DEVELOPMENT OF A MODEL HELICOPTER BASED FLIGHT TEST PLATFORM FOR MULTIVARIABLE FEEDBACK CONTROL

By
Thessygan Moodley

Submitted in fulfilment of the requirements for the degree of Master of Science in Engineering, completed at the School of Electrical, Electronic and Computer Engineering, University of Kwazulu Natal, under the supervision of Professor ES Boje.

July 2009

Abstract

The dissertation describes the development of a model helicopter based flight test platform for implementing autonomous six degree of freedom flight by a multiple input multiple output automatic control system.

The focus of the research is two fold:

i. Navigation system design centred about fusing multiple data and measurement sources using Kalman filtering techniques.
ii. Electrical engineering of a complete avionics package to support guidance, navigation and control functions.

Included are the results from several experiments conducted on the test platform, highlighting salient aspects and performance of the electrical and navigation systems.

Acknowledgements

Undertaking any endeavour of meaning cannot be done without the invaluable support of others. This project is no different, and, several individuals must be sincerely applauded for their efforts.

Firstly, my sincerest thanks are directed at Professor Edward Boje of the School of Electrical, Electronic and Computer Engineering (University of Kwazulu Natal), for his patient and expert supervision during the project. His knowledge and experience rendered the project period an interesting and precious learning experience for which I am extremely grateful.

Trevor Lorimer, the project test pilot, features on this list not only for his acrobatic efforts with the helicopter airborne, but, also for his dedication and mechanical wizardry with the craft on the ground. His patience and company during numerous test flights is also valued, as is his continued friendship.

The workshop staff at the School of Electrical, Electronic and Computer Engineering (University of Kwazulu Natal) deserves mention for the tireless effort involved in procuring components, printed circuit board manufacture and other assistance.

In terms of funding, I extend gratitude to the National Research Foundation (NRF) for the award of a Scarce Skills Scholarship which enabled me, financially, to study a post graduate degree in comfort. Additionally, research funds provided by the NRF helped facilitate smooth funding of the project.

Contents

List of figures

List of tables

Nomenclature, symbols and abbreviations

Nomenclature /Abbreviation	Description
(SA)RAM	(Single Access) Random Access Memory
‘BP3010	Micro inertial sensor unit BP3010
‘F2812	TMS320F2812 digital signal processor
A_{jk}	In matrix contexts, indicates the element in row j, column k of matrix A
ALU	Arithmetic logic unit
AT	ATtention (Hayes commands)
C/A	Coarse Acquisition
CMOS	Complementary metal oxide semiconductor
CMPS03	Two axes magnetic compass employed in the project
DC	Direct current
DSP	Digital Signal Processor
FM	Frequency modulation
GPS	Global Positioning System
GUI	Graphical user interface
Gyro	Gyroscope
I/O	Input/Output
I^2C	Inter Integrated Circuit
IC	Integrated Circuit
IMU	Inertial measurement unit
INS	Integrated navigation system
ISM	Instrumentation scientific measurement (band)
ISR	Interrupt Service Routine
LSB	Least significant bit
Matlab	Matlab version 7.0.1.24704 release 14
MIPs	Million Instructions Per Second
NMEA	National (American) Marine Electronics Association
NMEA 0183	Standard for interfacing marine electronic navigation devices
OEM	Original Equipment Manufacturer
PC	Personal Computer
PROM	Programmable Read Only Memory
PWM	Pulse width modulation
R/C or RC	Radio control(led)
RMS	Root mean square(d)
ROM	Read Only Memory
RS232	Asynchronous serial port signalling protocol
SCI	Serial Communications Interface
SDK	Software Development Kit
SPI	Serial Peripheral Interface
SPS	Standard Positioning Service
TSIP	Trimble Standard Interface Protocol
UART	Universal Asynchronous Receiver Transmitter
USD	American Dollars
VB	Visual Basic
VB2005	Visual Basic 2005
Windows	Microsoft Windows XP Professional Service Pack 2
Word	16 bits

Nomenclature /Abbreviation	Description
ZAR	South African Rands

Symbol/Unit	Description
/s or s^{-1}	Per second
$/s^2$ or s^{-2}	Per second squared
°	Degrees (unit of measurement for angles)
A	Ampere
Am^{-1}	Amperes per meter – unit of magnetic field strength
bar	Unit of pressure 1 bar = 10^5 Pa
g	Gravitation acceleration (about 9.81 m/s^2)
Hz	Hertz
m	Metre(s)
Pa	Pascal
rad	Radians
V	Volts
W	Watts
≈	Approximately
$\bullet^T$	Transpose of matrix or vector, ●
$\mathbf{A}, \mathbf{\Psi}$ (bold capital letters)	Indicates matrices.
$\mathbf{a}, \boldsymbol{\psi}$ (bold miniscule letters)	Indicates vectors.

1 Introduction

Helicopters number amongst the most adaptable and versatile vehicles in the modern world. Their remarkable capabilities give pilots access to all degrees of freedom of three dimensional space and enable the helicopter to travel to areas inaccessible to other flying machines. Characteristics such as vertical take off and landing, hover and cruise flight have made the helicopter the vehicle of choice for a wide range of applications in very varied environments. It is, however, these desirable characteristics, in addition to the number of degrees of freedom, interaction between channels, non-linearity and model uncertainty that also make the helicopter a complex machine that is difficult to control. Uncertainty in helicopter dynamics further compound the problem, making the development of helicopter flight controllers a challenging task.

For this project, initial steps toward six degree of freedom flight (autonomous flight) of a model helicopter were desired. The project furthered understanding of multivariable estimation through the integrated navigation system design. The practical component required the development of suitable hardware and software to interface the helicopter and the development of helicopter navigation algorithms and management systems. A ground station, in the form of a personal computer with custom application software, was also envisaged.

The scope of the project spanned research and significant hardware and software design culminating in the following project components which were developed from January 2005 to April 2007:

1. Understanding of current unmanned aerial vehicle hardware and software components with a view to helicopter avionics design. Avionics, in the scope of the project, implies all electrical hardware and software components necessary to accomplish the project objectives using the model helicopter as the flight test platform.
2. On board helicopter hardware design including processing core, navigation sensor suite, helicopter interface, wireless communications and power supply design with a view to creating a low cost integrated navigation system and eventual flight test platform.
3. Examination of estimation methods for the implementation of a suitable navigation system, and subsequent detailed exploration into the application of Kalman filtering techniques for integrated navigation system design.
4. Software design including helicopter avionics software, comprising command, telemetry and navigation functions, and ground side personal computer software for remote operation and monitoring of avionics.

The project encapsulates information drawn from several sources. Padfield (1995) takes significant steps toward solving the six degree of freedom orientation problem. Weston and Titterton (2002) provide a summary of modern inertial navigation technologies, techniques and their application, whilst Britting (1971), Farell and Barth (1999), Salychev (1998) and Grewal, Andrews and Weill (2001) provide deeper insight into the navigation problem and solution including mathematical descriptions, competing hardware and software schemes, use of the Kalman filter and valuable examples.

Maybeck (1999) provides a pleasant introduction to stochastic processes and estimation and, together with Levy (2002), assists with understanding the operation of the Kalman filter. These essentially lay the groundwork for more detailed Kalman filter and estimation texts including Brown and Hwang (1992), Grewal and Andrews (2001) and Kailath (2000). Brown and Hwang (1992) and Grewal and Andrews (2001) also provide brief discourses on INS integration schemes with Grewal and Andrews (2001) also delving into aspects related to digital computer implementation. Algorithms from Gerald and Wheatley (1999) and Welford (1962) are used in the digital navigation system implementation and ideas from Wilkinson (1965) and Press (1992) are used in analysing the numerical performance of the system. Hong et al. (2005), Rhee et al. (2004) and Ham and Brown (1983) quote information related to the observability of the global positioning system (GPS) aided integrated navigation system (INS).

Apart from the linear algebra, calculus and physical relations and theories found in Beauregard and Fraleigh (1995), Finney et al. (2001) and Young et al. (1996), several other texts, research papers and Internet sources were examined but not directly referenced. These are listed in the bibliography. Additionally, electronic hardware device specifications are extracted from manufacturers' data sheets, and, assistance for both the digital signal processor and personal computer software development kits was found in the respective online help systems.

In terms of document structure, Section 2, "Model helicopter description", provides a basic description of the model helicopter forming the test platform. Section 3, "Electronic and instrumentation systems", is a detailed examination of the on board avionics and ground side hardware and software designed for this project. Section 4, "Navigation", presents the GPS aided INS design and implementation. Several appendices also exist, which are referenced as required.

2 Model helicopter description

2.1 Overview

This section provides a basic description of the model helicopter used in this project including some of the key helicopter specifications.

2.2 General model helicopter information

The model helicopter employed in the project was a Hirobo BBC Stork intended for use by radio control (RC) enthusiasts. Figure 2.1 is a photograph of the uninstrumented flight platform highlighting key components.

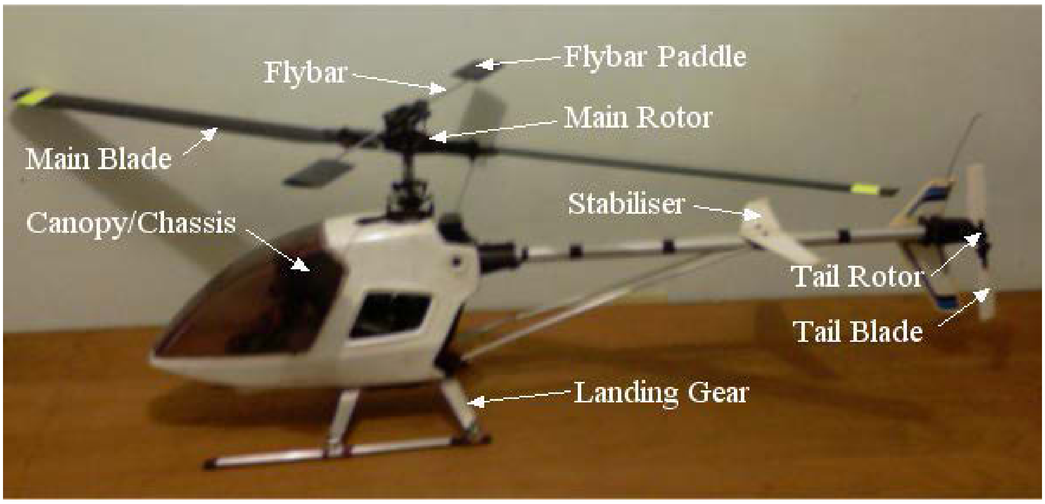


Figure 2.1. Uninstrumented model helicopter (Lorimer, 2006).

Table 2.1 provides a categorised list of key model helicopter parametrics. Reference is made to Figure 2.2, below, which defines a body fixed corordinate frame for the model helicopter.

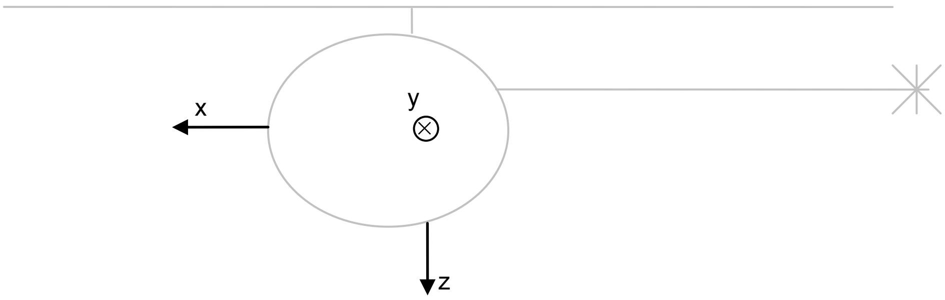


Figure 2.2. Coordinate frame used in descriptions of model helicopter specifications.

Table 2.1. Key helicopter specifications (Lorimer, 2006).

Parameter		Unit/Comment
Engine		
Cylinders	1	
Displacement	7.54×10^{-6} (0.46)	m^3 (in^3).

Parameter		**Unit/Comment**
Stroke	2	
Cooling	Air	
Power output	1.4	hp at 16 000 rpm.
Ignition	Glow plug.	
Fuel		
Capacity	240	ml.
Composition	Mixture	15 % Nitro, 17 % caster oil and 68 % methanol.
Main rotor		
Number of blades	2	
Blade length	0.675	m.
Blade chord	0.055	m.
Blade lift curve slope	6	
Stiffness	50	Nm/rad.
Moment of inertia	0.01868	kg/m^2.
Speed	1 500	rpm.
Tip speed	105 (380)	ms^{-1} (kph).
Tail rotor		
Number of blades	2	
Blade length	0.1	m.
Speed	6000	rpm.
Fly bar		
Type	Bell Hiller	
Length	0.065	m.
Chord	0.055	m.
Lift curve slope	6	
Moment of inertia	0.00437	kg/m^2.
Dimensions		
Helicopter mass	3.726	kg.
Instrumented mass	4.828	kg.
x Centre of mass	0.249	m.
y Centre of mass	0.110	m.
x Moment of inertia	0.082	kg/m^2.
y Moment of inertia	0.327	kg/m^2.
z Moment of inertia	0.244	kg/m^2.
Helicopter length	1.1	m (Nose to tail).
Helicopter width	0.245	m (Side to side at widest point).
Helicopter height	0.43	m (Skids to rotor top).
Flight time		
Hover	≈ 10	min (Considering fuel consumption).
Forward flight	≈ 20	min (Considering fuel consumption).

3 Electronic and instrumentation systems

3.1 Overview

To achieve the automatic control objectives of the project, an appropriate electronics package and instrument suite is required. This section furnishes details of all hardware, software and instruments that form the model helicopter avionics, as well as any support hardware and software e.g. Ground station devices.

Section 3.2, "Key functional requirements", highlights the main tasks that a generalised avionics package needs to achieve. Section 3.3, "Hardware", supplies particulars of electronic and instrumentation devices needed to fulfil the key functional requirements. The chapter ends with Section 3.4, "Software", which details the embedded software design for the avionics and which also includes information regarding ground side applications.

3.2 Key functional requirements

Before considering the design of hardware and software systems used, it is appropriate to mention the overall functional tasks that such systems would be required to perform.

Beginning with the flight platform, the model helicopter naturally demands that its servo motors be actuated. Such actuation calls for an appropriate interface to the servos. Furthermore, as will be discussed later, the model helicopter is flown with a standard radio controlled hobby receiver. This device also drives the servo motors and thus some arbitration is needed between signals from this unit and those from the avionics.

In terms of the eventual project objectives, particularly autonomous flight of the helicopter, several critical requirements arise. Chief among these is the need for some type of on board flight computer or processor. Being the core of the avionics (the flight platform cannot support multiple processors due to payload limitations), this device must possess capabilities that allow for the management of all other hardware. Additionally, it must have sufficient resources to support system software and adequate numerical processing power to handle the computational rigours of the navigation and flight control algorithms.

The navigation algorithms depend on another critical functional component: the inertial platform. An inertial platform is characterised by several hardware (IMU, aiding sources) and software (estimators) parts which must work in synchronism to produce dependable navigation solutions. These solutions, appropriate inertial data and reference trajectories are supplied to a flight control loop which is obviously necessary if the helicopter is to be flown sans pilot.

As the helicopter is extremely mobile (it has six degree of freedom movement in three dimensional space), data, command and status exchange with the avionics package can only be achieved through the use of a reliable wireless communications link. This channel should enable a ground station and operator to remotely monitor and log pertinent operating parameters. Monitoring and logging facilitates the in flight identification of faults (knowledge of which can prevent catastrophic failure), and, enables post flight analysis to be performed so that algorithms and systems can be adjusted.

Finally, the electronic and instrumentation package should include a suitable power source. As is to be expected with the model helicopter's payload constraints, the power source, and all avionics components, should be as small and light as possible.

Figure 3.1 summarises the key functional components of the model helicopter to achieve autonomous flight of the model helicopter. Inter connections between elements indicate functional components that depend on or communicate with each other in some way.

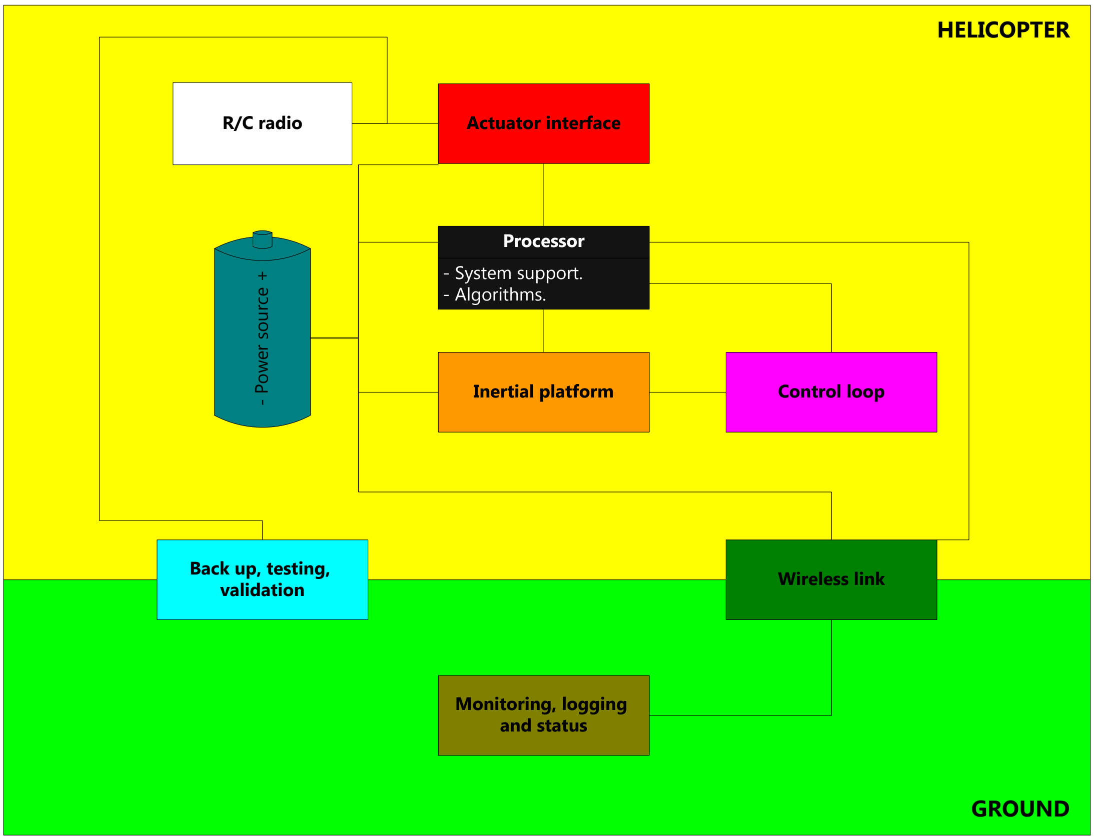


Figure 3.1. Summary of functional requirements.

3.3 Hardware

3.3.1 Hardware overview

Meeting the design challenges outlined in Section 3.2 required a variety of hardware components. Most components were acquired as original equipment manufacturer (OEM) modules, with the remaining functionality provided by custom designed hardware e.g. A motherboard containing all supporting circuitry (interfacing, voltage level translation etc.).

Processing is handled by a digital signal processor (DSP) which communicates, via the motherboard, with multiple devices external to itself. Among these the inertial measurement unit (IMU), GPS receiver, barometric altimeter and magnetic compass form the key hardware comprising the inertial platform. Signal switches multiplex either the signal processor or pilot commands (which arrive through a hobby radio control transmitter receiver pair) to the actuators. The primary wireless channel, however, is a Bluetooth serial data link, which, on the ground, is supported by a computer. Power to all on board systems is supplied by a rechargeable battery with appropriate regulation to meet voltage and current demands of all hardware.

Figure 3.2 shows the major electronic devices/systems used. Shading in the figure corresponds with that employed in Figure 3.1 and indicates those functional aspects impacted on by a particular piece of hardware. Interconnection between elements in the diagram implies electrical interaction (the exact nature of which will be considered later).

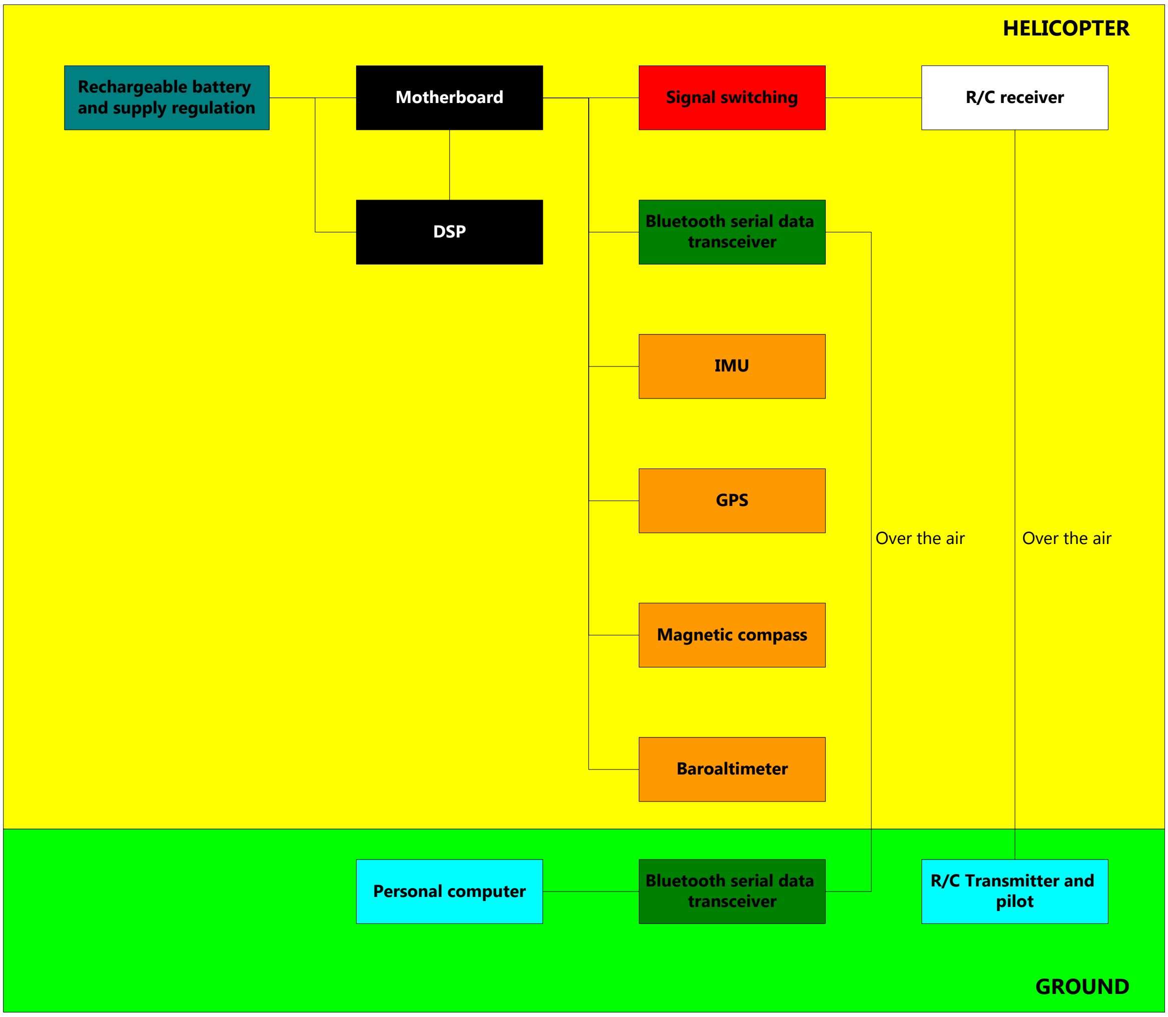


Figure 3.2. Summary of hardware setup.

3.3.2 Processing unit[1]

The DSP identified as the processing element in Section 3.3.1, is a Texas Instruments TMS320F2812 ('F2812). The processor is not used alone but resides on a DSP development board manufactured by Spectrum Digital – the eZdsp F2812 (Figure 3.3).

[1] Component parametrics and technical information presented in subsequent sub sections of Section 3.3 have been extracted from device data sheets and technical documents published by manufacturers. These documents are included in Appendix 3.

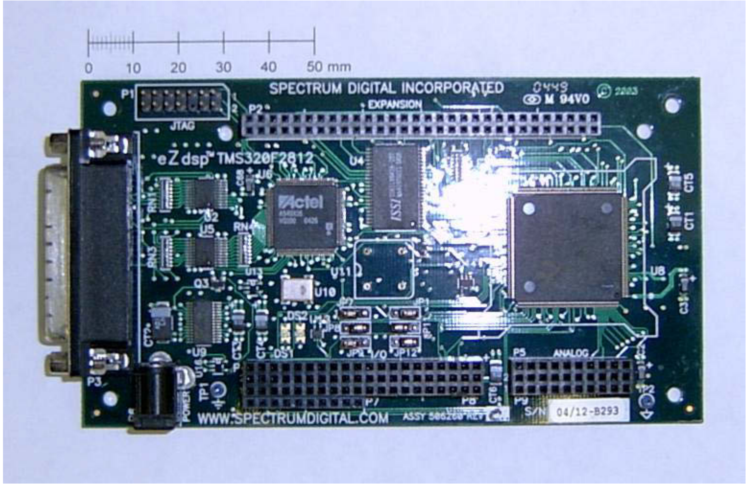


Figure 3.3. TMS320F2812 DSP development board (eZdsp F2812)

Although use of the eZdsp board may contradict the size and weight minimisation objective, the rapid and simple hardware development afforded by the eZdsp F2812 was traded against a small increase in size and weight. Apart from containing all DSP support circuitry (power supplies, decoupling capacitors, clocks/crystals) and convenient headers for I/O connection, the eZdsp board also included the debugging and programming interfaces to connect the processor to the personal computer (PC) based software development kit (SDK). Often, throughout this document, the term "DSP" will be used loosely to represent both the signal processor and the eZdsp development kit where distinction between the two is not essential.

The 'F2812 processor itself is a 32 bit processor designed specifically for digital control applications. The signal processor core is capable of single cycle 32 bit by 32 bit multiply and accumulate operations. With suitable software libraries, single and double precision floating point numbers can also be processed. A separate interrupt management unit handles automatic context saves and restores on interrupt requests and interrupt service routine (ISR) returns. This assures reduced latency when servicing asynchronous events (such fast interrupt response is ideal for digital control). Other features of the 'F2812, contributing to increased computational bandwidth, include specialised instructions, memory pipelines and the single cycle read modify write ability of the arithmetic logic unit (ALU). This is supported partly by the Harvard architecture involving multiple busses (program read bus, data read bus and data write bus) which allows the 'F2812 core to read an instruction, read a data operand, process and write a result in a single cycle.

Apart from the Harvard bus architecture, device memory is organised as follows:

1. 128 K words of flash memory that can be mapped to both code and memory space.
2. 128 K words of ROM.
3. 18 K words of single access RAM (SARAM) divided among five logical blocks. All blocks can be configured for code or data, with the exact designation of memory to code or data being determined by the linker and link map.
4. Three of the five SARAM blocks can be independently accessed thereby reducing pipeline stalls.

Figure 3.4 illustrates peripheral support provided on the 'F2812 as well as other elements discussed thus far.

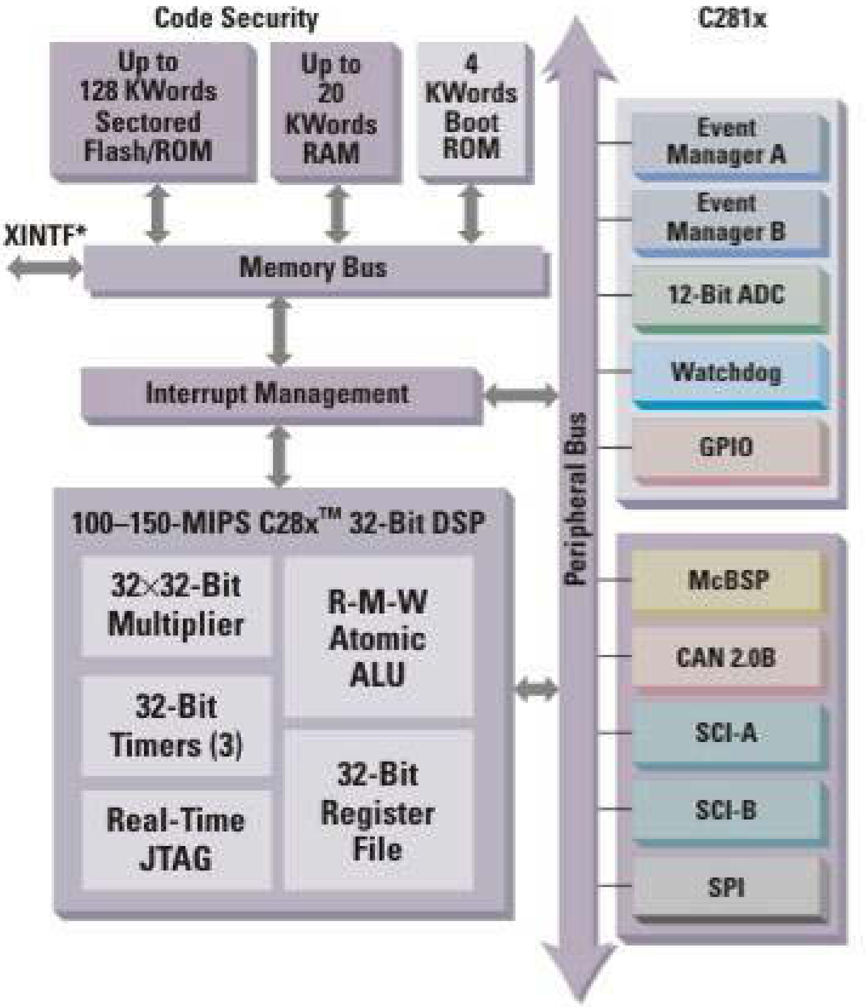


Figure 3.4. 'F2812 functional diagram (copied from Texas Instruments, 2006).

The signal processor's peripheral set is summarised in Table 3.1.

Table 3.1. TMS320F2812 peripherals and features (Texas Instruments, 2003).

Peripheral	**Quantity**	**Comments/Features**
Event Manager (EM)	2	With minimum CPU overhead can time events, control sampling and generate PWM.
Timers	2 per EM	
Compare units	3 per EM	12 PWM outputs available over both event managers.
Capture units	3 per EM	6 capture channels available over both event managers.
Analogue to digital converter	1	12 bit 16 channel analogue to digital converter featuring external and event triggered conversions resulting in minimum CPU overhead.
Serial peripheral interface	1	Synchronous serial port with enhancements: programmable clock rates and data lengths; and a 16 level deep FIFO buffer to reduce CPU overhead.
Serial communications interface	2	Asynchronous serial port with enhancements: Programmable clock, baud rate, data length and stop bits; and a 16 level deep FIFO buffer to reduce CPU load.

Peripheral	Quantity	Comments/Features
Multi-channel buffered serial port	1	High speed synchronous serial port that can support up to 256 channels creating provision for serial devices like CODECs and allowing high speed inter DSP communication for multi DSP systems.
Watchdog	1	
Controller area network	1	Support for the automotive multicast shared, differential serial bus.
General purpose inputs/outputs (GPIO)	Up to 56	I/O pins are multiplexed with peripheral signals. Directions are software selectable, and, GPIO system includes an input qualifier that can eliminate spurious signals on both digital and peripheral inputs.

The use of these peripherals is discussed throughout Section 3.3, when considering the interfacing needs of other hardware components.

Together with the substantial peripheral support and the fixed point ALU capable of 150 MIPS, the TMS320F2812 offers significant micro controller like I/O support but with much greater processing bandwidth.

3.3.3 Instrumentation

3.3.3.1 Inertial measurement unit

The IMU used at the heart of the navigation system is a Micro ISU BP3010 manufactured by Bulmer Electronics and Control. The device is a six degree of freedom inertial sensor measuring three orthogonal axes of acceleration (± 10 g per channel) and rotational velocity (± 300 °/s per channel).

Internally, the 'BP3010 employs Micro-Electro-Mechanical-Systems (MEMS) accelerometers and rate gyros manufactured by Analog Devices (the ADXL210E and ADXRS300 respectively). MEMS technology features the combination of mechanical elements (e.g. sensors and actuators) and electronic components (e.g. signal conditioning circuitry) on a common silicon substrate. The ADXL210E, for example, incorporates a surface micro-machined poly silicon structure suspended by poly silicon springs (forming the moving mass and spring central to acceleration measurement - Weston and Titterton, 2000). Deflection of this structure is indicative of acceleration and measurement is made via capacitive techniques with requisite signal conditioning on chip.

Additionally, the 'BP3010 incorporates:

1. A power supply that stabilises and regulates from a 5 V, 100 mA source.
2. Several micro controllers that perform temperature, scale factor, bias and misalignment compensation; and that manage the external data interface.
3. Sensor conditioning components providing sensor support circuitry and sensor to micro controller interface.

All six measurements are output at a maximum rate of 64 Hz in a 16 byte data frame via a UART. Acceleration and angular velocity is transmitted as signed scaled velocity increment and angular increment respectively.

In terms of error performance the IMU is specified as per Table 3.2.

Table 3.2. Specified error characteristics for the Micro ISU BP3010

Characteristic		Unit	Notes
Velocity increment output noise (rms)	0.005	ms^{-1}	1.
Angular increment output noise (rms)	5.236×10^{-4}	rad	2.
Axes misalignment	< 0.017	rad	< 0.1 °.
Residual gyro bias	0.0087	rad s^{-1}	3.
Residual accelerometer bias	0.049	ms^{-2}	5 mg.
Scale factor accuracy: Velocity increment	0.2	%	4.
Scale factor accuracy: Angular increment	0.2	%	5.

[1] At 64 Hz this translates to 0.32 ms^{-2} acceleration output noise (rms).

[2] At 64 Hz this translates to 0.0355 rad s^{-1} (approximately 2 °/s) angular velocity output noise (rms).

[3] Approximately 31.32 rad/hour (approximately 1795 °/hr)

[4] $\frac{1}{65535} \pm \frac{1}{327680}$ rad/LSB.

[5] 0.001 ± 0.0002 ms^{-1}/LSB.

Examining the data of Table 3.2, it can be shown that the 'BP3010 is not a high accuracy IMU. For instance, when considering bias, one of the most dominant error sources, the 'BP3010 is several orders of magnitude worse than typical navigation and tactical grade IMUs – consider the HG1700 tactical grade IMU, manufactured by Honeywell, which features worst case gyro bias of 30 °/hr (1-σ level is quoted as 10 °/hr)(Honeywell International Inc., 2006).

Although such inaccuracy compounded the challenge of creating a workable inertial platform, this IMU is attractive with respect to size and weight (see Figure 3.5), power consumption (0.5 W) and availability. Furthermore, at a fraction of the cost of higher quality devices (≈ ZAR 15 000/USD 2 500), the 'BP3010 presented an opportunity to develop a low cost inertial navigation system.

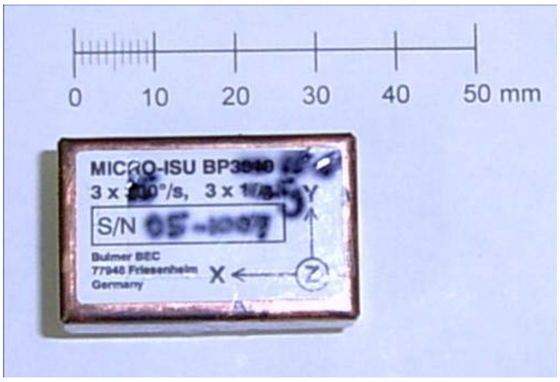


Figure 3.5. The Micro ISU BP3010 inertial measurement unit.

Prior to incorporating the 'BP3010 into the project, the intention for inertial measurement was to design a custom device using MEMS accelerometers and rate gyros. A significant time investment was made exploring this path, the results of which are provided in Appendix 3.

3.3.3.2 Global positioning system

The GPS receiver is the primary aiding source for the navigation system. The GPS receiver employed is a LassenIQ module manufactured by Trimble.

The LassenIQ is a 12 channel continuous tracking GPS receiver operating on the GPS Standard Positioning Service. It computes position, velocity and time fixes via a custom navigation processor working with the Coarse Acquisition (C/A) code broadcast on the GPS L1 frequency (1575.42 MHz).

The device algorithms also accommodate for tropospheric and ionospheric effects (through the use of approximate models); satellite clock error (by using satellite transmitted clock corrections); and satellite constellation (through optimum choice of satellite combination above the horizon). An enhanced sensitivity mode, that automatically increases receiver sensitivity should the satellite transmission be weak, further improves fix fidelity (Trimble Navigation Limited, 2004).

Table 3.3 and Table 3.4 indicate the accuracy and acquisition statistics for the LassenIQ module.

Table 3.3. LassenIQ GPS receiver accuracy (Trimble Navigation Limited, 2004).

Characteristic		**Unit**
Horizontal position	< 5 (50 %), < 8 (90 %)	m.
Altitude	< 10 (50 %), < 16 (90 %)	m.
Velocity (horizontal and vertical)	0.06	ms^{-1}.
Pulse per second[1]	± 50	ns.

[1] A 3.3 V CMOS compatible TTL level pulse once per second on the second accessible through one of the modules output pins.

Table 3.4. Acquisition times for the LassenIQ GPS receiver (Trimble Navigation Limited, 2004).

Acquisition from		**Unit**
Momentary signal loss (reacquisition)	< 2 (90 %)	s.
Hot start[2]	< 10 (50 %), < 13 (90 %)	s.
Warm start[3]	< 38 (50 %), < 42 (90 %)	s.
Cold start[4]	< 50 (50 %), < 84 (90 %)	s.

[2] Receiver has stored current almanac, initial position, time and valid ephemeris data (stored either using battery back up – secondary power source to keep RAM alive when module is powered down – or through data upload via the receiver interface port.

[3] Similar to hot start but implies an off time longer than that of the hot start. Consequently the ephemeris is invalid.

[4] Power on state with no initialisation of navigation data.

The GPS receiver, when computing fixes, outputs results at 1 Hz via a UART (9 600 bps, odd parity, no handshaking, 1 stop bit). UART transactions are formatted either using the NMEA 0183 protocol or the Trimble Standard Interface Protocol (TSIP) developed by the manufacturer. The module is compatible with several 3/3.3 V active antennas and a 3 V micro patch active antenna, fabricated by Trimble, is employed.

Dimensionally, the LassenIQ system has a small form factor:

1. Receiver: 26 mm x 26 mm x 6 mm shielded case weighing 6.5 grams.
2. Antenna: 42 mm x 50.5 mm x 13.8 mm weighing 20 grams.

The system power consumption is approximately 140 mW:

1. Receiver: 90 mW (3.3 V, 27 mA).
2. Antenna: 50 mW (3.3 V, 15 mA).

Size, mass and power consumption when coupled with low cost and availability rendered the LassenIQ and associated antenna an appropriate choice. Figure 3.6 shows the receiver and antenna.

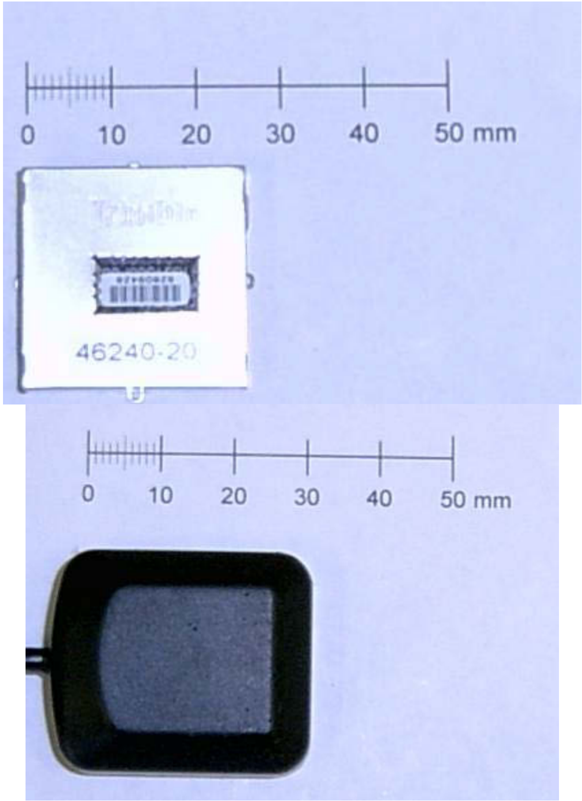


Figure 3.6. LassenIQ GPS receiver and antenna.

3.3.3.3 Barometer

With a view to supplementing altitude measurements from the GPS receiver, a barometer, the MS5534B, manufactured by Intersema, was used. The device is capable of sensing between 10 and 1100 mbar (1 – 110 kPa) with a minimum resolution of 0.1 mbar (10 Pa) via a piezo resistive sensor element and sensor interface integrated circuit (IC).

The IC serves a dual purpose: Digitisation of the uncompensated piezo resistive sensor output voltage and digitisation of the on chip temperature sensor voltage. The temperature is required for compensation routines as the pressure output is strongly temperature dependent. Additionally, compensation for temperature as well as process variation in device manufacture is provided for by a 64 bit programmable read only memory (PROM). The PROM holds six factory set calibration coefficients that must be used as part of the host application compensation routines if the specified performance (Table 3.5) is to be achieved.

Table 3.5. Accuracy/error specification for the MS5534B barometer (Intersema Sensoric SA, 2005).

Characteristic		**Unit**	**Notes**
Absolute pressure accuracy	± 1.5	mbar	± 150 Pa.
Relative pressure accuracy	± 0.5	mbar	1.
Accuracy over temperature (0 – 70 °C)	± 2	mbar	± 200 Pa.
Accuracy over supply voltage (2.2 – 3.6 V)	± 1.6	mbar	± 160 Pa.
Error with 3.3 V supply	0.4	mbar	2.

[1] ± 50 Pa when generating differenced pressure reading from some datum reading.

[2] 40 Pa due to device being operated with a 3.3 V supply rather than a 3.0 V supply. 3.3 V is readily available on hardware.

In term of this project, the barometer was used to determined altitutde. As such, altitutde errors resulting from the specifications of Table 3.5 are of more interest. These are determined in Section 4.5.3.2.2.

Data access to and from the MS5534B is via a three wire synchronous serial interface. In terms of power consumption the barometer requires approximately 3.3 mW (3.3 V, 1 mA) when producing a temperature or pressure conversion (when idle the device draws 1 µA). Together with the low power drain, this barometer is attractive due to its low cost, availability and compactness. It is packaged as a 9 mm x 9 mm x 3 mm surface mount device (Figure 3.7).

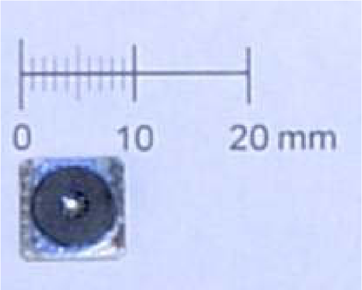


Figure 3.7. MS5534B barometer module.

3.3.3.4 Magnetic compass

A magnetic compass was employed for the purpose of yaw angle initialisation. The module used is a CMPS03 two axes magnetic sensor manufactured by Devantech (a company specialising in OEM electronics for hobby robot enthusiasts).

The compass, shown in Figure 3.8, which is intended to be a navigation aid for robots, is built around two orthogonally mounted magnetic sensors. The outputs from these sensors allows an on board processor to calculate the direction of the Earth's magnetic field relative to the sensor body. Technically, the device computes the direction of any present magnetic field, necessitating careful positioning of the CMPS03 module away from other magnetic sources in the end application. A further complication with the use of this module is that it has only two sensitive axes. Thus, to achieve sensible measurements, the plane spanned by the sensitive axes must be parallel to the Earth's surface. Such positioning will negate the effect of the dip of the planet's magnetic field toward the ground. As the magnetic compass is only used for initialisation, a process always performed with the navigation system reasonably level, this constraint proved not too severe.

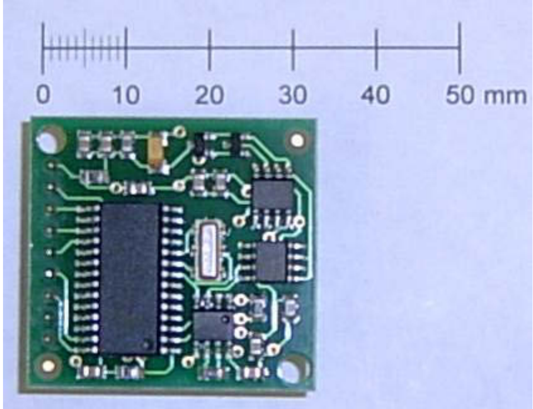


Figure 3.8. CMPS03 magnetic compass.

Internally, the CMPS03 uses two Philips KMZ51's as the sensing elements. The KMZ51 is a magnetic field sensor which exploits the magneto resistive properties of thin film permalloy to indicate magnetic field strength (Philips Semiconductors; 1998). Permalloy is a nickel iron alloy (usually 20 % iron, 80 % nickel) which, in terms of magnetic field sensing, has a very desirable characteristic: anisotropic magneto resistance (the electrical resistance changes by as much as 5 % contingent on applied magnetic field) (Webster, 2004). Each KMZ51 contains one permalloy based Wheatstone bridge resulting in magnetic field strength measurement sensitivity of 80 mV/(kAm^{-1}) (with a 5 V supply) and an operating bandwidth of 1 MHz (Philips Semiconductors; 1998).

The CMPS03 produces heading results between 0 and 359.9° with a resolution of 0.1° and an accuracy of 3 - 4° after calibration. Results can be collected via a PWM like signal, where positive pulse width represents bearing, or through the Inter Integrated Circuit (I^{2}C) data interface. I^{2}C is a two wire (clock and data) master/slave bus developed by Philips. Although it is possible for multiple masters to share the bus, common I^{2}C topologies include one master and multiple slaves (each identified by a unique seven bit address). I^{2}C supports three speeds: Standard (100 kHz), fast mode (400 kHz) and high speed (3.4 MHz). The CMPS03 results were obtained via the I^{2}C interface operating at the standard speed.

The device operates on 5 V drawing 150 mA (75 mW). Dimensionally, it measures 31.78 mm x 33.68 mm x 5 mm.

3.3.4 Communication units

3.3.4.1 Wireless data transceiver

As mentioned in Section 3.3.1, the primary wireless channel of the avionics is a Bluetooth based serial data link. This is achieved by using a pair of KC111 Bluetooth serial adapters manufactured by KCWirefree. One such module is attached to the avionics and the other is connected to the ground station personal computer.

The modules use the Bluetooth protocol (version 1.2) to facilitate wireless communication between each other (KC Wirefree LLC, 2004). Such communication is slightly more long range than typical Bluetooth applications as the module is based on the Bluetooth Class 1 specification (which offers the highest transmit power of all Bluetooth classes) (Bluetooth SIG, 2004). The manufacturer claims an effective range of 200 m. Tests in an open field disproved the 200 m assertion and, when on board the helicopter during manoeuvres (with the orientation of the ground device to the helicopter device often non optimal), about 50 m was the best range achieved. Additionally, the KC111 supports the Bluetooth Serial Port Profile (which can be used to interact with the remote transceiver as if it were a local communications port). Table 3.6 lists radio frequency (RF) and channel information for the KC111.

Table 3.6. KC111 RF and channel information (KC Wirefree, 2004 and Bluetooth SIG, 2004).

		Unit
Frequency	2.4 (2.4 – 2.4835)[1]	GHz.
Channel bandwidth	1	MHz.
Number of channels	79 (1 600 hops/s)	
Transmit power	100	mW.

[1] Frequency = (2 404 + k) MHz where k is the channel between 0 and 78. Lower guard band is 2 MHz. Upper guard band is 3.5 MHz.

The KC111 is design to be a complete serial cable replacement solution. As such it supports all standard data rates up to 921 kbits/s, features automatic point to point pairing between two preconfigured units, and, RS232 electrical signalling to and from the module.

Figure 3.9 illustrates the use of a pair of KC111 modules in a cable replacement application.

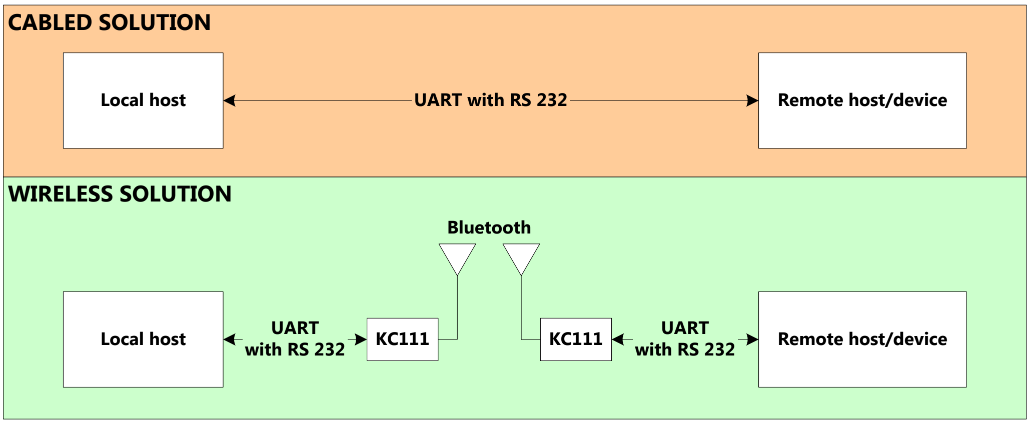


Figure 3.9. Serial communication application with and without KC111 cable replacement.

Use of the KC111 as per Figure 3.9 is subject to adhering to the manufacturer's configuration regimes. Essentially, the devices have two modes: command and bypass. The command mode allows the KC111 to be setup via Hayes like ATtention (AT) commands. Various parameters can be adjusted, including serial port parameters, security settings and automatic pairing configurations. In bypass mode, a pair of modules is functionally equivalent to a serial cable i.e. bytes transmitted by the host are received by the remote host/device and vice versa. Additionally, hardware flow control lines are manipulated appropriately (if hardware handshaking is enabled).

The KC111 draws power from any DC supply between 4 and 10 V and consumes 50 mA nominally (0.25 W at 5 V). When transmitting and receiving, current consumption up to 200 mA has been observed (1 W at 5 V). The device is a complete RF communications unit integrating all components including antenna in a 32 mm x 86 mm x 12 mm package (Figure 3.10).

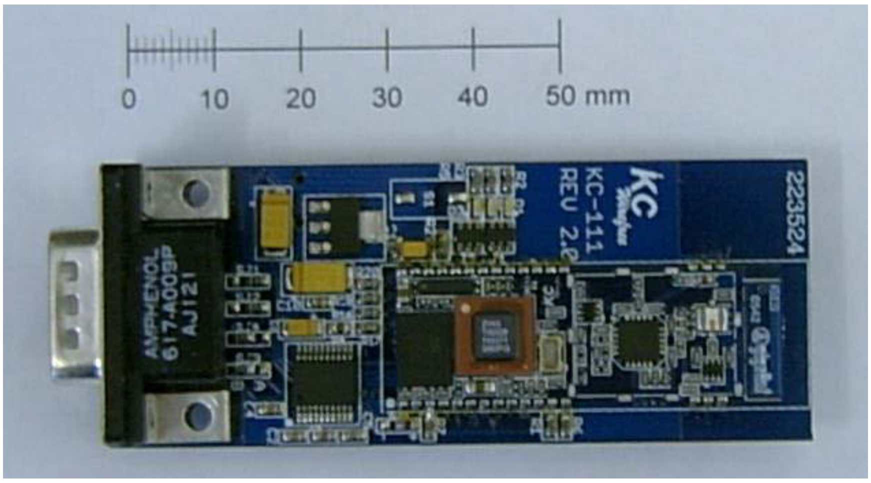


Figure 3.10. KC111 Bluetooth serial adapter.

3.3.4.2 Model helicopter radio system

Although the primary aim of the electronics package is autonomous flight, it must be possible to fly the model helicopter manually also. This is achieved by using a commercially available hobby radio controlled transmitter receiver pair. Both the transmitter and receiver are manufactured by Hitec and employ frequency modulation at 35.030 MHz to exchange data. The transmitter is the Optic 6 (Figure 3.11) and the receiver is the HF8-08RD (Figure 3.12).

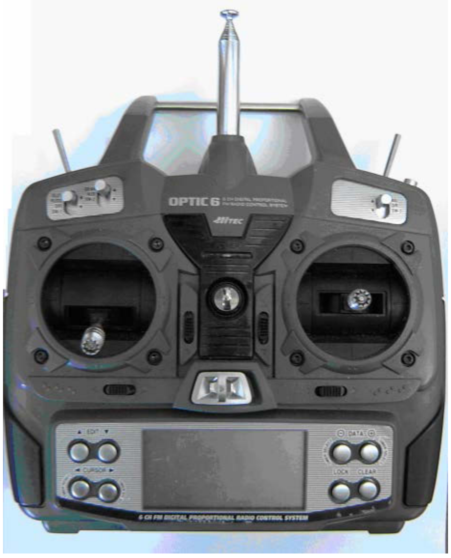


Figure 3.11. Optic 6 radio control transmitter.

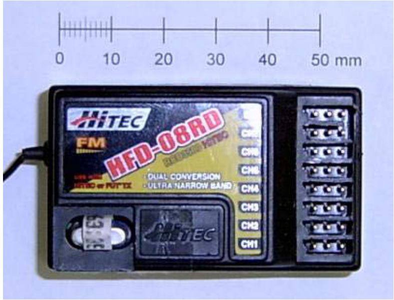


Figure 3.12. HF8-08RD radio control receiver.

The radio system:

1. Supports driving up to six servo motors directly.
2. Incorporates several digital mixing functions e.g. For coupling main rotor to throttle opening.
3. Includes retentive memory to hold all settings.

In terms of the helicopter, this system is configured to control the swash plate (lateral/aileron and longitudinal/elevator), collective pitch, throttle and tail rotor via a rate gyro/yaw rate controller combination. The yaw rate controller is a Futaba FP-G154 (Figure 3.13) proportional controller using the output of the rate gyro for yaw rate feedback.

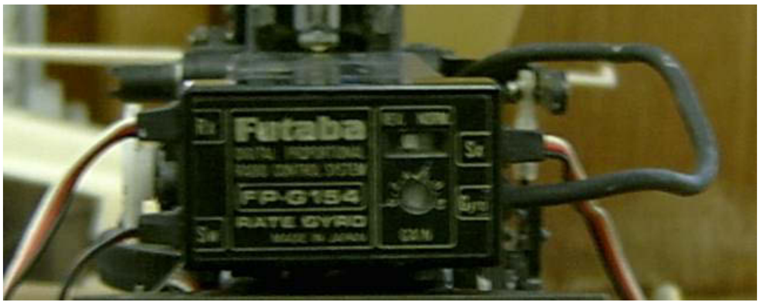


Figure 3.13. FP-G154 digital proportional controller.

The yaw controller has variable proportional gain allowing the sensitivity to pilot yaw commands to be adjusted.

The radio transmitter requires a 9.6 V supply that is provided by its own battery pack. Of more interest to the helicopter power design is the receiver, rate gyro and yaw controller. These all require 4.8 V nominally. The receiver draws 45 mA (216 mW at 4.8 V) and the combined current consumption of the rate gyro and yaw controller is 105 mA (504 mW at 4.8 V). Power use quoted here does not include power drawn by servo motors attached to the receiver.

3.3.5 Actuation and flight switch

The model helicopter control surfaces are actuated via several servo motors. These servo motors are driven either under pilot control from the model helicopter radio receiver or under automatic control from the DSP. Additionally, for the purposes of dynamic model validation and system identification, recording pilot inputs in manual flight is desired.

In terms of servo motor command generation under automatic control, the DSP contains multiple software configurable pulse width modulation (PWM) channels that can be employed to produce the positive pulse signals (technically, radio control servos motor position commands are encoded in the positive pulse width – no information is contained in the duty cycle as with traditional PWM applications). Either these PWM outputs or the radio receiver outputs must be connected to the servo motor command inputs. Such connection is achieved via hardware multiplexers with DSP signalling controlling the selection of PWM outputs or radio receiver outputs. Through the use of appropriate pull down resistors on the selection inputs of the multiplexers, a failure in the DSP would cause these devices to default to pilot control. Although such "fail safe" behaviour has been incorporated into the hardware design, it may be preferable, should an automatic controller ever be operational, for the selection of control source (either pilot or DSP) to rest with the pilot. This could be implemented through an available radio channel on the RC system.

The DSP also provides hardware support for capturing pilot inputs (in the form of positive pulse width durations sent to the servos) through the use of several timer-linked capture channels. The PWM and capture channels greatly simplify the design of the actuation and flight switch sub system, however, both suffer from a significant interfacing problem: DSP inputs and outputs are only 3.3 V tolerant while servo and R/C receiver signals operate with 4.8 V nominally (maximum 6.5 V). Dual supply logic level translators addressed this difficulty.

Figure 3.14 represents the actuation and flight switch sub system. The equivalent circuit, installed on the motherboard (Figure 3.2), is provided in Appendix 1. The circuit requires dual power supplies: 3.3 V and 4.8 V both consuming less than 1 mA each (excluding current sourced/sunk from/to logic outputs).

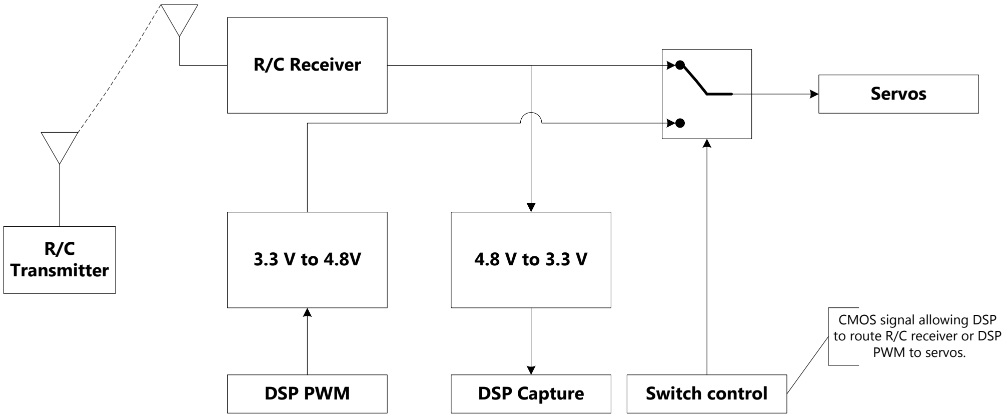


Figure 3.14. Functional view of actuation and flight switch sub system.

3.3.6 Power requirements and power supply design

The complete power supply design must meet the electrical needs of both the avionics and R/C components, be small and light weight and allow reasonable flight durations for the model helicopter. The model helicopter's maximum flight time is constrained to approximately twenty minutes by the fuel tank capacity. As a minimum, the power system should be healthy for this period. To facilitate convenient testing and use, the system was designed for a nominal operational life of one hour. Table 3.7 lists the power requirements for the main devices incorporated in the model helicopter.

Table 3.7. Voltage and current requirements of main components.

Component	Nominal voltage (V)	Maximum current (mA)
Avionics		
ezDSP	5	500.
KC111	5	200.
LassenIQ and antenna	3.3	39.
Compass	5	15.
IMU	6[1]	100.
Barometer	3.3	1.
Flight switch	3.3 and 4.8	0.056[5].
Interfacing/logic	3.3, 4.8, 5 and 6	4.51[6].
R/C components		
Radio receiver	4.8 to 6[2]	45.
Rate gyro and yaw controller	4.8 to 6[3]	105.
Servos	4.8 to 6[4]	3700.

[1] IMU 'BP3010 itself requires 5 V but interfacing/protection circuitry necessitates a 6 V supply.

[2, 3, 4] Nominal voltage of R/C components vary due to R/C battery charging regime differences.

[5, 6] Represents current consumption of logic and interfacing components when not sinking/sourcing current to/from inputs/outputs.

Bulk power is provided by a pair of rechargeable batteries. The avionics is supplied by a two cell Lithium Polymer (LiPo) battery (7.4 V nominally with 1 500 mAh capacity). The LiPo chemistry offers excellent energy density leading to a long lasting weight efficient solution. The R/C components use the four cell Nickel Cadmium (NiCd) battery supplied with the R/C kit (4.8 V nominally with 700 mAh capacity). Although this dual battery approach does not promote weight efficiency, in terms of ease of use and rapid development, this proved the best solution. The R/C components require 4.8 to 6 V but the LiPo battery has a nominal output voltage of 7.4 V. If the LiPo battery were to supply the R/C components also, then some type of regulation is necessary. This approach would become inconvenient when considering components like servo motors and rate gyros that can draw large currents, making regulation challenging. NiCd batteries, however, can supply comfortably up to ten times their C rate without the output voltage degrading. Powering the R/C components directly from the NiCd battery alleviated the problem.

In terms of the avionics, with the LiPo battery providing the raw power, the following supplies were included:

1. Two 5 V supplies – one for the ezDSP and the other servicing other 5 V peripherals.
2. One 3.3 V supply for peripherals.
3. One 6 V supply for the IMU and associated logic/interfacing.
4. One adjustable supply (between 2 V and 5.5 V) for other devices and/or future expansion.

For energy efficiency, switching regulators offer the best solution, but these often (depending on frequency) require large inductors which are heavy and difficult to acquire. To provide the necessary regulation, low drop out regulators were used instead. These represent a compromise between efficiency and ease of use. All regulators employed belong to the REG 104-X family manufactured by Texas Instruments. Key features include up to 1 A output current, drop out voltage of up to 580 mV and TTL compatible enable input. The latter was especially useful for starting/stopping peripherals (or groups thereof) as required, under DSP control, and for ensuring DSP power up guidelines were met – the key recommendation is that the DSP is powered up before any peripherals and that peripherals are powered down before the DSP. With respect to the enable input, the regulator supplying the DSP was always on. Several regulators source in excess of 100 mA. The regulator technical guideline indicated that significant heat sinking be employed in these cases. Size and weight constraints, however, prevented the use of large PCB copper areas and bulky metal heat sinks. Instead a small fan was used to address cooling needs.

The 4.8 V supplies from Table 3.7 come directly from the NiCd battery. Motherboard hardware requiring 4.8 V is used only to interface the R/C system. Should the NiCd battery not be at 4.8 V exactly (due to charging), such connection to the motherboard will ensure compatible logic levels.

Figure 3.15 is a block diagram of the power design used.

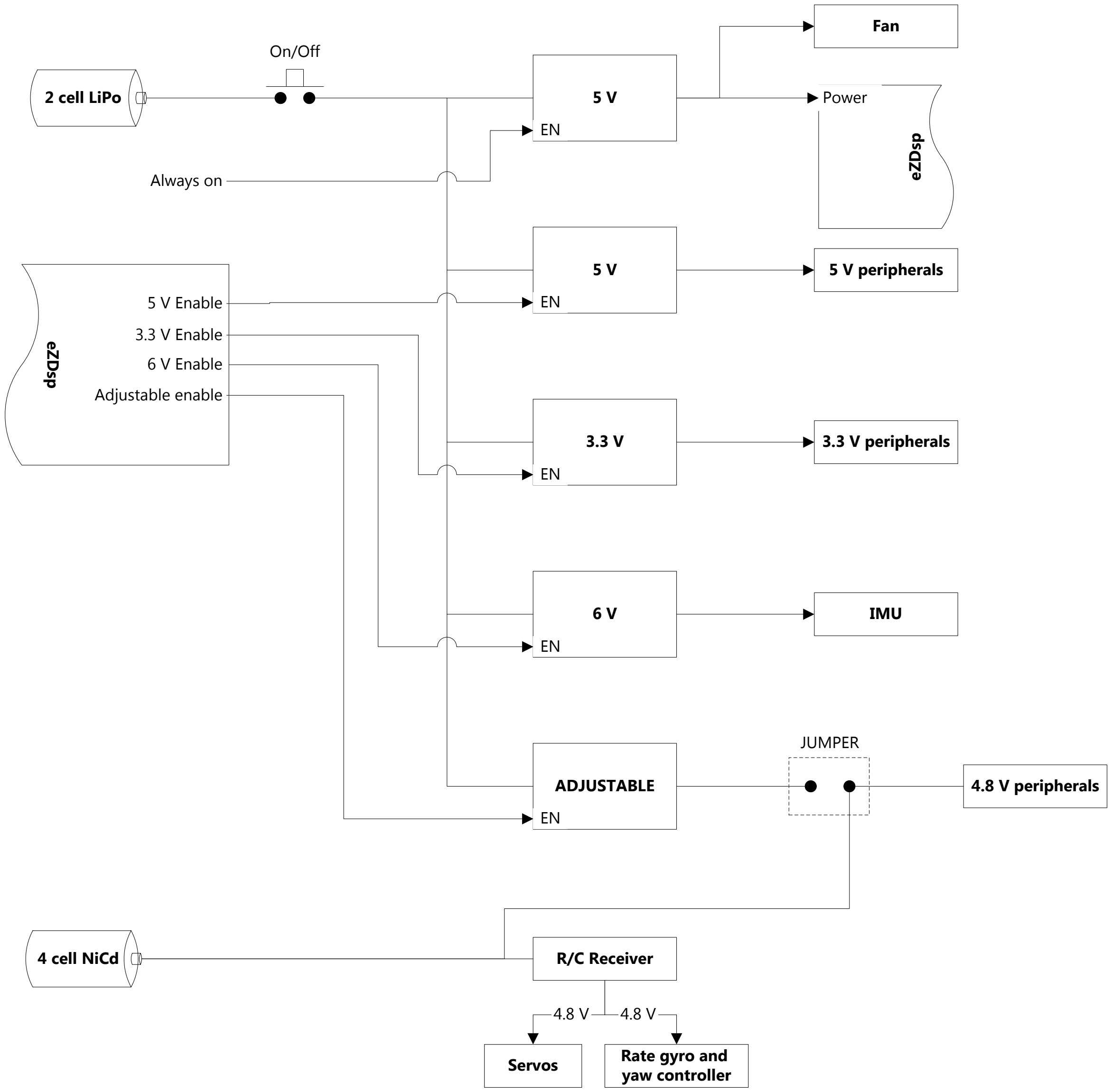


Figure 3.15. Schematic representation of power design.

The jumper in Figure 3.15 allows for the adjustable power to be routed to the 4.8 V supplies (if the adjustable regulator is configured to output 4.8 V).

3.3.7 Ground station

The ground station serves four key objectives: helicopter system monitoring, configuration, command and telemetry logging. The main hardware component of the ground station is a Microsoft Windows based PC. A laptop was selected for portability (small size and battery powered) and custom software was developed to achieve ground station functionality (Section 3.4.2). As mentioned in Section 3.3.4.1, a Bluetooth serial data transceiver is attached to the laptop to facilitate wireless communication with the avionics. A second LassenIQ GPS module and antenna are linked, via RS232 compatible UART, to the laptop. This GPS unit was to be kept stationary through the duration of a flight such that changes/errors in the GPS reading could be determined, communicated to the helicopter avionics and thereby used to correct readings from the on board GPS i.e. A rudimentary differential GPS configuration was attempted. Both the ground side KC111 and GPS system were powered from suitable low drop out regulated power supplies using a rechargeable 9 V Nickel Metal Hydride (NiMH) as the bulk power source.

The final elements of the ground station are the model helicopter radio transmitter (discussed in Section 3.3.4.2) and the pilot. This combination allows for manual control of the helicopter. Figure 3.16 is an overview of the ground station.

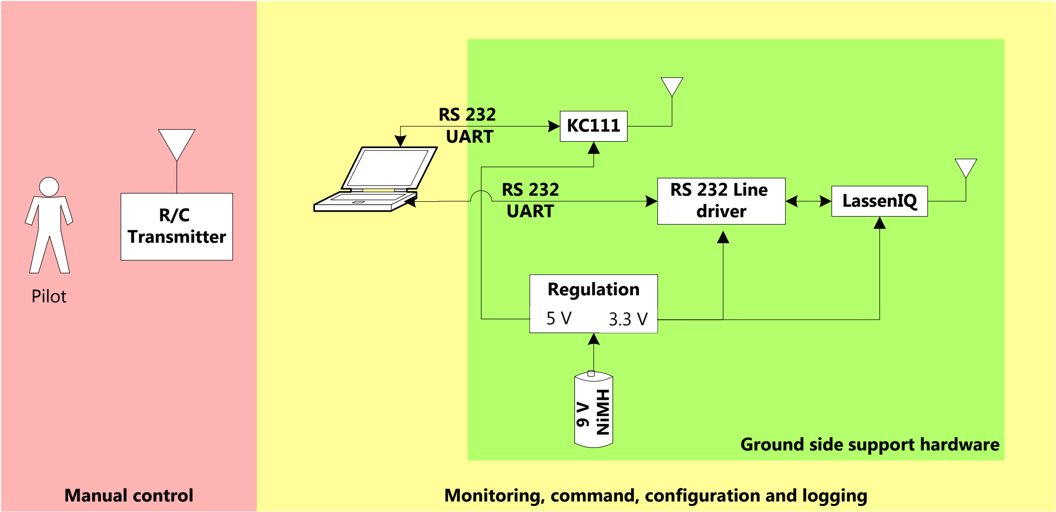


Figure 3.16. Ground station overview.

3.4 Software

3.4.1 Embedded software

3.4.1.1 Design methodology

Texas Instruments DSPs represent, in hardware terms, sophisticated processing cores and peripherals. To realise this hardware potential requires equally powerful software and software development tools. In this regard the 'F2812 is supported by a full software development kit (SDK). The key component of the SDK is Code Composer Studio (CCS) – A 32 bit Windows application integrating all functionality necessary to design, test and debug DSP solutions on Texas Instruments DSPs. The CCS integrated development environment incorporates:

1. Code generation tools (compiler, linker and optimiser)
2. Real time operating system support.
3. Real time debugging tools.
4. Simulator and instruction set simulator.
5. Editing tools with specialised text manipulation functions for convenient code editing.

DSP code can be written in C or C++. For this project C was used, as the object orientation benefits offered by C++ were not required and C still allowed rapid development in a high level language. Development effort was further reduced by employing CCS's configurable real time operating system (RTOS) for the DSP called DSP/BIOS. A RTOS is typically a small piece of code executing on the target processor that manages the real time schedule of an application and assists an embedded application in achieving its real time deadlines. Additionally, RTOSs offer significant benefits in terms of algorithm design simplicity. Algorithms can be written independently of scheduling as this is handled by the RTOS.

The 'F2812 RTOS, DSP/BIOS, provides run time services in a scalable run time library. Such services can be used as application building blocks, forming the back bone of a real time control application, to manage DSP resources and to oversee the real time schedule of an application on any of the TMS320 series of processors. The DSP/BIOS provides a means of structuring very intricate embedded programs, ultimately leading to increased development efficiency and maintainability.

The Texas Instruments RTOS consists of several parts. The real time kernel and the graphical configuration tool are the most important components with regard to this project. The former is DSP resident software while the latter forms part of the PC side tooling. At the heart of the real time kernel is a pre-emptive scheduler for program threads with multi tasking support. The kernel can support multiple thread types, with differing thread priorities facilitating the blocking and pre-emption that is synonymous with multi tasking. Table 3.8 lists the four thread classes.

Table 3.8. RTOS thread types (priority decreases from top to bottom).

Thread type	**Description**
Hardware interrupts	Triggered by hardware/peripheral events.
Software interrupts	Triggered by software posting the interrupt.
Tasks	
Idle tasks	Multiple functions executing in a continuous loop in the background.

The helicopter avionics software is almost exclusively interrupt driven with hardware and external events resulting in actions. In addition, several timer based software interrupts exist for time keeping, periodic ground station reporting, device monitoring etc. Figure 3.17 is a rudimentary representation of the software paradigm. The DSP drops into the DSP/BIOS "idle loop" following setup and initialisation which has been coded into the DSP software entry point – the C main() function. In the idle loop it is possible for idle tasks (as per Table 3.8) to execute, and, it is also possible for hardware and software interrupts to suspend the idle state such that the respective hardware and software ISRs may be executed. The DSP/BIOS RTOS facilitates the tranisition from idle to event processing by determining whether hardware and software interrupts have occurred and subsequently vectoring to the relevant ISRs.

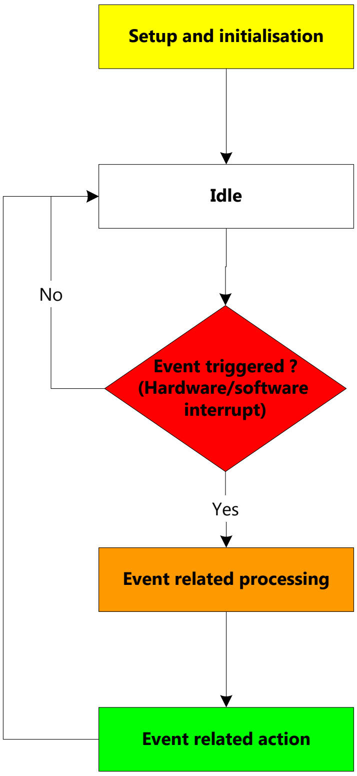


Figure 3.17. Software approach.

The other RTOS component of interest, the Graphical Configuration Tool provides “point and click” access to the static system setup of the DSP. Using this tool, the software designer can edit a large range of parameters including memory map, hardware interrupt service routines and system timers.

3.4.1.2 Embedded application overview

The DSP/BIOS lends itself to control applications through its real time scheduling capabilities. Chief among these is the efficient handling of hardware and software interrupts. As mentioned earlier, the helicopter software is mostly interrupt driven with application code written in C.

Figure 3.18 shows the overall DSP program flow from power on.

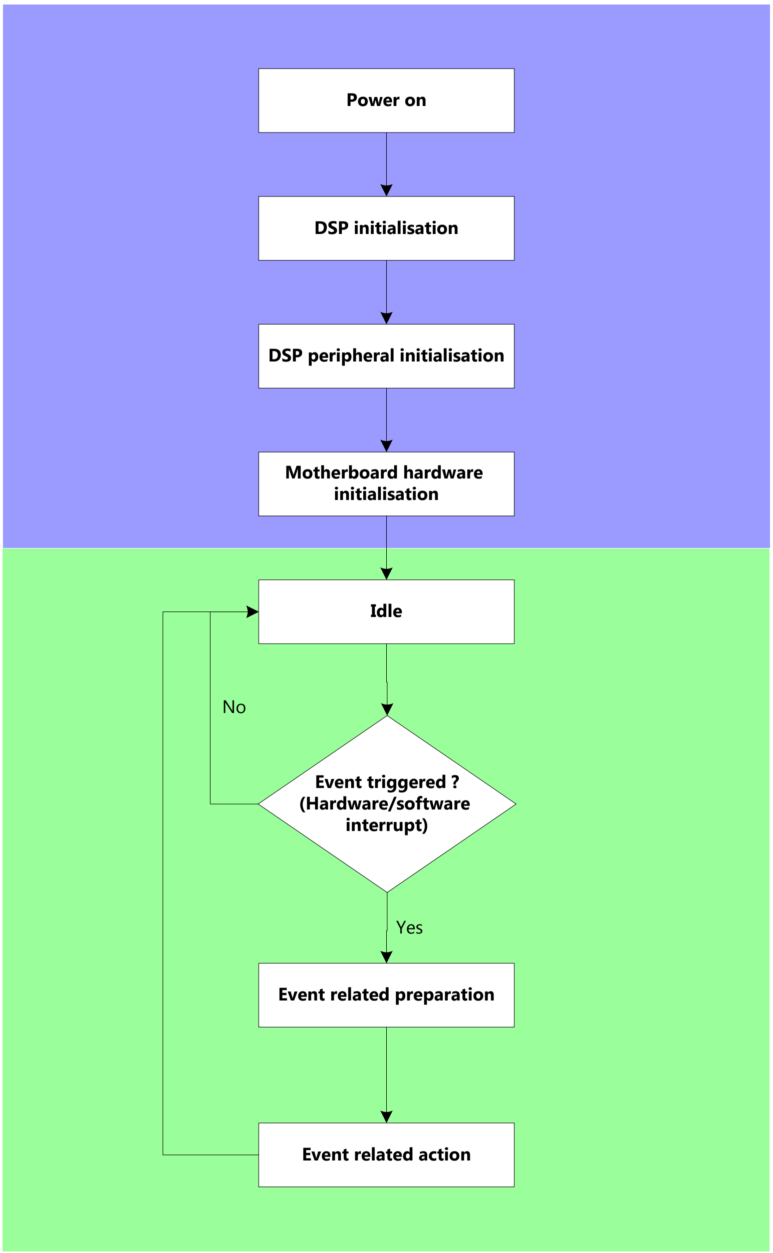

Power on
DSP initialisation
DSP peripheral initialisation
Motherboard hardware initialisation
Idle
No
Event triggered ? (Hardware/software interrupt)
Yes
Event related preparation
Event related action

Figure 3.18. DSP program flow.

As would be expected the DSP application runs mostly in the green section of the diagram with the blue section being executed once on start up. In terms of the green area:

1. Event related preparation refers to any calculation, logical tests or data preparation that must be preformed to facilitate subsequent actions. Typically, this could involve preparing a data buffer or checking software flags to determine the type of action.
2. Event related action refers to any steps taken by the software as a result of the tests performed above and/or using data from the preparation step. Typically, this is the response to the hardware or software interrupt triggers e.g. Executing the Kalman filter navigation routine. Here the trigger would be the reception of new IMU data and the preparation would involve determining whether the system mode was indeed "navigate".

Table 3.9 (following page) provides a categorised list of all triggers, subsequent preparation and actions, as well as DSP interrupt involved for each Trigger-Preparation-Action entry.

Table 3.9. All Trigger-Processing-Action triplets.

Trigger	**Preparation**	**Action**	**Triggered by**
Host communications			
Query	Prepare response.	Send response.	Software interrupt indicating that complete packet has been received. Individual byte reception is driven by a hardware interrupt (mixed trigger).
Flight switch: pilot		Change control source to pilot (manual).	As per mixed trigger..
Flight switch: DSP		Change control source to DSP (automatic).	As per mixed trigger.
Enable IMU/GPS/raw data transmit		Configure buffers/flags to send data.	As per mixed trigger.
Mode change: Idle		Change system mode to idle (no data exchange with DSP).	As per mixed trigger.
Mode change: Align		Change mode to alignment and initialise alignment process.	As per mixed trigger.
Mode change: Set origin		Change mode to set origin.	As per mixed trigger.
Mode change: Navigate		Set mode to navigate.	As per mixed trigger.
Parameter receive: Hardware		Set hardware parameters.	As per mixed trigger.
Parameter receive: Software		Set software parameters.	As per mixed trigger.
Software reset		Reset DSP software.	As per mixed trigger.
Pulse capture	Compute positive pulse width.	Save to pulse width structure.	Hardware interrupt
GPS Packet receive	Determine packet type.	Save fix or DOP data.	As per mixed trigger.
IMU Packet receive	Check mode.	Navigate: Run Kalman filter based navigation routine and send telemetry. Align: Run alignment routine and send telemetry. IMU log: Get/package raw data and send data. Set origin: Get new GPS fix, save as origin and send origin coordinates. Idle: No action.	As per mixed trigger.

It should be appreciated, when considering Table 3.9, that the DSP software takes a specific course of action dependent on the mode parameter. The possible mode parameters which are sent from the ground station host are:

1. Navigation:
   DSP software is processing navigation updates and returning relevant telemetry.
2. IMU logging:
   DSP software is collecting raw sensor data and transmitting these to the ground station.
3. Align:
   Software is generating coarse navigation system initialisation data and returning relevant telemetry.
4. Set origin:
   Software is waiting for a GPS sample such that the navigation system zero can be set. Again relevant data are returned to the host.
5. Idle:
   No processing or actions are in progress apart from system monitoring functions.

It should be noted that several timer based background tasks are always running. Table 3.10 lists these.

Table 3.10. Timer based background routines.

Task	**Description**	**Period**
System time	Causes system time to tick.	1 ms
Device watch dog reset	Clears a series of flags that indicates devices are working (device routines then set these again).	1 s
Battery monitor	Checks avionics battery voltage.	10 s

Many of the triggers discussed are hardware interrupts, others are software interrupts. However, several software interrupts are caused by underlying hardware interrupts e.g. Hardware interrupt service routine reads individual bytes from a serial port and decrements a mailbox. When the mailbox is zero, a software interrupt is triggered.

The embedded application also includes several global structures that hold pertinent application data and parameters necessary for algorithm execution and system functions, e.g. a structure to hold the most recent GPS fix and a structure to hold the general application status.

3.4.1.3 Timing and execution schedule

The major avionics tasks (navigation, logging, alignment and origin setting) are driven repeatedly by the periodic output of IMU samples. The IMU generates readings at 64 Hz, however, only every alternate sample is used, resulting in an effective rate of 32 Hz. This is done to ensure sufficient time for processing tasks between successive samples. It is expected that this will allow for sufficiently fast sampling time for the anticipated bandwith.

The GPS receiver generates fixes at 1 Hz asynchronously to the IMU sample output. As such, in the navigation and origin setting routines, the latest GPS sample is used. Furthermore, every seven IMU samples an altimeter reading is available and is assimilated into the next navigation system measurement update or origin setting. During alignment, the magnetic compass is queried per IMU reading and a heading is generated and included in the angle initialisation calculation.

In logging, navigation, alignment and origin setting modes, telemetry containing sensor data, key states etc., are sent to the ground station with every IMU sample. In navigation mode, data are sent in two stages such that some part of the transmission can be completed while the navigation algorithm is being processed. Remote commands and configuration are applied immediately they are received and such reception can occur at any time. Mode changes resulting from remote commands only take effect at the next IMU sample.

Other events occurring asynchronously that are of concern are the pilot command capture (when flown with R/C radio system) and servo command output. The most recent set of pulses are used when tasks (triggered by IMU update) require these and servo command outputs (in terms of pulses) are performed in the IMU triggered routine that computes their width.

As per Table 3.10, several periodic tasks are also executing continuously.

3.4.1.4 Telemetry, logging and remote operation

To facilitate performance analysis, the embedded application provides telemetry to the ground station for observation and archiving. The embedded application also uses several configurable parameters and modes that must be set. These functions are available over the Bluetooth serial data link. Table 3.11 details telemetry, logging and remote operation functionality provided.

Table 3.11. Telemetry, logging and remote operations supported by the avionics.

Operation	**Data direction**
Software reset.	Host to avionics.
Control source change.	Host to avionics.
Mode change.	Bidirectional.
Hardware parameter change.	Host to avionics.
Software parameter change.	Host to avionics.
DGPS correction reception.	Host to avionics.
Logging/telemetry.	Avionics to host.
Sensor data.	
Navigation states	
Alignment data	
Origin data	
Status information (device/sub system health, battery state).	

The avionics handles data transmission and reception using a packet concept. Transmit packets contain 130 bytes whilst received packets are 47 bytes long. At the maximum wireless data transmission rate of 115 200 bps, transmitted packets require approximately 11.3 ms to be sent. 130 byte send packets were decided upon with this send time and the IMU output rate of 64 Hz in mind. Essentially, when sending telemetry (sensor data, navigation states etc.), the full transmit packet must be sent in the time available between IMU samples (approximately 15.6 ms). It should be appreciated that the IMU output rate was eventually reduced to 32 Hz to allow for navigation algorithm processing. Received packets have been designed to be large enough to accommodate their maximum data payload. Data are sent as bytes with real values being encoded as IEEE 754 single precision numbers occupying four bytes each.

In terms of structure, transmitted and received packets are framed and include identifiers that describe the data contained within the packet. Figure 3.19 and Figure 3.20 show the packet structure for reception and transmission respectively.

Start byte	Identifier	44 data bytes	Stop byte
Byte 0	Byte 1	Bytes 2 to 45	Byte 46

Figure 3.19. General packet structure for reception.

Start byte	Identifier	127 data bytes	Stop byte
Byte 0	Byte 1	Bytes 2 to 128	Byte 129

Figure 3.20. General packet structure for transmission.

3.4.2 Ground station software

3.4.2.1 Design methodology

Ground station software for the Microsoft Windows XP (Windows) based laptop was authored in Microsoft Visual Basic 2005 (VB2005). Microsoft Visual Basic, in general, is development software specifically geared toward rapidly developing applications on 32 bit Windows platforms. It is largely event driven with processing tasks being initiated by user actions and hardware or software triggers.

The manifestation of VB2005 that was employed for the project was Visual Basic Express Edition 2005 which was a freely available development tool aimed particularly at academia and enthusiasts. The integrated development environment provides features for:

1. Creating projects.
2. Generating graphical user interfaces (GUI).
3. Writing Visual Basic code.
4. Testing and debugging code.
5. Distributing the end application.

Like VB2005, the Express Edition is heavily object orientated and class based and integrates tightly with the Microsoft .NET framework. The .NET framework is a fundamental Windows component that supports applications in run and design time comprising:

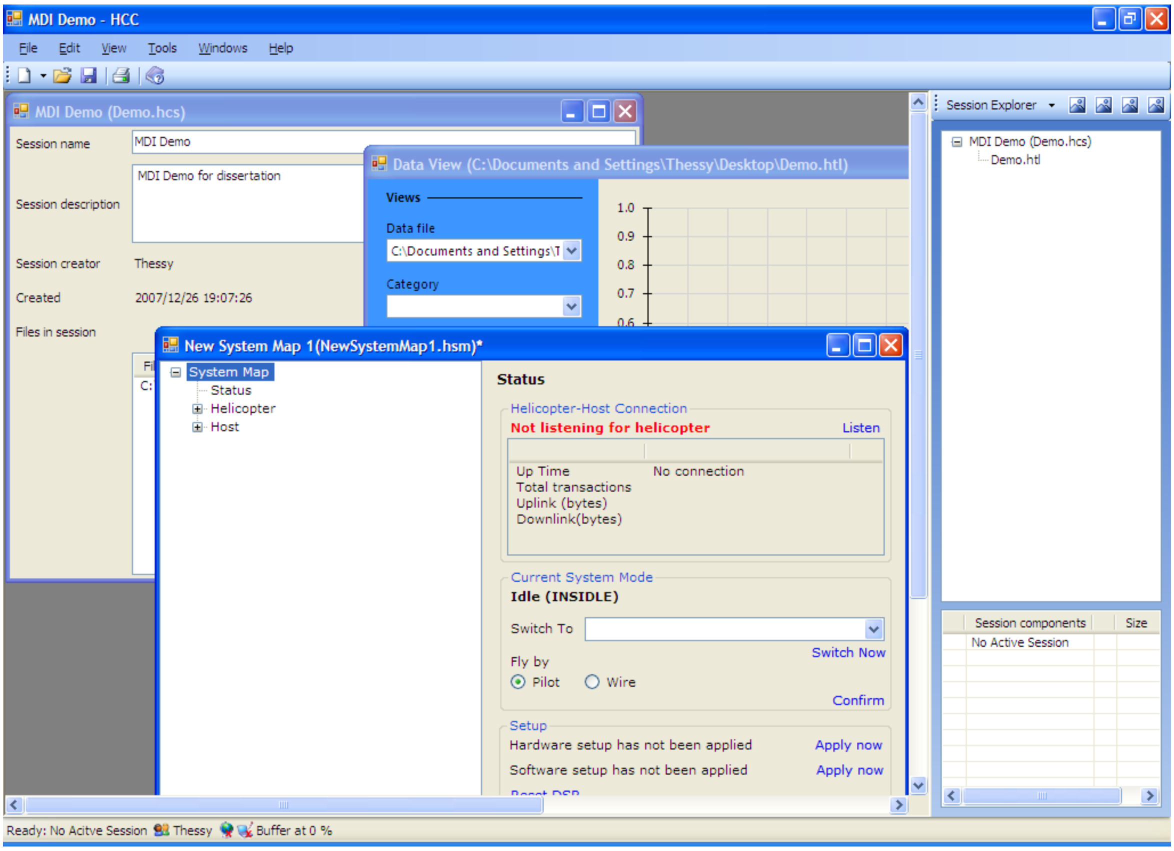
1. The common language run time:
   The foundation of the .NET framework which is essentially a run time component that manages code at execution time, provides core services (e.g. memory management) and that facilitates code security and robustness.

2. The .NET framework class libraries:
   These form a comprehensive object orientated collection of reusable types that can be inserted into an application under development e.g. Classes exist for the command button GUI component and for communication with a computer serial port.

The developed application takes full advantage of the inherent .NET framework support, particularly the class libraries which are used for GUI elements as well as hardware access and processing (e.g. threading class libraries). Furthermore, object orientation was often employed where encapsulation of a certain function or task was required (e.g. a library designed to interface the LassenIQ receiver).

3.4.2.2 Functionality and possible operations

The ground station application is written such that the main window is a multiple document interface (MDI) environment. This implies that several windows, each with possibly different categories of data and functions, can be contained in the main window (Figure 3.21).



Session name: MDI Demo

Session description: MDI Demo for dissertation

Session creator: Thessy

Created: 2007/12/26 19:07:26

Files in session

Data file: C:\Documents and Settings\T

Status

Helicopter-Host Connection

Not listening for helicopter

Up Time: No connection

Total transactions

Uplink (bytes)

Downlink(bytes)

Current System Mode

Idle (INSIDE)

Switch To

Fly by: ● Pilot ○ Wire

Setup

Hardware setup has not been applied

Software setup has not been applied

MDI Demo (Demo.hcs)
Demo.htl

Session components	Size
No Active Session	

Ready: No Acitve Session Thessy Buffer at 0 %

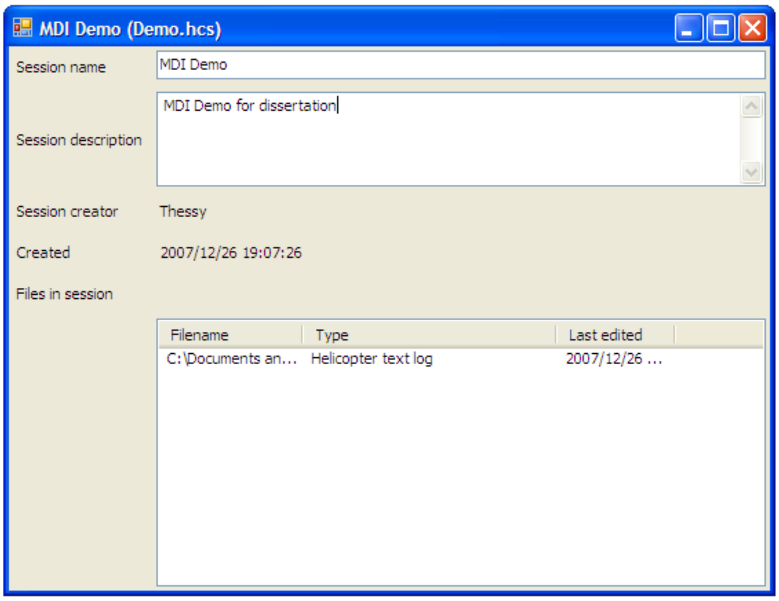
Figure 3.21. Multiple document interface example.

Whenever the helicopter is to be used for experiment, a new session is created and saved within the application and PC. The basic components of a session are two files:

1. The session description file which contains a description of the experiment to be performed (session name, experiment date, session notes etc.) and a list of all other files making up the session.
2. The helicopter system map file which contains detailed information regarding configurable embedded software parameters.

Additionally, multiple optional data log files can be added to the session to facilitate saving of telemetry to disk.

Each of the file types mentioned above can be opened and viewed via the main application window and are deployed into file type specific MDI windows. Figure 3.22 is the MDI window for the session description file.



MDI Demo (Demo.hcs)

Session name: MDI Demo

Session description: MDI Demo for dissertation

Session creator: Thessy

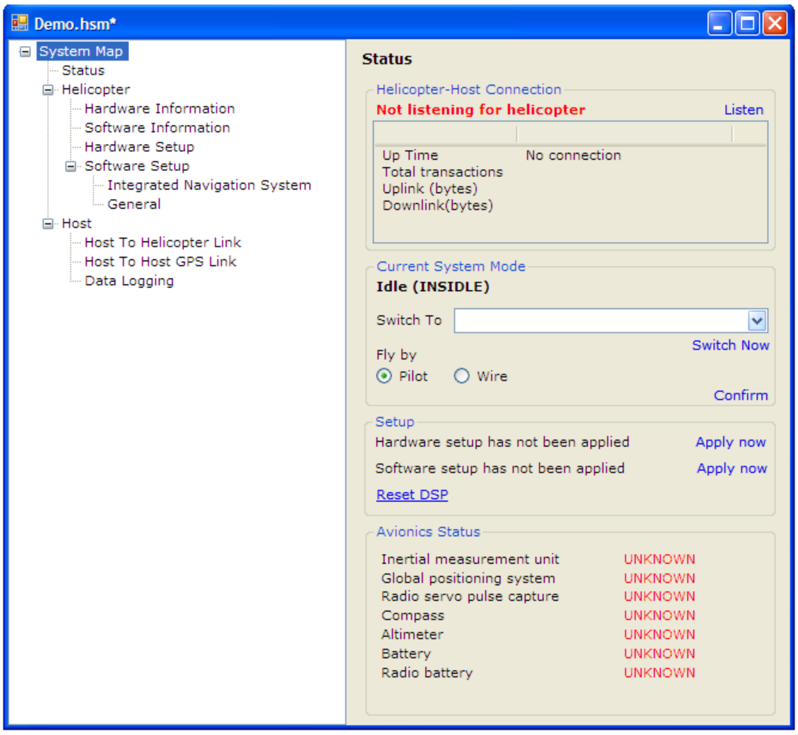
Created: 2007/12/26 19:07:26

Files in session

Filename	Type	Last edited
C:\Documents an...	Helicopter text log	2007/12/26 ...

Figure 3.22. Session description file MDI window.

Figure 3.23 is the MDI window for the helicopter system map file. Here the categorised list in the left hand tree controls the contents of the right hand pane. In Figure 3.23 the helicopter status display is selected.



Demo.hsm*

```
System Map
    Status
    Helicopter
        Hardware Information
        Software Information
        Hardware Setup
        Software Setup
            Integrated Navigation System
            General
    Host
        Host To Helicopter Link
        Host To Host GPS Link
        Data Logging
```

Status

Helicopter-Host Connection

Not listening for helicopter — Listen

Up Time — No connection
Total transactions
Uplink (bytes)
Downlink(bytes)

Current System Mode

Idle (INSIDLE)

Switch To

Switch Now

Fly by: ● Pilot ○ Wire

Confirm

Setup

Hardware setup has not been applied — Apply now

Software setup has not been applied — Apply now

Reset DSP

Avionics Status

Inertial measurement unit	UNKNOWN
Global positioning system	UNKNOWN
Radio servo pulse capture	UNKNOWN
Compass	UNKNOWN
Altimeter	UNKNOWN
Battery	UNKNOWN
Radio battery	UNKNOWN

Figure 3.23. Helicopter system map file MDI window.

Using the helicopter system map MDI window, the following hardware and software settings on the avionics can be adjusted:

1. Hardware settings
   a. Helicopter Bluetooth serial adapter device address.

    b. Serial port settings for wireless data transceiver.
    c. Serial port settings for ground side GPS transceiver.
    d. IMU orientation with respect to helicopter body.

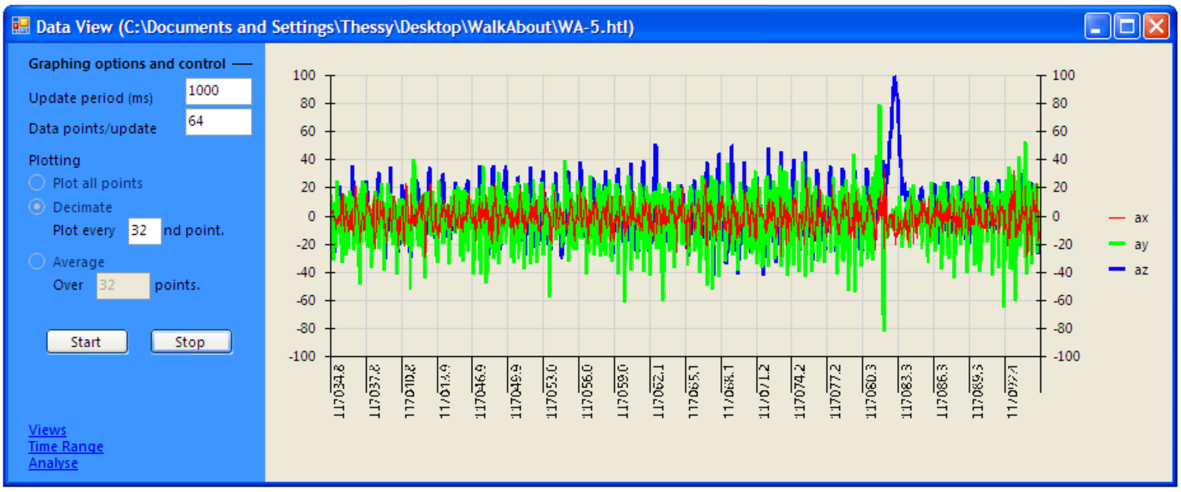
2. Software settings: Inertial navigation system.
    a. Initial state estimates.
    b. Initial state estimate covariances.
    c. GPS measurement covariances.
    d. Barometric altimeter covariance.
    e. Process noise covariances.
    f. Altitude measurement method (it is possible to use either the GPS only or a combination of GPS and barometer – see Section 4.5.2.2.1).

3. Software settings: General.
    a. Number of samples to use for alignment routine.
    b. Earth radii.
    c. GPS user equivalent range errors.
    d. Limits for accelerations and angular velocities.
    e. Compass calibration offset.

Furthermore, the helicopter system map MDI window manages several other tasks during an experimental session including:

1. Wireless connection to helicopter avionics via KC111 Bluetooth serial transceivers.
2. Connection to static GPS receiver, generation of DGPS like corrections and uploading of corrections to the avionics.
3. Uploading of all software and hardware settings to the avionics when such upload is initiated by user command via the GUI.
4. Monitoring avionics sub systems and device health returned by the embedded application for display on status page.
5. Marshalling of all other data received from the avionics and, if necessary, writing relevant data to disk.
6. Providing the user with data logging options and implementing subsequent user choices regarding:
    a. Data logging enable or disable.
    b. Data log file selection if multiple log files have been included in the session i.e. It is possible to include multiple data files per session, but, only one of these files is active at any give time. This allows for separation of data sets e.g. by flight manoeuvre.
7. Avionics mode changing through the use of GUI elements by the user.

Figure 3.24 is the MDI window for the data log file.



Data View (C:\Documents and Settings\Thessy\Desktop\WalkAbout\WA-5.htl)

Graphing options and control

Update period (ms) 1000

Data points/update 64

Plotting

○ Plot all points

● Decimate

Plot every 32 nd point.

○ Average

Over 32 points.

Figure 3.24. Data log file MDI window.

The fundamental purpose of the MDI window for the data log files is telemetry visualisation. This is achieved by displaying graphs of variables of interest from the data stored in the file. The MDI window contains tools for navigating data files by variable category (e.g. accelerations, angular velocities, INS states etc.) and time/sample number. In terms of data logging and data files, the ground side application includes a post processing tool that converts the binary data log files into comma delimited text files suitable for import into Microsoft Office Excel 2003 or Matlab.

4 Navigation

4.1 Overview

Navigation is the science of obtaining virtually instantaneous estimates of position, velocity and attitude with a view to craft guidance. Navigation systems are capable of providing detailed information related to vehicle location, motion and orientation. As such the navigation sub system offers a key function to any unmanned vehicle - the supply of accurate and reliable data for use in controlling the craft, alerting the pilot and managing and completing the mission. Apart from the development phase, where simulation may be used to test and assess a navigation system, these typically operate in real time, employing efficient processing techniques such that a variety of measurements may be incorporated into the position, velocity and attitude solution. The navigation problem is thus of the multiple input multiple output class; several measurements are processed, producing approximations to several quantities. In terms of this project, the quantities to be estimated included the helicopter's three dimensional position, velocity and orientation (attitude).

Section 4.2, "Navigation system architecture", describes various navigation system configurations employed today. Section 4.3, "Strap down INS", details the physical arrangement of the inertial platform as well as techniques necessary to use a strap down system. Section 4.4, "Kalman filtering overview", is a brief Kalman filtering primer touching on those aspects necessary for the navigation system design detailed in Section 4.5, "Navigation filter design".

4.2 Navigation system architecture

Conventional navigation systems are characterised by the navigation solution being generated from data collected from several navigation information sources. Redundancy in the data sources are common, however, the use of sensors with complementary characteristics provides far more value for the integrated navigation solution (Bar-Shalom et al., 2001 and Farell and Barth, 1999).

Contemporary navigation systems are centred about an inertial measurement system whose measurements are aided by data from other navigation instruments. Inertial measurement systems generate position, velocity and orientation estimates by implementing an advanced form of dead reckoning whereby accelerations and angular velocities are integrated. The process of integrating accelerations and angular velocities is neutrally stable i.e. Any inaccuracy or errors in the process will cause the position, velocity and orientation solution to diverge quickly (Angel and de La Parra, 2005). The aiding sources facilitate corrections to the solution by producing additional, usually complementary, information that is only available sporadically and that is fused with the inertial measurement system output to form an aided navigation system (Cramer, 1997 and Trommer and Wendel, 2004).

Durrant-Whyte and Nebot (1999) indicate that the basic anatomy of any aided inertial navigation system features: position sensing (low frequency); rate sensing (high frequency) and a model of the system. Low frequency position measurements are usually band limited signals corrupted by noise assumed to be broadband. The recovery of measurements requires low pass filtering causing data at frequencies beyond the corner frequency to be unusable. High frequency position measurements, typically in the guise of velocity, acceleration and/or angular velocity, facilitate the generation of position estimates beyond that of the corner frequency mentioned above. Such measurements should have sufficient bandwidth to be useful to the vehicle control loops and to facilitate attitude estimation. In the long term, rate sensing, and subsequent integration to produce position estimates, provides a poor navigation solution. In low dynamic environments motion and noise become indistinct. The need for the high frequency information to be augmented by the low frequency data is clear: Fusion of data from complementary sensors can exploit the benefits of each device, cancelling the deficiencies of some and resulting in an improved navigation solution. For example, the drift of position estimates from integration of high frequency acceleration measurements can be bounded by regular position estimates from a low frequency sensor. The system model provides the mechanism by which rate information can be linked to position information e.g. via the basic equations of kinematic motion.

As mentioned earlier most navigation systems utilise an IMU. In the navigation loop the IMU is the high frequency rate sensor. Low frequency sensing, for the type of vehicle used in this project, is usually provided by the GPS. Navigation performance of sensors is quoted in terms of accuracy, availability, integrity and continuity. To ensure a successful aided inertial navigation system, it is imperative that the collection of sensors be complementary in these broad categories. IMU derived position estimates are characterised by poor long term accuracy (unbounded error growth with time) whilst GPS readings have bounded error characteristics (Cramer, 1997). In terms of availability, integrity and continuity, GPS relies on the operation of a complex space infrastructure and the ability of the GPS receiver to communicate with GPS satellites. Due to the dependence on external factors, certain environments are unsuitable for obtaining GPS readings and as such this data can only be used sporadically. Conversely, the IMU (and supporting components that generate position estimates from rate data) is a completely self contained navigation device capable of providing a navigation solution continuously (Trommer and Wendel, 2004). It would appear that the GPS is the ideal foil for the IMU and indeed aided inertial navigation systems based on inertial measurement/GPS fusion offer improved performance when compared with either component alone.

4.2.1 Fusion configurations

Two fundamental methods exist for combining the information provided by the complementary data sources of a typical inertial measurement/GPS based inertial navigation system: The feed forward and feed back configurations. Both methods propose the estimation of inertial system errors which are consequently the states of the relevant Kalman filters. In both techniques, aiding source errors and inertial measurment errors are used as Kalman filter measurements and, as such, the estimation of errors follows naturally.

Brown (1992) provides a treatment of both schemes with emphasis on the Kalman filter. In terms of this, the feed forward method can be described by Figure 4.1.

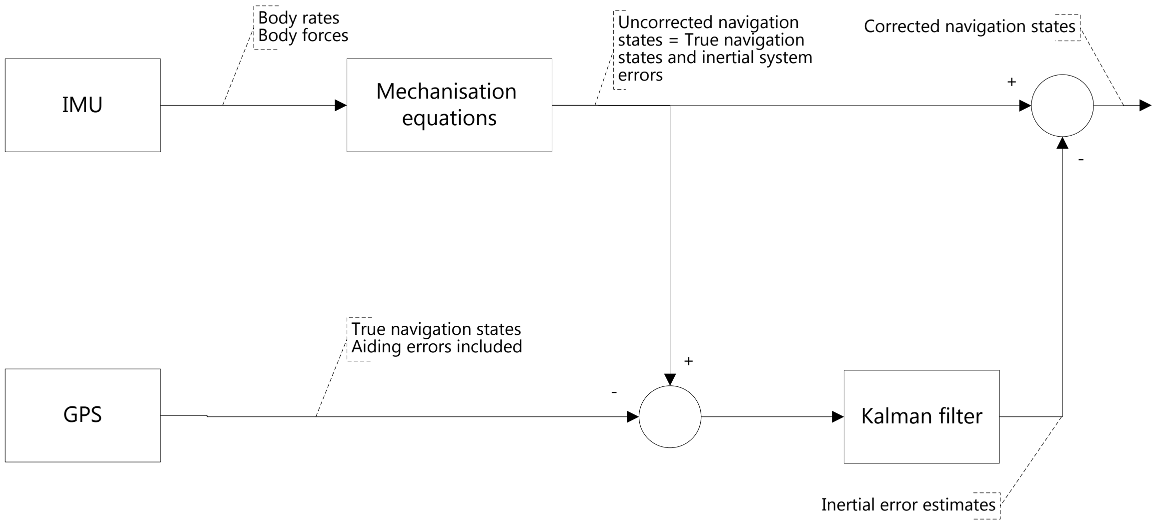
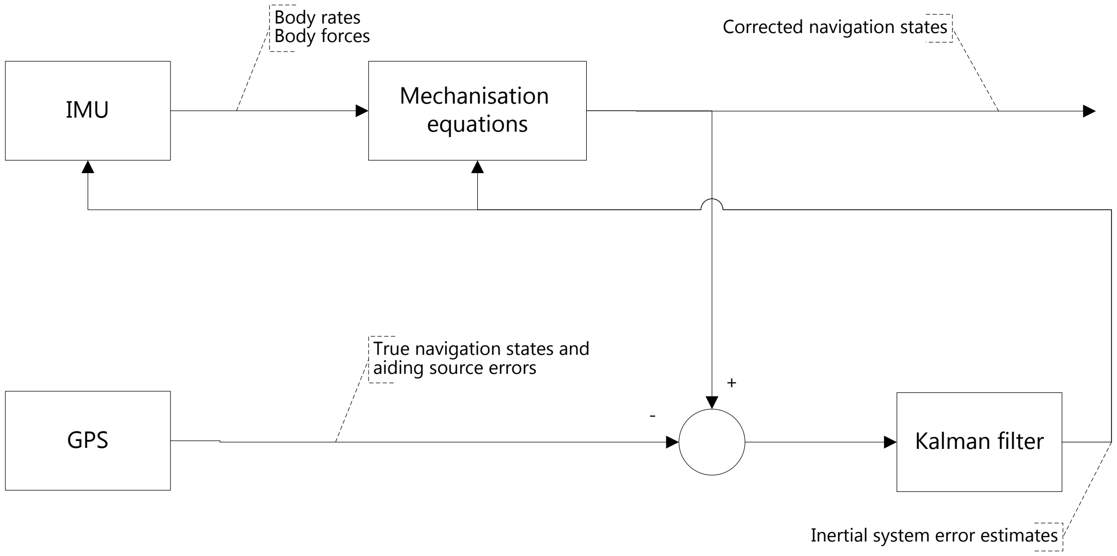


Figure 4.1. IMU/GPS combination via the feed forward method.

This method represents an open loop implementation where the Kalman filter is responsible for estimating inertial system errors. As is evident from Figure 4.1, the actual navigation quantities pass through the system without filtering (system errors are filtered). Such lack of distortion or lag is crucial in systems requiring high dynamic response and the feed forward configuration is therefore preferred in applications with high g manoeuvres. A major drawback of the feed forward technique, however, is that its success depends critically upon employing a costly high quality IMU. As there is no feedback to provide compensation to the IMU and mechanisation, the errors in the navigation states output by the mechanisation grow without bound. With medium to low accuracy IMUs, the error can become intolerable even for short mission times. Additionally, the measurement fed to the filter can become large enough to violate the linearity of the filter.

A better approach, in terms of achieving the project objectives with low cost sensors, is the feedback method presented in Figure 4.2.



Body rates
Body forces
Corrected navigation states
IMU
Mechanisation equations
True navigation states and aiding source errors
GPS
+
-
Kalman filter
Inertial system error estimates

Figure 4.2. IMU/GPS combination via the feedback method.

In the feedback method, estimates of the inertial navigation system errors are fed back to the IMU and mechanisation equation sub systems with a view to making corrections at the sub system level. Medium or low accuracy IMUs can then be used. The outputs from the IMU and mechanisation are corrected at each iteration, thereby preventing the pathological unbounded growth characterising the feed forward technique. These corrections are the Kalman filter outputs which arise from an estimate of the inertial system errors. In the Kalman filter, the estimates are generated either by pure prediction or, when an aiding source measurement is available, by a statistically optimal blending of the measurement and the prediction.

4.2.2 Coupling approaches

Farell and Barth (1999) describe the coupling approach as determining the flow of information within the IMU/GPS fusion. Two basic coupling approaches exist for integrating IMU and GPS: loose and tight coupling.

The loose coupling approach, illustrated by Figure 4.3, is characterised by the GPS generating independent navigation solutions possibily through the use of its own filter.

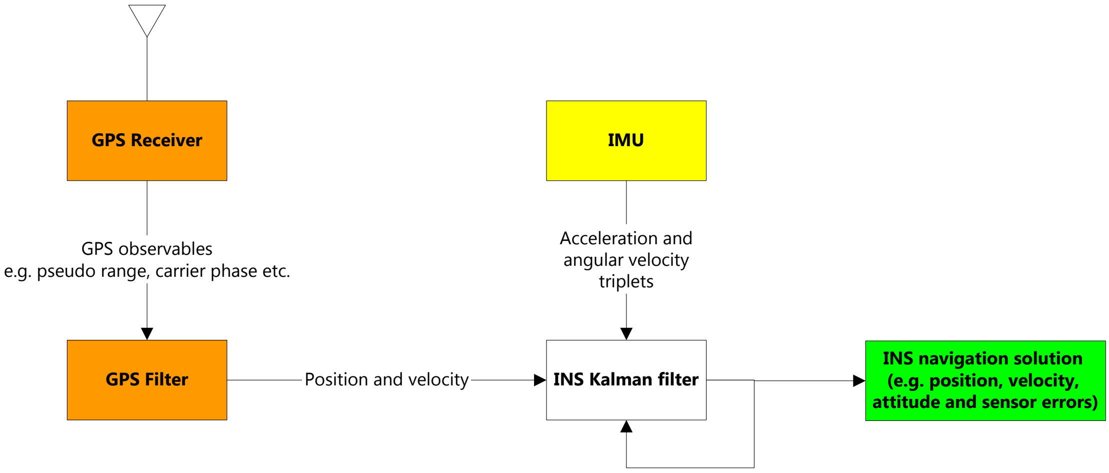


Figure 4.3. Schematic representation of loose coupling (Grewal, Andrews and Weill, 2001).

With this scheme, the INS Kalman filter does not include GPS state variables as part of the INS filter state vector. Instead, the navigation outputs of the GPS are employed as measurements to the INS Kalman filter, thereby using the GPS outputs for estimating IMU sensor and navigation variable errors. In terms of implementation, loose coupling provides good solutions if:

1. A high quality IMU (typically a navigation or tactical grade device) is used when long GPS outages are expected; or,
2. A low quality IMU (typically a low cost MEMs based device) is used when GPS outages are short and infrequent and if the feedback configuration (Section 4.2.1) is used, allowing the GPS reading to recalibrate the navigation solution at each GPS update epoch.

Tight coupling (Figure 4.4) is characterised by the GPS pseudo ranges and/or carrier phase and/or Doppler and satellite data being used directly in the INS Kalman filter.

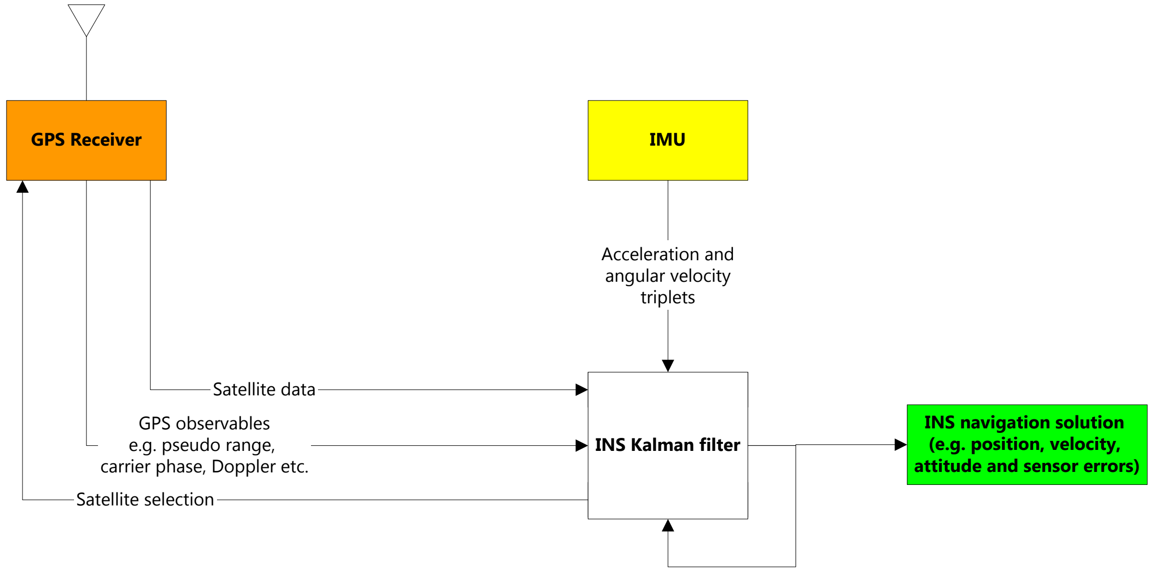

GPS Receiver
IMU
Acceleration and angular velocity triplets
Satellite data
GPS observables e.g. pseudo range, carrier phase, Doppler etc.
Satellite selection
INS Kalman filter
INS navigation solution (e.g. position, velocity, attitude and sensor errors)

Figure 4.4. Schematic view of tight coupling (Grewal and Andrews, 2001).

Farell and Barth (1999) provides a slightly different interpretation where the INS provides vehicle to satellite pseudo ranges and the GPS receiver generates corresponding measurements that are processed by the Kalman filter. In terms of nomenclature used in Farell and Barth (1999), the technique presented in Figure 4.4 is a refined form of loose coupling, however, the naming of Grewal, Andrews and Weill (2001) and Grewal and Andrews (2001) is used here, where loose coupling implies that the coupling variables are on the position and velocity level, and, tight coupling implies coupling variables on the pseudo range level. Farell and Barth (1999) consider aiding of carrier tracking loops with data from the INS solution to be essential for tight coupling.

The pros and cons of loose versus tight coupling cited by Grewal, Andrews and Weill (2001), Farell and Barth (1999) and Trommer and Wendel (2004) are listed below:

1. In terms of system integration loose coupling is characterised by simplicity. The GPS receiver can be considered a separate sub system requiring no adjustment. Additionally, intimate knowledge of processing basic GPS observables is unnecessary as is the subsequent implementation of a GPS position solution.
2. With loose coupling the GPS outputs are readily comparable with INS outputs (position, velocity and attitude if available).
3. Tight coupling generally results in a better solution as basic GPS observables are not as time correlated as GPS position and velocity measurements, and as the GPS position error is coloured. As neither covariance structure is known to the loosely coupled INS filter, sub optimal performance can result.
4. With tight coupling, partial GPS aiding of the INS is possible even with incomplete satellite constellations (less than four satellites). The raw GPS measurements to visible satellites can be utilised somewhat. This is unlike the loosely coupled approach, which, in a similar scenario, must ignore the GPS completely, leaving the INS unaided.

4.3 Strap down INS

4.3.1 Introductory concepts

Many flavours of inertial navigation systems exist, however, all can be categorised, with respect to the physical arrangement of inertial sensors, as either gimballed/mechanised platform or strap down inertial navigation systems. The former, although not primarily the subject of this section, is the traditional approach and is discussed for comparison with the strap down technique.

In the mechanised platform system, the rotation of a rigid frame with rotation bearings (the gimbal) isolates the inner section of the frame from vehicle rotations occurring about the bearing axes. Typically, slight frame rotations (due to force imbalance or bearing friction) are detected by gyroscopes on the inner section. Subsequently, restoring torques are provided by a feedback loop that attempts to cancel all frame rotations. Mechanically, the restoring torques are provided by servo motors – the so called gimbal actuators. Such cancellation is key to maintaining the alignment of the mechanised platform frame with the navigation frame. Figure 4.5 shows a typical gimballed setup.

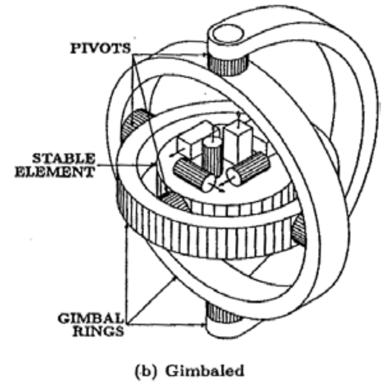


Figure 4.5. Example of mechanised platform system (Grewal, Andrews and Weill, 2001).

The method of Figure 4.5 results in the platform frame experiencing no rotation relative to the navigation frame (provided it was initially aligned with the navigation frame). Suitably aligned and orthogonally mounted accelerometers can then measure force in the navigation coordinates directly. Position and velocity can then be obtained by appropriate scaling and integration, and attitude is given by the relative angles between the vehicle and platform axes (Farell and Barth, 1999).

Salychev (1998) describes the gimballed approach as providing a physical realisation of the navigation frame (as defined in Section 4.3.2.1.2) through the use of a three axes gyroscope stabilised platform. This platform also uses a triad of orthogonal accelerometers. Conversely, the strap down approach, which is of more interest to this project, resolves the navigation frame analytically via digital calculations (primarily the six degree of freedom equations of motion). The calculation inputs are accelerometer and rate gyroscope readings from an inertial sensor cluster mounted directly onto the frame of the vehicle. The strap down instrument axes (accelerometer and rate gyroscope triplets) are aligned as closely as possible with the vehicle body frame (as defined in Section 4.3.2.1.3). Weston and Titterton (2000) refer to the strap down system as the electronic analogue of the mechanised platform system. This is an apt description when considering that rate gyroscopes in the strap down approach are employed to maintain a coordinate transformation from the accelerometer triad axes to directions in the navigation frame. In the mechanical platform system the rate gyroscopes are used to physically hold the accelerometer axes coincidental with the navigation frame. Figure 4.6 juxtaposes the strap down and mechanised platform approaches with respect to this electronic/mechanical comparison. The yellow area illustrates functionally equivalent components. The black blocks show functions performed in the on board computer.

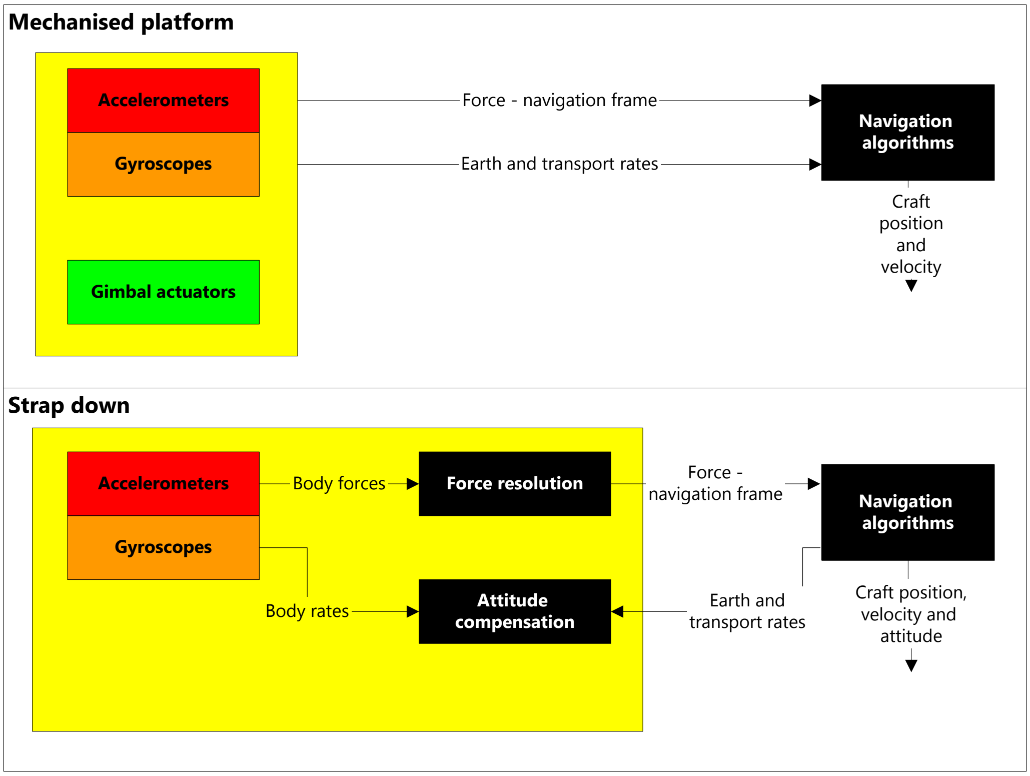


Figure 4.6. Schematic view of mechanised platform and strap down systems (Weston and Titterton, 2000).

Farell and Barth (1999) indicate that the choice between these competing methods be made taking cognisance of the following:

1. Cost of sensors.
2. Actuation versus computing.
3. Size and power consumption.
4. Accuracy needs.

In terms of cost, the gimballed setup is an intricate electro-mechanical device that is difficult to manufacture and consequently expensive. Strap down sensors tend to be cheaper but the complete navigation solution requires a more powerful on board computer. However, for low cost, the strap down approach is still favoured. Point (2) goes to robustness and reliability with the mechanised platform system comprising many moving mechanical parts while the strap down system relies heavily on the computational bandwidth and software quality of the on board computer. This computer must maintain relationships between coordinate systems (vehicle, navigation and inertial), perform integrations and produce eventual navigation results. The fact that the on board computer must maintain the vehicle orientation analytically, whereas the mechanised platform achieves this mechanically, results in the strap down approach requiring more computational power. With respect to size, the mechanised platform is typically larger than the strap down system due to the characteristic actuated platform. The gimbal actuation motors and requirement of motion render the frame more power hungry also. In terms of Point (4), accuracy, traditionally, standalone INSs requiring high accuracy were highly calibrated, precision machined mechanised platforms. This was fundamentally due to the fact that such systems expose the inertial sensors to a more benign inertial environment than the strap down equivalent where sensor clusters would experience the full dynamic motion of the vehicle (thus necessitating higher bandwidth and potentially noisier sensors). Lately, however, advances in sensor and computer technologies have made the strap down alternative more attractive (Farell and Barth, 1999).

With aiding, strap down systems feature performance comparable or better than mechanised platforms. This coupled with the cost benefit and saving in size and power has lead to a significant shift to the strap down approach.

4.3.2 Helicopter orientation

The helicopter can rotate freely in three dimensional space. In a strap down system, where the inertial sensors are fixed to the helcopiter, acceleration and angular velocity vectors are resolved along this instrument's sensitive axes. These vectors illustrate the net motion of the helicopter relative to an inertial reference frame by measuring acceleration and angular velocity along the helicopter fixed reference axes. While such measurements are the basis of any navigation system, to produce navigation results in a relevant navigation reference frame, these measurements must be converted to that navigation reference frame. Before continuing, it is necessary to define the reference frames required.

4.3.2.1 Reference frames

4.3.2.1.1 Inertial frame

An inertial reference frame is a frame in which Newton's laws of motion always apply. It is a theoretical construct as it is not feasible to achieve the totally non accelerating and non rotating frame (Salychev, 1998). The idealised frame can be approximated, for discussion and computation purposes, by selecting a frame that is stationary (or that is in uniform linear motion) relative to distant stars (Salychev, 1998). Regardless of the exact nature of the inertial frame, it must be noted that the inertial sensors produce accelerations and angular velocities relative to an inertial frame (but resolved along the instrument axes).

4.3.2.1.2 Navigation reference frame

The navigation reference frame will serve as the system's local navigation frame; allowing the translational and rotational motion of the helicopter to be related to the navigation system's origin. The navigation reference frame employed belongs to the class of local tangent plane frames, which define local reference directions for vehicle attitude, velocity and position on or near the surface of the Earth (Grewal, Andrews and Weill, 2001). Typically, such local tangent planes arise from the fitting of a tangent plane to the Earth's reference ellipse at some point (the frame origin) that is relevant for local/navigation measurements. In this project, the navigation reference frame is specifically defined by the x axes pointing north, the y axes east and the z axes down (Figure 4.7). The plane spanned by the x and y axes generates the local tangent plane. The navigation reference frame has been designated $\{x_n \quad y_n \quad z_n\}$.

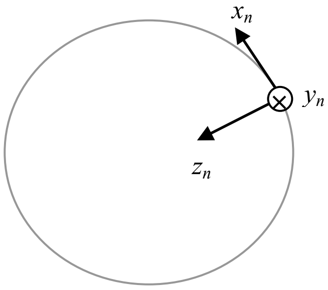


Figure 4.7. Navigation reference frame definition.

As is obvious, the navigation reference frame of Figure 4.7 is stationary with respect to the earth but it is not a true inertial frame as described by Britting (1971) and Salychev (1998). For the short mission times expected and limited spatial range that the model heliocopter will cover, the non stationary nature of the navigation reference frame (in the inertial sense) can be neglected.

4.3.2.1.3 Body frame

The helicopter has, at its centre of gravity, a coordinate frame: the body frame (Figure 4.8). It has been designated $\{x \quad y \quad z\}$. $\{x \quad y \quad z\}$ represent the common flight dynamics axes of roll, pitch and yaw. Relative to the inertial and navigation reference frames, this system of axes rotates and translates with the helicopter fuselage. Helicopter forces and moments are referenced to these axes.

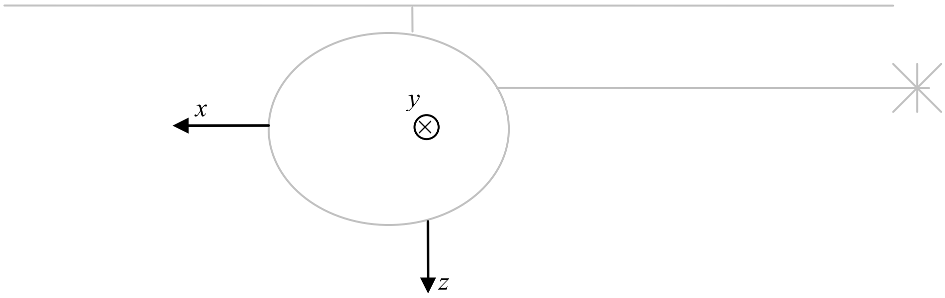


Figure 4.8. Helicopter body frame axes.

A method to relate angualr velocity and acceleration vectors measured in the body frame to the navigation reference frame is required for implementation of a strap down INS. In flight dynamics terms, the basis of this relationship is the relative rotations between the body frame axes and the navigation reference frame axes; commonly known as the Euler Angles (Padfield, 1995). In general, Euler angles allow the comparison of one coordinate system with another, by defining the relative rotation between them. Specifically, assume that the body frame $\{x \quad y \quad z\}$ and navigation reference frame $\{x_n \quad y_n \quad z_n\}$ were initially coincidental but, at some later time, following several rotations of the body frame, it is orientated arbitrarily relative to the navigation frame. Then:

1. The angle, φ, required to realign x_n with x is called roll.
2. The angle, θ, required to realign z_n with z is called pitch.
3. The angle, ψ, required to realign y_n with y is called yaw.

As already mentioned, inertial sensors in a strapdown system measure vehicle motion in the inertial frame. For simplicity, it will be assumed that the instrument sensitive axes are coindental with the body frame axes and thus:

1. The resolution of acceleration measurements along the body frame axes will be given by $\{a_x \quad a_y \quad a_z\}$, where the subscript indicates the body axis involved.
2. The resolution of angular velocity along the body frame axes will be denoted $\{p \quad q \quad r\}$. p is along the body x axis and is commonly referred to as roll angular velocity or roll rate; q is along the body y axis and is commonly referred to as pitch angular velocity or pitch

rate; and, r is along the body z axis and is commonly referred to as yaw angular velocity or yaw rate.

Both the navigation reference and body frames are right handed frames, with positive rotations being counter clockwise. Section 4.5.2.1 uses the frames mentioned here in the development of the navigation state mechanisation equations.

4.3.2.2 Orientation problem

Britting (1971) makes the observation that the body frame origin and the navigation reference frame origin are seldom coincident. This necessitates a transformation to relate motion in the body frame to the navigation system. Since the body frame is fixed at the helicopter's centre of mass, its orientation relative to the navigation reference frame changes. More importantly, as the forces and moments (and resulting accelerations and rotations) are referenced to the body frame, some transformation is required to relate helicopter motion to the "stationary" navigation reference frame (where such motion would be of interest for navigation and attitude determination purposes). This transformation is achieved using directional cosine matrices (DCM) based on the Euler angles.

4.3.2.2.1 Formulation of the directional cosine matrices

(This discussion, including Section 4.3.2.2.2, follows from Padfield, 1995)

Assume that:

1. The navigation reference frame $\{x_n \ y_n \ z_n\}$ has unit vectors $\{i_n \ j_n \ k_n\}$.
2. The body axes $\{x \ y \ z\}$ has unit vectors $\{i \ j \ k\}$.
3. Initially $\{i_n \ j_n \ k_n\}$ = $\{i \ j \ k\}$.

The formulation of the DCM will begin by rotating the body frame about k_n by angle ψ (yawing) (Figure 4.9). Assume that this generates a new set of unit vectors, $\{i_1 \ j_1 \ k_1\}$ representing the rotated position of $\{i \ j \ k\}$. Note that angles are positive in the counter clockwise direction (as is consistent with a right handed coordinate system).

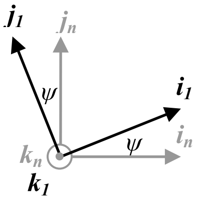


Figure 4.9. Body axes rotated about k_n.

Trigonometry allows the new coordinates, $\{i_1 \ j_1 \ k_1\}$, to be expressed in terms of the navigation reference frame, $\{i_n \ j_n \ k_n\}$,

$$\begin{bmatrix} i_1 \\ j_1 \\ k_1 \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_n \\ j_n \\ k_n \end{bmatrix} = \mathbf{\Psi} \begin{bmatrix} i_n \\ j_n \\ k_n \end{bmatrix} \qquad 4.1$$

Where matrix $\mathbf{\Psi}$ represents a transformation from the navigation frame to the new one.

Now, suppose the body frame, which is coincidental with $\{i_1\ j_1\ k_1\}$, is rotated about the j_1 axis, through angle θ (pitching), generating unit vectors $\{i_2\ j_2\ k_2\}$. This scenario is represented in Figure 4.10.

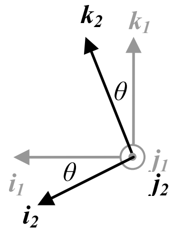


Figure 4.10. Coordinate frame rotated about j_1.

The $\{i_2\ j_2\ k_2\}$ system can be expressed in terms of the $\{i_1\ j_1\ k_1\}$ system,

$$\begin{bmatrix} i_2 \\ j_2 \\ k_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} i_1 \\ j_1 \\ k_1 \end{bmatrix} = \mathbf{\Theta} \begin{bmatrix} i_1 \\ j_1 \\ k_1 \end{bmatrix} \qquad 4.2$$

Where matrix $\mathbf{\Theta}$ represents a transformation matrix relating the $\{i_2\ j_2\ k_2\}$ coordinate system to the $\{i_1\ j_1\ k_1\}$ coordinate system.

Finally, suppose the body frame, which is now coincidental with $\{i_2\ j_2\ k_2\}$, is rotated about the i_2 axis, through angle φ (rolling), generating unit vectors $\{i_3\ j_3\ k_3\}$. This scenario is represented in Figure 4.11.

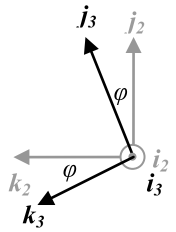


Figure 4.11. Coordinate frame rotated about i_2.

Again the $\{i_3\ j_3\ k_3\}$ system can be expressed in terms of the $\{i_2\ j_2\ k_2\}$ system,

$$\begin{bmatrix} i_3 \\ j_3 \\ k_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} i_2 \\ j_2 \\ k_2 \end{bmatrix} = \mathbf{\Phi} \begin{bmatrix} i_2 \\ j_2 \\ k_2 \end{bmatrix} \qquad 4.3$$

Where matrix $\mathbf{\Phi}$ represents a transformation matrix relating the $\{i_3\ j_3\ k_3\}$ coordinate system to the $\{i_2\ j_2\ k_2\}$ coordinate system.

In terms of the orientation of the helicopter, the $\{i_3\ j_3\ k_3\}$ coordinate system represents the body axes after arbitrary rotations of the helicopter or after it has yawed, pitched and rolled. Initially, the body frame and navigation reference frame were coincident, and vectors referred to the body frame could be directly referred to the navigation reference frame. Following the rotations, this cannot be done. A transformation matrix, the DCM, based on Equations 4.1, 4.2 and 4.3 can be formed to relate vectors between the frames. Equation 4.4, shows how this relation holds for the unit vectors.

$$\begin{bmatrix} i_3 \\ j_3 \\ k_3 \end{bmatrix} = \mathbf{\Phi\Theta\Psi} \begin{bmatrix} i_n \\ j_n \\ k_n \end{bmatrix} \qquad 4.4$$

Where,

$$\mathbf{\Phi\Theta\Psi} = \mathbf{\Gamma} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ (\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) & (\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) & \sin\varphi\cos\theta \\ (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) & (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) & \cos\varphi\cos\theta \end{bmatrix} \qquad 4.5$$

$\mathbf{\Gamma}$ represents a DCM transforming vectors from the navigation reference frame to the body frame. It should be appreciated that the order of rotations determines the exact form of the DCM and that changing the order of rotation would result in a different DCM. To transform from the body frame to the navigation reference frame $\mathbf{\Gamma}^{-1}$ must be used. $\mathbf{\Gamma}^{-1}$ can be found easily since $\mathbf{\Gamma}$ is orthonormal ($\mathbf{\Phi}, \mathbf{\Theta}$ and $\mathbf{\Psi}$ are orthonormal) i.e. $\mathbf{\Gamma}^{-1}$ = $\mathbf{\Gamma}^{\mathrm{T}}$ (Beauregard and Fraleigh, 1995).

$$\mathbf{\Gamma}^{-1} = \begin{bmatrix} \cos\theta\cos\psi & (\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) & (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \\ \cos\theta\sin\psi & (\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) & (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{bmatrix} \qquad 4.6$$

As an example, suppose that the body frame has been rotated arbitrarily relative to the navigation frame by angles φ, θ and ψ. Gravity, which in the navigation frame can be represented by $\mathbf{g} = g \cdot k_n$, could now be resolved (by a triad of body axes fixed accelerometers for example) along the vehicle body axes as $\mathbf{a}_{gb}$ using Equations 4.5.

$$\begin{aligned}\mathbf{a}_{gb} &= \mathbf{\Gamma g}\\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ (\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) & (\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) & \sin\varphi\cos\theta \\ (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) & (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) & \cos\varphi\cos\theta \end{bmatrix}\begin{bmatrix}0\\0\\g\end{bmatrix} \\ &= \begin{bmatrix} -g\sin\theta \\ g\sin\varphi\cos\theta \\ g\cos\varphi\cos\theta \end{bmatrix}\end{aligned} \qquad 4.7$$

4.3.2.2.2 Angular velocities and time derivatives of Euler angles

Padfield (1995) mentions one other addition to the orientation problem: finding a relationship between the fuselage angular velocities and time derivatives of the Euler angles. It is obvious that the angular velocities caused by the moments result in the changing Euler angles (Padfield, 1995),

$$p\, i_3 + q\, j_3 + r\, k_3 = \dot{\psi}\, k_n + \dot{\theta}\, j_1 + \dot{\varphi}\, i_2 \qquad 4.8$$

The left hand side of Equation 4.8 represents the angular velocities or rotation rate of an object relative to the inertial frame but resolved along the body frame axes of that object (as defined in Section 4.3.2.1.3). The right hand side of Equation 4.8 represents the rotation rate of the object resolved along the axes created during the discussions of Figure 4.9 to Figure 4.11 (the actual rotation angles are quoted in these but their time derivatives are used above for rates). As both the left and right sides represent the same rotation, the equality is valid.

To facilitate comparison of the components of Equation 4.8, both sides must be in the same coordinate system. This can be achieved by applying the appropriate transformation matrix (or product of matrices):

1. $\dot{\psi}$ is in the $\{i_n\ j_n\ k_n\}$ system and requires the full transformation matrix, $\mathbf{\Gamma}$, encompassing three rotations, $\mathbf{\Phi}, \mathbf{\Theta}$ and $\mathbf{\Psi}$, to express it in the body frame.
2. $\dot{\theta}$ has already undergone the $\mathbf{\Psi}$ transformation, but to express it in the body frame still requires two rotations given by the product $\mathbf{\Phi\Theta}$.
3. $\dot{\varphi}$ has already undergone two rotations and requires only that the rate be transformed from the $\{i_2\ j_2\ k_2\}$ system to the $\{i_3\ j_3\ k_3\}$ system which can be achieved via the $\mathbf{\Phi}$ transformation.

Equation 4.9 (Padfield, 1995) shows the result, where the matrix notation is used for clarity and where the right hand side of Equation 4.8 has undergone the appropriate transformation such that it matches the $\{i_3\ j_3\ k_3\}$ coordinate system.

$$\begin{bmatrix}p\\q\\r\end{bmatrix} = \mathbf{\Gamma}\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathbf{\Phi\Theta}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathbf{\Phi}\begin{bmatrix}\dot{\varphi}\\0\\0\end{bmatrix} \qquad 4.9$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot{\psi}\sin\theta \\ \dot{\psi}\sin\varphi\cos\theta \\ \dot{\psi}\cos\varphi\cos\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}\cos\varphi \\ -\dot{\theta}\sin\varphi \end{bmatrix} + \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} p &= \dot{\varphi} - \dot{\psi}\sin\theta \\ \Rightarrow q &= \dot{\theta}\cos\varphi + \dot{\psi}\sin\varphi\cos\theta \\ r &= \dot{\psi}\cos\varphi\cos\theta - \dot{\theta}\sin\varphi \end{aligned} \qquad 4.10$$

Solving for $\dot{\psi}$, $\dot{\theta}$ and $\dot{\varphi}$,

$$\begin{aligned} \dot{\psi} &= \frac{r\cos\varphi + q\sin\varphi}{\cos\theta} \\ \dot{\theta} &= q\cos\varphi - r\sin\varphi \\ \dot{\varphi} &= p + r\cos\varphi\tan\theta + q\sin\varphi\tan\theta \end{aligned} \qquad 4.11$$

4.3.2.2.3 GPS position outputs to navigation frame conversions

The Lassen IQ GPS receiver generates three dimensional position measurements in terms of latitude, longitude and altitude. The reference frame (Section 4.3.2.1.2) which will be used as the navigation frame for this project requires that measurements be in the north, east and down system. A method to convert the GPS unit outputs to the desired navigation frame coordinates is therefore needed.

A spherical earth model is a suitable starting point for the discussion. With this assumption, it is possible to formulate Equation 4.12 which generates north position readings from GPS latitudes.

$$r_N = R_{Earth}\Delta\nu \qquad 4.12$$

Equation 4.12 is a common arc length calculation (Young et al., 1996) where:

1. R_{Earth} is the radius of the arc (here the radius of the Earth).
2. $\Delta\nu$ represents a change in latitude angle expressed in radians. Changes in latitude, longitude and altitude are computed by initialising the navigation frame origin and subsequently using the initial latitude, longitude and altitude to generate differences ($\Delta\nu$, $\Delta\eta$ and Δh).
3. r_N is the distance travelled from the navigation frame origin in the North direction.

Similarly, east position readings can be determined from GPS longitudes via Equation 4.13 (r_E is east position and η is the longitude).

$$r_E = R_{Earth}\Delta\eta \qquad 4.13$$

As mentioned above, Equations 4.12 and 4.13 depends critically on the fact that the Earth is spherical, which is not the case. A more realistic and accurate Earth model is the ellipsoidal model. As indicated by the terminology, with this approach it is assumed that the Earth is elliptical. As such, the radius in Equations 4.12 and 4.13 is different in the north-south and east-west directions as the describing shape is an ellipse in the north-south plane. Additionally, in both the north-south and east-west directions the radii are a function of the geographic latitude. Grewal, Andrews and Weill (2001) quotes Equations 4.14 and 4.15 for the radii at a given latitude in the north-south and east-west directions respectively.

$$R_N = \frac{b}{a[1-e^2\sin^2(\nu)]^{3/2}} = \frac{a(1-e^2)}{[1-e^2\sin^2(\nu)]^{3/2}} \qquad 4.14$$

$$R_E = \frac{a}{\sqrt{1-e^2\sin^2(\nu)}} \qquad 4.15$$

In Equations 4.14 and 4.15:

1. a is the semi major axis of the ellipse (in the north-south plane).
2. b is the semi minor axis of the ellipse (in the north-south plane).
3. e is eccentricity of the ellipse in the north-south plane given by Equation 4.16.

$$e^2 = \frac{a^2 - b^2}{a^2} \qquad 4.16$$

Figure 4.12 shows where a and b originate from with respect to the ellipsoid Earth model. The gray north-south section on the left of Figure 4.12 intersects the ellipsoid Earth such that the line of intersection is an ellipse. This ellipse is shown on the right of the figure, with the semi major (a) and semi minor (b) axes marked for clarity.

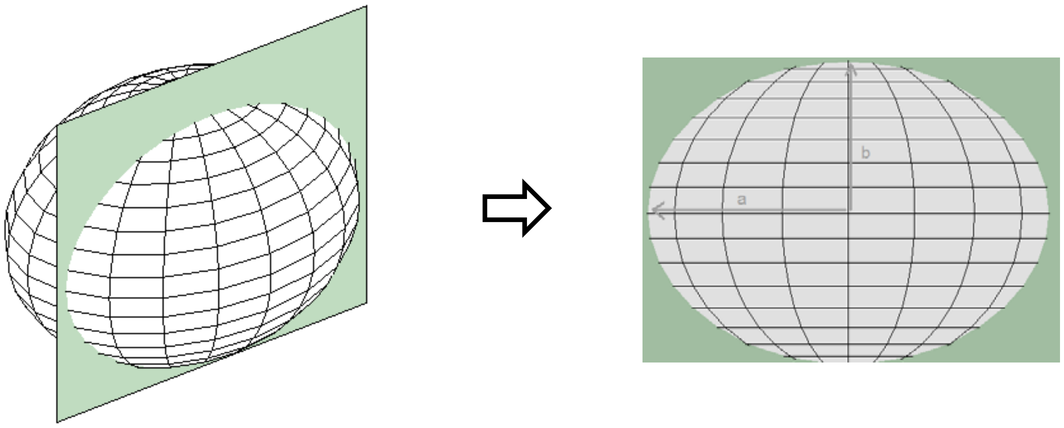


Figure 4.12. Illustration of north south plane generating ellipse.

Equations 4.12 and 4.13 can now be refined to incorporate the non spherical nature of the Earth by replacing the R_{Earth} with the appropriate radius from Equations 4.14 and 4.15 (R_N and R_E respectively).

Finally, the GPS unit generates altitude where up is positive. In terms of the desired navigation frame, down is needed as positive. Equation 4.17 effects the conversion necessary (r_D is the position in the down direction).

$$r_D = -\Delta h \qquad 4.17$$

4.4 Kalman filtering overview

4.4.1 Introductory theory and concepts

The Kalman filter is a set of recursive equations that estimate the instantaneous state of a linear dynamic system, perturbed by white noise, through observation of white noise corrupted measurements, linearly related to the state. Invented in 1960 by R.E Kalman, the filter is suited to applications where all desired variables are not readily available. It provides a technique to infer the unknown information from indirect and often noisy measurements (Grewal and Andrews, 2001). State estimates are generated by optimal (with respect to minimisation of variance) blending of past state prediction with present external measurement. Apart from being a pure state estimator, the filter also propagates the covariance of the estimated variables, thereby providing more complete knowledge of the instantaneous system state.

Kalman filters are almost exclusively implemented on digital computers, with their representation of the estimation problem by a finite number of variables making such implementation possible. It was indeed advances in digital computing that led to the initial popularity and practicability of the Kalman filter for several real time applications (Brown, 1992 and Kaliath et al, 2000).

The practicability of the filter is further enhanced by the fact that it is a recursive algorithm so that the recursive nature of the calculation circumvents the need to store and process all past data at each measurement epoch. The Kalman filter, through recursion, incorporates all past and present information and measurements such that the current system state may be estimated via:

1. Knowledge of the system comprising process dynamics and measurement relationships.
2. Statistical descriptions of:
    a. System noises (acting on the state differential equations).
    b. Measurement noises (acting on the algebraic output equations).
3. Other relevant information concerning variables of interest (Maybeck, 1999).

Figure 4.13 illustrates the need for a filter from a system perspective. The system is usually excited by several noisy inputs with noisy measuring devices capturing some system information.

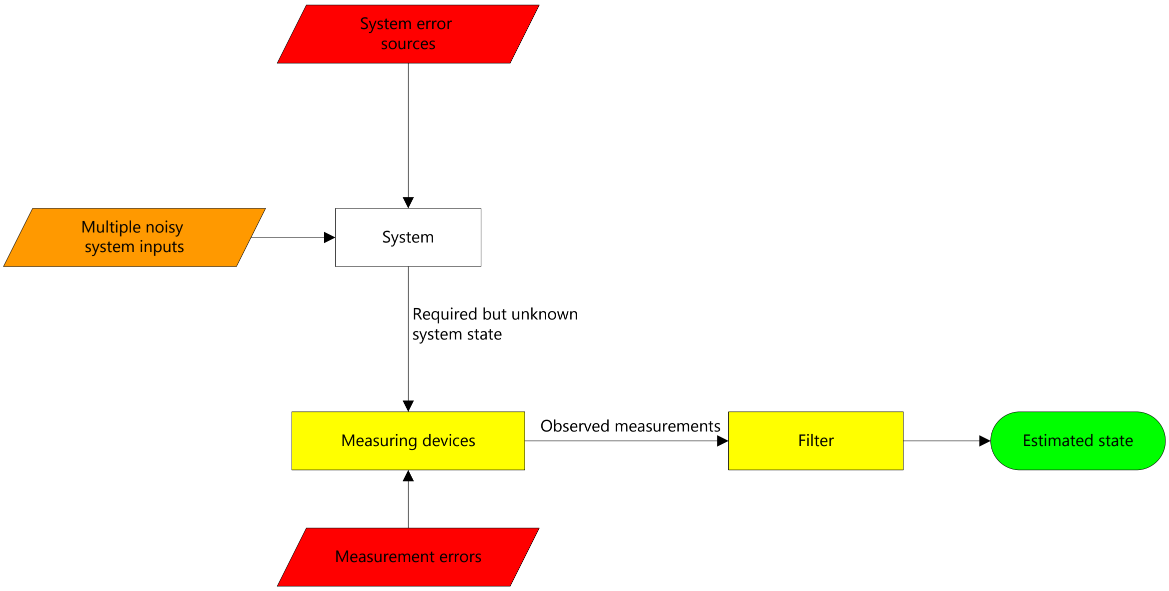


Figure 4.13. Kalman filter application with respect to system state determination (Maybeck, 1999 and Levy, 2002).

Typically, the Kalman filter attempts to achieve optimal estimation of relevant variables by using information provided by a noisy environment. This imperfect environment, from which the variables must be inferred, is often characterised by:

1. Noisy inputs.
2. Assumed state and measurement dynamics that, naturally, cannot model every aspect of the real system.
3. Measurements corrupted by noise, biases and device inaccuracies.

As alluded to earlier, it is also required to extract and process useful information from a variety of measurement sources with differing characteristics. The Kalman filter provides a systematic means of achieving this such that estimates may be generated where the estimation error is minimised statistically. This method is, however, subject to the following conditions:

1. The system model must be linear.
2. Measurement noises must be white.

Condition 1 is not severely limiting in that non linearities are usually overcome by model linearisation about a nominal point or trajectory leading to the linearised or extended Kalman filter (Brown, 1992). The requirement for whiteness is rooted in convenience, rendering the mathematics more tractable. For a physical system with a finite frequency pass band, wide band noise is equivalent to white noise leading to conditions 2 incurring minimum fidelity loss (Maybeck, 1999). Futhermore, although not essential, Gaussian measurement noises are preferred to ensure that the Kalman filter produces optimal estimates in a minimisation of mean square error sense. Failing this, the filter however, does generate the best linear estimate i.e. An estimate that produces the least mean square error considering all possible linear estimators.

Figure 4.14 shows, conceptually, the computational flow of a Kalman filter.

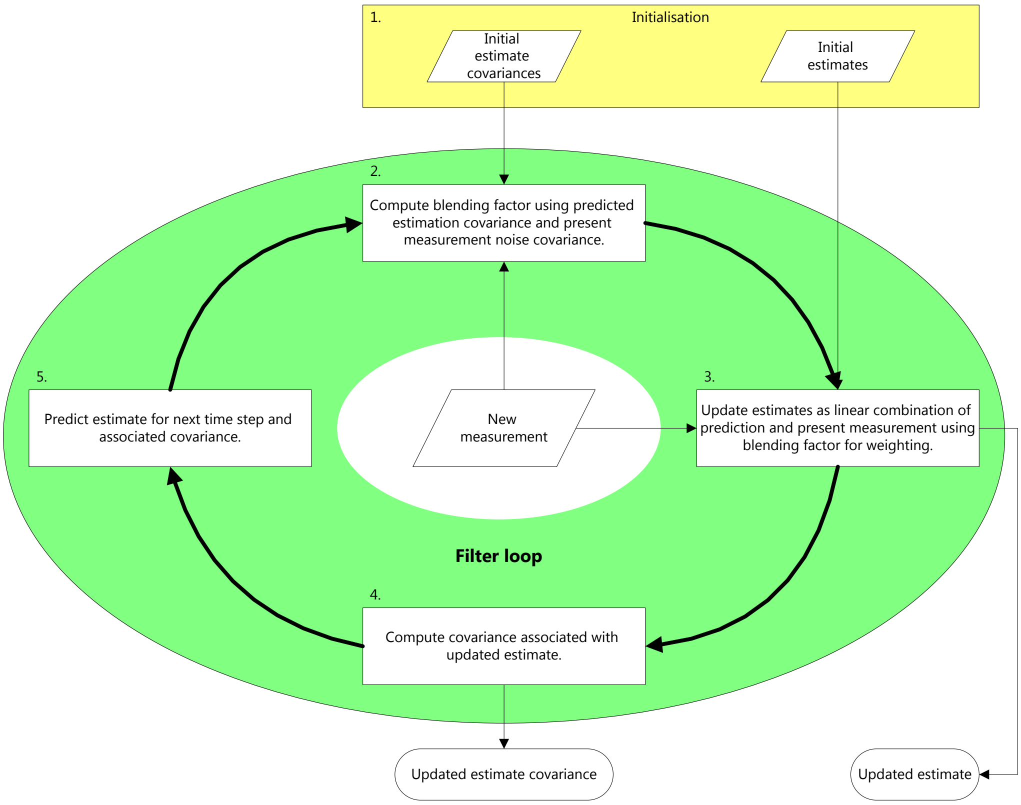

1.
Initialisation
Initial estimate covariances
Initial estimates
2.
Compute blending factor using predicted estimation covariance and present measurement noise covariance.
5.
Predict estimate for next time step and associated covariance.
New measurement
3.
Update estimates as linear combination of prediction and present measurement using blending factor for weighting.
Filter loop
4.
Compute covariance associated with updated estimate.
Updated estimate covariance
Updated estimate

Figure 4.14. Conceptualisation of Kalman filter recursive procedure (Levy, 2002).

The filter is initialised with a known initial estimate and associated covariance. Step 2 involves the calculation of a blending factor (the Kalman gain) that is used to weight the new measurement against the prediction (or initial value in the first iteration) when forming the updated estimate in Step 3. Qualitatively, the Kalman gain will be generated such that smaller measurement noise covariances will result in the measurement influencing the updated estimate more strongly than the predicted estimate. Conversely, a relatively small predicted estimate covariance (or initial estimate covariance for the first iteration) will result in the predicted estimate influencing the updated estimate more strongly than the measurement. Step 4 results in the information contributed by measurements in Step 3 being incorporated into the instantaneous statistical description of the estimates, by generating the updated estimate covariance (ideally, incorporation of measurements should result in smaller state error covariances implying reduced uncertainty). Finally, the filter prepares for the next time step (Step 5). Here predictions of the estimates and associated covariances at the next time step are made. Such projections are effected via noisy linear transformations that tend to increase the estimate uncertainty. Inclusion of information embedded in multiple measurements over multiple iterations has the opposite effect on estimate uncertainty - ultimately resulting in an "uncertainty steady state" where the uncertainty increase in Step 5 is offset by the uncertainty decrease in Step 3.

As is apparent from Figure 4.14 and the discussion above, the filter is a data processing algorithm that is, in terms of implementation, well suited to sampled operation. Consequently, subsequent discussion of the Kalman filter will focus on discrete time aspects and application. For a treatment of the continuous Kalman filter see Brown and Hwang (1992) or Grewal and Andrews (2001).

4.4.2 Applicability to the navigation problem

Kalman filters are effective and efficient stochastic estimators for a wide variety of real time and off line problems. This recursive algorithm has, since its invention, been extremely popular for merging navigation data in an optimal and performance conscious way. Before delving into the detail of the filter, it is appropriate to consider those aspects that allow it to be suited to the navigation solution.

Brown (1992) identifies the navigation problem as a suitable application for the Kalman filter since:

1. Vehicle state dynamics are often linear or can be linearised without significant fidelity loss.
2. Several navigation sources provide redundant measurement information that must be processed optimally and with maximum performance such that multiple quantities of interest (e.g. position, velocity and attitude) may be estimated.
3. Measurements must be processed on line to obtain real time navigation state estimates. Such processing should be as efficient as possible to ensure that real time deadlines are achieved.

In terms of the proposed GPS/INS integration (Section 4.2), Grewal and Andrews (2001) also note that the aiding of INS with GPS can be fitted to a Kalman filter as follows:

1. GPS and IMU outputs can be identified as noisy sensors.
2. Variables to be estimated (or system state) could include position, velocity, acceleration, attitude and angular velocity. Additional variables may be added to the estimation solution to account for time dependent sensor parameters (e.g. scale factors and biases) and to model correlated noises (which would, when unaccounted for, violate the white noise assumption mentioned in Section 4.4.1).
3. The system dynamic equations are possibly incomplete (unmodelled modes), can be affected by unpredictable disturbances, may include sensors whose parameters vary erratically, and are driven by corrupted inputs (all of which would contribute to uncertain state dynamics).

The Kalman filter, in this setting, often produces estimates of vehicle state (position, velocity and attitude) using statistical models to compute the blending between the current GPS measurement and the vehicle state as predicted by the uncertain vehicle state dynamics (which incorporates all past information) driven by the noisy IMU. Using statistical models, the Kalman filter is able to exploit the differing and complementary error behaviour (Section 4.2) of the GPS versus the IMU, so as to reduce the undesirable features of both. Additionally, the statistical models also address the challenge of including redundant measurement information (a common scenario in GPS/INS integration) in an optimal way.

4.4.3 Filter structure

As mentioned in Section 4.4.1, the Kalman filter is characterised by a set of recursive equations that estimate the instantaneous state of a dynamic system by minimising the estimation variance. The filter equation discussion, below, follows from Brown and Hwang (1992).

Firstly, consider the following mathematical description of a linear dynamic system to be estimated. The Kalman filter addresses the challenge of determining the state $\mathbf{x_k}$ ϵ R^n of the system. Assume that the stochastic process to be estimated can be modelled discretely as:

$$\mathbf{x_{k+1}} = \mathbf{A_k x_k} + \mathbf{w_k} \qquad 4.18$$

And assume that observations of the process, $\mathbf{z_k}$ ϵ R^m, are generated in accordance with:

$$\mathbf{z_k} = \mathbf{H_k x_k} + \mathbf{v_k} \tag{4.19}$$

Table 4.1 provides descriptions of the terms used in Equations 4.18 and 4.19.

Table 4.1. Dynamic system symbol definitions.

Variable/Symbol	Dimension	Description
$\bullet_k$	∈ Z	Sub script k indicates variable as at time t_k.
$\mathbf{A_k}$	∈ $R^{n \times n}$	Matrix defining transition of $\mathbf{x_k}$ from time t_k to t_{k+1} i.e. $\mathbf{x_k}$ to $\mathbf{x_{k+1}}$ in the absence of any forcing function.
$\mathbf{w_k}$	∈ $R^{n \times 1}$	Vector of white noises with known covariance matrix $\mathbf{Q_k}$.
$\mathbf{H_k}$	∈ $R^{m \times n}$	Matrix defining noiseless relationship between measurements, $\mathbf{z_k}$, and states, $\mathbf{x_k}$.
$\mathbf{v_k}$	∈ $R^{m \times 1}$	Vector of white noises defining observation errors with know covariance ($\mathbf{R_k}$) and zero cross correlation with $\mathbf{w_k}$.

Further characterisation of $\mathbf{w_k}$ and $\mathbf{v_k}$ is given by the respective covariance matrices below:

$$E[\mathbf{w_k w_i}^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \tag{4.20}$$

$$E[\mathbf{w_k w_i}^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \tag{4.21}$$

$$E[\mathbf{w_k v_i}^T] = 0, \quad \forall\, i,k \tag{4.22}$$

To begin the recursive computation of the Kalman filter, an initial estimate of the process incorporating all knowledge up to the starting point is required. This a priori estimate will be denoted $\hat{\mathbf{x}}_\mathbf{k}^-$ where the "^" denotes estimate and the "-" implies a priori estimate i.e. This is the best estimate without including the information provided by the measurement at t_k. Additionally, it is assumed that $\hat{\mathbf{x}}_\mathbf{k}^-$ has known error covariance matrix denoted by:

$$\mathbf{P_k^-} = E\,[\mathbf{e_k^- e_k^-}^T] = E\,[(\mathbf{x_k} - \hat{\mathbf{x}}_\mathbf{k}^-)(\mathbf{x_k} - \hat{\mathbf{x}}_\mathbf{k}^-)^T] \tag{4.23}$$

In Equation 4.23, the estimation error has been implicitly defined by Equation 4.24 having zero mean.

$$\mathbf{e_k^-} = (\mathbf{x_k} - \hat{\mathbf{x}}_\mathbf{k}^-) \tag{4.24}$$

With the a priori assumptions above, the measurement, $\mathbf{z_k}$, can now be assimilated into an improved estimate for the state using Equation 4.25.

$$\hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}\,(\mathbf{z_k} - \mathbf{H_k}\hat{\mathbf{x}}_\mathbf{k}^-) \tag{4.25}$$

In Equation 4.25, a linear blending of the noisy measurement and the a priori estimate via weighting factor $\mathbf{K_k}$ resulted in an optimal updated estimate, $\hat{\mathbf{x}}_\mathbf{k}$. Equation 4.25 is developed in Brown (1992) (Section 5.6). The optimality of the updated estimate lies solely in the choice of $\mathbf{K_k}$

which should be calculated to minimise the mean square estimation error. To arrive at an expression for $\mathbf{K_k}$, it is necessary to determine the error covariance associated with the updated estimate:

$$\mathbf{P_k} = E\,[\mathbf{e_k}\mathbf{e_k}^T] = E\,[(\mathbf{x_k} - \mathbf{\hat{x}_k})(\mathbf{x_k} - \mathbf{\hat{x}_k})^T] \qquad 4.26$$

Substituting Equation 4.19 into Equation 4.25, and then using the resulting expression for $\hat{x}_k$ in Equation 4.26 gives:

$$\begin{aligned}\mathbf{P_k} = E\,\Big\{&\big[(\mathbf{x_k} - \mathbf{\hat{x}_k^-}) - \mathbf{K_k}\,(\mathbf{H_k v_k} + \mathbf{v_k} - \mathbf{H_k \hat{x}_k^-})\big] \times \ldots \\ &\ldots \big[(\mathbf{x_k} - \mathbf{\hat{x}_k^-}) - \mathbf{K_k}\,(\mathbf{H_k v_k} + \mathbf{v_k} - \mathbf{H_k \hat{x}_k^-})\big]^T\Big\}\end{aligned} \qquad 4.27$$

Equation 4.27 can be simplified to Equation 4.28, which represents a general expression for the updated estimate error covariance, by completing the expectation using the fact that the a priori estimation error, $(\mathbf{x_k} - \mathbf{\hat{x}_k^-})$, is uncorrelated to $\mathbf{v_k}$.

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k H_k})\mathbf{P_k^-}(\mathbf{I} - \mathbf{K_k H_k})^T + \mathbf{K_k R_k K_k}^T \qquad 4.28$$

The optimisation problem can now be described in terms of Equation 4.28. It essentially involves minimising the individual terms along the major diagonal of $\mathbf{P_k}$ (these terms are the estimation error variances for the elements of the state vector being estimated). Analytically, this is achieved with differential calculus by differentiating the trace of $\mathbf{P_k}$ with respect to $\mathbf{K_k}$ (Equation 4.29).

$$\begin{aligned}&\frac{\partial\,(tr\,\mathbf{P_k})}{\partial \mathbf{K_k}} = 0 \\ &\Rightarrow \mathbf{K_k} = \mathbf{P_k^- H_k}^T(\mathbf{H_k P_k^- H_k}^T + \mathbf{R_k})^{-1}\end{aligned} \qquad 4.29$$

The trace of $\mathbf{P_k}$ is used since it can be argued that minimisation of the error sum implies that individual errors are also minimised. When Equation 4.29 is used, the blending factor $\mathbf{K_k}$ is referred to as the Kalman gain and the covariance matrix associated with the optimal estimate is calculated recursively via:

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k H_k})\mathbf{P_k^-} \qquad 4.30$$

A method for including the information embedded in $\mathbf{z_k}$ now exists via Equation 4.25 and the Kalman gain. However, for the relevant expressions to be solved, $\mathbf{\hat{x}_k^-}$ and $\mathbf{\hat{P}_k^-}$ were needed and, due to the recursive nature of the algorithm, a similar requirement exists to assimilate the data provided by measurement $\mathbf{z_{k+1}}$ i.e. for state estimate prediction at t_{k+1}, $\mathbf{\hat{x}_{k+1}^-}$ is needed as is its associated covariance matrix, $\mathbf{P_{k+1}^-}$. The a priori state estimate prediction is found by projecting the current updated estimate through the state transition matrix in the absence of noise (which is valid as noises here were considered to be zero mean and additive).

$$\mathbf{\hat{x}_{k+1}^-} = \mathbf{A_k \hat{x}_k} \qquad 4.31$$

In finding $\mathbf{P_{k+1}^-}$, first form the expression for the a priori error:

$$\begin{aligned}\mathbf{e}_{\mathbf{k+1}}^{-} &= \mathbf{x}_{\mathbf{k+1}} - \hat{\mathbf{x}}_{\mathbf{k+1}}^{-} \\ &= (\mathbf{A}_{\mathbf{k}}\mathbf{x}_{\mathbf{k}} + \mathbf{w}_{\mathbf{k}}) - \mathbf{A}_{\mathbf{k}}\hat{\mathbf{x}}_{\mathbf{k}} \\ &= \mathbf{A}_{\mathbf{k}}\mathbf{e}_{\mathbf{k}} + \mathbf{w}_{\mathbf{k}}\end{aligned} \qquad 4.32$$

From Equation 4.32,

$$\mathbf{P}_{\mathbf{k+1}}^{-} = E[\mathbf{e}_{\mathbf{k+1}}^{-}\left(\mathbf{e}_{\mathbf{k+1}}^{-}\right)^{T}] = \mathbf{A}_{\mathbf{k}}\mathbf{P}_{\mathbf{k}}\mathbf{A}_{\mathbf{k}}^{\ T} + \mathbf{Q}_{\mathbf{k}} \qquad 4.33$$

With Equations 4.31 and 4.33, the required a priori quantities are available, and $\mathbf{z}_{\mathbf{k+1}}$ can now be used as per Equation 4.25.

Grewal and Andrews (2001) describes the relationship between the discrete time system and the Kalman filter by Figure 4.15.

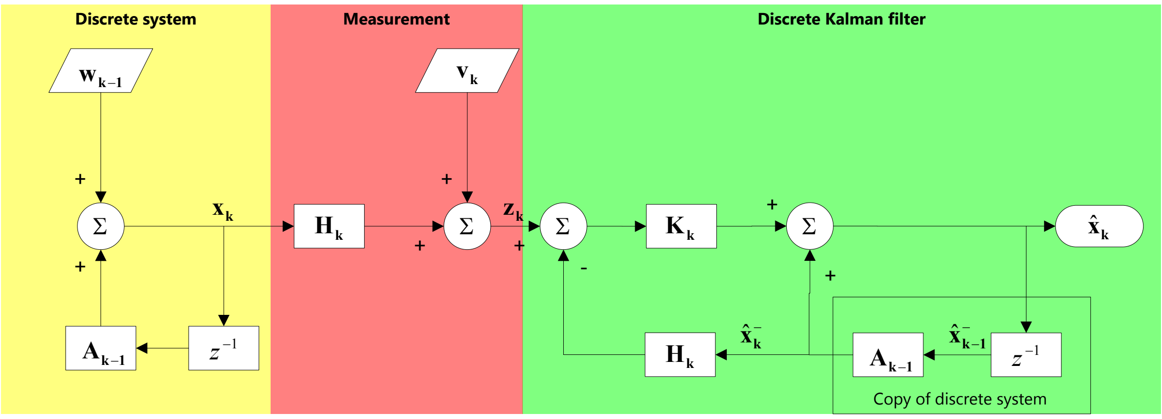

Discrete system
Measurement
Discrete Kalman filter
$\mathbf{w}_{\mathbf{k-1}}$
$\mathbf{v}_{\mathbf{k}}$
+
+
Σ
$\mathbf{x}_{\mathbf{k}}$
$\mathbf{H}_{\mathbf{k}}$
+
Σ
+
$\mathbf{z}_{\mathbf{k}}$
+
Σ
-
$\mathbf{K}_{\mathbf{k}}$
+
Σ
+
$\hat{\mathbf{x}}_{\mathbf{k}}$
+
$\mathbf{A}_{\mathbf{k-1}}$
z^{-1}
$\mathbf{H}_{\mathbf{k}}$
$\hat{\mathbf{x}}_{\mathbf{k}}^{-}$
$\mathbf{A}_{\mathbf{k-1}}$
$\hat{\mathbf{x}}_{\mathbf{k-1}}^{-}$
z^{-1}
Copy of discrete system

Figure 4.15. Conceptual relationship between discrete time system and Kalman filter.

Figure 4.16 further summarises the Kalman filter recursive equations with respect to the computational flow presented in Figure 4.14.

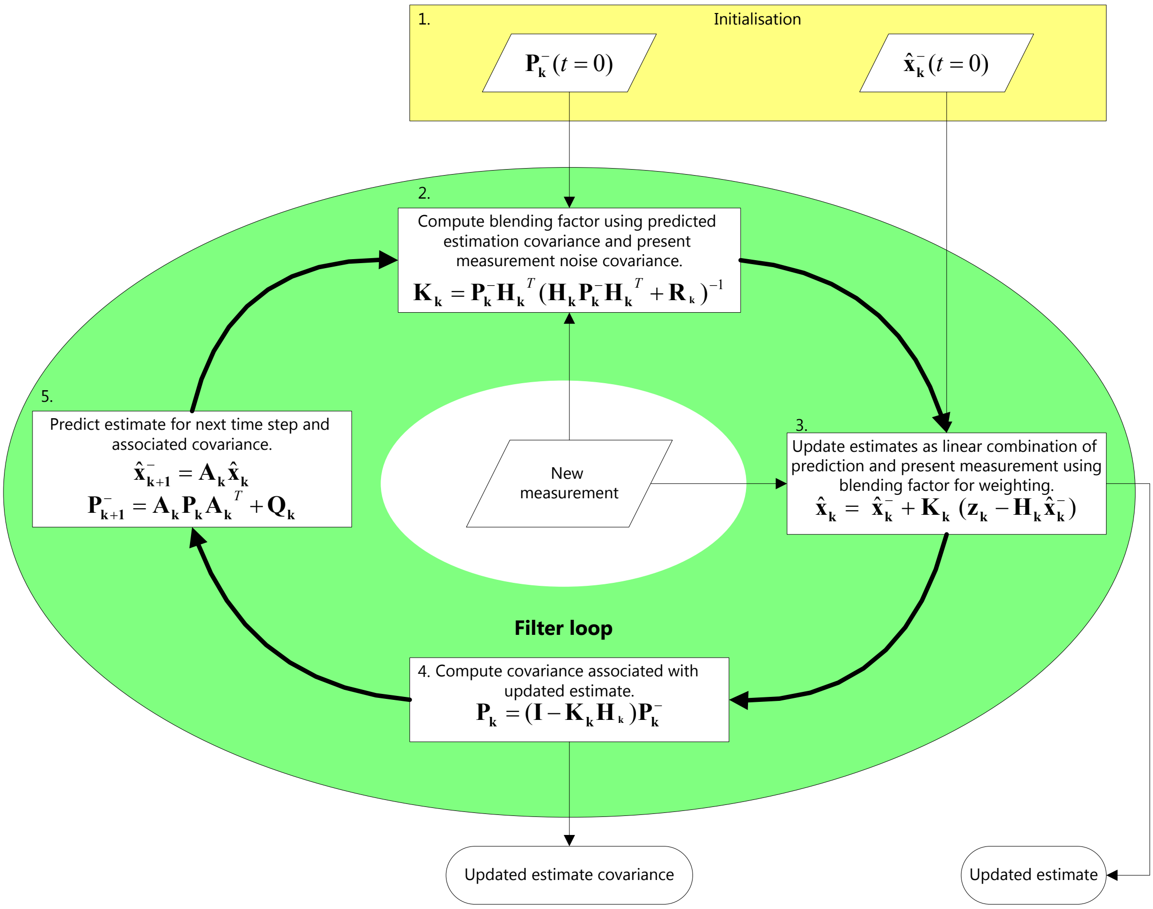

1.
Initialisation
$\mathbf{P}_k^-(t=0)$
$\hat{\mathbf{x}}_k^-(t=0)$
2.
Compute blending factor using predicted estimation covariance and present measurement noise covariance.
$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
5.
Predict estimate for next time step and associated covariance.
$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k\hat{\mathbf{x}}_k$
$\mathbf{P}_{k+1}^- = \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^T + \mathbf{Q}_k$
New measurement
3.
Update estimates as linear combination of prediction and present measurement using blending factor for weighting.
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$
Filter loop
4. Compute covariance associated with updated estimate.
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-$
Updated estimate covariance
Updated estimate

Figure 4.16. Kalman filter recursive equations (revised computational flow in terms of Figure 4.14).

Focussing on the filter operation between two successive time steps of the infinite loop of Figure 4.16, the transition diagram below (Figure 4.17) can be developed (Grewal and Andrews, 2001). Here:

1. Transitions between quantities are indicated by arrows.
2. Parentheses adjacent to arrows indicate variables effecting transition defined by the arrow.

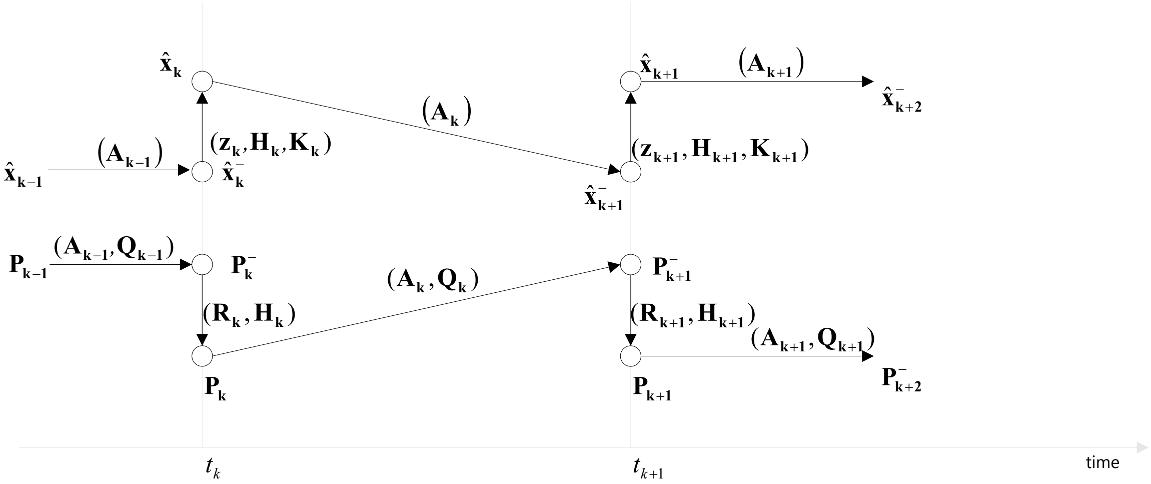

$\hat{\mathbf{x}}_k$
$(\mathbf{A}_k)$
$\hat{\mathbf{x}}_{k+1}$
$(\mathbf{A}_{k+1})$
$\hat{\mathbf{x}}_{k+2}^-$
$\hat{\mathbf{x}}_{k-1}$
$(\mathbf{A}_{k-1})$
$(\mathbf{z}_k, \mathbf{H}_k, \mathbf{K}_k)$
$\hat{\mathbf{x}}_k^-$
$(\mathbf{z}_{k+1}, \mathbf{H}_{k+1}, \mathbf{K}_{k+1})$
$\hat{\mathbf{x}}_{k+1}^-$
$\mathbf{P}_{k-1}$
$(\mathbf{A}_{k-1}, \mathbf{Q}_{k-1})$
$\mathbf{P}_k^-$
$(\mathbf{A}_k, \mathbf{Q}_k)$
$\mathbf{P}_{k+1}^-$
$(\mathbf{R}_k, \mathbf{H}_k)$
$(\mathbf{R}_{k+1}, \mathbf{H}_{k+1})$
$(\mathbf{A}_{k+1}, \mathbf{Q}_{k+1})$
$\mathbf{P}_k$
$\mathbf{P}_{k+1}$
$\mathbf{P}_{k+2}^-$
t_k
t_{k+1}
time

Figure 4.17. Filter variable transition from step k to k + 1.

It should be apparent that each variable to be estimated assumes two distinct values during the discrete epoch – an a priori value before any measurement information is incorporated and an updated value after the measurement is assimilated.

4.4.4 Non linear implementation

The Kalman filter algorithm assumes both linear process and measurement models. However, in many applications and especially in the integration of GPS and IMU, linearity is uncommon. The common engineering approach of linearising the problem may then find applicability to solve non linear estimation problems with the linear Kalman filter recursive solution.

With Kalman filtering, the linearization can take two forms:

1. The linearised Kalman filter, where the linearisation is performed about some predefined nominal trajectory in state space independent of measurement data.
2. The extended Kalman filter, where the linearisation is performed about a trajectory that is continually updated with estimates influenced by measurements.

The non linear Kalman filtering approach, presented in the following, is from Brown and Hwang (1992), but has been presented in discrete time for consistency with the linear filter discussion earlier. As the extended Kalman filter is an adaptation of the linearised Kalman filter, it is apt to begin with the linearised Kalman filter.

4.4.4.1 Linearised Kalman filter

The process to be estimated is now given by the mathematical description of Equations 4.34 and 4.35.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \qquad 4.34$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \qquad 4.35$$

Where:

1. $\mathbf{f}_k$ and $\mathbf{h}_k$ are now functions representing the noiseless state and measurement dynamics.
2. $\mathbf{u}_k$ is a known driving function.
3. $\mathbf{v}_k$ and $\mathbf{w}_k$ are white noise processes with zero cross correlation.
4. Non linearity has entered the system (as may be possible) in both the state and measurement equations.
5. The restrictive form of Equations 4.34 and 4.35, where $\mathbf{v}_k$ and $\mathbf{w}_k$ are separate additive terms, (and not independent variables of $\mathbf{f}_k$ and $\mathbf{h}_k$) has been chosen to simplify the discussion.

Now assume that a reference trajectory $\mathbf{x}^*_k$ exists in state space. From an illustrative plot of the reference and actual trajectories (Figure 4.18), it should be possible to express the actual trajectory as:

$$\mathbf{x}_k = \mathbf{x}^*_k + \Delta\mathbf{x}_k \qquad 4.36$$

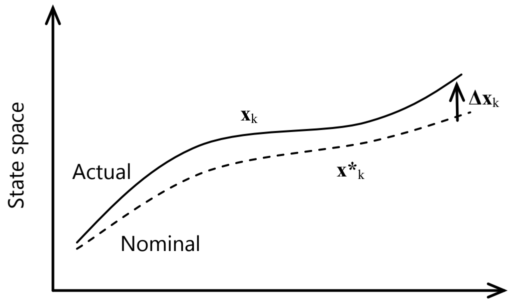


Figure 4.18. Representative view of actual versus nominal trajectory for linearised Kalman filter (after Brown and Hwang, 1992).

Substituting Equation 4.36 into Equations 4.34 and 4.35:

$$\mathbf{x}^*_{k+1} + \Delta\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}^*_k + \Delta\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \qquad 4.37$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}^*_k + \Delta\mathbf{x}_k) + \mathbf{v}_k \qquad 4.38$$

If $\Delta\mathbf{x}_k$ is small (as is to be expected if the nominal and actual trajectories are reasonably close), the state dynamic and measurement functions may be approximated by a Taylor series expansion (retaining only first order terms) as in Equations 4.39 and 4.40.

$$\mathbf{x}^*_{k+1} + \Delta\mathbf{x}_{k+1} \approx \mathbf{f}(\mathbf{x}^*_k, \mathbf{u}_k) + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} \cdot \Delta\mathbf{x}_k + \mathbf{w}_k \qquad 4.39$$

$$\mathbf{z}_k \approx \mathbf{h}(\mathbf{x}^*_k) + \left[\frac{\partial\mathbf{h}}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} \cdot \Delta\mathbf{x}_k + \mathbf{v}_k \qquad 4.40$$

Where:

$$\left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}\right|_{\mathbf{x}=\mathbf{x}^*_k} = \mathbf{A}_k \qquad 4.41$$

$$\left[\frac{\partial\mathbf{h}}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} = \left.\begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial h_2} & \cdots \\ \frac{\partial h_2}{\partial x_1} & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}\right|_{\mathbf{x}=\mathbf{x}^*_k} = \mathbf{H}_k \qquad 4.42$$

Usually the nominal trajectory is selected to satisfy:

$$\mathbf{x}^*_{k+1} = \mathbf{f}(\mathbf{x}^*_k, \mathbf{u}_k) \qquad 4.43$$

Using this result in Equation 4.39, and manipulating Equation 4.40 slightly, yields the linearised model of Equations 4.44 and 4.45.

$$\text{Linearised state dynamics: } \mathbf{\Delta x}_{k+1} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} \cdot \mathbf{\Delta x}_k + \mathbf{w}_k \qquad 4.44$$

$$\text{Linearised measurement model: } \mathbf{z}_k - \mathbf{h}(\mathbf{x}^*_k) = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*_k} \cdot \mathbf{\Delta x}_k + \mathbf{v}_k \qquad 4.45$$

From Equations 4.44 and 4.45:

1. The measurement (left hand side of Equation 4.45) is the actual measurement less the measurement predicted by evaluation of the non linear measurement dynamics (in the absence of noise) along the nominal trajectory.
2. The linear state dynamic and measurement dynamic matrices are evaluated along the nominal trajectory.

4.4.4.2 Extended Kalman filter

The extended Kalman filter differs from the linearised Kalman filter only in the determination of the nominal trajectory. Unlike the linearised Kalman filter that relies on a pre computed reference trajectory, the extended Kalman filter linearises about the filter's estimated trajectory.

Practically, this involves evaluating the partial derivatives of Equations 4.41 and 4.42 along a trajectory that is continually updated with the latest estimates. This implies that the nominal trajectory is ultimately a function of the measurement sequence. Naturally, this method is only applicable if the relevant process and measurement dynamics are differentiable, and, it should be appreciated that validating the stability of this implementation may be difficult.

4.4.4.3 Obtaining total quantities

In the linearised Kalman filter incremental quantities are the state variables. With the extended Kalman filter it may be more suitable to keep track of total estimates. Brown and Hwang (1992) suggest the following method to achieve this (and its justification).

The linearised measurement relation, Equation 4.45, results in the Kalman filter using $[\mathbf{z}_k - \mathbf{h}(\mathbf{x}^*_k)]$ as a measurement. Given the incremental update equation (Equation 4.46 below), if the measurement residual is formed by associating the underlined terms together, then the update equation can be written as Equation 4.47.

$$\mathbf{\Delta\hat{x}}_k = \mathbf{\Delta\hat{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \underline{\underline{\mathbf{h}(\mathbf{x}^*_k)}} - \underline{\underline{\mathbf{H}_k\mathbf{\Delta\hat{x}}_k^-}}] \qquad 4.46$$

$$\mathbf{\Delta\hat{x}}_k = \mathbf{\Delta\hat{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{\hat{z}}_k] \qquad 4.47$$

Equation 4.47 follows since an estimate of the measurement can be formed by:

$$\mathbf{\hat{z}}_k^- = \mathbf{h}(\mathbf{x}^*_k) + \mathbf{H}_k\mathbf{\Delta\hat{x}}_k^- \qquad 4.48$$

The measurement residual as per Equation 4.47 is the noisy measurement less the predicted measurement based on the corrected trajectory. Adding the nominal trajectory to both sides of Equation 4.47 gives Equation 4.49 which is the linear update equation in terms of total quantities.

$$\mathbf{x}^*_k + \Delta\hat{\mathbf{x}}_k = \mathbf{x}^*_k + \Delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k] \qquad 4.49$$

$$\hat{\mathbf{x}}_k \quad = \quad \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k] \qquad 4.50$$

Equation 4.50 operates in the usual Kalman filter paradigm – the updated estimate is computed from the sum of an a priori estimate with the appropriately weighted measurement residual. The a priori estimate must naturally be generated for the next step and this is done via the non linear dynamics:

$$\hat{\mathbf{x}}_{k+1}^- = \begin{cases} \textit{Solution of non linear differential equation}: \\ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_\mathbf{d}, t) \ \textit{at}\ t = t_{k+1} \\ \textit{Subject to initial condition}: \mathbf{x} = \hat{\mathbf{x}}_k \ \textit{at}\ t = t_k \end{cases} \qquad 4.51$$

Generation of $\hat{\mathbf{x}}_{k+1}{}^-$ by Equation 4.51 implies that the predictive measurement, $\hat{\mathbf{z}}_{k+1}{}^-$, can be provided through the measurement dynamics ($\mathbf{h}(\hat{\mathbf{x}}_{k+1}{}^-)$), and, with the measurement at t_{k+1}, the subsequent measurement residual is available. As such the filter can iterate. Equations 4.30 and 4.33 still hold for $\mathbf{P}_k$ and $\mathbf{P}_{k+1}{}^-$, but it should be appreciated that the $\mathbf{A}_k$, $\mathbf{H}_k$ and $\mathbf{Q}_k$ result now from the linearised model.

4.4.4.4 Linearised versus extended Kalman filtering

As noted by Grewal and Andrews (2001), the extended Kalman filter addresses a fundamental problem with the linearised Kalman filter in that the deviation of the pre computed nominal trajectory from the actual trajectory may become large enough to invalidate Equations 4.39 and 4.40 in the absence of Taylor series terms beyond first order. If, as in the extended Kalman filter, the nominal trajectory is updated with the estimated trajectory, this deviation is held sufficiently small that higher order Taylor terms do not corrupt the linearisation approximation. Conversely, Brown and Hwang (1992) indicate that the premise of using an updated nominal trajectory is not always favourable. As the updated nominal trajectory is only superior in a statistical sense, there is a chance that the updated trajectory is worse than the nominal one. This will lead to poorer estimates, causing further errors in the nominal trajectory, resulting ultimately in divergence of the filter. The extended Kalman filter is considered slightly more risky than the linearised Kalman filter, especially with large initial uncertainty and measurement errors.

A further disadvantage of the extended Kalman filter is cited by Grewal and Andrews (2001) – it requires increased computational power as several parameters including $\mathbf{A}_k$, $\mathbf{H}_k$ and $\mathbf{Q}_k$ must be calculated in real time as these depend on the estimates (an unpredictable nominal trajectory). Consequently, both $\mathbf{P}_k{}^-$ and $\mathbf{K}_k$ must also be computed in real time. With the linearised Kalman filter, the above parameters may be pre computed off line. As is evident from the ensuing discussion, preference of the linearised Kalman filter over the extended Kalman filter (or vice versa) is difficult to justify. The specific application often determines which method is more suitable.

4.5 Navigation filter design

4.5.1 Anatomy of the fusion Kalman filter

The Kalman filter implemented in the navigation system can now be characterised in terms of several parameters, some discussed earlier:

1. Feedback or feed forward configuration (Section 4.2.1).
2. Coupling approaches (Section 4.2.2).
3. Gimballed versus strap down mechanisation (Section 4.3).
4. Kalman filter states.
5. Kalman filter measurements.

The feedback configuration with loose coupling was used. The feedback method was suitable as a low accuracy IMU was used, and, due to the fact that the high dynamic fidelity of the feed forward configuration was unnecessary. Loose coupling was ideal as the GPS receiver provided only position and velocity outputs leading to integration on the position and velocity level only being possible. Additionally, even in the absence of the above limitation, the simplicity of loose coupling rendered it superior in that the time saved in implementing a loose coupling solution far out weighed the potential accuracy gain of the tight coupling approach. As would be expected from the type of IMU employed, a strap down system was used with the DCM of Section 4.3.2.2.1 providing the relevant transformation. Quaternions, which are an extension of complex numbers comprising a vector and scalar part, can also be used for this purpose. These complex numbers, having form $q_0 + q_1 i + q_2 j + q_3 k$, are used in several applications to compactly and completely represent rotations. The $\{i \quad j \quad k\}$ part specifies the vector about which rotation by angle q_0 occurs (Salychev, 1998). In many applications, quarternions are preferred due to the reduced computational effort and lack of pathology at pitch angle multiples of 90° (Merhav, 1996). With the helicopter pitched at ±90°, these so called gimbal lock singularities (a term from the era of mechanicised inertial platforms) manifest themselves and the Euler derivatives for pitch and yaw cannot be found (Equation 4.11 with θ=±90°). Essentailly, with nose up or nose down attitudes, roll and heading cannot be disambiguated from each other resulting in a loss of one degree of freedom of rotation (Merhav, 1996). However, here the DCM method is employed as; (1) the DSP has sufficient computational bandwidth; (2) the flight platform is envisioned for benign manoeuvres with shallow trajectories having pitch and roll angles bounded by ±45°; and, (3) the DCM method is more rapidly and easily implemented. Regardless of the technique used to maintain rotations, it should be noted that the strap down INS is the modern approach (especially with aiding) as a result of lower cost, size and power consumption.

With respect to the Kalman filter, Section 4.5.2 will show the non linear nature of the navigation system necessitating a linearised or extended Kalman filter. Also, in terms of the state configuration, Grewal and Andrews (2001) indicate that the choice of filter state variables forms a critical part of the Kalman filter design. Usually, nine basic states are included in any aided INS mechanisation and these have been included here (Grewal, Andrews and Weill, 2001):

1. The navigation solution comprising triplets of position and velocity in the navigation frame.
2. The craft orientation solution i.e. The Euler angle triplet of roll, pitch and yaw.

Additional states are selected considering the IMU sensor biases. Typically, major bias sources are augmented to the state vector and other ancillary errors are accounted for by the acceleration and angular velocity noise terms. Further, it is computationally impractical to implement very high fidelity models incorporating many error states. For the type of low cost IMU used, acceleration and angular velocity sensor biases were appended to the state vector as these tend to dominate the errors of such IMUs. This resulted in the fifteen element state vector presented in Equation 4.52.

$$\mathbf{x} = \left[r_N \; r_E \; r_D \; v_N \; v_E \; v_D \; \varphi \, \theta \, \psi \; b_x \; b_y \; b_z \; b_p \; b_q \; b_r \right]^T \qquad 4.52$$

Where:

1. r_N, r_E and r_D are the positions in the north, east and down directions.
2. v_N, v_E and v_D are the velocities in the north, east and down directions.
3. φ, θ and ψ represent the Euler angles of roll, pitch and yaw.
4. b_x, b_y and b_z are the accelerometer biases in the body x, y and z directions.
5. b_p, b_q and b_r are the rate gyro biases in the body p, q and r directions.

The Kalman filter estimating these states relies on measurements provided mostly by the GPS receiver. Six GPS measurements are used by the filter:

1. North, east and down positions.
2. North, east and down velocities.

GPS position readings are quoted in longitude, latitude and altitude with velocities given in the north, east and up frame. Conversions given in Section 4.3.2.2.3 result in the GPS receiver effectively providing positions in the north, east and down system. A barometer is also included, the output from which is converted to altitude and pre combined with the GPS vertical position, thereby providing a fused down position and covariance (Section 4.5.2.2.1). No direct measurement of attitude is available. The resulting measurement vector is given by Equation 4.53.

$$\mathbf{z} = \left[r_{N(GPS)} \; r_{E(GPS)} \; r_{D(MVE)} \; v_{N(GPS)} \; v_{E(GPS)} \; v_{D(GPS)} \right]^T \qquad 4.53$$

Where:

1. $r_{N(GPS)}$ and $r_{E(GPS)}$ are the GPS position readings expressed in the north and east directions.
2. $r_{D(MVE)}$ is the output of a minimum variance estimator combining GPS down position with a down position reading from a barometric altimeter.
3. $v_{N(GPS)}$, $v_{E(GPS)}$ and $v_{D(GPS)}$ are the GPS velocities in the north, east, down system.

4.5.2 Development of the Kalman filter process and measurement models

4.5.2.1 Navigation state mechanisation equations

The navigation state mechanisation equations are differential equations that relate the IMU measurements in the body frame to the desired navigation quantities in the navigation frame.

Farell and Barth (1999) propose, for position and velocity dynamics, state equations involving the basic kinematic relationships between position, velocity and acceleration (Equations 4.54 and 4.55 respectively):

$$\begin{bmatrix} \dfrac{d(latitude)}{dt} \\ \dfrac{d(longitude)}{dt} \\ \dfrac{d(altitude)}{dt} \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{1}{R+h} & 0 & 0 \\ 0 & \dfrac{1}{(R+h)\cos(lat)} & 0 \\ 0 & 0 & -1 \end{bmatrix}}_{A} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \quad 4.54$$

$$\dot{\mathbf{v}}_{\mathbf{e}}^{\mathbf{n}} = \mathbf{a}^{\mathbf{n}} + \mathbf{g}^{\mathbf{n}} + (\mathbf{\Omega}_{\mathbf{ne}} - 2\mathbf{\Omega}_{\mathbf{ie}}^{\mathbf{n}})\mathbf{v}_{\mathbf{e}}^{\mathbf{n}} \quad 4.55$$

In Equation 4.54 factors in the main diagonal of A facilitate conversion from a locally level tangent frame (v_N, v_E and v_D) to latitude, longitude and altitude representations of position. This is achieved by using the common radian measure, angle to arc length conversion where the arc radii facilitate the conversion. In Equation 4.54 the denominators of the main diagonal elements in A represent the radii (in metres) of the arcs spanned by latitude and longitude changes (in radians). These radii include the altitude of the craft. The velocities (v_N, v_E and v_D) are in metres per second. It should be appreciated that these equations are based on a spherical Earth model with R being the radius of the Earth. The $\cos(lat)$ term in the longitude equation is used to account for the shortening of the cross sectional East-West radius conversion term at latitudes away from the Equator. Figure 4.19 illustrates the geometry of the conversions.

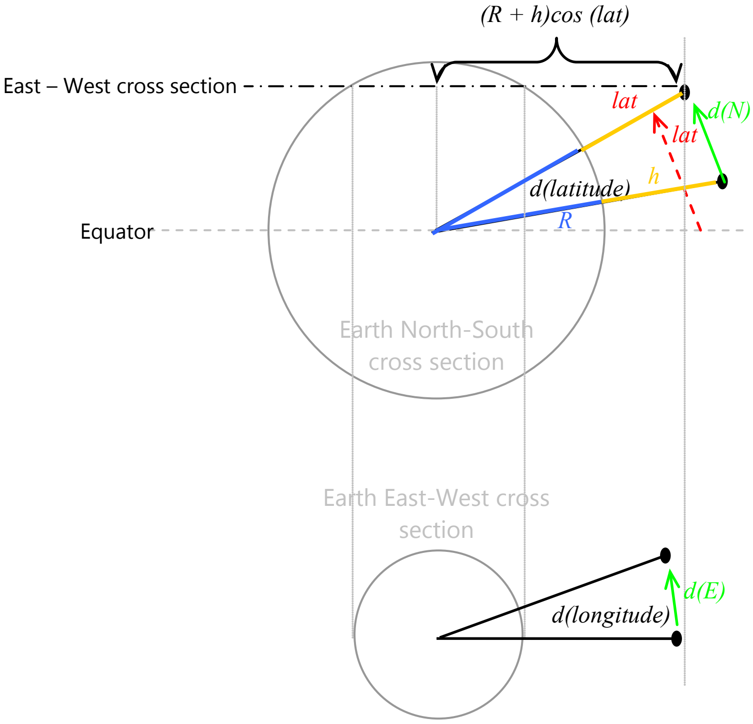

(R + h)cos (lat)
East – West cross section
lat
d(N)
lat
h
d(latitude)
R
Equator
Earth North-South cross section
Earth East-West cross section
d(E)
d(longitude)

Figure 4.19. Illustration of latitude/longitude change to North/East distance change conversion.

For this project, position solutions are required in the navigation frame, rendering the conversions of Equation 4.54 unnecessary. The subsequently simplified position dynamics are:

$$\begin{bmatrix} \dot{r}_N \\ \dot{r}_E \\ \dot{r}_D \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \tag{4.56}$$

The velocity dynamics quoted in Equation 4.55 relate the derivative of velocity in the navigation frame to:

1. Acceleration in the navigation frame: $\mathbf{a^n}$ and $\mathbf{g^n}$ (acceleration due to gravity).
2. Transport rate of the navigation frame relative to the Earth frame i.e. The skew symmetric matrix $\mathbf{\Omega_{ne}}$ representing the relative rotation of the navigation frame with respect to the Earth frame.
3. Inertial rotation rate of the Earth, given by the skew symmetric matrix $\mathbf{\Omega_{ie}}$.

In terms of Point (2), $\mathbf{\Omega_{ne}} = \mathbf{0}$ since the navigation frame in use is a tangent frame with fixed origin. Furthermore, with respect to (3) above, $\mathbf{\Omega_{ie}}$ is of the order of 10^{-5} rad/s and can be neglected in this application due to the short mission times and limited travel range of the helicopter. Additionally, the rate gyros used can only detect about 10^{-4} rad/s and, thus, the Earth rate is lost in gyro noise. This leads to the simplified velocity dynamics of Equation 4.57 which relates total acceleration in the navigation frame with velocity in the navigation frame.

$$\begin{bmatrix} \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \end{bmatrix} = \begin{bmatrix} a_N + g_N \\ a_E + g_E \\ a_D + g_D \end{bmatrix} \tag{4.57}$$

Since the navigation frame is defined as per Section 4.3.2.1.2, $g_N = g_E = 0$ and $g_D \approx 9.81\, ms^{-2}$. a_N, a_E and a_D however, represent acceleration of the helicopter expressed in the navigation frame. The IMU provides accelerations in the helicopter body coordinates, but, application of the DCM transformation from body to navigation frame (Section 4.3.2.2.1) will result in accelerations being expressed as desired. If the IMU body accelerations are a_x, a_y and a_z, the resulting velocity dynamic equations, with Euler angles φ (roll), θ (pitch) and ψ (yaw), are:

$$\begin{bmatrix} \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \end{bmatrix} = \begin{bmatrix} a_x \cos\theta\cos\psi + a_y(\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) + a_z(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \\ a_x \cos\theta\sin\psi + a_y(\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) + a_z(\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) \\ -a_x \sin\theta + a_y \sin\varphi\cos\theta + a_z \cos\varphi\cos\theta - g \end{bmatrix} \tag{4.58}$$

In terms of attitude dynamics i.e. state differential equations describing the evolution of the Euler angles, Salychev (1998) states that to find the DCM between body and navigation coordinates, Equation 4.59 must be solved for $\mathbf{\Gamma^{-1}}$.

$$\dot{\mathbf{\Gamma}}^{-1} = \mathbf{\Gamma}^{-1}\breve{\boldsymbol{\omega}}_\mathbf{b} - \breve{\boldsymbol{\omega}}_\mathbf{n}\mathbf{\Gamma}^{-1} \tag{4.59}$$

Where:

1. $\breve{\bullet}$, is the skew symmetric form of vector $\bullet = [\bullet_x \;\; \bullet_y \;\; \bullet_z]$, given by:

$$\breve{\bullet} = \begin{bmatrix} 0 & -\bullet_z & \bullet_y \\ \bullet_z & 0 & -\bullet_x \\ -\bullet_y & \bullet_x & 0 \end{bmatrix}$$

2. $\mathbf{\Gamma}$ is the DCM transforming vectors from the navigation frame to the body frame as defined in Section 4.3.2.2.1, with $\mathbf{\Gamma}^{-1}$ facilitating the reverse transformation.

Equation 4.59 can be simplified to Equation 4.60 when considering a fixed tangent plane navigation frame i.e. $\boldsymbol{\omega}_\mathbf{n} = \mathbf{0}$.

$$\dot{\mathbf{\Gamma}}^{-1} = \mathbf{\Gamma}^{-1}\breve{\boldsymbol{\omega}}_\mathbf{b} \qquad 4.60$$

Equation 4.60 illustrates the evolution of the DCM and not the Euler angles as desired. Padfield, however, provides relationships between fuselage angular velocities and Euler derivatives (Section 4.3.2.2.2) giving expressions for $d\varphi/dt$, $d\theta/dt$ and $d\psi/dt$. It can be shown that these expressions are equivalent to Equation 4.60 for attitude state differential equations by computing $d\mathbf{\Gamma}^{-1}/dt$ with Padfield's expressions for $d\varphi/dt$, $d\theta/dt$ and $d\psi/dt$ subsequently substituted. The resulting expression for $d\mathbf{\Gamma}^{-1}/dt$ then equals $\mathbf{\Gamma}^{-1}\breve{\boldsymbol{\omega}}_\mathbf{b}$ satisfying Equation 4.60. Thus attitude dynamics are:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + r\cos\varphi\tan\theta + q\sin\varphi\tan\theta \\ q\cos\varphi - r\sin\varphi \\ r\dfrac{\cos\varphi}{\cos\theta} + q\dfrac{\sin\varphi}{\cos\theta} \end{bmatrix} \qquad 4.61$$

Equation 4.61 is consistent, in form, with attitude dynamics quoted in Farell and Barth (1999) with differences attributable to the order of rotations (Section 4.3.2.2).

As mentioned in Section 4.5.1, it is required that estimates of IMU biases are also made. Typically, sensor biases are constant or vary slowly. Thus, these states are modelled as per Equation 4.62 (the mentioned slow variation is accounted for later).

$$\begin{bmatrix} \dot{b}_x \\ \dot{b}_y \\ \dot{b}_z \\ \dot{b}_p \\ \dot{b}_q \\ \dot{b}_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad 4.62$$

Incorporation of the bias terms will necessitate adjustment of the 9 variable navigation state differential equations such that the complete 15 state model is:

$$\begin{bmatrix} \dot{r}_N \\ \dot{r}_E \\ \dot{r}_D \\ \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{b}_x \\ \dot{b}_y \\ \dot{b}_z \\ \dot{b}_p \\ \dot{b}_q \\ \dot{b}_r \end{bmatrix} = \begin{bmatrix} \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \\ (a_x - b_x)\cos\theta\cos\psi + (a_y - b_y)(\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) + (a_z - b_z)(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \\ (a_x - b_x)\cos\theta\sin\psi + (a_y - b_y)(\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) + (a_z - b_z)(\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) \\ -(a_x - b_x)\sin\theta + (a_y - b_y)\sin\varphi\cos\theta + (a_z - b_z)\cos\varphi\cos\theta - g \\ (p - b_p) + (r - b_r)\cos\varphi\tan\theta + (q - b_q)\sin\varphi\tan\theta \\ (q - b_q)\cos\varphi - (r - b_r)\sin\varphi \\ (r - b_r)\frac{\cos\varphi}{\cos\theta} + (q - b_q)\frac{\sin\varphi}{\cos\theta} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad 4.63$$

$$= \mathbf{f}(\mathbf{x}, \mathbf{u_d}, t)$$

The discussion thus far has assumed that the accelerometers and rate gyros of the IMU are noiseless. Naturally, this is untrue. Assume noisy acceleration and rate gyro measurements:

$$\begin{aligned} \tilde{a}_x &= a_x + n_x \\ \tilde{a}_y &= a_y + n_y \\ \tilde{a}_z &= a_z + n_z \\ \tilde{p} &= p + n_p \\ \tilde{q} &= q + n_q \\ \tilde{r} &= r + n_r \end{aligned} \qquad 4.64$$

In Equation 4.64, the $n_\bullet$ terms indicate Gaussian white noise corrupting true IMU measurements producing noisy quantities (indicated by the over ~). It is assumed that these noise terms are uncorrelated with each other as the design of the IMU incorporates independent sensing elements for each channel represented (common mode noise, from a power supply, for example, is neglected). Additionally, assume process noises for bias states, n_{bx}, n_{by}, n_{bz}, n_{bp}, n_{bq} and n_{br} (also white and Gaussian), such that slow, small variations in the biases can be incorporated.

Using Equations 4.63 and 4.64, the complete process model/state differential equations can now be expressed as Equations 4.65, which describes the evolution of navigation states as the sum of a known process model driven by IMU inputs and unknown noise terms. Equation 4.65, additionally, represents a form suitable for use with the Kalman filter.

$$\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}_\mathbf{d}, t) + \mathbf{g}(\mathbf{x}, t)\mathbf{v}(t)$$

$$\mathbf{v}(t) = [n_x \ n_y \ n_z \ n_p \ n_q \ n_r \ n_{bx} \ n_{by} \ n_{bz} \ n_{bp} \ n_{bq} \ n_{br}]^T$$

$$\mathbf{g}(t) = \left[\begin{array}{ccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cos\theta\cos\psi & (\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) & (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cos\theta\sin\psi & (\sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi) & (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos\varphi & \sin\varphi & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right. \tag{4.65}$$

4.5.2.2 Measurement models

4.5.2.2.1 Down position minimum variance estimator

Before considering the measurement model employed, it is necessary to examine the generation of position measurements in the down channel. From the project outset, it was anticipated that the vertical measurement provided by the GPS receiver would be inadequate for control in this channel. The altitude measurements produced by the Lassen IQ module were not accurate enough for positioning the helicopter in the vertical channel which would be especially critical considering that the helicopter was flown near the ground. As such, supplementation of the GPS measurement was envisioned by other altitude measuring devices. Although the Kalman filter would provide a convenient method for adding such redundant information, the exact sensors to be employed were not known at time of filter design. Thus, it was decided to feed a single down position measurement to the Kalman filter which would consist of several independent measurement sources pre combined in some way. This combination was achieved with a linear minimum variance estimator, which ultimately generated one down position measurement, $r_{D(MVE)}$, and associated covariance, $\sigma^2{}_{D(MVE)}$.

Equations 4.66 show the expressions needed to combine altimeter and GPS positions and associated covariances, assuming that these measurement sources are independent and therefore uncorrelated. In Equations 4.66, y_1 and y_2 represent altimeter and GPS readings respectively, and, σ_1^2 and σ_2^2 represent altimeter and GPS covariances respectively.

$$\begin{aligned}
r_{D(MVE)} &= \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} y_2 \\
\sigma^2{}_{MVE} &= \left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2 \sigma_1^2 + \left(\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2 \sigma_2^2
\end{aligned} \tag{4.66}$$

Derivations for Equations 4.66, and for the case of 3 measurements are included in Appendix 2.

4.5.2.2.2 Model

In terms of the sensors used, only the GPS down position reading and barometric altimeter are pre combined as per Section 4.5.2.2.1.

The navigation Kalman filter will be supplied with noisy GPS measurements of position and velocity. GPS generated measurement of position in latitude, longitude and altitude are converted by expressions of Section 4.3.2.2.3 into position readings directly comparable with the INS Kalman filter states i.e.

$$\begin{aligned} r_{N(GPS)} &= r_N \\ r_{E(GPS)} &= r_E \\ r_{D(MVE)} &= r_D \end{aligned} \qquad 4.67$$

Further, the GPS velocities are related to the state variables thusly;

$$\begin{aligned} v_{N(GPS)} &= v_N \\ v_{E(GPS)} &= v_E \\ v_{U(GPS)} &= -v_D \end{aligned} \qquad 4.68$$

Finally, cognisance must be taken of the inaccuracies present in the GPS position and velocity solutions. Incorporating these, a measurement model, in line with the Kalman filtering structure, can be constructed (Equation 4.69).

$$\begin{bmatrix} r_{N(GPS)} \\ r_{E(GPS)} \\ r_{D(MVE)} \\ v_{N(GPS)} \\ v_{E(GPS)} \\ v_{U(GPS)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_N \\ r_E \\ r_D \\ v_N \\ v_E \\ v_D \\ \varphi \\ \theta \\ \psi \\ b_x \\ b_y \\ b_z \\ b_p \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} n_{N(GPS)} \\ n_{E(GPS)} \\ n_{D(MVE)} \\ n_{vN(GPS)} \\ n_{vE(GPS)} \\ n_{vU(GPS)} \end{bmatrix} \qquad 4.69$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, t) + \mathbf{w}(t)$$

Where $\mathbf{w}(t) = [n_{N(GPS)}\ n_{E(GPS)}\ n_{D(GPS)}\ n_{vN(GPS)}\ n_{vE(GPS)}\ n_{vU(GPS)}]^T$ represents Gaussian white noise terms corrupting external measurements.

4.5.3 Kalman filter parameters and setup

4.5.3.1 Use of the extended Kalman filter

As noted in Section 4.4.4, neither the linearised nor the extended Kalman filter is a perfect solution to the non linear filtering problem. For this project the extended Kalman filter was favoured since:

a. A nominal trajectory did not exist. The helicopter was flown by a human pilot, thus following of some predetermined trajectory could not be guaranteed to an extent that would hold deviations from nominal small.
b. The initial position of the helicopter was known almost exactly removing some of the extended Kalman filter risk of divergence identified by Brown and Hwang (1992).
c. Calculation issues noted by Grewal and Andrews (2001) were negated by the computational bandwidth offered by the 'F2812 DSP.
d. In terms of Section 4.5.2.1, the process model equations are non linear.

The extended Kalman filter will be used in conjunction with the concepts of Section 4.4.4.3 such that total navigation quantities may be computed. In terms of the method quoted therein:

1. State transition and measurement dynamics matrices ($\mathbf{A_k}$ and $\mathbf{H_k}$) from the linearised model are required for the Kalman filter.
2. A method for projecting $\hat{\mathbf{x}}_\mathbf{k}$ to $\hat{\mathbf{x}}_\mathbf{k+1}^-$ by solving the differential equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u_d}, t)\big|_{\mathbf{x}=\hat{\mathbf{x}}_\mathbf{k}, t=t_k}$ is required. $\mathbf{f}$ is the non linear noiseless process dynamics.
3. The process noise covariance associated with the linearised model, $\mathbf{Q_k}$, is needed.

Generating $\mathbf{A_k}$ involves computing, in real time, the Jacobian of the noiseless non linear process dynamics. Following which, the Jacobian is discretised for digital implementation. Symbolically (Section 4.5.2.1 for $\mathbf{f}$):

$$\begin{aligned} \mathbf{T_k} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, t_k\right) \\ and\ \mathbf{A_k} &= e^{\mathbf{T_k}\Delta t} \end{aligned} \qquad 4.70$$

Practically, the Jacobian was computed using the symbolic tools available in Matlab (as the result is sparse) and discretisation, which requires application of the matrix exponent to the Jacobian multiplied by the calculation intervals, is achieved by the approximation, $e^{\bullet} \approx \mathbf{I} + \bullet$.

From Section 4.5.2.2.2, it should be clear that the measurement model requires no adjustment as it is linear.

With reference to Point (2), a priori estimate generation is achieved by numerically integrating $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u_d}, t)$ using the Explicit Euler method,

$$\begin{aligned} &\mathbf{x_{i+1}} = \mathbf{x_i} + \mathbf{f}\Delta t \quad (generally) \\ &\Rightarrow \hat{\mathbf{x}}_\mathbf{k+1}{}^- = \hat{\mathbf{x}}_\mathbf{k} + \mathbf{f}\Delta t \end{aligned} \qquad 4.71$$

In Equation 4.71:

1. $\mathbf{f}$ represents process dynamics in the absence of noise.
2. Δt represents the calculation intervals.
3. $\hat{\mathbf{x}}_{\mathbf{k}}$ is the estimate for the previous time step.
4. $\hat{\mathbf{x}}_{\mathbf{k+1}}^{-}$ is the required a priori estimate.

A discussion of the development of $\mathbf{Q}_{\mathbf{k}}$ for the extended Kalman filter follows in Section 4.5.3.2.

4.5.3.2 Determination of process and measurement noise covariances

The process and measurement noise covariances provide the statistical descriptions of the Kalman filter instrument noises required to stochastically blend estimates and measurements. Process noise covariances are mostly directly related to IMU errors, whereas, measurement noise covariances are linked to GPS and barometric altimeter errors.

4.5.3.2.1 Process noise covariances

$\mathbf{Q}_{\mathbf{k}}$, the process noise covariance, is determined by computing the covariance of the noise terms associated with the process dynamics (Equation 4.65).

$$\mathbf{Q}_{\mathbf{k}} = \mathrm{cov}[\mathbf{Gn}] \qquad 4.72$$

Where:

1. cov represents covariance.
2. $\mathbf{G}$ is the matrix form of $\mathbf{g}(\mathbf{x},t)$ from Equation 4.65.
3. $\mathbf{n}$ is the noise vector from Equation 4.65.

Once the current filter estimate is substituted into $\mathbf{G}$, it is merely a matrix of scalings that can easily be manipulated into an expression for $\mathbf{Q}_{\mathbf{k}}$. In such an expression, the only unknown is the covariance of $\mathbf{n}$ (Equation 4.73).

$$\mathbf{Q}_{\mathbf{k}} = \mathbf{G}\,\mathrm{cov}[\mathbf{n}]\mathbf{G}^{T} = \mathbf{G}E(\mathbf{nn}^{T})\mathbf{G}^{T} \qquad 4.73$$

$\mathbf{G}$ is readily available at each epoch implying that only the covariance of the signals $\mathbf{n}$ is required to completely solve for $\mathbf{Q}_{\mathbf{k}}$. $E(\mathbf{nn}^{T})$ is the expectation of $\mathbf{nn}^{T}$, which, for zero mean signals is the covariance matrix with elements representing individual covariances along the main diagonal (Equation 4.74).

$$E(\mathbf{nn}^{T}) = diag\left[\sigma^{2}{}_{nx} \quad \sigma^{2}{}_{ny} \quad \sigma^{2}{}_{nz} \quad \sigma^{2}{}_{np} \quad \sigma^{2}{}_{nq} \quad \sigma^{2}{}_{nr} \quad \sigma^{2}{}_{nbx} \quad \sigma^{2}{}_{nby} \quad \sigma^{2}{}_{nbz} \quad \sigma^{2}{}_{nbp} \quad \sigma^{2}{}_{nbq} \quad \sigma^{2}{}_{nbr}\right] \qquad 4.74$$

Where subscripts relate to noise terms from Equation 4.65 in Section 4.5.2.1. The first six elements from Equation 4.74 relate to covariances associated with IMU noise. From Section 3.3.3.1, IMU noises are quoted as RMS quantities. The RMS, mean and covariance can be related by Equation 4.75, where x is a random variable and the over bar indicates mean.

$${x_{rms}}^{2} = \bar{x}^{2} + {\sigma_{x}}^{2} \qquad 4.75$$

If it is assumed that the noise is zero mean, then the covariance is merely the square of the quoted RMS noise. In terms of noise parameters provided, an additional complication exists in that these are given as incremental quantities that must be combined with the output sample rate of the IMU to produce noise specifications for acceleration and angular velocity. Thus:

1. For accelerations the quoted 0.005 m/s incremental RMS velocity noise becomes, at 64 Hz output rate, acceleration RMS noise of 0.32ms^{-2}. This is as a result of the output format of acceleration data from the IMU – incremental velocity. In general, if from one 64 Hz sample to the next, the output incremental velocity is v m/s, then the acceleration, a, can be given by Equation 4.76, which has been applied to convert the incremental RMS velocity noise to acceleration RMS noise (above).

$$a = v \ {}^{m}\!/_{s} \times {}^{64}\!/_{s} = (v \times 64) \ {}^{m}\!/_{s^2} \tag{4.76}$$

   In the preceeding discussion, 64 Hz is used (even though every second IMU sample is employed, effectively, producing IMU data at 32 Hz) as the IMU output rate is still 64 Hz. Incremental inertial measurements, as produced by the unit, must be viewed in this context. Decimation to 32 Hz, to allow sufficient inter sample processing time, occurs via the avionics software.

   Once the conversion from velocity increment to acceleration has been made, Equation 4.75 can be applied to produce the acceleration noise covariances of Equation 4.77.

$$\sigma^2{}_{nx} = \sigma^2{}_{ny} = \sigma^2{}_{nz} = 0.1024 \ (ms^{-2})^2 \tag{4.77}$$

2. For angular velocities the quoted 5.236 x 10^{-4} rad incremental angular noise per sample becomes, at 64 Hz output rate (the comments from Point (1) above relating to sampling rate applies), angular velocity RMS noise of 0.0355 rad/s. As per the incremental velocity/acceleration case, this is as a result of the output format of angular velocity from the IMU – incremental angle. In general, if from one 64 Hz sample to the next, the output incremental angle is α rad, then the angular velocity, ρ, can be given by Equation 4.78, which has been applied to convert the incremental angular noise to angular velocity RMS noise.

$$\rho = \alpha \ rad \times {}^{64}\!/_{s} = (\alpha \times 64) \ {}^{rad}\!/_{s}. \tag{4.78}$$

   Once this conversion has been made, Equation 4.75 is applied to arrive at the noise covariance.

$$\sigma^2{}_{np} = \sigma^2{}_{nq} = \sigma^2{}_{nr} = 1.123 \times 10^{-4} \ (rad/s)^2. \tag{4.79}$$

The latter six elements of Equation 4.74 describe the statistics of noises associated with the IMU bias model included in the state mechanisation equations. As mentioned in Section 4.5.2.1 these should encapsulate the variation of biases. From Section 3.3.3.1, the IMU has associated worst case residual bias of 5 mg and 0.5 °/s. No bias drift characteristic is provided, thus, assume that the bias cannot change by more than one residual bias unit per hour. Implying standard deviations of:

1. Acceleration bias drift of approximately 0.05 (ms^{-2})/h.
2. Angular velocity bias drift of approximately 8.727 x 10^{-3} (rad/s)/h.

However,

1. The minimum quantisation for accelerometers is 0.064 ms^{-2} implying that the bias drift will be seen as 0.064 (ms^{-2})/h which translates to 5.556 x 10^{-7} (ms^{-2})/sample.
2. Similarly for angular velocity with minimum quantisation of 9.758 x 10^{-4} rad/s (implying that 9 quantisations are required to encapsulate hourly drift above), the bias drift is 7.6305 x 10^{-8} (rad/s)/sample.

Points (1) and (2) above represent the maximum drift per sample. It should be appreciated that these are worst case approximations resulting from quantisation effects. If it is assumed that the drift per sample is subsequently uniformly distributed and ideally zero (implying zero mean), an expression for the covariance can be developed. Consider the general scenario below with uniformly distributed random variable X having maximum/minimum value ±m and probability density function equal to a.

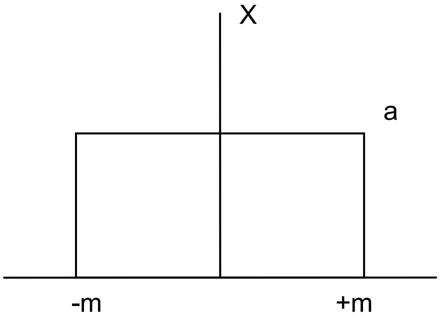


Figure 4.20. Uniformly distributed random variable X.

From basic statistics,

$$\int_{-m}^{m} a \, dX = a(2m) = 1 \Rightarrow a = \frac{1}{2m} \qquad 4.80$$

With X zero mean,

$$\sigma^2{}_X = \frac{1}{2m} \int_{-m}^{m} (x - \bar{x}) \, dX = \frac{m^2}{3} \qquad 4.81$$

Equation 4.81 can now be employed to find bias noise covariances, with m from Points (1) and (2) above.

$$\begin{aligned} \sigma^2{}_{nbx} &= \sigma^2{}_{nby} = \sigma^2{}_{nbz} = (1/3)(7.630 \times 10^{-8})^2 = 1.9403 \times 10^{-15} \, (ms^{-2})^2 \\ \sigma^2{}_{nbp} &= \sigma^2{}_{nbq} = \sigma^2{}_{nbr} = (1/3)(5.556 \times 10^{-7})^2 = 1.029 \times 10^{-13} \, (rad/s)^2 \end{aligned} \qquad 4.82$$

Results from Equations 4.77, 4.79 and 4.82 can be substituted into Equation 4.74. Thus, as matrix $\mathbf{G}$ is computed at each epoch, Equation 4.73 can be employed to calculate $\mathbf{Q_k}$ online.

4.5.3.2.2 Measurement noise covariances

The measurement noise covariance matrix, $\mathbf{R_k}$, defines the statistical characteristic of errors in external measurements processed by the Kalman filter i.e. Diagonal elements of $\mathbf{R_k}$ define covariances of measurements in measurement vector, $\mathbf{z_k}$.

The measurement vector consists of three position measurements and three velocity measurements (both triplets in the north, east and down system). Generation of the measurement covariance matrix requires that covariances associated with positions and velocities are found. Thus:

1. The covariances of the GPS north and east position measurements are required.
2. The covariance of the GPS down position measurement is required for fusion with the barometric altimeter measurement producing an overall down measurement covariance (Section 4.5.2.2.1).
3. The covariances of GPS north, east and down velocity readings are required.

In terms of GPS position solution, AMCE (2003) quotes two key factors that degrade measurement accuracy:

1. The user equivalent range error (UERE), which relates to the accuracy of pseudo range measurements and which depends on satellite, atmospheric conditions, receiver and GPS positioning code employed.
2. The geometric dilution of precision (GDOP) which relates to the spatial relationship of the satellites relative to the receiver and, which, due to the continuous motion of the satellites, changes. Three dimesional GPS position calculations are made using range measurements to a minimum of four satellites. Should these satellites be clustered close together, these range measurements are nearly equal, causing small relative errors to be exaggerated when range measurements are differenced to form position solutions (Grewal, Andrews and Weill, 2001). Related parameters also exist for horizontal and vertical dilution of precision parametrics: HDOP and VDOP respectively.

AMCE (2003) additionally gives an expression for computing the position solution accuracy using the relevant dilution of precision and the UERE:

$$\sigma = \sigma_R \cdot DOP \qquad 4.83$$

Where:

1. σ is the position solution standard deviation.
2. σ_R is the UERE.
3. DOP is the horizontal or vertical dilution of precision.

The LassenIQ GPS receiver data frame provides the position and velocity solution, and the estimated DOPs for both the horizontal and vertical accuracy determination as per Equation 4.83. The UERE is, however, required. Manufacturer specifications for the LassenIQ module quote horizontal and vertical position accuracy as <6 m (50 %) and <8 m (50 %). Although not explicitly stated, it is assumed, due to the format in which the above characteristics are quoted, that this relates to the Circular Error Probable (CEP) statistic. Using a conversion table found in AMCE (2003), CEP statistics can be converted to standard deviations as follows:

1. For linear measurement e.g. Vertical.

$$0.6745\sigma = CEP \tag{4.84}$$

2. For two dimensional measurements e.g. Horizontal.

$$1.177\sigma = CEP \tag{4.85}$$

Applying Equations 4.84 and 4.85 to the manufacturer's error statistics gives:

$$\begin{aligned} \sigma_{R(VERTICAL)} &= 14.83\ m \\ \sigma_{R(HORIZONTAL)} &= 6.007\ m \end{aligned} \tag{4.86}$$

Thus, with Equation 4.83, and the relevant online DOP output by the GPS unit, it is possible to generate the required position solution covariances as each GPS reading is output. These covariances are then computed in real time, dependent on the number and orientation of GPS satellites visible to the receiver.

Additionally, to generate the GPS and barometric altimeter fused covariance (as per Section 4.5.2.2.1), the covariance of the barometric altimeter is needed. It is known from the barometer data sheet that the absolute pressure accuracy is ±1.5 mbar and that the error caused by using a 3.3 V power supply is ≤0.4 mbar. Thus, in terms of error budget, it is safe to assume a worst case pressure error of approximately 2 mbar. This can be translated to worst case altitude error using the pressure to altitude relationship (Equation 4.87 from Young et al., 1996). Equation 4.87 is further used to generate down position readings from measured barometric pressures.

$$p = p_0 \exp\left(-\frac{Mgy}{RT}\right) \tag{4.87}$$

Where

1. p_0 is the pressure at sea level (1013.25 mbar or 1.013 x 10^5 Pa).
2. R is the gas constant (8.315 J/(mol·K)).
3. T is the ambient temperature in Kelvin.
4. M is the molar mass of air (28.8 x 10^{-3} kg/mol).
5. p is the actual pressure (p_0 and p must be expressed in the same pressure unit).
6. y is the altitude above sea level in metres.

Equation 4.87, in differential terms, with pressure error Δp gives altitude error Δy (in metres).

$$\Delta y = -\frac{RT}{Mg} \cdot \frac{\Delta p}{p} \tag{4.88}$$

Assuming the following nominal conditions and worst case Δp of ±2 mbar, Equation 4.88 gives a maximum altitude error of approximately ±17.54 m.

1. T = 298 K.
2. g = 9.81 ms^{-2}.
3. p = 1013.25 mbar.

Again, assuming a uniform distribution about zero mean bounded by ±17.54 m, Equation 4.81 can be applied giving:

$$\sigma_{alt}{}^{2} \approx 102\ m^{2} \Rightarrow \sigma_{alt} \approx 10\ m. \qquad 4.89$$

Finally, the GPS velocity solution is assigned an accuracy of 0.06 m/s from the device data sheet. No other information is provided, and, as such, Equation 4.81 and the assumptions thereof were used to develop the requisite covariances of 0.0012 (m/s)^2 (0.035 m/s). The covariance is the same for all channels – only one velocity statistic is available.

4.5.3.3 Navigation system initialisation and alignment

Grewal, Andrews and Weill (2001) refer to INS initialisation as the process of determining initial values for position, velocity and biases. Attitude (Euler angle) initialisation is known as alignment. Prior to the navigation Kalman filter entering the iterative computation loop, all filter states and associated state covariances must be initialised. This will form the first a priori state ($\hat{\mathbf{x}}_{t=0}{}^{-}$) and associated state error covariance ($\mathbf{P}_{t=0}{}^{-}$).

Usually position initialisation is executed by employing an external source e.g. GPS reading or manual crew entry (Grewal, Andrews and Weill, 2001). Here GPS data are used indirectly as it is assumed that the initial spatial position of the INS is the origin for the navigation frame i.e. The INS is positioned on the test field and the position at which the INS is initialised is noted as the navigation frame origin. Naturally, the corresponding GPS location at this spot is recorded such that future GPS data may be referenced to this point. Thus for initialisation of the first three states:

$$\left[\hat{r}_N{}^{-} \quad \hat{r}_E{}^{-} \quad \hat{r}_D{}^{-}\right]_{t=0} = [0 \quad 0 \quad 0] \qquad 4.90$$

As, by the above assumption, the position is known exactly, the initial state error covariance is:

$$\left[\sigma_{rN}{}^{2} \quad \sigma_{rE}{}^{2} \quad \sigma_{rD}{}^{2}\right]_{t=0} = [0 \quad 0 \quad 0] \qquad 4.91$$

For the velocity triplet, initialisation is typically done at rest and, as the helicopter is always stationary on the test field when the filter is started (the INS is always started with the helicopter waiting for take off), this is the scenario here also, implying:

$$\left[\hat{v}_N{}^{-} \quad \hat{v}_E{}^{-} \quad \hat{v}_D{}^{-}\right]_{t=0} = [0 \quad 0 \quad 0] \qquad 4.92$$

Again, the initial state is known exactly, thus:

$$\left[\sigma_{vN}{}^{2} \quad \sigma_{vE}{}^{2} \quad \sigma_{vD}{}^{2}\right]_{t=0} = [0 \quad 0 \quad 0] \qquad 4.93$$

Grewal, Andrews and Weill (2001) further refers to the alignment process of a strap down system as being key to determining the initial coordinate transformation from the body to the navigation frame. Typically, there are two components to the alignment procedure: horizontal alignment (Euler roll and pitch angles) and azimuth alignment (Euler yaw angle). Alignment can further be decomposed into coarse and fine alignment.

A common method for coarse alignment, loosely referred to as gyro compassing by Grewal, Andrews and Weill (2001), involves employing acceleration and angular velocity data directly. This method, which is used in the developed INS, uses accelerometer readings to obtain the local vertical and hence determine pitch and roll angles by using the DCM i.e. The system of Equations 4.94 is solved for φ and θ, using accelerometer readings $[a_x \; a_y \; a_z]$ and the norm of the accelerometer triplet as gravity, g:

$$\begin{gathered} g = \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2} \\ \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} -g\sin\theta \\ g\sin\varphi\cos\theta \\ g\cos\varphi\cos\theta \end{bmatrix} \end{gathered} \qquad 4.94$$

Practically, even with the helicopter engine off, the IMU readings are somewhat corrupted by noise. As such, the computation of Equations 4.94 is processed repeatedly over a specific time period such that average results for initial φ and θ can be found. A consequence of this is that initial covariances for φ and θ can also be computed. Additionally, it should be noted that gravity, as seen by the IMU, is also repeatedly calculated during this process and averaging of these results was used to assign gravity in terms of the hardware used.

Azimuth alignment, with respect to gyro compassing, involves measurement of the Earth's rotation to determine initial yaw. This method could not be used, as the IMU gyroscopes lack the sensitivity required to discriminate noise from the Earth's rotation. As such, the output of the two axes magnetic compass, with the INS nearly level (as was usually the start position), was used to determine the initial yaw. Again, an averaging process over multiple samples was used to obtain a mean initial yaw and covariance.

Finally, the bias states need to be initialised. It was assumed that all biases are initially zero. The gyro compassing process unfortunately accounts for any accelerometer bias by manipulating the appropriate horizontal tilt angle (pitch or roll). This is unavoidable, however, the initial bias covariances were non zero, allowing the filter to adjust these. It is known that the residual accelerometer and rate gyro biases are 5 mg and 0.5 °/s respectively. Applying the method of Equation 4.81 and assuming that the initial bias is distributed uniformly between ±5 mg and ±0.5 °/s yields:

$$\begin{gathered} {\sigma_{bx}}^2 = {\sigma_{by}}^2 = {\sigma_{bz}}^2 = \frac{(0.005g)^2}{3} \\ {\sigma_{bp}}^2 = {\sigma_{bq}}^2 = {\sigma_{br}}^2 = 2.5403 \times 10^{-5} \; (rad/s)^2 \end{gathered} \qquad 4.95$$

4.5.4 Navigation filter computation cycle

Figure 4.22 illustrates the navigation filters computation cycle. These iterative calculations are triggered by the IMU outputting samples when the avionics system is in navigation mode. As mentioned in Section 3.4.1.3, the IMU and GPS samples are asynchronous, with the latest new GPS sample being incorporated into the filter with the first IMU sample following such GPS reception (Figure 4.21). This results in a maximum delay of one IMU sample (1/32 seconds) before GPS data are assimilated into the filter. In terms of this project, only slow, benign movements of the helicopter were envisioned and, as such, the jitter mentioned is small enough to be neglected.

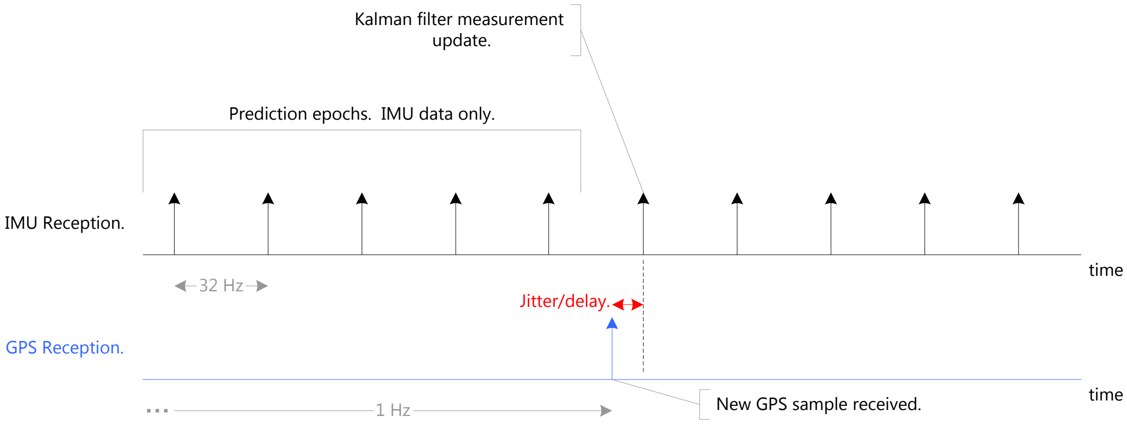
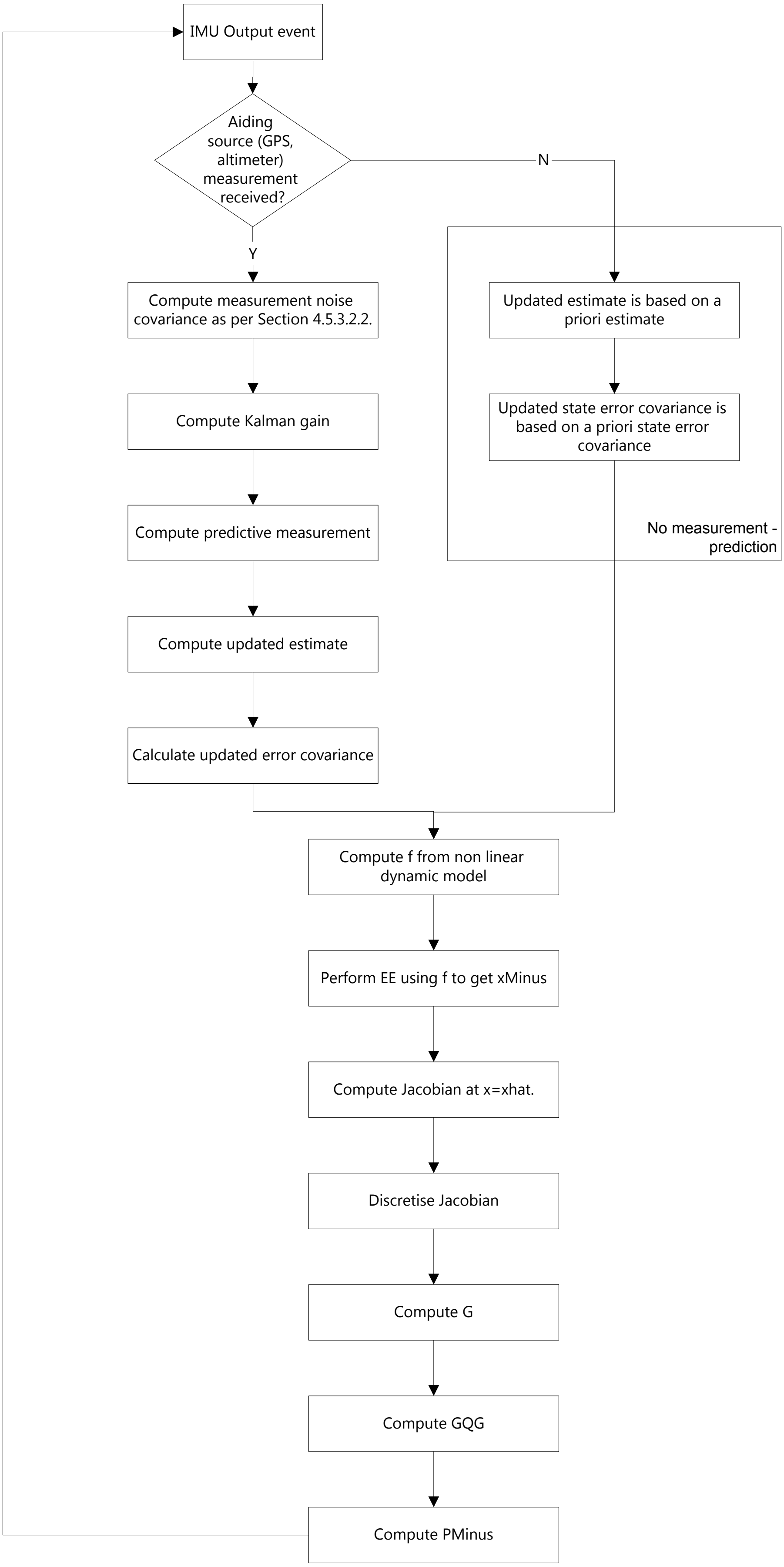


Figure 4.21. Asynchronisity of GPS and IMU samples.

It should be noted, from Figure 4.22, that, in the absence of an external measurement update, the filter is operated with prediction only. As per Grewal and Andrews (2001), this essentially requires that the measurement sensitivity, $\mathbf{H}_\mathbf{k} = \mathbf{0}$. As is evident from the Kalman gain equation (repeated below), with the measurement sensitivity equal to zero, the Kalman gain, $\mathbf{K}_\mathbf{k}$, also collapses to zero. Naturally, in this scenario, $\mathbf{R}_\mathbf{k}$, representing the non existent measurements' covariances, can be any value except zero. Thus Equations 4.96 result providing the "updated" estimate and state error covariance.

$$
\begin{aligned}
\mathbf{K}_\mathbf{k} &= \mathbf{P}_\mathbf{k}\overbrace{\mathbf{H}_\mathbf{k}{}^{T}}^{=0}\left(\mathbf{H}_\mathbf{k}\mathbf{P}_\mathbf{k}{}^{-}\mathbf{H}_\mathbf{k}{}^{T} + \mathbf{R}_\mathbf{k}\right)^{-1} = \mathbf{0} \\
\hat{\mathbf{x}}_\mathbf{k} &= \hat{\mathbf{x}}_\mathbf{k}^{-} + \overbrace{\mathbf{K}_\mathbf{k}}^{0}(\mathbf{z}_\mathbf{k} - \overbrace{\mathbf{H}_\mathbf{k}}^{0}\hat{\mathbf{x}}_\mathbf{k}^{-}) \quad \Rightarrow \quad \hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{k}{}^{-} \\
\mathbf{P}_\mathbf{k} &= (\mathbf{I} - \overbrace{\mathbf{K}_\mathbf{k}}^{0}\overbrace{\mathbf{H}_\mathbf{k}}^{0})\mathbf{P}_\mathbf{k}^{-} \quad \Rightarrow \quad \mathbf{P}_\mathbf{k} = \mathbf{P}_\mathbf{k}{}^{-}
\end{aligned}
\qquad 4.96
$$

Practically, in the absence of a measurement update, it is unnecessary to compute $\mathbf{K}_\mathbf{k}$ using $\mathbf{H}_\mathbf{k} = \mathbf{0}$. Instead, the filter is run with predicition only, with Equations 4.96 facilitating continuation of the recusive loop.



IMU Output event
Aiding source (GPS, altimeter) measurement received?
N
Y
Compute measurement noise covariance as per Section 4.5.3.2.2.
Compute Kalman gain
Compute predictive measurement
Compute updated estimate
Calculate updated error covariance
Updated estimate is based on a priori estimate
Updated state error covariance is based on a priori state error covariance
No measurement - prediction
Compute f from non linear dynamic model
Perform EE using f to get xMinus
Compute Jacobian at x=xhat.
Discretise Jacobian
Compute G
Compute GQG
Compute PMinus

Figure 4.22. Navigation filter computational cycle.

4.5.5 Matlab filter implementation

In an effort to test and examine the operation of the navigation system, the Kalman filter designed thus far was implemented with Matlab. Once the operation of the filter algorithm was verified, the algorithm was translated into equivalent C code for embedded execution on the DSP. The Matlab implementation provides the added benefit of verifying DSP computed results. DSP computed states can easily be compared to the Matlab filter run offline with the same experimental inputs.

4.5.5.1 Baseline implementation

The basic Matlab solution consisted of a combination of Matlab scripts, functions and Simulink models. Appendix 3 provides a copy of the baseline Matlab solution discussed here. It should be appreciated that the Matlab filter implementation was developed with embedded system deployment in mind. As such the structure of the solution is as follows:

1. A primary Kalman filtering Matlab function that performs Kalman filtering measurement updates or prediction as per Section 4.5.4 (KFilterDOP.m).
2. A Simulink model that wraps the above function such that logged accelerations, angular velocities and GPS data may be supplied to the filter and navigation outputs generated (FiltExpDOP.m). The Simulink model calls the KFilterDOP function for each time step and provides the appropriate acceleration, angular velocity and GPS data for that time. Navigation outputs are sent to the Matlab workspace. In this manner, offline navigation results can be obtained if raw sensor data is available.
3. A Matlab function (ResetFilter.m) is used to reset the filter such that for multiple executions of the offline computation, the filter states and covariances are returned to their initial values.
4. The Matlab script file (Parameters.m) is used to assign all physical constants, masses and other constant data.
5. The Matlab script (LoadFTData.m) loads raw sensor data, as collected from the helicopter during flight tests. Additionally, some pre processing of the raw data occurs:
    a. GPS position samples are converted from the latitude/longitude/altitude to positions in the north, east and down frame using methods of Section 4.3.2.2.3.
    b. GPS velocity samples are converted from the north, east and up frame to the north, east and down frame (Section 4.3.2.2.3).
    c. IMU accelerations and angular velocities are converted from quantisations to values in ms^{-2} and rad/s.
    d. Start and end times, used in the Simulink environment, are extracted.
6. The Matlab script, InitFilter.m, is a wrapper script for elements (1)-(5) above that, apart from calling the above components, prepares the Matlab workspace (e.g. Declares global variables). Figure 4.23 illustrates the flow of operations initiated by InitFilter.m

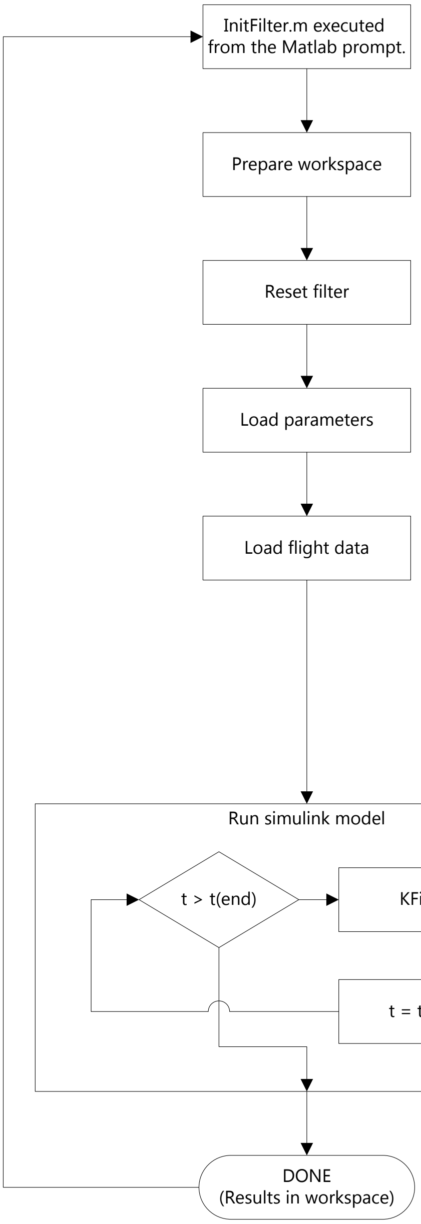


Figure 4.23. Operations initiated by InitFilter.m.

4.5.5.2 Initial filter test

Operation of the filter of Section 4.5.5.1 was initially validated by capturing the raw IMU and GPS inputs from the helicopter and passing these through the Matlab setup offline. As should be clear from the previous section, the Matlab function KFilterDOP is key to the INS operation and the tests listed here were fundamentally to confirm reasonable operation of this function.

Tweaking of parameters and initialisation quantities was done for testing of the embedded INS solution (Section 4.5.6). It should also be noted that the barometric altimeter and compass were unavailable when these experiments were conducted (these items were still on order from respective manufacturers). It was still possible to validate the filter operation since:

1. The altimeter result is incorporated into the down position measurement external to the Kalman filter. For this validation GPS altitude was used solely.

2. The compass was used only for yaw initialisation which, in the absence of the CMPS03, was effected using an external magnetic compass, the reading from which was manually entered into the system.

Two flights were performed with the helicopter avionics configured to output sensor data for logging to the PC ground station. A human pilot was in control of the helicopter. Table 4.2 lists the attempted manoeuvres and approximate times for each phase of the flight.

Table 4.2. Initial flight test - flight plans.

Flight test 1		**Flight test 2**	
Manoeuvre/Phase	**Duration (s)**	**Manoeuvre/Phase**	**Duration (s)**
Stationary (on ground, motor running)	20	Stationary (on ground, motor running)	10
Ascent	20	Ascent	10
Hover	20	Forward flight (nose in)	15
Descent	20	Backward flight (tail in)	15
Stationary	20	Descent	10
		Stationary	10

As far as possible, the pilot attempted to:

1. Ensure vertical ascents and descents.
2. Land at the point of take off.

The flights of Table 4.2 did not follow a strict path as, at this stage of the project, the pilot was still honing his skills. As such the tests here were more for validating the "shape" of the solution and ensuring no unforeseen numerical pitfalls. Section 4.5.6 provides results from more strictly controlled tests.

The fifteen state variables, as computed offline, using the logged data from the flight tests are plotted in Figure 4.24 to Figure 4.54 (below). For position and velocity, the relevant plots include the GPS measurements. Additionally, for position, velocity and Euler angles, the relevant plots show these quantities determined from pure integration of the respective rate data (all reflected on a much larger scale on the right hand sides of the appropriate figures):

1. For position, double integration of the acceleration measurements.
2. For velocity, single integration of the acceleration data.
3. For Euler angle, single integration of the angular velocity data.

The above additions to the result set are included for comparison in the absence of filtering. Plots for flight test 2 also include a horizontal position plot illustrating the attempted flight path.

4.5.5.2.1 Navigation result: Flight test 1

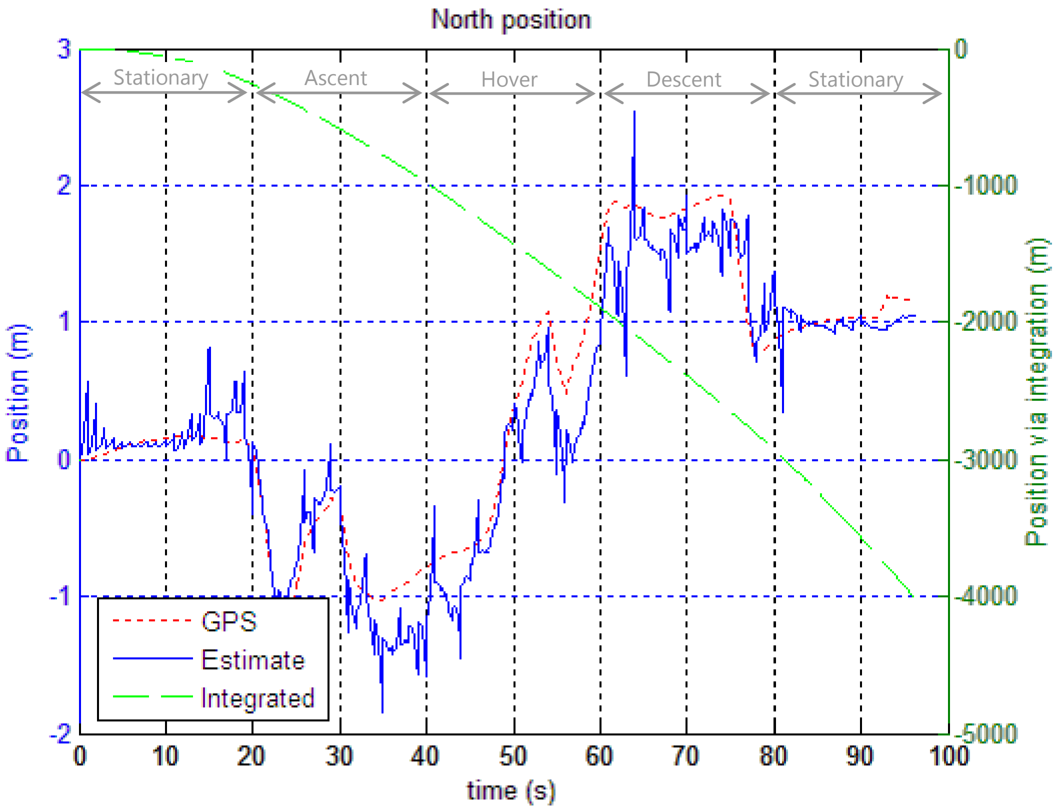


Figure 4.24. Flight test 1: North position results.

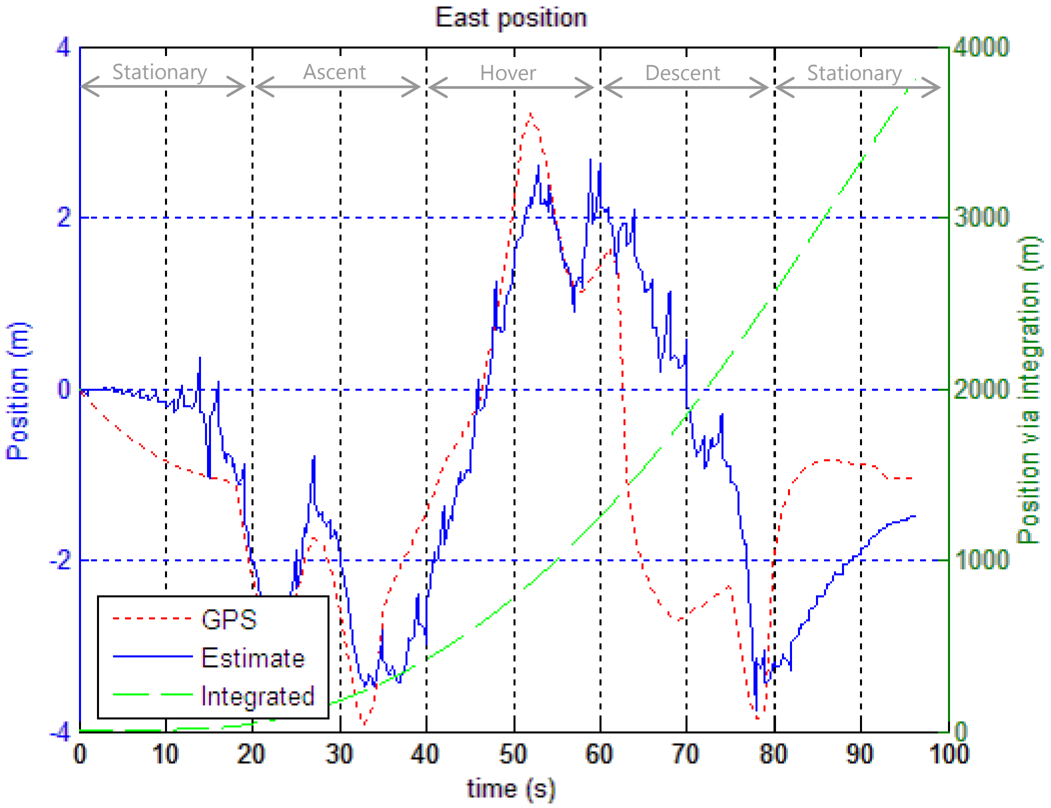


Figure 4.25. Flight test 1: East position results.

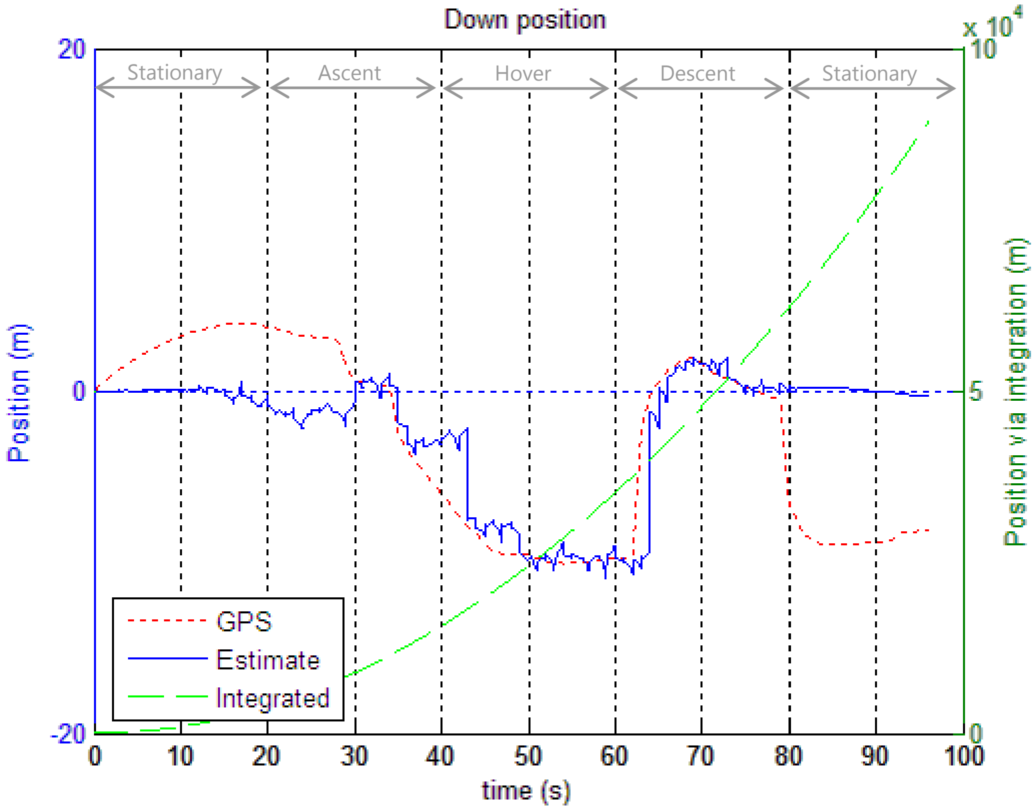


Figure 4.26. Flight test 1: Down position results.

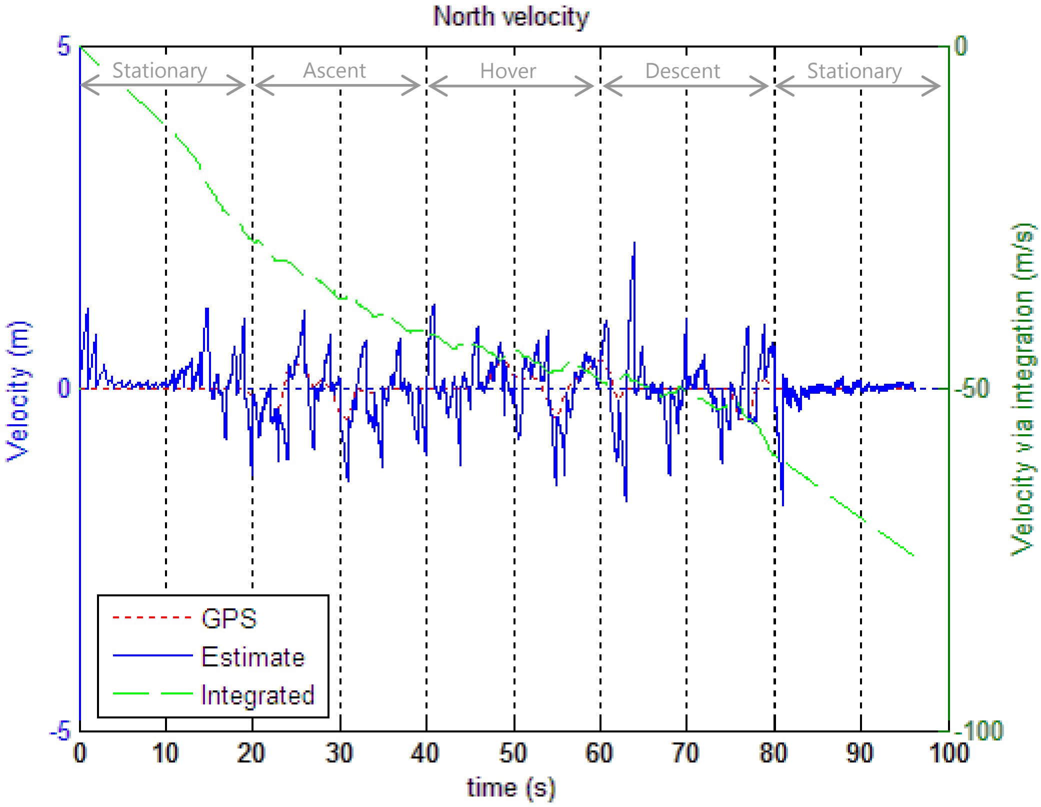


Figure 4.27. Flight test 1: North velocity results.

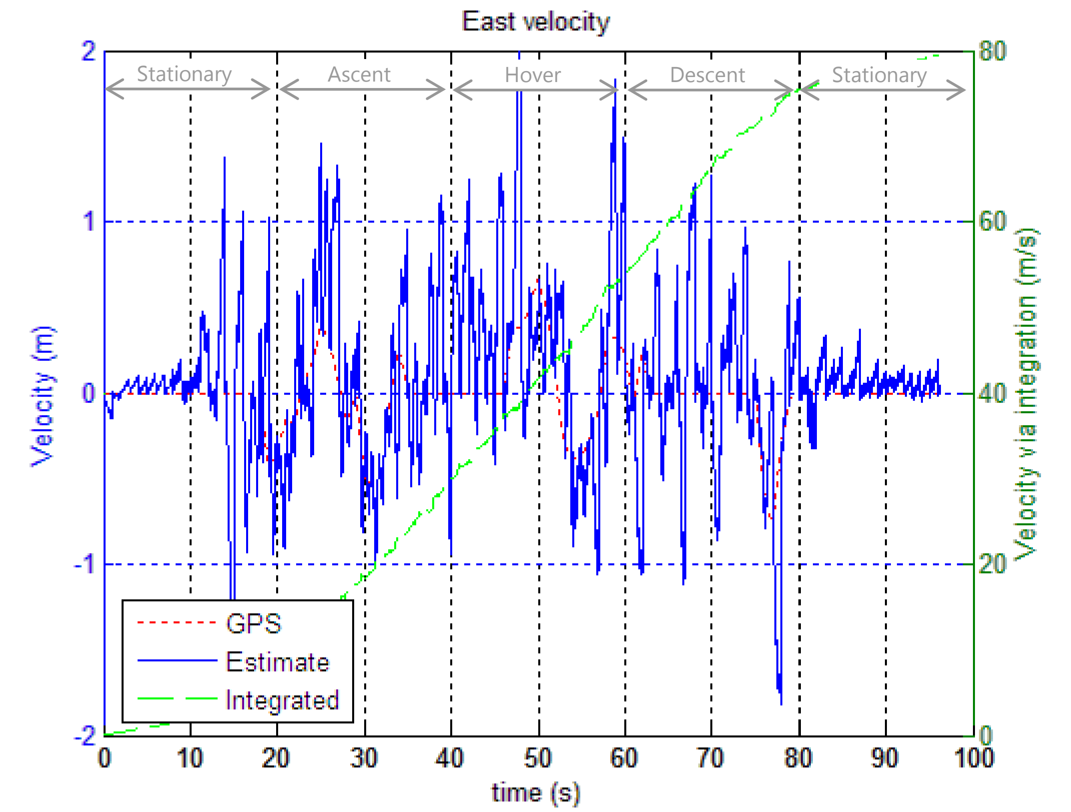


Figure 4.28. Flight test 1: East velocity results.

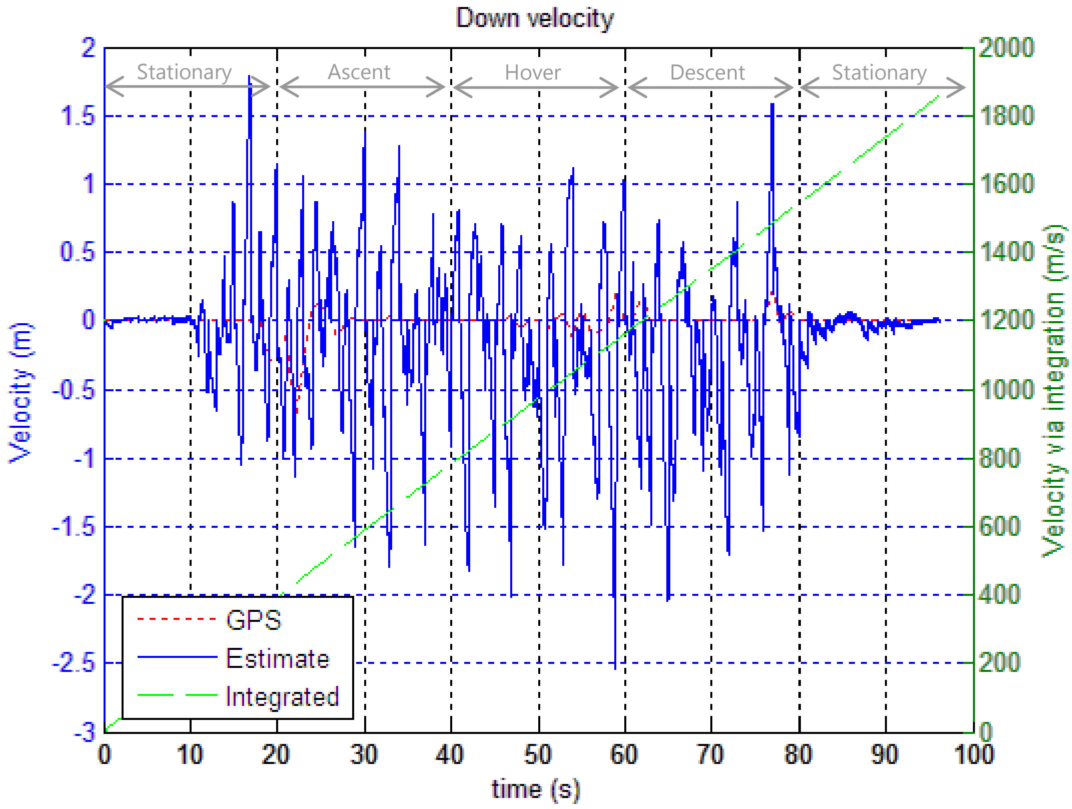


Figure 4.29. Flight test 1: Down velocity results.

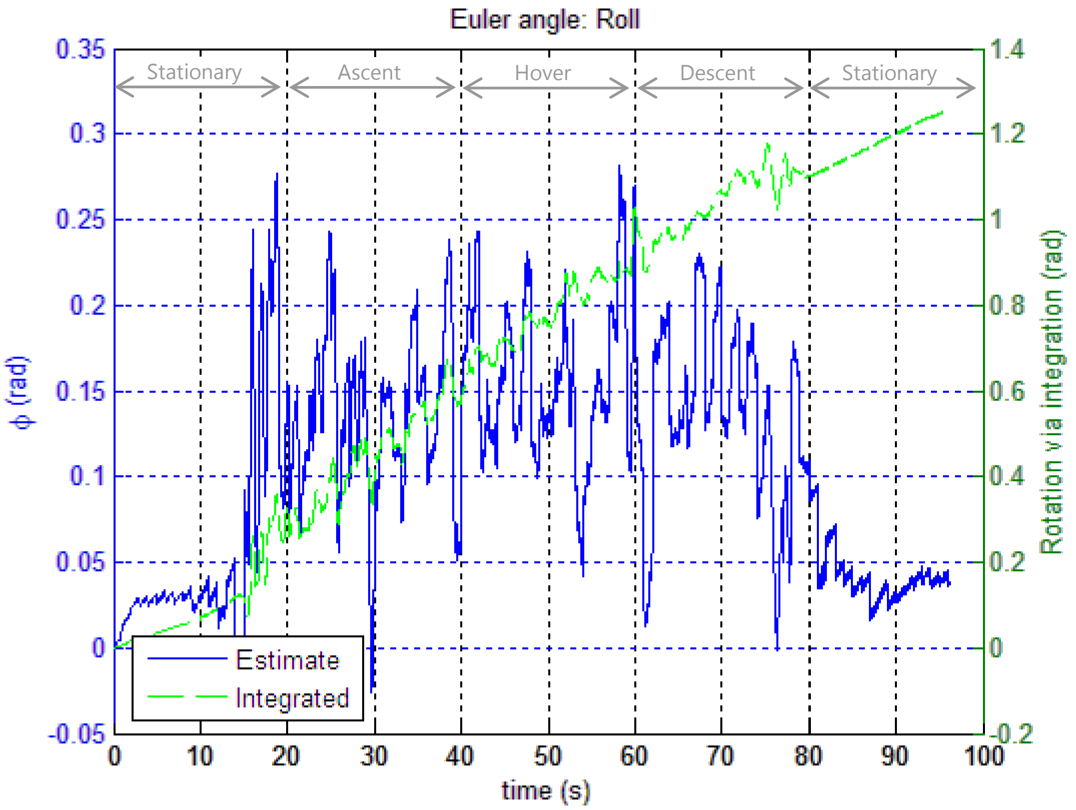


Figure 4.30. Flight test 1: Euler roll results.

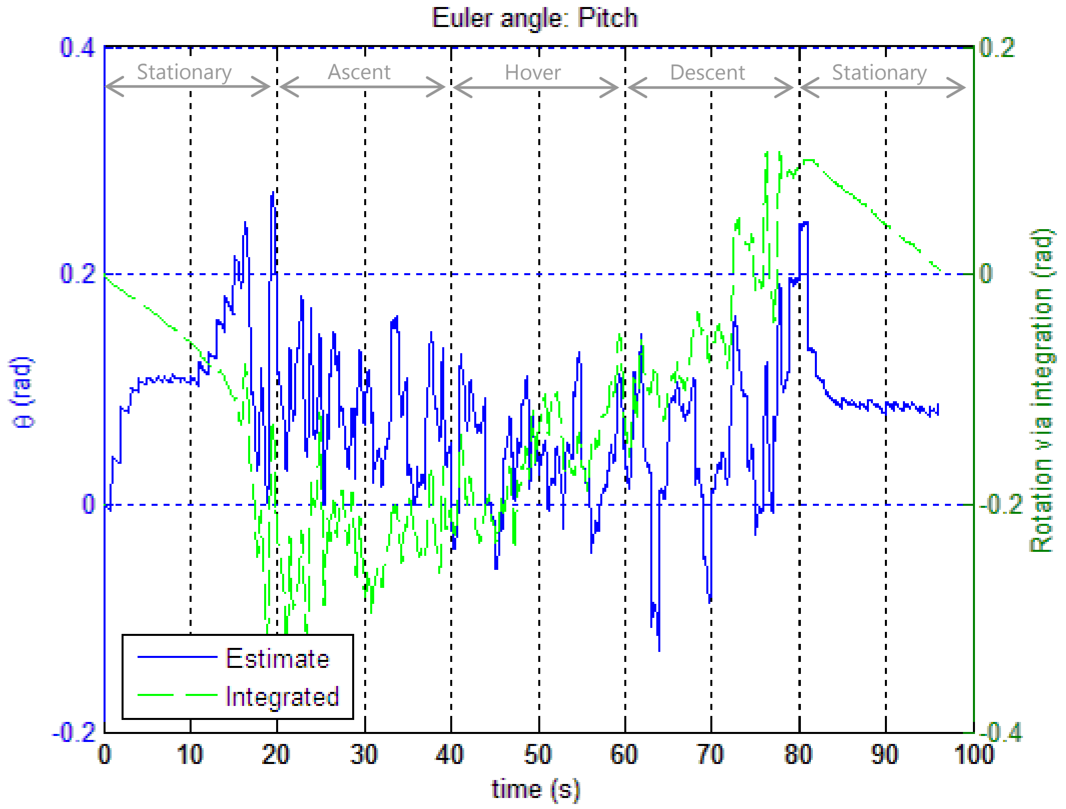


Figure 4.31. Flight test 1: Euler pitch results.

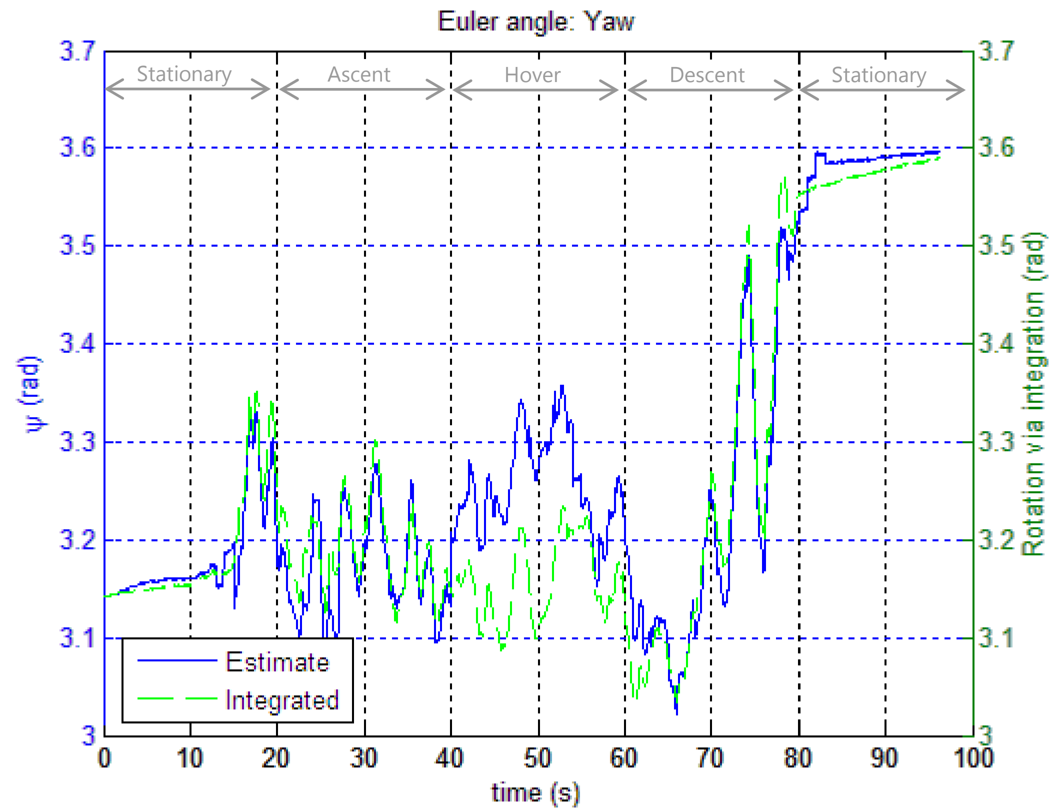


Figure 4.32. Flight test 1: Euler yaw results.

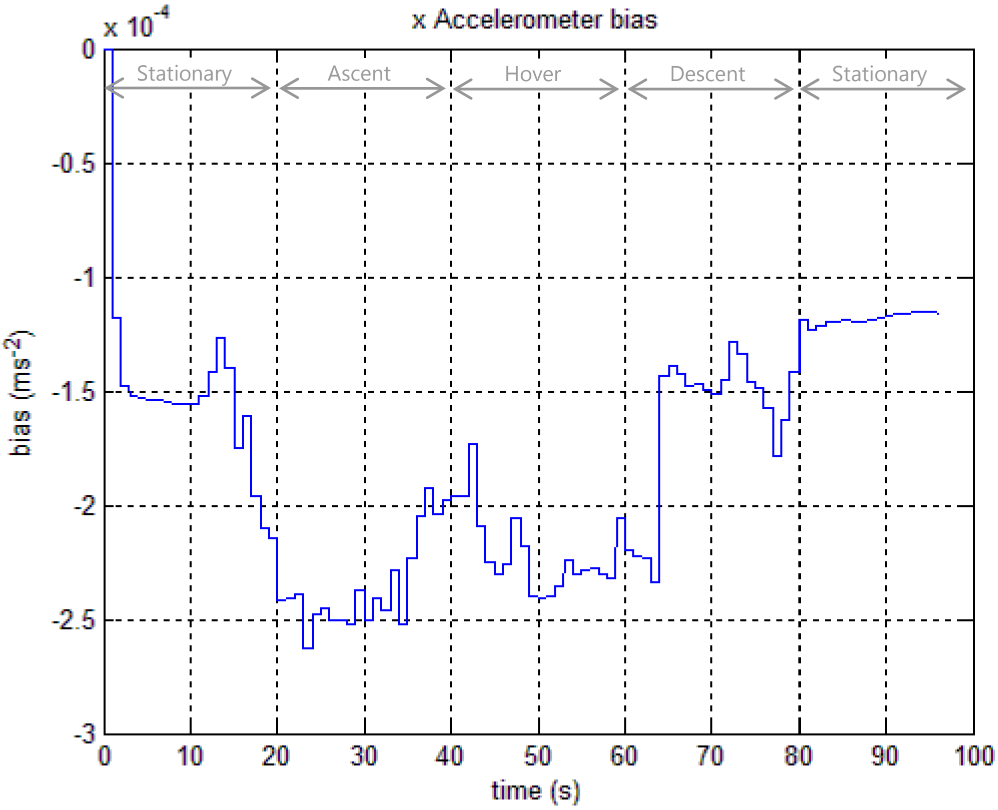


Figure 4.33. Flight test 1: Bias estimate for the x accelerometer.

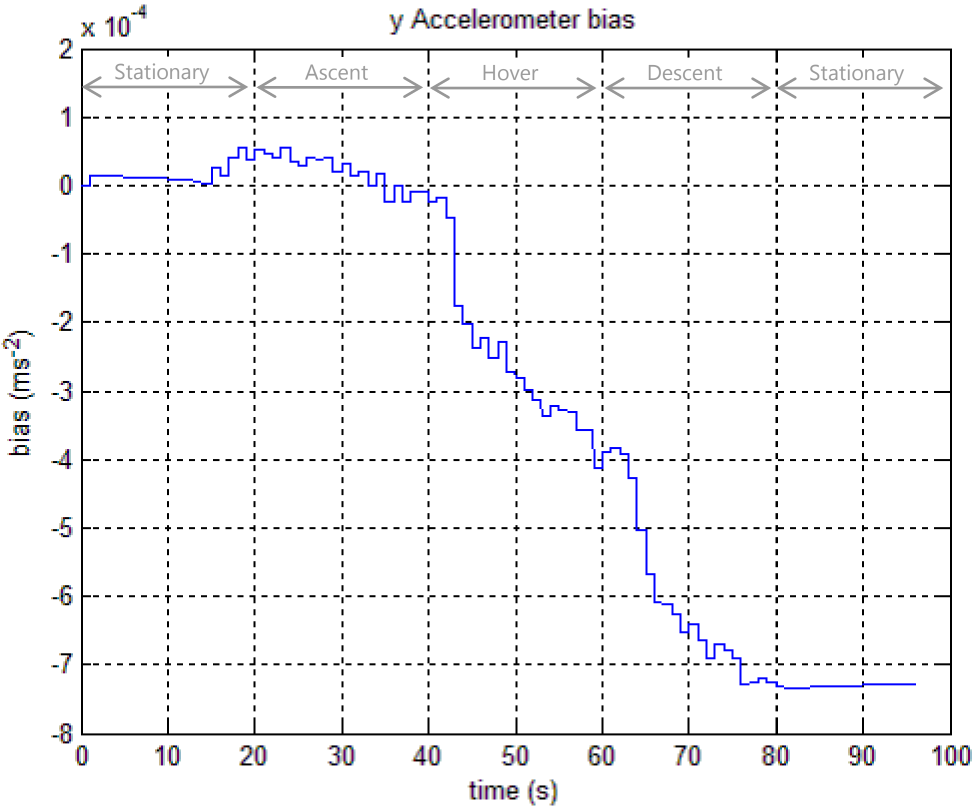


Figure 4.34. Flight test 1: Bias estimate for the y accelerometer.

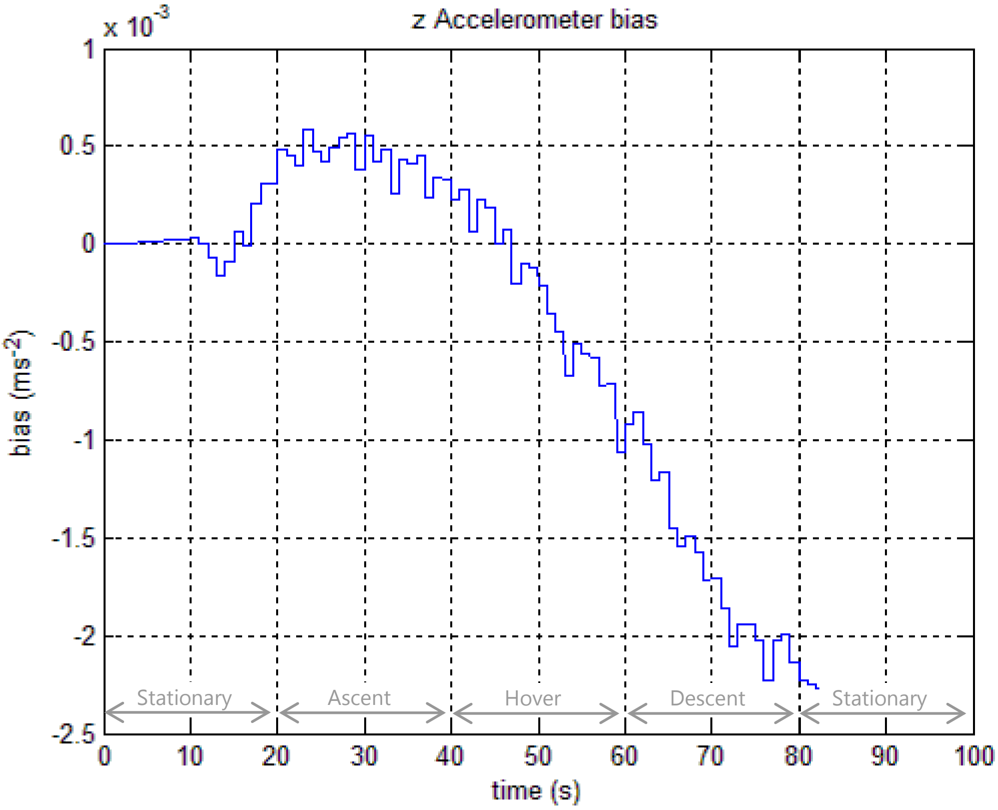


Figure 4.35. Flight test 1: Bias estimate for the z accelerometer.

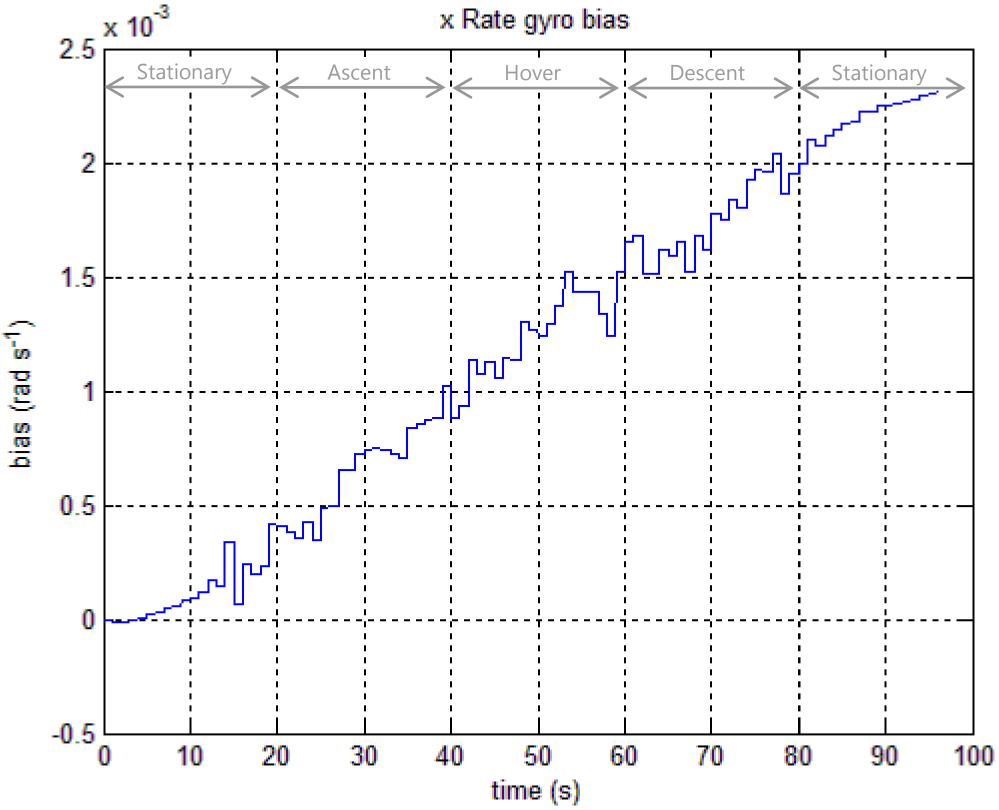


Figure 4.36. Flight test 1: Bias estimate for the x rate gyro.

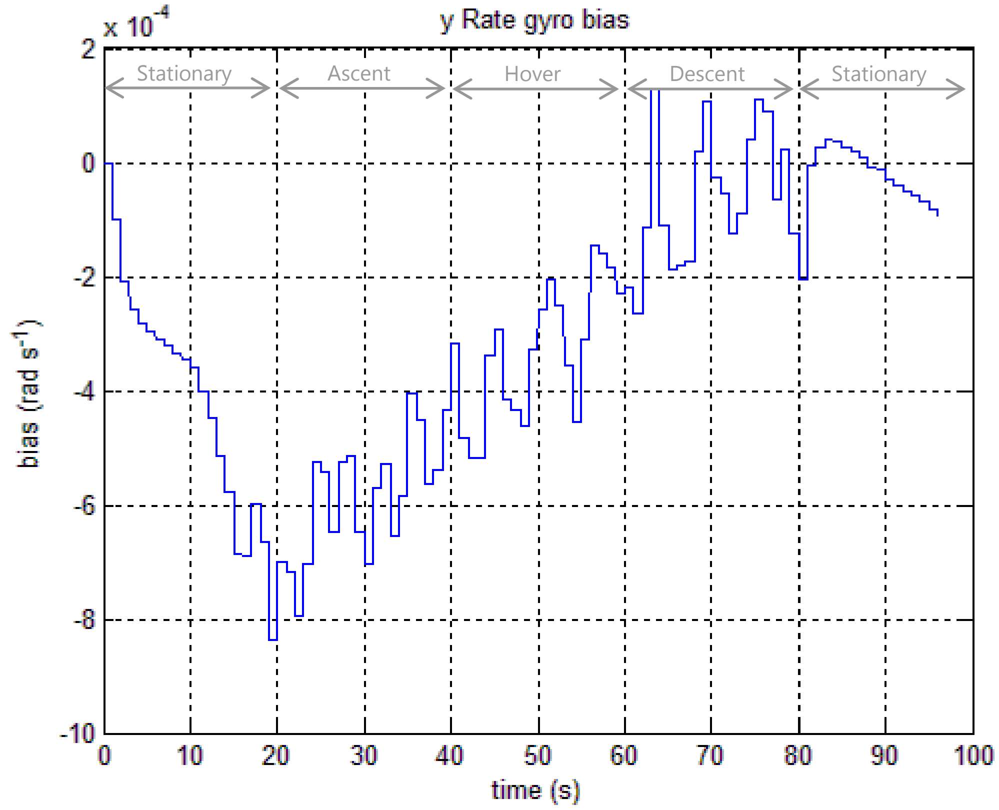


Figure 4.37. Flight test 1: Bias estimate for the y rate gyro.

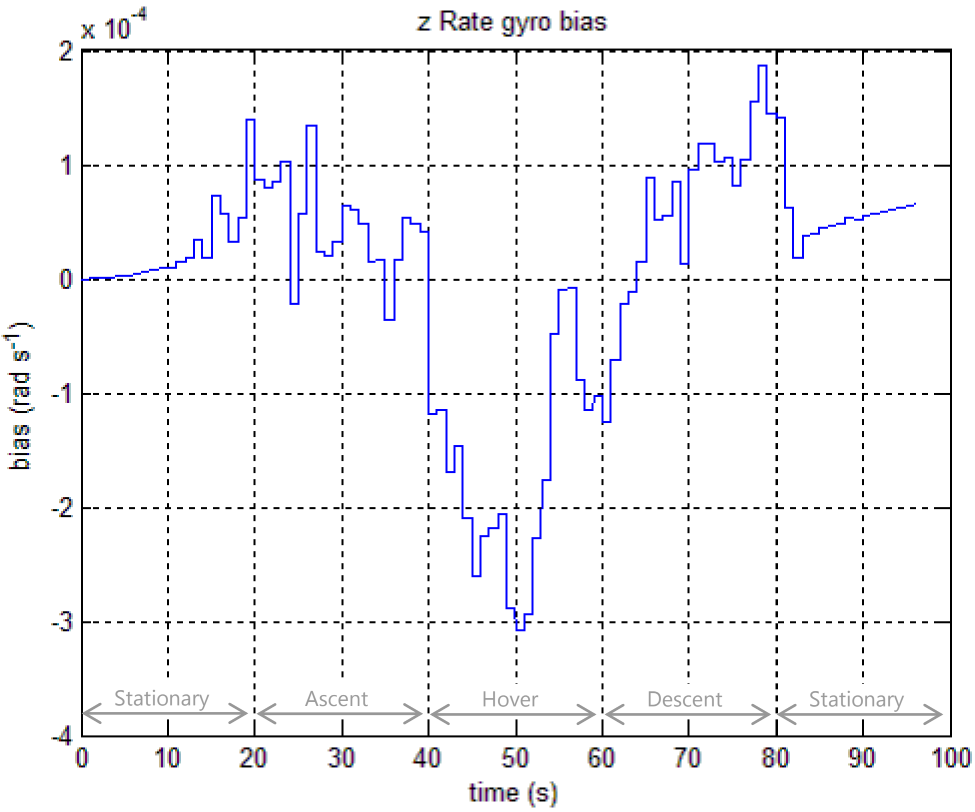


Figure 4.38. Flight test 1: Bias estimate for the z rate gyro.

4.5.5.2.2 Navigation result: Flight test 2

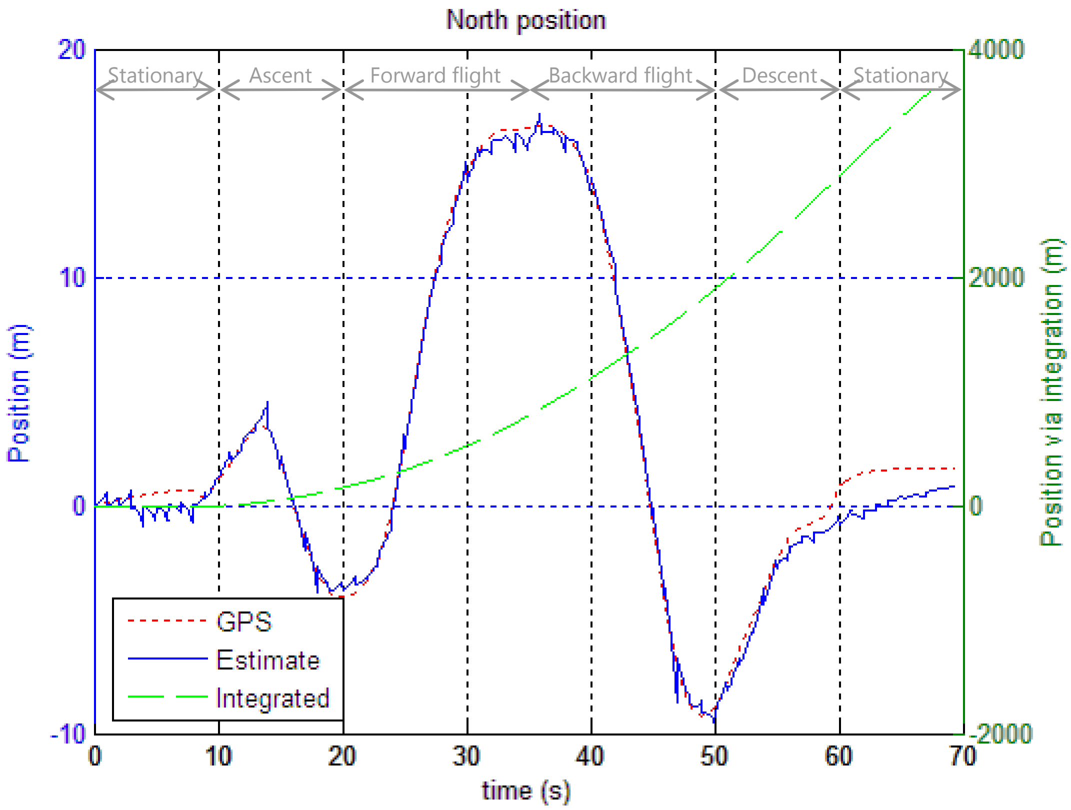


Figure 4.39. Flight test 2: North position results.

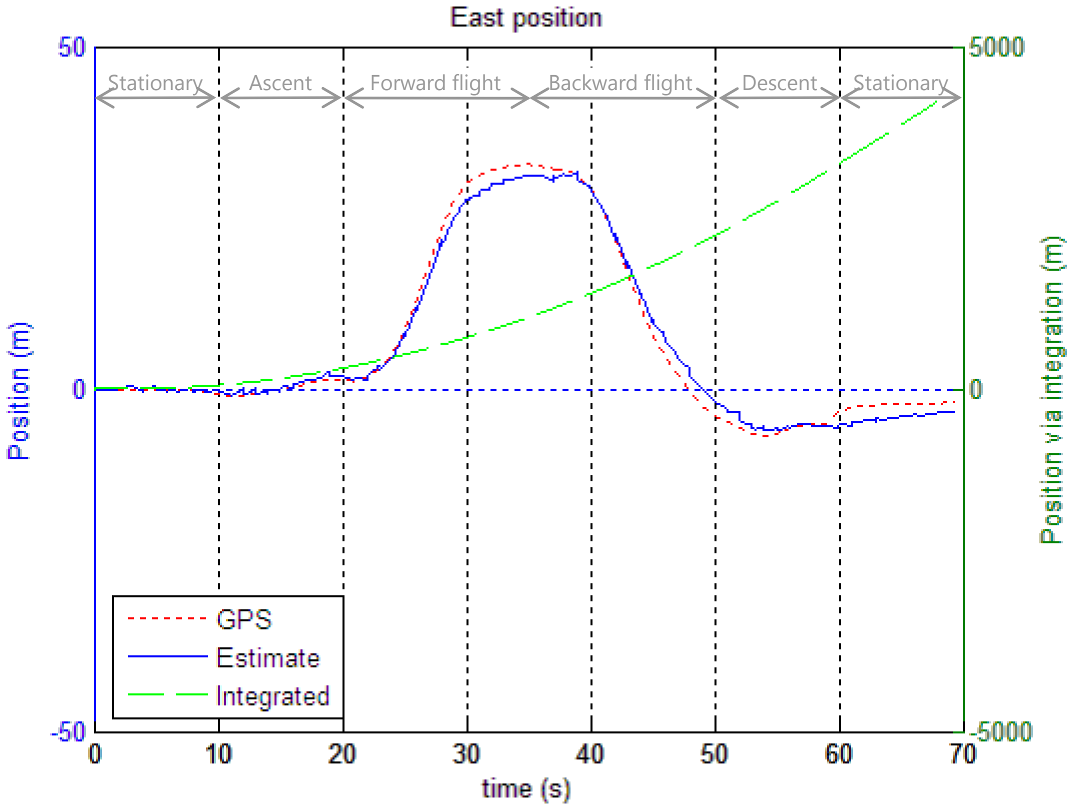


Figure 4.40. Flight test 2: East position results.

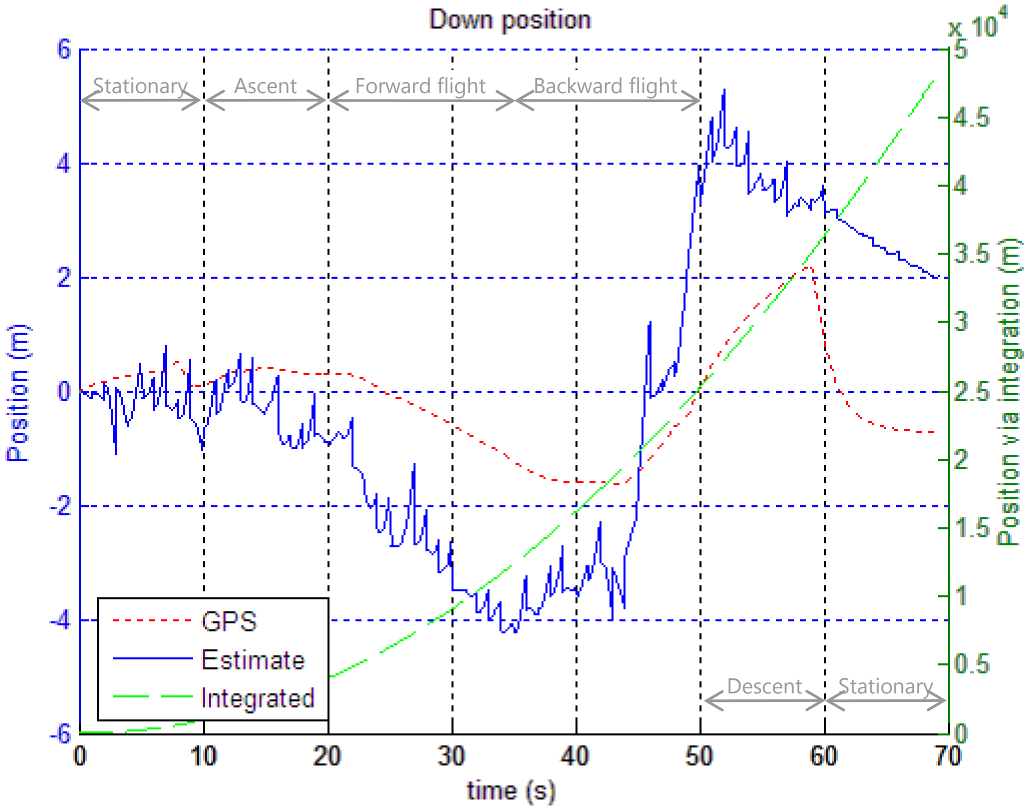


Figure 4.41. Flight test 2: Down position results.

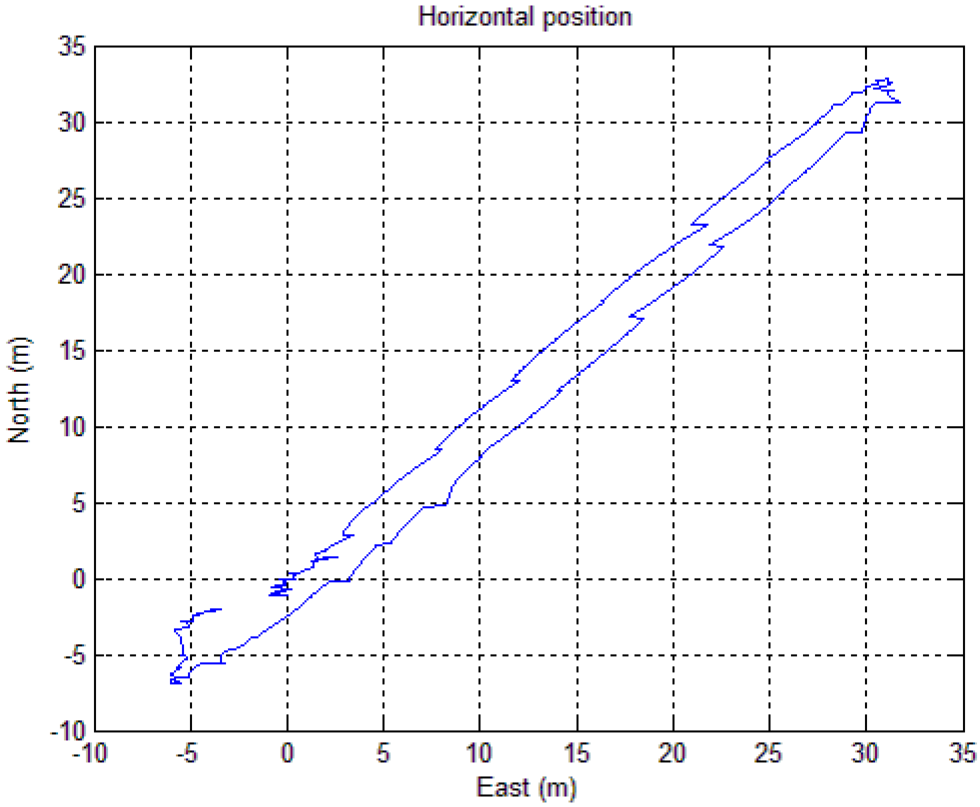


Figure 4.42. Flight test 2: Horizontal position estimate.

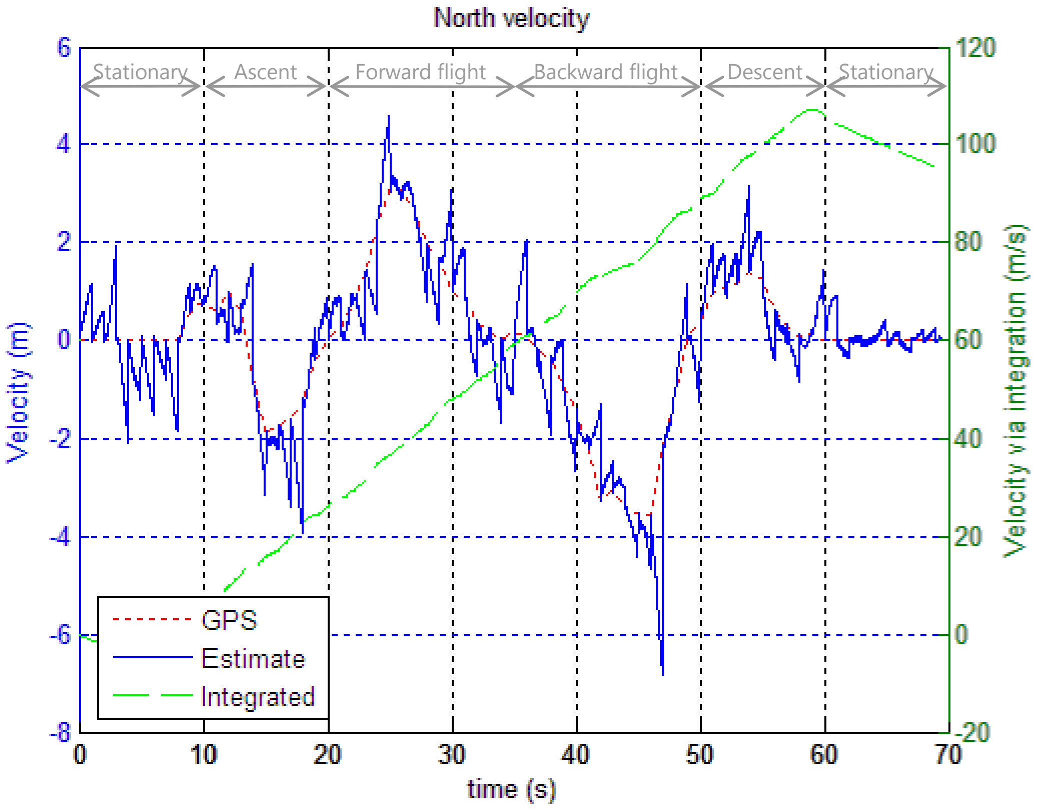


Figure 4.43. Flight test 2: North velocity results.

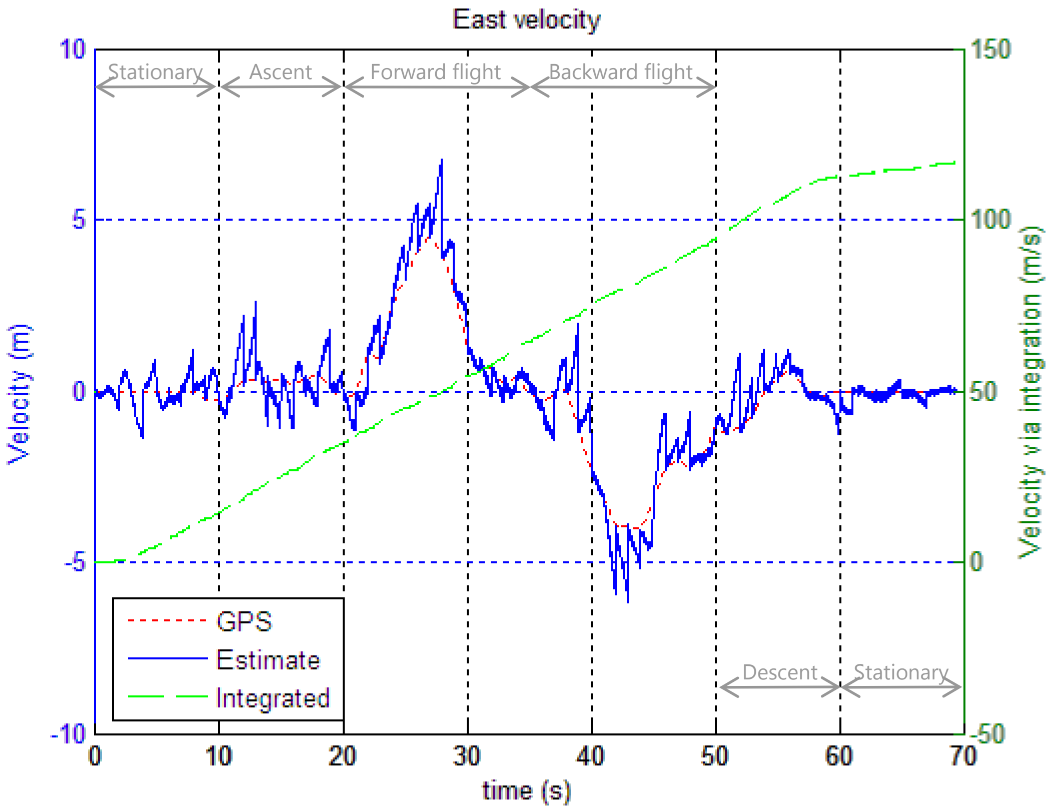


Figure 4.44. Flight test 2: East velocity results.

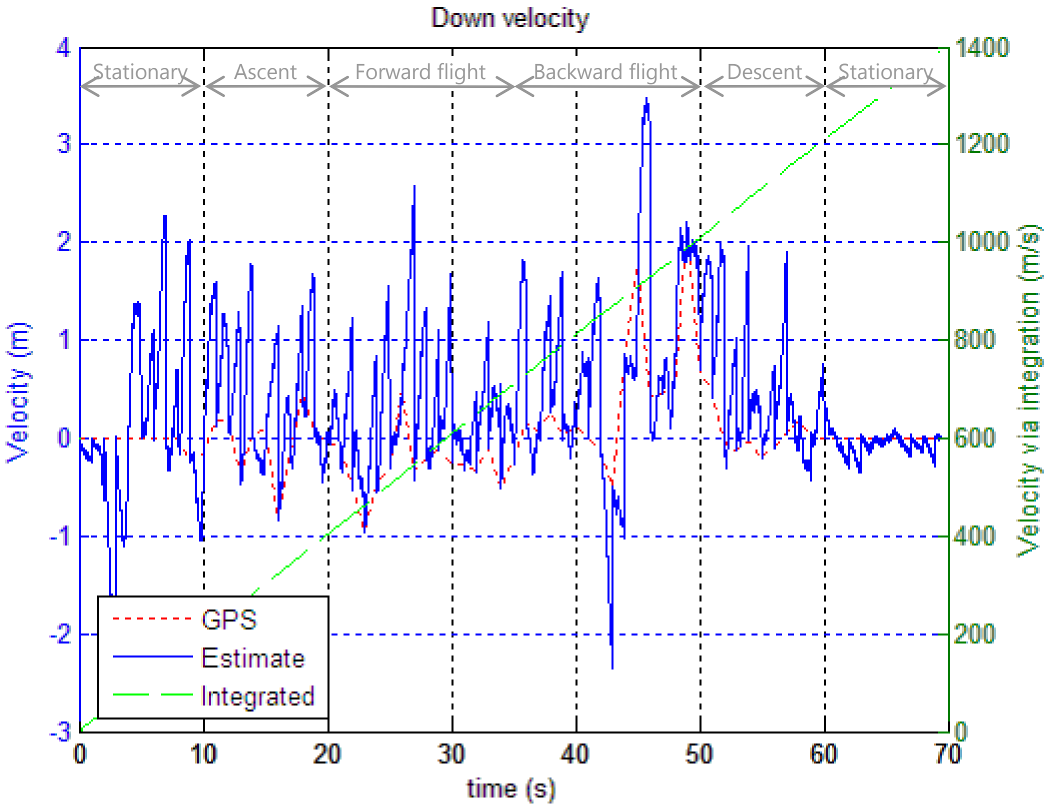


Figure 4.45. Flight test 2: Down velocity results.

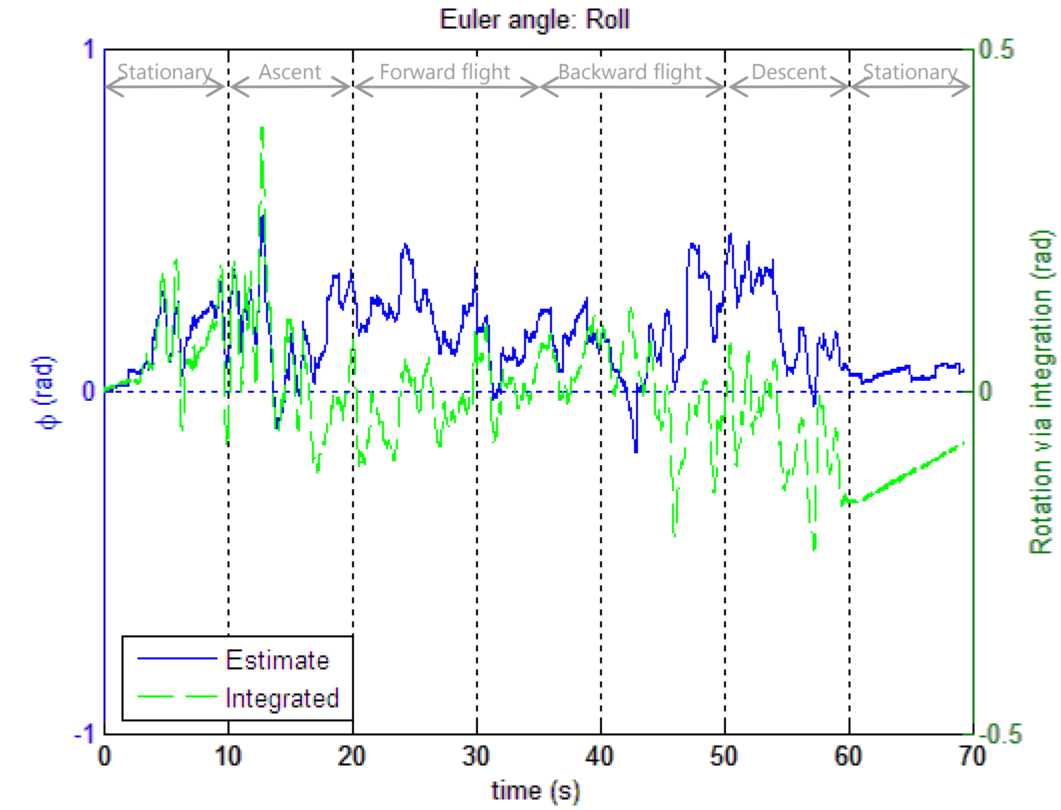


Figure 4.46. Flight test 2: Euler roll results.

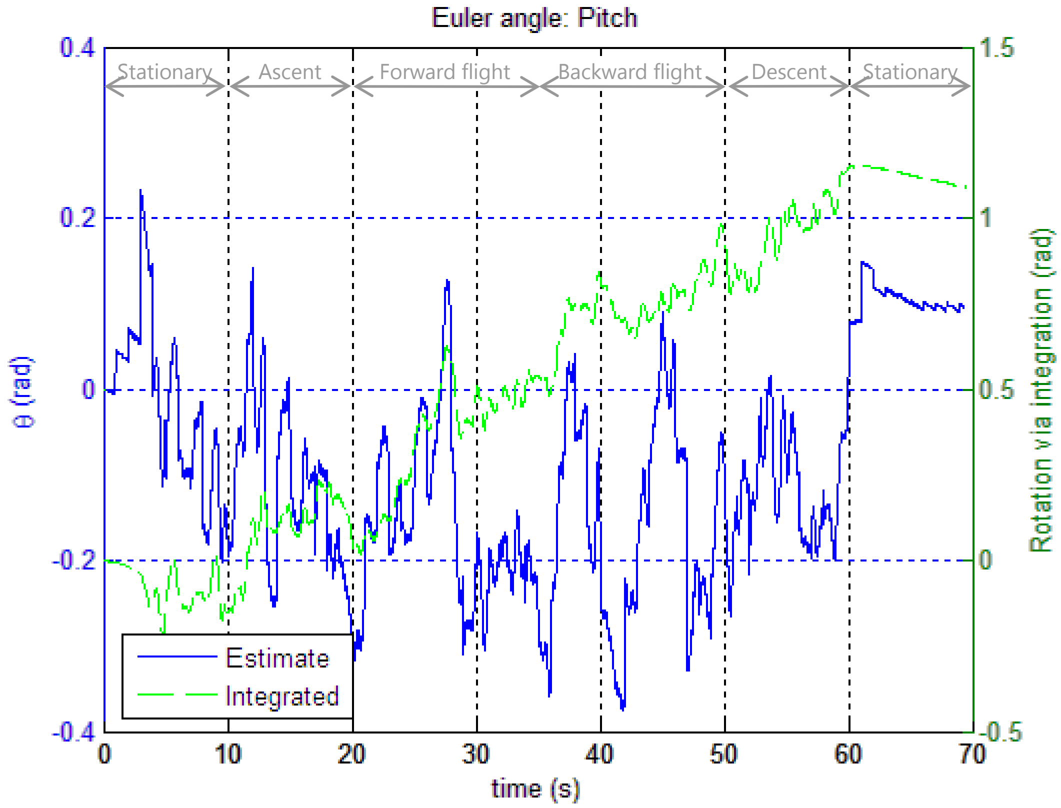


Figure 4.47. Flight test 2: Euler pitch results.

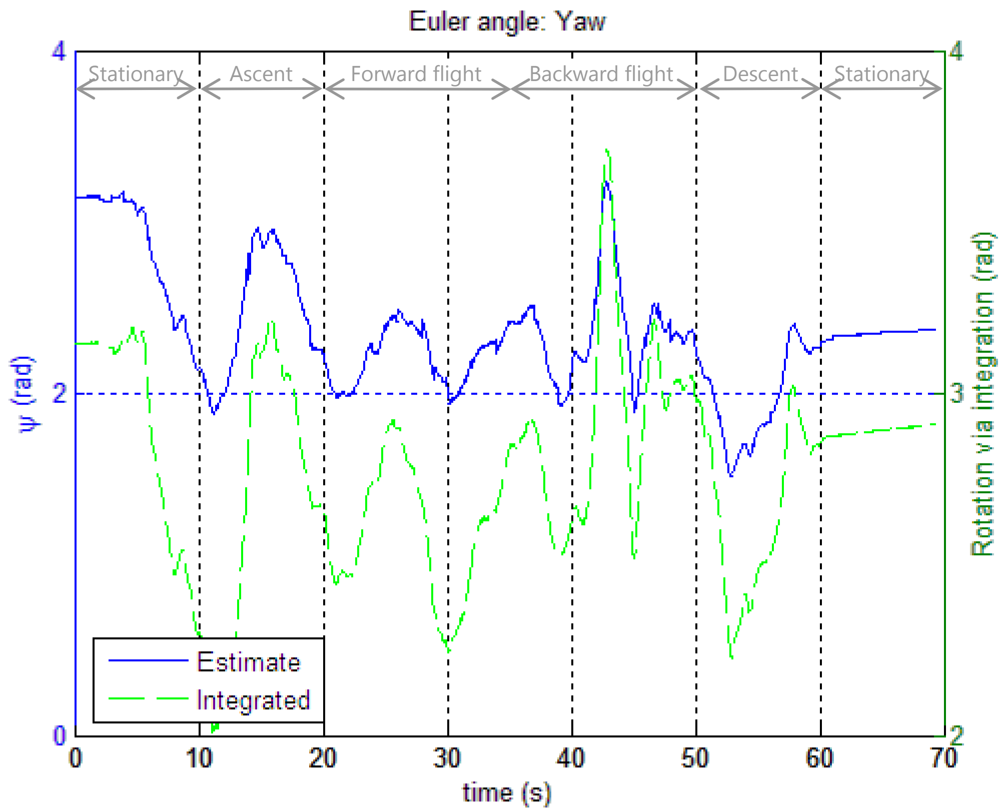


Figure 4.48. Flight test 2: Euler yaw results.

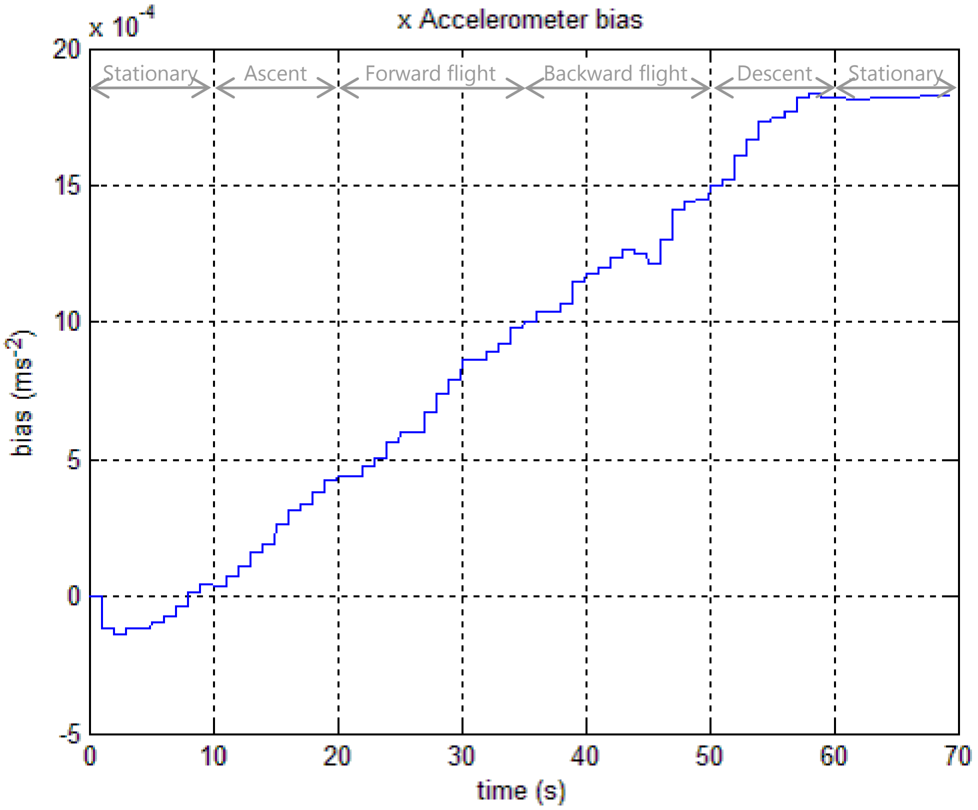


Figure 4.49. Flight test 2: Bias estimate for the x accelerometer.

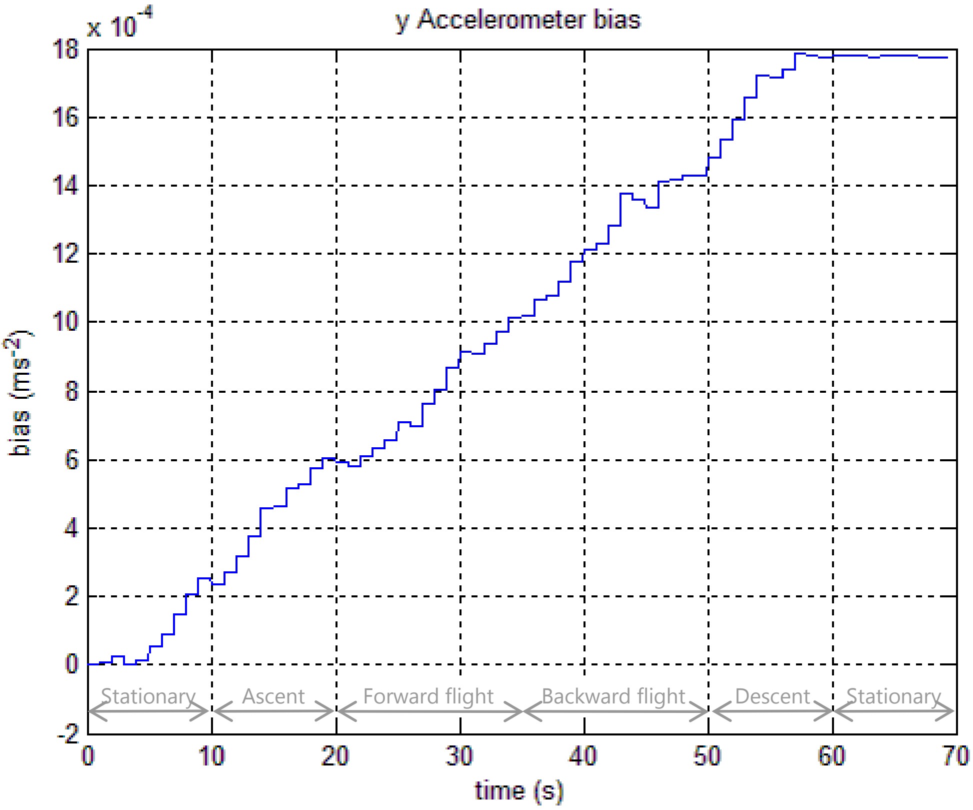


Figure 4.50. Flight test 2: Bias estimate for the y accelerometer.

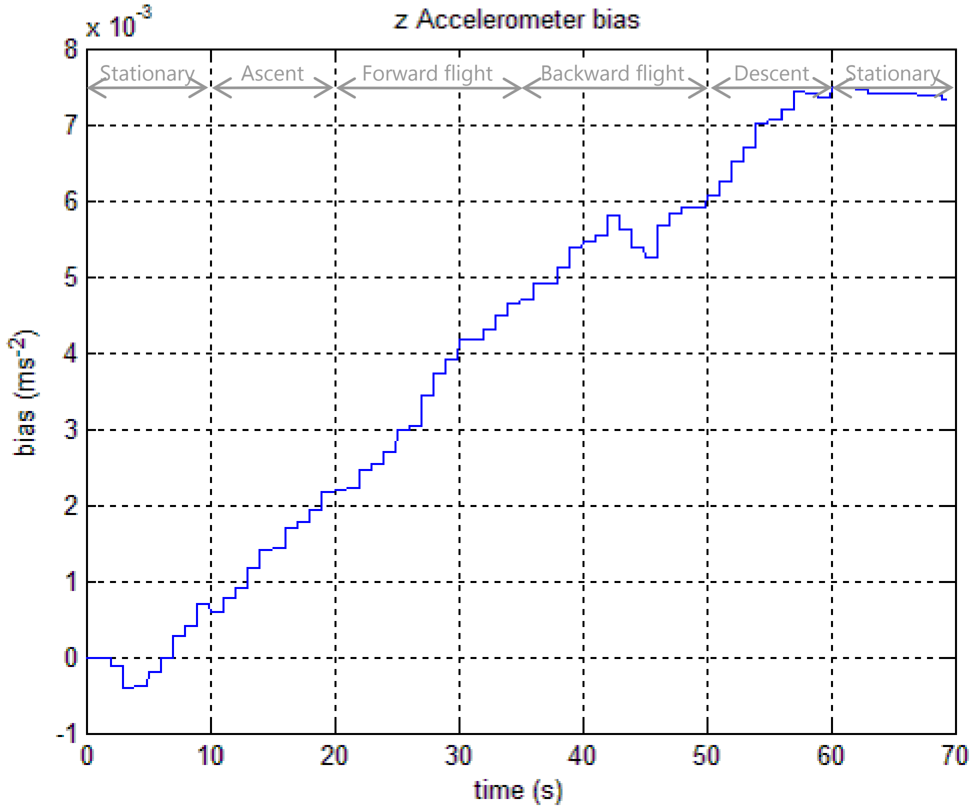


Figure 4.51. Flight test 2: Bias estimate for the z accelerometer.

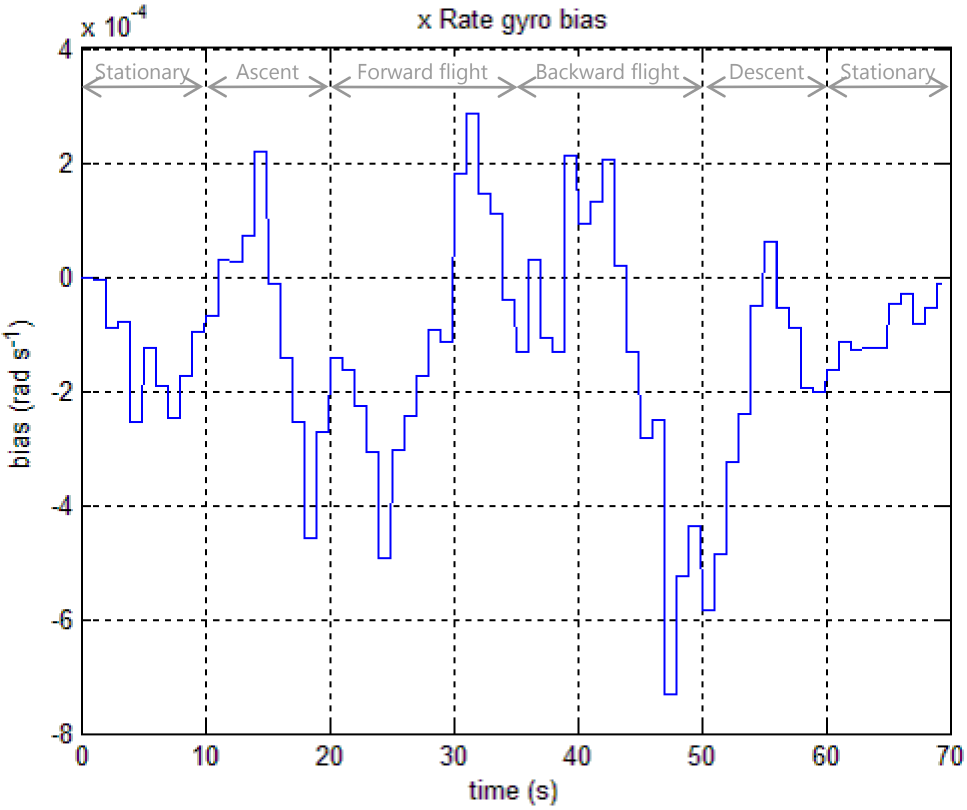


Figure 4.52. Flight test 2: Bias estimate for the x rate gyro.

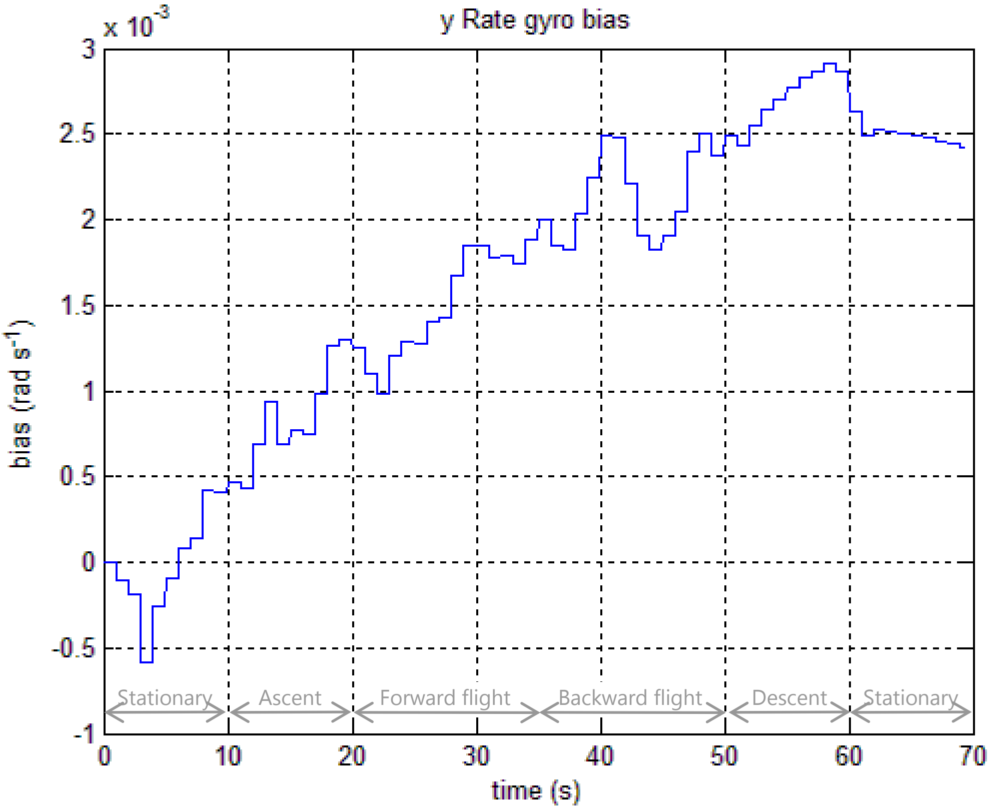


Figure 4.53. Flight test 2: Bias estimate for the y rate gyro.

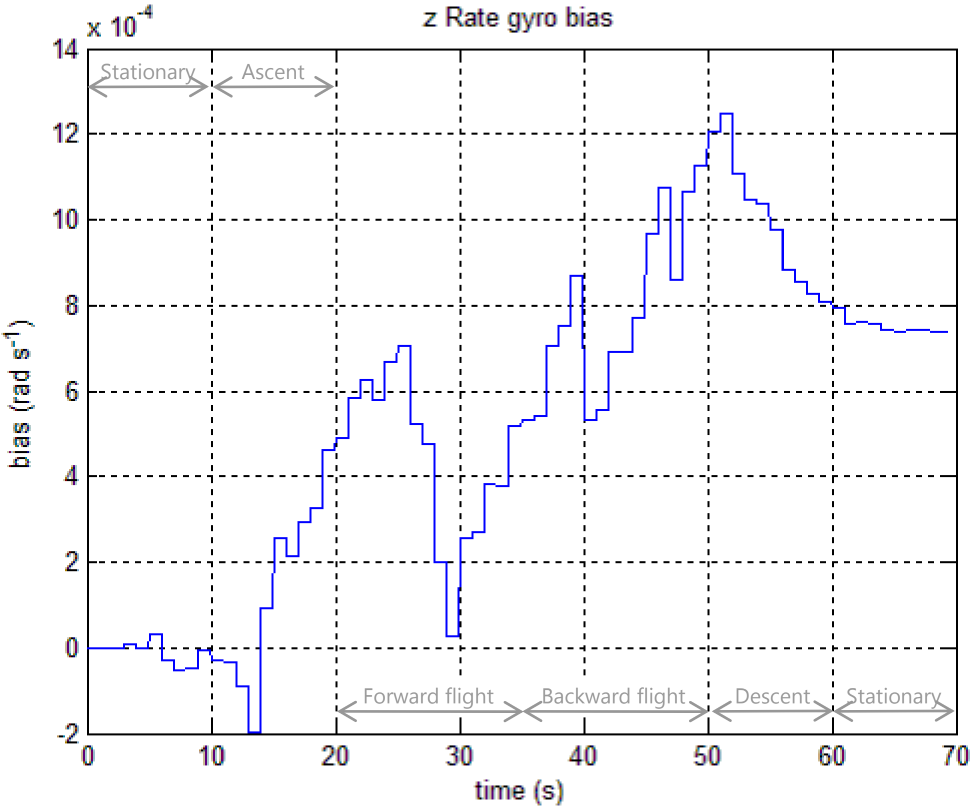


Figure 4.54. Flight test 2: Bias estimate for the z rate gyro.

4.5.5.2.3 Discussion

Position results from both experiments showed estimates that tracked the attempted flight manoeuvres.

For flight test 1, Figure 4.26 is of particular interest illustrating the vertical flight and approximately 20 seconds of hover. In the horizontal plane, deviations from zero, likely associated with imperfect pilot control, were recorded. The helicopter however, did return to within 1 metre of the starting point.

In flight test 2, the forward flight pattern is clearly shown in Figure 4.42, with the down position channel (Figure 4.41) showing take off and altitude during flight as well as dubious GPS samples causing the estimate to become poor following landing (beyond about 50 seconds).

The velocity plots for both tests indicate a strong following of GPS velocity data. This was expected due to the low GPS velocity measurement error covariance. Pitch and roll estimates are difficult to validate, however, due to the benign nature of manoeuvres attempted, the expected maximum/minimum pitch and roll was no more that 30° (≈ ±0.5 rad). Figures above show that this was not exceeded.

For both flights, the estimates indicate periods of prediction (e.g. for position characterised by diverging solution similar to the integrated solution) and correction (measurement update arrives bounding growth). The corrections appear critical to obtaining sensible estimates as the navigation system cannot be run with IMU data only, even for a few seconds. This stems possibly from the significant vibration experienced with the helicopter engine running and, naturally, from inaccuracies in the IMU. Unfortuantely, these cannot be low pass filtered out of the solution, as errors propagate non linearly through the model. It should also be apparent that integration of IMU data alone is completely insufficient when trying to produce the navigation solution e.g. Position and velocity solutions from integration feature quadratic and linear growth (symptomatic of sensor bias) even with the attempted bias compensation via the bias states.

More detailed analysis of the filter is given when considering the embedded test flight results. At this juncture, with respect to confirming reasonable filter operation, flight tests 1 and 2 provided sufficient positive results to allow embedded filter development and testing.

4.5.6 Embedded implementation

4.5.6.1 Matrix operations

The Matlab filter solution is implemented on the DSP in C. Such embedded deployment required that certain Matlab native matrix functions were custom written (or derived from the sources mentioned below) for the DSP Kalman filter. The basic matrix operations necessary for the functioning of the Kalman filter in C are:

1. Matrix assignment.
2. Matrix addition.
3. Matrix subtraction.
4. Matrix scaling.
5. Matrix transpose.
6. Matrix multiplication.
7. Matrix inverse.

The operations mentioned in Points 1 to 5 above are fairly simple to implement in C. These are based heavily on element wise arithmetic operators (standard C operators) on floating point elements of the input matrices. Table 4.3 provides a brief summary of each sub routine implementing Points 1 to 5.

Table 4.3. Basic matrix operation descriptions.

Operation	**Matrix assignment**
Description	The elements of an input matrix are assigned to the elements of an output matrix.
Inputs	Input matrix, $\mathbf{A}$, and input matrix dimensions, rCount and cCount.
Outputs	Output matrix, $\mathbf{B}$.
Pseudo code	for each row in $\mathbf{A}$, index j, up to rCount for each column in $\mathbf{A}$, index k, up to cCount $\mathbf{B}_{jk} = \mathbf{A}_{jk}$
Operation	**Matrix addition/subtraction**
Description	The sums/differences of corresponding elements of input matrices are computed.
Inputs	Input matrices, $\mathbf{A}$ and $\mathbf{B}$, and input matrix dimensions, rCount and cCount.
Outputs	Output matrix, $\mathbf{C}$.
Pseudo code	for each row in $\mathbf{A}$, index j, up to rCount for each column in $\mathbf{A}$, index k, up to cCount $\mathbf{C}_{jk} = \mathbf{A}_{jk} + \mathbf{B}_{jk}$ (for subtraction $\mathbf{C}_{jk} = \mathbf{A}_{jk} - \mathbf{B}_{jk}$)

(Table 4.3 continued.)

Operation	**Matrix scaling**
Description	The elements of an input matrix are scaled by a given constant.
Inputs	Input matrix, $\mathbf{A}$, input matrix dimensions, rCount and cCount, and scaling factor, c.
Outputs	Output matrix, $\mathbf{A}$.
Pseudo code	for each row in $\mathbf{A}$, index j, up to rCount for each column in $\mathbf{A}$, index k, up to cCount $\mathbf{B}_{jk} = c \times \mathbf{A}_{jk}$
Operation	**Matrix transpose**
Description	The rows of an input matrix are swapped with the columns.
Inputs	Input matrix, $\mathbf{A}$, and input matrix dimensions, rCount and cCount.
Outputs	Output matrix, $\mathbf{B}$.
Pseudo code	for each row in $\mathbf{A}$, index j, up to rCount for each column in $\mathbf{A}$, index k, up to cCount $\mathbf{B}_{kj} = \mathbf{A}_{jk}$

Matrix multiplication is also implemented using the standard method i.e. The product, $\mathbf{C}$, of two matrices, $\mathbf{A}$ and $\mathbf{B}$, is constructed such that the element $\mathbf{C}_{jk}$ is determined by computing the scalar ("dot") product of the jth row of $\mathbf{A}$ with the kth column of $\mathbf{B}$. The pseudo code is presented in Listing 4.1 where: (1) n is the row dimension of $\mathbf{A}$; (2) m is the common inner dimension (column count of $\mathbf{A}$ and row count of $\mathbf{B}$), and; (3) p is the column dimension of $\mathbf{B}$.

Listing 4.1. Pseudo code for matrix multiplication.

```
for each row in $\mathbf{A}$, index j, up to n
    for each column in $\mathbf{B}$, index k, up to p
        $\mathbf{C}_{jk} = 0$
        for each column in $\mathbf{A}$, index i, up to m
            $\mathbf{C}_{jk} = \mathbf{C}_{jk} + \mathbf{A}_{ji} \times \mathbf{B}_{ik}$
```

Matrix inversion is implemented using Gaussian elimination with back substitution. The method, from Gerald and Wheatley (1999), accepts as inputs the matrix to be inverted, $\mathbf{A}$, and one parameter defining the dimension of the matrix (invertible matrices are square, thus only one dimension is needed), n. Pseudo code Listing 4.2 illustrates the method.

Listing 4.2. Matrix inversion by Gaussian elimination with back substitution (Gerald and Wheatley, 1999)

```
(Initialise nxn identity matrix that will form matrix where the columns are vectors of RHS's to be used in multiple solutions of $\mathbf{Ax}=\mathbf{b}_k$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \dots\ \mathbf{b}_n]$)
$\mathbf{B} = \mathbf{0}$
for each column in $\mathbf{B}$, index k, up to n
    $\mathbf{B}_{kk} = 1$

for each row in $\mathbf{A}$, index j, up to (n-1)
    m = maximum value from absolute values of row j in matrix $\mathbf{A}$
    pvt = absolute value of $\mathbf{A}_{jj}/m$
    $pivot_j = j$
    t_pvt = j

    (Find pivot row)
    for each row of $\mathbf{A}$, index i, from (j + 1) to n
        if absolute value of $\mathbf{A}_{ij} > pvt$
```

```
            pvt = absolute value of $\mathbf{A}_{ij}$
            t_pvt = i
        end if
    end for i

    (If required switch rows)
    if pivot$_j$ ≠ t_pvt
        Interchange rows j and t_pvt of matrix $\mathbf{A}$

    (Store multipliers)
    for each row of $\mathbf{A}$, index i, from (j + 1) to n
        $\mathbf{A}_{ij} = \mathbf{A}_{ij}/\mathbf{A}_{jj}$

    (Create zeros below the main diagonal)
    for each row in $\mathbf{A}$, index i, from (j+1) to n
        for each column in $\mathbf{A}$, index k, from (j+1) to n
            $\mathbf{A}_{ik} = \mathbf{A}_{ik} - \mathbf{A}_{ij} \times \mathbf{A}_{jk}$

    for each column in $\mathbf{B}$, index k, up to n
        $\mathbf{B}_{ik} = \mathbf{B}_{ik} - \mathbf{A}_{ij} \times \mathbf{B}_{jk}$

    end for i
end for j

(Back substitution)
for each column in $\mathbf{B}$, index c, up to n (for each RHS in $\mathbf{B}$ find solution and store in column of $\mathbf{A}^{-1}$)
    $\mathbf{A}^{-1}{}_{nc} = \mathbf{B}_{nc}/\mathbf{A}_{nn}$
    for each row in $\mathbf{A}^{-1}$, index j, from (n-1) down to 1
        $\mathbf{A}^{-1}{}_{jc} = \mathbf{B}_{jc}$
        for  each column in $\mathbf{A}^{-1}$, index k, from (n) to (j+1)
            $\mathbf{A}^{-1}{}_{jc} = \mathbf{A}^{-1}{}_{jc} - \mathbf{A}^{-1}{}_{kc} * \mathbf{A}_{jk}$

        $\mathbf{A}^{-1}{}_{jc} = \mathbf{A}^{-1}{}_{jc}/\mathbf{A}_{jj}$
    end for j
end for c
```

It should be appreciated that the method presented in Listing 4.2 does not address the problem of rank deficiency in **A**. As the determinant of **A** arises efficiently as a result of the Gaussian elimination technique (Gerald and Wheatley, 1999), it may be prudent to to test the absolute value of this determinant, which is zero if rank deficiency is present. This could at least allow the code to detect problems with the invertibility of **A**. As an alternate to the online determinant check, Section 4.5.6.3 presents, briefly, a condition number based analysis that was conducted, offline, on the test data.

The sub routines described above were initially tested by writing them in Matlab using only scalar operations (operations present in C also). These Matlab versions were used to validate the correct functioning of the developed matrix operations. Following which, the Matlab functions were used as a "template" to aid coding in C on the DSP. Appendix 3 provides the C code.

4.5.6.2 Filter alignment

Section 4.5.3.3 mentions that the filter alignment procedure for determining initial roll, pitch and yaw requires averaging and a variance calculation. To achieve a reasonable result from this process, a relatively large number of samples needs be averaged. Up to five minutes of samples is desired, with the exact number being configurable from the ground side PC.

The simplest averaging and variance strategies involve collecting all relevant samples and then performing expectation and variance calculations. However, for the desired maximum number of samples (five minutes or 9 600 samples), assuming four quantities are required (roll, pitch, yaw and gravity), 150 KB of memory is required (for IEEE 754 four byte floating point numbers). This exceeds the memory capacity of the DSP.

Welford (1962) suggests a solution where a method for computing running means and covariances is presented. Figure 4.55 shows the implementation of this method in the embedded navigation application (Green highlighting indicates algorithmic elements from Welford, 1962).

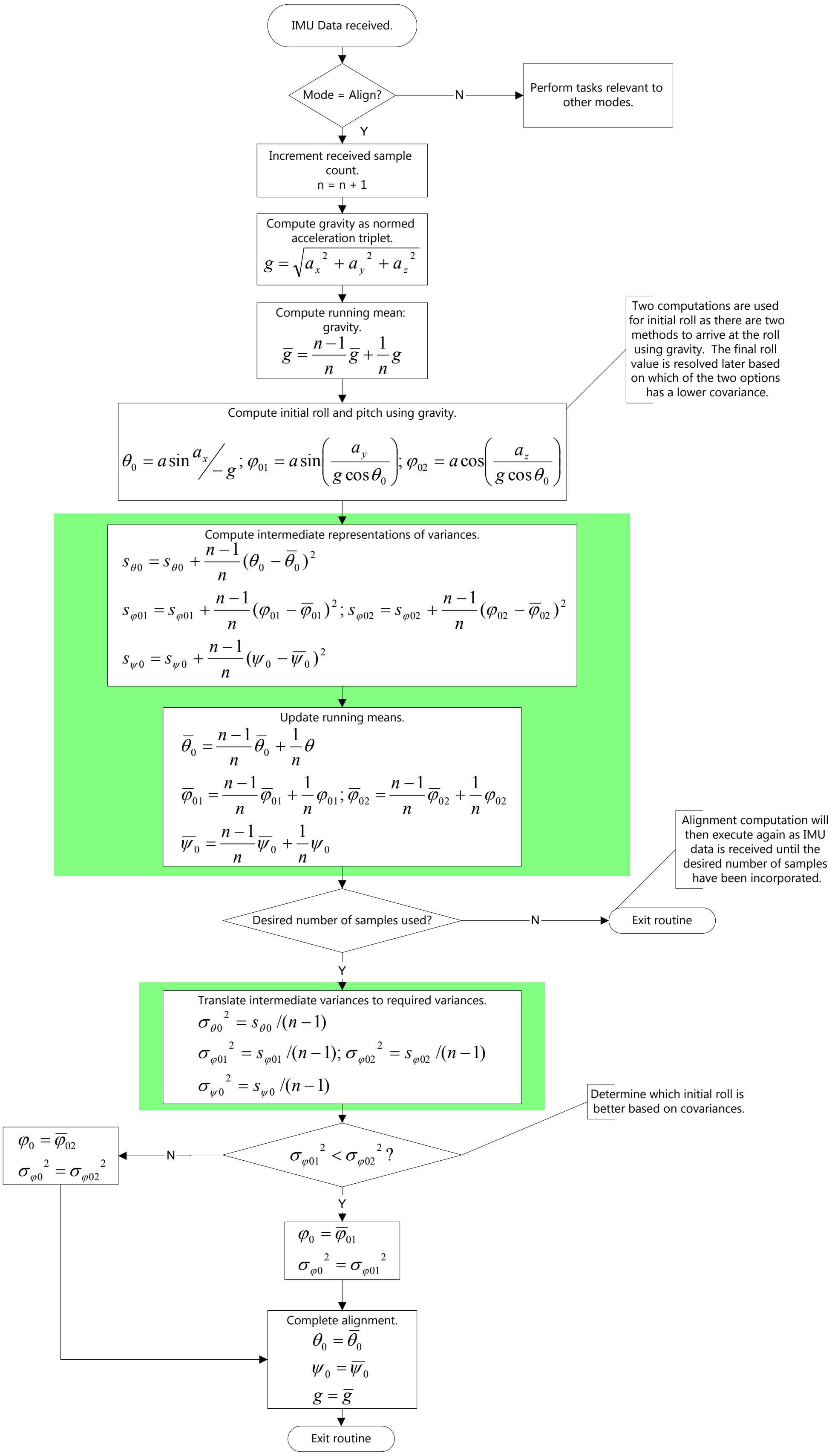

IMU Data received.
Mode = Align?
N
Perform tasks relevant to other modes.
Y
Increment received sample count.
n = n + 1
Compute gravity as normed acceleration triplet.
$g=\sqrt{a_x^2+a_y^2+a_z^2}$
Compute running mean: gravity.
$\overline{g}=\frac{n-1}{n}\overline{g}+\frac{1}{n}g$
Two computations are used for initial roll as there are two methods to arrive at the roll using gravity. The final roll value is resolved later based on which of the two options has a lower covariance.
Compute initial roll and pitch using gravity.
$\theta_0=a\sin{}^{a_x}\!/_{-g};\ \varphi_{01}=a\sin\left(\frac{a_y}{g\cos\theta_0}\right);\ \varphi_{02}=a\cos\left(\frac{a_z}{g\cos\theta_0}\right)$
Compute intermediate representations of variances.
$s_{\theta 0}=s_{\theta 0}+\frac{n-1}{n}(\theta_0-\overline{\theta}_0)^2$
$s_{\varphi 01}=s_{\varphi 01}+\frac{n-1}{n}(\varphi_{01}-\overline{\varphi}_{01})^2;\ s_{\varphi 02}=s_{\varphi 02}+\frac{n-1}{n}(\varphi_{02}-\overline{\varphi}_{02})^2$
$s_{\psi 0}=s_{\psi 0}+\frac{n-1}{n}(\psi_0-\overline{\psi}_0)^2$
Update running means.
$\overline{\theta}_0=\frac{n-1}{n}\overline{\theta}_0+\frac{1}{n}\theta$
$\overline{\varphi}_{01}=\frac{n-1}{n}\overline{\varphi}_{01}+\frac{1}{n}\varphi_{01};\ \overline{\varphi}_{02}=\frac{n-1}{n}\overline{\varphi}_{02}+\frac{1}{n}\varphi_{02}$
$\overline{\psi}_0=\frac{n-1}{n}\overline{\psi}_0+\frac{1}{n}\psi_0$
Alignment computation will then execute again as IMU data is received until the desired number of samples have been incorporated.
Desired number of samples used?
N
Exit routine
Y
Translate intermediate variances to required variances.
$\sigma_{\theta 0}{}^2=s_{\theta 0}/(n-1)$
$\sigma_{\varphi 01}{}^2=s_{\varphi 01}/(n-1);\ \sigma_{\varphi 02}{}^2=s_{\varphi 02}/(n-1)$
$\sigma_{\psi 0}{}^2=s_{\psi 0}/(n-1)$
Determine which initial roll is better based on covariances.
$\sigma_{\varphi 01}{}^2<\sigma_{\varphi 02}{}^2\,?$
N
$\varphi_0=\overline{\varphi}_{02}$
$\sigma_{\varphi 0}{}^2=\sigma_{\varphi 02}{}^2$
Y
$\varphi_0=\overline{\varphi}_{01}$
$\sigma_{\varphi 0}{}^2=\sigma_{\varphi 01}{}^2$
Complete alignment.
$\theta_0=\overline{\theta}_0$
$\psi_0=\overline{\psi}_0$
$g=\overline{g}$
Exit routine

Figure 4.55. Alignment routine using method from Welford (1962).

Naturally, several calculation variables (running means, intermediate covariances and received sample counter) must be initialised to zero and this is done when the INS mode is changed to "alignment" (prior to the first execution of the algorithm above). Each execution of the algorithm is driven by the reception of the IMU data, with relevant mode flags in the IMU receive data routine indicating the current purpose of the IMU data – alignment. The routine represented by Figure 4.55 is then executed.

4.5.6.3 Numerical considerations

Grewal and Andrews (2001) list several pitfalls contributing to ill conditioning of any practically implemented digital Kalman filter. The following notes quote these, as well as some project relevant discussion:

1. Large uncertainties in matrices **A, Q, R** and **H**.
   $\mathbf{Q}$ and $\mathbf{R}$ are derived from manufacturer data and $\mathbf{H}$ is known exactly. $\mathbf{A}$, however, is an approximation of the matrix exponent i.e. $\mathbf{A} = \mathbf{I} + \mathbf{T}\Delta t \approx e^{\mathbf{T}\Delta t}$. In this project, the maximum value of the neglected lower order terms was of the order of ≈ 10^{-3} (implying small uncertainty in $\mathbf{A}$).

2. Large ranges of the actual values of matrices, measurements and state variables.
   Matrices employed do not have elements with widely ranging values, except the $\mathbf{Q}$ matrix (elements between 10^{-1} and 10^{-15}) which could have been problematic. Additionally, measurements and state variables are all in the same coordinate frame with similar scales resulting in values within the same range.

3. Ill conditioning of the matrix inversion required for the Kalman gain calculation.
   The conditioning of the inversion was assessed by calculating the condition number of the matrix, $[\mathbf{H_k}\mathbf{P_k^-}\mathbf{H_k}^T + \mathbf{R_k}]$. Press (1992) indicates that if the condition number is large, such that its reciprocal approaches the floating point precision of the embedded system, then the operation is poorly conditioned. Beauregard and Fraleigh (1995) and Wilkinson (1965) provide a brief discourse on conditioning using the $\mathbf{Ax} = \mathbf{b}$ example (simultaneous equations). Essentially, a poorly conditioned system results in solutions being hyper sensitive to small variations in coefficients (**A**) or vector **b**. This creates problems in digital computations where small round off errors can corrupt accurate results. The condition number is usually:

$$\kappa = {}^{\sigma_{\max}}\!/_{\sigma_{\min}} \qquad 4.97$$

   Where $\sigma_{\max/\min}$ is the maximum/minimum singular values of the matrix. For the flight tests presented in Section 4.5.6.4, the condition number of $[\mathbf{H_k}\mathbf{P_k^-}\mathbf{H_k}^T + \mathbf{R_k}]$ was computed offline at each measurement update and found to always be ≤ 70. The DSP uses IEEE 754 single precision floats with machine precision $\varepsilon \approx 10^{-7}$. Since $\kappa^{-1} \geq 10^{-2} >> \varepsilon$, the criterion of Press (1992) for well conditioning is met.

4. Large matrix dimensions can create problems as the number of arithmetic operations increase quadratically or cubically with dimension, thereby increasing the opportunity for round off errors.
   The filtering problem here is of relatively low dimension with no matrices larger than fifteen rows by fifteen columns. Several of the filter matrices (e.g. **R**, **H**, **Q**, **G** etc.) are sparse, and the matrix operation routines for addition and multiplication (Section 4.5.6.1) were adjusted to reduce the number of arithmetic operations. Addition and multiplication

routines where one matrix operand consisted mostly of zero elements, benefited greatly from the use of a binary matrix indicating the location of non zero values. The use of the binary matrix allows the code to skip unnecessary floating point additions and multiplications as the result of these are known.

5. Poor machine precision causing relative round off errors to be large.
The C environment of the DSP supports only IEEE 754 single precision floats. Having a 24 bit mantissa these result in precision of only $\varepsilon \approx 10^{-7}$. This is the best that can be achieved with the system employed.

Several issues raised by Grewal and Andrews (2001) appear to have acceptable behaviour in terms of the navigation solution. Through the embedded filter tests, however, all areas of concern were monitored for pathology.

4.5.6.4 Embedded filter tests

Operation of the embedded Kalman filter, executing on the DSP and helicopter avionics, was tested online and in real time by performing several flight tests. All parameters and initialisation quantities were set or determined to their previously discussed values. Both the barometer and magnetic compass were available and included in the experimental hardware and software. Additionally, all pertinent data, including sensor outputs and DSP calculated filter states, were logged on the ground station, and a video camera was used to graphically record events. Section 4.5.6.4.1 provides details of the flight tests performed on an open sports field.

4.5.6.4.1 Flight test descriptions

Three flight tests, with the patterns depicted in Figure 4.56 to Figure 4.58, were done. Symbols used in these figures are defined in Table 4.4.

Table 4.4. Symbols used in graphical flight test descriptions.

🟩	Flight start point.
🟥	Flight end point.
⟶	Indicates section of flight path to be followed between way points.
x	Way point marker where x represents the number of a way point in a sequential list of points to be achieved.

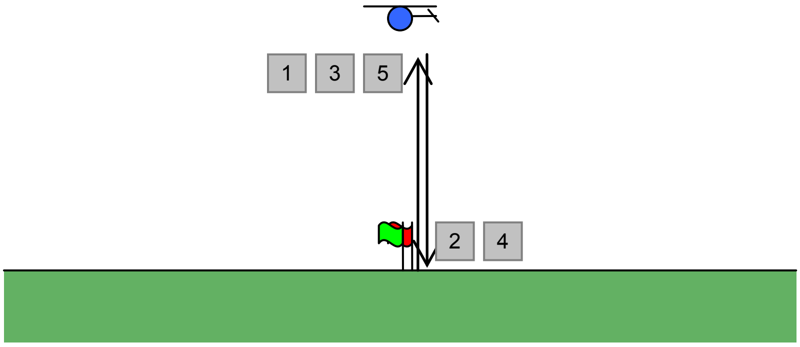


Figure 4.56. Flight test 3: Vertical flight with three ascent/descent pairs.

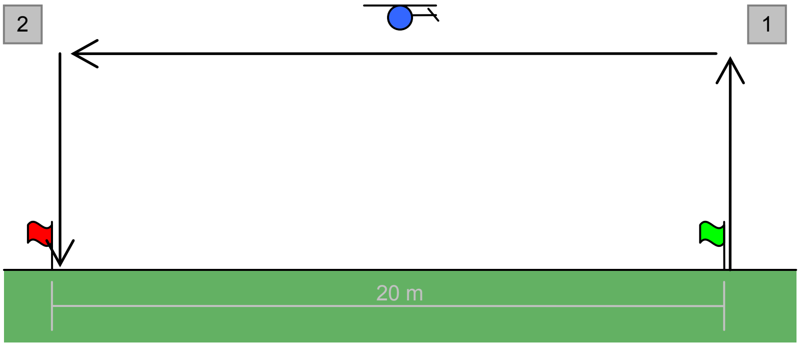


Figure 4.57. Flight test 4: Horizontal flight with no turns along a 20 m line on a bearing of 15°.

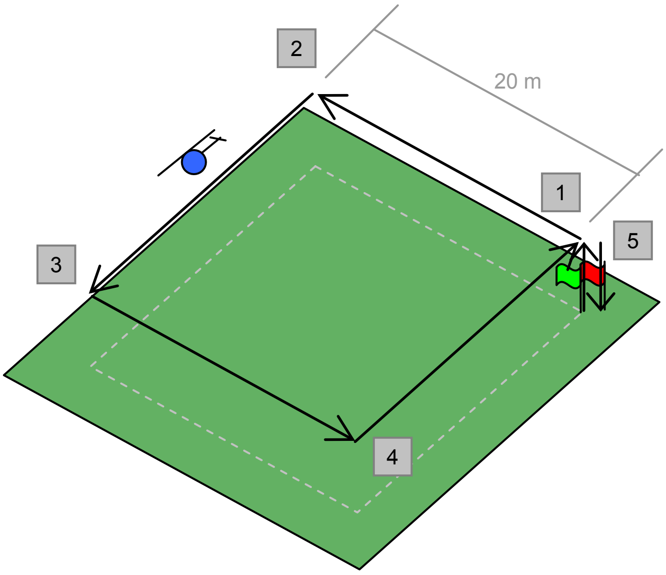


Figure 4.58. Flight test 5: Horizontal flight with turns along a square having 20 m sides. Turns occurred at markers 2, 3 and 4. An initial bearing (between markers 1 and 2) of 15° was attempted.

For all tests, markers were placed on the test field to aid the pilot, however, altitude references are both difficult to provide and follow. As such, validation of the down position estimate was provided by reviewing captures from the video log.

4.5.6.4.2 Navigation result descriptions

The fifteen state variables and associated estimation standard deviations, as computed by the embedded filter, are plotted in Section 4.5.6.4.3 to Section 4.5.6.4.5, below. Symbols from Table 4.4 are used to annotate the graphs. Plots for flight tests 4 and 5 include a horizontal position plot indicating the path followed. Result sets for all flight tests also include a plot of $\|\mathbf{I}-\mathbf{K}_{\mathbf{k}}\mathbf{H}_{\mathbf{k}}\|$ and $\|\mathbf{K}_{\mathbf{k}}\|$ for the duration of the flight where $\|\bullet\|$ is the 2-norm of matrix $\bullet$. This was incorporated in an effort to determine the effect of the filter versus the model i.e. Whether the filter was merely using measurement data or actually blending measurement data with a priori estimates. Plotting the norms, mentioned above, can be useful as the measurement update equation (Equation 4.25) can be arranged as Equation 4.98.

$$\hat{\mathbf{x}}_{\mathbf{k}} = (\mathbf{I}-\mathbf{K}_{\mathbf{k}}\mathbf{H}_{\mathbf{k}})\hat{\mathbf{x}}_{\mathbf{k}}^{-} + \mathbf{K}_{\mathbf{k}}\mathbf{z}_{\mathbf{k}} \qquad 4.98$$

From Equation 4.98, it should be apparent that:

1. $(\mathbf{I}-\mathbf{K_k H_k})$ equal to or near zero would imply that the a priori estimates are being ignored.
2. $(\mathbf{I}-\mathbf{K_k H_k}) << \mathbf{K_k}$ would imply that the measurement is far more strongly represented in the estimate than the a priori estimate. Naturally, $(\mathbf{I}-\mathbf{K_k H_k})$ and $\mathbf{K_k}$ must be multiplied by the relevant signals before going into the estimate, and, as such, comparison of these may not be valid. However, the a priori estimate and measurement are in the same unit, and it is likely that these are in roughly the same numeric range.
3. The quantities of interest from Points (1) and (2) are matrices. Their matrix norms are plotted so as to compactly acquire a sense for the magnitude of the elements of the matrices.

Table 4.5 lists the navigation results presented in Section 4.5.6.4.3 to Section 4.5.6.4.5, below, including relevant figure numbers.

Table 4.5. Navigation result - figure number cross reference.

Result	**Relevant figures**		
	Flight test 3	**Flight test 4**	**Flight test 5**
State estimates: Position.	Figure 4.59 to Figure 4.61	Figure 4.77 to Figure 4.79	Figure 4.96 to Figure 4.98
State estimates: Velocity.	Figure 4.62 to Figure 4.64	Figure 4.81 to Figure 4.83	Figure 4.100 to Figure 4.102
State estimates: Euler angle.	Figure 4.65 to Figure 4.67	Figure 4.84 to Figure 4.86	Figure 4.103 to Figure 4.105
State estimates: Accelerometer bias.	Figure 4.68 to Figure 4.70	Figure 4.87 to Figure 4.89	Figure 4.106 to Figure 4.108
State estimates: Rate gyro bias	Figure 4.71 to Figure 4.73	Figure 4.90 to Figure 4.92	Figure 4.109 to Figure 4.111
Horizontal position	None	Figure 4.80	Figure 4.99
$\|\| \mathbf{I}-\mathbf{K_k H_k} \|\|$ and $\|\| \mathbf{K_k} \|\|$	Figure 4.74	Figure 4.93	Figure 4.112
Standard deviations: Navigation	Figure 4.75	Figure 4.94	Figure 4.113
Standard deviations: Biases	Figure 4.76	Figure 4.95	Figure 4.114

4.5.6.4.3 Navigation result: Flight test 3

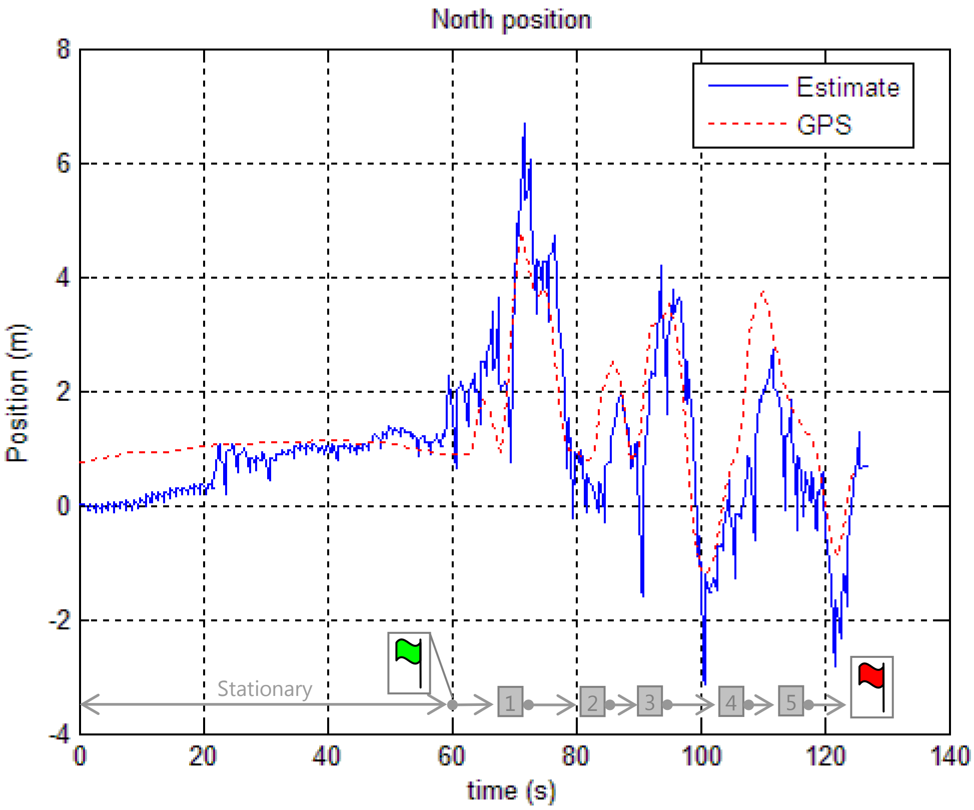


Figure 4.59. Flight test 3: North position estimate.

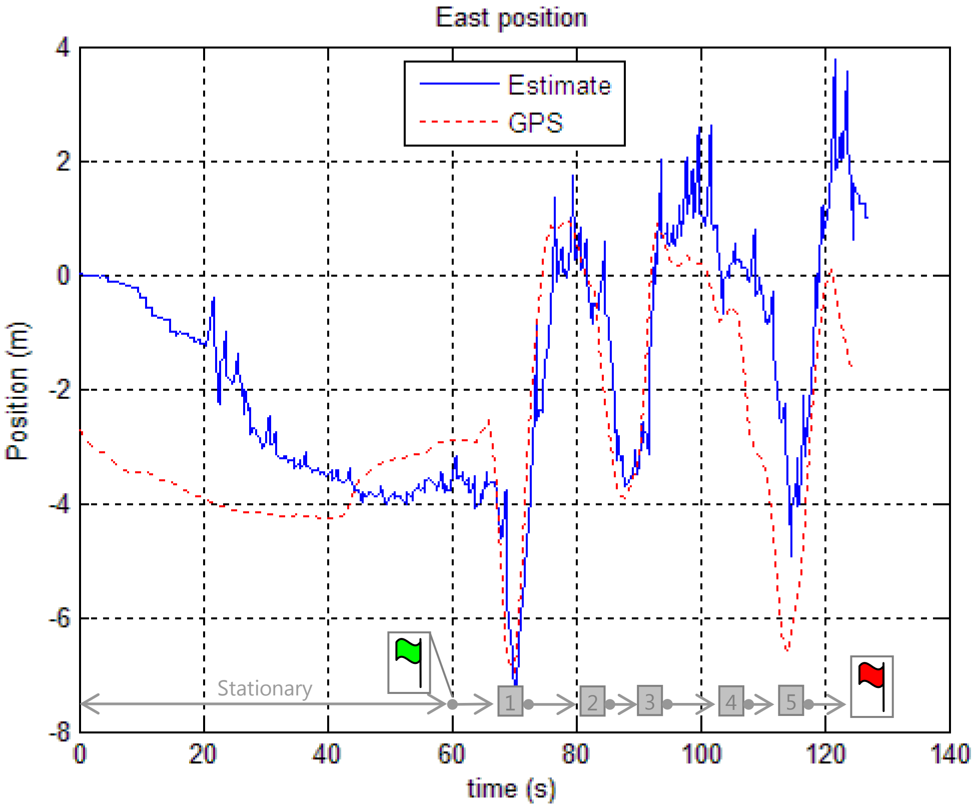


Figure 4.60. Flight test 3: East position estimate.

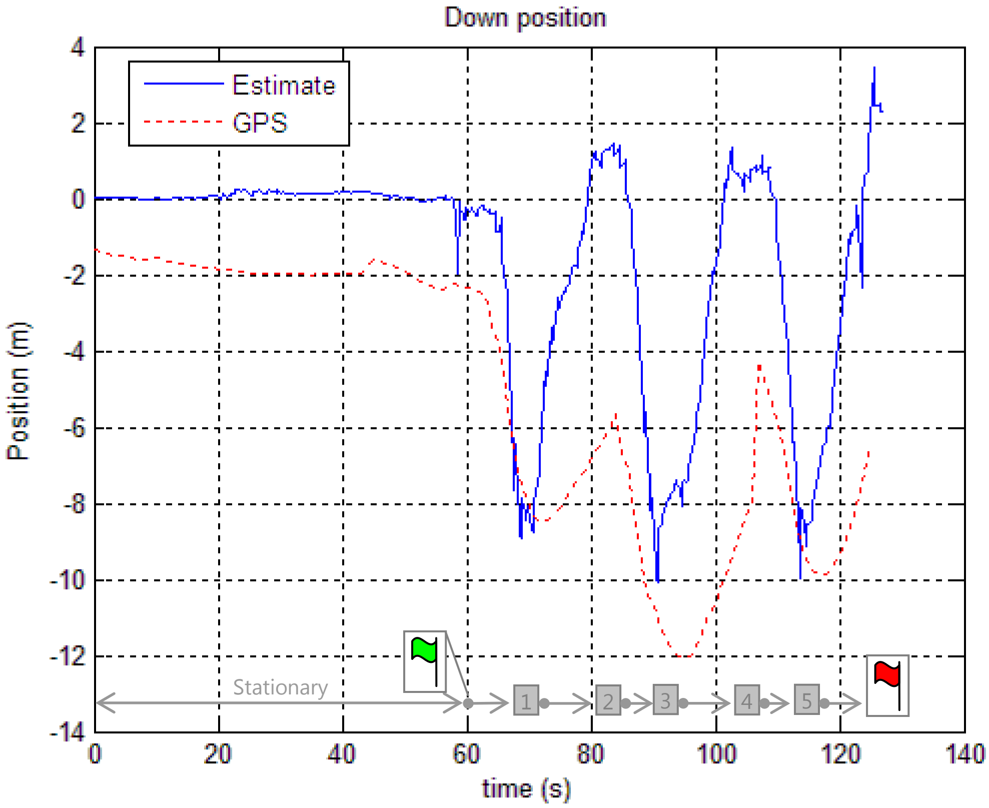


Figure 4.61. Flight test 3: Down position estimate.

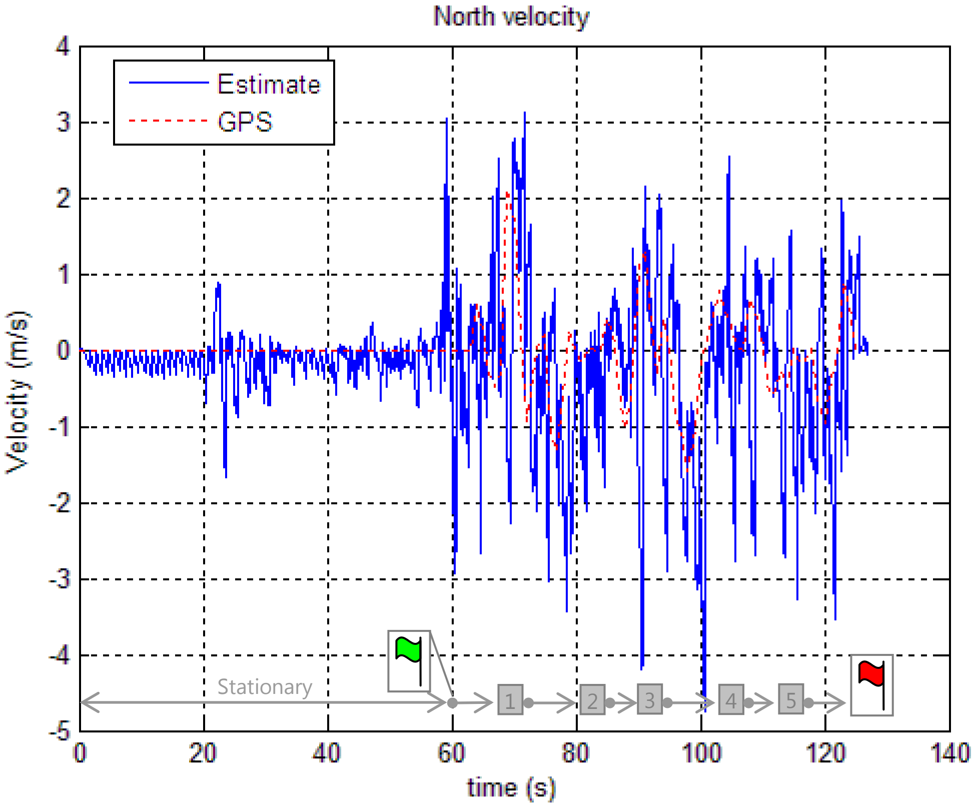


Figure 4.62. Flight test 3: North velocity estimate.

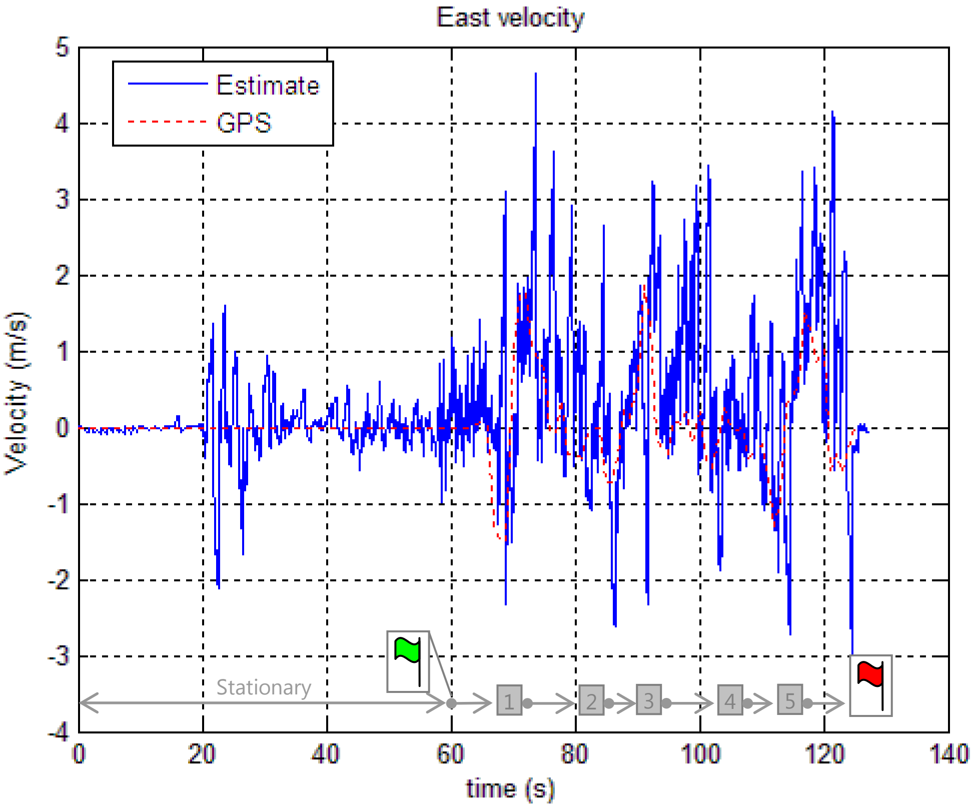


Figure 4.63. Flight test 3: East velocity estimate.

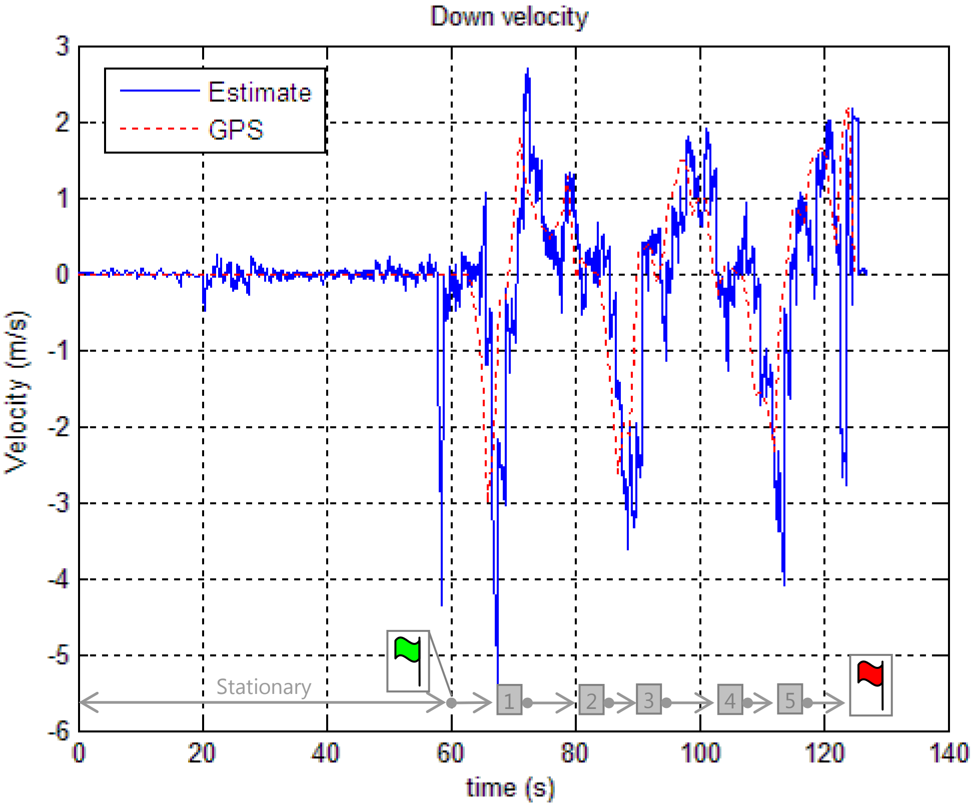


Figure 4.64. Flight test 3: Down velocity estimate.

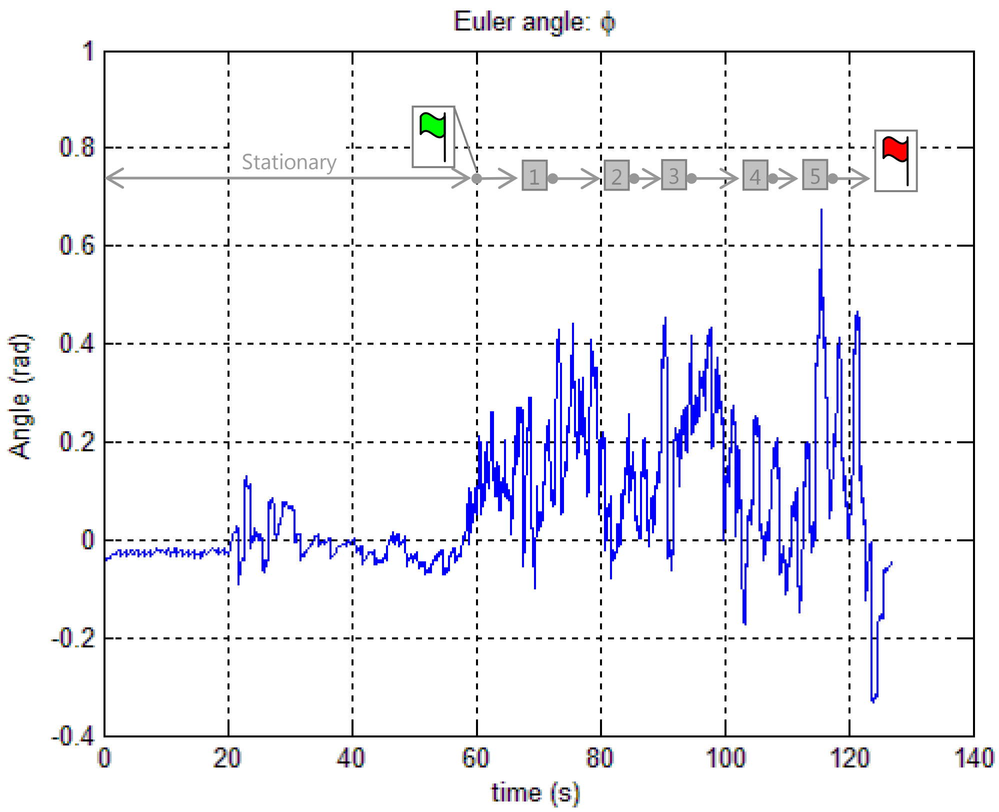


Figure 4.65. Flight test 3: Euler estimate for roll.

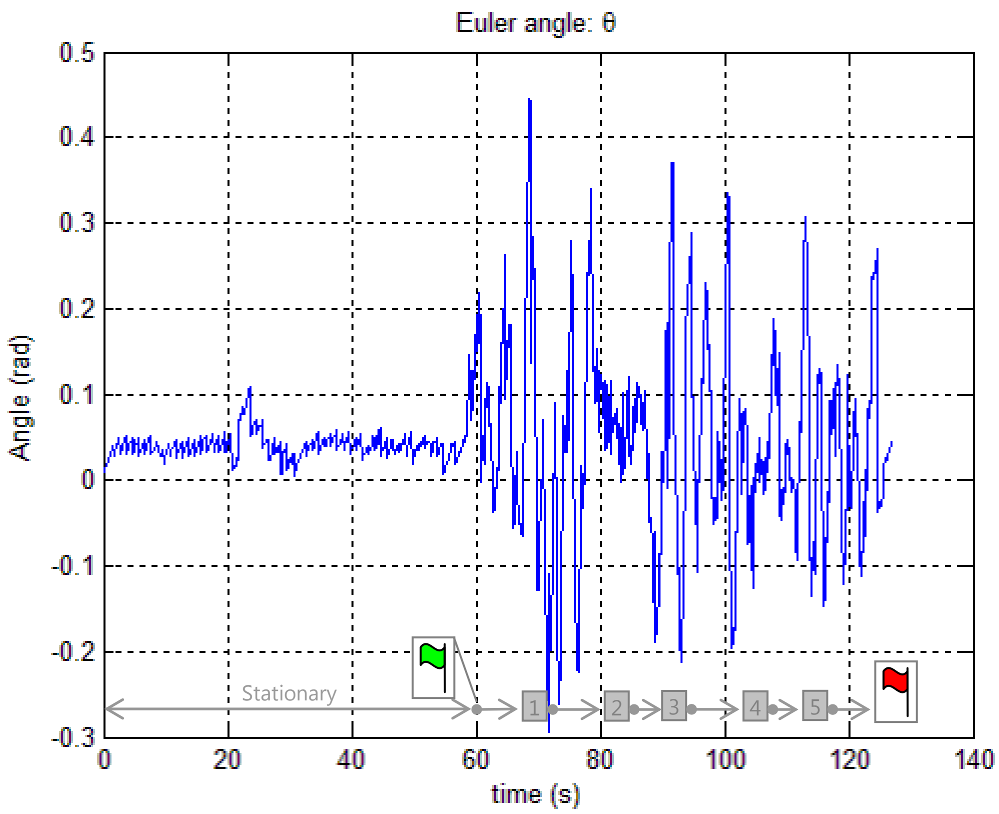


Figure 4.66. Flight test 3: Euler estimate for pitch.

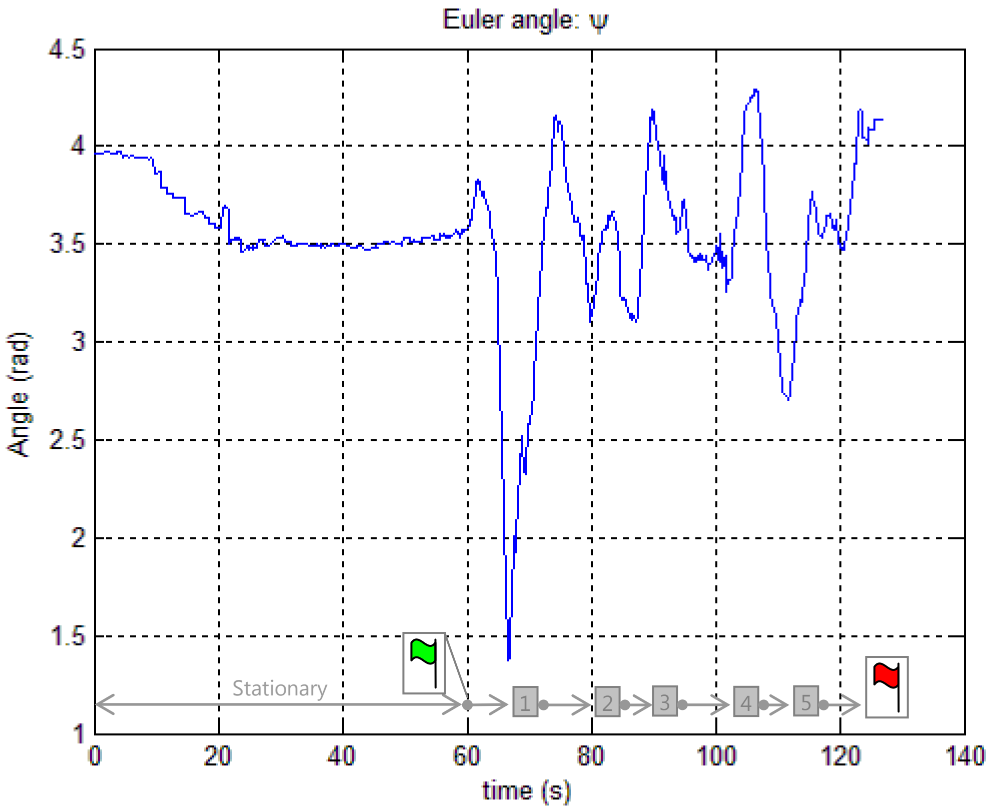


Figure 4.67. Flight test 3: Euler estimate for yaw.

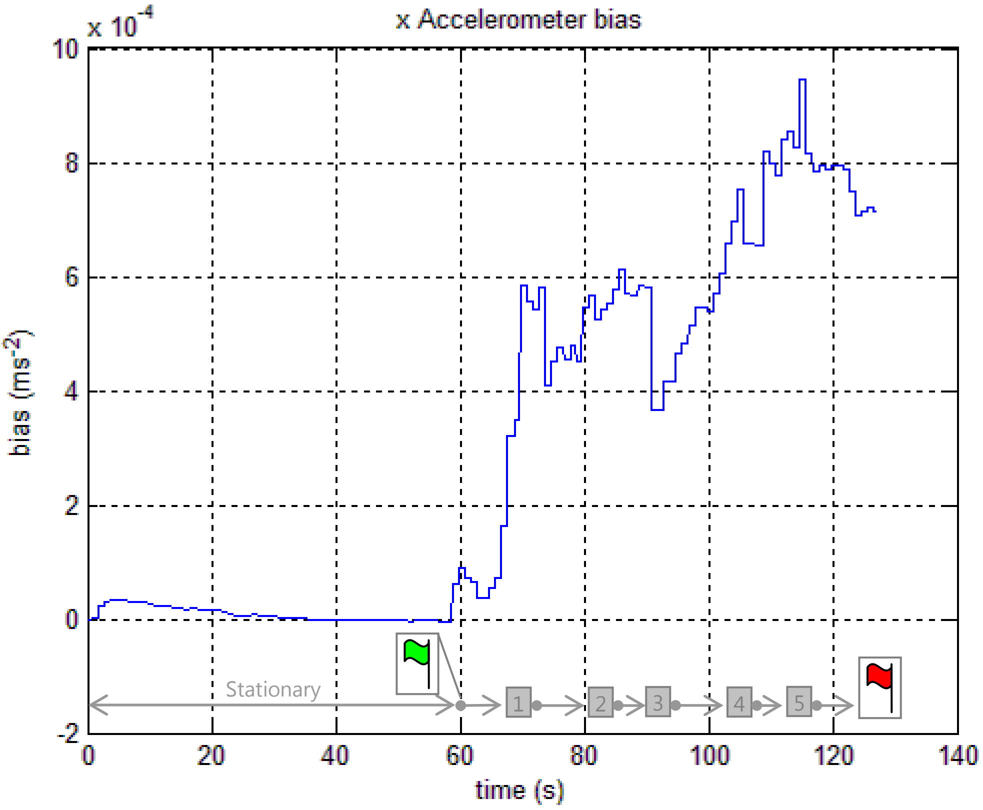


Figure 4.68. Flight test 3: Estimate for x accelerometer bias.

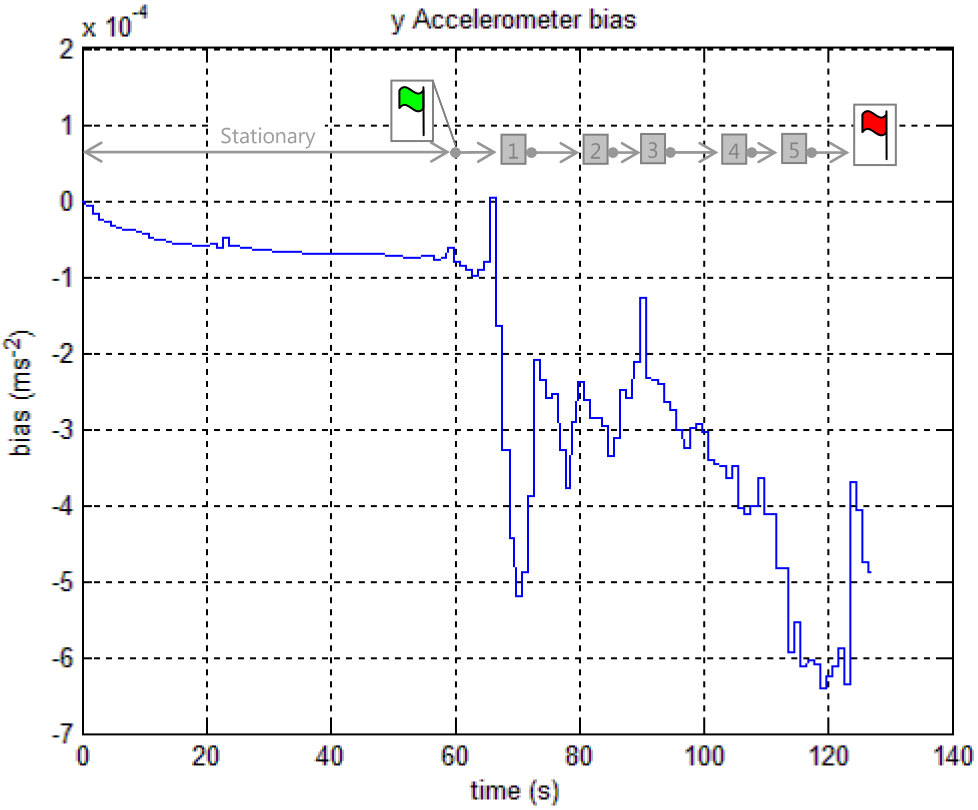


Figure 4.69. Flight test 3: Estimate for y accelerometer bias.

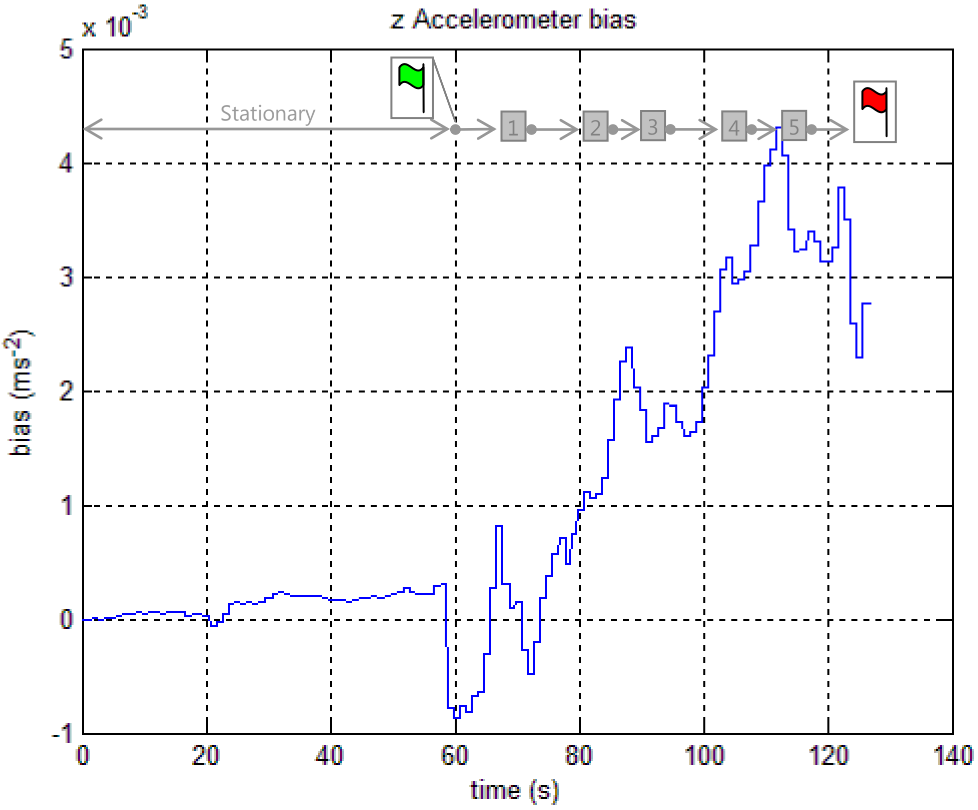


Figure 4.70. Flight test 3: Estimate for z accelerometer bias.

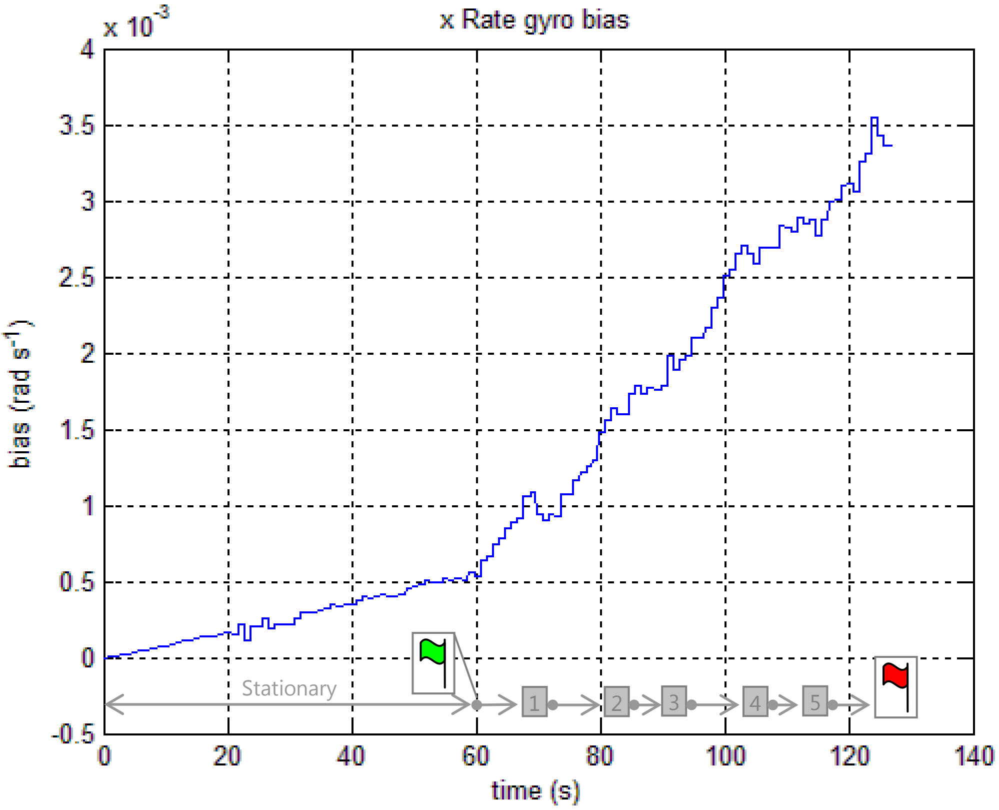


Figure 4.71. Flight test 3: Estimate for x rate gyro bias.

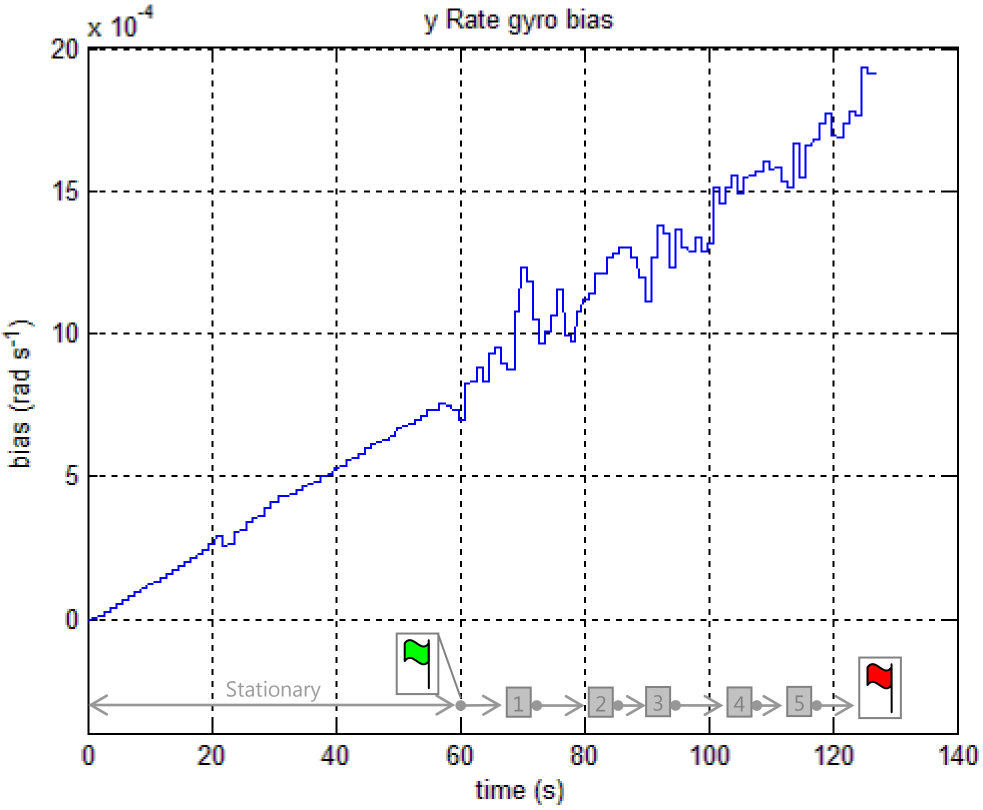


Figure 4.72. Flight test 3: Estimate for y rate gyro bias.

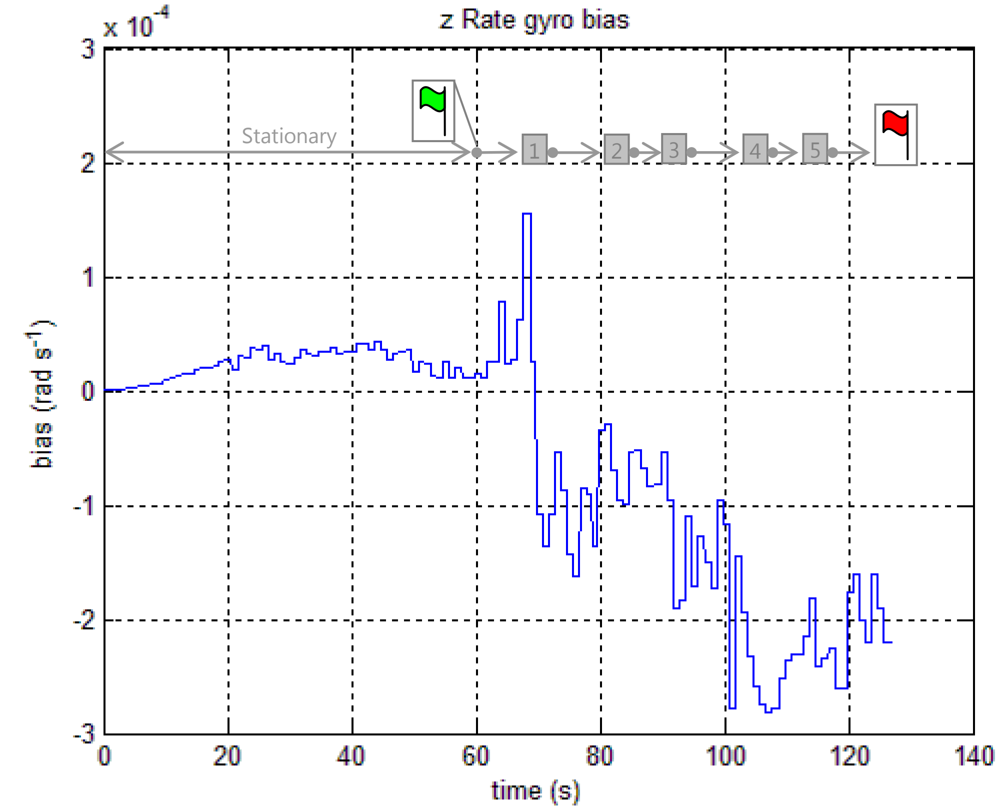


Figure 4.73. Flight test 3: Estimate for z rate gyro bias.

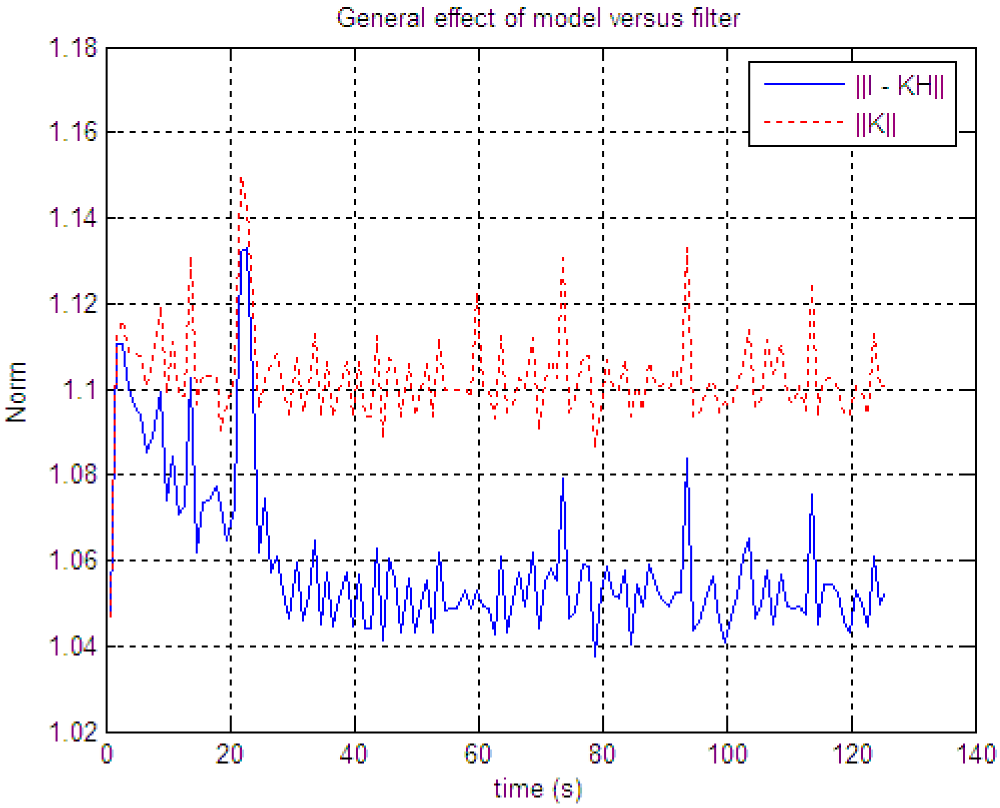


Figure 4.74. Flight test 3: General effect of model versus filter.

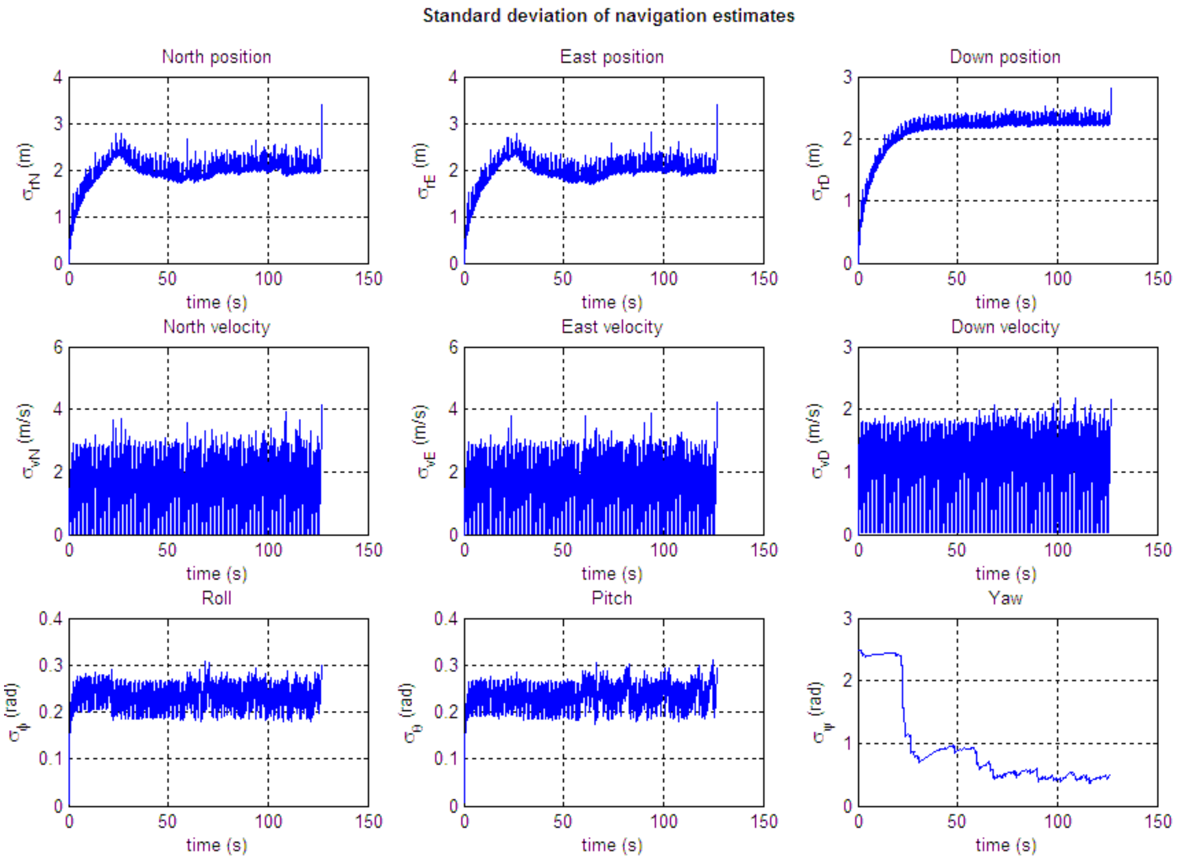


Figure 4.75. Flight test 3: Standard deviation of navigation estimates.

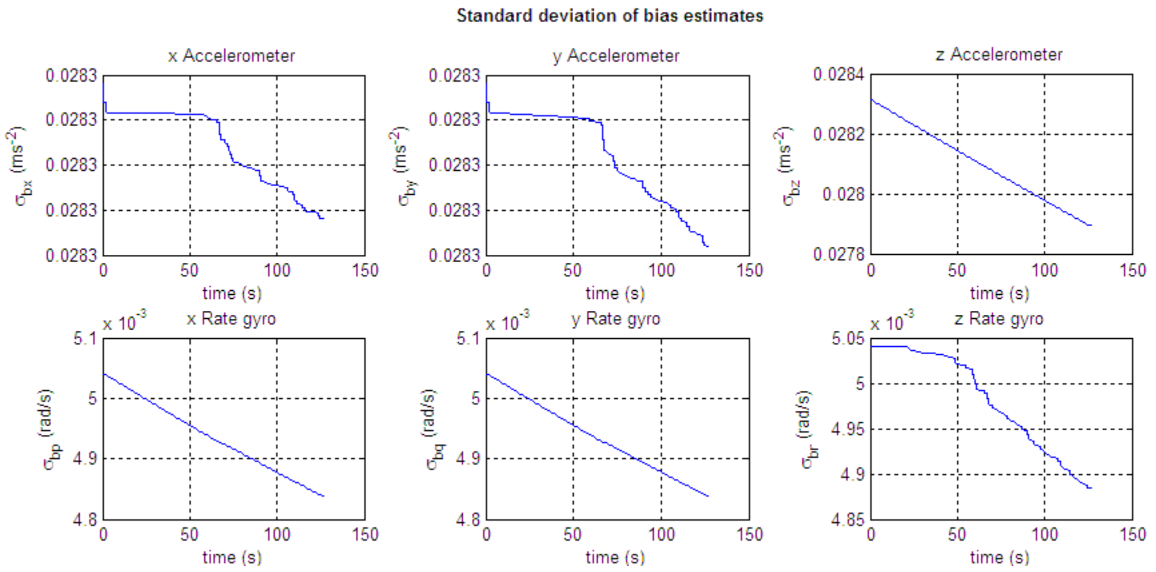


Figure 4.76. Flight test 3: Standard deviation of bias estimates.

4.5.6.4.4 Navigation result: Flight test 4

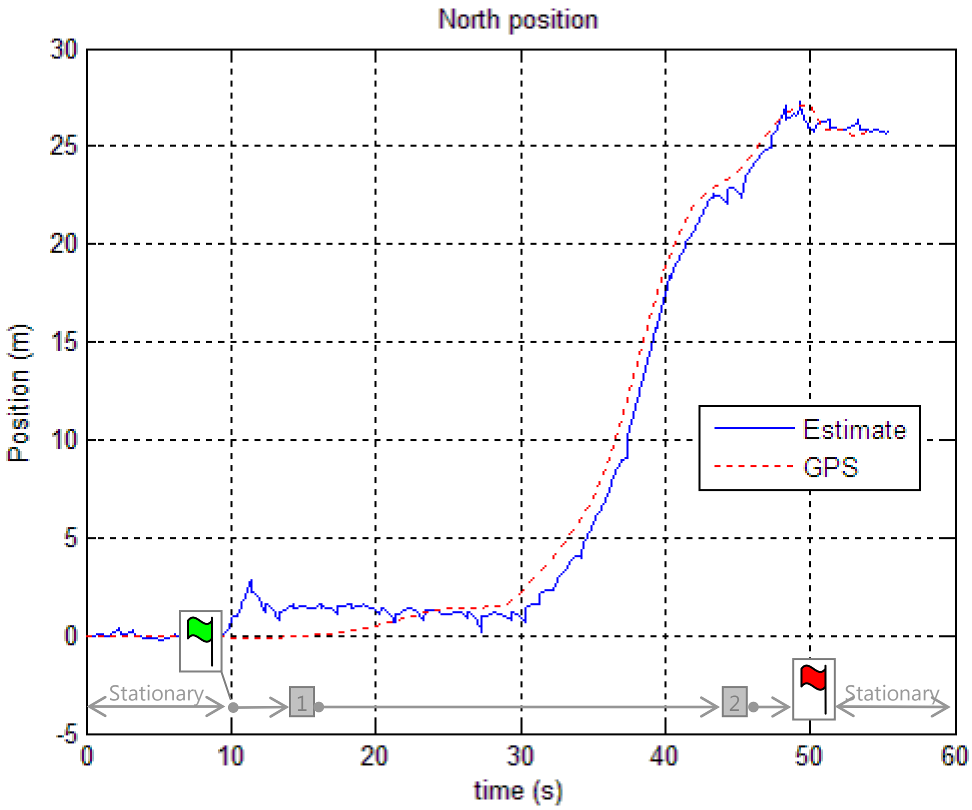


Figure 4.77. Flight test 4: North position estimate.

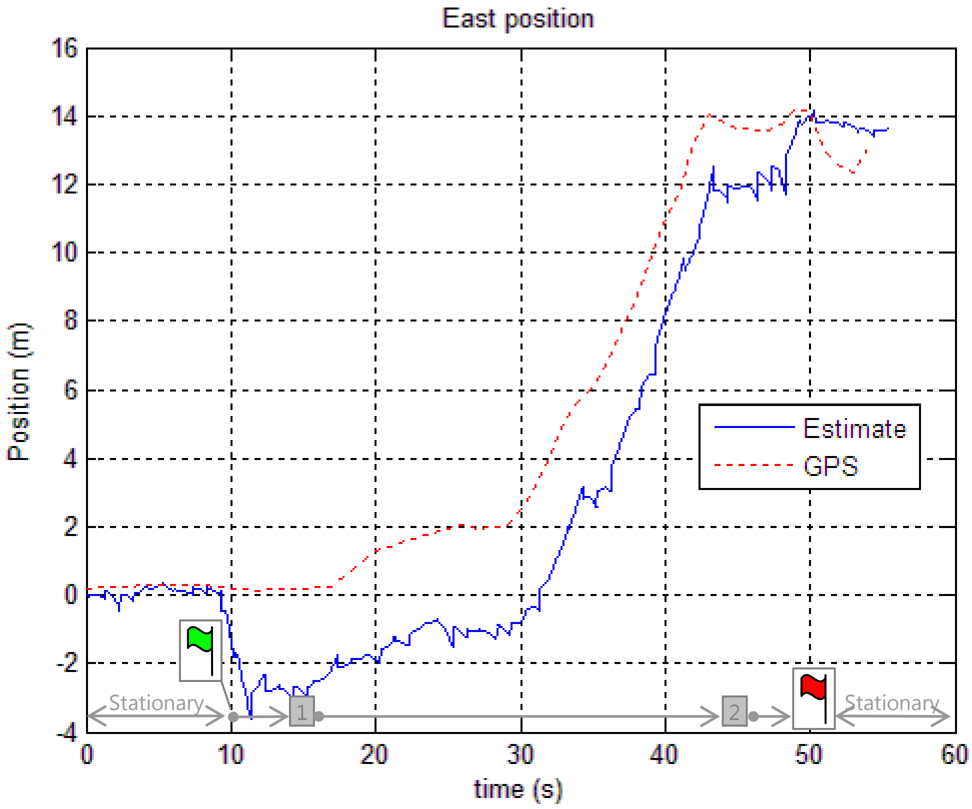


Figure 4.78. Flight test 4: East position estimate.

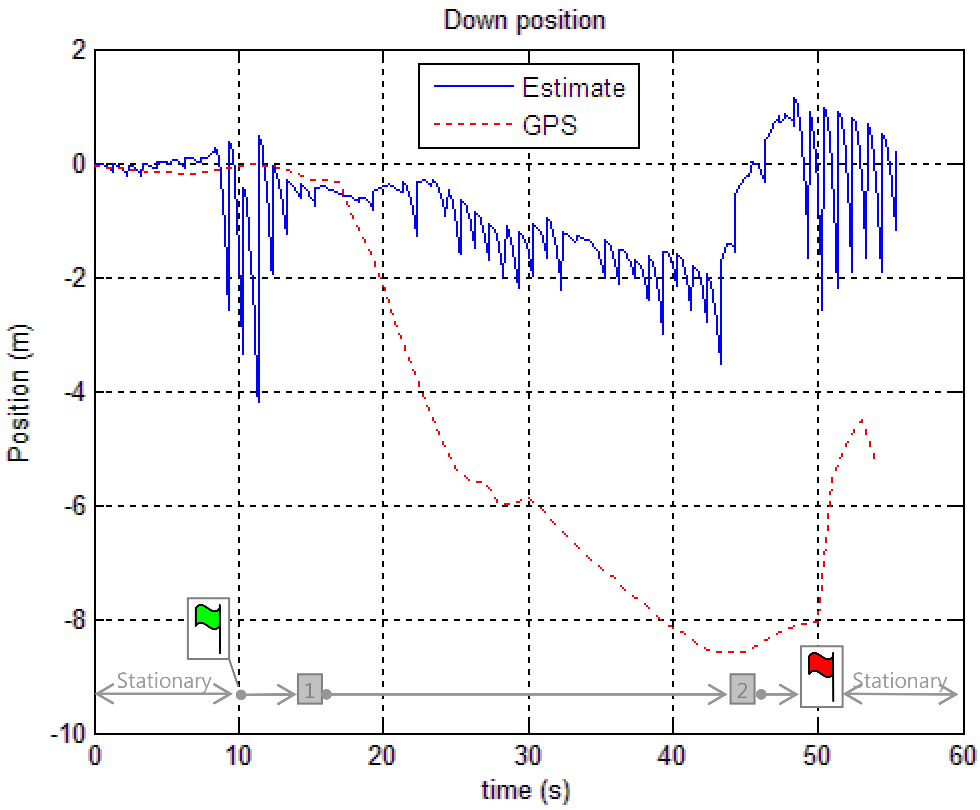


Figure 4.79. Flight test 4: Down position estimate.

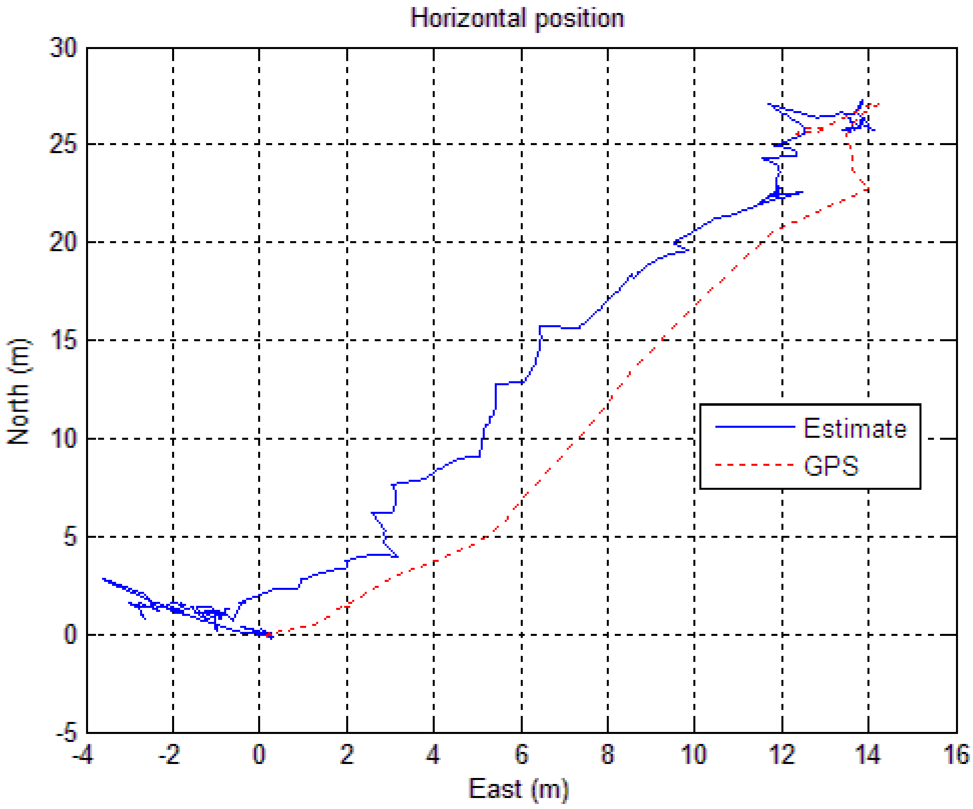


Figure 4.80. Flight test 4: Horizontal position estimate.

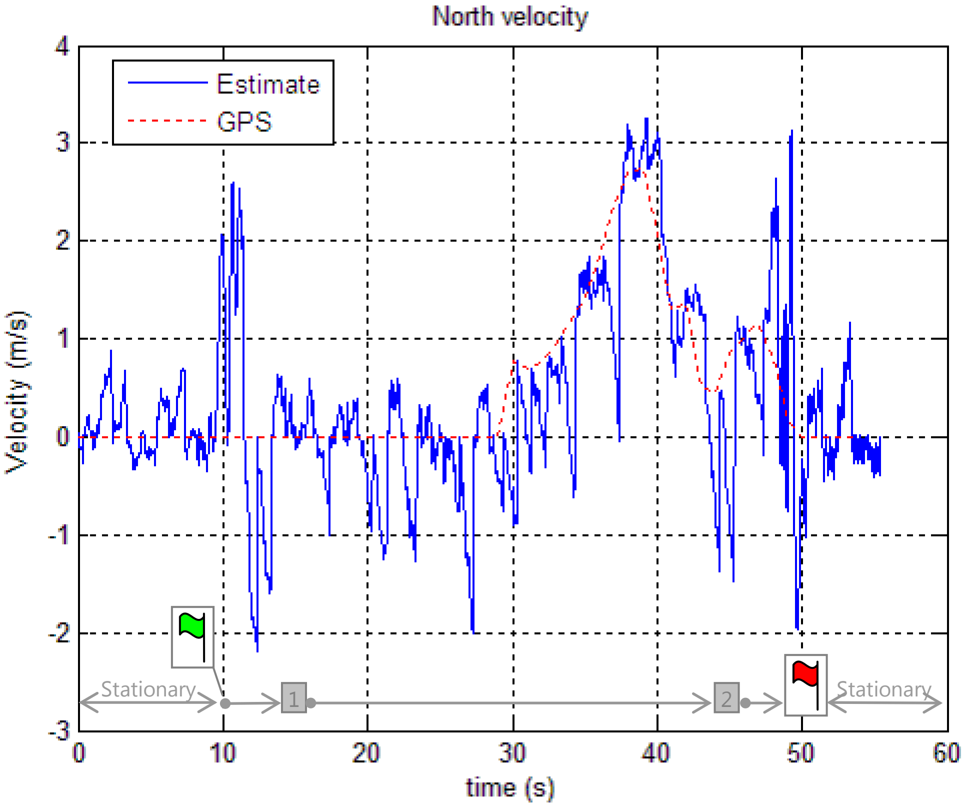


Figure 4.81. Flight test 4: North velocity estimate.

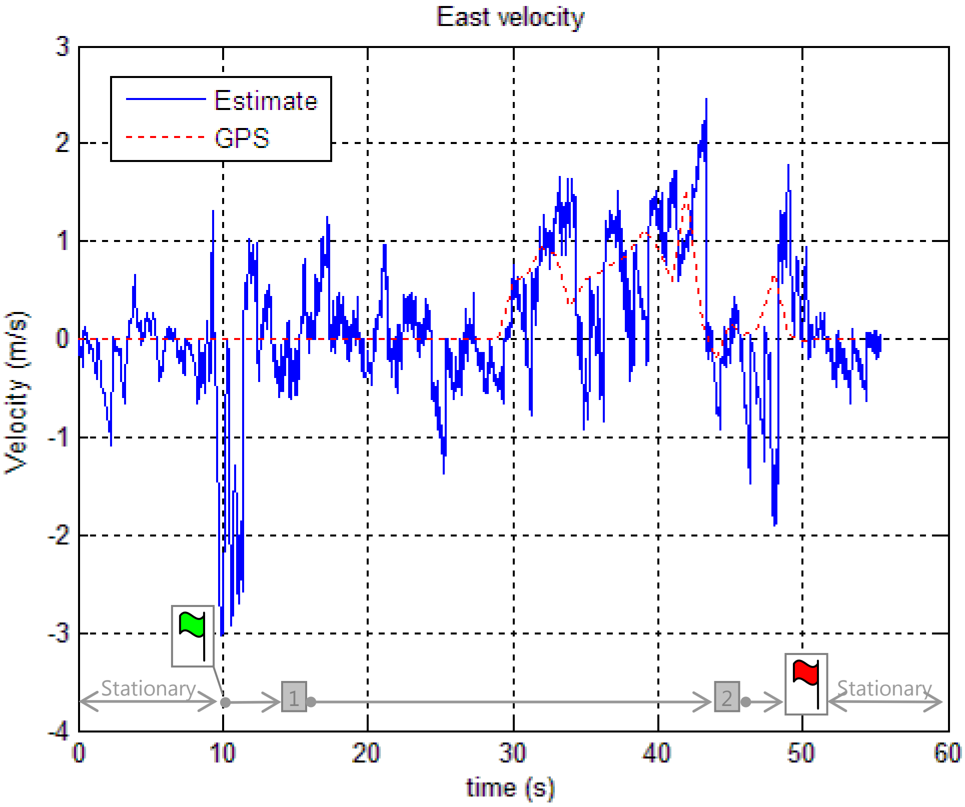


Figure 4.82. Flight test 4: East velocity estimate.

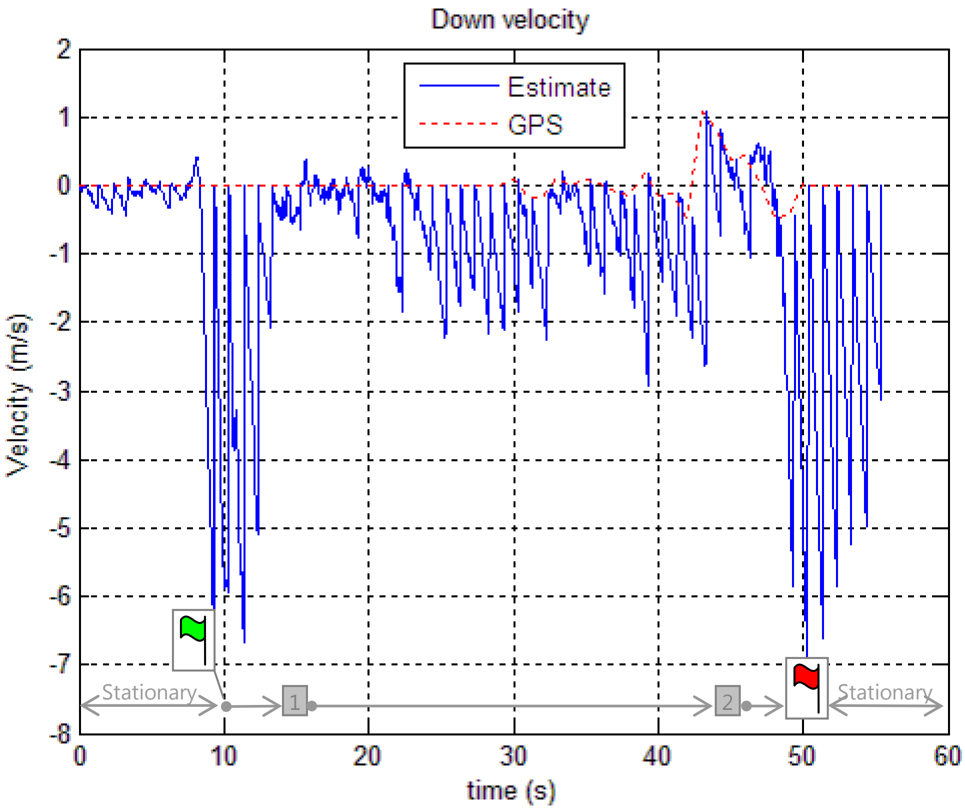


Figure 4.83. Flight test 4: Down velocity estimate.

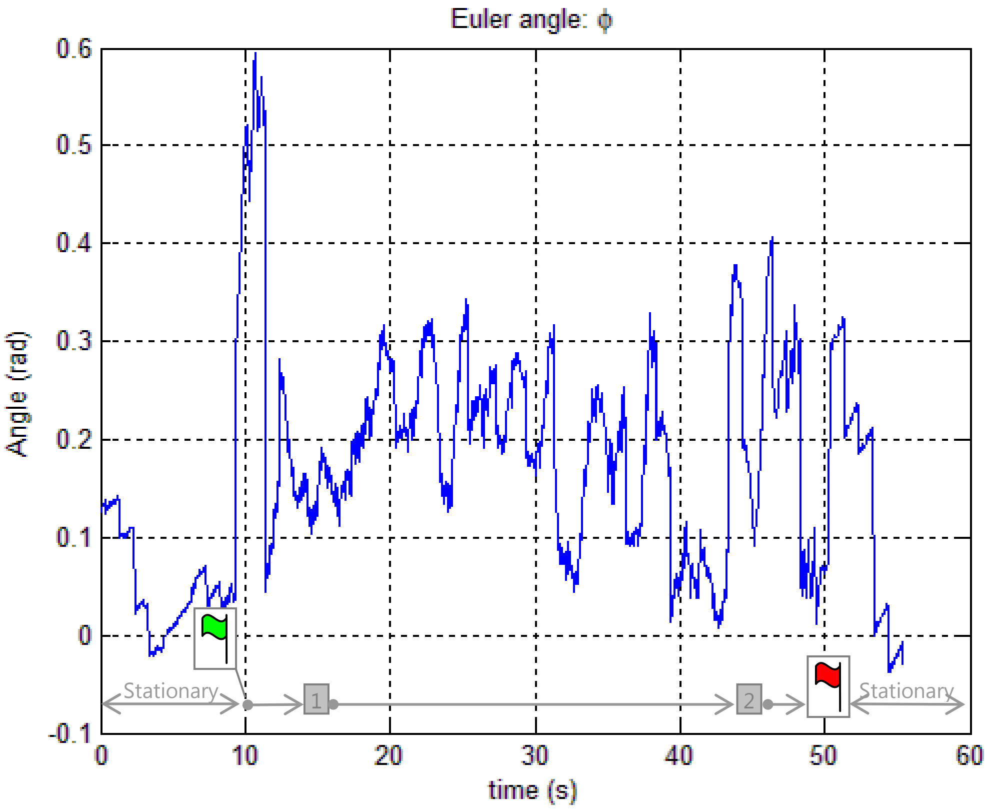


Figure 4.84. Flight test 4: Euler estimate for roll.

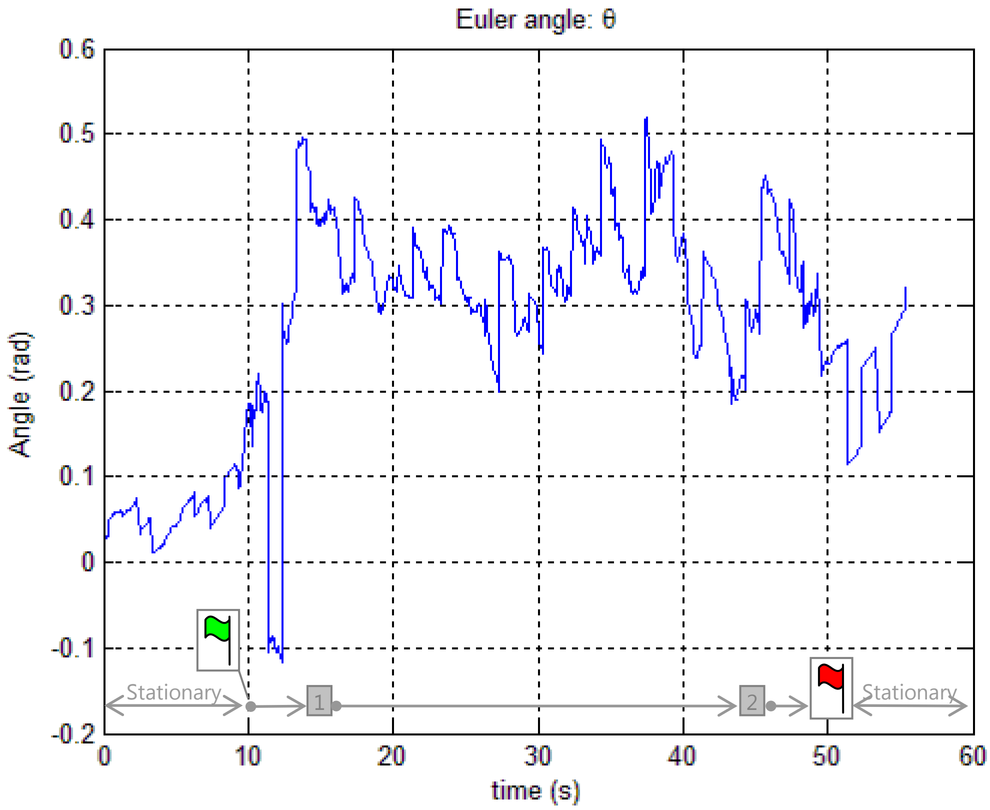


Figure 4.85. Flight test 4: Euler estimate for pitch.

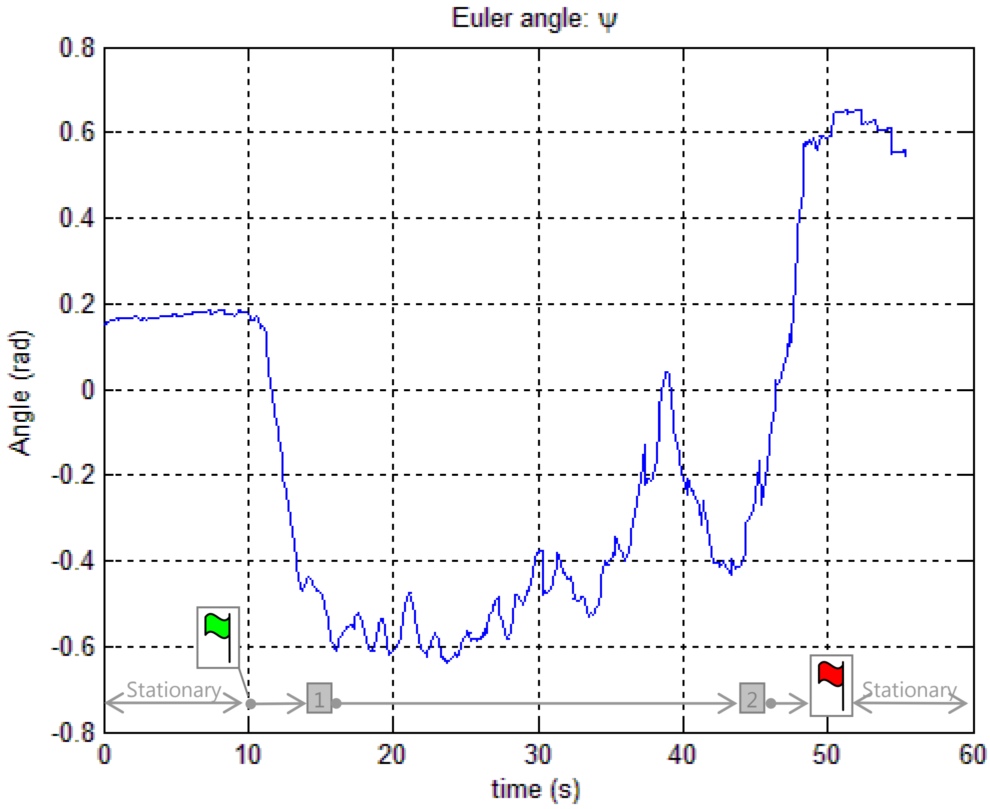


Figure 4.86. Flight test 4: Euler estimate for yaw.

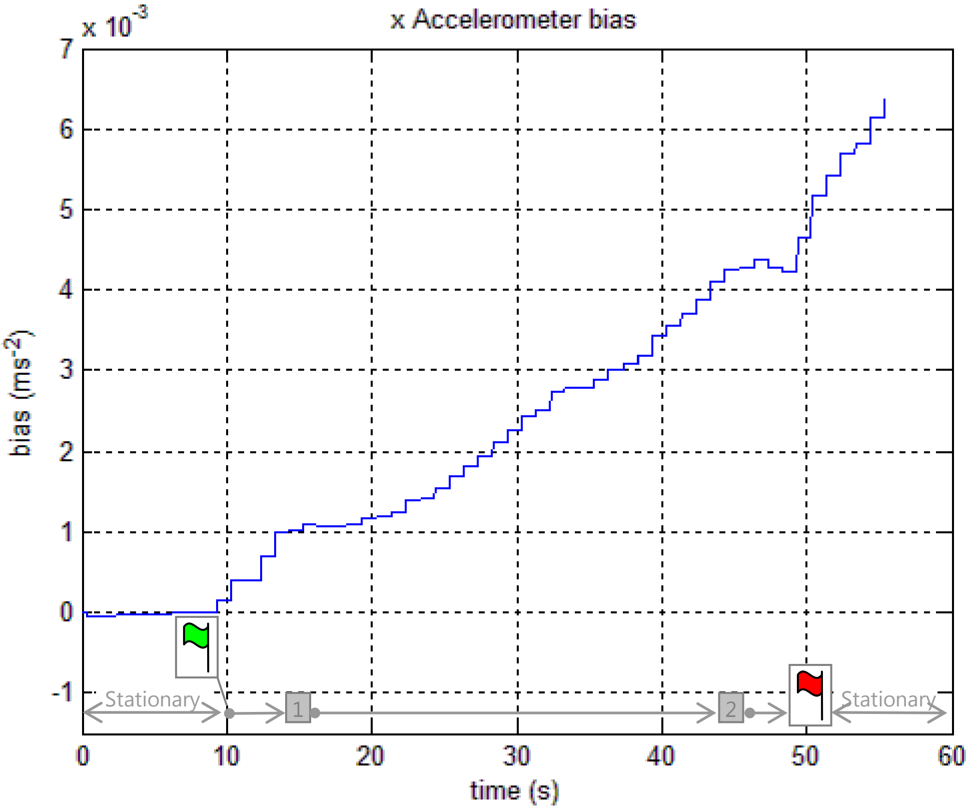


Figure 4.87. Flight test 4: Estimate for x accelerometer bias.

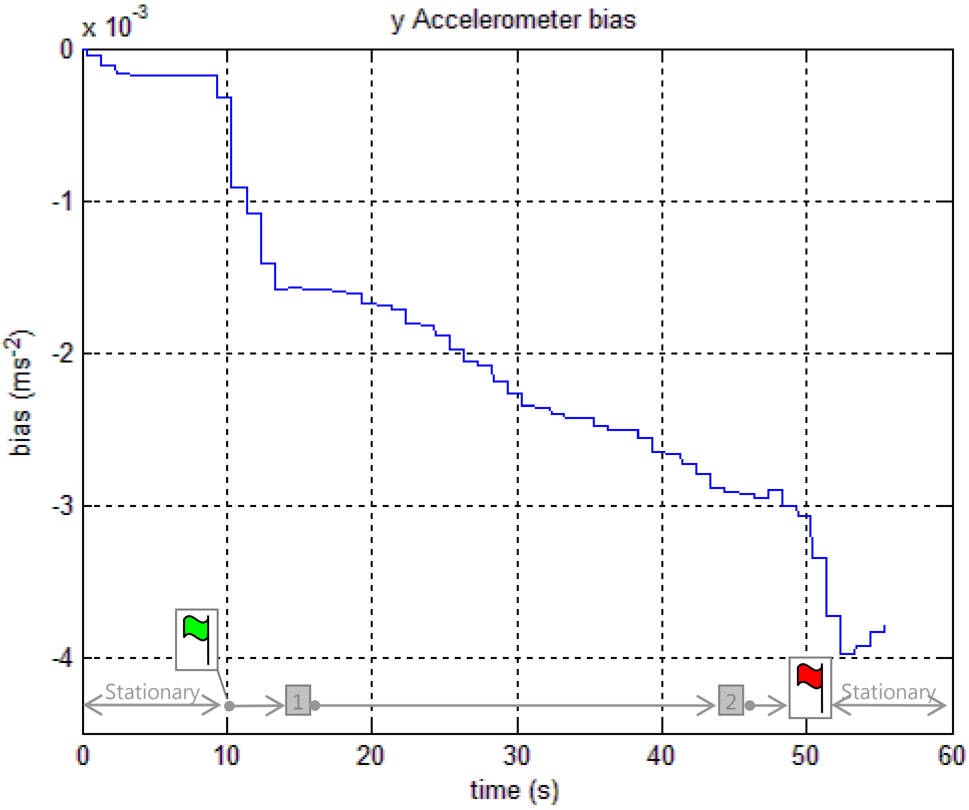


Figure 4.88. Flight test 4: Estimate for y accelerometer bias.

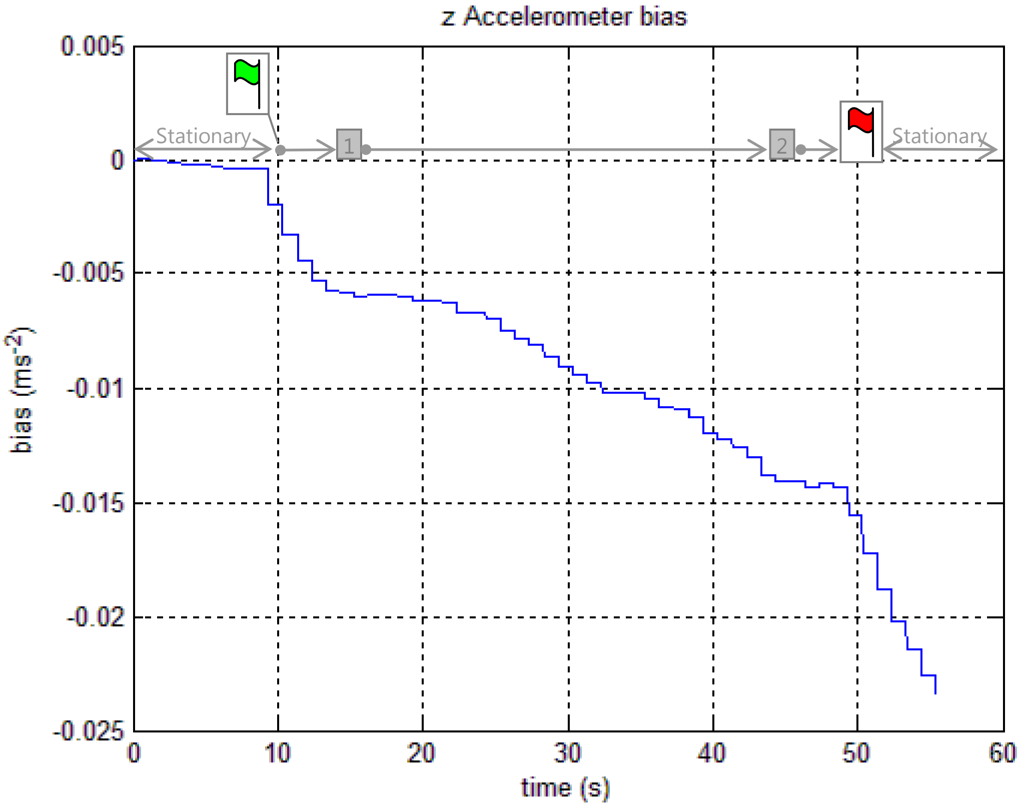


Figure 4.89. Flight test 4: Estimate for z accelerometer bias.

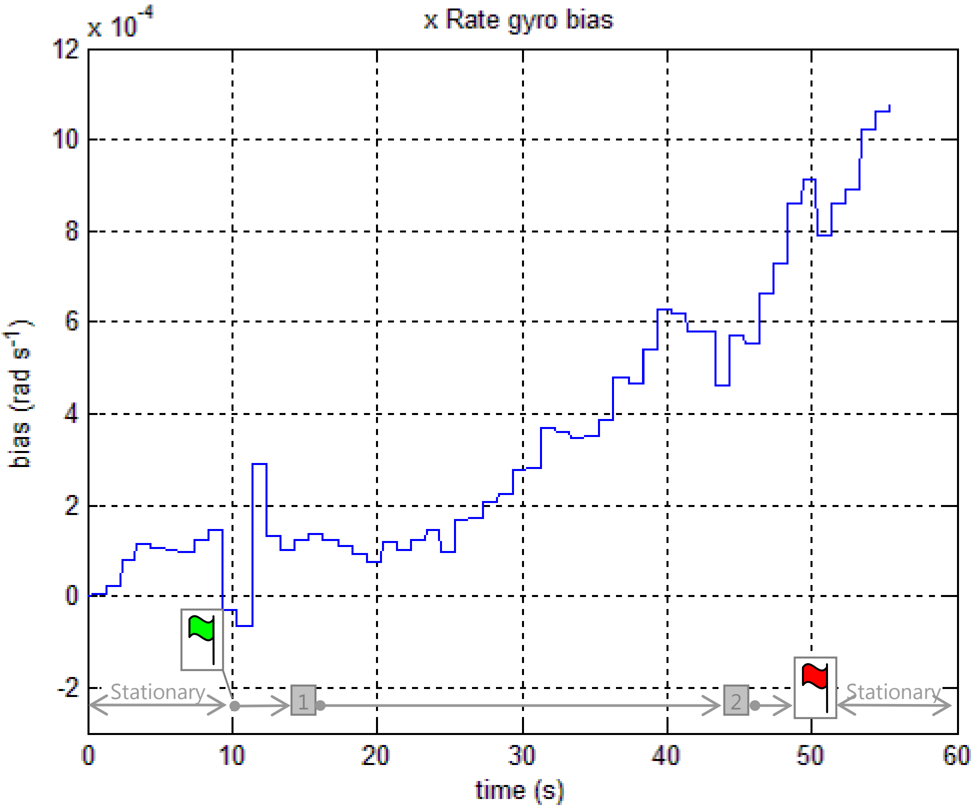


Figure 4.90. Flight test 4: Estimate for x rate gyro bias.

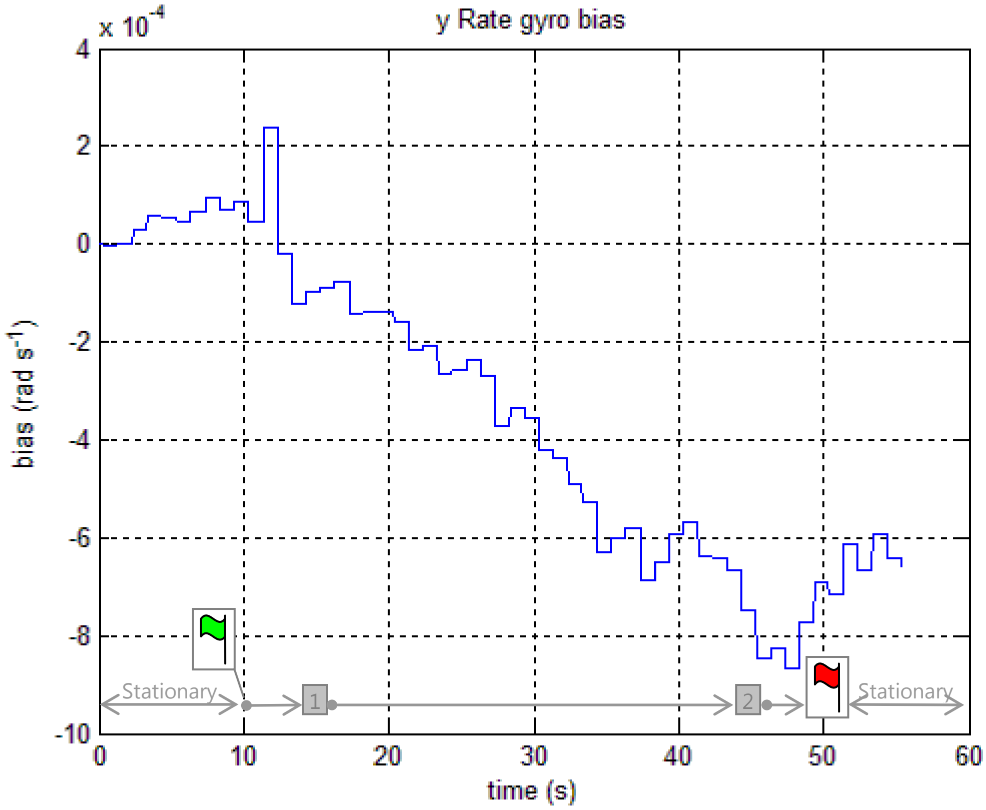


Figure 4.91. Flight test 4: Estimate for y rate gyro bias.

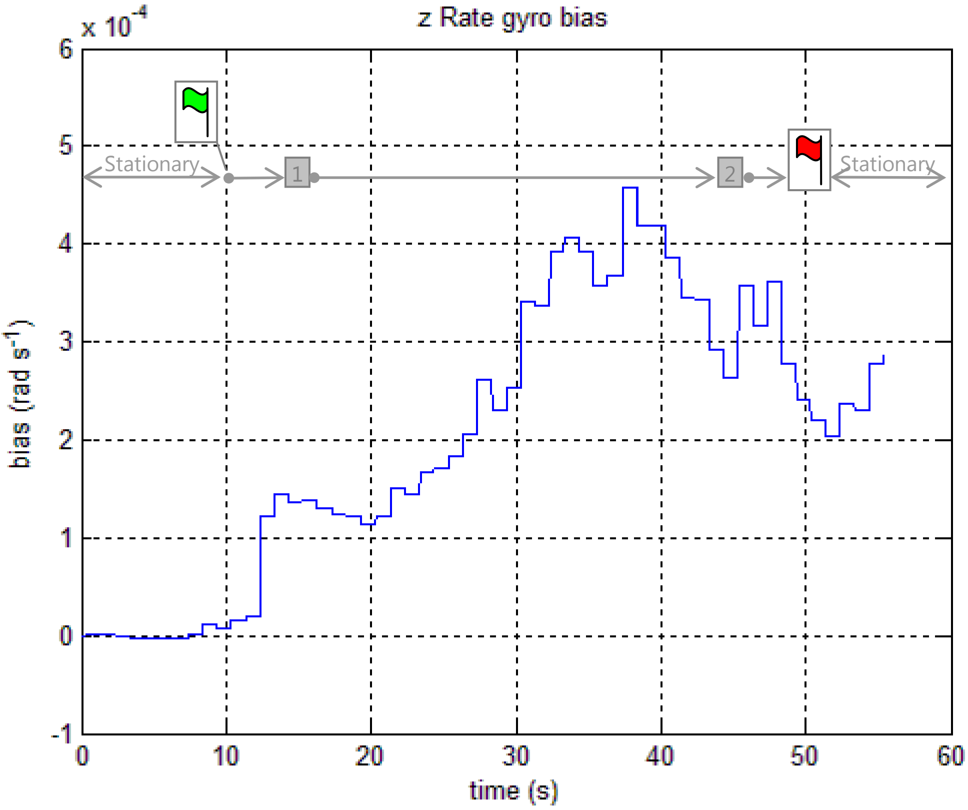


Figure 4.92. Flight test 4: Estimate for z rate gyro bias.

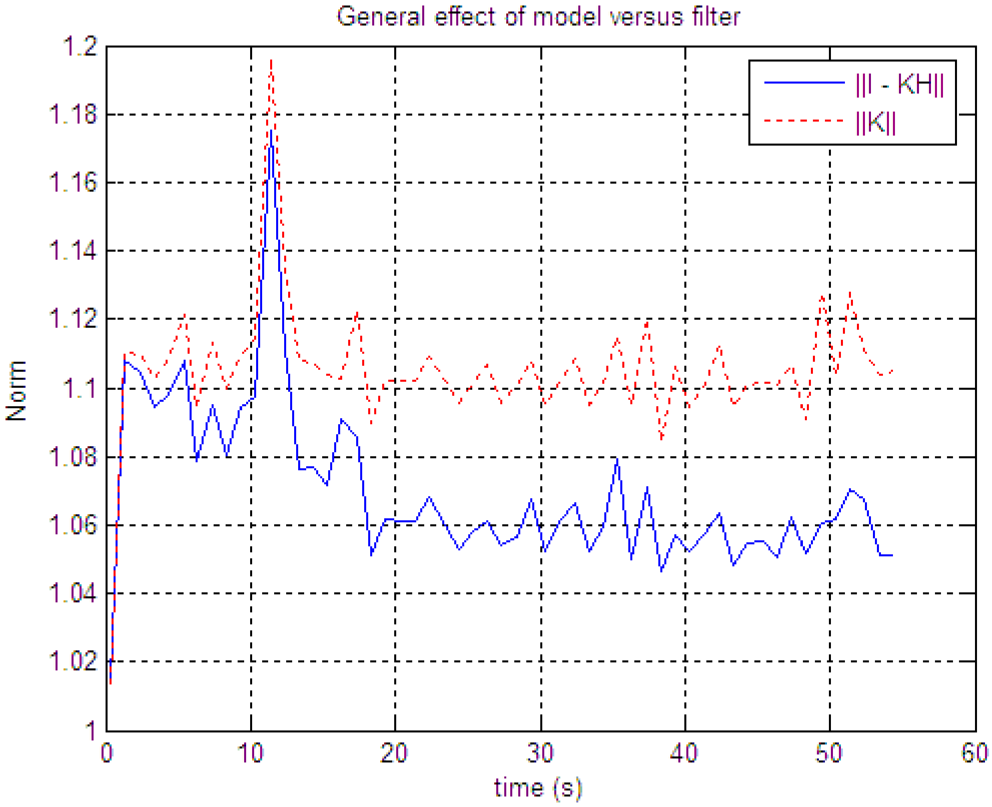


Figure 4.93. Flight test 4: General effect of model versus filter.

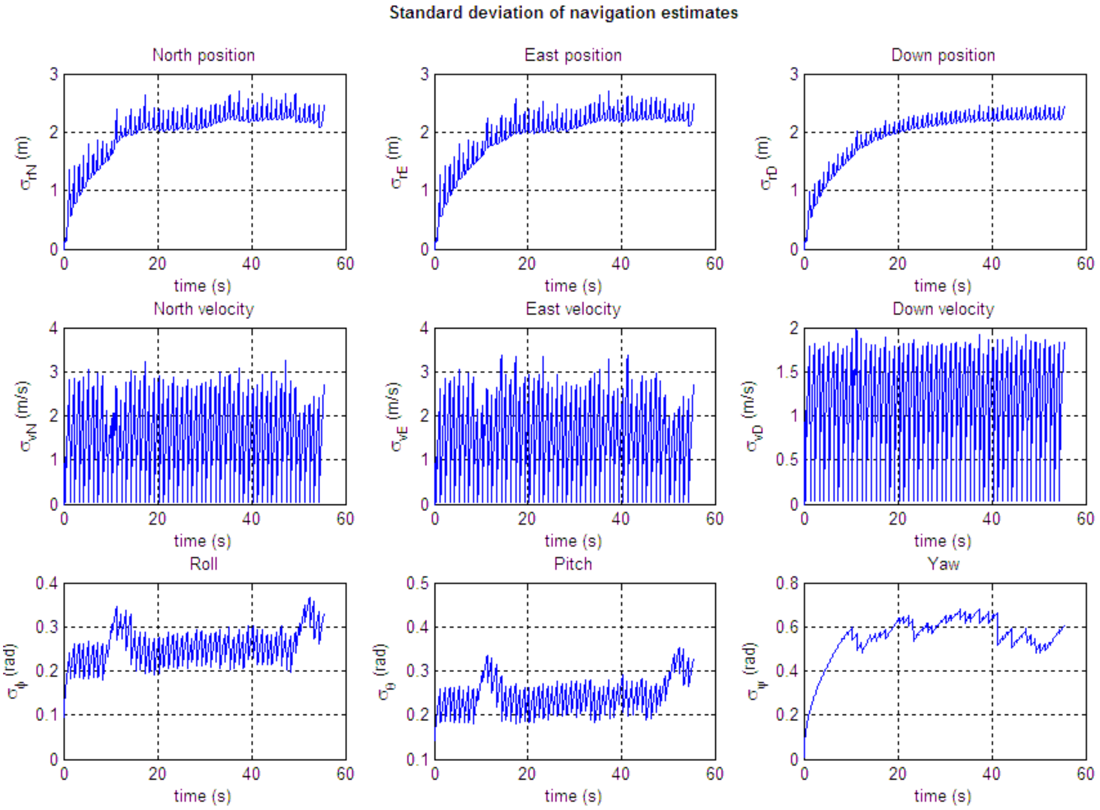


Figure 4.94. Flight test 4: Standard deviation of navigation estimates.

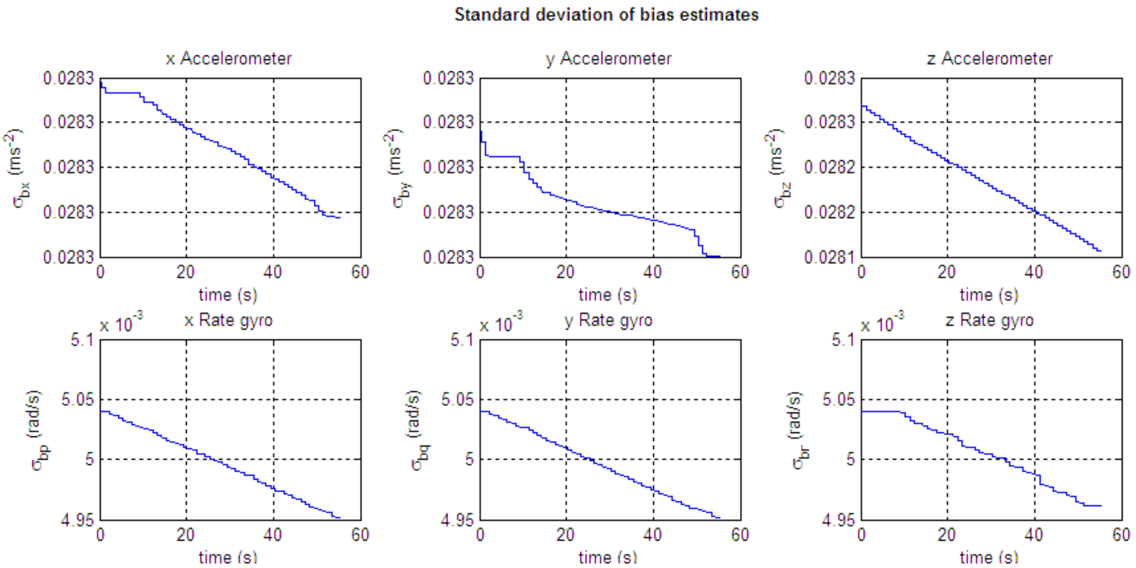


Figure 4.95. Flight test 4: Standard deviation of bias estimates.

4.5.6.4.5 Navigation result: Flight test 5

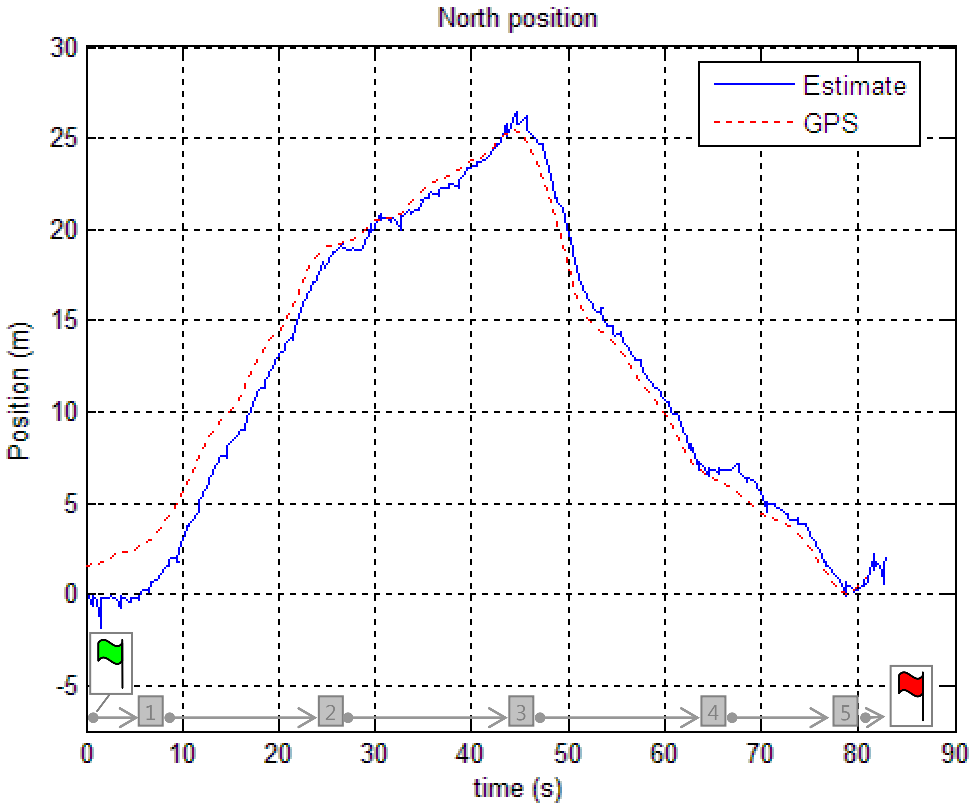


Figure 4.96. Flight test 5: North position estimate.

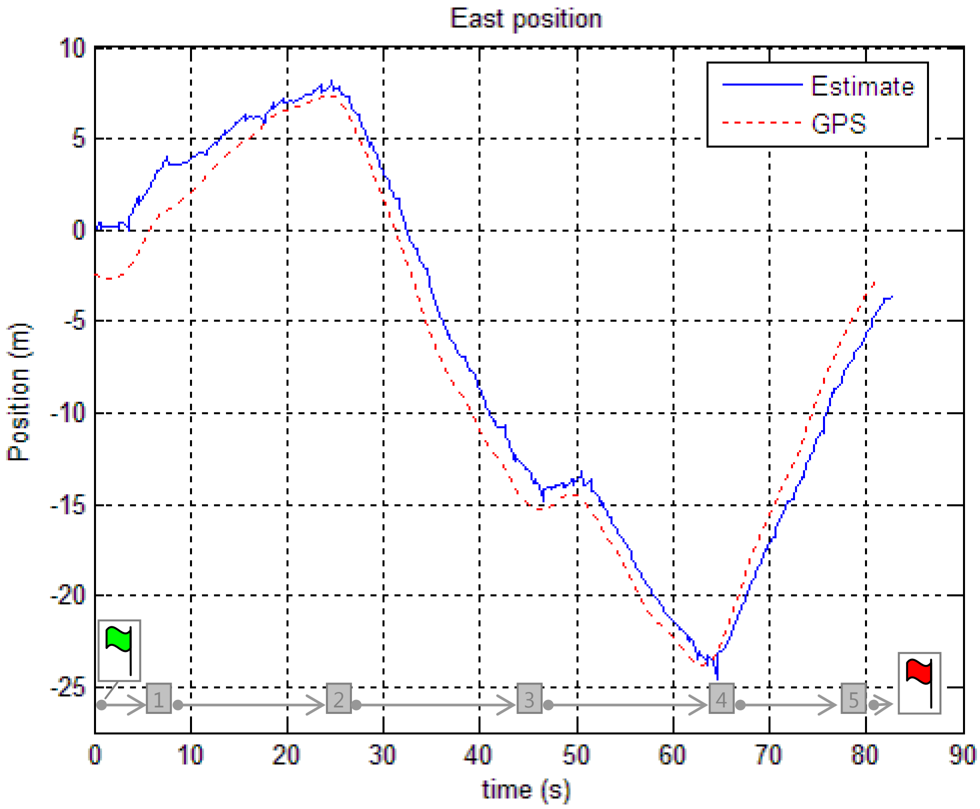


Figure 4.97. Flight test 5: East position estimate.

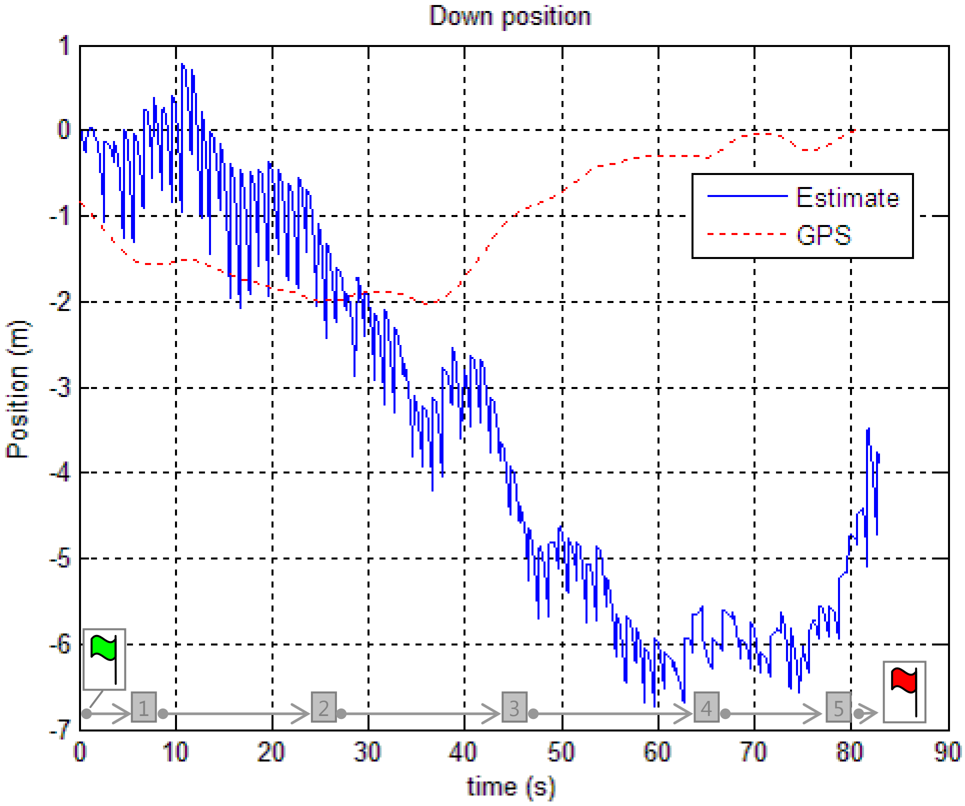


Figure 4.98. Flight test 5: Down position estimate.

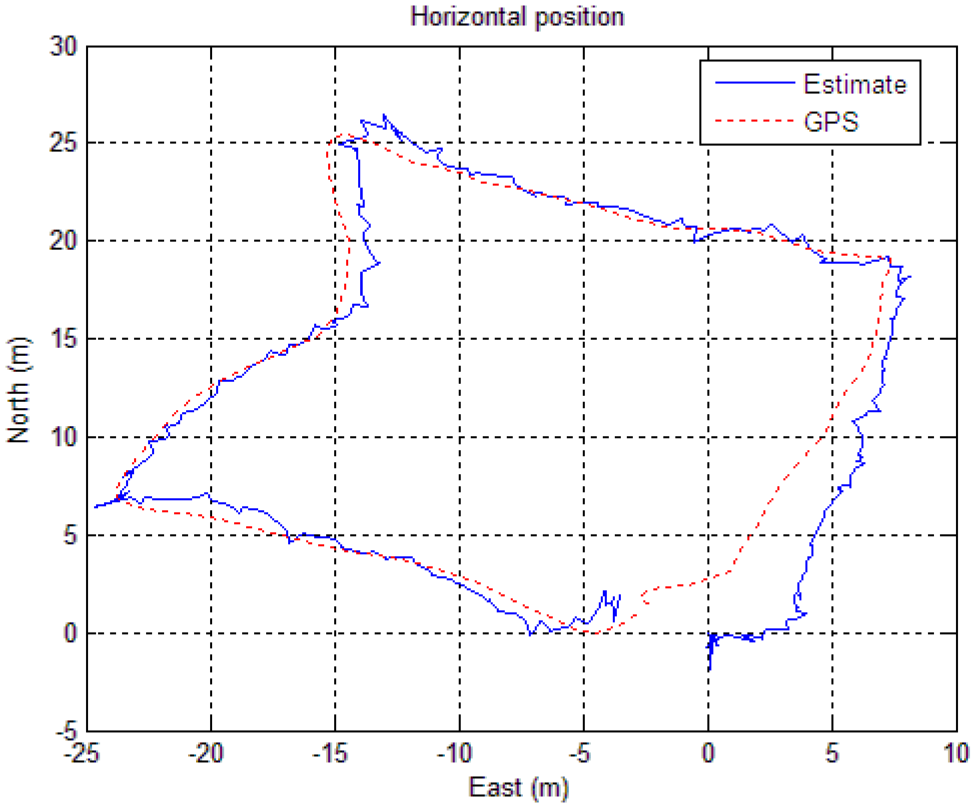


Figure 4.99. Flight test 5: Horizontal position estimate.

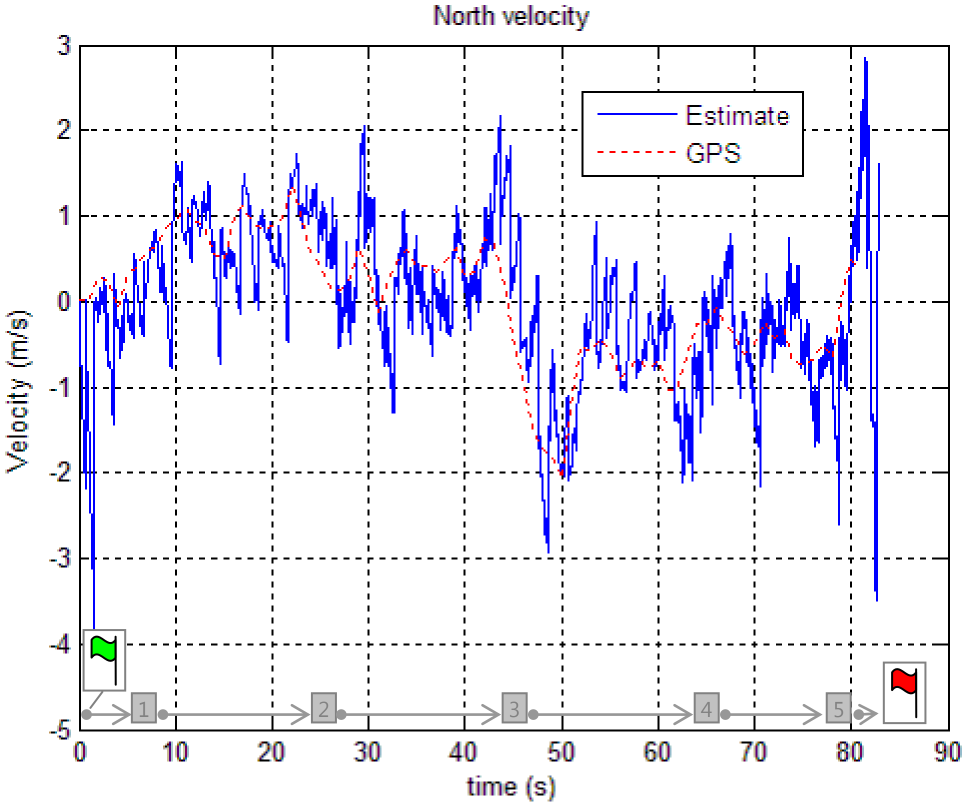


Figure 4.100. Flight test 5: North velocity estimate.

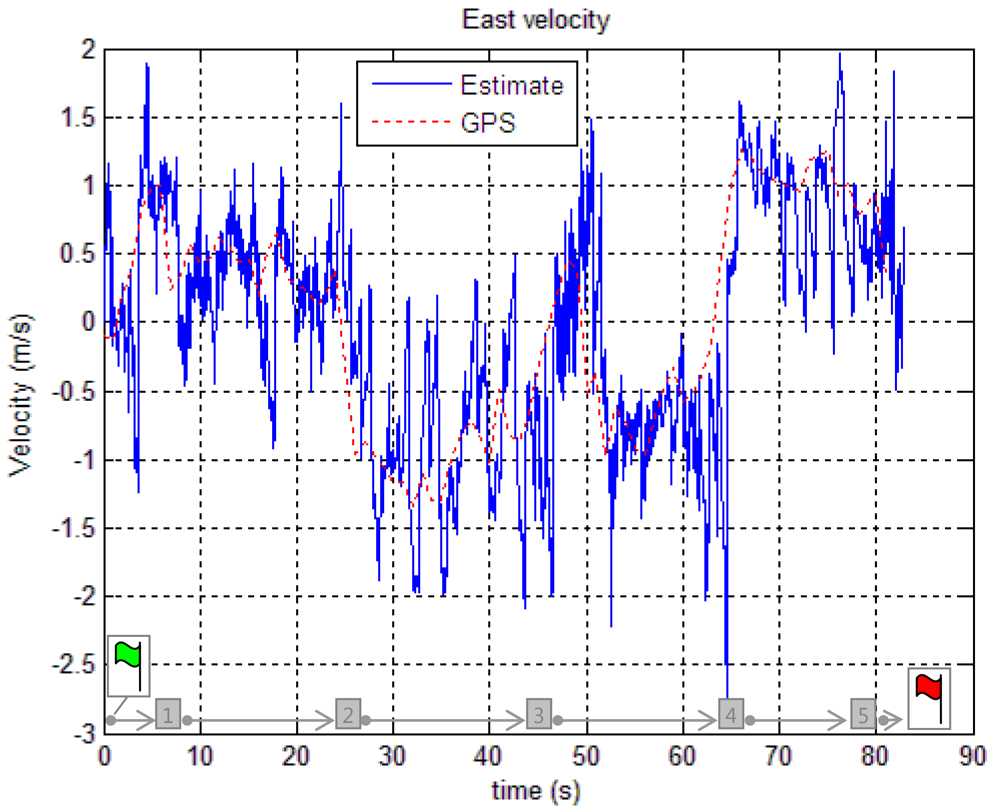


Figure 4.101. Flight test 5: East velocity estimate.

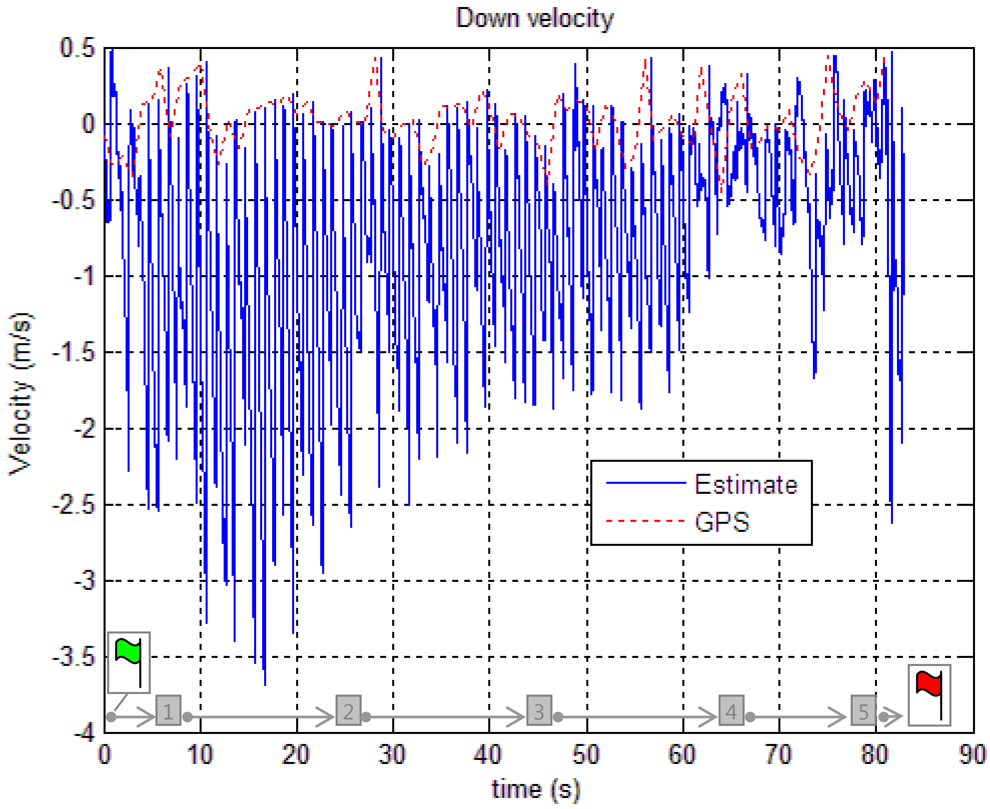


Figure 4.102. Flight test 5: Down velocity estimate.

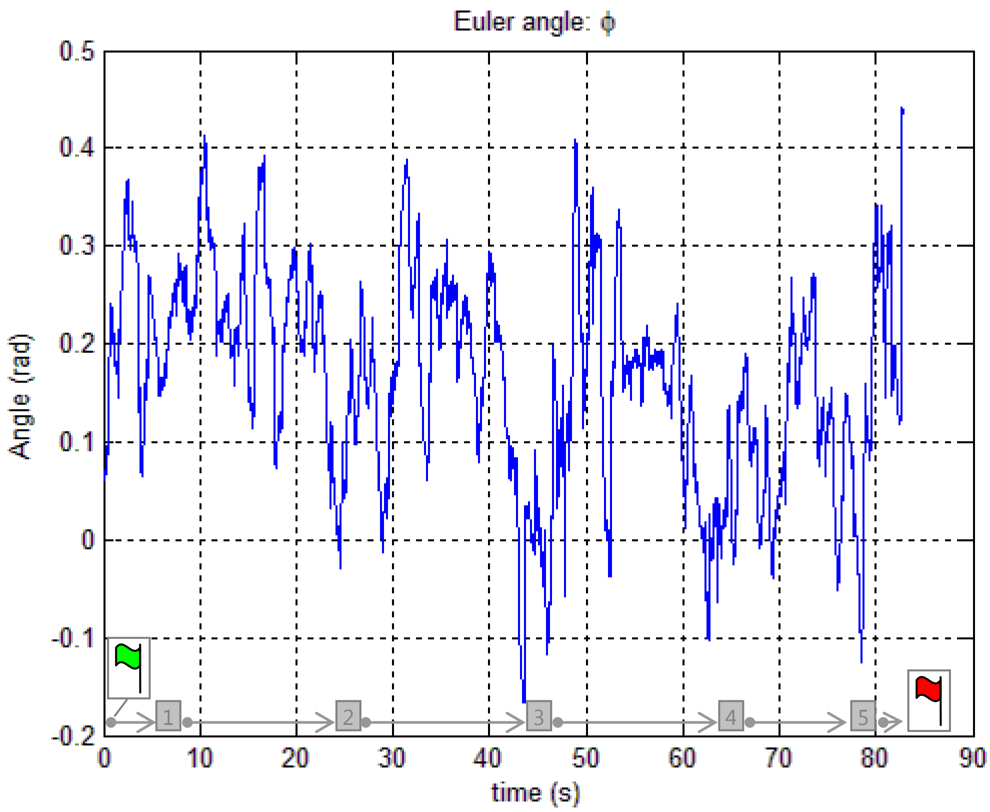


Figure 4.103. Flight test 5: Euler estimate for roll.

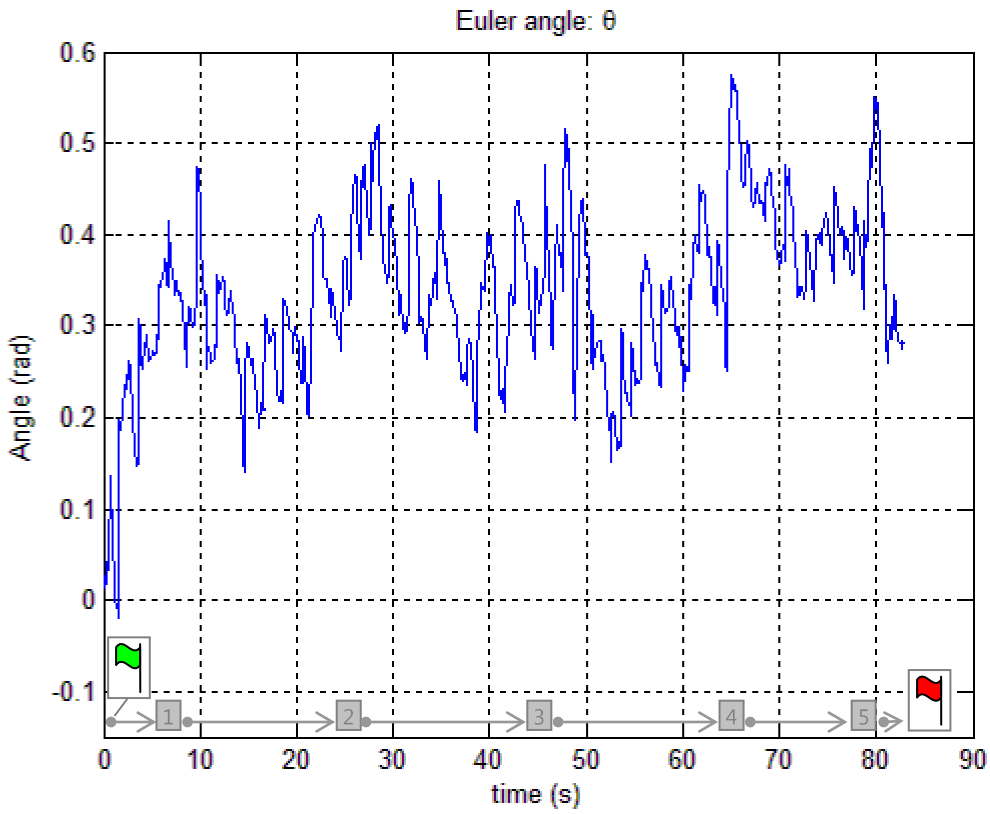


Figure 4.104. Flight test 5: Euler estimate for pitch.

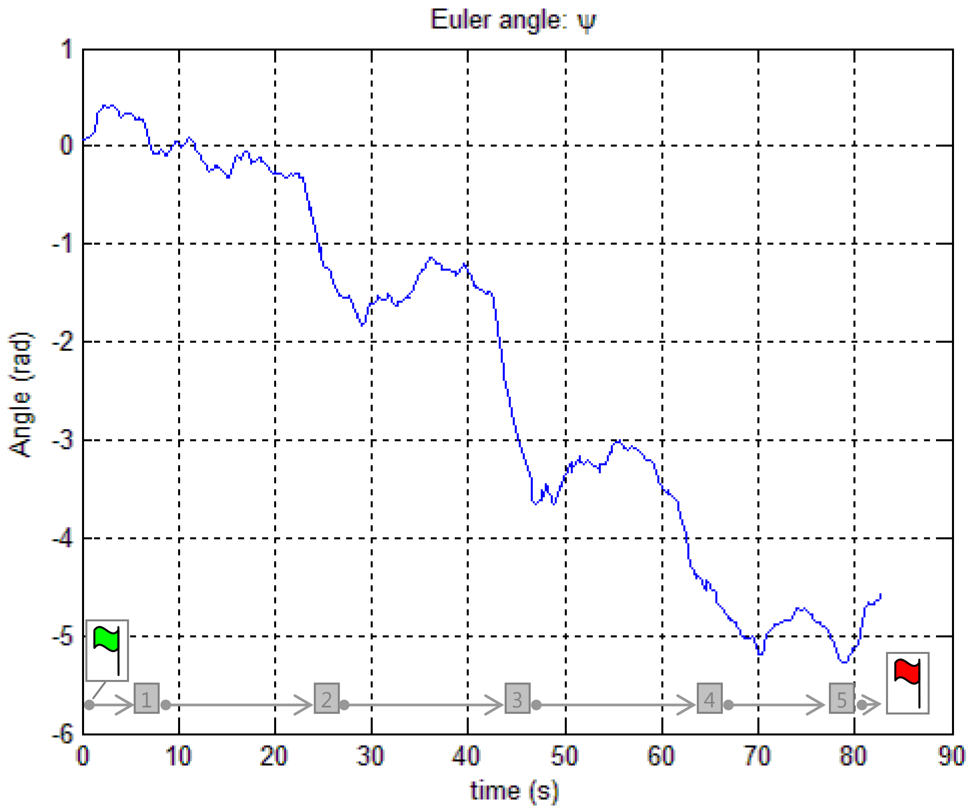


Figure 4.105. Flight test 5: Euler estimate for yaw.

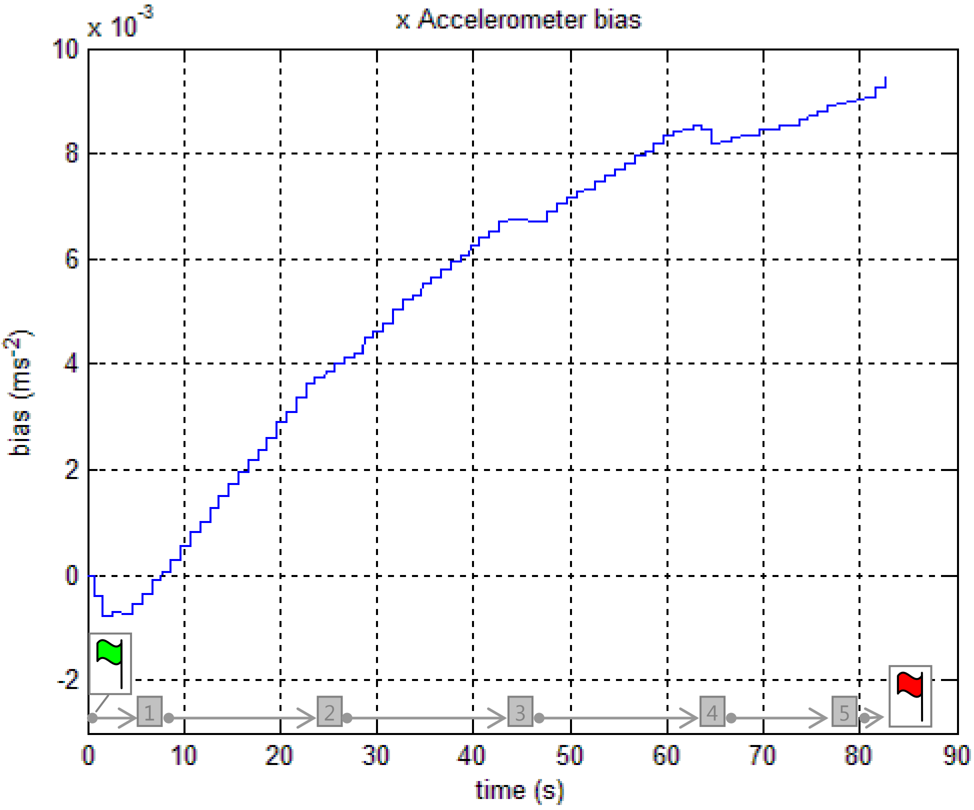


Figure 4.106. Flight test 5: Estimate for x accelerometer bias.

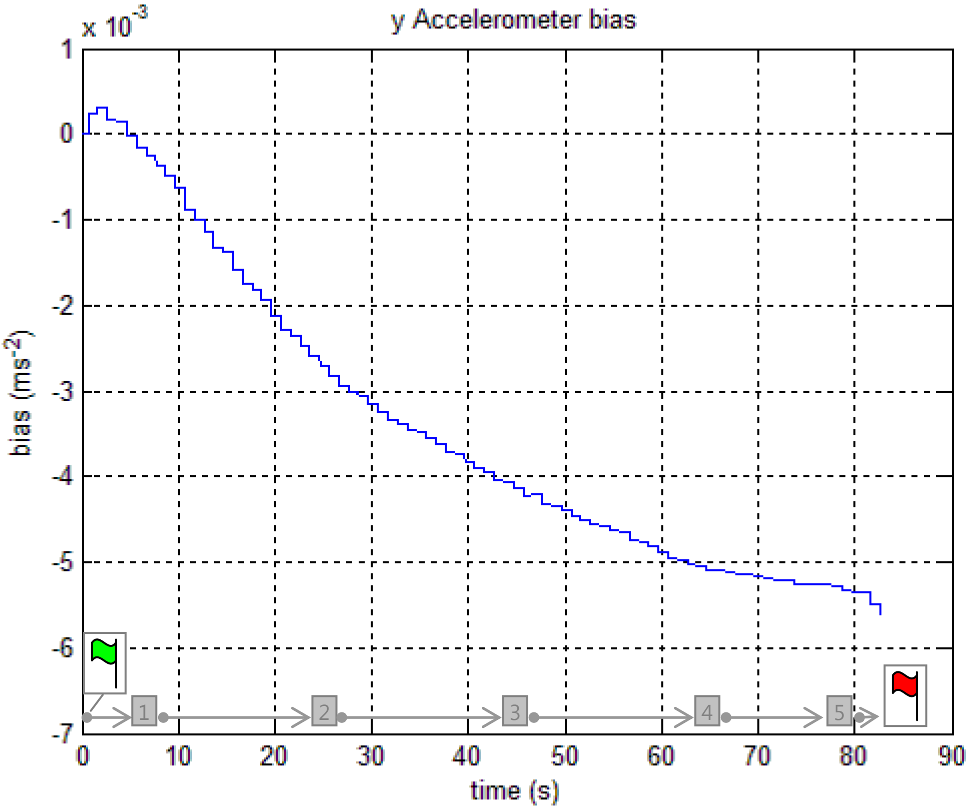


Figure 4.107. Flight test 5: Estimate for y accelerometer bias.

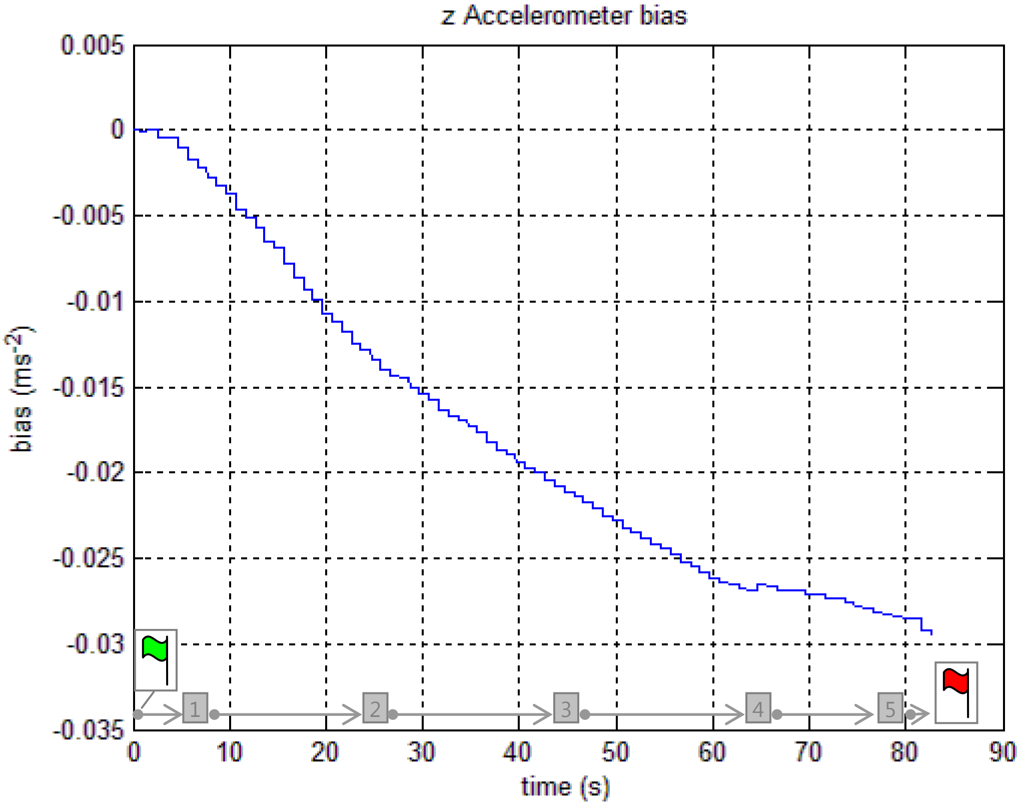


Figure 4.108. Flight test 5: Estimate for z accelerometer bias.

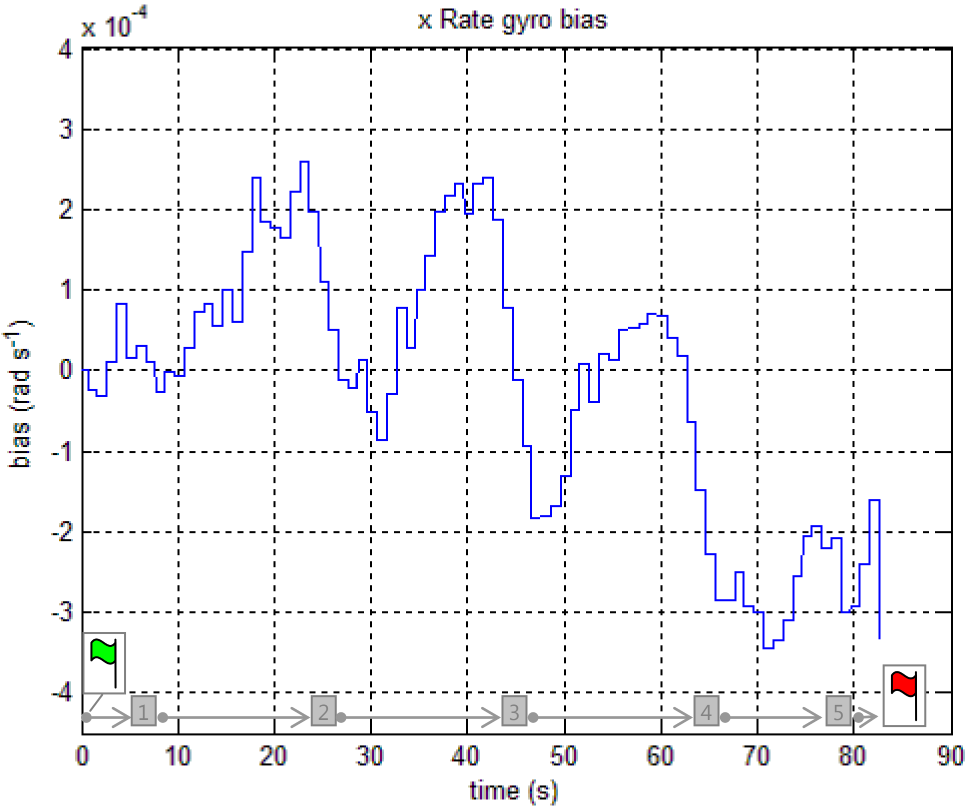


Figure 4.109. Flight test 5: Estimate for x rate gyro bias.

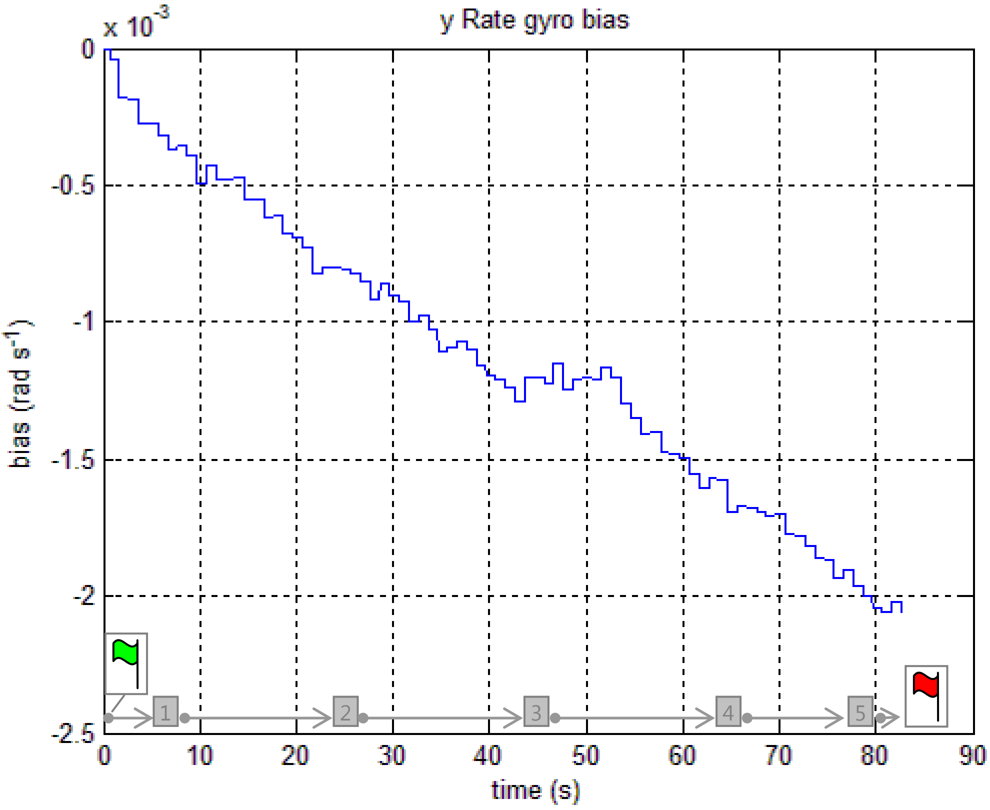


Figure 4.110. Flight test 5: Estimate for y rate gyro bias.

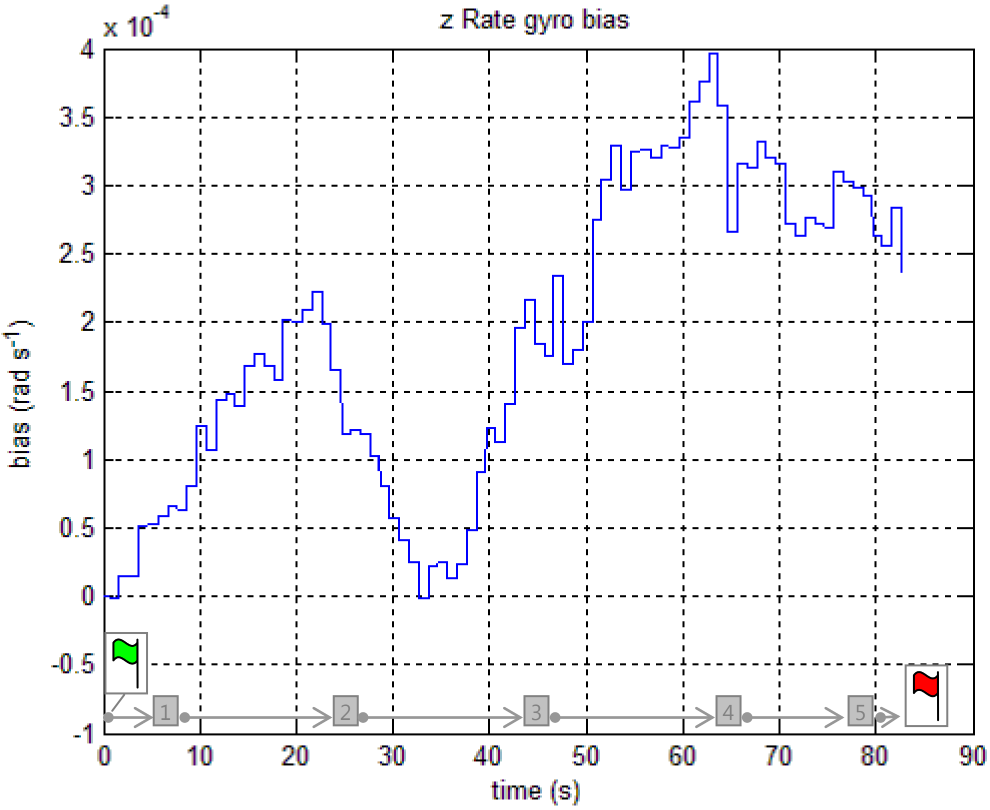


Figure 4.111. Flight test 5: Estimate for z rate gyro bias.

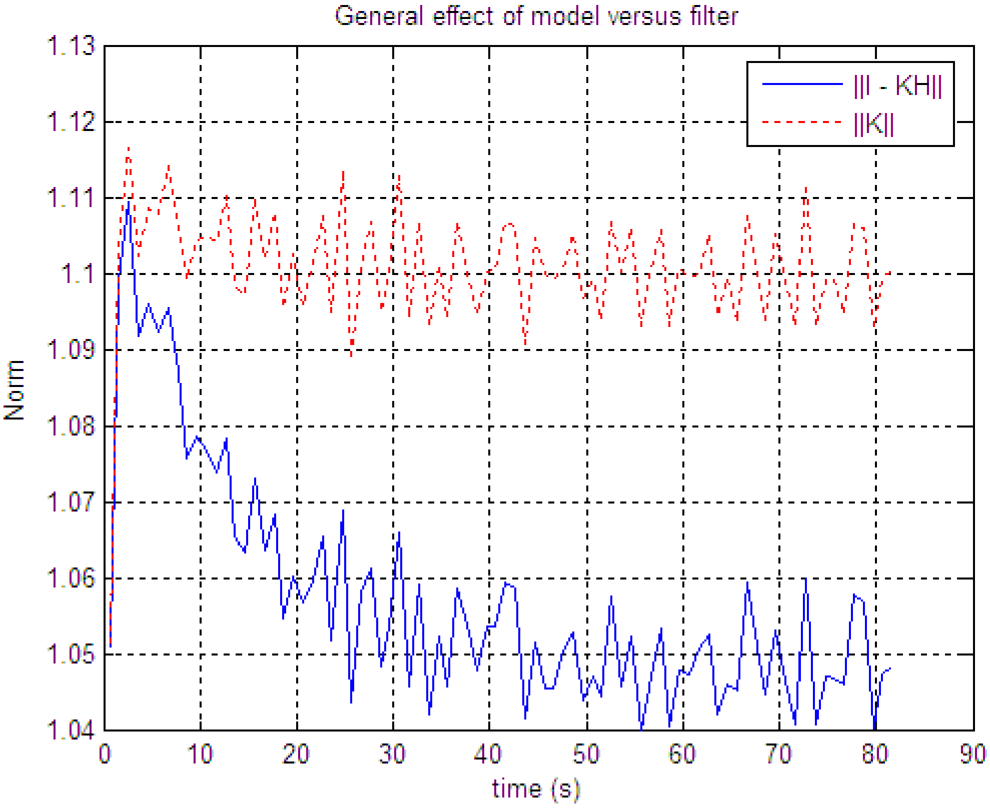


Figure 4.112. Flight test 5: General effect of model versus filter.

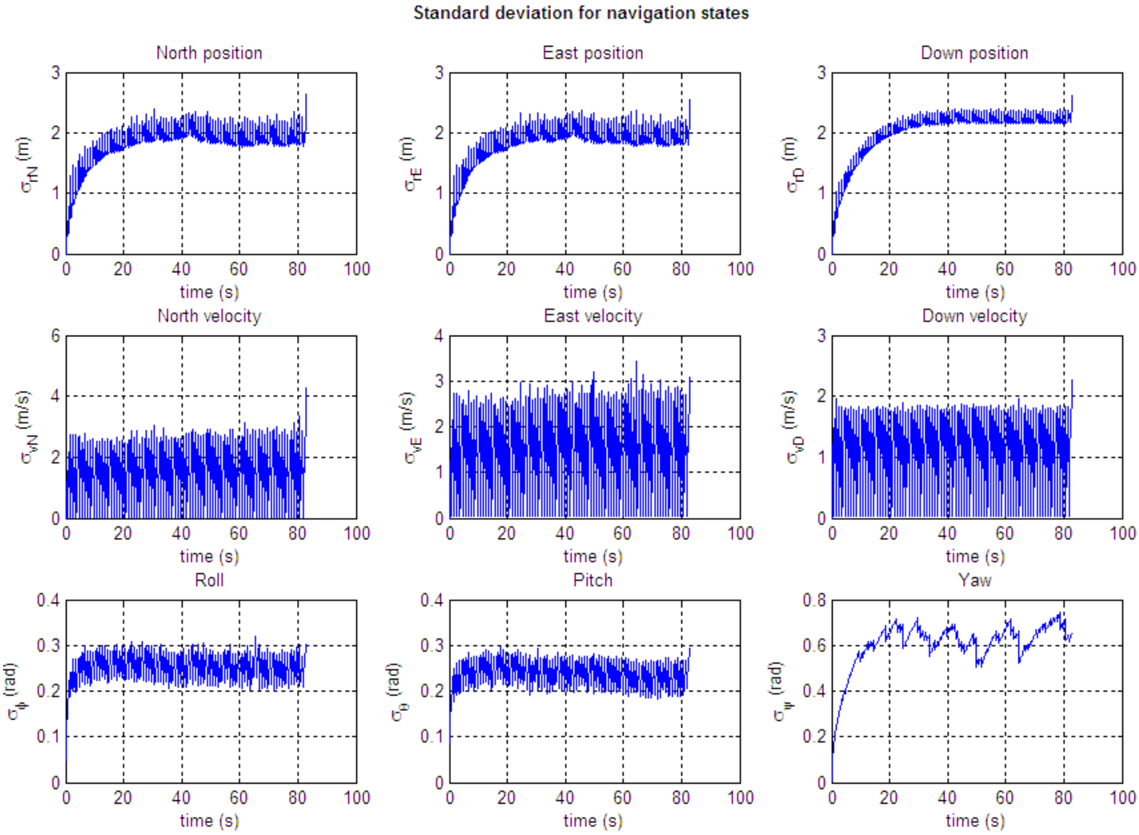


Figure 4.113. Flight test 5: Standard deviation of navigation estimates.

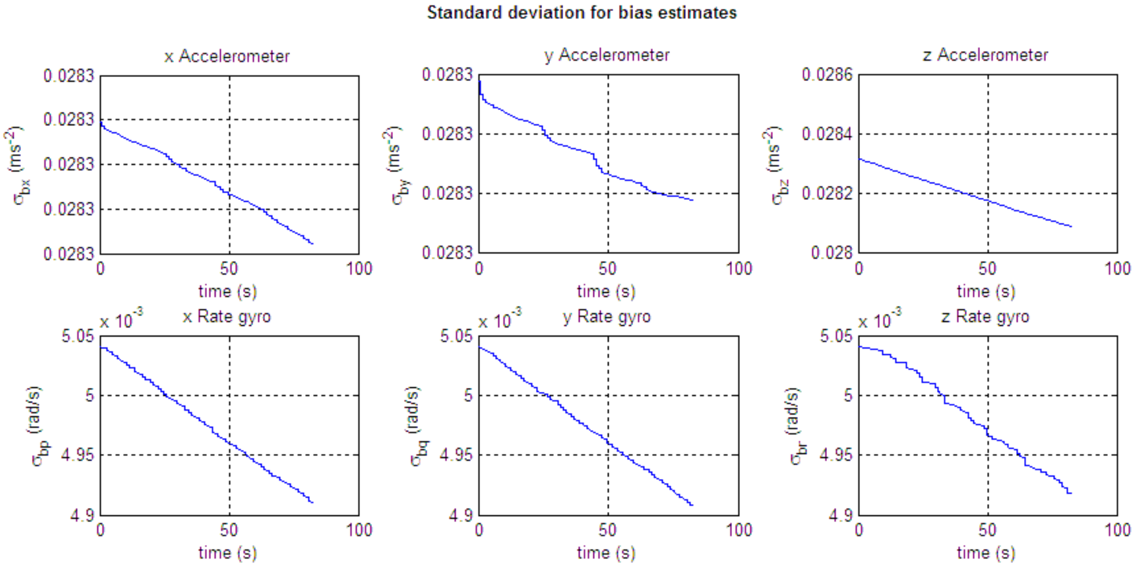


Figure 4.114. Flight test 5: Standard deviation of bias estimates.

4.5.6.4.6 Discussion

Position results from all flight tests showed results that tracked attempted flight paths. Flight test 3, having three vertical ascent-descent pairs, is well represented by Figure 4.61 which confirms the flight pattern. The altitude reflected in the down position plot (about 10.5 m from lowest to highest point) is acceptable when considering Figure 4.115, below, which shows an image capture from the video log of this flight where the helicopter (red circle) is indicated at about 11 m (The overlay grid spacing is 2 m).

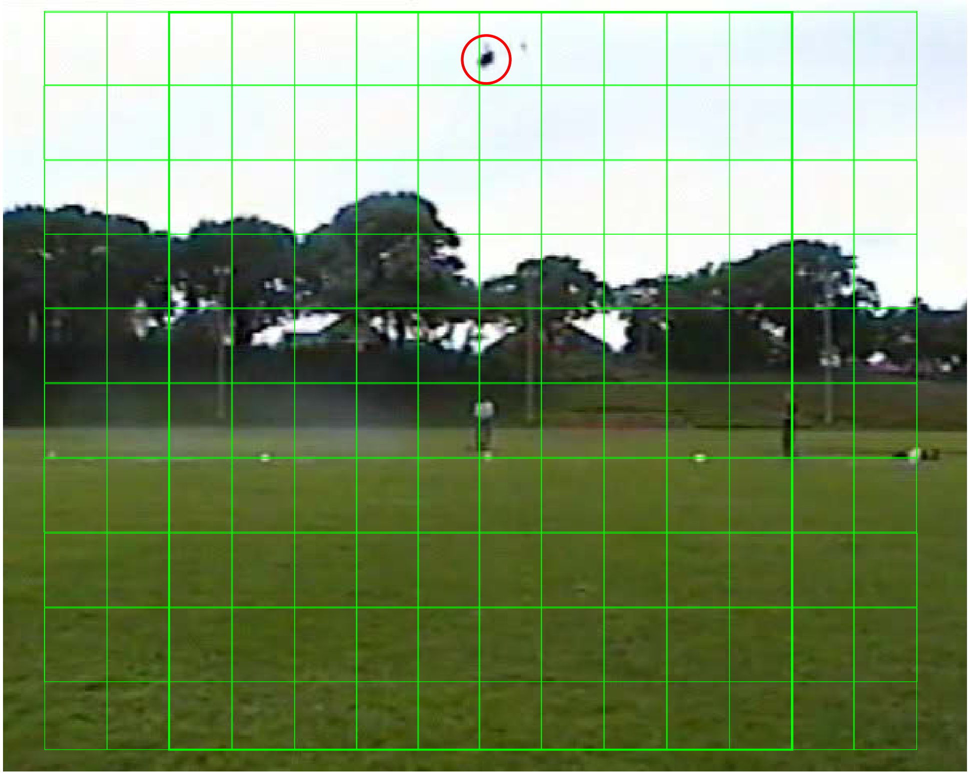


Figure 4.115. Altitude verification for flight test 3.

The North and East positions for flight test 3, should, ideally, have been zero, however, as can be seen from the video log (Appendix 3), the horizontal position of the helicopter was not maintained as required, nor was it landed at exactly the take off point, possibly leading to the metre or so mismatch in horizontal start and end positions.

In terms of flight tests 4 and 5, the horizontal position plots (Figure 4.80 and Figure 4.99) are most illuminating. These show the expected horizontal manoeuvres clearly. However:

1. For flight test 4, Figure 4.80 indicates a line length of about 28 m. Examination of the video footage (Appendix 3) shows that the end of the flight path was overrun, leading to the inconsistent length.
2. For flight test 5, the length of the sides of the square obtained was about 20 m each, with slight overruns again indicated by the video footage. The significant kink in the horizontal plot between way points 2 and 3 is clearly visible in the video log that shows the pilot struggling to control the helicopter during this time.

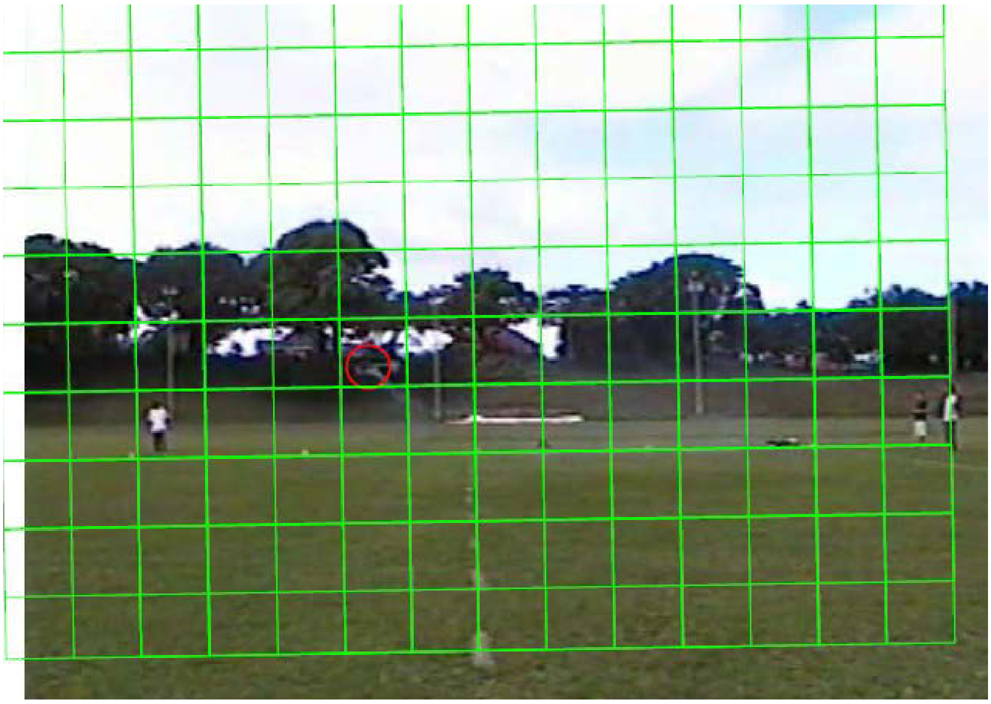


Figure 4.116. Capture highlighting altitude during flight test 4.

Figure 4.116 shows the maximum altitude achieved during flight test 4 as reconstructed from the video. As is evident, the attitude is between 2 and 3 m. This is roughly reflected in Figure 4.79, however, as anticipated, the GPS and altimeter inaccuracy are too large to affect accurate estimates in this channel. This problem manifested itself more obviously in flight test 5, where the down position estimate reaches 7 m (Figure 4.98), although the helicopter was never more than 3 m off the ground.

The velocity plots for all tests are difficult to analyse or draw conclusions from due to the significant inter measurement update growth of the estimate. Sudden, large changes in estimates at measurement update cycles are likely to be caused by the low GPS measurement error covariances causing GPS velocities to dominate predicted velocities. This was expected due to the low GPS velocity measurement error covariance.

Pitch and roll estimates for all flights can only be validated by considering that the manoeuvres attempted were benign, and, as such, it was expected that the helicopter would never pitch or roll greater than ±45° (≈ ±0.8 rad). This is evident in the relevant plots for flight tests 3 to 5. In terms of yaw, flight tests 4 and 5 feature acceptable results for yaw estimates under certain conditions. Based on the layout of the flight test field and flight paths, test 4 showed that the heading along the test varied between 325° and 0°. A satellite photograph of the test field was used to confirm this. Additionally, the yaw result for flight test 5 shows three turns of approximately 90° - indicative of the square that was followed.

The yaw for flight test 3 and for the beginning and end periods of flight tests 4 and 5, however, is dubious. This uncertainty is likely related to the observability of the problem, insight into which can be gained by examining the observability matrix which is generated by Equation 4.99.

$$\mathbf{O}(\partial\mathbf{f}/\partial\mathbf{x},\mathbf{H}) = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\cdot(\partial\mathbf{f}/\partial\mathbf{x}) \\ \mathbf{H}\cdot(\partial\mathbf{f}/\partial\mathbf{x})^2 \\ \vdots \\ \mathbf{H}\cdot(\partial\mathbf{f}/\partial\mathbf{x})^{14} \end{bmatrix} \qquad 4.99$$

$\mathbf{O}(\partial\mathbf{f}/\partial\mathbf{x},\mathbf{H})$, is not full rank at any computational iteration indicating that, for the present system and measurement setup, some states are unobservable. Hong et al. (2005) show that insight into observability of a GPS/INS problem may be gained by performing a covariance analysis. This involves juxtaposing standard deviations of state estimates with acceleration and angular velocity inputs. Figure 4.117 and Figure 4.118 show filtered accelerations and angular velocities for flight test 4 (for which the covariance method presented in Hong et al., 2005, will be attempted here). To aid with such analysis:

1. The accelerations and angular velocities logged during flight test 4 have been filtered in Matlab. Unfiltered data is characterised by significant noise and vibration making any visual analysis onerous.
2. The initial state estimate covariances have been adjusted such that the $\mathbf{P}_0^- = [10\,10\,10\,10\,10\,10\,2\,2\,2\,2\,2\,2\,2\,2\,2]$. This was done so that clear trends in the standard deviation plots could be readily identified.

The filter was then run, offline, in Matlab, to obtain the results below with flight test 4 IMU, GPS and altimeter data supplied to the algorithms.

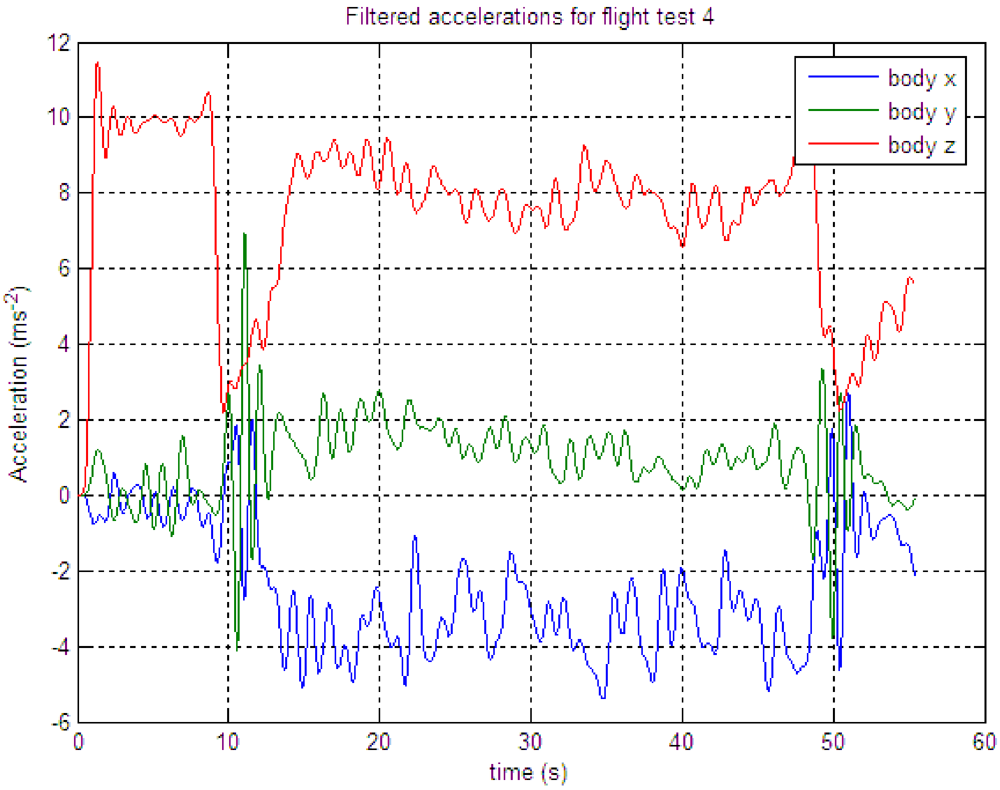


Figure 4.117. Accelerations for flight test 4.

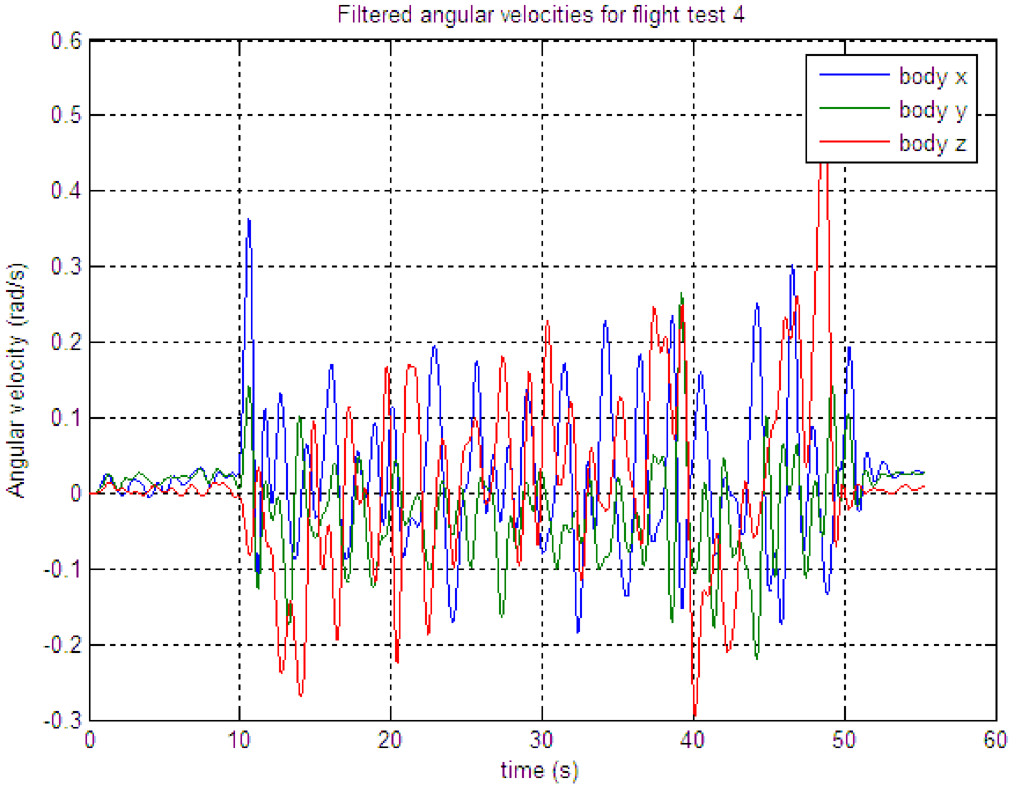


Figure 4.118. Angular velocities for flight test 4.

The standard deviation for the yaw estimate for flight test 4 is plotted below. As is evident, initially, the yaw estimation standard deviation increases indicating poor observability. However, at approximately 10 s the standard deviation begins to reduce rapidly. This concurs with the results from Hong et al. (2005), which essentially illustrates that the yaw becomes observable provided there is some horizontal specific force (consider Figure 4.117 where, at about 10 s, the body x and y experienced acceleration).

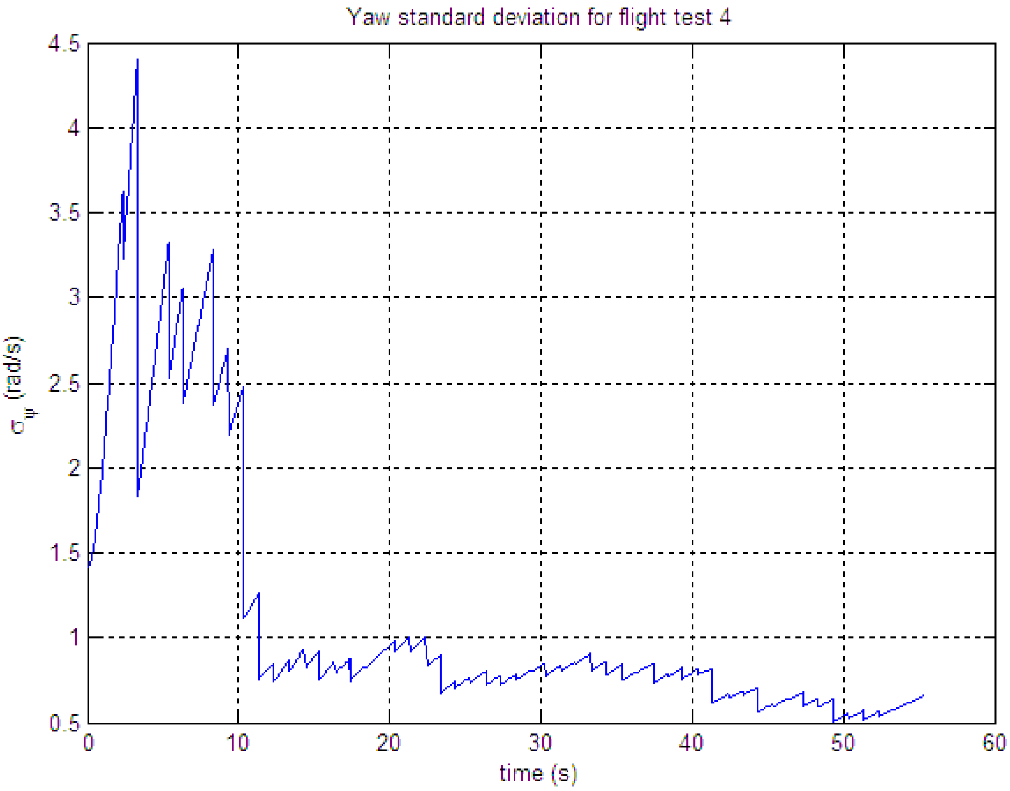


Figure 4.119. Yaw standard deviation from flight test 4.

Other variables of interest in terms of this standard deviation analysis are presented in Figure 4.120 and Figure 4.121 that illustrate the standard deviations for bias states.

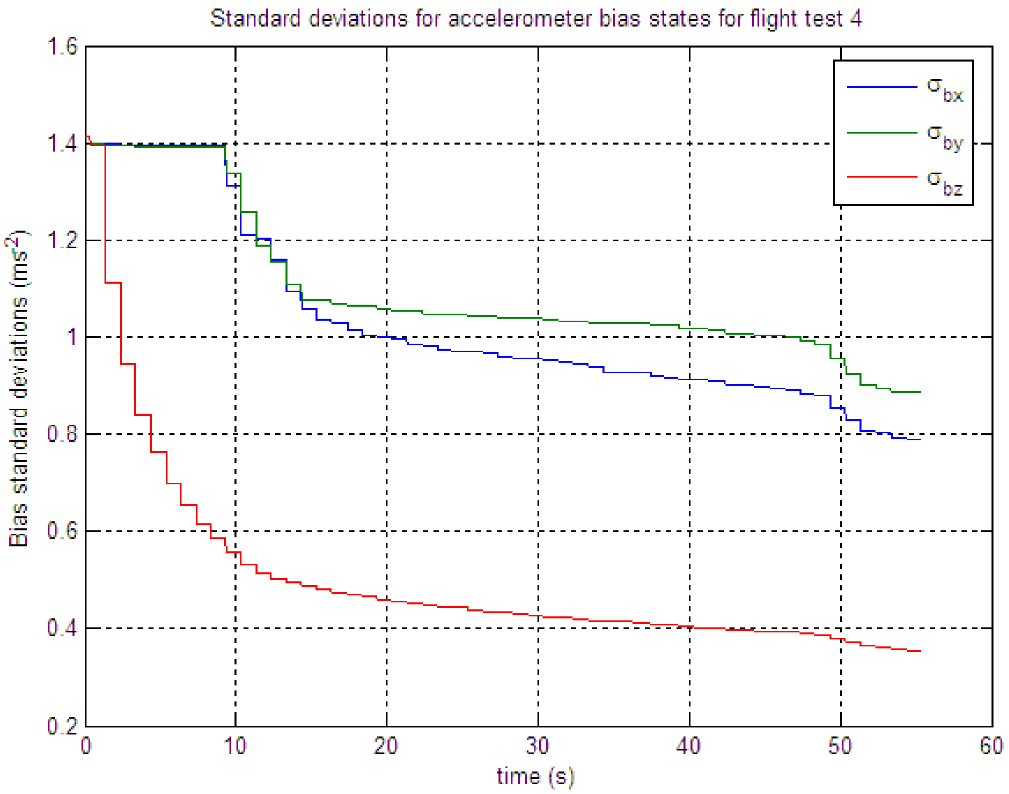


Figure 4.120. Accelerometer bias standard deviations.

Horizontal accelerometer biases, as mentioned by Hong et al. (2005) and Rhee et al. (2004), are poorly observable as it is difficult to disambiguate horizontal accelerometer biases from horizontal tilt angle (pitch and roll) errors. Further, as in Hong et al. (2005) and Rhee et al. (2004), Figure 4.120 confirms that the vertical component of accelerometer bias is always observable (rapid reduction of standard deviation) possibly due to the presence of a force (gravity) in that channel for most of the flight. Figure 4.121 further agrees with the propositions of Hong et al. (2005) with respect to horizontal gyro bias components that appear observable even in the absence of motion. The vertical gyro bias component should only become observable with simultaneous changes in specific force and attitude, however, from the data available, this cannot be verified.

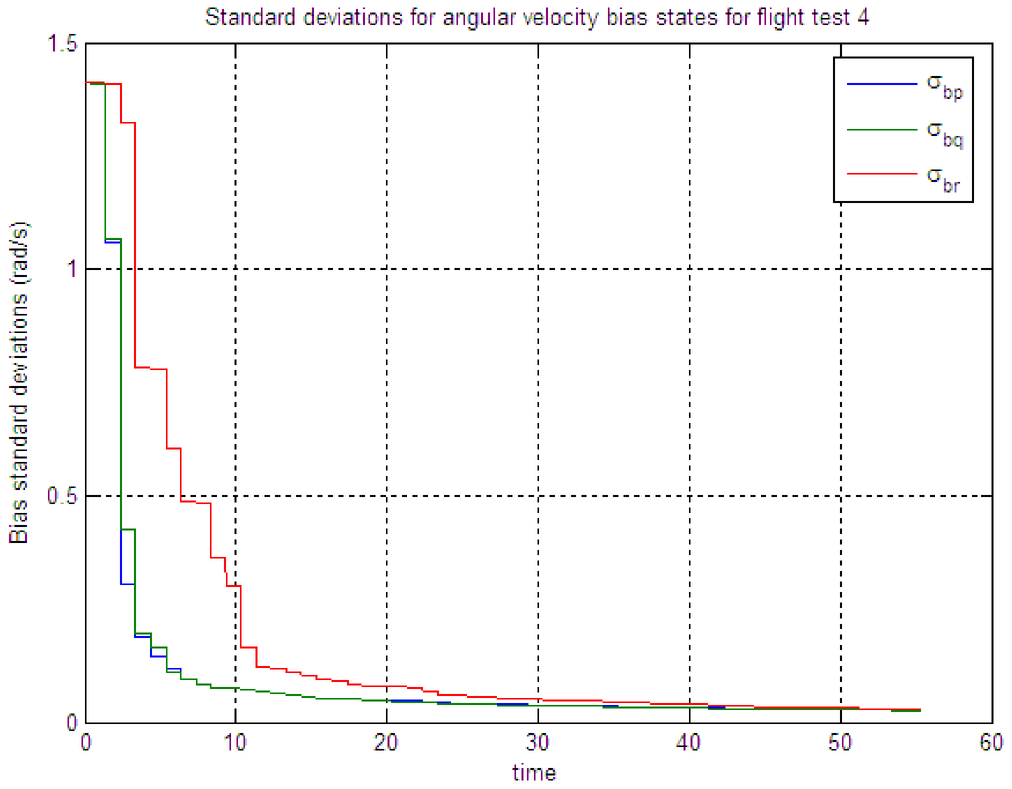


Figure 4.121. Angular velocity bias standard deviations.

Ham and Brown (1983) present a method based on the eigenvalues and eigenvectors of the normalised covariance matrix for determining the degree of observability. More specifically, it is possible for individual states to be unobservable, whilst linear combinations of states are observable. The proposed method assists in finding such combinations, however, upon application, in Matlab, no conclusive results were obtained (which is a distinct possibility noted by the authors).

As was evident from the initial filter tests (Section 4.5.5.2), the navigation solutions feature periods of prediction and correction (when measurements were applied). The inter measurement update divergence of the estimates was partly attributed to vibration of the helicopter due to the running engine and rotating blades. Analysis of the frequency spectrum of the acceleration and angular velocity data fed into the filter can facilitate some testing of this hypothesis. As an example, this was attempted for the data of flight test 4. It was anticipated, due to the benign nature of the flight manoeuvres, that the maximum sensor bandwidth required would be between 5 and 10 Hz. Figure 4.122 shows the power spectral density for all acceleration and angular velocity channels.

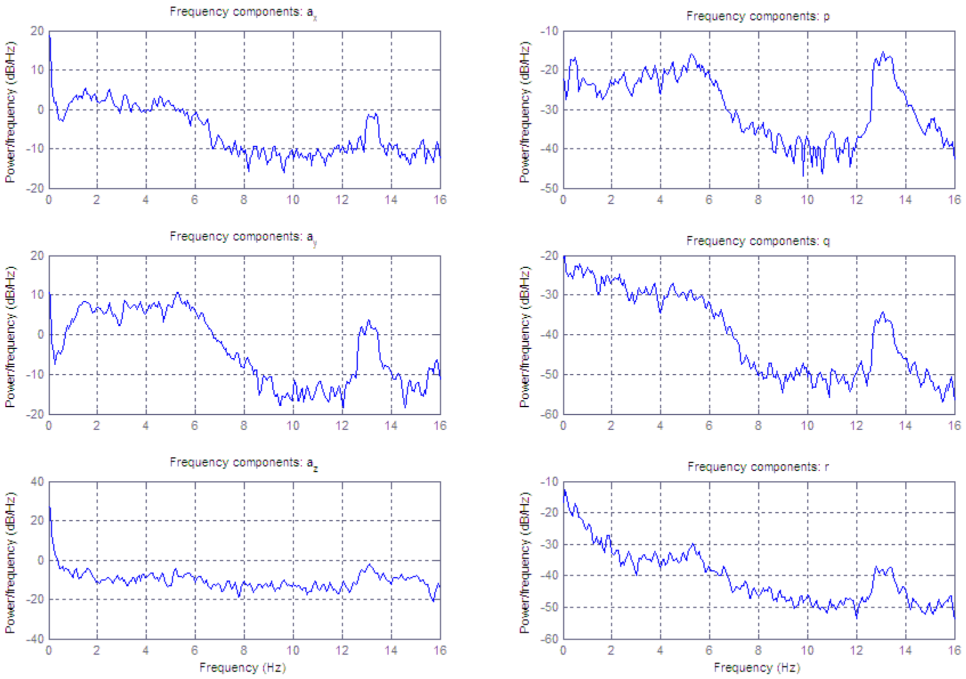


Figure 4.122. Power spectral density for IMU data.

Below about 8 Hz, the frequency distribution indicated by Figure 4.122 is consistent with the gentle motion of flight test 4. However, anomalies are present at about 12.5 Hz, where all channels have significant signal power. The main rotor frequency is 25 Hz and the tail rotor frequency is 100 Hz (Table 2.1). It is therefore likely that vibration steming from these is aliasing into the inertial sensor measurement at 12.5 Hz. Filtering, as mentioned in Section 4.5.5.2, is difficult due to the non linear dynamics and, the fact that the IMU bandwidth is unspecified, makes aliasing due to inadequate filtering a distinct possibility.

The behaviour of the estimates between measurement updates is also of concern when considering any future automatic flight control algorithm. Typical control schemes for this type of problem would require a sampling rate of approximately 10 Hz implying that these must rely on the IMU data in between measurement updates. For position estimates, the IMU data and model appears to be capable of propagating the state, between measurement updates, to within approximately 1 m of the updated estimate. For velocity, the situation is slightly worse, fundamentally due to the fact that measurements are "trusted" far more than the a priori estimates as the GPS velocity measurement error covariances are extremely low. From the curves presented, however, it should be clear that the estimates are not merely the measurement data, and that the IMU data propagates the states. Such propagation, however, must only be considered valid for short periods of time as regular measurement updates are essential to bounding the error growth of the navigation solution. In the continued absence of measurement updates, said growth is indeed too rapid, rendering this IMU alone impractical for the provision of accurate navigation data. The inability of the INS to operate adequately with the IMU only, is indeed characteristic of navigation systems integrated with low cost IMUs.

Naturally, even in the low cost MEMs IMU arena, devices of differing qualities exist which could possibly allow the INS to tolerate measurement update absences for longer periods of time (but still not facilitate inertial sensor only navigation). Typically, these devices feature superior bias and noise performance e.g. Juxtapose the MMQ50 IMU, manufactured by Systron Donner, having rate gyro bias of 200 °/hr and accelerometer bias of 3 mg against the 'BP3010. In terms of the position estimates, 1 s of prediction, at the 'BP3010 3σ accelerometer noise of 0.96 ms^{-2}, would theoretically produces position deviations of 0.48 m whilst the same scenario with the MMQ50 (accelerometer noise of 0.0155 ms^{-2} at 10 Hz) leads to 0.0077 m. Although this is a rather simplistic view, that neglects other errors, it serves to illustrate that a superior quality device can improve the system. Better quality devices, however, were available at a cost premimum that would have impacted the project budget far more severely.

5 Conclusion

The fundamental result of the research work documented herein was the development of an operational integrated navigation system based on the aiding of an IMU with external data (e.g. GPS) via the Kalman filter. The developed navigation system features acceptable performance especially considering that low cost sensors were employed. The approximate total project expense (excluding model helicopter) including DSP, MEMS IMU and other sensors and hardware was ZAR 25 000. With respect to the navigation system design, the development process documented herein is complete. It includes navigation system structure and high level overviews, navigation models and state mechanisation equation development, Kalman filtering application, simulation and real world testing. Some of navigation results obtained, however, are non ideal and these are perhaps a function of the quality of sensors employed (e.g. rapid growth in state errors in the absence of measurement updates).

The other outcome of the project is the DSP based avionics system incorporating all sensors and communications devices. The avionics were custom designed to support the navigation system and eventually autonomous flight of the helicopter. As such, the helicopter hardware and software can communicate with multiple sensors over several standard interfaces, actuate the helicopter control surfaces and has the computational bandwidth to process the current navigation algorithms and any future unmanned control schemes. With respect to improvements, naturally, higher quality inertial sensors are desirable. A more accurate means of altitude determination is also essential for better readings in the down position channel and, with autonomous flight, for near ground manoeuvres. It may, for the purposes of further testing of the navigation system, be beneficial to fly higher, although, this renders following a marked path difficult for a human pilot. Additionally, through flight testing, the Bluetooth based wireless transceivers behaved erratically and should possibly be replaced with more robust devices – preferably ones with omni directional antenna and larger range. Other useful additions to the sensor suite may be a three axes magnetic compass to provide heading data and a multi antenna GPS capable of resolving tilt.

The long term objective of the project is, naturally, achieving automonous flight of the model helicopter. Future steps to this end could include the development of a dynamic model for the helicopter which could be validated on the platform developed herein. Following which the validated model can be used in an investigation into competing methods for robust controller design and eventual testing of several control schemes on the instrumented test platform developed.

6 References

AMCE; 2003, *"Navstar global positioning system surveying"*, p4-4 to 4-7, p4-14, US Army corps engineering, Washington.

Angel J and de La Parra S; 2005, *"Low cost navigation system for UAV's"*, Aerospace science and technology, No. 9, p505.

Bar-Shalom Y, Xiao-Rong L and Thiagalingam K; 2001, *"Estimation with applications to tracking and navigation: Theory, algorithms and software"*, p524, Wiley-IEEE.

Beauregard, RA and Fraleigh JB; 1995, *"Linear algebra"*, 3rd ed., p350/1, p533, Addison Wesley, United States of America.

Bluetooth SIG; 2004, *"Specification of the Bluetooth system"*, Vol. 2, p29, p31, Available from http://www.bluetooth.org.

Britting, KR; 1971 *"Inertial navigation systems analysis"*, p1-37, Wiley Interscience, United States of America.

Brown RG and Hwang PYC; 1992, *"Introduction to random signals and applied Kalman filtering"*, 2nd ed., p265-267, p366-370, John Wiley and Sons, United States of America.

Cramer M; 1997, *"GPS/INS Integration"*, p2-6, Photogrammetric Week '97, Heidelberg.

Durrant-Whyte H and Nebot EM; 1999, *"A high integrity navigation architecture for outdoor autonomous vehicles"*, Robotics and autonomous systems, No. 26, p82.

Farell J and Barth M; 1999, *"The global positioning system and inertial navigation"*, p10, 11, 15, 16, 24, 246, McGraw Hill, United States of America.

Finney RL, Weir MD and Giordano FR; 2001, "Thomas' calculus", 10th ed., p948-954, Addison Wesley Longman, United States of America.

Gerald CF and Wheatley PO; 1999, "Applied numerical analysis", 6th ed., p126-133, Addison Wesley Longman, United States of America.

Grewal MS and Andrews AP; 2001, *"Kalman filtering theory and practice using Matlab"*, 2nd ed., p1/2/114/115, p121-123/127, p170/171/175/180/209/345/346, John Wiley and Sons, United States of America.

Grewal MS, Andrews AP and Weill LR; 2001, *"Global positioning systems, inertial navigation, and integration"*, p12/18-24/156/163/164/255/256/338, John Wiley and Sons, United States of America.

Ham FM and Brown RG; 1983, *"Observability, eigenvalues and Kalman filtering"*, IEEE transactions on aerospace and electronic systems", Vol AES-19, No 2, p269-273.

Honeywell International Inc.; 2006, *"HG1700 Inertial Measurement Unit"*, p2.

Hong S, Lee MH, Chun H, Kwon S and Speyer JL; 2005, *"Observability of error states in GPS/INS integration"*, IEEE Transactions on vehicular technology, Vol 54, No 2, p731-743.

Intersema Sensoric SA; 2005, *"MS5534B Barometer module"*, p5-8.

Kaliath T, Sayed AH and Hassibi B; 2000, "Linear estimation", Prentice Hall Inc., United States of America.

KC Wirefree LLC; 2004, "KC111 Bluetooth serial adapter datasheet", p1.

Kim JH, Wishart S and Sukkarieh S; 2006, *"Real time navigation, guidance and control of a UAV using low cost sensors"*, Australian Centre for Field Robotics, University of Sydney, Australia.

Levy LJ; 2002, *"The Kalman filter: Navigation's integration workhorse"*, p1-4, p8/9, http://www.cs.unc.edu/~welch/kalman/Levy1997/index.html.

Lorimer, T; 2006, *"Mathematical modelling"*, p1-50

Maybeck PS, 1999; *"Stochastic models, estimation and control"*, Vol 1, p4-7, Academic Press, United States of America.

Merhav S, 1996; *"Aerospace sensor systems and applications"*, p317/320, Springer, United States of America.

Padfield, GD; 1995 *"Helicopter flight dynamics: The theory and application of flying qualities and simulation modelling"*, p27, p30/31, p33-35, ,p99/100, p107, p147/148, p177/178, AIAA education series, Great Britain.

Philips Semiconductors; 1998, *"KMZ51 Magnetic field sensor"*, p2-4.

Press WH; 1992, *"Numerical recipes in C: The art of scientific computing"*, 2nd ed., p61, Cambridge University Press, Great Britain.

Rhee I, Abdel-Hafez M, Speyer JL; 2004, *"Observability of integrated GPS/INS during manoeuvres"*, IEEE transactions on aerospace and electronic systems, Vol 40, No 2, p526-535.

Salychev OS; 1998 *"Inertial systems in navigation and geophysics"*, p43, Bauman MSTU Press, Moscow.

Texas Instruments; 2006, *"C2000 Controllers Architecture & Peripherals from Texas Instruments"*, http://focus.ti.com/en/graphics/dsp/general/dspdetails/c28x_bd.jpg.

Texas Instruments; 2003, *"TMS320x281x, 280x DSP peripheral reference guide"*, p1-12.

Trimble Navigation Limited; 2004, *"Lassen iQ GPS module low-power, high-quality GPS solution for your mobile products"*, p1/2.

Trommer GF and Wendel J; 2004, *"Tightly coupled GPS/INS integration for missile applications"*, p627/628, Aerospace science and technology.

Webster JG; 2004, *"Electrical measurement, signal processing and displays"*, p12-19, 12-20, CRC Press, United States of America.

Welford BP; 1962, *"Note on method for calculating corrected sums of squares and products"*, Technometrics, Vol. 4 No. 3, p419/420.

Weston JL and Titterton DH; 2000, *"Modern inertial navigation technology and its applications"*, p49-53, Electronics and communication engineering journal.

Wilkinson JH; 1965, *"The algebraic eigenvalue problem"*, p61, Clarendon Press, Great Britain.

Young, HD, Freedman RA, Sandin TR and Lewis Ford A; 1996 *"University Physics"*, 9th ed., p503, p269, Addison Wesley Publishing Company Inc., United States of America.

7 Bibliography

Blewitt G; 1997, *"Basics of the GPS technique: Observation equations"*, Appears in text "Geodetic applications of GPS", Swedish Land Survey.

Caron F, Duflos E, Pomorski D and Vanheeghe, P; 2004, *"GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects"*, Information fusion.

Eissfeller B and Spietz P; 1989, *"Basic filter concepts for the integration of GPS and an inertial ring laser gyro strapdown system"*, Manuscripta geodaetica, No. 14, p166-182.

Hollasch S; 2005, *"IEEE Standard 754 floating point numbers"*, http://steve.hollasch.net/cgindex/coding/ieeefloat.html.

Kailath T; 1974, *"A view of three decades of linear filtering theory"*, IEEE Transactions on information theory, Vol. IT-20, No. 2, p146-174.

Kim JH and Sukkarieh S; 2003, *"A baro-altimeter augmented INS/GPS navigation system for an uninhabited aerial vehicle"*, Presented at: The 6th international symposium on satellite navigation technology including mobile positioning and location services, July 2003, Australia.

Maciejowski JM; 1989, *"Multivariable feedback design"*, Addison-Wesley Publishing, Great Britain.

O' Hara M; 1998, *"EMC at component and board level"*, Newnes.

Ogata K; 2002, *"Modern control engineering"*, 4th ed., Prentice-Hall, Inc., United States of America.

Omerbashich M; 2002, *"Integrated INS/GPS navigation from a popular perspective"*, Journal of air transportation, Vol. 7, No. 1, p103-118.

Salychev OS, Voronov VV, Cannon ME, Nayak R and Lachapelle; 2000, *"Low cost INS/GPS integration: Concepts and testing"*, Presented at the Institute of navigation: National technical meeting.

Schimel BD and Grantham WJ; 1995, *"The engineering handbook"*, CRC Press, United States of America.

Sorenson HW; 1970, *"Least squares estimation: from Gauss to Kalman"*, IEEE Spectrum, July 1970, p63-68.

Wang CJ; 1999, *"Controllability and observability of linear time varying singular systems"*, IEEE Transactions on automatic control, Vol. 44, No. 10, p1901-1905.

Appendix 1. Avionics hardware schematics

Flight switch

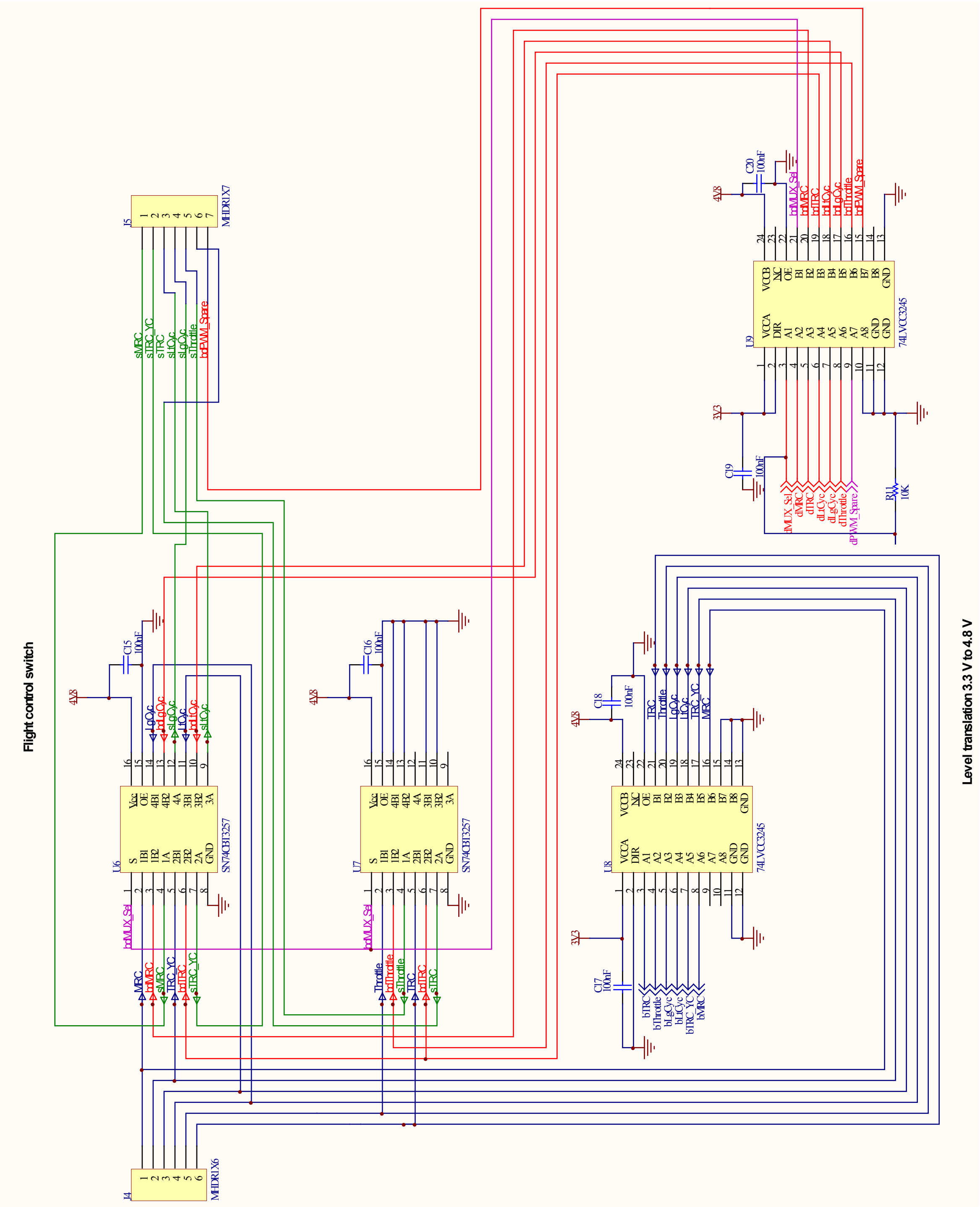


Figure A1.1. Hardware schematic for flight switch.

IMU Interface

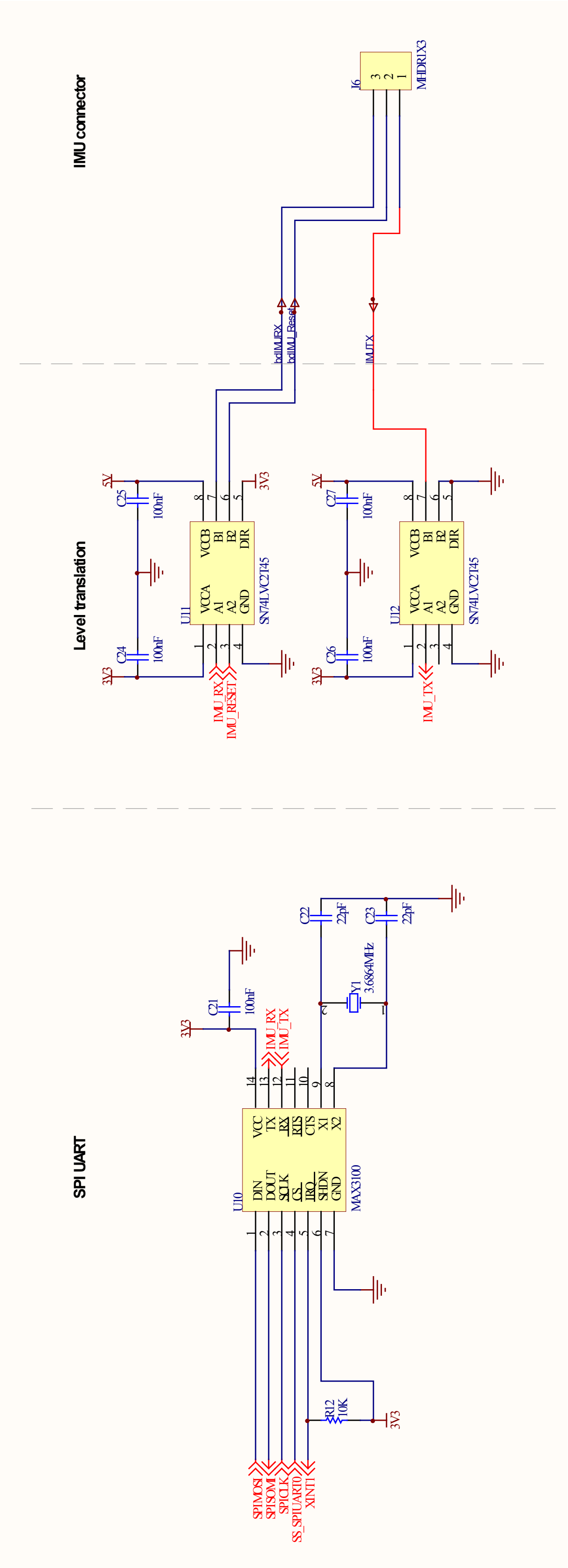


Figure A1.2. Hardware schematic for the IMU interface.

GPS Receiver interface

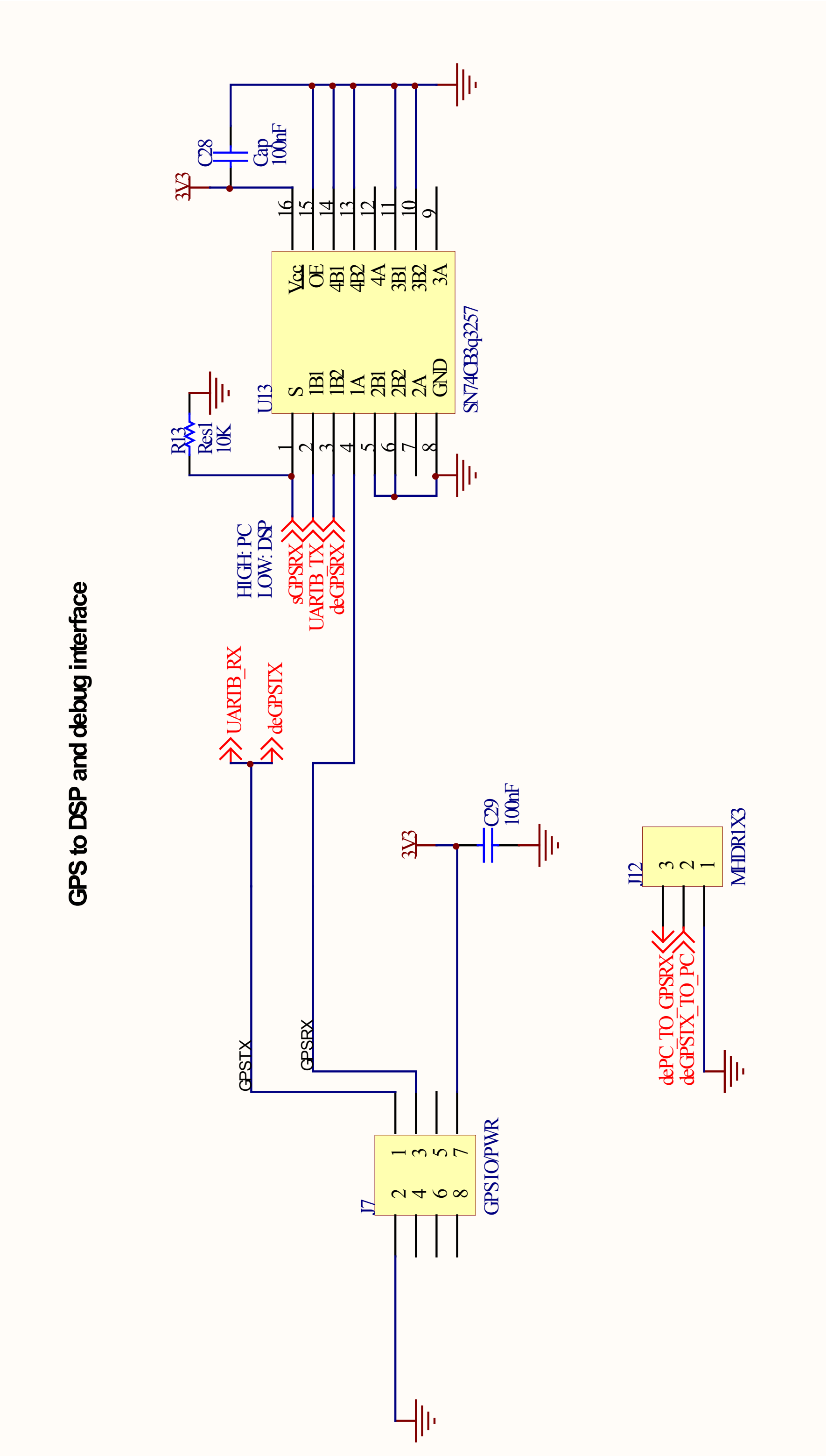


Figure A1.3. Hardware schematic for the GPS receiver interface.

Wireless transceiver: RS232 interface

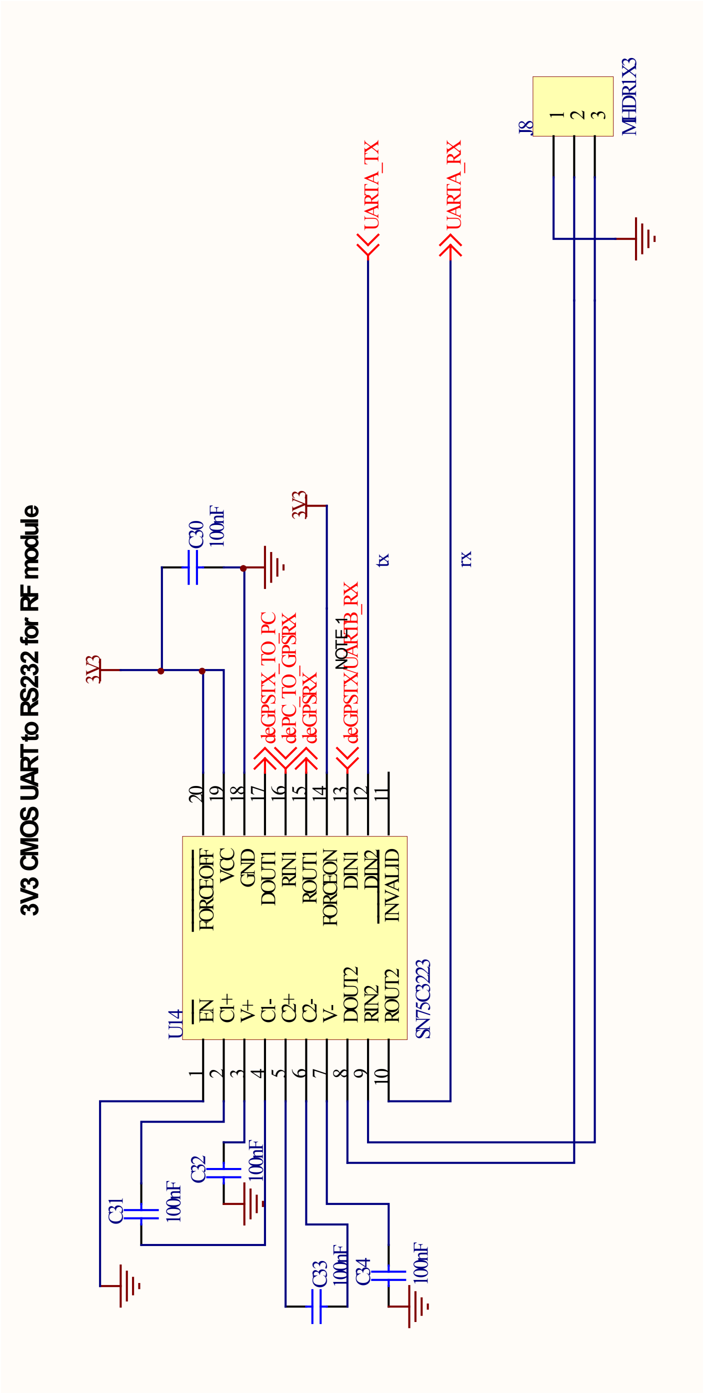


Figure A1.4. Hardware schematic for the RS232 interface for the KC 111.

Magnetic compass interface and support circuitry

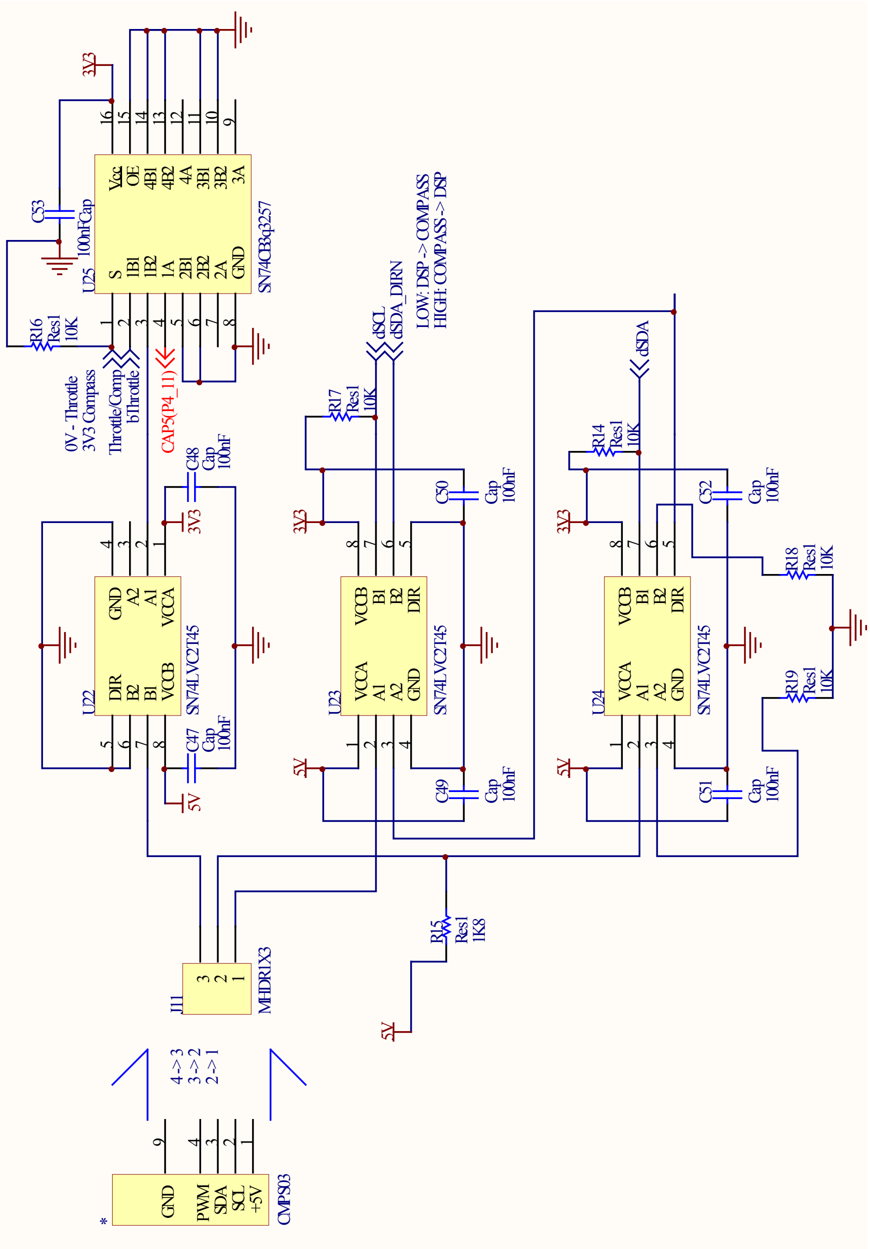


Figure A1.5. Hardware schematic for the magnetic compass interface.

SPI Support

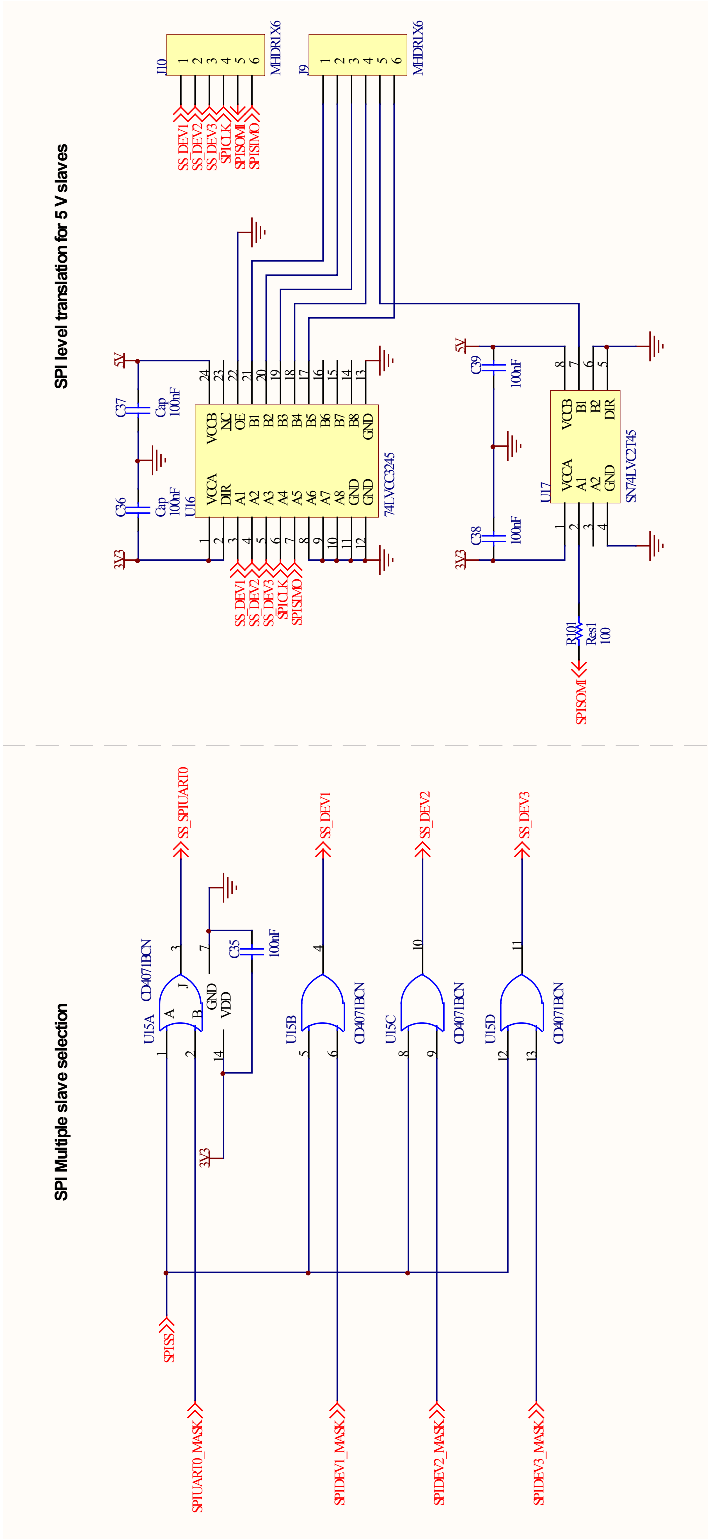


Figure A1.6. Hardware schematic for SPI support components.

Barometer

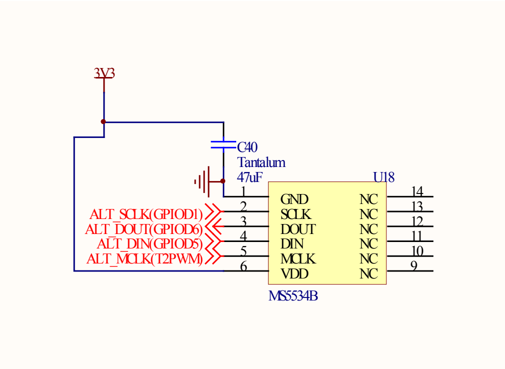


Figure A1.7. Barometer connections.

Connections to the ezDSP development board

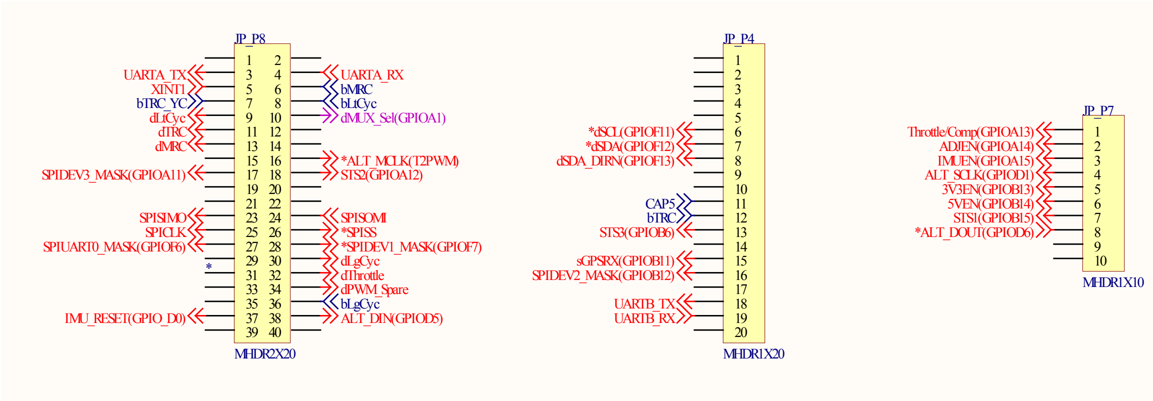


Figure A1.8. Connections between hardware devices and the ezDSP development board.

Appendix 2. Minimum variance estimator

Assume that there exists three measurement sources for some quantity, x. Each measurement produces one noise corrupted reading for that quantity, with associated noise covariances (provided all noises are zero mean).

$$\begin{aligned} y_1 &= x + n_1 \\ y_2 &= x + n_2 \\ y_3 &= x + n_3 \end{aligned} \qquad \text{A2.1}$$

$$\begin{aligned} \sigma_1{}^2 &= \mathrm{cov}(n_1) \\ \sigma_2{}^2 &= \mathrm{cov}(n_2) \\ \sigma_3{}^2 &= \mathrm{cov}(n_3) \end{aligned} \qquad \text{A2.2}$$

It is required that a linear estimator (Equation A2.3) be found, with coefficients a_1, a_2 and a_3, such that the variance of the estimate is minimised subject to the constraint $a_1 + a_2 + a_3 = 1$.

$$\hat{x} = a_1 y_1 + a_2 y_2 + a_3 y_3 \qquad \text{A2.3}$$

Substituting Equation A2.1 into Equation A2.3,

$$\begin{aligned} \hat{x} &= a_1(x + n_1) + a_2(x + n_2) + a_3(x + n_3) \\ &= \underbrace{x(a_1 + a_2 + a_3)}_{A} + \underbrace{a_1 n_1 + a_2 n_2 + a_3 n_3}_{B} \end{aligned} \qquad \text{A2.4}$$

The variance of the estimate, in Equation A2.4 must be minimised and the constraint mentioned above follows naturally from A.

Thus, $\hat{\sigma}_x^2$, below, must be minimised.

$$\begin{aligned} \hat{\sigma}_x^2 &= E\left\{(\hat{x} - x)^2\right\} \\ &= E\left\{\left[x(\underbrace{a_1 + a_2 + a_3}_{=1\ from\ constra\mathrm{int}}) - x + a_1 n_1 + a_2 n_2 + a_3 n_3 \right]^2\right\} \\ &= E\left\{(a_1 n_1 + a_2 n_2 + a_3 n_3)^2\right\} \\ &= a_1{}^2\sigma_1{}^2 + a_2{}^2\sigma_2{}^2 + a_3{}^2\sigma_3{}^2 \\ &(for\ independent\ noises) \end{aligned} \qquad \text{A2.5}$$

The minimisation problem can now be seen as minimising:

$$\begin{gathered} \hat{\sigma}_x^2 = a_1{}^2\sigma_1{}^2 + a_2{}^2\sigma_2{}^2 + a_3{}^2\sigma_3{}^2 \\ \text{Subject to: } a_1 + a_2 + a_3 = 1 \end{gathered} \qquad \text{A2.6}$$

The Lagrange Multiplier will be applied to perform the minimisation (Finney et al., 2001). For notational brevity,

$$\begin{aligned} b &= \hat{\sigma}_x^2 \\ c &= {\sigma_1}^2 \\ d &= {\sigma_2}^2 \\ e &= {\sigma_3}^2 \end{aligned} \qquad \text{A2.7}$$

Thus minimise,

$$\begin{gathered} b = {a_1}^2 c + {a_2}^2 d + {a_3}^2 e \\ Subject \quad to \quad a_1 + a_2 + a_3 = 1 \quad \Rightarrow \quad Let\ g = a_1 + a_2 + a_3 - 1 \end{gathered} \qquad \text{A2.8}$$

Using the method of Lagrange Multipliers,

$$\begin{gathered} \nabla b = \lambda \nabla g \\ 2a_1 c\, i + 2a_2 d\ j + 2a_3 e\, k = \lambda i + \lambda j + \lambda k \end{gathered} \qquad \text{A2.9}$$

Where:

1. ∇ indicates the gradient of the subsequent function.
2. λ is the Lagrange multiplier relating the gradient of the function to be minimised to the gradient of the constraint.
3. i, j and k are the unit vectors in the a_1, a_2 and a_3 directions respectively (allowing abstraction of the minimisation problem to the vector space).

Solving simultaneously for a_1, a_2 and a_3 (using $a_1 + a_2 + a_3 = 1$ also), gives the coefficients required to solve the problem.

$$\begin{aligned} a_1 &= \frac{ed}{ed + ce + cd} \\ a_2 &= \frac{ce}{ed + ce + cd} \\ a_3 &= \frac{cd}{ed + ce + cd} \end{aligned} \qquad \text{A2.10}$$

Using these results in Equations A2.3 and A2.5 produces the minimum variance estimate and variance respectively (Equations A2.11)

$$\hat{x} = \left(\frac{\sigma_3{}^2\sigma_2{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right) y_1 + \left(\frac{\sigma_1{}^2\sigma_3{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right) y_2 + \left(\frac{\sigma_1{}^2\sigma_2{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right) y_3$$

A2.11

$$\hat{\sigma}_x^2 = \left(\frac{\sigma_3{}^2\sigma_2{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right)^2 \sigma_1{}^2 + \left(\frac{\sigma_1{}^2\sigma_3{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right)^2 \sigma_2{}^2 + \left(\frac{\sigma_1{}^2\sigma_2{}^2}{\sigma_3{}^2\sigma_2{}^2 + \sigma_1{}^2\sigma_3{}^2 + \sigma_1{}^2\sigma_2{}^2} \right)^2 \sigma_3{}^2$$

For the case where there are two measurements that must be combined, it is required that a linear estimator (Equation A2.12) be found, with coefficients a_1 and a_2, such that the variance of the estimate is minimised subject to the constraint $a_1 + a_2 = 1$.

$$\hat{x} = a_1 y_1 + a_2 y_2 \qquad \text{A2.12}$$

Substituting the relevant parts of Equation A2.1 into Equation A2.12,

$$\begin{aligned} \hat{x} &= a_1(x + n_1) + a_2(x + n_2) \\ &= \underbrace{x(a_1 + a_2)}_{A} + \underbrace{a_1 n_1 + a_2 n_2}_{B} \end{aligned} \qquad \text{A2.13}$$

The variance of the estimate, in Equation A2.13 must be minimised and the constraint mentioned above follows naturally from A.

Thus, $\hat{\sigma}_x^2$,below, must be minimised.

$$\begin{aligned} \hat{\sigma}_x^2 &= E\left\{ (\hat{x} - x)^2 \right\} \\ &= E\left\{ \left[x(\underbrace{a_1 + a_2}_{=1 \; from \; constra \, \text{int}}) - x + a_1 n_1 + a_2 n_2 \right]^2 \right\} \\ &= E\left\{ (a_1 n_1 + a_2 n_2)^2 \right\} \\ &= a_1{}^2\sigma_1{}^2 + a_2{}^2\sigma_2{}^2 \\ &\quad (for\ independent\ noises) \end{aligned} \qquad \text{A2.14}$$

The minimisation problem can now be seen as minimising:

$$\begin{aligned} \hat{\sigma}_x^2 &= a_1{}^2\sigma_1{}^2 + a_2{}^2\sigma_2{}^2 \\ \text{Subject to: } & a_1 + a_2 = 1 \end{aligned} \qquad \text{A2.15}$$

The Lagrange Multiplier will again be applied to perform the minimisation (Finney et al., 2001). For notational brevity,

$$f = \hat{\sigma}_x \qquad \text{A2.16}$$

Thus minimise,

$$\begin{gathered} f = a_1{}^2 c + a_2{}^2 d \\ Subject \quad to \quad a_1 + a_2 = 1 \quad \Rightarrow \quad Let\ g = a_1 + a_2 - 1 \end{gathered} \qquad \text{A2.17}$$

Using the method of Lagrange Multipliers,

$$\begin{gathered} \nabla f = \lambda \nabla g \\ 2a_1 c\ i + 2a_2 d\ j = \lambda i + \lambda j \end{gathered} \qquad \text{A2.18}$$

Solving simultaneously for a_1 and a_2 (using $a_1 + a_2 = 1$ also), gives the coefficients required to solve the problem.

$$\begin{gathered} a_1 = \frac{d}{c+d} \\ a_2 = \frac{c}{c+d} \end{gathered} \qquad \text{A2.19}$$

Using these results in Equations A2.12 and A2.14 produces the minimum variance estimate and variance respectively (Equations A2.20)

$$\begin{gathered} \hat{x} = \left(\frac{\sigma_2{}^2}{\sigma_1{}^2 + \sigma_2{}^2} \right) y_1 + \left(\frac{\sigma_1{}^2}{\sigma_1{}^2 + \sigma_2{}^2} \right) y_2 \\ \hat{\sigma}_x^2 = \left(\frac{\sigma_2{}^2}{\sigma_1{}^2 + \sigma_2{}^2} \right)^2 \sigma_1{}^2 + \left(\frac{\sigma_1{}^2}{\sigma_1{}^2 + \sigma_2{}^2} \right)^2 \sigma_2{}^2 \end{gathered} \qquad \text{A2.20}$$

Appendix 3. Software, datasheets and other digital content

The DVD-ROM below contains several items referred to in this document. Microsoft Windows operating systems should automatically open a HTML document upon reading the disc. If this does not occur, execute the following command from a command line or the Windows "Run" dialogue box.

```
<DVD drive letter>:\index.htm
```

The DVD contains:

1. Hardware device data sheets for avionics components.
2. Avionics hardware design files (schematics, printed circuit boards and libraries) designed with Protel DXP as well as a hardware design document detailing a custom IMU design.
3. Avionics software design files including all C code and ground station Visual Basic project files.
4. Photographs and videos of the helicopter and test flights.
5. Matlab files required for the Matlab and embedded filter implementations.
6. Several software installation packages for software used in the project.