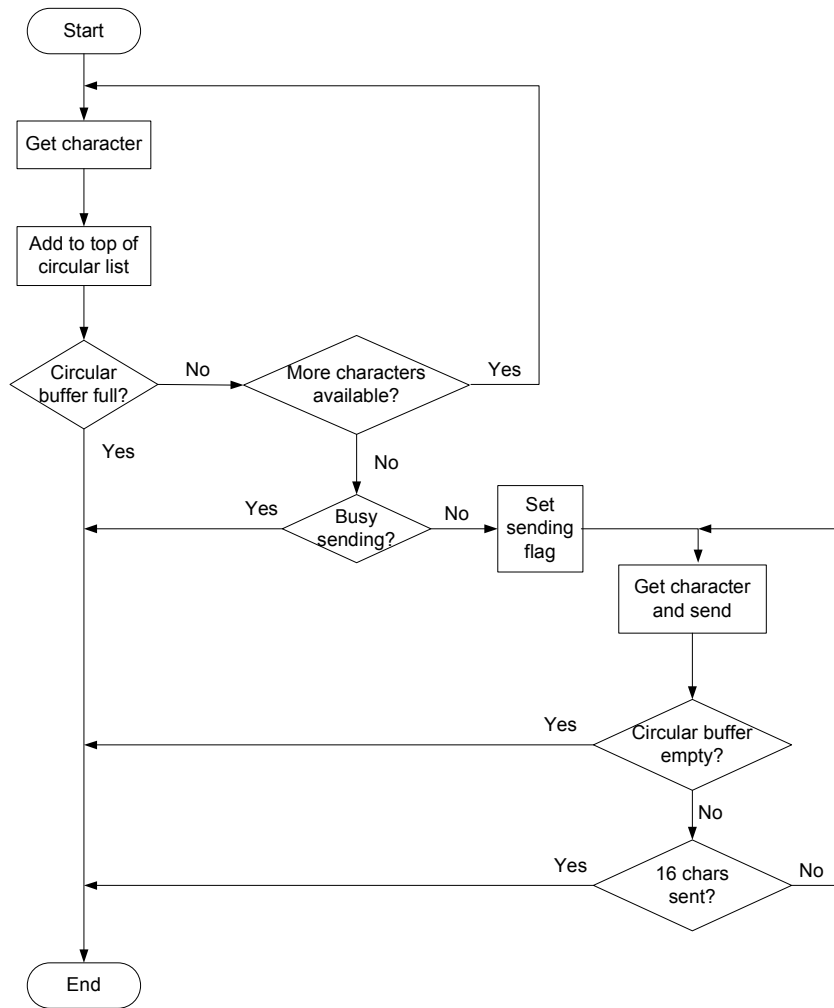


Figure G- 4: COM port transmit procedure.

Task	Priority
Clock tick	1
Keyboard tick	2
Transmit data on COM1	30
Transmit data on COM2	31
Transmit data on COM3	32
Transmit data on COM4	33
Print a message on the display	50
COM port data receiver	60

Table G- 3: Task priorities in AMX.

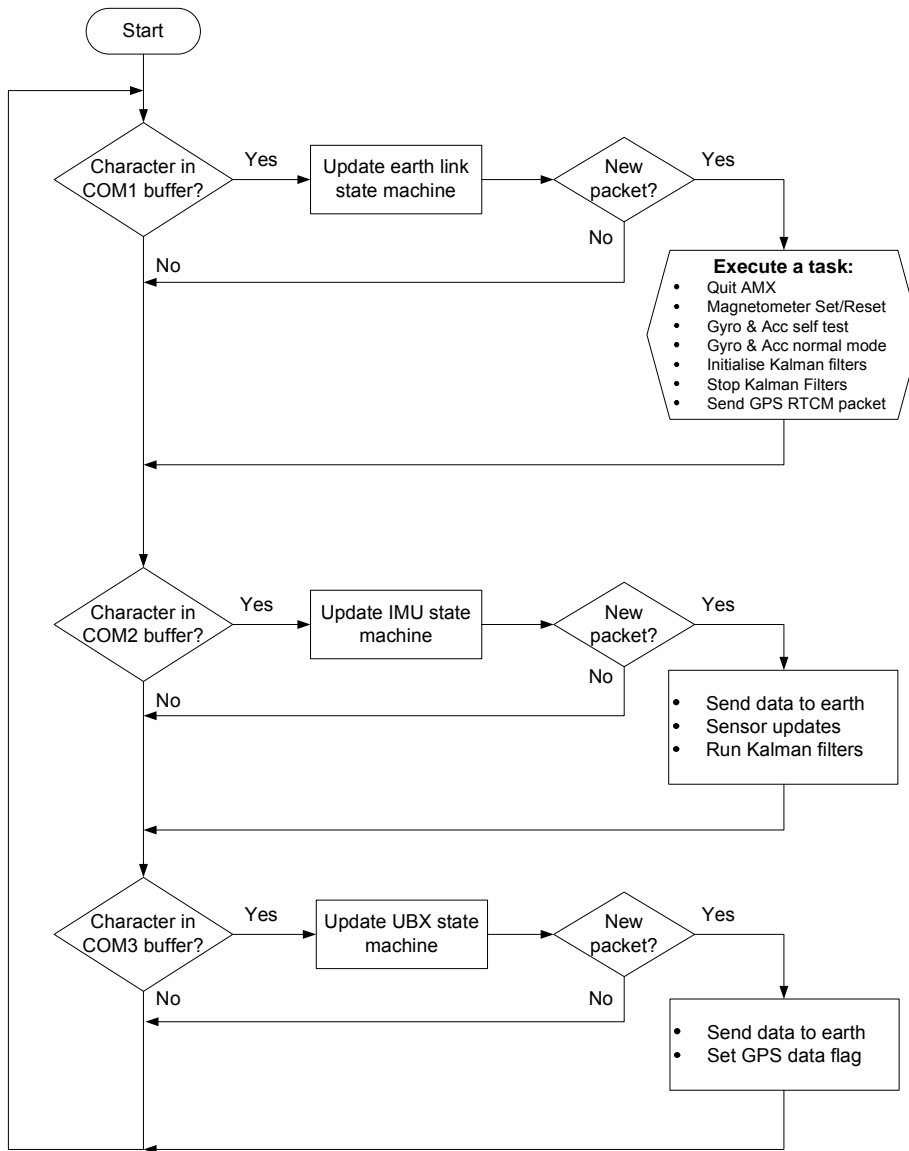


Figure G- 5: AMX receiver task.