The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Nottingham Geospatial Institute

Advanced Navigation Architecture for Low-cost Unmanned Aerial Vehicles

Hery A Mwenegoha

Thesis submitted to the University of Nottingham for the degree of

Doctor of Philosophy

November 2021

To my parents

'If I have seen further, it is by standing on the shoulders of giants.'

Isaac Newton, 1643-1727

Abstract

This thesis details the effective integration of global navigation satellite systems (GNSS) with an inertial navigation system (INS) to meet the requirements for use in small, mass-market unmanned aerial vehicles (UAVs). A key focus is on the addition of the input from the vehicle dynamic model (VDM).

The dominant navigation system for most small, mass-market UAVs is based on INS/GNSS integration. The integration of the two systems provides a navigation solution with both short-term and long-term accuracy. However, during a GNSS outage, the navigation solution drifts. This can happen due to severe multipath, intentional or unintentional interference, even against cryptographically secured GNSS signals, rapid dynamics and loss of line of sight to the satellites. Most small UAVs use low-cost inertial sensors, which during a GNSS outage, will cause the navigation solution to drift rapidly. Traditionally, additional aiding sensors such as cameras and range finders have been used to reduce the rapid drift of the navigation solution. However, this approach adds extra weight and additional cost to the overall system. More recently, the use of a VDM in providing improved navigation performance has gained research popularity, especially for small, mass-market UAVs. This approach preserves the autonomy of the navigation system while avoiding extra cost, additional weight, and power requirements, essential for low-cost applications.

This thesis presents a VDM navigation architecture suitable for a fixed-wing UAV fitted with low-cost inertial sensors and a GNSS receiver during periods of extended GNSS outage. The thesis presents and examines state-of-the-art VDM navigation techniques, quantifies their limitations and identifies approaches to reduce navigation solution drift during GNSS outages. An integration algorithm that implements the approaches and overcomes these limitations is developed and evaluated via a Monte Carlo simulation study. The integration algorithm is then tested on real flight data gathered from a test flight using a small fixed-wing UAV.

The thesis identifies that most current VDM integration schemes use a loosely coupled configuration, using position and velocity measurements from a GNSS receiver. This work shows that the VDM navigation solution can drift significantly with this configuration during an extended GNSS outage. A novel VDM-based architecture is then developed to reduce the navigation solution drift during extended GNSS outages. The architecture, referred to as a tightly coupled VDM-based integration architecture (or simply TCVDM), uses raw GNSS observables and measurements from inertial sensors to aid the navigation solution even when tracking less than four satellites. The architecture uses an extended Kalman filter (EKF) to estimate the navigation solution errors. A software-based GNSS measurement simulator is also developed to generate the raw GNSS observables.

Simulation results reveal significant improvements in navigation accuracy during GNSS outages. In the case of a GNSS outage lasting over two minutes,

results show that position accuracy improves by one to two orders of magnitude compared to a tightly coupled INS/GNSS integration scheme (TCINS) and by a factor of four compared to the state-of-the-art VDM integration architecture. In addition to the navigation states, the filter also estimates wind velocity components, VDM parameters and the receiver clock offset and drift. The estimation of wind velocity components is achieved even without an air data system. It is found that the architecture only resolves 40% of the initial error in the model parameters. This is found to be sufficient for navigation with randomly distributed errors of 10% in the model parameters.

The developed architecture is also tested on real flight data gathered using a small fixed-wing UAV. A custom flight control system (FCS) that houses a low-cost inertial measurement unit (IMU), barometer and a data logging module is used on the UAV. The FCS is used for guidance, navigation and logging control inputs and different measurements. Three GNSS receivers are installed on the UAV and used to derive a reference position, velocity and attitude solution. Test results show that the position error estimation performance for the TCVDM scheme improves by a factor of 43 compared to a TCINS scheme with two satellites visible during a GNSS outage. The velocity error estimation performance for the TCVDM scheme improves by a factor of 7 across all channels compared to a TCINS scheme during the outage. However, the TCVDM scheme shows poor attitude estimation performance. This is attributed to the lack of accurate VDM parameters, especially the torque coefficients, which leads to significantly worse yaw angle estimation performance.

This work presents an alternative, low-cost navigation scheme for small UAVs that uses sensors usually available in most UAVs. The navigation scheme can work alongside existing integration architectures to provide a secondary navigation solution for improved reliability and integrity monitoring.

Acknowledgements

The research undertaken for this thesis was conducted at the Institute for Aerospace Technology and the Nottingham Geospatial Institute (NGI). I would like to thank my academic supervisors Dr James Pinchin, Dr Mark Jabbal, and Professor Terry Moore (Professor Emeritus), for their support and guidance throughout this research.

The work is funded by the INNOVATIVE doctoral programme. The INNOVATIVE programme is partially funded by the Marie Curie Initial Training Networks (ITN) action (project number 665468) and partially by the Institute for Aerospace Technology (IAT) at the University of Nottingham.

I would like to thank Francesco Basile for the discussion we had at the beginning of my research on software-based GNSS measurement simulators. This discussion helped and motivated the development and testing of a GNSS measurement simulator used in this research.

For their contribution to the fieldwork undertaken for this thesis, especially on the use of GNSS equipment, I would like to thank Sean Ince and Norma Oldfield. I would also like to thank Andrew Hallam and Thomas Haskins for their help with setting up the full-scale oscillation rig and support in preparing the small UAV for flight tests. I would also like to thank Julien Martin for carrying out the wind tunnel tests and full-scale oscillation tests.

Outside of the University of Nottingham, I would like to thank the Hucknall Model Aircraft Flying Club administration for their guidance and support in conducting flight tests. I would like to thank Alan Falconbridge for his flexibility and willingness to pilot the UAV whilst offering valuable feedback.

I would also like to thank colleagues and friends in and outside the Department. I would like to thank my colleagues within the Aerospace Technology Centre (ATC) and the NGI for the great support and memories built over the years.

And finally, and most importantly, I would like to thank my lovely parents for their unlimited and unconditional love, encouragement, sacrifice, and support in my years of study. It is without a doubt that without their support, this work would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Publications

During this Research

Mwenegoha, H., Moore, T., Pinchin, J. and Jabbal, M. (2019) 'Model-Based Autonomous Navigation with Moment of Inertia Estimation for Unmanned Aerial Vehicles', *Sensors*, 19(11).

Mwenegoha, H. A., Moore, T., Pinchin, J. and Jabbal, M. (2019) 'Enhanced Fixed Wing UAV Navigation in Extended GNSS Outages using a Vehicle Dynamics Model and Raw GNSS Observables', in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. Miami, Florida, pp. 2552–2565.

Mwenegoha, H. A., Moore, T., Pinchin, J. and Jabbal, M. (2020) 'A Model-based Tightly Coupled Architecture for Low-Cost Unmanned Aerial Vehicles for Real-Time Applications', *IEEE Access*.

Mwenegoha, H. A., Moore, T., Pinchin, J. and Jabbal, M. (2021) 'Error characteristics of a model-based integration approach for fixed-wing unmanned aerial vehicles', *Journal of Navigation,* pp. 1-14.

Before this Research

Mwenegoha, H. A. and Jabbal, M. (2013) 'Investigation of Passive Flow Control Techniques to Enhance the Stall Characteristics of a Microlight Aircraft', *International Journal of Flow Control*, 5(3), pp. 215–242.

List of Abbreviations

ANO	Air Navigation Order
BPSK	Binary Phase Shift Keying
BLDC	Brushless Direct Current
BVLOS	Beyond Visual Line of Sight
CAA	Civil Aviation Authority
C/A	Coarse/Acquisition
CDMA	Code Division Multiple Access
CEP	Circular Error Probable
DDCP	Double Differenced Carrier Phase
DLL	Delay Lock Loop
DOF	Degree-of-Freedom
DOY	Day of the Year
DR	Dead Reckoning
ECEF	Earth-Centred Earth-Fixed
ECI	Earth-Centred Inertial
EKF	Extended Kalman Filter
ESA	European Space Agency
ESC	Electronic Speed Controller
FCS	Flight Control System
FOG	Fibre Optic Gyro
GLONASS	Global'naya Navigatsionaya Sputnikovaya Sistema
GNC	Guidance, Navigation, and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HALE	High-Altitude Long-Endurance
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISP	In-System Programming
LAMBDA	Least-Squares Ambiguity Decorrelation Adjustment
LCVDM	Loosely Coupled Vehicle Dynamic Model
MEMS	Micro-Electro-Mechanical System
MIPS	Million Instructions Per Second
MTOW	Maximum Take-off Weight

NAVSAT	Navy Navigation Satellite System
NAVSTAR	Navigation System with Timing And Ranging
NED	North East Down
NLOS	Non-Line-of-Sight
OCXO	Oven-Controlled Crystal Oscillator
PCO	Phase Centre Offset
PCV	Phase Centre Variation
PDOP	Position Dilution of Precision
PLL	Phase Lock Loop
ppm	Parts per Million
PSD	Power Spectral Density
PVAT	Position, Velocity, Attitude and Time
PVT	Position, Velocity and Time
RF	Radio Frequency
RLG	Ring Laser Gyro
RMS	Root Mean Square
RPM	Revolutions per Minute
SEP	Spherical Error Probable
TCINS	Tightly Coupled Inertial Navigation System
TCVDM	Tightly Coupled Vehicle Dynamic Model
TCXO	Temperature-Compensated Crystal Oscillator
TDCP	Time Differenced Carrier Phase
TECS	Total Energy Control System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VDM	Vehicle Dynamic Model
VTOL	Vertical Take-off and Landing

List of Symbols

α	angle of attack
α_k	GPS ionospheric delay correction parameter set
β	sideslip angle
β_k	GPS ionospheric delay correction parameter set
δ_α	aileron deflection
δ_e	elevator deflection
δ_r	rudder deflection
η	track capture angle
η_{esc}	ESC efficiency
λ	longitude
μ	latitude
μ_E	Earth's gravitational constant
ρ	air density
ρ_r^s	geometric range between (r) and (s)
σ	standard deviation
σ_{iono}	standard deviation of the residual ionospheric error
σ_{sclock}	standard deviation of the GPS broadcast clock
σ_{tropo}	standard deviation of the residual tropospheric error
τ	torque
τ	time constant
τ_n	motor-propeller time constant
ϕ	roll angle
ϕ^b	rotation vector
ϕ_r^s	carrier phase between (r) and (s)
Φ_k	State transition matrix at epoch k
θ	pitch angle
ψ	yaw angle
ω_{en}^n	transport-rate
ω_{ib}^b	rotation rate vector in the body frame
ω_{ie}	Earth rotation rate
ω_x	roll rate
$\overline{\omega}_x$	dimensionless roll rate
ω_y	pitch rate
$\overline{\omega}_y$	dimensionless pitch rate

ω_z	yaw rate
$\bar{\omega}_z$	dimensionless yaw rate
a	temperature lapse rate
b	wingspan
b_1	tail-baseline vector
b_2	left-wing baseline vector
b_{a_1}	accelerometer bias in the x-axis
b_{a_2}	accelerometer bias in the y-axis
b_{a_3}	accelerometer bias in the z-axis
b_{clk}	receiver clock offset
b_{g_1}	gyroscope bias in the x-axis
b_{g_2}	gyroscope bias in the y-axis
b_{g_3}	gyroscope bias in the z-axis
c	speed of light in free space
$\bar{c}$	mean aerodynamic chord
CD	total drag force coefficient (experiment and AVL)
CF_{T_1}	static thrust force coefficient
CF_{T_2}	dynamic thrust force coefficient
CF_{T_3}	dynamic thrust force coefficient
CF_X	total drag force coefficient
CF_{X_1}	drag force coefficient
CF_{X_α}	drag force derivative with angle of attack
$CF_{X_{\alpha 2}}$	drag force derivative with angle of attack
$CF_{X_{\beta 2}}$	drag force derivative with sideslip angle
CF_{Y1}	lateral force coefficient
CF_{Z_1}	lift force coefficient at zero angle of attack
CF_{Z_α}	lift curve slope
CL	lift force coefficient
Cl	roll moment coefficient (experiment and AVL)
Cm	pitch moment coefficient (experiment and AVL)
CM_p	propeller torque coefficient
CM_{X_β}	roll moment derivative with sideslip angle
$CM_{X_{\delta_\alpha}}$	roll moment derivative with aileron deflection
$CM_{X_{\bar{\omega}_x}}$	roll moment derivative with dimensionless roll rate
$CM_{X_{\bar{\omega}_z}}$	roll moment derivative with dimensionless yaw rate

CM_{Y1}	pitch moment coefficient at the aerodynamic centre
CM_{Y_α}	pitch moment derivative with angle of attack
$CM_{Y_{\delta_e}}$	pitch moment derivative with elevator deflection
$CM_{Y_{\bar{\omega}_y}}$	pitch moment derivative with dimensionless pitch rate
$CM_{Z_{\bar{\omega}_z}}$	yaw moment derivative with dimensionless yaw rate
CM_{Z_β}	yaw moment derivative with sideslip angle
$CM_{Z_{\delta_r}}$	yaw moment derivative with rudder deflection
Cn	yaw moment coefficient (experiment and AVL)
CY	lateral force coefficient (experiment and AVL)
d_{clk}	receiver clock drift
D	propeller diameter
D_r^s	Doppler frequency between (r) and (s)
e_r^s	line of sight vector from (r) to (s)
E^s	elevation of satellite (s)
F	linearised dynamic matrix
f_i	carrier frequency in the L(i) band
f_{ib}^b	specific force in the body frame
F_T	thrust force
F_X^w	drag force
F_Y^w	lateral force
F_Z^w	lift force
g	gravity acceleration
G_{clk}	noise shaping matrix for the GNSS receiver clock errors
G_e	noise shaping matrix for the IMU errors
G_n	noise shaping matrix for the navigation states
G_p	noise shaping matrix for the VDM parameters
G_w	noise shaping matrix for the wind velocity components
h	geodetic height
h_o	basepoint altitude in the standard atmosphere
H	linearised observation matrix
I^b	aircraft inertia matrix in the body-fixed frame
I_{xx}	moment of inertia component about the x-axis
I_{yy}	moment of inertia component about the y-axis
I_{zz}	moment of inertia component about the z-axis
I_{xz}	product of inertia
I_r^s	ionospheric delay between (r) and (s)

J	advance ratio
K_k	Kalman gain
m	aircraft mass
M	vector of aircraft moments
M_p	propeller torque
M_P	error due to multipath
M_X^b	roll moment
M_Y^b	pitch moment
M_Z^b	yaw moment
n	propeller speed
n_c	commanded propeller speed
N_r^s	carrier integer ambiguity between (r) and (s)
p	roll rate
P_k	Error covariance matrix
P_0	basepoint in the standard atmosphere
P_0	initial covariance matrix
P_r^s	pseudorange between (r) and (s)
q	pitch rate
$\bar{q}$	dynamic pressure
q_b^n	quaternion rotation vector from the body frame to local navigation frame
q_0	quaternion scalar component
q_1	quaternion vector component
q_2	quaternion vector component
q_3	quaternion vector component
Q_{clk}	process noise covariance matrix for the GNSS receiver clock
Q_k	process noise covariance matrix at epoch k
r	yaw rate
r_{br}^n	baseline vector between (b) and (r)
r_{er}^e	receiver position vector in ECEF frame
r_{eb}^n	aircraft position vector in the local frame
r_{es}^e	satellite position vector in ECEF frame
R	rotation matrix
R_a	gas constant for air
R_b^n	transformation matrix from (b) to (n)
R_k	measurement covariance
R_M	meridian radius of curvature

R_p	prime vertical radius of curvature
R_σ	code to carrier-phase error ratio
S	aircraft wing area
t_r	receiver time of signal reception
Th	GNSS receiver thermal noise
T_0	Basepoint temperature in the standard atmosphere
T_r^s	tropospheric delay between (r) and (s)
T_s	satellite time of signal transmission
u_f	GNSS receiver clock model additive white noise
u_g	GNSS receiver clock model driving white noise
U	Control input vector
v_{er}^e	receiver velocity vector in ECEF frame
v_{eb}^n	aircraft velocity vector in the local frame
v_{es}^e	satellite velocity vector in ECEF frame
V	airspeed
V^b	airspeed vector
V_x^b	airspeed component along X_b
V_y^b	airspeed component along Y_b
V_z^b	airspeed component along Z_b
w_g	GNSS noise vector
w_i	IMU noise vector
w_k	measurement noise vector
w_N	wind velocity component along X_N
w_E	wind velocity component along Y_E
w_D	wind velocity component along Z_D
W^n	wind velocity vector
x	state vector
x_f	GNSS receiver clock frequency
x_p	GNSS receiver clock phase
X_{clk}	receiver clock states
X_e	IMU error states
X_n	navigation states
X_p	VDM parameters
X_w	wind velocity states
Z_k	measurement vector
ZTD_d	zenith tropospheric dry delay
ZTD_w	zenith tropospheric wet delay

1 Introduction

1.1. Background

Navigation involves the determination of the position and velocity of a moving object with respect to a known reference and guiding it to a specific destination. Navigation techniques usually fall into two main categories, namely position fixing and dead reckoning.

Position fixing refers to the different techniques used to determine the position and velocity of an object using measurements with respect to known reference points. Usually, the measurements are via radio frequency (RF) transmission, and one such system is the global navigation satellite system (GNSS). As the name dictates, position fixing alone does not offer the spatial orientation of an object (Tawk, 2013).

Dead reckoning (DR) is a relative positioning technique in which the current position is determined from the previous position and measurements of the direction of motion and distance travelled. A DR system's performance depends on the accuracy of the initial states and the accuracy with which velocity and orientation can be determined. An inertial navigation system (INS) is an example of a DR system. Error accumulation with time is the limiting factor for most DR systems (Tawk, 2013).

An inertial navigation system consists of a system of inertial sensors, also called an inertial measurement unit (IMU), which measure specific force and angular velocity with respect to an inertial frame. An INS also includes a computing element that calculates a moving object's position, velocity, and orientation. Inertial sensors consist of accelerometers and gyroscopes (gyros). An INS is self-contained. It does not depend on exteroceptive sensing in the computation of the navigation states upon initialisation, thus making it immune to jamming, spoofing and interference. Figure 1.1 shows the main blocks of a simple INS.

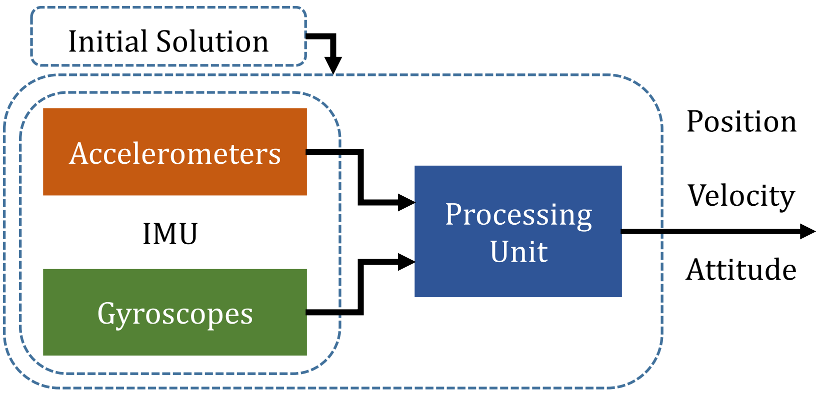


Figure 1.1. A simple INS setup.

An INS integrates measurements from inertial sensors to estimate a moving object's position, velocity, and orientation through its computing element. Therefore, any initial errors or measurement errors build up to significant navigational errors. The rate of divergence of the navigation solution depends on the quality of the inertial sensors used. Different types of errors corrupt inertial sensors' measurements, such as random noise, scale-factor, bias instability, bias variation with temperature and cross-coupling errors. To prevent error growth, an INS is usually integrated with other sensors and systems such as GNSS, magnetometers, and range finders.

Unmanned aerial vehicles (UAVs) have found wide use in so-called 'D-D-D' (Dull-Dangerous-Dirty) fields (Jiménez López and Mulero-Pázmány, 2019). Aerial photography, mapping, search and rescue, surveillance and reconnaissance, resource management, border patrol and inspections, anti-poaching campaigns are amongst a few areas where UAVs are being used. UAVs are dominated by two main types, fixed-wing conventional aircraft and rotary-wing vertical take-off and landing (VTOL) aircraft, as shown in Figure 1.2 (a, b) (Saeed *et al.*, 2015).

(a) A fixed-wing UAV

(b) A rotary-wing VTOL UAV(quadrotor)

(c) A hybrid UAV (tilt-rotor)

Figure 1.2. Different types of UAVs (Yu *et al.*, 2016).

Each type has its limitations on endurance, payload capacity, range, controllability and manoeuvrability. For instance, fixed-wing UAVs have significantly better endurance and payload capacity as opposed to the rotary-wing type. Rotary-wing VTOL UAVs can easily take off and land without requiring a dedicated runway. The inherent limitations of the two types have led to the development of fixed-wing VTOL UAVs or hybrid UAVs (shown in Figure 1.2 (c)) that inherit both types' advantages (Saeed *et al.*, 2015). In the United Kingdom, the basic rules for operating UAVs are governed by the Air Navigation Order (ANO) 2016. At the time of this writing, this has been amended by Air Navigation (Amendment) Orders 2017/1112, 2018/623 and 2018/1160. The ANO defines a small UAV as any unmanned aircraft other than a balloon or a kite with a mass not exceeding 25 kg (CAA, 2016).

A navigation system is an integral part of a UAV estimating its position, velocity, and attitude used in guidance and control of the aircraft, as shown in Figure 1.3.

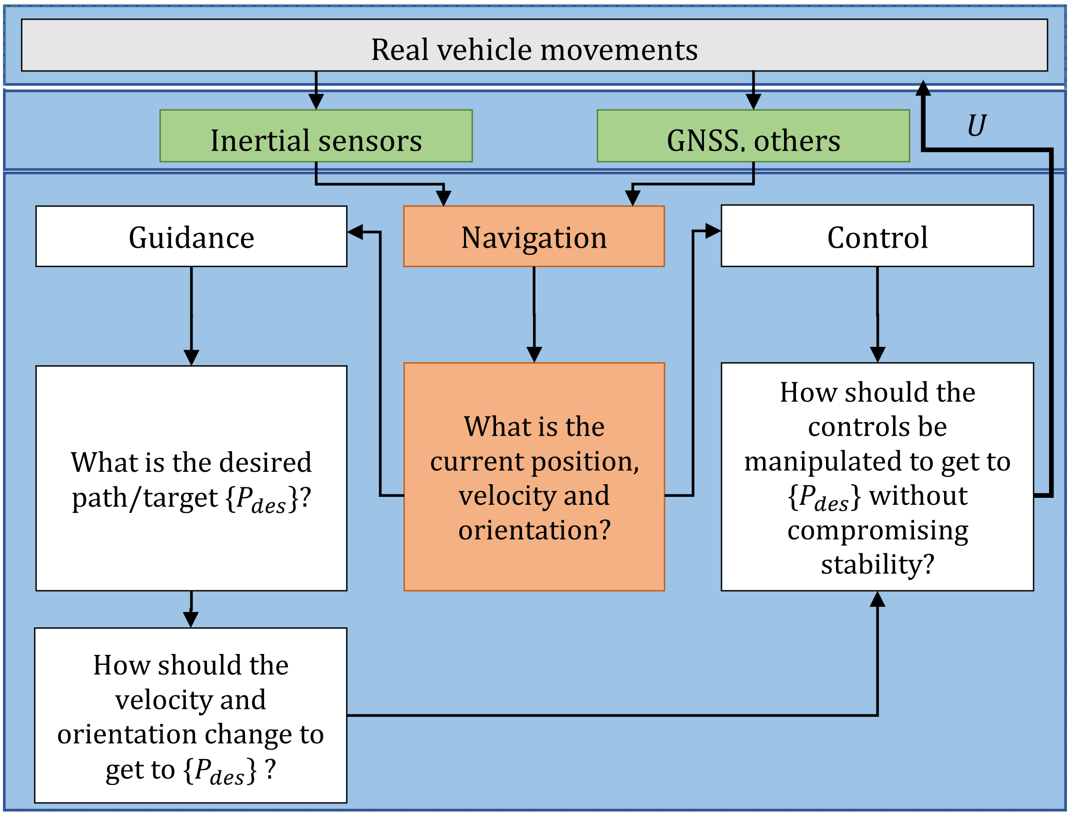


Figure 1.3. Guidance, Navigation and Control architecture. In the figure, $\{P_{des}\}$ represents the set of desired states.

Improved navigation reliability has been achieved through hardware redundancy which increases cost, power consumption, and weight. Technological advancements from ring laser gyros (RLG), fibre optic gyros (FOG) to micro-electro-mechanical systems (MEMS) inertial sensors alongside developments in GNSS have paved the way to a wealth of new commercial applications (Tawk, 2013). MEMS-based inertial sensors have enabled a dramatic reduction in INS size, weight, and power consumption, allowing its use in new applications and instruments such as wildlife tracking, medical instruments, and smartphones (Nusbaum and Klein, 2017). Moreover, the significant reduction in size, weight, and power consumption of MEMS-based inertial sensors has enabled the use of INS for guidance, navigation and control (GNC) in small-scale to large-scale UAVs. The quality of the inertial sensors used determines the grade of the INS. Table 1.1 summarises three broad categories based on error characteristics and cost. MEMS inertial sensors usually include a triad of sensors on a single silicon wafer. The significant reduction in size of the sensing elements has imposed performance limits for MEMS-based sensors.

Table 1.1. Different grades of IMU (Hide, 2003; Honeywell, 2018; Xsens, 2018).

Grade	Navigation	Tactical	Automotive
Example	Honeywell HG9900	Northrop LN200	Xsens MTi 100
Dimensions (cm)	13.91x16.26x13.56	8.9x8.9x8.5	5.7x4.2x2.35
Cost (£ approx.)	>100k	20k	1.5k
Gyro	Ring laser	Fibre Optic	MEMS
Bias ($°/h$)	0.0035	1-10	<720
Scale Factor (ppm)	5.0	100	<1%
Noise ($°/h/\sqrt{\text{Hz}}$)	0.002	0.04-0.1	<36
Accelerometer	Silicon	Silicon	Silicon
Bias (mg)	0.025	$0.2 - 1$	$< \pm 5$
Scale Factor (ppm)	100	300	$< \pm 1\%$
Noise ($m/s/\sqrt{hr}$)	$7\ \mu g$	0.03	< 0.15

1.2. Research Question and Motivation

The research question being addressed is:

"To what extent can knowledge of the vehicle dynamic model and associated control inputs be used with low-cost MEMS-grade inertial sensors and mass-market GNSS receivers to reduce drift in the navigation solution during a GNSS outage ?"

The GNSS market report by the European GNSS Agency (2019) indicated that the number of GNSS units shipped on UAVs (Drones) of different categories exceeded 10 million units in 2018. Further, the report highlighted that UAVs have become a third market segment for GNSS shipments and account for a large proportion of installed units after the consumer solutions and road applications, as shown in Figure *1.4*.

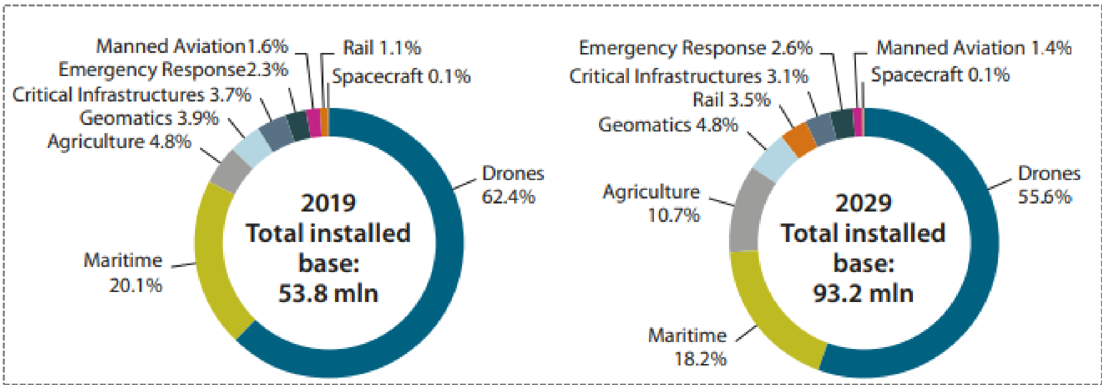


Figure 1.4. Installed GNSS base for categories other than consumer solutions and road applications (European GNSS Agency, 2019).

The report also highlighted different efforts in developing UAVs for beyond visual line of sight (BVLOS) operations and package delivery in urban environments, which require accurate position information for mission success. It was indicated that the reliance on GNSS for position information is expected to increase. In Europe alone, the UAV service revenue in different service areas such as surveying, rail inspections, agriculture, delivery, and e-commerce is projected to increase from 50 million euros in 2019 to over 700 million euros in 2029. The report also highlighted that even though the demand for GNSS positioning is growing, GNSS alone can not meet the accuracy requirements for all settings/environments.

Most low-cost, mass-market UAVs use an INS integrated with GNSS to provide a navigation solution with both short-term and long-term accuracy (Kim and Sukkarieh, 2003; George and Sukkarieh, 2005; Babu and Wang, 2009; Brown and Hwang, 2012; Falco, Pini and Marucco, 2017). A GNSS receiver needs to track at least four satellites to output the absolute position information that can be used to correct the accumulated error in the inertial navigation solution. However, problems tend to arise during a GNSS outage where the integrated navigation solution drifts (Hide, 2003; Wang *et al.*, 2004; Lau, Liu and Lin, 2013; Quinchia *et al.*, 2013). This can happen due to intentional or unintentional corruption (even against cryptographically secured GNSS signals), rapid dynamics, and severe multipath (Groves, 2008; Papadimitratos and Jovanovic, 2008; Tawk *et al.*, 2014). The rate of navigation solution drift depends on the quality of the inertial sensors used. In most small UAVs, the quality of the inertial sensors is relatively low. As a result, the position uncertainty is far from being of practical use after only a few seconds of GNSS outage (Khaghani and Skaloud, 2016b).

Some authors used additional aiding sensors, such as cameras and range finders, to reduce rapid drift in the navigation solution during GNSS outages (Kim and Sukkarieh, 2003; Wang *et al.*, 2004; Madison *et al.*, 2007; Vasconcelos *et al.*, 2010). Besides adding extra weight and additional costs, these sensors suffer from inherent limitations due to dependency on external sensing. Other authors have explored advanced integration schemes, while others have investigated advanced error modelling schemes, saving on weight but introducing additional software complexities (George and Sukkarieh, 2005; El-Diasty and Pagiatakis, 2009; Quinchia *et al.*, 2013; Tawk *et al.*, 2014).

More recently, research has been conducted on the use of vehicle dynamic models (VDM) to reduce the drift of the navigation solution during periods of extended GNSS outage (Koifman and Bar-Itzhack, 1999; Bryson and Sukkarieh, 2004; Vasconcelos *et al.*, 2010; Crocoll *et al.*, 2013; Khaghani and Skaloud, 2016a). This approach preserves the system's autonomy and avoids extra cost and weight on the host platform. Figure 1.5 shows the typical VDM integration schemes. In contrast to a strapdown algorithm (SDA) which relies on inertial sensors to propagate the navigation solution, a VDM relies on control inputs.

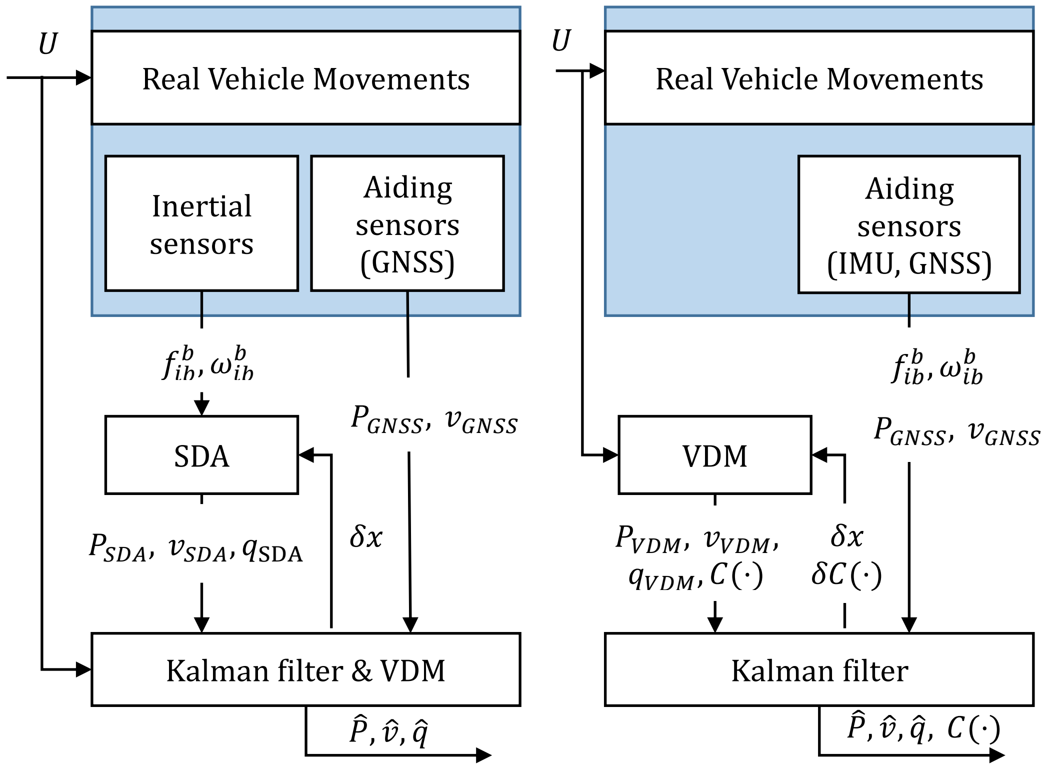


Figure 1.5. Embedded VDM navigation (left), external VDM navigation (right). In the figure $[P, v, q]_{[SDA,VDM]}$ represent the position, velocity and orientation (in quaternion) states by the SDA and VDM, respectively; $C(\cdot)$ represents the VDM parameters; δx represents the corrections to the navigation states.

Even though different in structure and implementation, the current VDM integration schemes use GNSS position and velocity measurements to update the navigation solution. As explained previously, these measurements will not be available during a GNSS outage. When coupled with modelling assumptions and errors in the VDM parameters will cause a VDM-based navigation solution to drift, in some cases more rapidly than others (Bryson and Sukkarieh, 2004). However, even when tracking less than four satellites, a GNSS receiver can still output useful information, such as the pseudorange and Doppler frequency measurements. These can be used to provide a quasi-continuous integrated VDM navigation solution with reduced error growth.

1.3 Research Aim and Objectives

The aim of this research is to investigate and test a VDM navigation architecture suitable for a fixed-wing UAV fitted with low-cost MEMS-grade inertial sensors and a GNSS receiver during periods of extended GNSS outage. The term ‘low-cost’ is used to represent MEMS-grade sensors that cost typically less than £5,000 for an IMU assembly.

A fixed-wing UAV is considered in this research due to its advantages over rotary-wing VTOL UAVs, such as better range, endurance, and payload capacity. The increased range and endurance of fixed-wing UAVs makes them ideal for BVLOS operations. Most BVLOS operations involving fixed-wing UAVs are autonomous, and therefore it is important that the GNSS receiver used is able to provide a navigation solution at all times. The lack of this navigation solution can compromise the stability of the aircraft resulting in property damage and even loss of the aircraft. Most small rotary-wing VTOL UAVs usually operate within visual line of sight. In the case of compromised stability, an operator may be able to perform a controlled flight to land. Since most BVLOS operations and other similar operations involving fixed-wing UAVs will be in open sky conditions, the GNSS receivers on these platforms can become increasingly susceptible to intentional and unintentional GNSS signal interference. This may result in a GNSS outage for an extended period. In this case and other similar scenarios where a GNSS outage could occur, a VDM integration architecture may offer a navigation solution that allows the continued and safe operation of the aircraft without adding extra weight and cost to the overall platform.

Therefore, to achieve the research aim, the following objectives were determined:

- To investigate the navigation performance and quantify limitations of the current state-of-the-art VDM navigation scheme(s) during GNSS outages.
- To propose a novel integration algorithm that reduces drift in the navigation solution during an extended GNSS outage lasting over one minute without adding extra weight and cost to small UAVs.
- To undertake simulated data testing and practical testing of the proposed integration algorithm.

1.4 Contribution to Knowledge

This thesis makes a contribution to knowledge in the area of VDM navigation with specific applications to fixed-wing UAVs. This is demonstrated by the publication of three journal papers and one conference paper, which can be found in the List of Publications. The contribution is made in six steps:

i. The review of existing VDM integration algorithms and identification of architectures that give the most robust navigation solution during GNSS outages using low-cost sensors.

ii. The development of a six-degree-of-freedom (6DOF) fixed-wing aircraft model to generate GNSS and IMU datasets to investigate the performance of the integration architecture(s).

iii. The identification of approaches that can mitigate rapid error growth during GNSS outages without adding extra weight and cost to a fixed-wing UAV.

iv. The development and testing of a novel integration architecture that implements the approaches in (iii) using the available dataset from (ii).

v. The characterisation of the aerodynamic and propulsion model of a small (MTOW < 4 kg) off-the-shelf fixed-wing UAV to obtain model parameters used to test the developed architecture.

vi. The testing of the developed algorithm using flight data gathered from the small UAV fitted with a MEMS-grade IMU and a GNSS receiver to validate the results obtained in simulation.

1.5 Thesis Outline

Chapter 2 gives an overview of GNSS principles and strapdown inertial navigation. The GPS L1 signal is introduced as well as the different errors affecting the signal. The errors exhibited by inertial sensors are also be presented, followed by a review of common integration architectures and filters. The chapter then presents a detailed review of different VDM navigation schemes, highlighting their strengths and weaknesses.

Chapter 3 gives an overview of the main building blocks of a 6DOF aircraft model used in this research. Different ways of representing the aircraft's attitude are discussed, and the equations of motion are presented. The chapter then goes on to highlight the atmospheric model and gravity model used in this research. The aerodynamic and propulsion models are also presented, followed by the implementation of the 6DOF aircraft model in a simulation environment.

Chapter 4 evaluates the navigation performance of the current state-of-the-art VDM navigation techniques. The chapter identifies and quantifies the limitations of the most recent VDM navigation scheme by comparing its performance to a model-based navigation architecture developed during the research. The chapter then presents the characteristics of the navigation solution errors in a VDM navigation scheme during different GNSS outage intervals followed by reacquisitions. The chapter concludes by highlighting the main limitations of state-of-the-art VDM navigation techniques and reframes the research question.

Chapter 5 details the development of a novel, tightly coupled VDM-based integration architecture that reduces the growth of the navigation solution errors during GNSS outages compared to state-of-the-art techniques. The architecture uses a VDM as the main process model, while raw GNSS observables alongside IMU measurements aid the navigation solution. The chapter also details the development of a software-based GNSS measurement simulator alongside the error models used to generate the raw GNSS observables. The chapter then presents the simulation setup used to evaluate the performance of the developed algorithm, followed by a detailed discussion of the results.

Chapter 6 presents the flight testing campaign to validate simulation results of the proposed architecture using flight data gathered from a small fixed-wing UAV. The chapter then presents the characterisation of the aerodynamic and propulsion models using a combination of wind tunnel testing, full-scale oscillation tests and a geometry-based technique. The chapter then describes the test flight conducted to gather data used to test the architecture. The derivation of the reference position, velocity and attitude solution is described, followed by a discussion of the results.

Finally, in Chapter 7, a detailed summary is presented. Conclusions are drawn alongside recommendations for future work.

Figure 1.6 shows a breakdown of the objectives and a high-level overview of the work carried out in this thesis, including the main highlights of each chapter as outlined in this section.

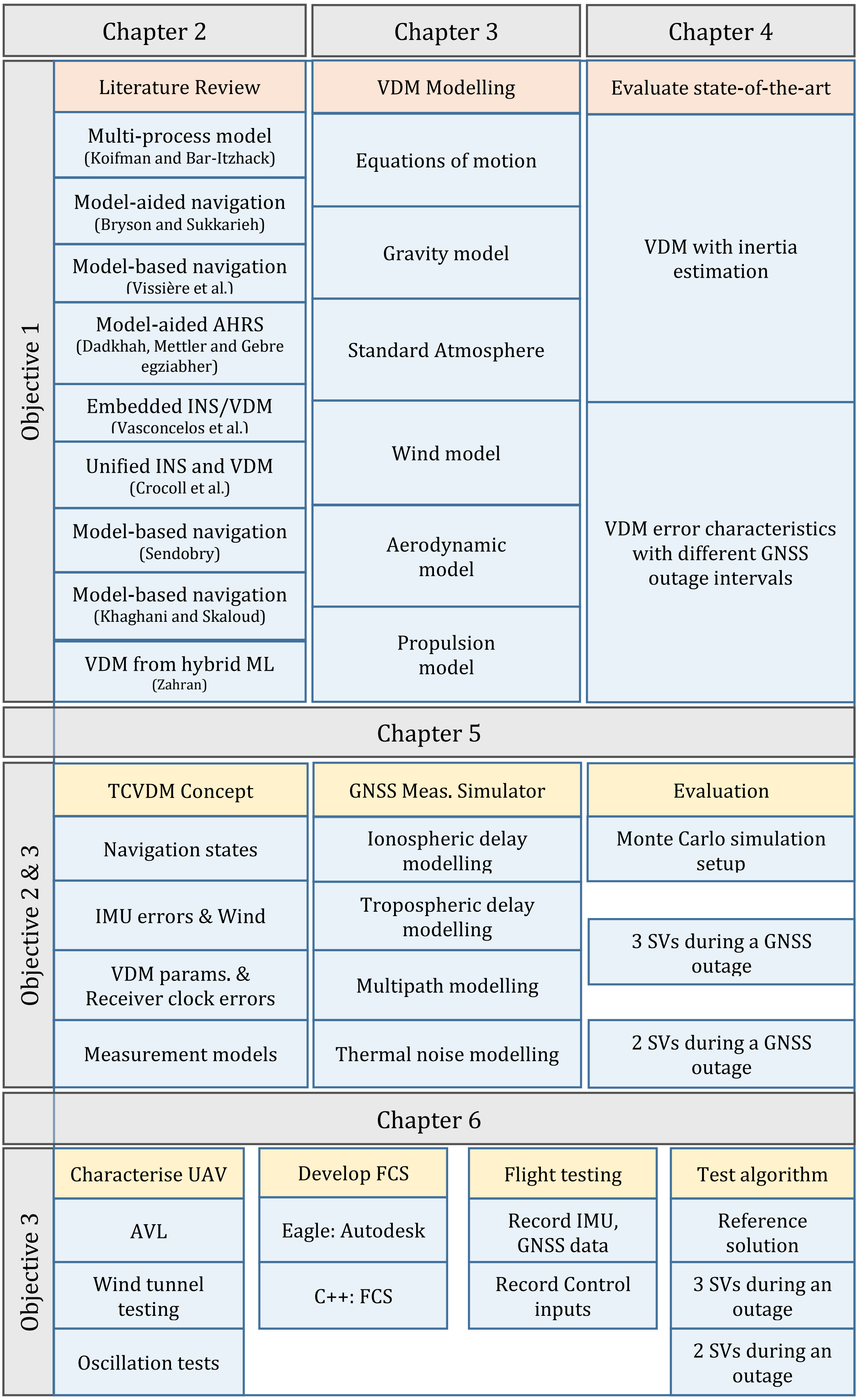


Figure 1.6. Breakdown of research objectives.

2 Background and Literature Review

2.1. Introduction

The primary function of an aircraft navigation system is to provide an accurate and consistent estimate of the aircraft's position, velocity and attitude. UAVs commonly use an inertial navigation system (INS) integrated with a global navigation satellite system (GNSS) to provide a filtered and quasi-continuous navigation solution. In low-cost applications, the quality of inertial sensors used is relatively low, affecting the performance of the integrated navigation solution, especially during a GNSS outage. Therefore, this chapter reviews the different systems and sensors found on most small UAVs used to provide an integrated navigation solution.

This chapter is organised as follows: Section 2.2 presents an overview of GNSS principles, focusing on the United States (U.S.) Global Positioning System (GPS). Section 2.3 discusses strapdown inertial navigation. The section focuses on low-cost MEMS-grade inertial sensors suitable for use in small UAVs. Different errors exhibited by these sensors are discussed, followed by a brief overview of integration architectures and filters. Justification for integrating an INS with a GNSS is also provided by briefly outlining the complementary characteristics of the two systems. Section 2.4 presents the current VDM navigation architectures, highlighting their strengths and weaknesses. Section 2.5 presents a summary of this chapter.

2.2. GNSS Principles

This section gives a brief overview of GNSS principles. The current systems are briefly discussed, followed by a review of different segments. The GPS signal acquisition and tracking principles are also briefly reviewed. The section places emphasis on different errors affecting GNSS ranging signals. A complete description of different GNSS systems and methods is beyond the scope of this section; instead, the reader is directed to works such as Kaplan and Hegarty (2017), Hofmann-Wellenhof, Lichtenegger and Wasle (2008), and Groves (2013).

Global navigation satellite systems have now been available for civilian use for almost three decades. The primary use for most civilian applications is in absolute positioning and timing. With knowledge of satellite positions, GNSS receivers can compute the absolute position through a process called trilateration. The range to each satellite is determined using a binary code signal borne on a radio frequency carrier signal and used to estimate the receiver's position. The GNSS signal propagating from the satellite is influenced by different error sources, which eventually reduce the accuracy of the computed navigation solution. The vast majority of GNSS receivers use a quartz oscillator, with more using temperature-compensated crystal oscillators (TCXO) as the frequency standard (Groves, 2013). For instance, the u-blox NEO-M8N, NEO-M8T, and the NoVAtel OEMStar GNSS receivers use a TCXO (NovAtel, 2011; u-blox, 2020). These oscillators are relatively low cost. They can introduce significant errors to the computed solution due to the lack of synchronisation with the transmitting satellites. The lack of synchronisation between the receiver and transmitting satellites leads to a common timing offset for all received satellite signals. This common offset also needs to be resolved for the receiver to output useful position information. For this reason, a GNSS receiver needs to track at least four satellites to resolve the absolute position and timing offset fully. There are usually more than four satellites in view at any given time, and therefore the position can be refined and consistency checks performed.

2.2.1. Current Systems

The first satellite navigation system was called TRANSIT, also known as NAVSAT (Navy Navigation Satellite System). The system was operative from 1964 and used by the U.S. Navy to periodically calibrate inertial systems on their submarines (Capuano, 2016). Position determination was accomplished using Doppler shift of radio signals transmitted by a limited number of satellites (in Earth orbit ~ 1100 km), providing a fix only every hour or more. Russia operated a similar system around the time known as Tsikada (Groves, 2008).

At the time of this writing, operational GNSS include the Global Positioning System (GPS), owned and operated by the U.S government, GLONASS, owned and operated by Russia, Galileo, funded by the European Union and managed by the European Space Agency (ESA), and BeiDou, owned and operated by China. The status of these systems at the time of this writing is presented in Table 2.1.

Table 2.1. Status of current GNSS systems (European Union, 2016; Russian Space Systems, 2016; China Satellite Navigation Office, 2017; Dunn, 2018).

System	Country	Coding	Orbital height and Period	Operational satellites	Status
GPS	US	CDMA	20,200 km 12h	≥ 30	Operational
GLONASS	Russia	FDMA CDMA	19,100 km 11.3h	≥ 23	Operational
Galileo	EU	CDMA	23,222km 14.1h	≥24	Operational
BeiDou	China	CDMA	21,528 km 12.6h	≥35	Operational

2.2.2. GNSS Segments

Any GNSS can be divided into three main segments: the space segment, the user segment, and the control segment, as shown in Figure 2.1.

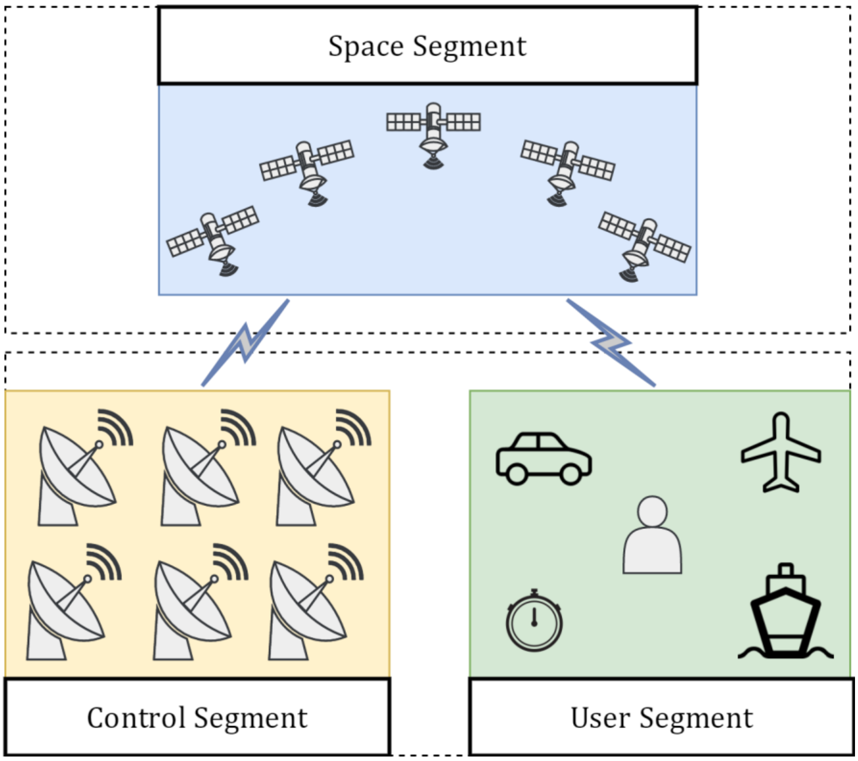


Figure 2.1. GNSS segments.

The space segment consists of the constellation of GNSS satellites used in positioning and timing applications. Sufficient satellites are needed to ensure global availability at all times. All current systems use satellites in medium earth orbit (MEO), approximately 20,000 km from the Earth's surface. The satellites usually have different orbit configurations to meet specific needs and achieve a certain level of performance. The satellites, otherwise called space vehicles, broadcast signals to both the control and user segments. Each GNSS broadcasts a range of different signals, many of which are open to all users free of charge. Others are restricted to military users, emergency services, commercial subscribers and security services.

A network of ground monitoring stations, uplink stations and one or more control stations make up the control segment (CS). This segment continuously monitors each satellite and provides 'health warnings' in the event of a malfunction. The control stations compute the satellite orbits and produce 'ephemerides' to enable the user to compute the satellite position. Further, the stations monitor the satellite clocks and provide correcting information in the satellite transmitted message that the user can use to correct any clock errors.

The user segment consists of the GNSS receivers that utilise the information received from the satellites for positioning and timing. Modern low-cost receivers can have more than 50 channels and track multiple satellites from different constellations, which improves availability and helps with integrity monitoring. It is even possible to have relatively cheap (<£400) GNSS receivers that can track satellite signals on multiple frequencies and multiple constellations, such as the u-blox ZED-F9P. The use of multiple frequencies can significantly improve navigation performance by eliminating the ionospheric delay through linear combinations of the measurements. These advantages were only available to high-end receivers which generally have a larger form factor, require more power, and specialised antennas.

2.2.3. GPS

The GPS is currently undergoing a modernisation process that began in 2000 that aims to guarantee compatibility with other systems and facilitate interoperability. The modernisation process is a multibillion-dollar effort to upgrade the features and overall performance of the system. At the time of this writing, the GPS constellation consists of a mix of new and old satellites. The current constellation even includes GPS III/IIIF satellites, with the first one launched in 2018. The last Block IIA (2nd generation, "Advanced") satellite was decommissioned in 2019. The Block IIA and Block IIR ("Replenishment") satellites are categorised as legacy satellites. Block IIR-M ("Modernised"), Block IIF ("Follow-on"), GPS III, and GPS IIIF ("Follow-on") are considered the modernised satellites. Figure 2.2 shows different GPS blocks and the modernisation effort.

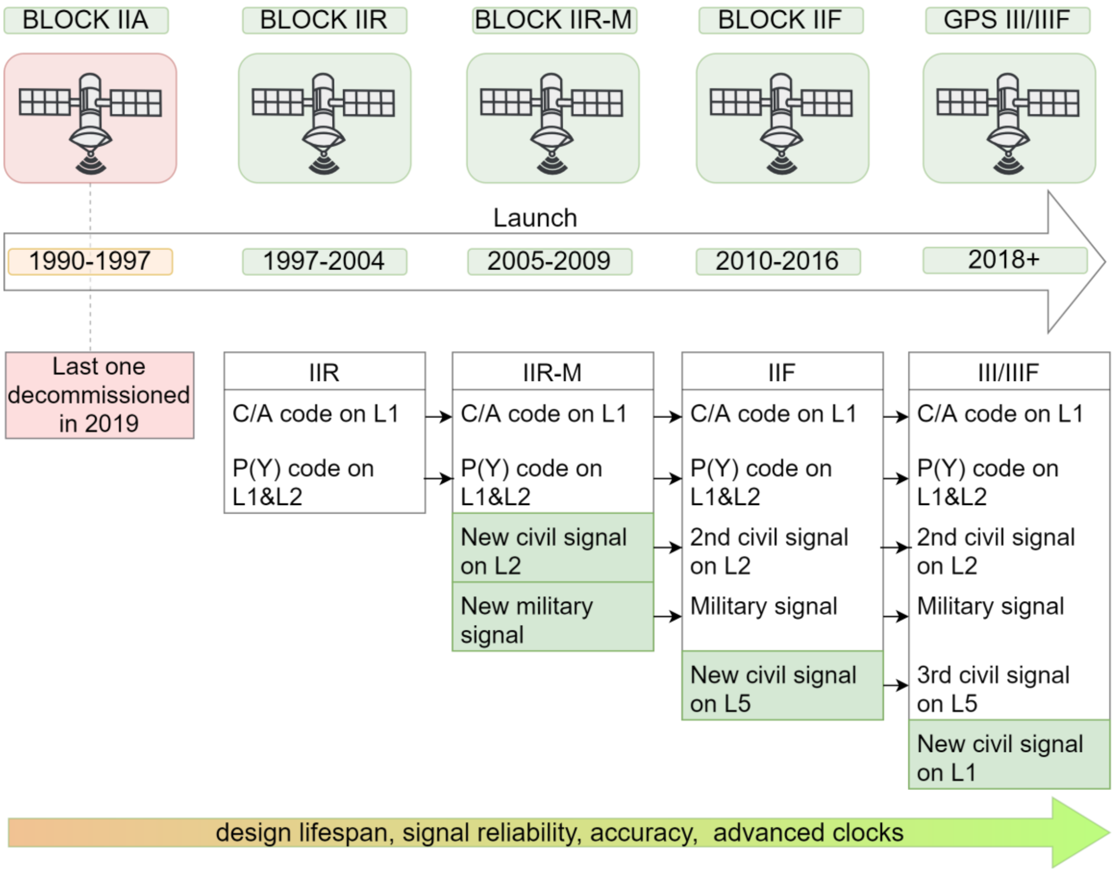


Figure 2.2. GPS satellite blocks showing the modernisation effort.

At the time of writing, there are eight Block IIR satellites in operation in addition to seven Block IIR-M satellites, twelve Block IIF satellites and three GPS III/IIIF satellites. This makes a total of thirty operational satellites in orbit.

The three frequencies bands used by GPS include L1 (1575.42 MHz), L2 (1227.6 MHz) and L5 (1176.45 MHz). Most low-cost mass-market GNSS receivers track the legacy GPS L1 C/A signal, and cheap dual-frequency receivers such as the u-blox ZED-F9P can track both L1 C/A and the civil signal on L2 (L2C). Due to a plethora of compatible low-cost GPS L1 C/A receiver equipment, the next section and this thesis, in general, will focus on this signal.

2.2.4. GPS L1 Open Signals

The GPS 'course/acquisition' (C/A) code is a 1023 bit pseudorandom binary code (PRN) that exhibits excellent correlation properties allowing each satellite to broadcast its unique code on the same frequency without significant interference. This technique is called code division multiple access (CDMA), in which satellites use different ranging codes that have low cross-correlation properties with respect to one another (Tawk, 2013). The code is broadcast repeatedly with a period of 1 ms. A ranging code and navigation message modulate the carrier wave leaving a satellite. The navigation message carries information about the satellite's orbit and clock. The ranging signal enables the determination of the time of transmission of the received signal. When used with the information in the navigation message, the receiver's position can be computed.

The C/A code is modulated on the L1 carrier phase signal using binary phase shift keying (BPSK). With this modulation, for each bit transition in the C/A code sequence, a phase shift of 180° is introduced to the carrier phase.

Besides information about the satellite's orbit and clock, the navigation message also contains information about the satellite's health and almanac data. The almanac can be used during signal acquisition since it contains coarse orbital information about other satellites. The message is broadcast at 50 bits per second and added to the C/A code before being modulated onto the carrier wave (Tawk, 2013).

Another civilian signal, L1C, also modulates the L1 carrier. The first satellite featuring the L1C signal was launched in 2018, as shown in Figure 2.2. This signal has a data and pilot channel. The pilot channel is useful in acquisition and tracking because the absence of data bit transition allows longer integration, improving sensitivity and robustness.

2.2.5. GPS Signal Acquisition and Tracking

A full description of the GPS signal acquisition and tracking is beyond the scope of this section. A good description can be found in (Borre *et al.*, 2007; Groves, 2013; Tawk, 2013). A GPS receiver needs to process the received satellite signals to output a navigation solution. The incoming carrier signal is first downconverted from the original L-band radio frequency to a lower intermediate frequency (IF) to allow a lower sampling rate to be used. The intermediate frequency has the same modulation as the incoming signal. The signal is then digitised for further processing including acquisition and tracking. Two parameters need to be determined during signal acquisition so that all the visible satellites can be identified. These parameters include the Doppler-shifted carrier frequency and the code phase.

Relative motion between a satellite and a receiver introduces a Doppler shift ($\Delta\tilde{f}^{s}_{ca,r}(\tilde{t}_r)$). This Doppler shift can also occur due to local oscillator frequency drift. Determination of the code phase enables the receiver to determine the start of the C/A code frame with respect to the receiver time ($\tilde{t}_r$). The signal transmission time ($\tilde{T}^s$) can then be deduced, which in turn is used to estimate the range from the receiver to the satellite. Because the satellite clocks and receiver clock are not perfectly synchronised to the GPS system time, this estimate is usually called a pseudorange ($\tilde{P}_r^s$).

The signal acquisition process provides coarse estimates of the Doppler-shifted carrier frequency and the code phase. The purpose of tracking is to refine these values and keep track of their change over time. A phase lock loop (PLL) is used to track the carrier phase ($\tilde{\phi}_{ca,r}^s$), and the code signal is tracked by a delay lock loop (DLL). These loops are used to generate error signals which are fed back to the oscillator to align the received signal with a locally generated replica. Figure 2.3 shows some of the blocks of a simple GPS receiver. The figure also shows some of the measurements output by the ranging processor (which acquires and tracks a satellite signal).

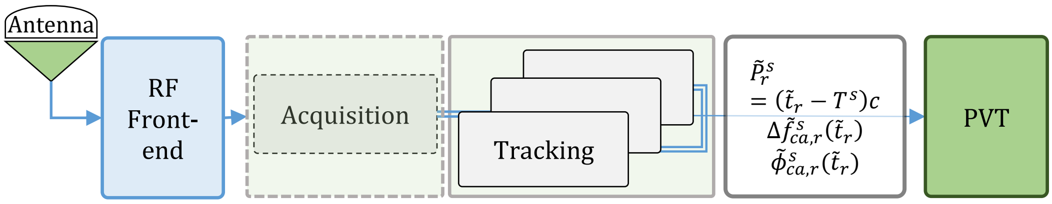


Figure 2.3. Blocks of a simple GPS receiver showing the raw GNSS observables.

The availability of raw GNSS observables (pseudoranges, Doppler frequencies) allows the receiver to compute its position and velocity. This is accomplished once the tracking loops are in lock and the ephemeris information decoded from the navigation data message. For GPS L1 C/A, the navigation message is a 1500 bit long frame lasting 30 seconds. It contains five subframes, with each frame lasting 6 seconds. One frame is required for the ephemeris and 25 frames for the almanac. The receiver synchronises to the start of each subframe using a unique preamble to decode the ephemeris. By synchronising to the preamble on each channel, the receiver can determine relative transit time with respect to a reference channel. This enables the computation of the first set of pseudoranges which are then propagated using the code phase measurements from the code tracking loop.

2.2.6. Error Sources

The derived pseudoranges alongside other observables are influenced by errors from different sources. These are discussed in this section and the modelling effort presented in Chapter 5.

The satellite clocks exhibit an error due to the cumulative effect of the oscillator noise. The ground control stations continuously monitor the satellite clocks, and clock correction parameters are made available to the receivers through the navigation message. The residual range error due to the satellite clock ranges from 0.3 m to 4 m, depending on the type of satellite and age of broadcast (Kaplan and Hegarty, 2006).

The control segment's prediction of a satellite's position will be different from its actual position. Therefore, the ephemeris error is the error in the prediction of the satellite positions. These errors are generally small in the radial direction (from a satellite toward the Earth's centre). The along-track and cross-track components are generally much larger and more difficult for the control segment to observe because they do not project significantly onto the line-of-sight vector toward the Earth. For the same reasons, a user does not experience large measurement errors due to the largest ephemeris error components with an effective range error on the order of 0.8 m (Kaplan and Hegarty, 2006).

The atmosphere influences the propagation speed of a GPS signal which manifests as a bias in the derived pseudorange and carrier-phase. In GPS processing, the atmosphere is usually modelled as being composed of two parts, the ionosphere and the troposphere. For the most part, a signal leaving the satellite travels at the speed of light in free space. However, as the signal enters the atmosphere, the signal propagation speed changes. The ionosphere is the electrically charged part of the atmosphere that extends from 70 km and extends to 1000 km above the Earth's surface. It is composed of charged particles that influence the signal propagation speed. Ultraviolet rays from the sun ionise portions of gas molecules and release free electrons. The electron density along the signal propagation path influences the propagation speed. The electron density varies throughout the day and time of the year, largely influenced by solar activity. The error introduced by the ionosphere ranges from a few metres and can reach 100 m (Pinchin, 2011). On the other hand, the troposphere is electrically neutral and extends from the surface of the Earth to a height of about 40 km. The error induced by the troposphere is a function of the local temperature, pressure, relative humidity and receiver's altitude. The uncompensated range equivalent of this delay can vary from about 2.4 m for a satellite at the zenith and a receiver at sea level to about 25 m for a satellite at a low elevation angle (Kaplan and Hegarty, 2006). The troposphere is usually modelled as being comprised of a dry part and a wet part. The dry component consists mainly of dry air and constitutes 90% of the total delay. The dry component can be predicted very accurately, while the wet component is often difficult to predict due to the uncertainties in the atmospheric distribution. The navigation message includes correction parameters for the ionospheric error that can correct up to 50% of the error. This correction model is useful for single-frequency users and is also utilised in this research, as explained in Chapter 5. The residual ionospheric delay averaged over the globe (and elevation angles) is around 7 m (Kaplan and Hegarty, 2006).

The receiver noise can be thought of as being comprised of system noise and tracking loop noise. System noise is generated by the receiver electronic hardware and includes thermal noise, which also includes the contribution of the antenna. The quality of the components affects the receiver's performance, with lower quality components producing more noise. The tracking loop noise is determined by the loop bandwidth and the incoming signal strength (Pinchin, 2011; Richardson, Hill and Moore, 2016). The range equivalent for this error has been shown to have more variation with the incoming signal strength than its phase equivalent counterpart (Richardson, Hill and Moore, 2016). The receiver settings can be changed to improve tracking loop performance to cope with high dynamics situations at the expense of increased noise. Pinchin (2011) showed that platform dynamics directly impact the tracking performance, which in his experiments resulted in the loss of phase lock (indicated by cycle slips) and increased biases in the code measurements. Most methods used within a receiver are proprietary, and therefore the exact cause for loss of phase tracking and code biases was unknown.

Multipath can also significantly contribute to the pseudorange error budget. This error is caused by reflected signals from a satellite arriving at the antenna in addition to the direct signal. For air applications, most signal reflections are from the host-vehicle body. The reflected signals arrive with a delay and a different amplitude compared to the direct signal. These signals influence the correlation properties of the line-of-sight signal and eventually influence the receiver's tracking performance. Consequently, this introduces ranging errors (code multipath) and carrier-phase errors (carrier multipath). Because the wavelength of the PRN code is much larger than the wavelength of the carrier, the multipath error on code measurements is larger. Different techniques exist to mitigate multipath, including antenna designs, improved correlation techniques, which are mostly proprietary for most commercial receivers, and improved signal design with new modulation schemes that improve correlation properties. Another approach includes estimating the different multipath components and removing their effects in the channel observations (Tawk, 2013). It is also important to mention that the error also depends on the environment in which the receiver operates and can be different even for receivers within one metre of each other (Pinchin, 2011).

2.3. Strapdown Inertial Navigation

A strapdown inertial navigation system is a dead-reckoning form of navigation with an IMU fixed and aligned with the orthogonal body axes of the host platform. The performance, size and mass of inertial sensors within an IMU vary by several orders of magnitude. In low-cost applications, the quality of the inertial sensors used is relatively low. As explained in the previous chapter, if used alone, errors in these sensors will cause the navigation solution to drift.

This section presents an overview of inertial sensors and, particularly, MEMS inertial sensors. Typical errors affecting these sensors are also presented. State-of-the-art INS/GNSS coupling techniques are also discussed, and some of their limitations are highlighted.

2.3.1. MEMS Inertial Sensors

MEMS accelerometers usually consist of a proof mass constrained to move in a single axis and a pickoff to sense the applied specific force. The common types of accelerometers are pendulous and vibrating beams (Groves, 2008). With a pendulous accelerometer, the proof mass forms a pendulum with the accelerometer case. Vibrating-beam accelerometers consist of a vibrating beam supporting the proof mass along the sensitive axis. Motion causes a change in the beam's resonant frequency, and this can be used to deduce the applied specific force in the sensitive axis. Vibrating-beam accelerometers tend to have higher accuracy capability than pendulous types (Titterton and Weston, 2004). Other types of MEMS accelerometers include the tunnelling type and electrostatically levitated type. The tunnelling type accelerometer uses a control electrode to deflect a cantilevered beam (proof mass) using electrostatic force into a position called the tunnelling position. The control electrode registers an applied acceleration force on the proof mass as the change in the applied potential to maintain the beam tunnelling position. These devices, however, have a limited dynamic range (Titterton and Weston, 2004).

Gyroscopes typically form the most expensive part of an inertial navigation system, with their performance often being the limiting factor for the overall navigation solution accuracy (Tawk, 2013). Low-cost MEMS-grade gyros contain a vibrating silicon structure used to measure the angular rotation about an input axis using the deflection caused by the Coriolis acceleration perpendicular to the input axis and proportional to the input rotation (Hide, 2003). There are generally three practical sensor configurations based on this principle, including simple oscillators (single vibrating mass), balanced oscillators (tuning fork gyroscope) and shell resonators (cylinder, ring oscillators) (Titterton and Weston, 2004).

As a result of high volume manufacturing using chemical etching and batch processing, MEMS-based inertial sensors are generally low cost. They also have a small size, low weight, rugged construction, and low power consumption (Tawk, 2013). Consequently, this size reduction creates challenges in attaining good performance and has led to decreased sensitivity and an increase in noise (Titterton and Weston, 2004). Figure 2.4 shows typical 3-axis MEMS accelerometer and gyroscope chips found in most low-cost applications.

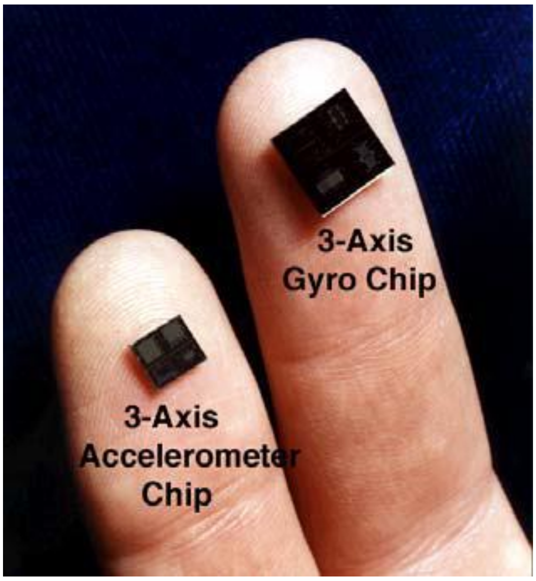


Figure 2.4. Typical 3-axis MEMS accelerometer and gyroscope chips (Barbour, 2011).

2.3.2. Error Characteristics

Inertial sensors exhibit different types of errors, including biases, scale-factor, cross-coupling and random noise. The order of these errors depends on the quality of the sensors. Some systematic errors, such as temperature-dependent bias variation, can be removed through a laboratory calibration routine, and others can only be corrected by integrating with other navigation sensors. Typically, stochastic processes are used to model different errors exhibited by these sensors.

Sensor bias comprises a static component and a dynamic component. The static component (also known as the turn-on-bias or bias repeatability) varies from run to run but remains fixed during each run. The dynamic component is usually referred to as in-run bias variation or bias instability and varies during each run and incorporates a residual temperature-dependent variation following a laboratory calibration.

Scale-factor, sometimes called sensitivity, is the ratio between the measured output and the change in the sensed input. A deviation of this input-output gradient is the scale-factor error. Scale-factor errors in low-cost MEMS-grade sensors can be as high as 10% (Groves, 2013).

Cross-coupling errors arise from misalignment of the sensitive axes of the sensors with respect to the orthogonal axes of the body on which the sensor assembly is mounted. As a result of this misalignment, the inertial sensors along a given sensitive axis become sensitive to inputs on other orthogonal axes. This can result in additional scale-factor errors but are usually several orders of magnitude smaller than the cross-coupling errors (Groves, 2013). It is possible to have cross-coupling errors as a result of cross-talk between individual sensors.

Random noise can result from several sources. For very weak signals, electrical noise could limit the resolution of the sensors, and for vibratory sensors, high-frequency resonances could significantly degrade the performance. For frequencies below 1 Hz, the spectrum for accelerometer and gyro noise is approximately white (Groves, 2013). The lack of correlation of white noise samples means it cannot be compensated. The direct integration of random white noise results in random walk in attitude for the case of gyros and random walk in velocity for the case of accelerometers. Operating in an environment with high vibrations can effectively increase the random white noise exhibited by the sensors. It can also potentially introduce time-correlated components if the external vibration frequency is close to the inertial sensor's resonant frequency. Filtering and De-noising techniques can be used to reduce high-frequency noise (Quinchia *et al.*, 2013). More recently, deep-learning techniques have been applied to de-noise gyroscope measurements using a dilated convolutional neural network (Brossard, 2020).

2.3.3. Error Modelling

Usually, different stochastic processes are used to model the different errors exhibited by inertial sensors. The common stochastic processes used include a random constant process, a random walk process, and a first-order Gauss-Markov process.

A random constant process can be used to approximate a constant error for a given time. The differential equation for this process is given by:

$$\dot{x}(t) = 0 \tag{2.1}$$

$$x_{k+1} = x_k$$

where k represents the time index. The static component of the sensor bias can be modelled by this process using information such as the mean and variance of the error.

A random walk process is a result of integrating random white noise ($w(t)$). The differential equation for this process is given by:

$$\dot{x}(t) = w(t) \tag{2.2}$$

$$x_{k+1} = x_k + w_k$$

As previously explained, the integration of white noise in the specific force measurements leads to velocity random walk, while the integration of white noise in the gyroscope measurements leads to angle random walk.

A first-order Gauss-Markov process is usually used to describe a coloured noise signal. Its differential equation is given by:

$$\dot{x}(t) = -\beta x(t) + w(t) \tag{2.3}$$

$$x_{k+1} = e^{-\beta \Delta t} x_k + w_k$$

where β is the inverse of the correlation time, τ_c, and Δt is the integration interval. With a significantly large correlation time (as β approaches zero), the process resembles a random walk process, while with a very short correlation time, the process resembles white noise. The scale-factor errors and the dynamic component of the sensor bias can be modelled as a first-order Gauss-Markov process and estimated in an integration architecture.

2.3.4. Integration Architectures

The integration of INS and GNSS has been widely adopted to improve overall navigation performance. The two systems have complementary characteristics, which makes them a perfect match when integrated. An INS generally has good short-term stability but poor long-term performance. On the other hand, GNSS has good long-term stability and limited short-term performance. Errors in the receiver tracking loops, clock instability, multipath, variation in satellite geometry, and low received signal strength can significantly affect the accuracy of the navigation solution output by a GNSS receiver. The integration of the two systems leads to an integrated navigation solution with the combined advantages of the two systems. Some of the main characteristics of the two systems are presented in Table 2.2. An integrated navigation system provides both short- and long-term stability, improved availability and greater integrity. When the quality of the inertial sensors used is low, the INS solution in an INS/GNSS integration architecture is typically corrected using a closed-loop implementation. The errors estimated in a closed-loop implementation are continuously fed back to correct the INS. This keeps the estimated errors small. Other configurations include the open-loop and total-state implementation discussed in detail in Groves (2013).

Table 2.2. Pros and Cons of INS and GNSS.

	INS	GNSS
Pros	High output data rate	No initial information required
	Provides relative position, velocity and attitude information	Provides absolute position, velocity and time information
	Good short-term stability	Good long-term stability
	Partially independent of the operating environment	Time Independent
	Not susceptible to RF interference	Time standard
Cons	Needs a good initial estimate	Susceptible to RF interference
	Poor long-term stability	Limited short-term stability
	Can be influenced by external vibrations	Environment dependent
	Time-dependent	Low data rate

The integration of INS and GNSS can be grouped into three typical integration strategies: loosely coupled integration, tightly coupled integration, and deeply coupled integration.

Loosely coupled integration architectures fuse independent position and velocity solutions (P, V) from the two systems to provide a blended navigation solution, as shown in Figure 2.5. This is the simplest method of integrating an INS with a GNSS receiver.

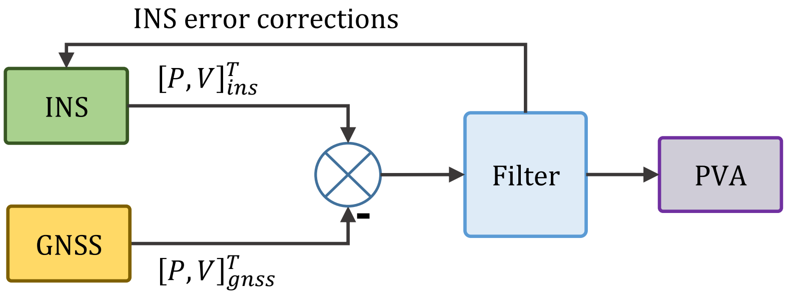


Figure 2.5. A loosely coupled integration architecture.

Usually, an extended Kalman filter (EKF) is used to blend the two solutions with increased interest over the last decade in the use of non-linear estimators such as the unscented Kalman filter (UKF) and even particle filters (Tawk, 2013). The two filters, the EKF and UKF, will be explained further in the next section due to their relevance to this research. Loosely coupled schemes inherently require at least four satellites to provide a drift-free navigation solution. With less than four satellites, no position and velocity information from the GNSS receiver will be available for the filter update step (assuming the receiver is outputting a single-epoch PVT solution). Therefore during an outage (when tracking less than four satellites), the accuracy of the integrated solution heavily depends on the quality of the inertial sensors used.

Tightly coupled integration architectures fuse raw GNSS observables, typically pseudorange and Doppler frequency measurements, with their INS estimates to produce a single navigation solution, as shown in Figure 2.6. This generally improves accuracy because the raw observables are not as correlated as the position and velocity estimates used in the loosely coupled approach (Tawk, 2013). Further, such as scheme can take advantage of measurements available even during a GNSS outage to limit the error growth. It is possible to have Doppler aiding information fed back to the GNSS receiver in this architecture, as done by Tawk *et al.* (2014). This can significantly improve tracking loop performance due to narrower tracking-loop bandwidth leading to improved noise resistance and sensitivity (Groves, 2008).

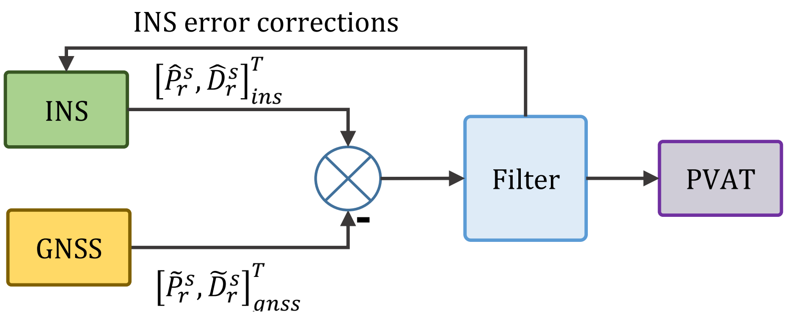


Figure 2.6. A tightly coupled integration architecture. The vectors $[\tilde{P}_r^s, \tilde{D}_r^s]_{gnss}^T$ and $[\hat{P}_r^s, \hat{D}_r^s]_{ins}^T$ represent the measured and estimated pseudorange and Doppler frequency from a receiver (r) to a satellite (s) by the GNSS receiver and INS, respectively.

Deep integration architectures combine both GNSS tracking and navigation into a single integration filter, as shown in Figure 2.7.

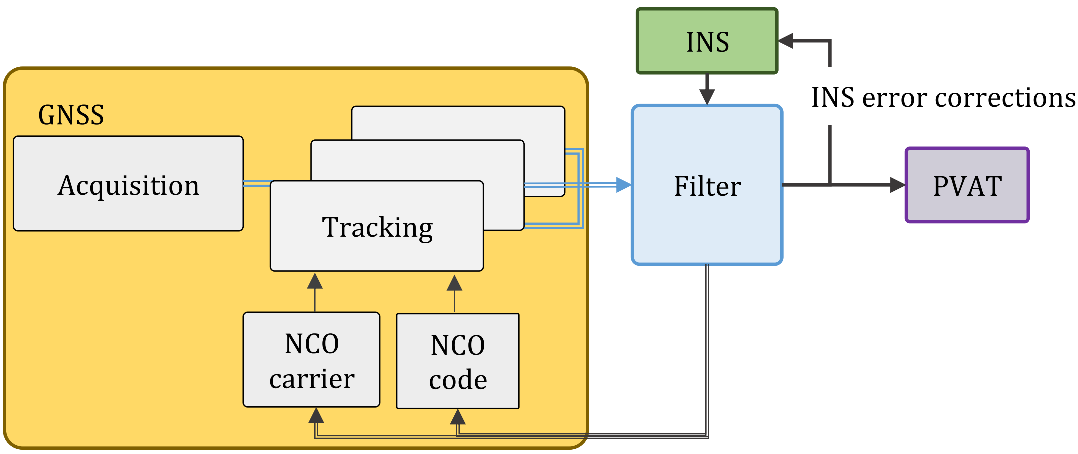


Figure 2.7. A deep integration architecture.

In this architecture, the integrated navigation solution directly controls the numerically controlled oscillators of the underlying tracking loops. Usually, the tracking architecture is modified from scalar to vector, where all the channels are dependent on each other and controlled by the integration filter. The integration filter can be fed with information from the discriminators or the correlators. As a result of improved tracking performance, this architecture has good performance even during high dynamics, improved sensitivity to weak signals and even improved anti-jamming performance (Groves, 2013). The benefits of deep integration come at the expense of increased complexity and computation cost with tight time synchronization.

2.3.5. Integration Filters

An integration filter is an estimator or a mathematical algorithm that systematically combines information from multiple sources. A common estimator used for integrating an INS and GNSS is the Kalman filter introduced by Kalman (1960). Rather than a filter, it is a Bayesian estimation technique that works recursively to update its estimates as a weighted average of the current measurement data and previous estimates. A standard Kalman filter is structured to produce an unbiased estimate for a linear system. For a nonlinear system or measurement model, a standard Kalman filter is no longer optimal. The usual approach is to linearise the models about a continually updated trajectory by new measurements (Groves, 2008; Brown and Hwang, 2012; Tawk, 2013). The resulting filter is called an extended Kalman filter (EKF).

An extended Kalman filter uses the first-order terms of the Taylor series expansion of a nonlinear system and measurement model and applies the standard Kalman filter theory (Gelb *et al.*, 1974). A nonlinear system and observation model is given by:

$$\dot{x}(t) = f(x(t), u_d, t) + G(t)w(t) \tag{2.4}$$

$$z = h(x, t) + v(t) \tag{2.5}$$

where: f is the nonlinear system model,
G is the noise shaping matrix,
w is the system noise vector assumed to be Gaussian,
u_d is a deterministic forcing function,
h is the nonlinear observation model,
v is the measurement white noise vector,
x, z represent the state vector and measurement vector, respectively.

Assuming the error in the estimated states is much smaller than the states themselves, a linear dynamic and measurement model is given by:

$$\dot{X} = F \cdot X + Gw(t) \tag{2.6}$$

$$Z = H \cdot X + v(t) \tag{2.7}$$

where F is the linearised dynamic matrix, H is the linearised observation matrix, and X is the state vector.

The discrete propagation of the states and associated error states over an iteration of the filter is given by:

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} + \int_{t-\tau_s}^{t} f(\hat{x}, u_d, t')dt' \tag{2.8}$$
$$\hat{X}_{k|k-1} = \Phi_{k-1}\hat{X}_{k-1|k-1}$$

and the associated predicted error covariance matrix is given by:

$$P_{k|k-1} = \Phi_{k-1}P_{k-1|k-1}\Phi_{k-1}^{T} + Q_{k-1} \tag{2.9}$$

The state transition matrix can be approximated as:

$$\Phi_{k-1} = \exp(F_{k-1}\tau_s) \tag{2.10}$$

which can be computed as a power-series expansion of the dynamic matrix, F, and propagation interval, τ_s:

$$\Phi_{k-1} = \sum_{m=0}^{\infty} \frac{F_{k-1}^{m}\tau_s^{m}}{m!} \tag{2.11}$$

where the dynamic matrix is given by:

$$F_{k-1} = \frac{\partial f(x, u_d)}{\partial x} \bigg| x = \hat{x}_{k-1|k-1}, u_d = u_{k-1} \tag{2.12}$$

The simplest form of the process noise covariance, Q_{k-1}, is obtained by neglecting the time propagation of the system noise over an integration interval (Groves, 2013). This is given by:

$$Q_{k-1} = G_{k-1}E\left(\int_{t-\tau_s}^{t}\int_{t-\tau_s}^{t} w(t')w^{T}(t'')dt'dt''\right)G_{k-1}^{T} \tag{2.13}$$

And in the limit, $\tau_s \to 0$, the equation simplifies to (the impulse approximation):

$$Q_{k-1} = G_{k-1}Q_{s,k-1}G_{k-1}^{T}\tau_s \tag{2.14}$$

where $Q_{s,k-1}$ is the spectral density matrix.

The updated state estimate is given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h\left(\hat{x}_{k|k-1}\right)\right) \tag{2.15}$$

and the observation matrix is given by:

$$H_k = \frac{\partial h(x)}{\partial x} \bigg| x = \hat{x}_{k|k-1} \tag{2.16}$$

and the updated state covariance matrix is given by:

$$P_{k|k} = (I - K_kH_k)P_{k|k-1}(I - K_kH_k)^{T} + K_kR_kK_k^{T} \tag{2.17}$$

where K_k is the Kalman gain and is computed as:

$$K_k = \frac{P_{k|k-1} H_k^T}{H_k P_{k|k-1} H_k^T + R_k} \tag{2.18}$$

and R_k is the measurement noise covariance matrix, and the denominator in the computation of the Kalman gain is called the innovation covariance (S_k).

Terms such as efficiency and consistency are typically used when dealing with Kalman filters. Here, these terms are briefly defined. If a second random variable, y is related to x through a nonlinear transformation, the transformed statistics are consistent given the following inequality (Julier and Uhlmann, 1997):

$$P_{yy} - E[\{y - \bar{y}\}\{y - \bar{y}\}^T] \geq 0 \tag{2.19}$$

where: $\bar{y}$ is the mean and P_{yy} is the covariance of y.

If the statistics are not consistent, the value of P_{yy} is under-estimated. If the Kalman filter uses the inconsistent set of statistics, it could lead to divergence since the filter places too much weight on the information and underestimates the covariance. Efficient transformation dictates that the value on the left-hand side of the inequality is minimised, which implies the covariance of the transformed random variable (P_{yy}) should closely match the actual mean squared error. And for an unbiased estimate: $\bar{y} \approx E[y]$. The EKF uses the linearised system and measurement models to propagate the mean and covariance of a random variable. The approximation is accurate only if the second and higher-order terms of δx in the mean, and fourth and higher-order terms of δx in the covariance are negligible (Julier and Uhlmann, 1997). Therefore, for a highly nonlinear system and measurement model, this linearisation can introduce biases that can significantly affect the filter's performance. This problem is amplified when dealing with large state errors and very precise measurements. In this case, applying the standard extended Kalman filter equations leads to a condition where the covariance matrix decreases more rapidly than the actual state errors. This under-estimation eventually leads to the filter ignoring new measurements even in the presence of large residuals (the filter no longer gives consistent estimates). It is possible to include second-order terms, but this comes at a high computational cost. The space shuttle, for example, utilised an *ad hoc* technique known as underweighting to account for second-order terms (Zanetti, DeMars and Bishop, 2009). Underweighting slows down the convergence of the covariance matrix. For a nonlinear measurement model, the innovations truncated to first-order are given by:

$$\epsilon_k = z_k - h\left(\hat{x}_{k|k-1}\right) \approx H_k e_{k|k-1} + v_k \tag{2.20}$$

where: $e_{k|k-1}$ is the *a priori* estimation error.

And the innovation covariance matrix is given by:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.21}$$

The updated covariance matrix can be multiplied by H_k and H_k^T and represented in the following form (Zanetti, DeMars and Bishop, 2009).

$$H_k P_{k|k} H_k^T = H_k P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k\right)^{-1} R_k \tag{2.22}$$

And if the contribution of the *a priori* estimation error $H_k P_{k|k-1} H_k^T$ to the innovation covariance is much larger than the measurement covariance matrix R_k as a result of very precise measurement, the EKF will produce an updated covariance matrix that approximates the measurement noise covariance matrix.

$$H_k P_{k|k} H_k^T \approx R_k \tag{2.23}$$

If the contributions of the second-order terms of the Taylor series expansion of $h(x)$ are given by B_k, and have a comparable magnitude to the measurement error, the innovation covariance matrix is modified to:

$$S_k = H_k P_{k|k-1} H_k^T + B_k + R_k \tag{2.24}$$

Then the *a posterior* covariance can be approximated by: $H_k P_{k|k} H_k^T \approx R_k + B_k$. Therefore, in the presence of nonlinearities and by truncating the Taylor series to first-order, the EKF can underestimate the *a posterior* covariance. To overcome some of these challenges (when using the EKF), the unscented Kalman filter is briefly discussed.

The unscented Kalman filter uses a deterministic choice of sampling points, usually called sigma points, to represent the state estimate's conditional density (Julier and Uhlmann, 1997; Brown and Hwang, 2012). The filter's name takes after the Unscented Transform (UT). This is used to calculate the statistics of a random variable following nonlinear transformation. The discrete sigma points are projected through a nonlinear transform and the Unscented Transform provides an estimate of the mean and covariance of the associated random variable. For a Gaussian random variable, the UKF can accurately capture the posterior mean and covariance to the third-order. For non-Gaussian inputs, approximations are accurate to at least the second-order (Julier and Uhlmann, 1997).

For a given nonlinear function $f(\cdot)$ that maps x to y, the Unscented Transform uses the mean and covariance information for x and set of sample points to estimate the mean and covariance information for y. The sample points are selected from the probability distribution of x. For an N-dimensional state, $2N + 1$ points are chosen to be a minimal set of the random variable over the probability distribution domain of the variable.

The sigma points are generated using:

$$x_i = \begin{cases} \bar{x} & i = 0 \\ \bar{x} + \left(\sqrt{(N+\lambda)P_x}\right)_i & i = 1, \dots N \\ \bar{x} - \left(\sqrt{(N+\lambda)P_x}\right)_{i-N} & i = N+1, \dots 2N \end{cases} \tag{2.25}$$

where: $\lambda = \alpha^2(N + \kappa) - N$,

α determines the spread of the sigma points about $\bar{x}$,

κ is a scaling factor,
N is the dimension of x.

The transformation of the generated sigma points is given by:

$$y_i = f(x_i) \tag{2.26}$$

To compute the mean and covariance of the transformed points, weights need to be defined. These are given by:

$$weights: \begin{cases} W_i^{(m)} = \dfrac{\lambda}{\lambda + N} & i = 0 \\ W_i^{(m)} = \dfrac{1}{2(\lambda + N)} & i = 1, \dots 2N \\ W_i^{(c)} = \dfrac{\lambda}{(\lambda + N)} + 1 - \alpha^2 + \beta & i = 0 \\ W_i^{(c)} = \dfrac{1}{2(\lambda + N)} & i = 1, \dots 2N \end{cases} \tag{2.27}$$

where: β is dependent on the knowledge of the distribution, a value of 2 is usually used for Gaussian distributions,
$W^{(m)}$ denotes the mean weights,
$W^{(c)}$ denotes the covariance weights.

The weighted mean and associated covariance are then given by:

$$\hat{y} = \sum_{i=0}^{2N} W_i^{(m)} y_i \tag{2.28}$$

$$P_y = \sum_{i=0}^{2N} W_i^{(c)} (y_i - \hat{y})(y_i - \hat{y})^T \tag{2.29}$$

The unscented Kalman filter applies the Unscented Transform to both the system and measurement model to compute the posterior mean and covariance matrices. These are then used with the standard Kalman filter equations to update the propagated states. For brevity, the individual steps are not repeated here as they can be found in many references (Julier and Uhlmann, 1997; Wan and Merwe, 2000; Brown and Hwang, 2012).

2.4. VDM Navigation

The use of a vehicle dynamic model (VDM) for navigation is not an entirely new concept. It dates back to the early 1990s (Koifman and Merhav, 1991). Research explores two main concepts in using a VDM for navigation, namely model-aided and model-based navigation. A model-aided approach employs an INS as the main process model and a VDM as an aiding tool. A model-based approach is the less common, more recent approach that uses a VDM as the main process model and an INS as the aiding system. Essentially, the two schemes use control inputs from either the autopilot system or manual flight commands to propagate a navigation solution using a set of equations that describe the motion of a vehicle under the influence of applied forces and moments. A VDM operates on dead reckoning principles, similar to an INS. If a VDM is operating alone, any errors in the initial estimate will lead to rapid drift in the navigation solution. Even with perfect knowledge of the vehicle dynamics, a VDM solution is still prone to numerical errors, which will accumulate, leading to drift in the navigation solution. The two schemes will be discussed in this section. The different VDM navigation schemes are discussed in chronological order, highlighting the base integration architecture, navigation performance results, and limitations of the adopted scheme.

Koifman and Bar-Itzhack (1999) present one of the early works using the aircraft dynamics model to aid an INS using an EKF. With perfectly known dynamics, position error for the aided INS was below 30 km during the entire flight (which lasted five hours), while for the pure INS case, the maximum error reached 1000 km. In this case, it was assumed that the main sources of error stem from the lack of knowledge of wind velocity and the IMU errors. To remedy this, the state vector included both wind velocity components and IMU errors. It was also indicated that without slalom-like manoeuvres with a period of 100 s and a roll amplitude of 15°, the filter diverges and renders the integrated navigation solution unusable. The manoeuvres enhanced the observability of the modes which would otherwise be unobservable in a straight and level flight. The integration approach considered both the VDM and the INS at the same level in a multi-process model approach. It included duplicate states in position, velocity and orientation, as shown in Figure 2.8. Additionally, the authors indicated using low-grade inertial sensors, but the presented error stochastics suggest high-end sensors.

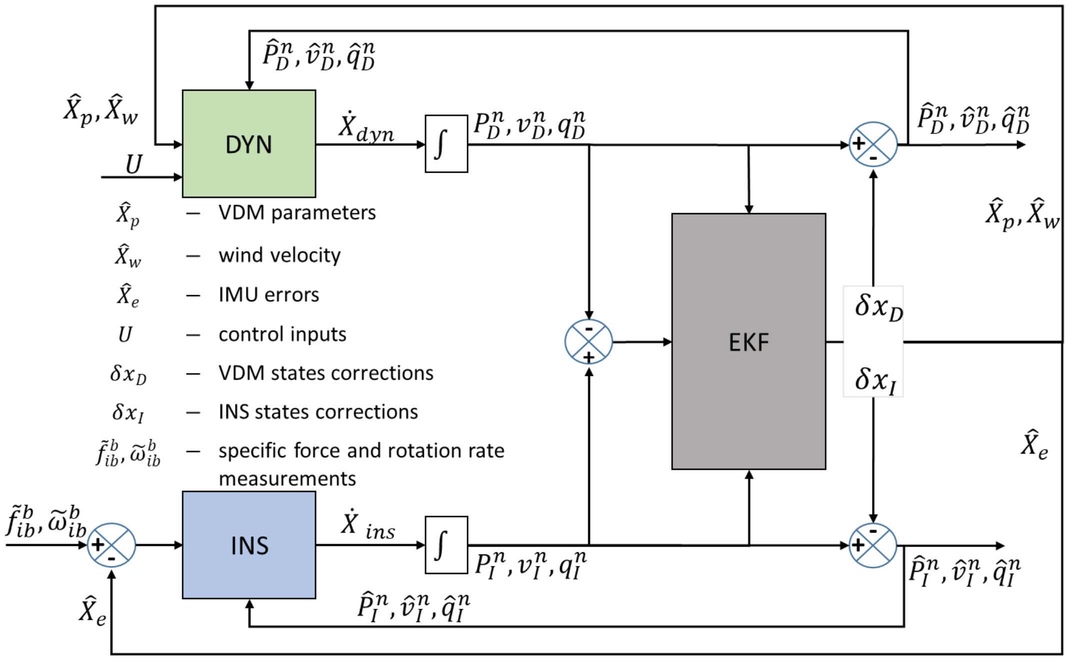


Figure 2.8. A multi-process model scheme. In the figure $P_{D,I}^n, v_{D,I}^n, q_{D,I}^n$ represent the position, velocity and quaternion states computed by the VDM (D) and INS (I), respectively; $\hat{P}_{D,I}^n, \hat{v}_{D,I}^n, \hat{q}_{D,I}^n$ represent the corrected navigation states.

A sensitivity analysis was performed by varying the VDM parameters from the nominal values, one at a time. It was found that for the whole flight profile, incorrect parameters in the order of 10% of the nominal values caused the navigation error to grow, making the aided system unusable. It was also found that the system's performance was more sensitive to errors in the aircraft lateral dynamics. To improve the navigation performance, the state vector was augmented to include 21 aerodynamic coefficients. This prevented divergence and improved the navigation accuracy of the aided navigation system with perturbed parameters. It was also suggested that not all parameters were estimated individually but rather as groups. The estimation was deemed good enough for navigation with the maximum position error during the entire flight being less than 30 km, similar to the aided INS case with perfectly-known dynamics. Further, the estimation of both wind velocity components and IMU errors was very similar to the case without any uncertainty in the model parameters. The estimation of wind velocity was possible even without an air data system.

Julier and Durrant-whyte (2003) investigated the role of vehicle process models in sensor-based navigation systems for autonomous land vehicles using an EKF. Using a high fidelity model of an automated ground vehicle implemented in the multibody dynamics and motion analysis software (ADAMS), the study showed that higher-order models suffer from observability problems in VDM parameters. However, it was shown that imposing weak constraints (treating a constraint as extra observation with a nonzero uncertainty) reduced the problem. The authors showed that the error between the true vehicle dynamics and the process model manifests itself in terms of a penalty that must be applied to the process noise covariance. The nature of this penalty was found to be time-varying. It was shown that some changes to the process model could reduce orientation errors by 90% and position errors by 40%. The authors focused on land vehicles using constraints in their process model, which are not directly applicable to fixed-wing aircraft. Further, the error characteristics of the sensors used were not clearly outlined. However, the general principles and deductions highlight the importance of a VDM in improved navigation performance.

Bryson and Sukkarieh (2004) investigated the use of a VDM in aiding position, velocity and orientation estimates provided by an INS with low-cost inertial sensors for a fixed-wing UAV using an EKF. Two approaches were considered in their investigation. The first approach is shown in Figure 2.9, and the second approach in Figure 2.10. The first approach compared and corrected velocity and attitude estimates as predicted by both the INS and VDM. The second approach used the VDM’s predicted acceleration and rotation rates to correct IMU errors directly. In the second approach, the Jacobian matrix for the VDM acceleration errors was evaluated numerically (the rate of change of the body axes acceleration was evaluated for varying values of the rotation rate perturbed about the value calculated by the VDM). In both configurations, the INS formed the main process model, and VDM aiding was activated during a GNSS outage. In the first configuration, the errors in the position states were not estimated in the filter due to the lack of coupling between them and other states (Bryson and Sukkarieh, 2004). It was argued that variations in atmospheric density with altitude or rotation of the navigation frame as a function of position resulting from operating over a large area of the Earth's surface might induce weak coupling leading to the observability of the position states.

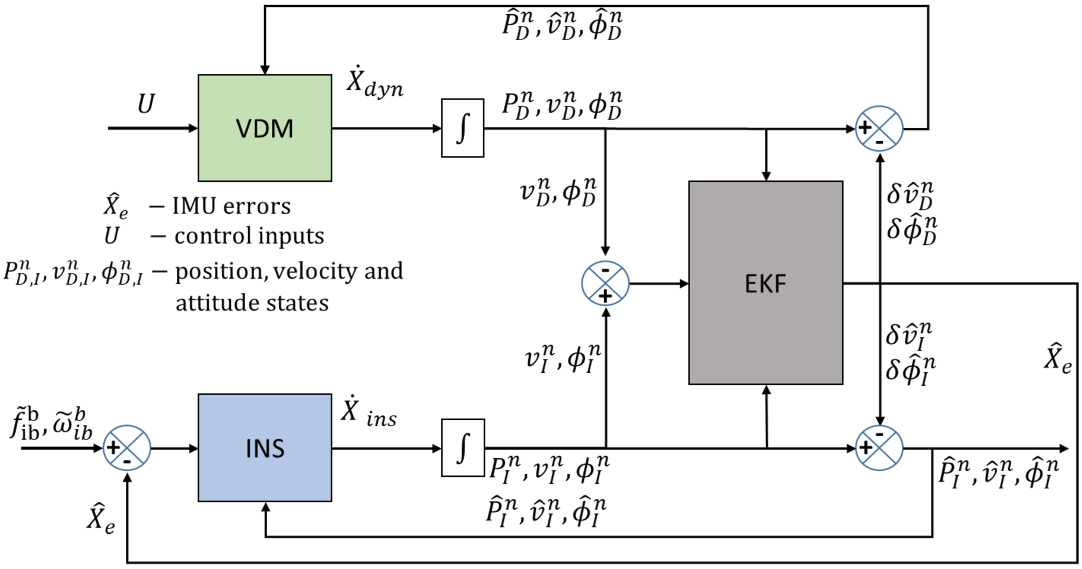


Figure 2.9. Configuration 1 of the model-aided architecture investigated by Bryson and Sukkarieh.

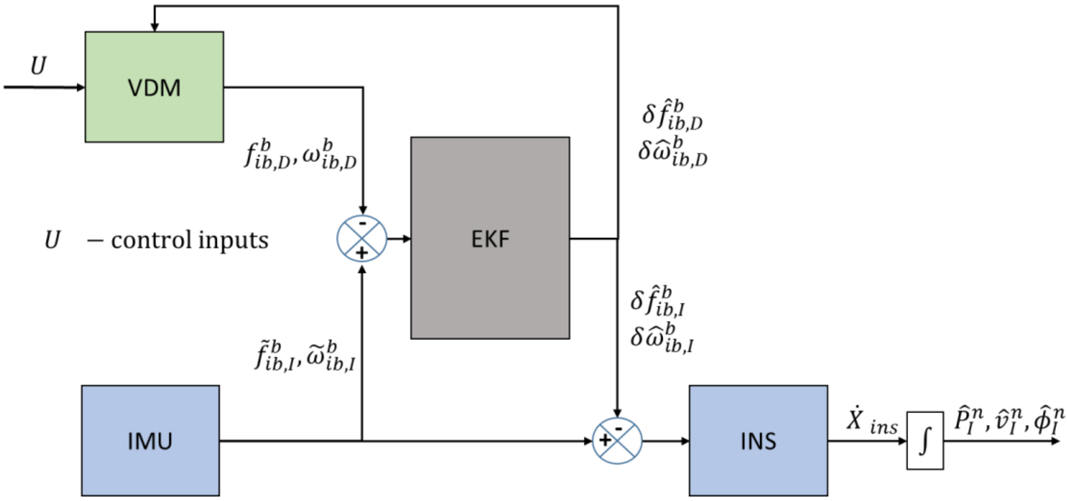


Figure 2.10. Configuration 2 of the model-aided architecture investigated by Bryson and Sukkarieh.

With a 5% uncertainty on VDM parameters, the east position error was below 100m for the first configuration and above 800m for the second configuration after 50 seconds of GNSS outage, indicating the first configuration's superior performance. The good performance in the first configuration was attributed to the marginal error growth in velocity and attitude that can be estimated and rejected with greater ease than the rapid error dynamics in acceleration and rotation rates. In both configurations, the mechanism to estimate wind velocity and VDM parameters was not included. The final navigation solution was still dependent on the INS, which would be disabled in case of IMU failure.

Vissière *et al.* (2008) reported a successful hovering flight of a Benzin Acrobatic model helicopter from Vario™ with low-cost inertial sensors by adding an accurate dynamics model, which improved the prediction of the EKF. CATIA was used to model 688 different parts to obtain the mass moment of inertia matrix and centre of gravity position (Vissière *et al.*, 2008). An autonomous outdoor flight under a 20km/h wind showed that position errors were within 1 m vertically and 3 m horizontally. The estimated attitude errors remained bounded to within 3 degrees in roll and pitch. However, the estimated yaw angle error was within 15 degrees. The architecture did not include a mechanism to estimate or calibrate the mass moment of inertia. Instead, CATIA was used for this purpose which can be time-consuming.

Dadkhah, Mettler and Gebre-egziabher (2008) investigated the use of a helicopter dynamic model to aid an attitude heading reference system (AHRS) incorporating low-cost rate gyros using an EKF. The helicopter model was developed using frequency-domain system identification using attitude and position information gathered using six high-speed MX-40 cameras (Dadkhah, Mettler and Gebre-egziabher, 2008). It was argued that parametric errors in the EKF measurement stream resulting from the helicopter dynamic model were the main cause of the suboptimal performance in the estimation of gyro biases. The authors argued that state vector augmentation to account for correlation of the model parameters could improve the solution. The online calibration of model parameters was not considered, even though the authors mentioned the potential benefits of such capability. Wind velocity components were neither estimated directly nor modelled as unknown external disturbances in the system design in which the final navigation solution was still dependent on an INS.

Vasconcelos *et al.* (2010) implemented an embedded INS with a VDM for a model helicopter using low-cost inertial sensors. The navigation performance of the embedded approach was compared to an external model-aided approach. The embedded approach used both error states and total states, as can be seen in Figure 2.11. In the embedded approach, the VDM was used to form the measurement innovation using INS states in the equations of motion. This reduced duplicate states and allowed choosing the appropriate dynamics (linear or rotational) to include. The execution time of the embedded VDM was 400 seconds, 26.3% lower than external VDM aiding using both angular and linear velocity inputs. With only linear velocity aiding, the execution time for the embedded scheme was 310 seconds, 42.9% lower than external VDM aiding. However, both external and embedded VDM aiding were computationally intensive as opposed to the classical INS/GNSS integration scheme. The navigation performance of the embedded VDM approach was similar to the external VDM approach. However, the use of both total and error states presents a complex integration approach.

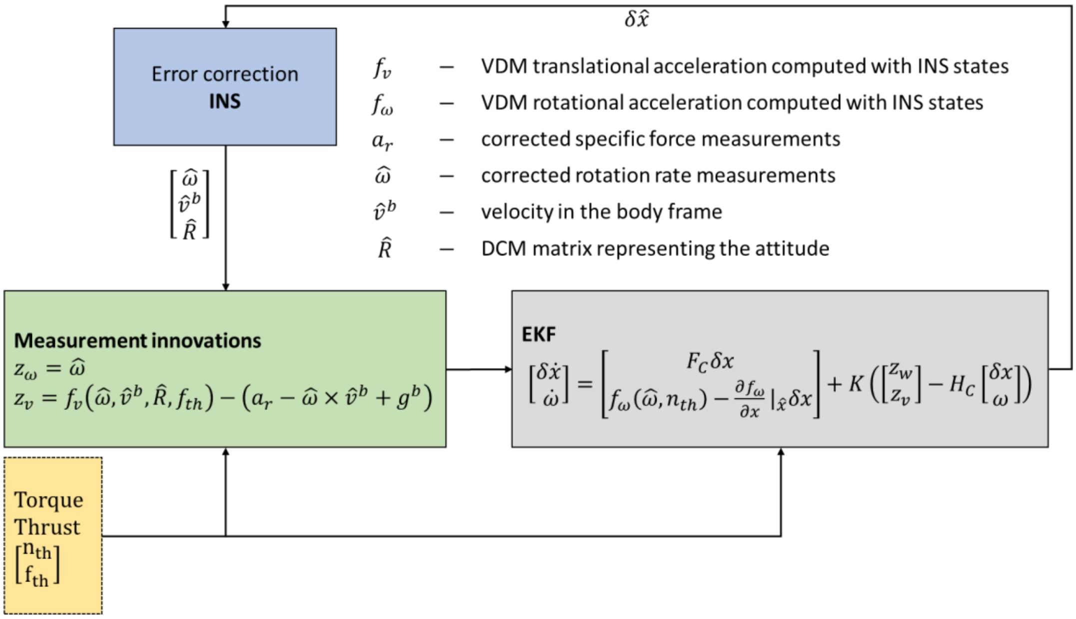


Figure 2.11. The embedded INS with VDM.

Crocoll *et al.* (2013) investigated a unified INS and VDM approach using a modified EKF. The unified approach incorporated two valid state predictions (using the INS and VDM), achieving a reduced state vector size and computational load over the classical model-aided scheme in Figure 2.8. It was shown analytically that the inclusion of position states as pseudo-measurements leads to the divergence of the navigation filter due to zero process noise. A higher update rate using the predicted velocity and orientation states improved the accuracy of the navigation solution. It was shown that the accuracy of the unified approach is similar to that presented by Koifman and Bar-Itzhack (1999). The architecture was then extended to include the capability for online VDM parameter calibration (Crocoll and Trommer, 2014) and the capability to estimate wind velocity states (Mueller, Crocoll and Trommer, 2016). Online parameter calibration and wind estimation significantly improved the navigation performance of the architecture. Further, it was shown that the quality of the IMU plays a vital role in wind velocity estimation during a GNSS outage, with a higher grade IMU showing improved wind estimation (Mueller, Crocoll and Trommer, 2016). However, the approach only considered the translational dynamic model and ignored the rotational model, and the architecture was only investigated for a quadrotor.

Sendobry (2014) completely avoids using duplicate states (position, velocity and attitude) by propagating the state vector using the VDM only as opposed to the unified scheme proposed by Crocoll *et al.* (2013). The state vector was augmented to include vehicle accelerations and moment biases in the EKF and applied to a quadrotor. The quadrotor propulsion model was parameterised through wind tunnel testing. Simulation results showed the importance of estimating the propulsion coefficients, which resulted in a consistent estimate of the navigation solution. An experimental investigation showed the robustness of the proposed solution using a ground vehicle. The position solution showed a drift-free navigation performance near buildings where the GNSS solution presented erroneous measurements and sometimes even total outage. The architecture, however, was not investigated during periods of extended GNSS outage and simulation studies were based on a quadrotor and did not consider a fixed-wing aircraft. The wind velocity components were not directly estimated in the filter even though moment biases due to asymmetric mass distribution and drag torque were estimated.

Khaghani and Skaloud (2016) presented an extension to the model-based approach presented by Sendobry (2014), with specific application to fixed-wing UAVs. Measurements from a GNSS receiver and a low-cost IMU were used to estimate corrections to the VDM solution using an EKF, as shown in Figure 2.12. With an initial uncertainty of 10% in the VDM parameters, simulation results indicated two orders of magnitude improvement in position estimation as opposed to conventional INS/GNSS integration for a GNSS outage lasting five minutes. Similarly, roll and pitch angle estimation improved by more than two orders of magnitude while yaw angle estimation improved by more than one order of magnitude as opposed to an INS/GNSS scheme. In further developments, experimental results indicated attitude errors of a VDM/GNSS integration scheme (IMU not used) being one to two orders of magnitude greater than a conventional INS/GNSS integration architecture (Khaghani and Skaloud, 2018b). This was mainly attributed to unmodeled dynamics, especially in lateral moments that failed to accurately track the vehicle movements in the absence of IMU data. The architecture included the mechanism to estimate wind velocity even without an air data system which reduced rapid growth in position error during a GNSS outage. The architecture used the position and velocity measurements from a GNSS receiver that are not available during a GNSS outage.

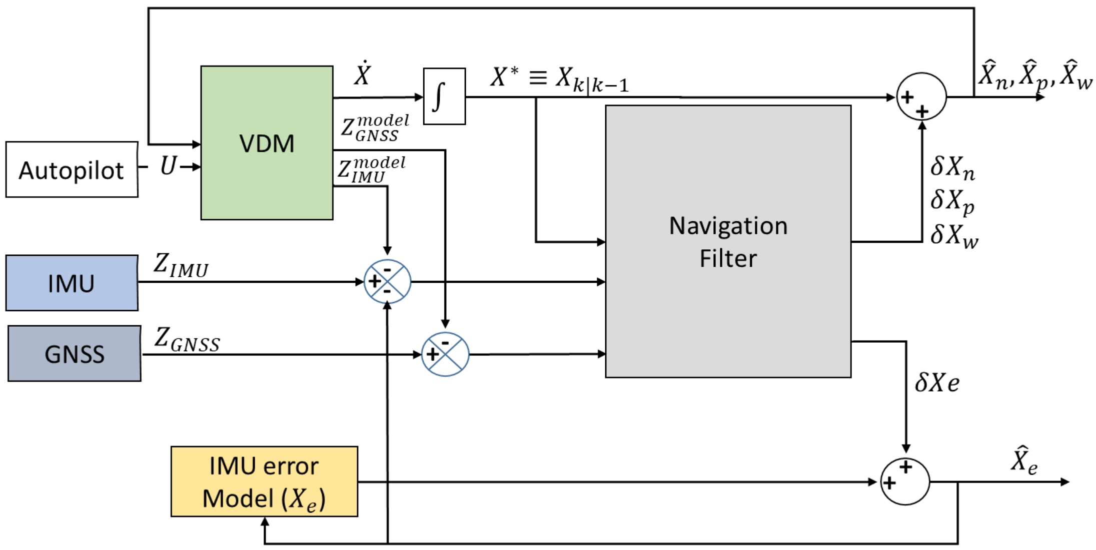


Figure 2.12. VDM integration architecture proposed by Khaghani and Skaloud.

Zahran *et al.* (2018) derived a VDM from a hybrid machine learning scheme utilising a bagged regression and classification technique to aid an INS during a GNSS outage for a quadcopter. The machine learning module (regression and classification) acted as a substitute to provide position and velocity information during periods of GNSS outage, as shown in Figure 2.13. An EKF with 21 states is used as the fusion filter. Using both regression and classification schemes resulted in lower position errors during GNSS outages as opposed to using only a regression scheme. The classification step reduced the navigation solution drift by using derived terms of the motors' speeds to classify the vehicle movement into specific modes (acceleration/deceleration, constant velocity and hover), which acted as velocity and attitude constraints. Compound manoeuvres not included in the training data seemed to degrade the performance during a GNSS outage. For an outage lasting over 100 s, the position error reached 16.8 m with compound manoeuvres as opposed to 5.5 m with single-axis manoeuvres, and both cases showed an order of magnitude improvement in position estimation as opposed to a pure INS case. The performance of the algorithm was investigated only through simulations for a quadcopter and did not include the mechanism to estimate wind velocity directly.

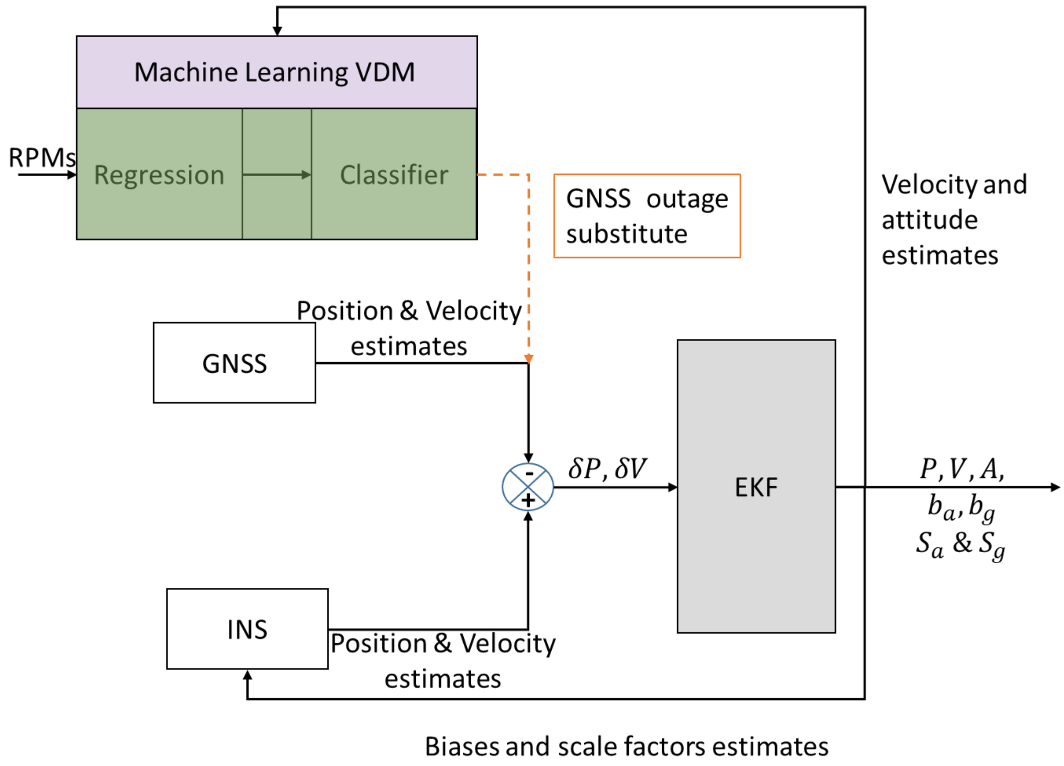


Figure 2.13. A hybrid machine learning VDM integration architecture.

Youn *et al.* (2020) proposed a model-aided state estimation scheme for a high-altitude long-endurance (HALE) fixed-wing solar-powered UAV developed by the Korea Aerospace Research Institute (KARI). The approach used synthetic measurements of angle of attack and sideslip angle derived using VDM parameters and accelerometer measurements with small-angle approximations. Angular rate measurements from gyros were used directly as measurements in the architecture. Specific force measurements from the accelerometers were used to derive synthetic measurements and propagate the airspeed, angle of attack, and sideslip angle states. Other sensors considered included a magnetometer for heading derivation, airspeed sensors and a GPS receiver. The approach used a UKF as the navigation filter. Results from real flight tests demonstrated that the architecture accurately estimated critical states, including wind velocity states, without direct measurements of angle of attack and sideslip angle. Even though the approach included the mechanism to estimate wind velocity components, the impact of the errors of VDM parameters on navigation performance was not investigated. Further, the impact of a GNSS outage on navigation performance was also not investigated.

2.5. Summary

This chapter has reviewed different technologies and systems used in forming an integrated navigation solution typically used in small UAVs. GNSS principles alongside strapdown inertial navigation system basics have been discussed. The errors that affect GNSS signals and inertial sensors have been independently discussed, and the complementary characteristics of the systems have been tabulated. This has served as justification for integrating the two systems. Further, different VDM navigation architectures have been discussed, identifying their strengths and limitations.

Two concepts in using the VDM in an integrated architecture have been discussed, namely model-aided and model-based schemes. It has been identified that VDM navigation schemes can mitigate rapid error growth in the navigation solution during GNSS outages following a clear mathematical structure representing the dynamics of the host platform type and an accurate set of model parameters. Some common limitations identified on the currently available integrated VDM architectures include the lack of a mechanism to estimate wind velocity and online parameter estimation. Further, the current VDM navigation schemes tend to use a loosely coupled configuration, using available position and velocity from a GNSS receiver in the fusion filter. This limits the performance of the integrated VDM scheme during an outage leading to drift in the navigation solution even with perfect knowledge of the vehicle dynamics. This limitation in the currently available integrated VDM architectures serves as the main motivation for the work carried out in this thesis.

3 Vehicle Dynamics Modelling

3.1. Introduction

In the previous chapter, different VDM navigation schemes in the archival literature were discussed. As previously highlighted, a VDM requires a clear mathematical structure for the host platform type and an accurate set of model parameters to provide useful information. This research aims to develop an integrated navigation architecture that utilises the dynamic model of a fixed-wing UAV. A six-degree-of-freedom (6DOF) model of a fixed-wing UAV is required to evaluate such an architecture in a simulation environment. The 6DOF model is essentially the physics or math model that defines the movement of an aircraft under the forces and moments applied to it using the control inputs as well as other external influences. For instance, the ability of a fixed-wing UAV to generate enough lift to overcome the Earth's gravitational force depends on the geometry of its lifting surfaces, the local atmosphere and the relative airflow around the UAV. A fully representative model will include the translational and rotational dynamics of the UAV, a local model of the atmosphere, a gravity model and any other external disturbances such as wind. Further, depending on the objectives of the simulation, the 6DOF model can include simple models valid for a limited flight regime or more complex models valid for a wider flight envelope. Once a 6DOF aircraft model is defined, motion variables can be derived from it and used to assess the performance of an integration architecture. Therefore, this chapter provides the details of the dynamics modelling effort, including the definition of the coordinate frames and different models used to develop a 6DOF aircraft model.

The chapter is organised as follows. Section 3.2 presents the coordinate frames used in this research. Section 3.3 presents the different ways of representing the relative orientation between coordinate frames (attitude). In Section 3.4, the rigid body equations of motion used in this research are presented. These are essentially the equations that will be used to obtain the motion variables from the forces and moments applied to the aircraft. In Section 3.5, the atmospheric model and gravity model used in this research are presented, and in Section 3.6, the wind model is presented. Section 3.7 presents the aerodynamic and propulsion models that characterise a fixed-wing UAV. The section also discusses some of the limitations of the models. Section 3.8 presents the 6DOF aircraft model implemented in a simulation environment and some details about the guidance and control scheme utilised. The Chapter summary is given in Section 3.9.

3.2. Coordinate Frames

Figure 3.1 shows the main coordinate frames used in this research. An inertial frame (not shown in the figure) is a non-rotating and non-accelerating frame with respect to the rest of the universe. For the purpose of navigation, an Earth-centred inertial frame is usually used. This frame is usually defined with its x-axis pointing from the Earth to the Sun at the vernal equinox (Groves, 2013). The z-axis points along the direction of the Earth's axis of rotation from the centre to the celestial north pole. The y-axis completes the 3D right-handed Cartesian system. This frame is denoted by the symbol i.

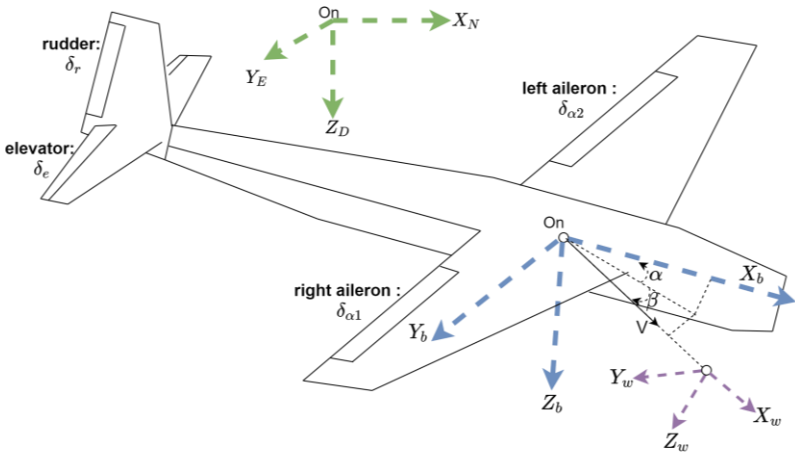


Figure 3.1. Coordinate frames, airspeed, and control surfaces.

A local navigation frame (X_N, Y_E, Z_D) also called a local level frame, is a local geodetic frame usually used as the resolving frame for the navigation solution. The origin of this frame is the point where the navigation solution is sought (centre of mass of the aircraft). Its z-axis points along the normal to the surface of the reference ellipsoid, as can be seen in Figure 3.2. The x-axis is the projection in the plane orthogonal to the z-axis of the line that points to the north pole, and the y-axis completes the orthogonal set (points east). This frame is usually abbreviated to the NED frame (north-east-down).

A body-fixed frame (X_b, Y_b, Z_b), denoted by the symbol b, has its origin at the centre of mass of the aircraft, and the angle of attack (α) and sideslip angle (β) are defined relative to it. Its x-axis usually points in the forward direction, the z-axis points down in the usual direction of gravity, and the y-axis completes the 3D right-handed Cartesian system. For angular motion, the x-axis, y-axis, and z-axis are usually referred to as the roll axis, pitch axis and yaw axis, respectively. The speed of the aircraft relative to the surrounding air is called airspeed, denoted V. This is usually aligned with the x-axis of the wind frame (X_w). The z-axis of the wind frame (Z_w) is taken along the lift line of action (but points in the opposite direction to the lift force) and the y-axis (Y_w) completes the right-handed orthogonal set.

An Earth-centred Earth-fixed frame (ECEF), denoted by the symbol e, has its origin at the centre of the ellipsoid modelling the Earth's surface and is used as the reference frame for the navigation solution. This frame remains fixed with respect to the Earth. This has its x-axis pointing from the centre to the intersection of the conventional zero meridian with the equator. The z-axis points along the Earth's axis of rotation to the true north pole, and the y-axis completes the orthogonal set, as shown in Figure 3.2. In this work, the WGS84 realisation of the ECEF frame has been used because it is the datum used by the GPS system.

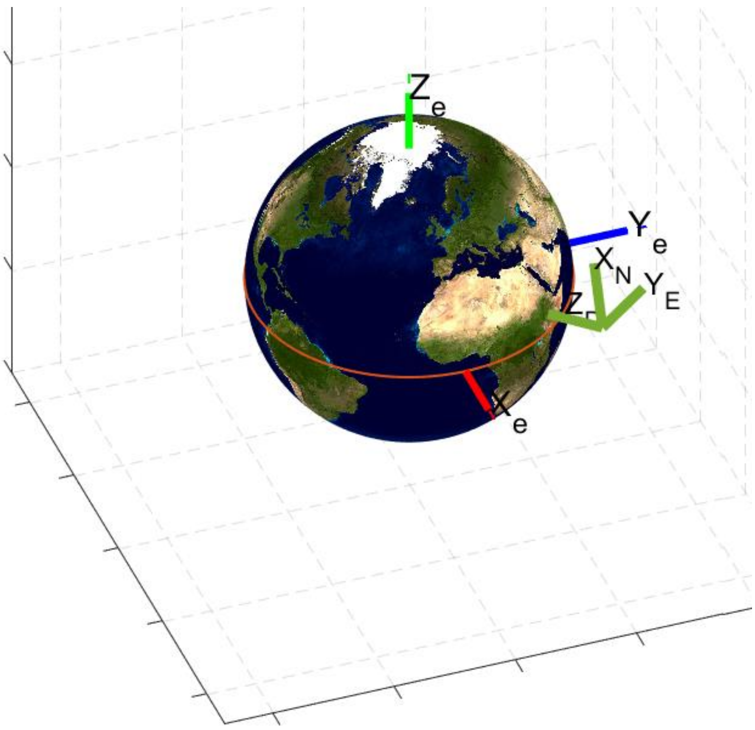


Figure 3.2. The ECEF frame and Local navigation (NED) coordinate frame.

The main parameters defining the WGS84 datum are (Dunn, 2018):

- Equatorial radius, $R_e = 6378137$ m
- Flattening of the ellipsoid, $f = 1/298.257223563$
- Eccentricity, $e = 0.0818191908426215$
- Earth's gravitational constant, $\mu_E = 3.9860050 \times 10^{14}$ m^3/s^2
- Earth's rotation rate, $\omega_{ie} = 7.2921151467 \times 10^{-5}$ rad/s

The airspeed magnitude, V, angle of attack, α, and sideslip angle, β, are given by:

$$V^b = \left[V_x^b, V_y^b, V_z^b\right]^T \tag{3.1}$$

$$V = \|V^b\| \tag{3.2}$$

$$\alpha = \arctan\left(\frac{V_z^b}{V_x^b}\right) \tag{3.3}$$

$$\beta = \arcsin\left(\frac{V_y^b}{V}\right) \tag{3.4}$$

Based on the airspeed vector and wind velocity vector, the aircraft's ground velocity vector in the NED frame is given by:

$$v^n = V^n + W^n \tag{3.5}$$

In Equation (3.5), W^n is the wind velocity vector in the NED frame.

3.3. Attitude Representation

The relative orientation between two coordinate frames is known as attitude. For aircraft navigation, the orientation of the body-fixed frame relative to the local navigation frame is usually of interest. Typically, an integrated architecture keeps track of the change in the relative orientation between the frames. The relative orientation between coordinate frames can be expressed in different ways. The most common representations include direction cosines, Euler angles (three successive scalar rotations), and quaternions (single rotation about a particular axis).

3.3.1. Direction Cosine Matrix

A direction cosine matrix is a 3x3 matrix, denoted by the symbol R_b^n representing the rotation from the body-fixed frame (b) to the navigation frame (n). The elements of the matrix are the product of the unit vectors describing the axes of the two frames. The rate of change of R_b^n is given by (Titterton and Weston, 2004):

$$\dot{R}_b^n = R_b^n \Omega_{nb}^b \tag{3.6}$$

where Ω_{nb}^b is the skew-symmetric form of the angular rate vector, $\omega_{nb}^b = [\omega_1, \omega_2, \omega_3]^T$, of the body frame with respect to the local navigation frame and resolved in the body frame. The skew-symmetric matrix is given by:

$$\Omega_{nb}^b = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3.7}$$

3.3.2. Euler Angles

Euler angles intuitively represent the orientation of one coordinate frame with respect to another using three successive rotations. Euler rotations do not commute because each rotation is performed in a different coordinate frame (the order of the rotations is critical). They exhibit a singularity when the pitch angle is $\pm$ 90° such that roll and yaw become indistinguishable. The rotations may be expressed as direction cosine matrices, and the most common rotation convention for aircraft navigation is the 3-2-1 convention visualised as a yaw rotation, then a pitch rotation and finally a roll rotation $(\psi \to \theta \to \phi)$. The corresponding rotation matrices are given by:

$$\begin{aligned} R_n^b &= R_1(\phi)R_2(\theta)R_3(\psi) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.8}$$

The propagation of Euler angles is given by:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = R_w \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tag{3.9}$$

$$R_w = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\,/\cos\theta & \cos\phi\,/\cos\theta \end{bmatrix}$$

3.3.3. Quaternions

A rotation from one coordinate frame to another may be completed by a single rotation about some axis. A quaternion is a four-element vector that may be used to represent rotation about this axis in three-dimensional space. A quaternion vector is given by:

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.10}$$

where the first element, q_0, is the scalar part which defines the magnitude of rotation and the other three elements, $q_1, q_2,$ and q_3, define the unit vector of the axis of rotation. A quaternion vector representing the rotation from the body-fixed frame to the navigation is denoted q_b^n. A lot has been written about quaternions, and in this work, only the relevant formulations are given. For a more comprehensive review of the subject, the reader is directed to the appropriate references (Kuipers, 1999; Titterton and Weston, 2004; Solà, 2017). The product of quaternions, denoted by the symbol ⊗, may be expressed in matrix form as:

$$\begin{aligned} p \otimes q = [p]_L \cdot q &= \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \\ = [q]_R \cdot p &= \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \end{aligned} \tag{3.11}$$

where $[p]_L$ and $[q]_R$ represent the left- and right-quaternion-product matrices (Solà, 2017). In relation to the navigation frame and the body-fixed frame, the propagation of a quaternion vector is given by:

$$\begin{gathered} \dot{q} = \frac{1}{2} q \otimes [\omega] \\ [\omega] = [0, \omega_1, \omega_2, \omega_3]^T \end{gathered} \tag{3.12}$$

The direction cosine matrix can be represented in quaternion form as:

$$R_b^n = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_0 q_3 + q_1 q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.13}$$

In this work, quaternions have been used to represent attitude because they do not have the singularity exhibited by Euler angles at $\pm$ 90° of pitch and are more computationally efficient than the direction cosine matrix. Table 3.1 summarises some of the properties of the rotation matrix (direction cosine matrix) and the quaternion.

Table 3.1. Properties of the rotation matrix and the quaternion (Solà, 2017).

	Rotation matrix R	Quaternion q
Parameters	$3x3 = 9$	$1 + 3 = 4$
Identity	$I_{3\times3}$	$[1,0,0,0]^T$
Inverse	R^T	q^*
Composition	R_1R_2	$q_1 \otimes q_2$
Rotation operator	$R = I + \sin\phi[\boldsymbol{u}]_\times + (1-\cos\phi)[\boldsymbol{u}]_\times^2$	$q = \cos\frac{\phi}{2} + \boldsymbol{u}\sin\frac{\phi}{2}$
Rotation action	Rx	$q\otimes x\otimes q^*$
Constraint	$RR^T = I$; $\det(R) = +1$	$q\otimes q^* = [1,0,0,0]^T$
ODE	$\dot{R} = R[\omega]_\times$	$\dot{q} = 1/2\ q\otimes\omega$
Exponential map	$R = \exp([\boldsymbol{u}\phi]_\times)$	$q = \exp(\boldsymbol{u}\phi/2)$
Logarithmic map	$\log(R) = [\boldsymbol{u}\phi]_\times$	$\log(q) = \boldsymbol{u}\phi/2$
Perturbations	$R(t) = R\exp([\boldsymbol{u}\Delta\phi]_\times)$	$q(t) = q\otimes\delta q$

where: $\boldsymbol{u}$ is the axis of rotation, ϕ is the magnitude of rotation,
$v = \boldsymbol{u}\phi$ is a rotation vector,
$2\delta\dot{q} = [0, \delta\dot{v}]^T$.

3.4. Equations of Motion

This section presents the equations of motion for a rigid body. These equations are used to generate the aircraft motion variables. The equations are generally platform-independent but will be presented in relation to a fixed-wing UAV since this is the platform used in this thesis. The inputs to these equations are the forces and moments applied to the platform. These are generated using the control inputs and also depend on the local atmosphere and the local airflow.

The main motion variables (navigation states) considered in modelling the dynamics of a fixed-wing UAV include:

$$X_n = \left[\mu, \lambda, h, v_{eb,N}^n, v_{eb,E}^n, v_{eb,D}^n, q_0, q_1, q_2, q_3, \omega_x, \omega_y, \omega_z, n\right]^T \tag{3.14}$$

where the geodetic position $p_b = [\mu, \lambda, h]^T$ represents the latitude, longitude and height of the UAV, respectively. The UAV's velocity vector $v_{eb}^n = \left[v_{eb,N}^n, v_{eb,E}^n, v_{eb,D}^n\right]^T$ is expressed with respect to the ECEF frame and resolved in the local navigation frame. The quaternion rotation vector $q_b^n = [q_0, q_1, q_2, q_3]^T$ represents a rotation from the body frame to the NED frame. The UAV's rotation rate vector around its axes $\omega_{ib}^b = \left[\omega_x, \omega_y, \omega_z\right]^T$ is expressed with respect to an inertial frame and resolved in the body frame, and n is the propeller rotation rate.

The equations of motion, alongside a first-order model for the propeller dynamics, used in this research are given by (Khaghani and Skaloud, 2016a):

$$\dot{p}_b = \left[\frac{v^n_{eb,N}}{R_M + h}, \frac{v^n_{eb,E}}{(R_P + h)\cos(\mu)}, -v^n_{eb,D}\right]^T \quad (3.15)$$

$$\dot{v}^n_{eb} = R^n_b f^b_{ib} + g^n - (2\Omega^n_{\mathrm{ie}} + \Omega^n_{en})v^n_{eb} \quad (3.16)$$

$$\begin{aligned} \dot{q}^n_b &= \tfrac{1}{2} q^n_b \otimes \left[\omega^b_{nb}\right] \\ &= \tfrac{1}{2}\left[\omega^b_{nb}\right]_R q^n_b \end{aligned} \quad (3.17)$$

$$\dot{\omega}^b_{ib} = (I^b)^{-1}\left(M - \omega^b_{ib} \times I^b \omega^b_{ib}\right) \quad (3.18)$$

$$\dot{n} = \frac{n_c}{\tau_n} - \frac{n}{\tau_n} \quad (3.19)$$

Equation (3.16) and Equation (3.18) define the translational and rotational dynamics of the UAV through the applied forces, f^b_{ib}, and moments, M. These can generally be derived using Newton's laws of motion (which apply with respect to inertial frames). For the interested reader, a brief discussion and derivation of the linear and rotational dynamics is included in Section D.1 and Section D.2 of Appendix D of this thesis.

In Equation (3.15), R_M and R_P represent the meridian radius of curvature and prime vertical radius of curvature, respectively.

$$R_M = \frac{R_e(1-e^2)}{(1-e^2\sin^2(\mu))^{3/2}} \quad (3.20)$$

$$R_P = \frac{R_e}{(1-e^2\sin^2(\mu))^{\frac{1}{2}}} \quad (3.21)$$

In Equation (3.16), the rotation matrix R^n_b transforms vectors from the body-fixed frame to the NED frame and f^b_{ib} is the specific force vector of the body frame with respect to an inertial frame resolved in the body frame. g^n is the gravity vector in the NED frame. In Equation (3.17), ω^b_{nb} is given by:

$$\omega^b_{nb} = \omega^b_{ib} - (R^n_b)^T(\omega^n_{ie} + \omega^n_{en}) \quad (3.22)$$

the transport rate, ω^n_{en}, and the Earth's rotation rate in the NED frame, ω^n_{ie}, are defined as (Groves, 2013):

$$\omega^n_{en} = [\dot{\lambda}\cos\mu \quad -\dot{\mu} \quad -\dot{\lambda}\sin\mu]^T \quad (3.23)$$

$$\omega^n_{ie} = [\omega_{ie}\cos\mu \quad 0 \quad -\omega_{ie}\sin\mu]^T \quad (3.24)$$

In Equation (3.18), I^b represents the mass moment of inertia matrix. This gives the body mass distribution around the origin. The mass moment of inertia matrix is given:

$$I^b = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.25}$$

where I_{xx}, I_{yy} and I_{zz} represent the moment of inertia terms about the aircraft's x-, y-, and z-axis, respectively. Conventional fixed-wing UAVs are usually symmetric about the x - z plane. As a result, the products of inertia, I_{xy} and I_{yz}, are assumed zero (Cork, 2014).

In Equation (3.19), n_c and τ_n represent the commanded propeller speed and time constant, respectively.

In Section 3.1, it was mentioned that the applied forces and moments acting on an aircraft also depend on the local atmosphere and local airflow. So, the next section will first review the atmospheric and gravity model used in this research, and the following section will give an overview of the wind model adopted.

3.5. Standard Atmosphere and Gravity Model

The international standard atmosphere may be used to represent the static atmosphere (pressure, temperature, and density of the Earth's atmosphere) void of any dynamic effects such as wind and turbulence.

In this research, the International Civil Aviation Organization (ICAO) standard atmosphere is used to model the static atmosphere. It consists of a tabulation of values at various altitudes. In the standard atmosphere, the temperature within different layers of the atmosphere is taken as a linear function of the altitude above mean sea level (ICAO, 1993). The temperature, pressure and associated temperature gradient (lapse rate) for some layers of the atmosphere are presented in Table 3.2. The standard atmosphere exhibits both spatial and temporal correlations, and therefore local variations in pressure, temperature and density are not reflected in the model. However, this does not present a problem in assessing the navigation performance of an integration architecture.

Table 3.2. The altitude, temperature, pressure and lapse rate within the troposphere and stratosphere.

Layer	h_0 [m]	T_0 [K]	P_0 [Pa]	a [°C/m]
Troposphere	0	288.15	101,325	-0.0065
Tropopause	11,000	216.65	22,632	0.0
Stratosphere	20,000	216.65	5,475	0.0010
Stratosphere	32,000	228.65	868	0.0028

where: h_0 is the basepoint height above mean sea level for the specific layer,
T_0 is the basepoint temperature (sea level for the troposphere),
P_0 is the basepoint pressure (sea level for the troposphere),
a is the temperature lapse rate.

The atmospheric data presented in Table 3.2 allows the calculation of the local air temperature and density given by:

$$T = T_0\left[1 + a\frac{h_{amsl}-h_0}{T_0}\right] \tag{3.26}$$

$$\rho = \frac{P_0\left[1+a\frac{h_{amsl}-h_0}{T_0}\right]^{5.2561}}{R_a T} \tag{3.27}$$

where: R_a is the gas constant for air (J/kg · K),
h_{amsl} is the height above mean sea level (m).

The approximate height above mean sea level h_{amsl} is related to the geodetic height h by:

$$h_{amsl} = h - N(\mu, \lambda) \tag{3.28}$$

where N is the height of the geoid (constant gravity potential model of the Earth's surface) relative to the ellipsoid. As depicted in Figure 3.3, Equation (3.28) is only an approximation.

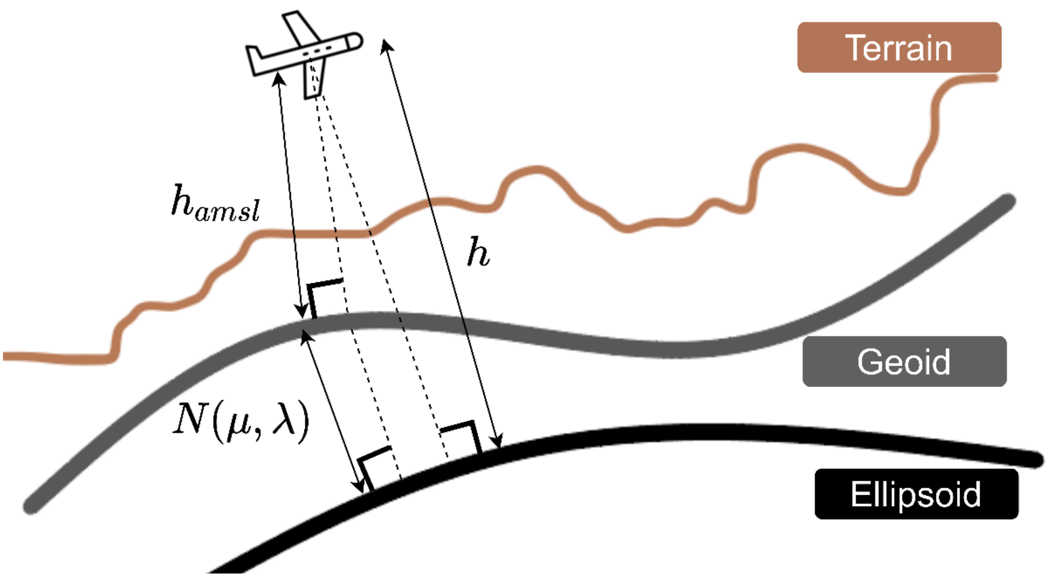


Figure 3.3. Orthometric h_{amsl} and Geodetic height h.

The WGS84 datum provides a simple representation of the acceleration due to gravity at the surface of the reference ellipsoid given as a function of the latitude at a particular location. The model is referred to as the Somigliana model (Groves, 2013). Other higher precision models exist, such as the EGM 2008 and EGM 96. These contain a large set of coefficients used to compute a much higher precision of the acceleration due to gravity. These are computationally intensive, especially for low-cost applications involving UAVs that operate relatively close to Earth's surface. The Somigliana gravity model is given by:

$$g_0(\mu) = g_e\frac{(1 + 0.001931853\sin^2\mu)}{(1 - e^2\sin^2\mu)^{1/2}} \tag{3.29}$$

where: g_e is the theoretical gravity at the equator,
e is the first eccentricity of the ellipsoid,
μ is the geodetic latitude.

For heights less than 10 km, the down component of the gravity model can be approximated as:

$$g_{b,D}^{n}(\mu, h_b) = g_0(\mu)\left[1 - \frac{2}{R_0}\left(1 + f + \frac{\omega_{ie}^2 R_0^2 R_p}{\mu_E}\right)h_b + \frac{3}{R_0^2}h_b^2\right] \qquad (3.30)$$

where: μ_E is the Earth's gravitational constant,
R_0 is the equatorial Earth radius,
R_p is the polar Earth radius,
f is the flattening of the ellipsoid.

3.6. Atmospheric Disturbances

To introduce unknown disturbances in the 6DOF aircraft model, an engineering model of atmospheric disturbance is required. Various models exist that capture the complex interaction of wind shear, vector shear, turbulence and gusts. The most common models used in aircraft navigation are detailed in the military specification document MIL-F-8785C (Moorhouse and Woodcock, 1982). The use of different models is driven by the objectives of the simulation, such as the study and evaluation of stability and control characteristics of an aircraft (flying qualities). An empirical scale is also used to describe wind speed in relation to observed conditions at sea or on land. This scale is referred to as the Beaufort scale and is presented in Table 3.3. It provides a convenient interpretation of wind speed.

Table 3.3. Beaufort wind scale.

Beaufort Scale	Mean wind [m/s]	Wind Limits [m/s]	Description
0	0.5	0.0 − 1.0	Calm
1	1.5	1.0 − 2.0	Light air
2	2.5	2.0 − 3.0	Light breeze
3	4.5	4.0 − 5.0	Gentle breeze
4	7.0	6.0 − 8.0	Moderate breeze
5	10.0	9.0 − 11.0	Fresh breeze
6	12.5	11.0 − 14.0	Strong breeze
7	15.5	14.0 − 17.0	Near gale
8	19.0	17.0 − 21.0	Gale
9	22.5	21.0 − 24.0	Severe gale
10	26.5	25.0 − 28.0	Storm
11	30.5	29.0 − 32.0	Violent storm
12	−	33.0 +	Hurricane

To study the navigation performance of different integration techniques, different authors have proposed and used simpler models. For instance, Koifman and Bar-

Itzhack (1999) used a two-component wind model with a mean wind speed and a gust component described by a first-order Markov process.

In this research, a stochastic wind model given by a Gauss-Markov process is used to assess aircraft navigation performance. The wind velocity model has a constant component of a southeasterly wind of 3.8 m/s magnitude, a correlation time of 200 seconds, and a process uncertainty of 0.1 m/s. On the Beaufort scale, this would be described as a gentle breeze.

3.7. Aerodynamics and Propulsion

This section presents the aerodynamic and propulsion models for a conventional fixed-wing UAV. These models are used to provide the applied forces and moments on the UAV during a flight through the combined effect of the control inputs, local atmosphere and local airflow.

3.7.1. Mass, Inertia and Geometry

A UAV's mass, inertia and geometrical characteristics are important parameters in the aerodynamic model. They play a key role in the translational and rotational dynamics of the aircraft. For a small UAV, its mass and geometry can easily be derived using measurements. The mass moment of inertia, however, can be difficult to derive with simple measurements. The mass moment of inertia of a small UAV is usually derived from a CAD (computer-aided design) package or experimentally from full-scale oscillation tests.

In this research, the aircraft is modelled as a rigid body. It is assumed that the mass of the aircraft does not change. This is also the case for the mass moment of inertia matrix. The aircraft's geometry is also assumed to be fixed. This greatly simplifies the translational and rotational dynamics. The mass, inertia and geometrical properties of the UAV modelled in this research are presented in Table 3.4. The reference values can also be found in Ducard (2007).

Table 3.4. Aircraft mass and inertia data.

Parameter		Value	Units
Mass	m	28	kg
Roll moment of inertia	I_{xx}	2.56	kg m^2
Pitch moment of inertia	I_{yy}	10.9	kg m^2
Yaw moment of inertia	I_{zz}	11.3	kg m^2
Product of inertia	I_{xz}	0.5	kg m^2
Product of inertia	I_{xy}	0.0	kg m^2
Product of inertia	I_{yz}	0.0	kg m^2
Wingspan	b	3.1	m
Wing area	S	1.8	m^2
Mean aerodynamic chord	$\bar{c}$	0.58	m
Propeller diameter	D	0.79	m

3.7.2. Thrust

In most small fixed-wing UAVs, a propeller generates the required thrust force. The propeller is usually installed on a brushless DC motor connected to a power source (usually a battery) via a motor controller. The thrust force generated by a propeller is given by:

$$F_T = \rho n^2 D^4 CF_T(J) \tag{3.31}$$

where the thrust coefficient CF_T depends on the advance ratio and modelled according to Khaghani and Skaloud (2016) and Ducard (2007).

$$CF_T(J) = CF_{T_1} + CF_{T_2} J + CF_{T_3} J^2$$
$$J = \frac{V}{D\pi n} \tag{3.32}$$

The model above ignores the variation of the thrust coefficient with propeller rotational speed (Brandt and Selig, 2011). Typically, look-up tables are more suited for characterising thrust coefficient with advance ratio and propeller rotational speed. Brandt and Selig (2011) showed that the variation of the coefficients is more significant around the high-efficiency region for propellers typically used in small UAVs. For simplicity, the model also assumes that the thrust vector is aligned with the aircraft's longitudinal axis, even though it is possible to have other components of the thrust force vector that can be resolved into the body frame.

3.7.3. Lift Force

The lift force coefficient is modelled as a linear function of the lift curve slope (CF_{Z_α}) and the lift coefficient at zero angle of attack (CF_{Z1}). For the purpose of navigation and operating within a limited range of angles of attack, this simple model is sufficient (Khaghani and Skaloud, 2016a).

$$F_Z^w = \bar{q} S CF_Z(\alpha) \tag{3.33}$$
$$CF_Z = CF_{Z1} + CF_{Z_\alpha}\alpha$$
$$\bar{q} = \frac{1}{2}\rho V^2$$

It is not difficult to see that the lift equation above is only valid for a limited range of angles of attack since it does not take into account nonlinear effects such as stall and unsteady aerodynamics. For instance, a fixed-wing UAV with a sharp leading edge and low aspect ratio wing design might experience trailing-edge separation at high angles of attack and sometimes even leading-edge vortex breakdown. Thin wings with sharp leading edges, such as delta wings, might cause leading-edge vortices to form that travel down the wing and break down due to turbulence. The breakdown of these vortices leads to a loss in vortex lift (Khan and Nahon, 2016).

3.7.4. Lateral Force

Lateral force coefficient is modelled as a function of sideslip angle β and is given by:

$$F_Y^w = \bar{q}SCF_Y(\beta)$$
$$CF_Y = CF_{Y1}\beta \tag{3.34}$$

This model assumes that the aircraft is symmetric about the plane formed between the body x-axis and z-axis.

3.7.5. Drag Force

Drag is the force that resists the movement of an aircraft through air. There are generally two contributions to the drag force acting on an aircraft: parasite drag and lift induced drag. Parasite drag includes form drag (drag due to the shape of the aircraft), interference drag, and skin friction. The lift induced drag results from the downwash created by wingtip vortices. The wingtip vortices are created from the spanwise flow of air from the lower surface (high pressure) to the upper lower-pressure surface. To capture both parasitic and lift induced drag, the drag coefficient is modelled as a function of both the angle of attack, α, and the sideslip angle, β, as found in Ducard (2007).

$$F_X^w = \bar{q}SCF_X(\alpha, \beta)$$
$$CF_X = CF_{X1} + CF_{X\alpha}\alpha + CF_{X\alpha 2}\alpha^2 + CF_{X\beta 2}\beta^2 \tag{3.35}$$

Simple models exist that model the drag coefficient as a linear function of the angle of attack. However, these models incorrectly predict the drag force when the angle of attack becomes sufficiently negative (Beard and McLain, 2013).

3.7.6. Roll Moment

The dimensionless roll moment coefficient is modelled as a linear function of aileron deflection, dimensionless roll rate ($\bar{\omega}_x$), dimensionless yaw rate ($\bar{\omega}_z$) and sideslip angle (Ducard, 2007).

$$M_X^b = \bar{q}SbCM_X(\,\delta_\alpha, \bar{\omega}_x, \bar{\omega}_z, \beta) \tag{3.36}$$
$$CM_X = CM_{X_{\delta_\alpha}}\delta_\alpha + CM_{X_{\bar{\omega}_x}}\,\bar{\omega}_x + CM_{X_{\bar{\omega}_z}}\,\bar{\omega}_z + CM_{X_\beta}\beta$$

The term $CM_{X_{\delta_\alpha}}$ is associated with the deflection of the ailerons and is referred to as the primary roll control derivative. The term $CM_{X_{\bar{\omega}_x}}$ is roll damping derivative.

3.7.7. Pitch Moment

The dimensionless pitch moment coefficient is modelled as a linear function of the elevator deflection, the dimensionless pitch rate ($\bar{\omega}_y$), the dimensionless yaw rate ($\bar{\omega}_z$), the angle of attack and the pitching moment coefficient at zero angle of attack (Ducard, 2007; Khaghani and Skaloud, 2016a).

$$M_Y^b = \bar{q}S\bar{c}CM_Y\big(\,\delta_e, \bar{\omega}_y, \alpha\big) \tag{3.37}$$

$$CM_Y = CM_{Y_{\delta_e}} \delta_e + CM_{Y_{\bar{\omega}_y}} \bar{\omega}_y + CM_{Y_\alpha} \alpha + CM_{Y1}$$

The term $CM_{Y_{\delta_e}}$ is associated with the deflection of the elevator and is referred to as the primary pitch control derivative. The term CM_{Y_α} is the longitudinal static stability derivative and $CM_{Y_{\bar{\omega}_y}}$ is the pitch damping derivative.

3.7.8. Yaw Moment

The yawing moment coefficient is modelled as a linear function of rudder deflection, dimensionless yaw rate ($\bar{\omega}_z$), and sideslip angle (Ducard, 2007; Khaghani and Skaloud, 2016a).

$$M_Z^b = \bar{q}SbCM_Z(\delta_r, \bar{\omega}_z, \beta) \tag{3.38}$$

$$CM_Z = CM_{Z_{\delta_r}} \delta_r + CM_{Z_{\bar{\omega}_z}} \bar{\omega}_z + CM_{Z_\beta} \beta$$

The term $CM_{Z_{\delta_r}}$ is associated with the deflection of the rudder and is referred to as the primary yaw control derivative. The term $CM_{Z_{\bar{\omega}_z}}$ is the yaw damping derivative and CM_{Z_β} is the weathercock stability derivative.

In Equation (3.36) - (3.38), the dimensionless roll rate, pitch rate and yaw rate are given by:

$$\bar{\omega}_x = \frac{\omega_x b}{2V}, \bar{\omega}_y = \frac{\omega_y \bar{c}}{2V}, \bar{\omega}_z = \frac{\omega_z b}{2V} \tag{3.39}$$

The control surface deflections are normalised such that the range is the same.

$$\delta_a, \delta_e, \delta_r \in [-1, 1] \tag{3.40}$$

Table 3.5 shows the control surface sign convention used in this research.

Table 3.5. Control surface sign convention.

Control	Deflection	Sense	Primary effect
δ_a	Right aileron up	+	Positive roll moment
δ_e	Elevator up	+	Positive pitch moment
δ_r	Rudder right	+	Positive yaw moment

3.7.9. Propeller Torque

Typically in a small fixed-wing UAV, a motor spins the propeller to generate thrust. This results in an equal and opposite torque applied by the propeller to the body on which the motor-propeller assembly is mounted. This is given by:

$$M_p = CM_p \rho n^2 D^5 \tag{3.41}$$

where CM_p is the torque coefficient. It is generally a function of the Reynolds' number, tip Mach number, propeller design and the advance ratio. Knowledge of the torque load can also be used to estimate the power required to drive the propeller. For a fixed-wing UAV, the effects of the propeller torque are usually relatively minor. In this thesis, the propeller torque is ignored alongside other gyroscopic effects.

The specific force vector can now be represented by:

$$f_{ib}^{b} = \frac{1}{m}\left(R_{w}^{b} \begin{bmatrix} F_X^w \\ F_Y^w \\ F_Z^w \end{bmatrix} + \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix} \right) \tag{3.42}$$

and the moment term can be represented by:

$$M = \left[M_X^b, M_Y^b, M_Z^b \right]^T \tag{3.43}$$

In Equation (3.42), R_w^b is the vector transformation matrix from the wind frame to the body frame and is given by:

$$R_w^b = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \cos\beta\sin\alpha & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{3.44}$$

In this thesis, the reference values for the presented aerodynamic and propulsion model are obtained from Ducard (2007). The values are also presented in Table 3.6.

Table 3.6. Reference values for the aerodynamic and propulsion models presented.

Property	Value	Units	Property	Value	Units
CF_{T_1}	$8.42 \cdot 10^{-2}$	[−]	CM_{X_β}	$-1.30 \cdot 10^{-2}$	[rad^{-1}]
CF_{T_2}	$-1.36 \cdot 10^{-1}$	[−]	$CM_{X_{\bar{\omega}_x}}$	$-1.92 \cdot 10^{-1}$	[−]
CF_{T_3}	$-9.28 \cdot 10^{-1}$	[−]	$CM_{X_{\bar{\omega}_z}}$	$3.61 \cdot 10^{-2}$	[−]
CF_{X_1}	$-2.12 \cdot 10^{-2}$	[−]	CM_{Y1}	$2.08 \cdot 10^{-2}$	[−]
CF_{X_α}	$-2.66 \cdot 10^{-2}$	[rad^{-1}]	CM_{Y_α}	$-9.03 \cdot 10^{-2}$	[rad^{-1}]
$CF_{X_{\alpha 2}}$	-1.55	[rad^{-2}]	$CM_{Y_{\delta_e}}$	$5.45 \cdot 10^{-1}$	[−]
$CF_{X_{\beta 2}}$	$-4.01 \cdot 10^{-1}$	[rad^{-2}]	$CM_{Y_{\bar{\omega}_y}}$	-9.83	[−]
CF_{Z1}	$1.29 \cdot 10^{-2}$	[−]	$CM_{Z_{\delta_r}}$	$5.34 \cdot 10^{-2}$	[−]
CF_{Z_α}	-3.25	[rad^{-1}]	$CM_{Z_{\bar{\omega}_z}}$	$-2.14 \cdot 10^{-1}$	[−]
CF_{Y1}	$-3.79 \cdot 10^{-1}$	[rad^{-1}]	CM_{Z_β}	$8.67 \cdot 10^{-2}$	[rad^{-1}]
$CM_{X_{\delta_\alpha}}$	$6.79 \cdot 10^{-2}$	[−]	τ_n	0.4	s

3.8. Matlab/Simulink Implementation

This section presents a high-level overview of the blocks defining the 6DOF aircraft model based on the presented rigid body equations of motion alongside the aerodynamic, propulsion, gravity and atmospheric models. The 6DOF aircraft model is implemented in Matlab/Simulink, as can be seen in Figure 3.4. Matlab provides a seamless environment to develop and test algorithms. The Matlab-based graphical programming language (Simulink) provides a customisable set of block libraries ideal for model-based design, testing and even code generation to other programming languages. The 6DOF aircraft model is used to generate IMU and GNSS datasets to evaluate the navigation performance of VDM integration architectures, as discussed in the following chapters.

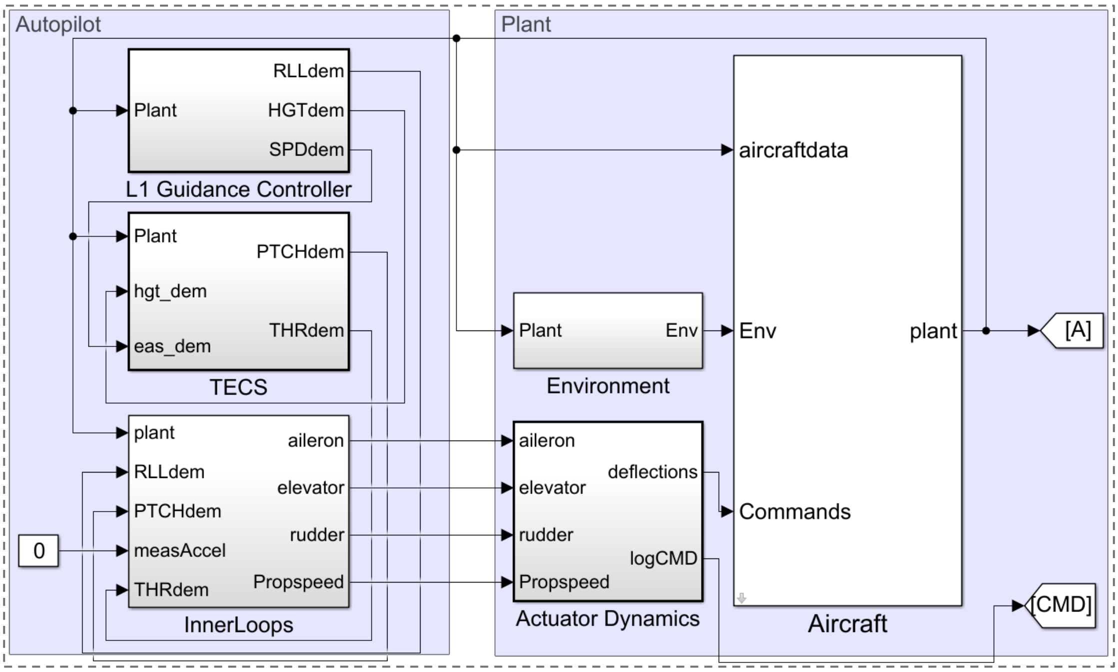


Figure 3.4. Simulink implementation of the VDM.

The main implementation of the presented equations of motion is contained within the aircraft block. The block contains the aerodynamic model and propulsion model used to output the forces and moments acting on the aircraft, as shown in Figure 3.5. These are then used to generate the motion variables and the data used for the sensor models through the equations of motion presented in Section 3.4 of this chapter. The implementation of the atmospheric and gravity model in the simulation environment is contained within the environment block. The block also contains a simple wind model previously explained in Section 3.6.

A simple mission planning interface has also been implemented in Matlab to allow a user to quickly define a set of waypoints alongside autopilot settings at each waypoint. This mission planning interface is shown in Figure 3.6 alongside a small dialog box used to define the autopilot settings for each waypoint. Some of these settings include the altitude of the waypoint, the waypoint radius, the command at each waypoint, the aircraft speed at a particular waypoint and the number of turns if the aircraft is commanded to loiter at a particular waypoint. The interface can also be used to load waypoints defined in a text file and can also be used to replay the flight. After defining the waypoints, the simulation is started, and the data is recorded into a structure for further processing.

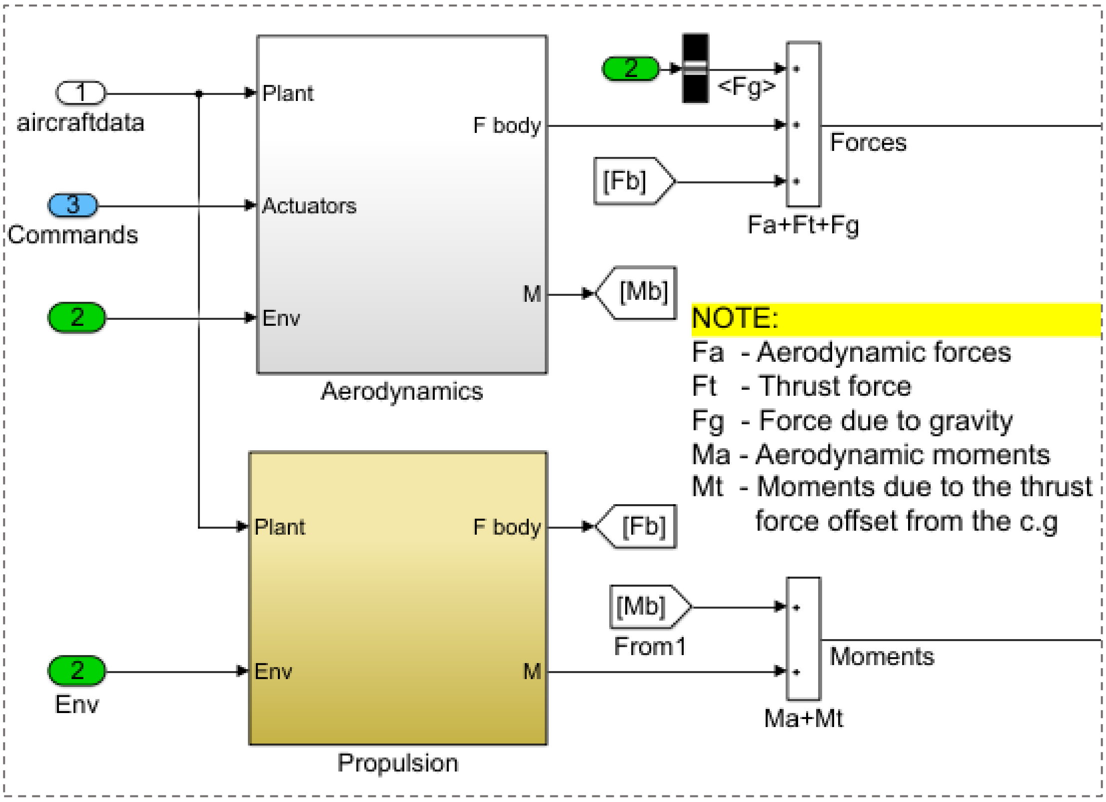


Figure 3.5. Aerodynamics and propulsion block.

The autopilot system consists of a nonlinear guidance logic (L1 guidance controller), a throttle and pitch controller (Total Energy Control System - TECS controller) and inner loops to control roll, pitch and yaw angle demands. The nonlinear guidance logic generates a lateral acceleration command based on the cross-track error. The acceleration command is then used to generate a bank angle command that returns the aircraft to the desired track. The TECS controller decouples the dynamic response of altitude and airspeed, enabling their efficient control using the combined effect of throttle and elevator inputs. In the implementation, it is assumed that the throttle controls the total energy of the aircraft, and the elevator controls the energy distribution (Balmer, 2015). Further details of the controllers are given in Appendix A.

The autopilot system is manually tuned using nominal VDM parameters (Table 3.6) to achieve desired tracking performance. Tracking performance is then tested against random changes in the VDM parameters reaching 10% of the initial values for a simple trajectory, and the results are presented in Figure 3.7. The final 3D position error does not exceed 60 metres for the presented flight segment.

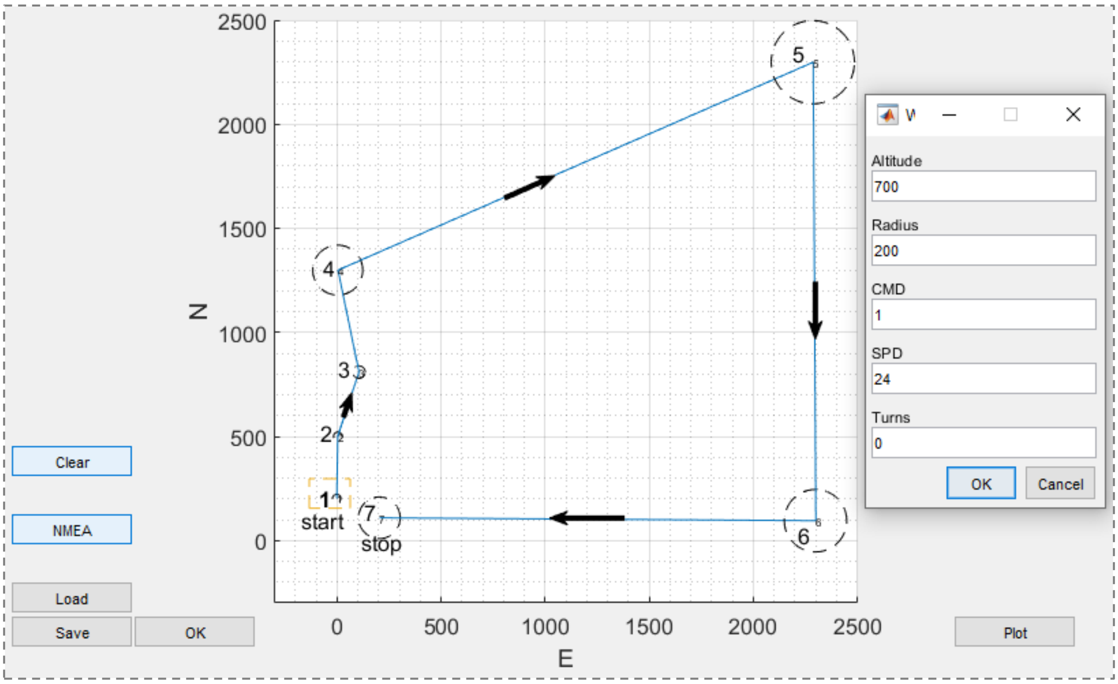


Figure 3.6. A simple mission planning interface. The dashed circles represent the waypoints, and the arrows represent the direction of the flight. The radius parameter determines the size of the circle.

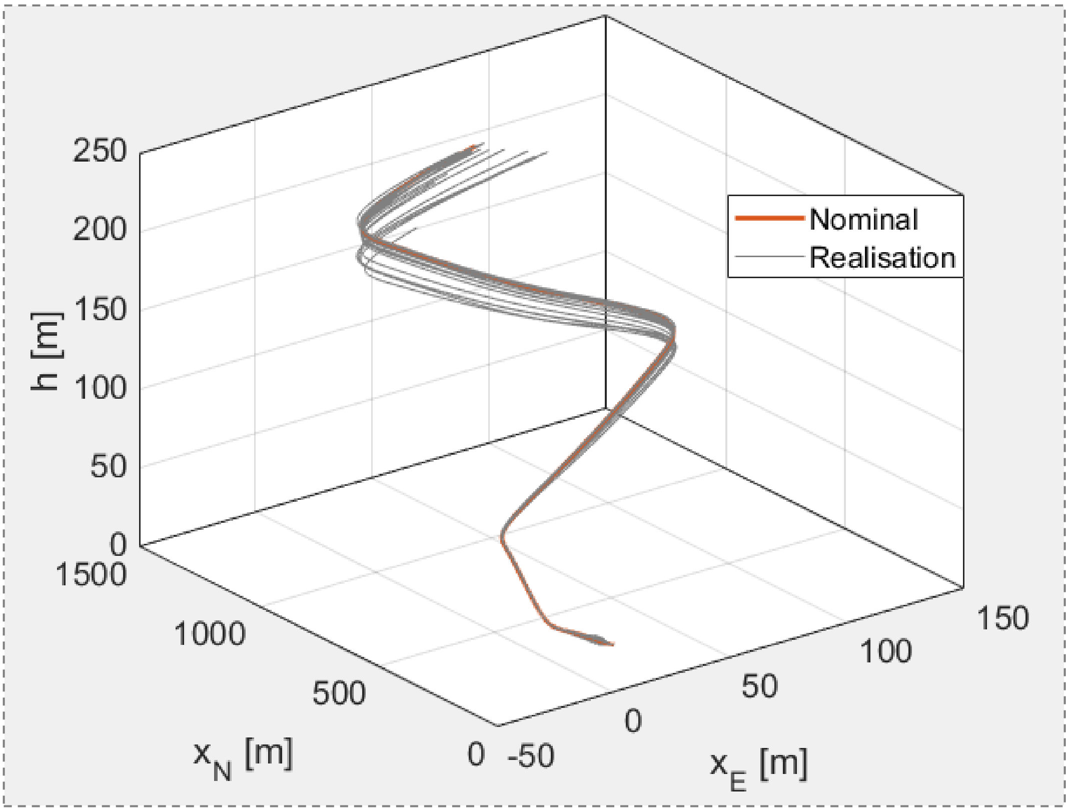


Figure 3.7. A simple trajectory generated by flying the aircraft between waypoints. The grey lines represent the trajectory with random variations (10%) in VDM parameters.

3.9. Summary

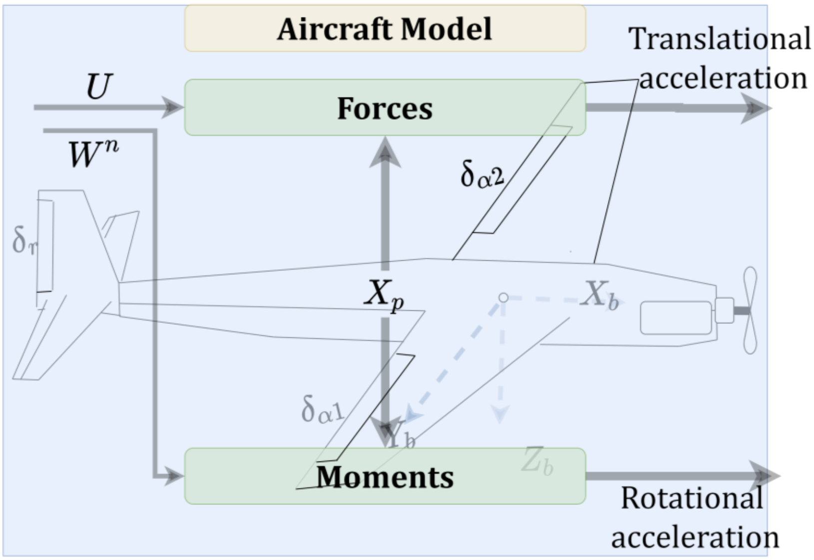
This chapter has presented the modelling of the aircraft dynamics used to develop a six-degree-of-freedom (6DOF) model of a fixed-wing UAV. The chapter highlighted the coordinate frames used and the set of equations that govern the motion of the aircraft under the influence of applied forces and moments. The chapter then presented the atmospheric and gravity models used in the 6DOF aircraft model. An aerodynamic model for a conventional fixed-wing UAV was presented, highlighting some of its limitations. For instance, the lift model adopted is only valid for a limited range of angles of attack and ignores other effects such as stall. The chapter then presented a high-level overview of the 6DOF aircraft model implemented in Matlab/Simulink, highlighting the main blocks corresponding to different models presented earlier in the chapter. The chapter also presented a simple mission planning interface used to define the mission profile, change the autopilot settings, and review a mission. The autopilot was tested for random changes in the VDM parameters reaching 10% of the initial values with the final position error not exceeding 60 metres.

The next chapter will use the developed 6DOF aircraft model to evaluate the performance of the state-of-the-art VDM integration architecture.

4 Loosely Coupled VDM

4.1. Introduction

In Chapter 2, two concepts in using the VDM in an integrated architecture were introduced, namely model-aided and model-based schemes. Model-aided schemes typically use the VDM to aid an INS solution, while in a model-based scheme, the VDM is the main process model aided by available systems and sensors such as GNSS and inertial sensors. Essentially, the two concepts use control inputs to propagate navigation states using a set of equations that describe the motion of the vehicle under the influence of applied forces and moments, as presented in the previous chapter. An additional input to the VDM architecture is wind velocity, as can be seen in Figure 4.1.



$$U = [\delta_\alpha, \delta_e, \delta_r, n_c]^T \qquad W^n = [w_N, w_E, w_D]^T$$

Figure 4.1. VDM with control and wind velocity input.

The current VDM navigation schemes, presented in Section 2.4, tend to use a loosely coupled configuration, using available position and velocity from a GNSS receiver in the fusion filter. In most implementations, it is assumed that the mass moment of inertia terms are perfectly known. The impact of the mass moment of inertia errors on the navigation solution has not been analysed. In most cases, it is assumed that a GNSS receiver does not recover following a GNSS outage. Moreover, the characteristics of the resulting errors in different GNSS outage intervals have not been examined. Having identified these challenges, the chapter presents and analyses a VDM navigation scheme that estimates the mass moment of inertia terms and compares its navigation performance to an existing model-based integration approach during a GNSS outage. The chapter then goes on to present the error characteristics of a VDM navigation scheme during different GNSS outage intervals followed by periods of reacquisition.

The chapter is organised as follows. In Section 4.2, the current state-of-the-art VDM navigation approach is briefly presented. Section 4.3 presents a VDM navigation scheme with inertia estimation. Section 4.4 presents the navigation solution error characteristics of a VDM navigation scheme during different GNSS outage intervals, and the chapter summary is presented in Section 4.5.

The work presented in this chapter has been published in Mwenegoha *et al.* (2019a).

4.2. Current VDM Navigation

The current state-of-the-art VDM integration scheme falls under the model-based category (shown in Figure 4.2). Since a VDM navigation architecture is simply a mathematical model, it is robust against platform vibrations and thermal effects, usually affecting other systems such as an INS, which rely on IMU data to propagate the navigation states. This makes a model-based scheme more attractive than a model-aided scheme, especially in applications where the quality of the inertial sensors is low. Additionally, results presented by Khaghani and Skaloud (2018b) demonstrate the resilience provided by a VDM/GNSS integration approach in the absence of IMU data. Their simulation results show that a model-based approach can provide a navigation solution that is sufficient for UAV guidance and control in the absence of IMU measurements. This is unlike a model-aided scheme that would disable the navigation solution altogether. Multi-process models which rely on the INS for the final navigation solution, such as the one proposed by Koifman and Bar-Itzhack (1999), have similar limitations to model-aided schemes.

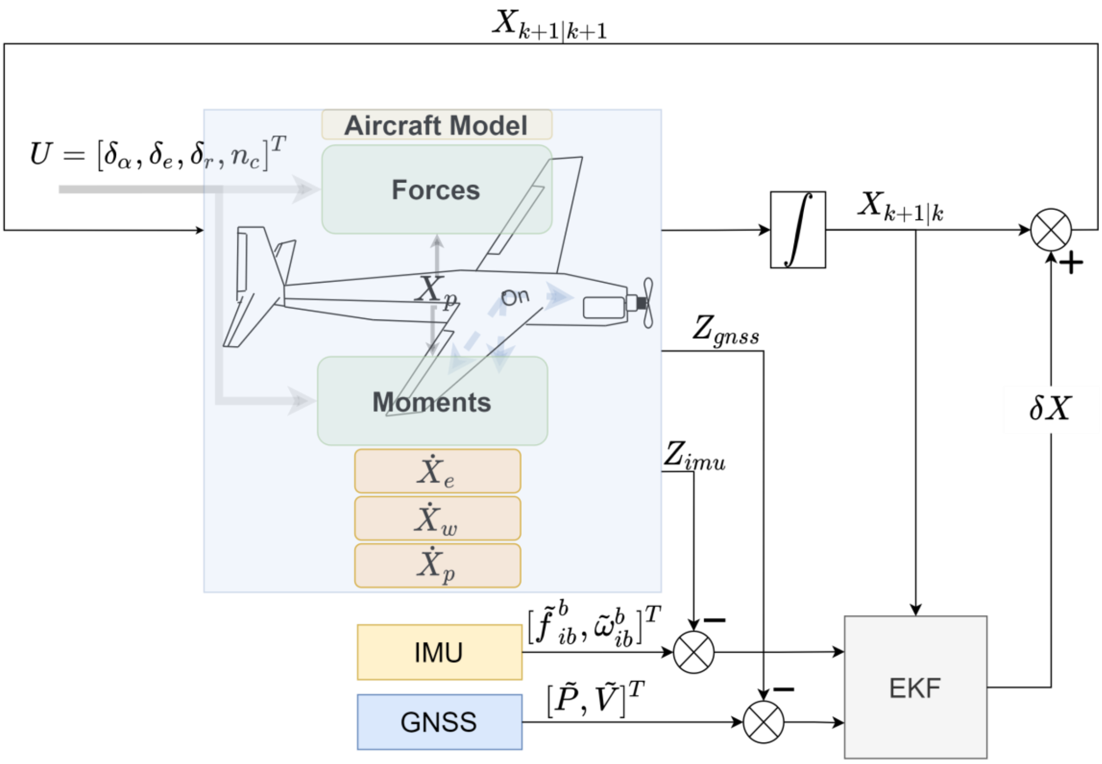


Figure 4.2. Model-based integration architecture.

Even though a model-based scheme is more attractive than a model-aided scheme, it is still limited by initialisation errors, VDM parameter uncertainties, wind velocity uncertainty and numerical integration errors. These limitations mean that such a system also requires an absolute positioning system to provide an error-bound navigation solution.

The implementation of the current state-of-the-art model-based integration architecture for a fixed-wing UAV is shown in Figure 4.2. The architecture uses an extended Kalman filter (EKF) alongside measurements from a GNSS receiver and a MEMS-grade IMU to aid the solution. Other than the navigation states, the state vector also includes IMU error states, wind velocity states and VDM parameters.

4.3. VDM with Inertia Estimation

This section presents and analyses a loosely coupled model-based scheme that includes the mechanism to estimate the mass moment of inertia directly. The architecture is tested using both GNSS and IMU measurements during an extended GNSS outage. The unscented Kalman filter (UKF) is used as the fusion filter in estimating corrections to the navigation states.

The architecture is evaluated via a Monte Carlo simulation study with a predefined trajectory and a variable wind profile assuming commercial off-the-shelf low-quality MEMS-grade inertial sensors. The assessment is made in terms of navigation accuracy and filter consistency, especially during periods of extended GNSS outage. This section addresses two key questions:

- *Is the navigation performance significantly affected by any errors in the moment of inertia terms, especially during a GNSS outage?*
- *Does the choice of the navigation filter significantly affect navigation performance?*

In addressing the questions, the architecture is briefly described, followed by a description of the filtering methodology used to assess the performance of the architecture. Simulation results are then presented at the end of the section.

4.3.1. Description

Figure 4.3 shows the architecture proposed to study the navigation performance of a loosely coupled architecture in comparison to the state-of-the-art scheme. The architecture uses the unscented Kalman filter to estimate corrections to the navigation states using IMU and GNSS measurements. Other than the navigation states X_n, the state vector is augmented to include wind velocity states X_w, IMU error terms X_e, and VDM parameters X_p. The VDM parameters also include the moment of inertia terms. The difference between the state-of-the-art model-based scheme and the presented architecture lies in the choice of the filter and the augmentation of the state vector to include the mass moment of inertia terms.

As presented in the previous chapters, a VDM requires a set of parameters (X_P) used to derive the moments and forces acting on the aircraft. Pre-calibration of these parameters before a flight is possible, but this can be time-consuming and usually requires expensive equipment defeating the whole purpose of a low-cost routine. The VDM structure allows the calibration of these parameters during a flight reducing the effort in obtaining accurate parameters.

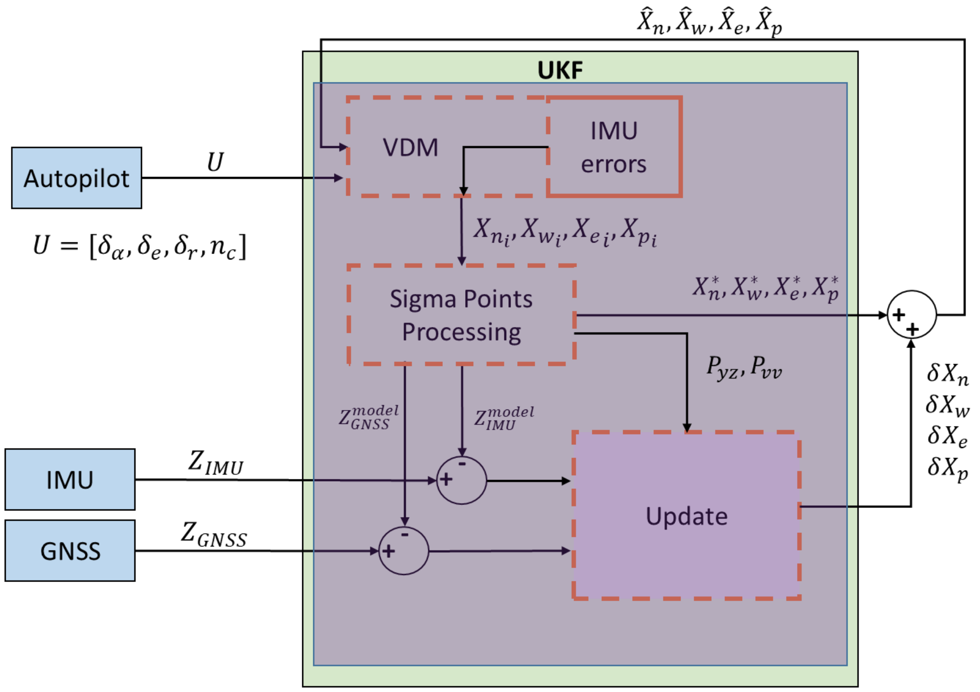


Figure 4.3. VDM-based UKF architecture.

4.3.2. Filtering Methodology

The UKF is chosen to serve as the navigation filter in this study. This filter has been discussed in Chapter 2.

The navigation filter uses a set of appropriately chosen weighted points to parameterise the mean and covariance of a probability distribution. Figure 4.4 shows the underlying mechanism of generating the sigma points followed by appropriate transformations and the calculation of the covariance matrices.

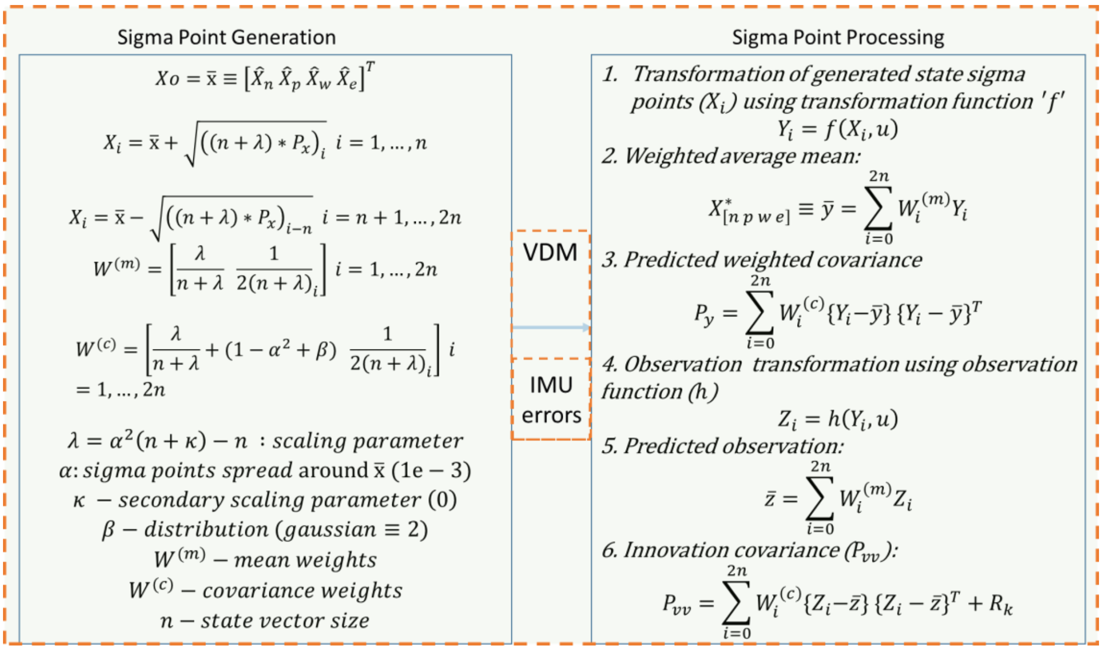


Figure 4.4. The mechanism to generate sigma points.

The navigation states are propagated using the standard rigid body equations of motion presented in Chapter 3 with some simplifications. The aircraft is assumed to be flying over a small region, and therefore the Earth is assumed to be locally flat. Coriolis acceleration due to the rotation of the Earth is ignored, and a local navigation frame (NED) fixed at the take-off point is considered the local inertial frame. Therefore, the navigation states considered in this study include:

$$X_n = \left[x_N, x_E, x_D, v_x^b, v_y^b, v_z^b, \phi, \theta, \psi, \omega_x, \omega_y, \omega_z, n\right]^T \tag{4.1}$$

the position vector $x^n = [x_N, x_E, x_D]^T$ is in the NED frame relative to the take-off point. Based on the assumptions, the simplified equations of motion are given by:

$$\dot{x}^n = R_b^n v^b \tag{4.2}$$

$$\dot{v}^b = f_{ib}^b + (R_b^n)^T g^n - \omega_{ib}^b \times v^b \tag{4.3}$$

$$\dot{\phi}_{nb} = R_w \, \omega_{ib}^b \tag{4.4}$$

$$\dot{\omega}_{ib}^b = (I^b)^{-1}(M - \omega_{ib}^b \times I^b \omega_{ib}^b) \tag{4.5}$$

where $v^b = \left[v_x^b, v_y^b, v_z^b\right]^T$ is the inertial velocity vector in the body frame. The transformation matrix R_b^n in terms of the Euler angles $\phi_{nb} = [\phi, \theta, \psi]^T$ is given by:

$$R_b^n = R_3^T(\psi) R_2^T(\theta) R_1^T(\phi) \tag{4.6}$$

and R_w is given by:

$$R_w = \begin{bmatrix} 1 & \tan\theta \, sin\phi & \tan\theta \cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \, / \cos\theta & \cos\phi \, / \cos\theta \end{bmatrix} \tag{4.7}$$

It is assumed that the pitch angle θ of the aircraft does not reach $\pm 90°$ and therefore, the matrix R_w is always defined.

To capture slow transitions in wind velocity, a random walk process is used in the navigation filter (Khaghani and Skaloud, 2016a). The wind velocity vector is defined as:

$$X_w = [w_N, w_E, w_D]^T \tag{4.8}$$

No deterministic part is considered in the process model, and only white noise is considered to rule the transition in time. This process model is given by:

$$\dot{X}_w = G_w W_w \tag{4.9}$$

$$G_w = [I]_{3\times 3}$$

where G_w is the noise shaping matrix, W_w is the noise vector, and $[I]_{m\times n}$ represents the identity matrix with m rows and n columns (note: $m = n$).

A random walk process model is also used to model the IMU error terms. The IMU error states are defined as:

$$X_e = [b_{ax}, b_{ay}, b_{az}, b_{gx}, b_{gy}, b_{gz}]^T \tag{4.10}$$

where $b_{a[x,y,z]}$ and $b_{g[x,y,z]}$ represent the accelerometer and gyroscope biases, respectively. The process model contains only white noise and is given by:

$$\dot{X}_e = G_e W_e \tag{4.11}$$

$$G_e = [I]_{6\times6}$$

where G_e is the noise shaping matrix and W_e is the noise vector for the IMU error terms. In the simulation, the IMU error terms are modelled using a first-order Gauss-Markov process with specific details given in the next section.

In the filter, VDM parameters are assumed to contain some initial uncertainty. However, during a flight, these parameters are considered static. Twenty-two model parameters and four mass moment of inertia terms are considered. These are defined as:

$$X_p = \begin{bmatrix} CF_{T_1}, & CF_{T_2}, & CF_{T_3}, & CF_{X1}, & CF_{X\alpha}, & \ldots \\ \ldots CF_{X\alpha 2}, & CF_{X\beta 2}, & CF_{Z1}, & CF_{Z_\alpha}, & CF_{Y1}, & \ldots \\ \ldots CM_{X_{\delta_\alpha}}, & CM_{X_\beta}, & CM_{X_{\bar{\omega}_x}}, & CM_{X_{\bar{\omega}_z}}, & CM_{Y1}, & \ldots \\ \ldots CM_{Y_\alpha}, & CM_{Y_{\delta_e}}, & CM_{Y_{\bar{\omega}_y}}, & CM_{Z_{\delta_r}}, & CM_{Z_{\bar{\omega}_z}}, & \ldots \\ \ldots CM_{Z_\beta}, & \tau_n, & I_{xx}, & I_{yy}, & I_{zz}, & \ldots \\ \ldots I_{xz} & & & & & \end{bmatrix}^T \tag{4.12}$$

A random walk process with a small process noise is used to model the mass moment of inertia terms and model parameters collectively referred to as VDM terms (X_p). The process model is formulated as:

$$\dot{X}_p = G_p W_p \tag{4.13}$$

$$G_p = [I]_{26\times26}$$

where G_p is the noise shaping matrix and W_p is the small noise vector. As previously stated, during a flight, these parameters are considered fixed.

Therefore, the state vector considered in the architecture is given by:

$$X = [X_n^T, X_w^T, X_e^T, X_p^T]^T \tag{4.14}$$

The measurement vector Z consists of the accelerometer and gyroscope outputs as well as GNSS receiver measurements. The IMU measurements are given by:

$$Z_{IMU} = \begin{bmatrix} f_{ib}^b \\ \omega_{ib}^b \end{bmatrix} + X_e + w_i \tag{4.15}$$

and the GNSS measurements are given by:

$$Z_{GNSS} = \begin{bmatrix} x_N \\ x_E \\ x_D \end{bmatrix} + w_g \tag{4.16}$$

where w_i and w_g represent the residual error for the IMU and GNSS measurements, respectively, modelled as Gaussian white noise. The measurement covariance matrices are obtained from the simulated error characteristics presented in the next section.

4.3.3. Simulation Setup

The trajectory used in the simulation is shown in Figure 4.5. The flight profile includes a take-off segment, a climb segment to an altitude of 700 m, and a cruise segment where a GNSS outage is induced, followed by a descent and approach segment. The outage is induced 200 seconds into the flight and assumed to last for the remainder of the flight (lasts for 140 seconds). The total flight time is 340 seconds.

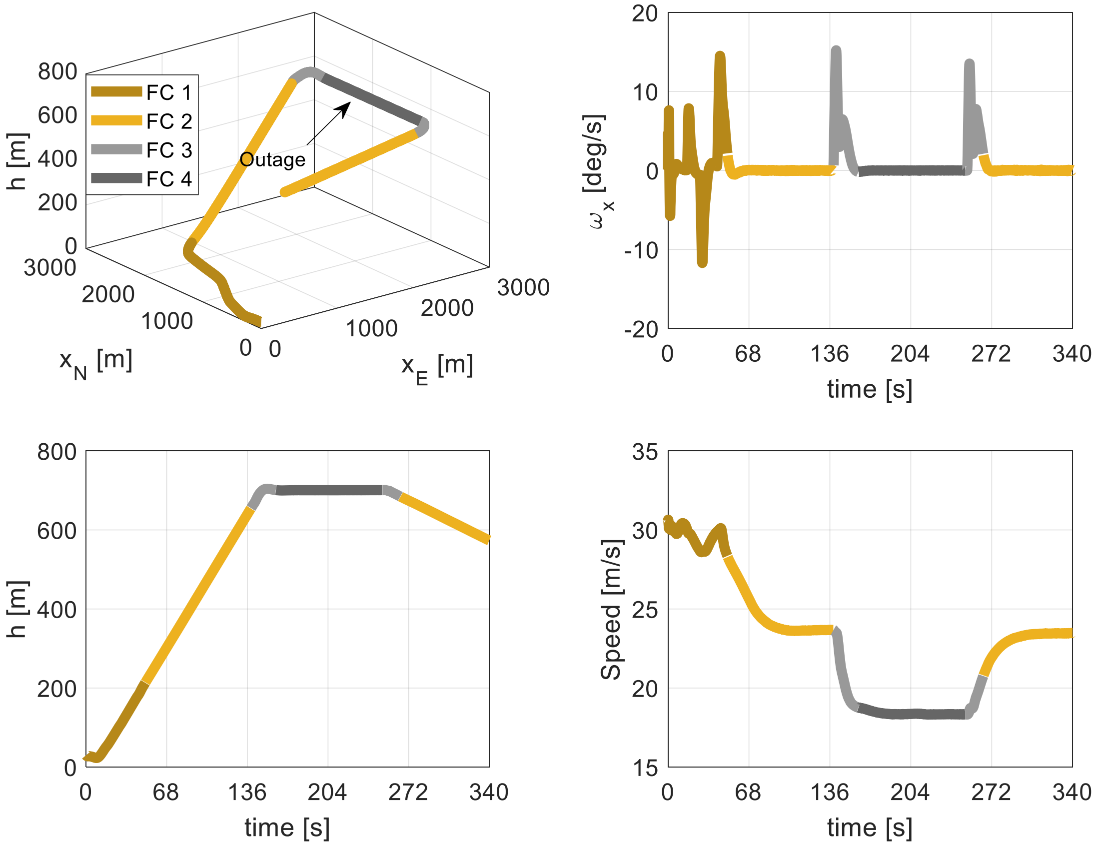


Figure 4.5. Trajectory used in the simulation (top left). Roll rate (top right), altitude (bottom left) and aircraft speed (bottom right) during different flight phases.

The trajectory is derived using error-free sensors. In the simulation, guidance and control are independent of the architecture under investigation. The control inputs are assumed to be available from the autopilot system. Such an extended GNSS outage may occur due to external interference, such as from a GNSS signal jammer. Wilde *et al.* (2016) demonstrated how a 10 mW chirp jammer could cause a GNSS outage in a low-cost receiver on a UAV for an extended range reaching 1 km. During this period, the receiver did not output a navigation solution (no position and velocity measurements).

The flight profile is categorised according to four flight characteristic regions (FC) based on speed changes, altitude changes and aircraft manoeuvres. Table 4.1 shows the changes in speed, altitude, and aircraft manoeuvres during different phases. A segment with a significant change in altitude coupled with compound manoeuvres is denoted FC 1. The take-off segment constitutes FC 1. A segment with a significant change in both speed and altitude is denoted FC 2, which in Figure 4.5 includes the climb and descent segments. A segment with significant manoeuvres and speed changes but with little or no change in altitude is denoted FC 3. A constant speed and constant altitude phase without manoeuvres is denoted FC 4 (straight and level, unaccelerated flight - SLUF).

The stochastic properties of the IMU considered in the simulation are given in Table 4.2. Similar values have been used in Khaghani and Skaloud (2018). The simulated IMU measurements are assumed to contain a random turn-on bias component, a dynamic bias component given by a first-order Gauss-Markov process and white noise. The simulated IMU is assumed to be sampled at 100 Hz, and all other errors such as misalignment errors and scale-factor variations are ignored. In practice, the true IMU error characteristics will not be known, and the measurements from the IMU may contain unmodeled correlated components. To overcome the limitations of models used in a Kalman filter, sufficient noise must be modelled to overbound the real system's behaviour (Groves, 2013). Therefore, to maintain a situation close to reality, the navigation filter uses scaled values (in the range from 1 to 2 applied to each axis).

Table 4.1. Flight characteristics.

FC	Speed change	Altitude change	Manoeuvre
1			
2			
3			
4			
	No Significant change	Significant change	

Table 4.2. IMU errors in the simulation.

Property	Accelerometer	Gyroscope
Random bias (σ)	10 mg	1000 °/hr
White noise (PSD)	100 μg/√Hz	21.6 °/hr/√Hz
First-order Gauss-Markov (σ)	0.05 mg	20 °/hr
Correlation Time (τ)	200 s	200 s
Sampling Frequency	100 Hz	100 Hz

In the simulation, GNSS measurements are assumed to contain white noise with a standard deviation of 1 m sampled at 1 Hz, similar to the modelling effort in Khaghani and Skaloud (2016, 2018).

Table 4.3 presents the standard deviation of the initial error for the navigation states and VDM terms. Here, the VDM solution is assumed to be initialised using a different mechanism, such as from an INS/GNSS integration scheme.

Table 4.3. The standard deviation of the initial errors for the navigation states and VDM terms.

State	Standard deviation (1σ)
Position	[1.0, 1.0, 1.0] m
Velocity	[1.0, 0.5, 0.5] m/s
Attitude	[3.5°, 3.5°, 5°]
Rotation rates	1.5 °/s
Propeller speed	15 rad/s
Model parameters	10 %
Moment of Inertia terms	10%

The initial uncertainties for the navigation states range from one to two times the initial errors described. The initial values for IMU error terms and wind velocity components are set to zero. An initial uncertainty of 1.5 m/s is used for each wind velocity component. The initial uncertainties in the state covariance matrix for the IMU error terms are in general agreement with the IMU stochastic properties.

Table 4.4 presents the standard deviation of the tuned process noise used in the filter.

Table 4.4. Process noise.

State	Standard deviation (1σ)
Position	10^{-6} m
Velocity	0.008 m/s
Attitude	10^{-6} rad
Rotation rates	10^{-4} rad/s
Propeller speed	10^{-4} rad/s
Accelerometer Bias	2×10^{-5} m/s^2
Gyroscope Bias	2×10^{-6} rad/s
Wind	10^{-3} m/s
Model parameters	0.015% of True Values
Moment of Inertia	0.015% of True Values

One hundred Monte Carlo runs are performed in Matlab to evaluate the autonomous navigation performance and investigate the system's robustness against random initialisation and sensor errors. The justification of the number of simulations is described in Section B.1 of Appendix B.

4.3.4. Results

This section presents the navigation performance results of the model-based scheme using a UKF (UKF/VDM) compared to a model-based scheme using an EKF (EKF/VDM). The UKF/VDM architecture includes the mechanism to estimate the mass moment of inertia terms as described in the previous sections.

The position and velocity errors are presented in Figure 4.6 (w-I: with the mass moment of inertia estimated, w/o-I: without estimating the mass moment of inertia) in the NED and body-fixed coordinate frames, respectively. For the most part, during periods of GNSS availability, the predicted confidence values (1σ) are consistent with the empirical RMS of estimation errors except during short periods of high dynamics between 34 seconds and 50 seconds. During this time, the filter underestimates the longitudinal and lateral velocity errors leading to a slight underestimation of north and east position errors. This is attributed to unresolved initialisation errors. During periods of GNSS outage, position errors grow gradually, reaching only 14.5m in the north channel, 8m in the east and 4.3m in the down direction after 140 seconds of VDM coasting. Most of the velocity errors seem to be estimated within the first 60 seconds of GNSS availability. The error is seen to be less than 0.1m/s after 150 seconds of GNSS availability across all channels. There is a marginal growth in velocity errors following a GNSS outage, but these remain well within 0.1m/s across all channels.

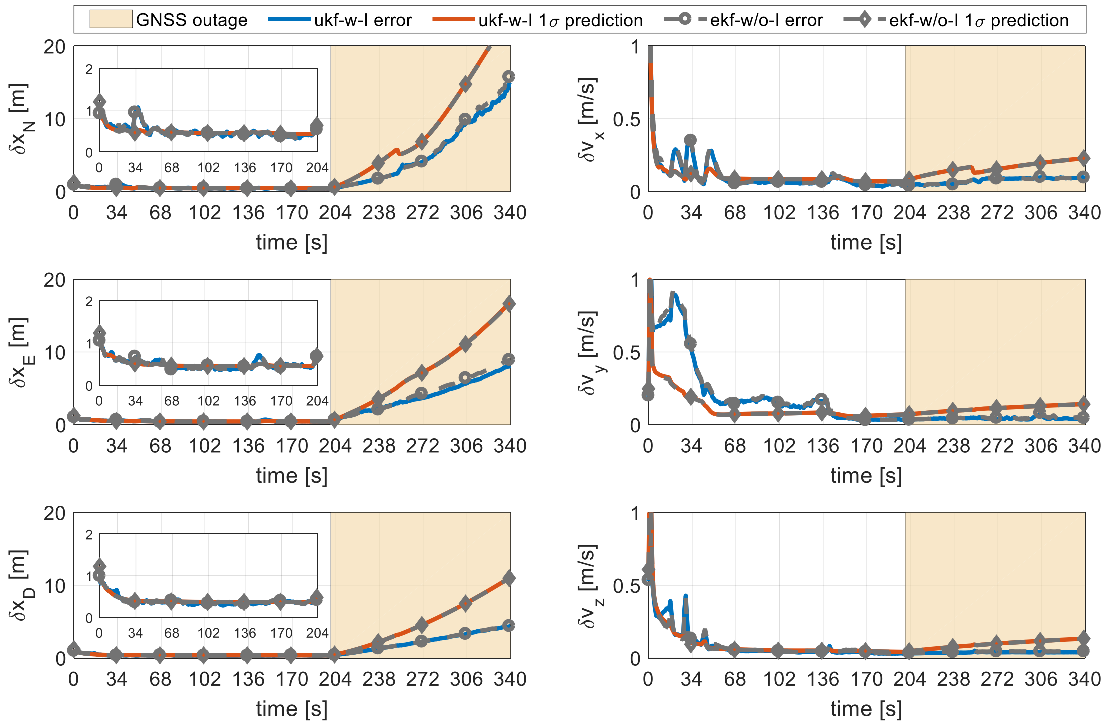


Figure 4.6. RMS of position errors for all 100 runs (left); RMS of velocity errors for all 100 runs (right).

Figure 4.7 shows the RMS of position errors for the proposed UKF/VDM architecture with the perturbed and augmented moment of inertia terms compared to the EKF/VDM architecture.

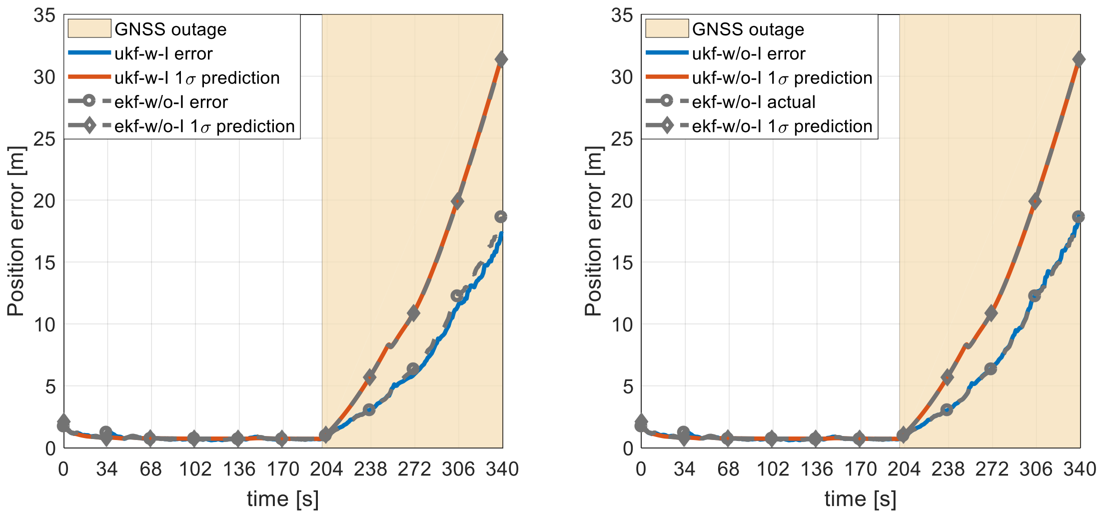


Figure 4.7. RMS of 3D position errors for all 100 runs. UKF/VDM architecture with- (left) and without- (right) moment of inertia terms.

Evidently, both setups show similar position error estimation performance. The differences are well within the precision of the Monte Carlo runs. It is important to note that these deductions are relatively similar for the remaining navigation states.

Most roll and pitch angle errors are estimated well within the first 50 seconds of GNSS availability. However, yaw angle errors are slightly delayed and only well resolved after 160 seconds of GNSS availability, as shown in Figure 4.8.

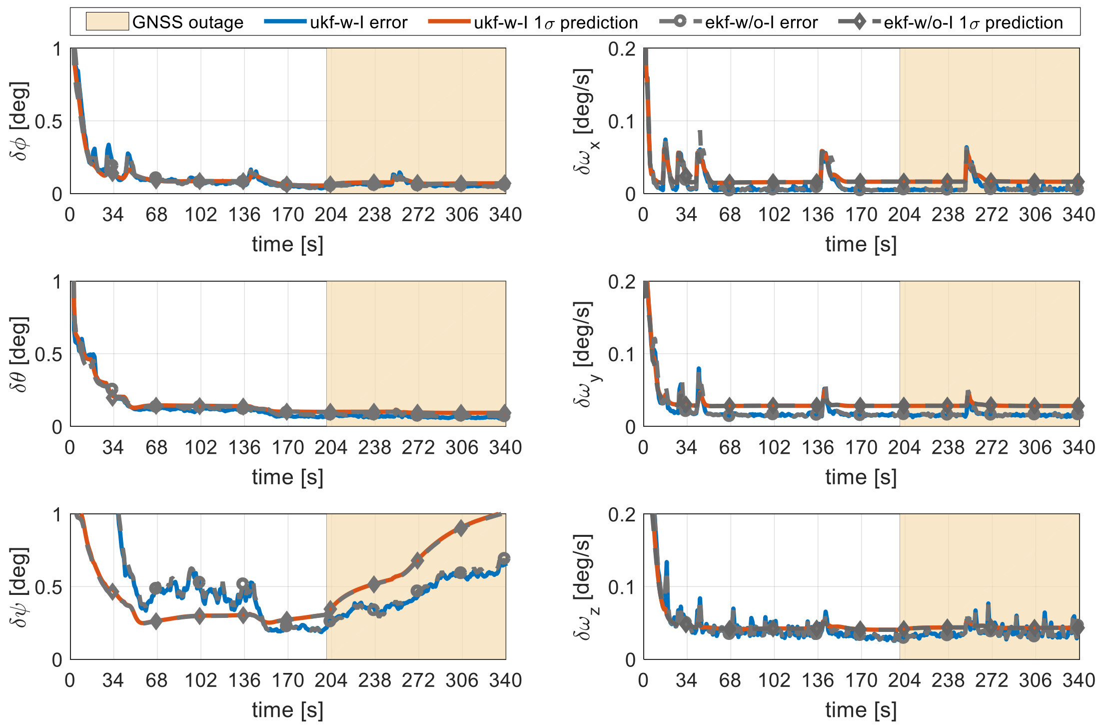


Figure 4.8. RMS of attitude errors (left) and angular velocity (right) errors for all 100 runs.

The lack of a direct heading reference and large initialisation errors in the yaw angle might be the cause of the delayed yaw angle error estimation during GNSS availability. The angular rates are quickly estimated within 20 seconds of GNSS availability due to the presence of direct observations from the gyroscopes. Roll and pitch angle errors remain within 0.06 degrees after 140 seconds of GNSS outage, while the yaw angle error increases gracefully to 0.65 degrees after 140 seconds of GNSS outage. The 1σ predictions also seem to be consistent with the empirical RMS of estimation errors even during periods of GNSS outage.

Figure 4.9 shows the RMS of estimation errors for the propeller speed and wind speed for 100 Monte Carlo runs. Even without direct RPM measurements, most of the error in the propeller speed seems to be estimated within the first 50 seconds of the flight. As a consequence of the first-order model used for the propeller speed and the unresolved initial errors, some spikes are noticeable during periods with significant commanded propeller speed inputs. However, the error does not increase during the GNSS outage indicating the filter's ability to keep track of this error even in the absence of GNSS data.

Wind speed is estimated well within the first 60 seconds of GNSS availability, and the error is less than 0.12 m/s, 150 seconds into the flight. The error in wind speed estimation is relatively small even after 140 seconds of GNSS outage, with the final estimation error being less than 0.2m/s. The filter also seems to be consistent in estimating wind speed, as can be seen from the predicted confidence values (1σ), even without an air data system, attributed to correctness in the filter setup. The navigation performance of a model-based approach is generally prone to errors resulting from unknown external disturbances such as wind. Therefore, the filter's ability to estimate wind even without an air data system makes this approach robust against external wind disturbances.

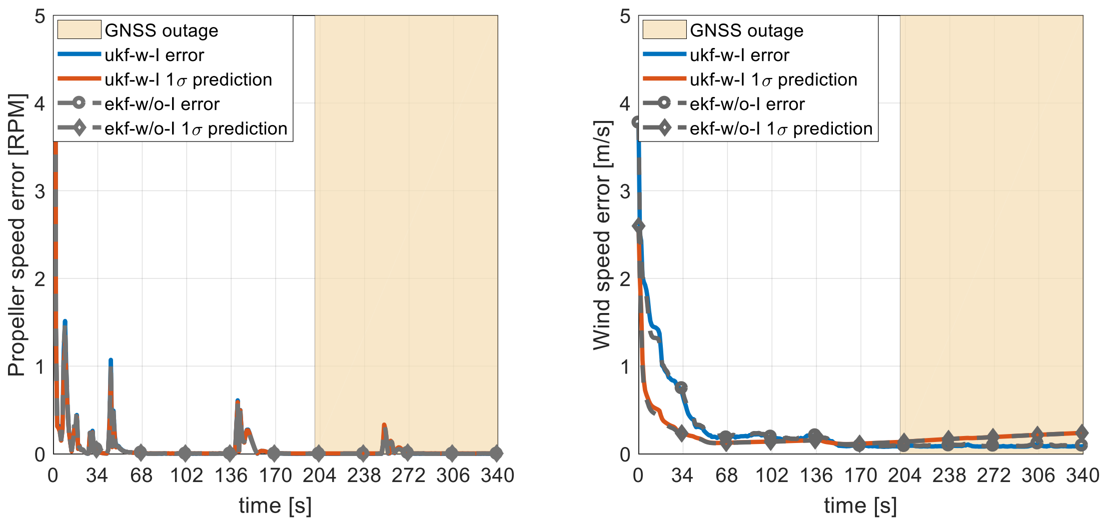


Figure 4.9. RMS of propeller speed errors (left) and wind speed errors (right) for all 100 runs.

Figure 4.10 shows the accelerometer and gyroscope bias estimation errors. The filter effectively and consistently estimates the accelerometer and gyroscope biases within the first 50 seconds of GNSS availability. The filter resolves 98% of the initial turn-on bias in the gyroscope measurements well within the first 100 seconds of GNSS availability. The estimation seems to improve even during 140 seconds of GNSS outage, thanks to the mitigation provided by the VDM. Also, the predicted confidence values (1σ) seem to be consistent with the empirical RMS of estimation errors, attributed to the correctness in filter setup.

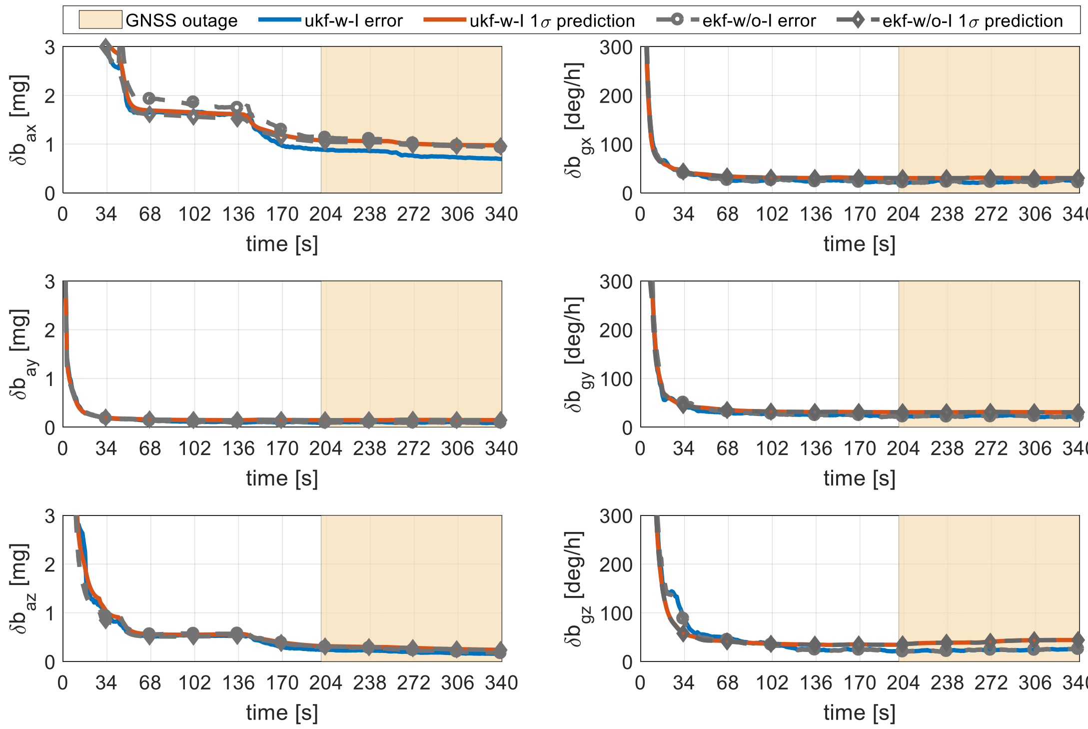


Figure 4.10. RMS of accelerometer errors (left) and gyroscope errors (right) for all 100 runs.

Results indicate that the z-axis and x-axis accelerometer biases (b_{az}, b_{ax}) are slightly delayed in their estimation. This might be attributed to the coupling between the attitude and accelerometer errors under low dynamics, and more separation and observability might be achieved with high dynamics.

The RMS of the mean error of the VDM terms is presented in Figure 4.11. With an initial uncertainty of 10% considered in each of the VDM terms, the filter is slightly optimistic in estimating VDM terms. Generally, the initial error in the VDM terms is seen to reduce quickly during periods of GNSS availability and remains bounded during periods of GNSS outage. The VDM terms error reduces quickly to less than 7.5% within 50 seconds of GNSS availability and then gradually to 6.6% at the onset of the GNSS outage. Strong coupling and correlation of the VDM terms might be the reason for the slight inconsistent estimate, with the difference between the mean error and the filter's prediction being less than 19% at the end of the flight. Perhaps more dynamic manoeuvres, exciting different modes, could improve the overall estimation performance of the VDM terms, as demonstrated in Laupré and Skaloud (2020). However, the current results are deemed well enough for navigation due to the consistent estimate in navigation states. It is worth mentioning that, for a similar setup, the EKF/VDM architecture, with perturbed but not augmented inertia terms, is found to be overly inconsistent in the estimation of the model parameters with a difference of 48.5% between the mean error and the prediction at the end of the flight.

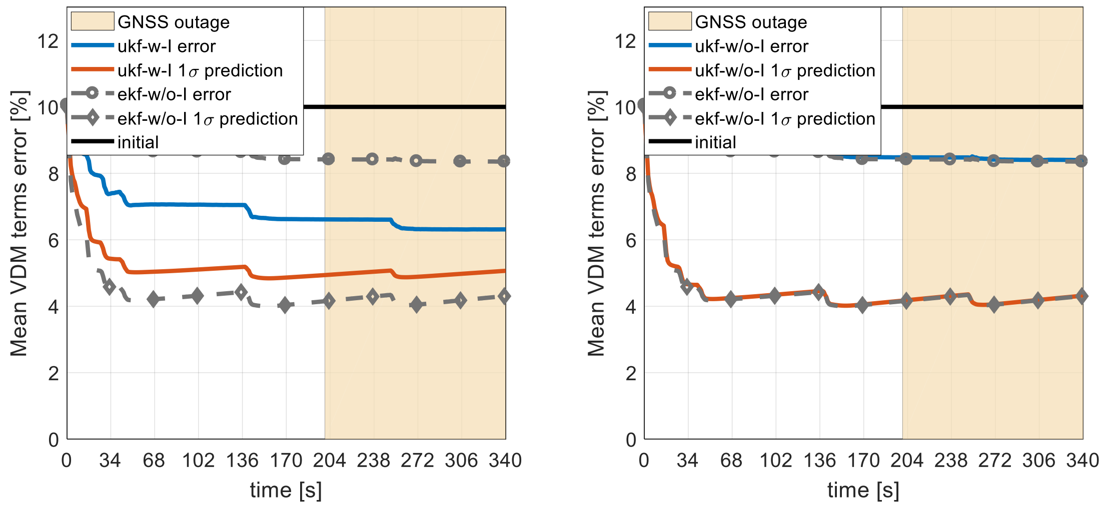


Figure 4.11. RMS of the mean error of all VDM terms. UKF/VDM architecture with (left) and without (right) moment of inertia terms.

This is not surprising and highlights the importance of estimating the moment of inertia terms, especially when they contain some errors. This also shows that the proposed architecture could be used with greater confidence in the presence of inertia tensor perturbations saving both time and cost associated with inertia modelling. Further, Figure 4.11 (on the right) shows the performance of both a UKF/VDM and EKF/VDM architecture without the moment of inertia terms in the state vector. The results show that the choice of the filter does not significantly influence the estimation performance of the model parameters.

Figure 4.12 shows the mean estimation error for the thrust and drag force coefficients. The navigation filter estimates about 45% of the initial error in the thrust coefficients within the first 50 seconds of GNSS availability. The first 50 seconds of the flight make up the FC1 flight segment (see Table 4.1) marked by significant manoeuvres and altitude changes. The errors do not grow during the GNSS outage lasting 140 seconds. About 25% of the initial drag coefficient errors are estimated within the first 50 seconds of GNSS availability. The drag force coefficient errors also do not grow during the GNSS outage.

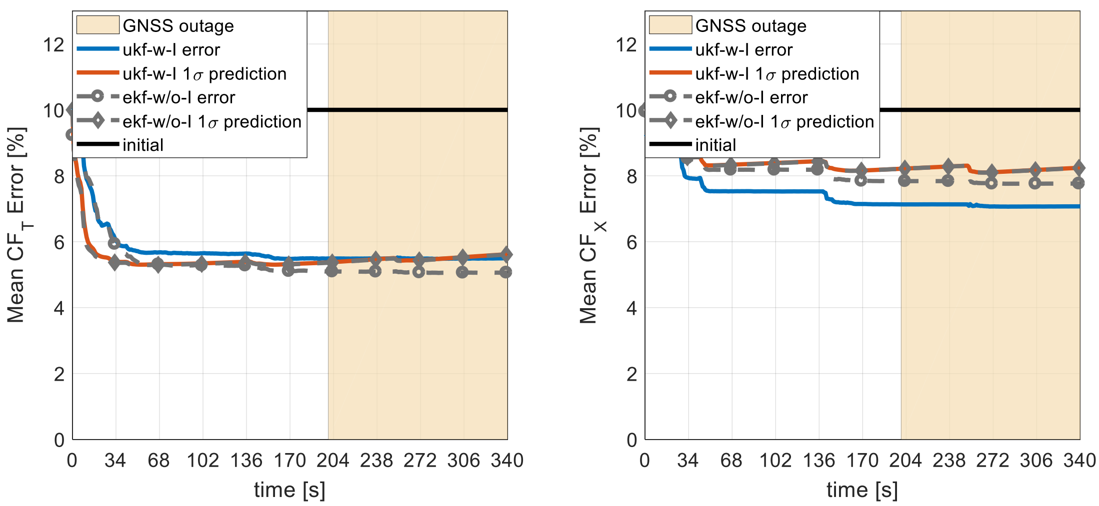


Figure 4.12. RMS of the mean error for thrust force coefficients (left) and drag force coefficients (right) for all 100 runs.

Figure 4.13 shows the mean error in the estimation of lift and lateral force coefficients.

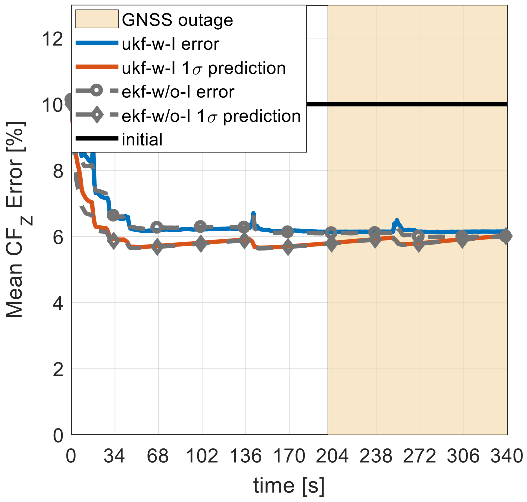


Figure 4.13. RMS of the mean error for lift force coefficients (left) and the lateral force coefficient (right) for all 100 runs.

The estimation error of the lift force coefficients seems to reduce quickly in the first 50 seconds of the flight. During this period, the aircraft experiences a significant change in altitude and compound manoeuvres. A combination of altitude changes and compound manoeuvres did not seem to improve the estimation of the lateral force coefficient. Sharp turns with associated changes in aircraft speed at constant altitude seem to improve the estimation of the lateral force coefficient, indicated by the improved estimate around 136 seconds and 250 seconds. A GNSS outage does not seem to influence the estimation performance of both the lift and lateral force coefficients.

Figure 4.14 shows the estimation error for the rolling and pitching moment coefficients. Altitude changes and compound manoeuvres (FC1) seem to improve the estimation of rolling moment coefficients gradually.

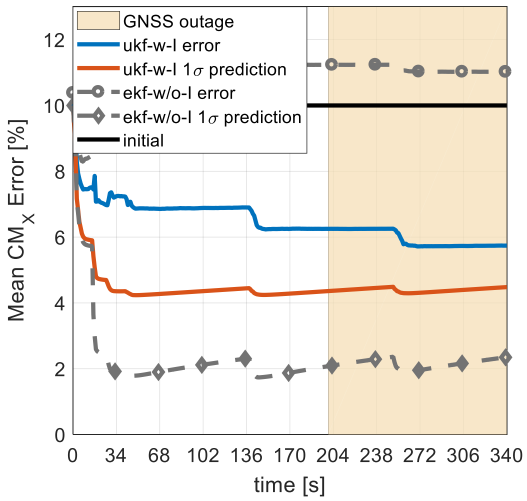


Figure 4.14. RMS of the mean error for roll moment coefficients (left) and pitch moment coefficients (right) for all 100 runs.

On the other hand, sharp turns and significant speed changes (FC3) at constant altitude seem to rapidly improve the estimation of rolling moment coefficients even during a GNSS outage. The mean error of the pitching moment coefficients seems to rapidly improve in the presence of altitude changes and compound manoeuvres (FC1). The error improves slightly during sharp turns at constant altitudes. There is also a noticeable difference between the UKF/VDM architecture with mass moment of inertia error estimation and the EKF/VDM architecture that does not estimate the mass moment of inertia error. The error in the mass moment of inertia terms influences the torque derivatives because they are significantly correlated.

Figure 4.15 shows the estimation error for the yawing moment coefficients and propulsion unit time constant. The estimation of yawing moment coefficients seems to improve only slightly during the GNSS availability period attributed to the significant correlation within the terms, which limits their observability. More dynamic manoeuvres might help improve the estimation of these coefficients. Most of the error in the propulsion unit time constant seems to be estimated during the GNSS availability period. Only 30% of the initial error remains at the end of the flight. The estimation even improves during the GNSS outage period.

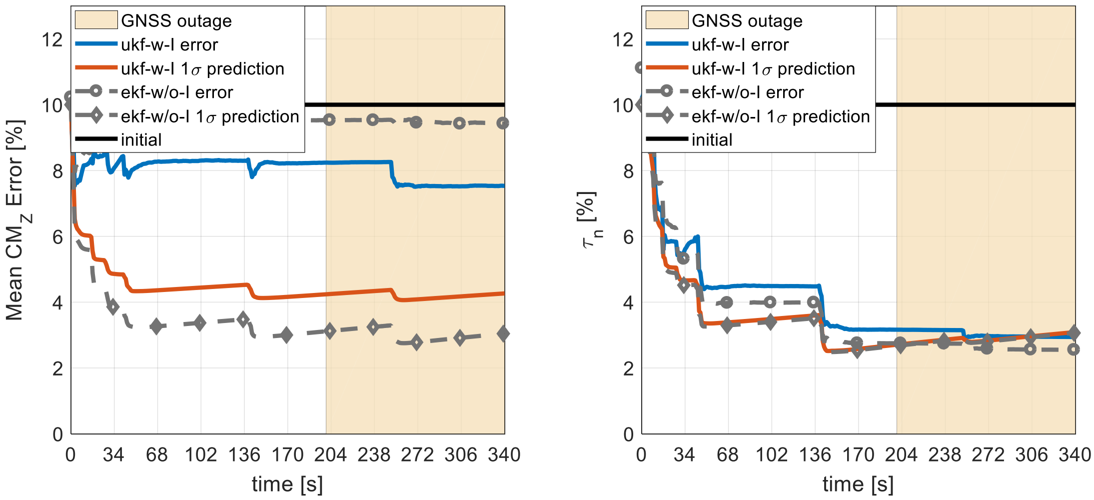


Figure 4.15. RMS of the mean error for yawing moment derivatives (left) and the propeller time constant (right) for all 100 runs.

The error in the estimation of the mass moment of inertia terms is presented in Figure 4.16. With an initial uncertainty of 10% in the moment of inertia terms, it can be seen that errors in the roll and pitch terms reduce quickly within the first 50 seconds of GNSS availability. However, the filter appears to be slightly optimistic in estimating these terms. The estimated errors do not grow even during periods of extended GNSS outage lasting 140 seconds. The difference between the filter's 1σ prediction and the empirical RMS of estimation errors at the end of the flight is 24.4% for the roll axis (I_{xx}) and 33% for pitch axis (I_{yy}). The mass moment of inertia term in the yaw axis (I_{zz}) seems to be resolved after the initial errors in the roll and pitch terms have been resolved. The mass moment of inertia term in the yaw axis continues to be observable even during periods of GNSS outage, with a difference of 23% between the filter's 1σ prediction and the empirical RMS of estimation errors at the end of the flight. Sharp turns with rapid changes in speed seem to significantly improve the estimation of the mass moment of inertia term in the yaw axis. The estimation of the product of inertia seems to improve in the initial phase of the flight within the first 30 seconds but gradually diverges for the remainder of the flight. This might be attributed to the degree of coupling and lack of enough excitation for the product of inertia to be observable, and perhaps more dynamic manoeuvres could improve the overall estimation.

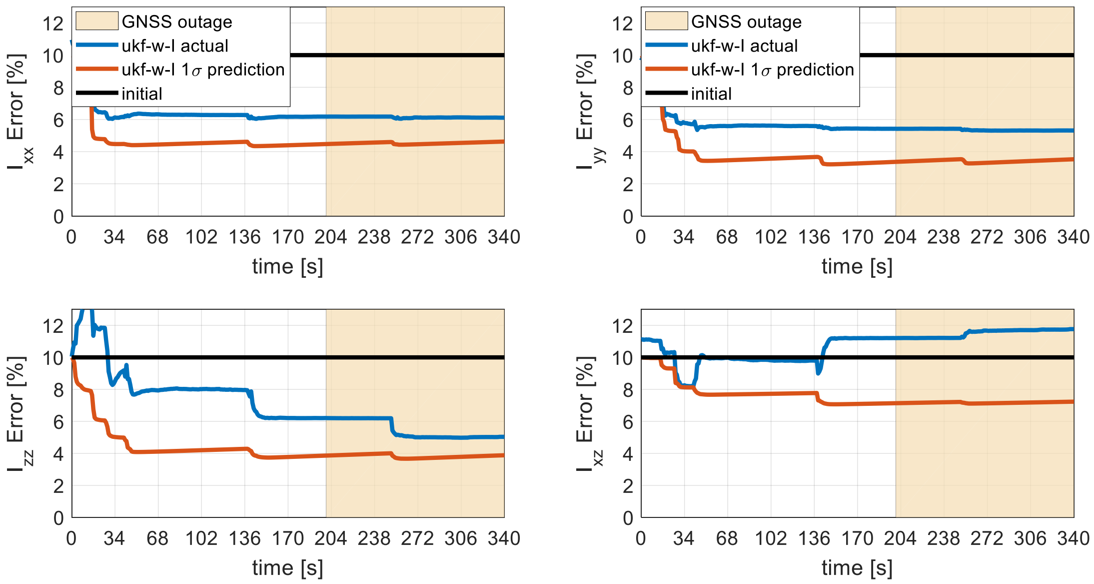


Figure 4.16. RMS of estimation errors for the mass moment of inertia terms for all 100 runs for the UKF/VDM scheme.

For a better understanding of the correlation properties between the different states, a correlation matrix is presented in Figure 4.17. Generally, the mass moment of inertial terms seem to be decorrelated from most of the navigation states. However, they seem to be significantly correlated with the moment derivatives which is not surprising, owing to the formulation of the rigid body equations. Further, the moment derivatives show significant correlation within groups. The pitching and yawing moment coefficients show significant cross-correlation. The rolling moment coefficients seem to be decorrelated from the rest of the moment derivatives. This is attributed to the high level of dynamics in the roll axis, which improves their overall group observability. It is also important to note that the model parameters showed some correlation with some navigation states, which is essential for improved observability of the parameters. The wind velocity states seem to be significantly correlated with the position and velocity states and could help explain the significant growth in position error during periods of GNSS outage.

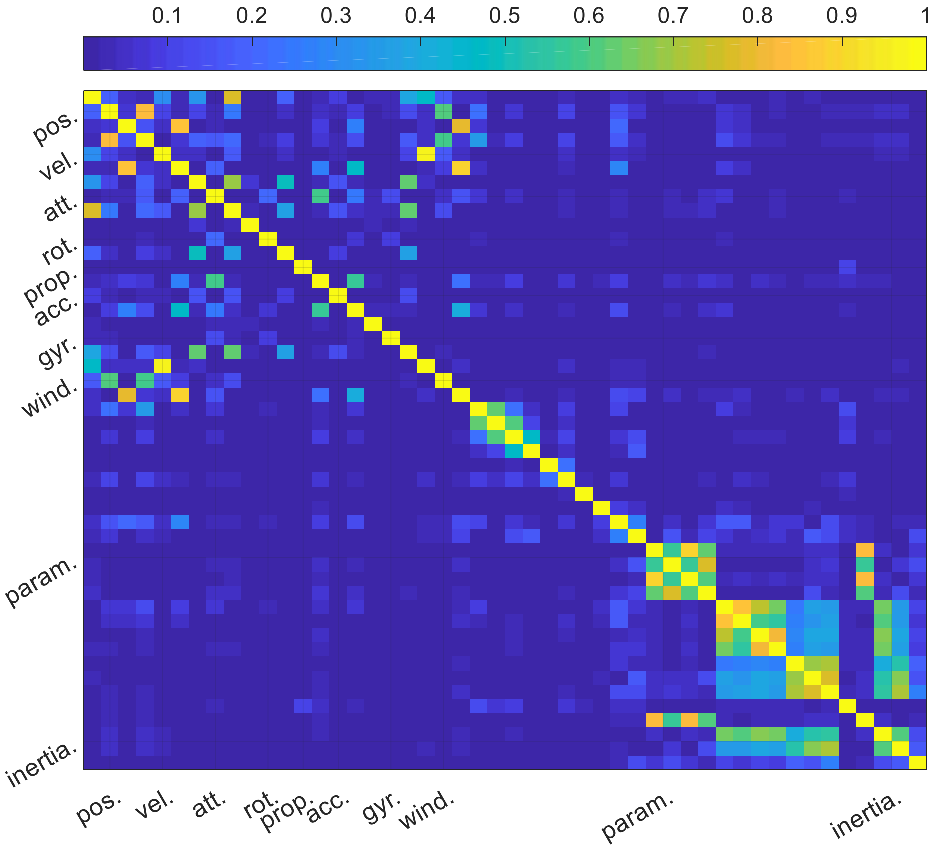


Figure 4.17. Correlation matrix at the end of the flight for one realisation.

4.3.5. Summary

This section presented and analysed a UKF/VDM integration architecture that uses measurements from a MEMS-grade IMU and position measurements output by a GNSS receiver with specific application to a fixed-wing UAV. The architecture included the mechanism to estimate the mass moment of inertia terms directly, which reduces the need for laborious routines in estimating these terms. The performance of the architecture was compared to the state-of-the-art EKF/VDM architecture, which does not include the mechanism to estimate the mass moment of inertia terms. A Monte Carlo simulation study was used to evaluate the architectures. The VDM scheme was assumed to be initialised from a different integration approach, such as an INS/GNSS scheme. A GNSS outage was induced 200 seconds into the flight and lasted for 140 seconds. During the outage, position measurements were not available to the filter.

Several deductions were made, and three important ones are highlighted here:

- In a VDM-based architecture that utilises IMU measurements and position measurements from a GNSS receiver, errors (within 10% of the true values) in the moment of inertia terms do not significantly influence the estimation performance of the navigation states.
- With an initial uncertainty of 10% in the model parameters, the choice of the navigation filter, either a UKF or EKF, does not significantly affect the estimation performance of the navigation states.

- Errors in the moment of inertia terms strongly influence the estimation performance of the torque coefficients due to the significant correlation between them. However, the estimation of inertia terms reduces the errors in the torque coefficients.

The investigation in this section has addressed the two questions posed at the beginning of the section. However, throughout the investigation, it has been assumed that a GNSS outage occurred only once. The outage interval could vary, and the navigation solution errors might present different characteristics with different GNSS outage intervals. The next section investigates the characteristics of these errors for different outage intervals.

4.4. VDM Error Characteristics

In the previous section, the navigation performance of a VDM-based integration architecture was investigated assuming the GNSS outage occurs only once during the flight. In a typical scenario, a GNSS receiver could recover tracking of the satellites lost during the outage. Further, the outage interval could be different based on the persistence of the conditions causing the outage. Therefore, this section investigates the characteristics of the navigation solution errors of a model-based integration architecture during different lengths of GNSS outage.

First, the integration architecture used in the investigation is described, followed by a description of the filtering methodology. The simulation setup is then described, followed by a discussion of the results.

4.4.1. Description

The model-based integration architecture used to characterise the navigation solution errors during different lengths of GNSS outages is similar to that shown in Figure 4.2. In this architecture, the dynamic model of a fixed-wing UAV is used as the main process model. Both IMU measurements and position measurements from a GNSS receiver are used in the navigation filter. Here, an EKF is used as the fusion filter.

4.4.2. Filtering Methodology

The use of an EKF as the fusion filter requires the computation of the Jacobians from the process and observation models. The process and observation models are similar to those presented in Section 4.3.2. The process models are linearised to provide dynamic matrices used in the EKF. The linearised process model is of the form:

$$\dot{X} = FX + GW_s \tag{4.17}$$

this is equivalent to the form:

$$\begin{bmatrix} \dot{X}_n \\ \dot{X}_e \\ \dot{X}_w \\ \dot{X}_p \end{bmatrix} = \begin{bmatrix} F_{nn} & F_{ne} & F_{nw} & F_{np} \\ F_{en} & F_{ee} & F_{ew} & F_{ep} \\ F_{wn} & F_{we} & F_{ww} & F_{wp} \\ F_{pn} & F_{pe} & F_{pw} & F_{pp} \end{bmatrix} \begin{bmatrix} X_n \\ X_e \\ X_w \\ X_p \end{bmatrix} + \begin{bmatrix} G_n & 0 & 0 & 0 \\ 0 & G_e & 0 & 0 \\ 0 & 0 & G_w & 0 \\ 0 & 0 & 0 & G_p \end{bmatrix} \begin{bmatrix} W_n \\ W_e \\ W_w \\ W_p \end{bmatrix} \tag{4.18}$$

the process models are linearised with respect to all the states given by:

$$F_{ij} = \partial \dot{X}_i / \partial X_j \tag{4.19}$$

The dynamic matrix has been evaluated analytically even though no significant changes in the results were noticed with numerically evaluated matrices. The numerical evaluation of the dynamic matrix is usually done using complex-step derivatives (Squire and Trapp, 1998). The analytical form is numerically stable and computationally efficient (Khaghani and Skaloud, 2016a). It is assumed that the sub-state vectors are uncorrelated, and therefore the off-diagonal blocks in the G matrix are all zero. For the navigation states, uncertainty terms apply only to linear and rotational accelerations as well as the propeller rotational acceleration. Therefore, the standard deviation of the driving noise on the navigation states is given by:

$$\sigma_{W_n} = \left[\sigma_{\dot{v}_x}, \sigma_{\dot{v}_y}, \sigma_{\dot{v}_z}, \sigma_{\dot{\omega}_x}, \sigma_{\dot{\omega}_y}, \sigma_{\dot{\omega}_z}, \sigma_{\dot{n}}\right]^T \tag{4.20}$$

The process models for IMU errors and wind velocity components are linear, and therefore, the noise vectors are similar to Equation (4.11) and (4.9), respectively.

The measurement vector consists of IMU measurements and position measurements from a GNSS receiver.

$$Z = \left[Z_{imu}^T, Z_{gnss}^T\right]^T \tag{4.21}$$

To update the predicted state and covariance matrix, an EKF uses a linearised observation model given by:

$$Z = HX + r \tag{4.22}$$

where r is the measurement noise vector with covariance R. The measurement covariance matrices are obtained from the simulated error characteristics presented in the next section. The observation matrix, H, is given by:

$$\begin{aligned} H = \frac{\partial Z}{\partial X} &= \frac{\partial\left[Z_{imu}^T, Z_{gnss}^T\right]^T}{\partial\left[X_n^T, X_e^T, X_w^T, X_p^T\right]^T} \\ &= \begin{bmatrix} H_{imu,n} & H_{imu,e} & H_{imu,w} & H_{imu,p} \\ H_{gnss,n} & H_{gnss,e} & H_{gnss,w} & H_{gnss,p} \end{bmatrix} \end{aligned} \tag{4.23}$$

4.4.3. Simulation Setup

The trajectory used in the simulation is presented in Figure 4.18. It partly captures what would be experienced in a typical mapping or surveying mission. The trajectory consists of five GNSS outage segments of different intervals (10 s, 20 s, 30 s, 60 s, and 90 s, respectively). The outages are induced when the aircraft turns, followed by a reacquisition period upon completion of the turn. The entire flight lasted 780 seconds. Three scenarios are investigated with the aircraft roll rate limited to 15 deg/s, 30 deg/s and 60 deg/s. The initial 250 seconds of the flight are used for convergence and are not included in the discussion.

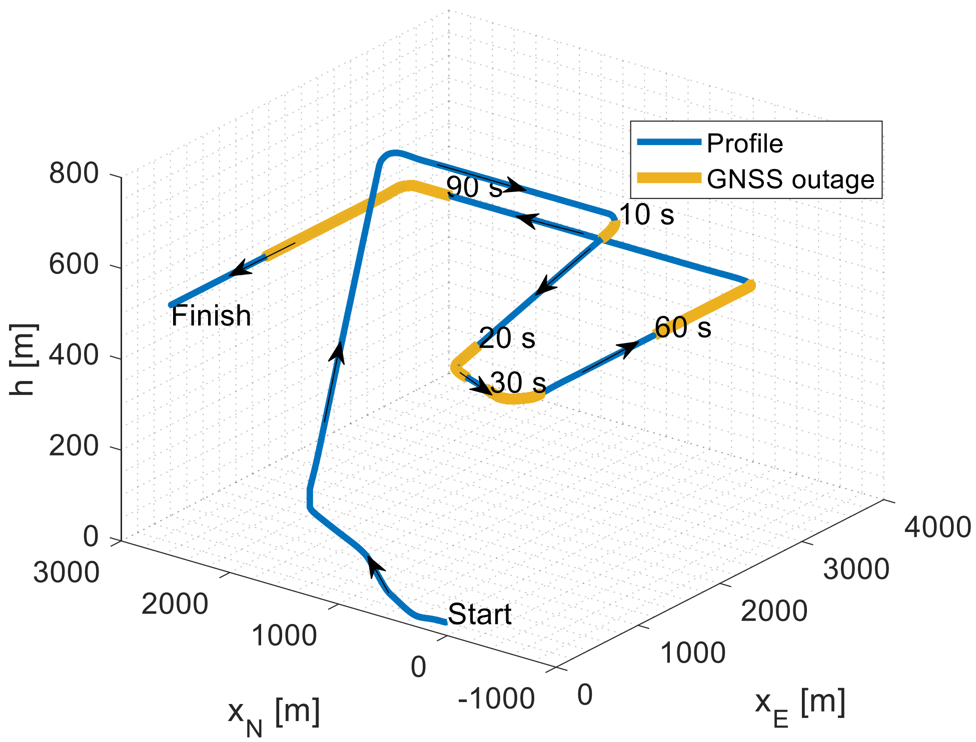


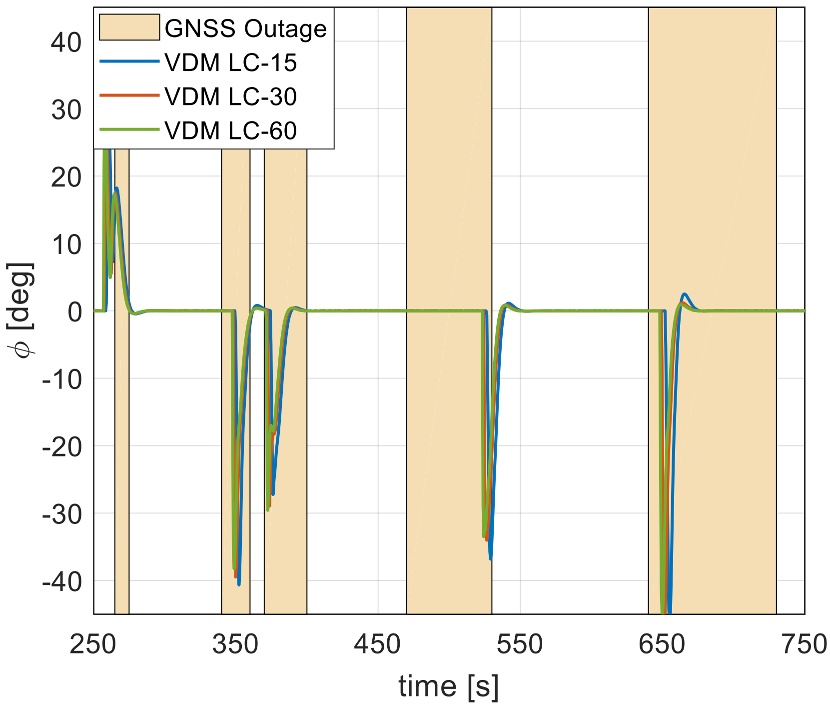
Figure 4.18. Trajectory used in the Monte Carlo simulation study.

Figure 4.19 shows the GNSS outage and reacquisition segments as well as the roll rates and roll angles achieved during the three runs. The outage period is set to 10 s, 20 s, 30 s, 60 s, and 90 s, respectively, during each simulation run. In the figure, VDM LC-15 represents the simulation run with a 15 deg/s rate limit; VDM LC-30 represents the simulation run with a 30 deg/s rate limit; VDM LC-60 represents the simulation run with a 60 deg/s rate limit.

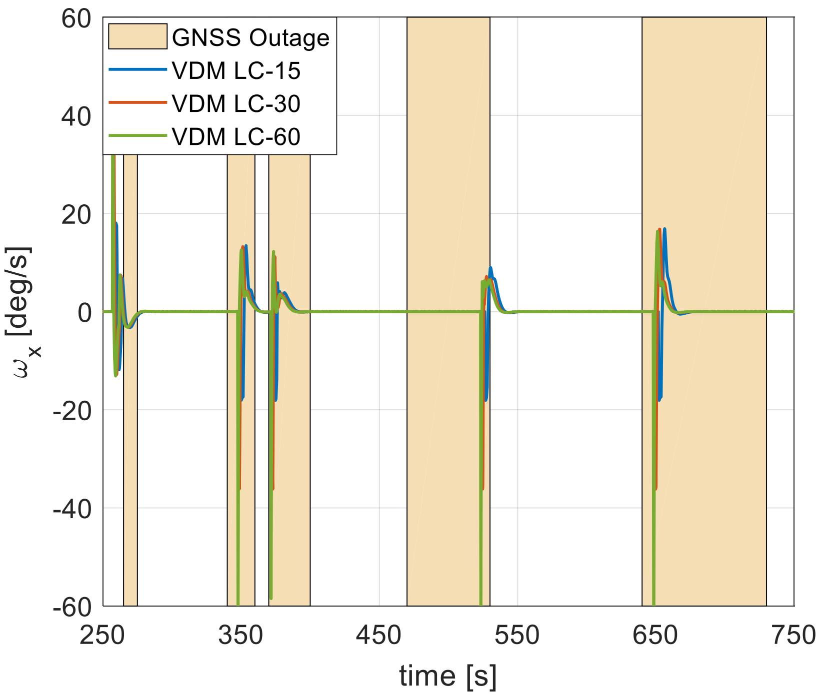
The stochastic properties of the IMU and GNSS receiver considered in the simulation are presented in Table 4.5. This setup assumes the use of a low-grade GNSS receiver. It is also assumed that the navigation solution is not used for georeferencing in the mapping or surveying mission.

Table 4.5. Stochastic properties for IMU and GNSS receiver.

Sensor	Type	Value
Accelerometer	Random bias (σ)	10 mg
	White noise (PSD)	100 $\mu g/\sqrt{Hz}$
	GM-Process	0.05 mg
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
Gyroscope	Random bias (σ)	1000 °/hr
	White noise (PSD)	21.6 $°/hr/\sqrt{Hz}$
	GM-Process	20 °/hr
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
GNSS Receiver	White noise (σ)	5 m
	Sampling Frequency	1 Hz



(a) Roll angle



(b) Roll rate

Figure 4.19. Dynamics in terms of roll angle (a) and roll rate (b) for the three simulation runs.

The reported position accuracy in some low-cost receivers is around 2.5 m (Circular Error Probable - CEP) to 3.5 m (Spherical Error Probable - SEP) using static data collected over 24 hours (u-blox, 2020). Further, GNSS positioning error analysis shows that the positioning error contains correlated noise components other than just simply white noise (Niu *et al.*, 2014; Konrad *et al.*, 2017). Dynamic performance results reported in Lim *et al.* (2019) show that the 3D position error of a low-grade GNSS receiver can reach 4 m in stand-alone positioning when mounted on a quadcopter flying a modest trajectory with four waypoints.

One hundred Monte Carlo runs are used to evaluate the autonomous navigation performance and investigate the system's robustness against random initialisation and sensor errors. The values of the standard deviation of the initial errors and the process noise covariance matrix values are the same as the ones presented in Section 4.3.3.

4.4.4. Results

Results from the simulation study are presented in this section for the VDM-based approach, and some comparison is made to a standard INS/GNSS approach described in Section E.1 of Appendix E.

Position and velocity estimation performance results are presented in Figure 4.20.

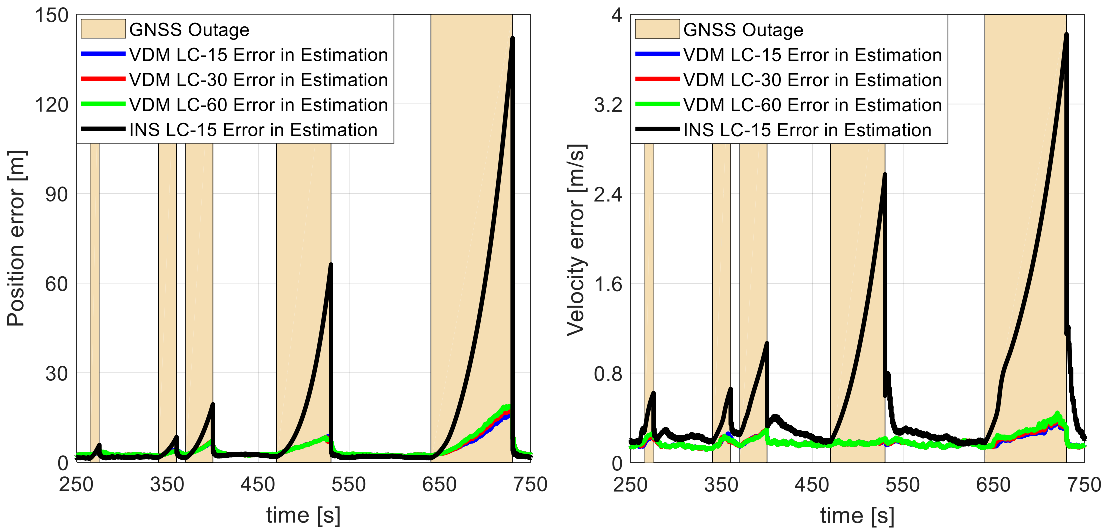


Figure 4.20. RMS of 3D position errors (left) and velocity magnitude errors (right) for the VDM vs INS approach.

For GNSS outages lasting up to 60 seconds (fourth outage), the results showed that position and velocity errors for the VDM-based approach with different rotation rates were similar. The position error reached 8.5 m for the different rate cases, well within the 2σ of the GNSS receiver as opposed to 61 m for the INS/GNSS integration architecture. Large rotation rates seemed to mostly influence the position and velocity errors for a GNSS outage lasting 90 seconds (last outage phase). For the VDM-based approach with a roll rate limit of 60 deg/s, the maximum position error was 16 % greater than that observed with a rate limit of 15 deg/s. The maximum position error for the INS/GNSS approach reached 142 m. This was greater than the maximum position error for the VDM-based approach by more than a factor of 7. Similarly, the maximum velocity magnitude error for the INS/GNSS approach during the last GNSS outage reached 3.8 m/s, which was greater than the error for the VDM-based approach (with a rate limit of 60 deg/s) by more than a factor of 8. It is important to note that the use of the

VDM allowed the velocity error to quickly converge to a lower RMS of estimation error as opposed to the INS/GNSS scheme after each outage.

The attitude estimation errors are presented in Figure 4.21.

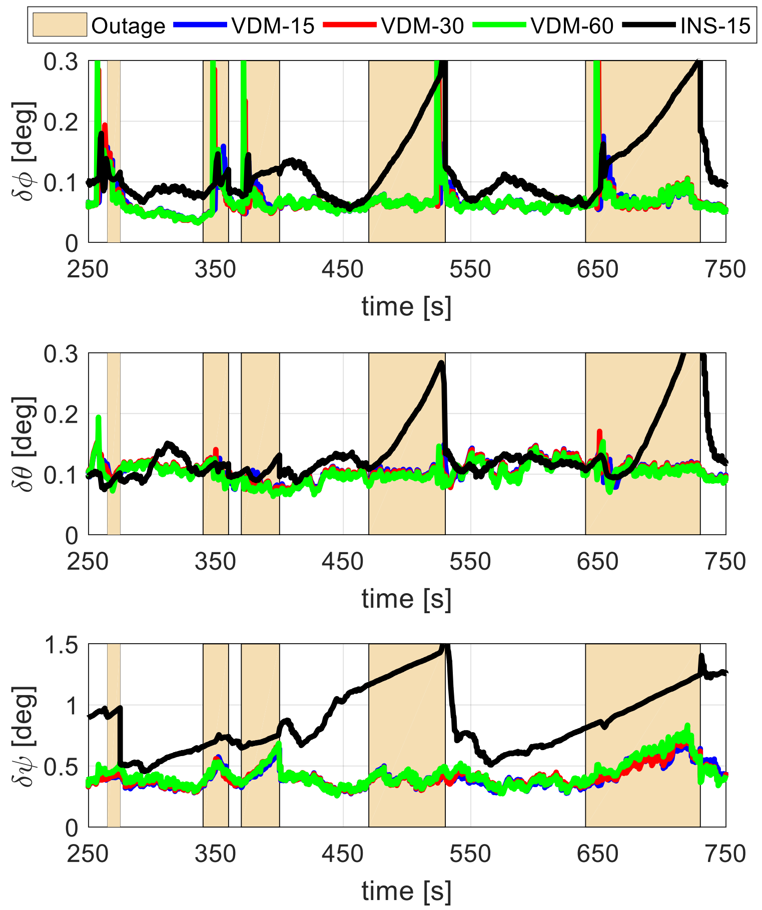


Figure 4.21. RMS of attitude estimation errors for the VDM vs INS approach.

For different GNSS outage intervals, the rotation rate limit did not significantly influence the roll and pitch angle errors for the VDM-based approach. Yaw angle error was found to increase only when the aircraft turned. During the fourth outage (470 seconds – 530 seconds), the aircraft experienced large rotation rates during the last phase of the outage, leading to a maximum error of 0.5 degrees. Between 470 seconds and 520 seconds, the aircraft was flying mostly straight and level, causing only a slight growth in yaw angle error, as shown in Figure 4.21. During a turn, the maximum rotation rate achieved was found to influence mostly roll angle errors and slightly pitch angle errors reaching a maximum of 0.46 degrees (648 seconds) with a rate limit of 60 deg/s. With the VDM approach, the roll angle error quickly recovered after short periods of rapid roll dynamics. The large increase in roll angle error during sections with large rotation rates occurred even with GNSS availability, as can be seen in the interval between 250 seconds and 265 seconds. The large instantaneous error, correlated with the rotation rate, is mainly attributed to the remaining part of the initialisation errors, especially in the VDM parameters (see sensitivity analysis in Section B.2 of Appendix B). The use of the VDM prevented further growth of the attitude errors following rapid dynamics even in periods of extended GNSS outage lasting 90 seconds. On the other hand, with an INS/GNSS approach, the attitude errors increased rapidly, with the maximum error observed being correlated with the length of the outage period.

The accelerometer bias estimation errors are presented in Figure 4.22. After the filter converged, the accelerometer bias estimation errors did not grow during the GNSS outages for both the model-based approach and the INS/GNSS integration scheme. The simple random walk model used to estimate the accelerometer biases provided good results enabling good navigation performance for the flight lasting 780 seconds with short periods of GNSS outage in between. During an outage, large roll rotation rates did not seem to influence the estimation performance of the accelerometer biases following the convergence of the filter.

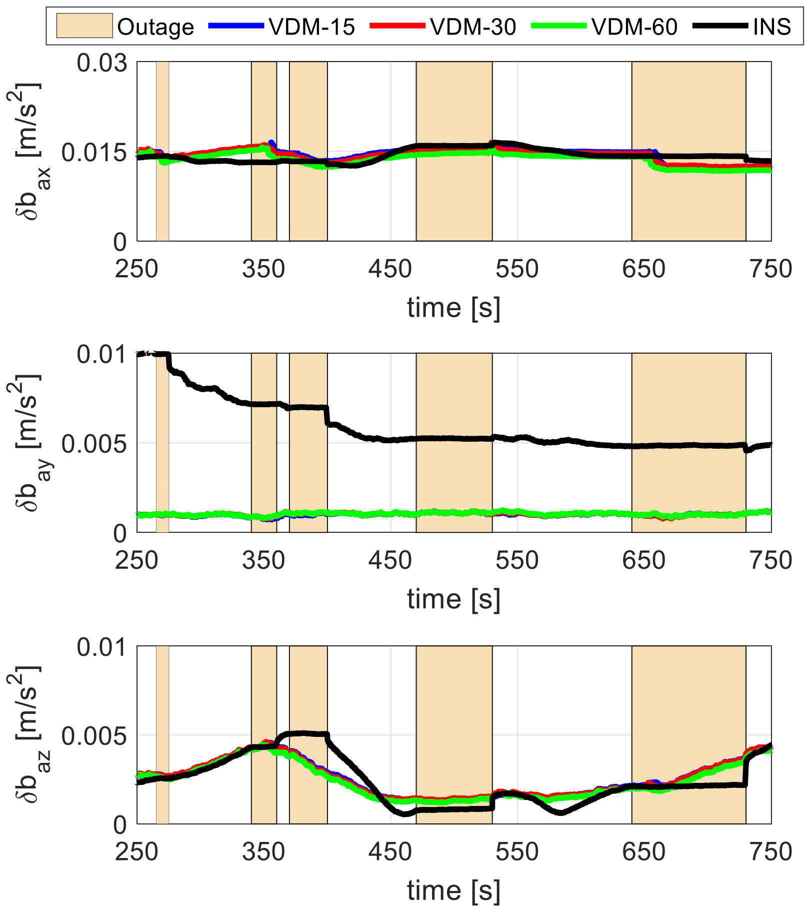


Figure 4.22. RMS of accelerometer bias estimation errors for the VDM approach and INS/GNSS architecture.

It is also noted that the use of the VDM allowed all the accelerometer biases to be easily observable as opposed to the INS/GNSS approach. The bias on the y-axis for the INS/GNSS approach was not fully converged by 250 s, which can be attributed to the correlation with the attitude states.

The gyroscope bias estimation errors are presented in Figure 4.23. Ninety-five percent of the initial gyroscope bias estimation errors are resolved well within 100 seconds of GNSS availability. Like the accelerometer biases, GNSS outages of different lengths, 10 s, 20 s, 30 s, 60 s and 90 s, did not seem to influence the estimation error of the gyroscope bias owing to the use of the VDM and direct IMU measurements during the outage. Further, large roll rotation rates reaching 60 deg/s during the GNSS outages did not seem to influence the estimation performance of the errors in different gyroscope measurements. In contrast, the INS/GNSS integration approach seemed to have lower RMS estimation errors even though these seemed to increase during the outages. The relatively larger estimation errors for the VDM-based approach as opposed to the INS/GNSS approach might be attributed to the lack of enough manoeuvres to excite different modes to make all the VDM parameters observable and, in turn, improve the observability of the IMU error terms.

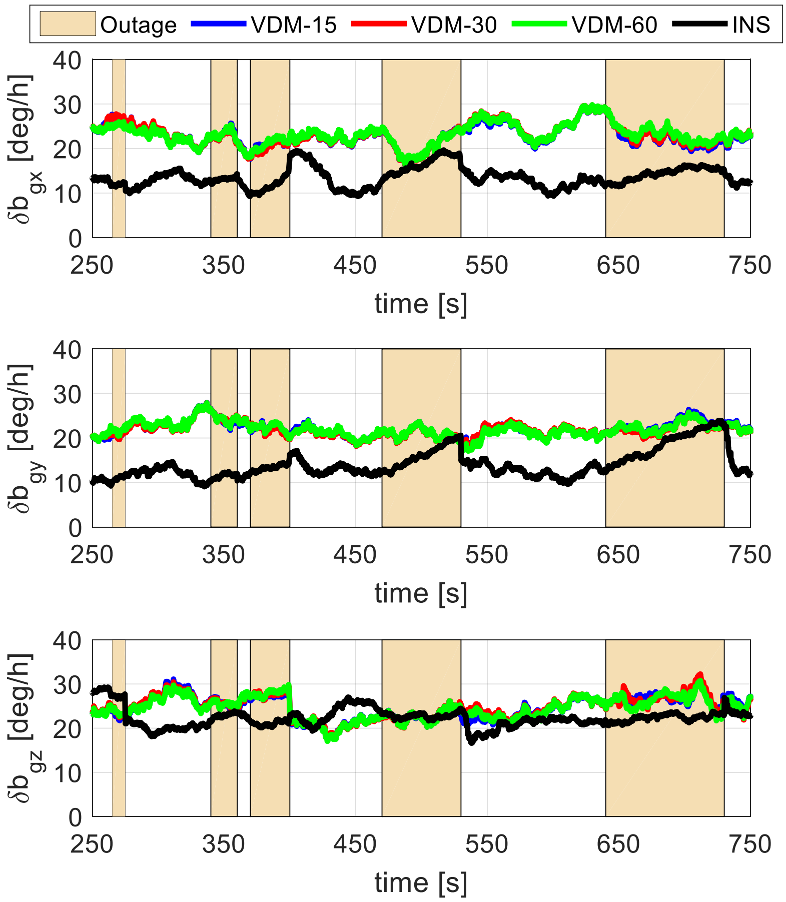


Figure 4.23. RMS of gyroscope bias estimation errors for the VDM approach.

Figure 4.24 shows the RMS of wind magnitude errors. For the VDM-based approach, the ability to estimate wind velocity improved the navigation solution during GNSS outages as opposed to an INS/GNSS integration architecture. It was found that GNSS outages lasting less than 60 seconds did not significantly influence wind estimation performance. With a GNSS outage lasting 90 seconds (fifth outage), the error in the estimated wind magnitude was found to grow gradually, reaching only 0.2 m/s at the end of the fifth outage. The level of dynamics, especially in the roll axis, did not seem to influence the estimation of wind magnitude errors.

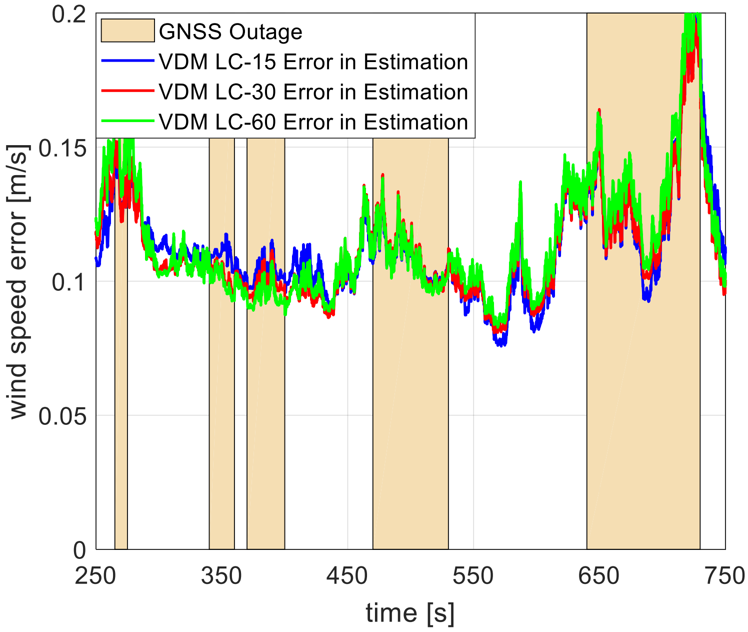


Figure 4.24. RMS of wind speed estimation errors for the VDM approach.

The RMS of the mean error of twenty-two VDM parameters is presented in Figure 4.25. A GNSS outage alone lasting up to 90 seconds did not seem to influence the VDM parameter estimation performance. However, turning during a GNSS outage led to improved observability of the VDM parameters thanks to the availability of IMU measurements during this period. Further, a rotation rate of 15 deg/s in the roll axis led to slightly better observability of VDM parameters as opposed to 60 deg/s, but the difference was less than 1%.

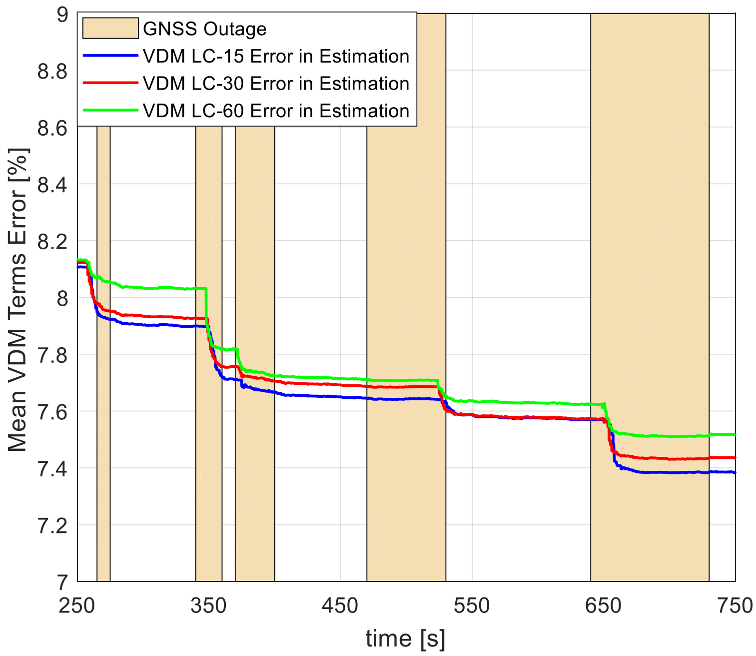


Figure 4.25. RMS of the mean error of VDM parameters.

4.5. Summary

This chapter presented and analysed state-of-the-art model-based navigation architectures for fixed-wing UAVs.

An improved model-based integration architecture was presented and analysed. The proposed concept uses a VDM as the main process model to propagate a navigation solution whilst IMU and GNSS measurements were fused in a navigation filter to estimate corrections for the states. The architecture utilised a UKF as the navigation filter and included the mechanism for estimating the mass moment of inertia. The performance of the proposed concept was compared to the state-of-the-art model-based architecture that utilises an EKF as the navigation filter but does not include the mechanism for mass moment of inertia estimation. A Monte Carlo simulation study was used to evaluate the performance of the proposed scheme assuming the use of a fixed-wing UAV fitted with low-cost MEMS-grade IMU. The proposed concept is referred to as the UKF/VDM integration architecture, and the state-of-the-art model-based scheme is referred to as the EKF/VDM integration architecture.

It was found that the approach consistently and efficiently estimated the navigation states whilst also estimating model parameters. Further, it was found that the filter was able to estimate the mass moment of inertia terms even with an initial uncertainty of 10% in the nominal values. The filter was also able to consistently estimate wind velocity without additional sensors. This improved the navigation solution, especially during a GNSS outage, where the position error in all directions was less than 14.5 m.

Other important conclusions include:

- Errors (10%-1σ of the true values) in the moment of inertia terms did not significantly influence the estimation performance of the navigation states.
- The choice of the navigation filter, either a UKF or EKF, did not significantly influence the estimation performance of the navigation states.
- Errors in the moment of inertia terms strongly influenced the estimation performance of the torque coefficients due to the significant correlation between them. However, the estimation of inertia terms reduced the errors in the torque coefficients.
- Sharp turns with rapid speed changes at constant altitude improved the estimation performance of the lateral force coefficient (CF_{Y1}), rolling moment coefficients ($CM_{X\ldots}$) and the yawing moment coefficients ($CM_{Z\ldots}$). Changes in altitude with associated compound manoeuvres improved the estimation performance of the pitching moment coefficients (CM_Y) and the lift force coefficients (CF_Z). Further, the estimation of thrust coefficients seemed to improve when the aircraft manoeuvred and changed altitude.
- Little and in some instances, no improvement in VDM parameter estimation was noticed in the absence of manoeuvres even when the aircraft experienced a significant change in speed and altitude (FC2). This was the case even during a GNSS outage.

Throughout the investigation, it was assumed that a GNSS outage occurred only once. In a typical scenario, the outage interval could vary, and the GNSS receiver could reacquire the lost satellites. Therefore, the chapter then investigated the characteristics of the navigation solution errors during different GNSS outage intervals. The chapter evaluated a model-based integration architecture during different GNSS outage intervals via a Monte Carlo simulation study. Five different outage intervals were considered: 10 s, 20 s, 30 s, 60 s, and 90 s. Each outage interval was followed by a reacquisition phase where position measurements were available to the navigation filter. The error characteristics were evaluated at different rotation rate limits imposed on the aircraft ranging from 15 deg/s, 30 deg/s and 60 deg/s. The performance was compared to a standard INS/GNSS integration architecture.

Results showed that position error increased proportionally with roll rate for an extended GNSS outage lasting 90 seconds. Attitude errors were not significantly influenced by GNSS outages lasting up to 60 seconds, with extended outages (90 seconds) mostly influencing yaw error. Further, it was found that the VDM parameters continued to be observable even during a GNSS outage provided the aircraft manoeuvres during this period. Also, it was found that the level of dynamics in the roll axis did not significantly influence the growth of wind magnitude errors.

The presented architecture has shown superior navigation performance with varying roll rates as opposed to an INS/GNSS approach operating with a modest rate of 15 deg/s. The approach has the potential to work alongside a conventional INS/GNSS integration architecture, especially in applications where the aircraft could experience rapid dynamics or GNSS interference causing GNSS outages.

Current model-based and even model-aided integration architectures rely on using position and velocity measurements from a GNSS receiver. This chapter has shown that a VDM-based integration architecture will experience drift in the navigation solution during a GNSS outage. This raises an interesting question and is the main motivation behind the work carried out during the research. This question reads:

Can the navigation performance of a VDM-based navigation scheme be improved using the available raw GNSS observables (pseudoranges and Doppler frequencies) when tracking less than four satellites?

This question will be addressed in the following chapter.

5 Tightly Coupled VDM

5.1. Introduction

This chapter presents a novel, tightly coupled integration architecture for use in fixed-wing UAVs.

Different VDM integration schemes available in the literature were discussed in the previous chapters, and the state-of-the-art model-based scheme was evaluated. The available schemes were found to rely on the position and velocity measurements output by a GNSS receiver to provide a bounded navigation solution in a so-called loosely coupled integration architecture. These measurements from a GNSS receiver are usually not available during a GNSS outage (when the receiver is tracking less than four satellites), as shown in Figure 5.1. This can cause the navigation solution to drift even when using a VDM, as shown in the previous chapter.

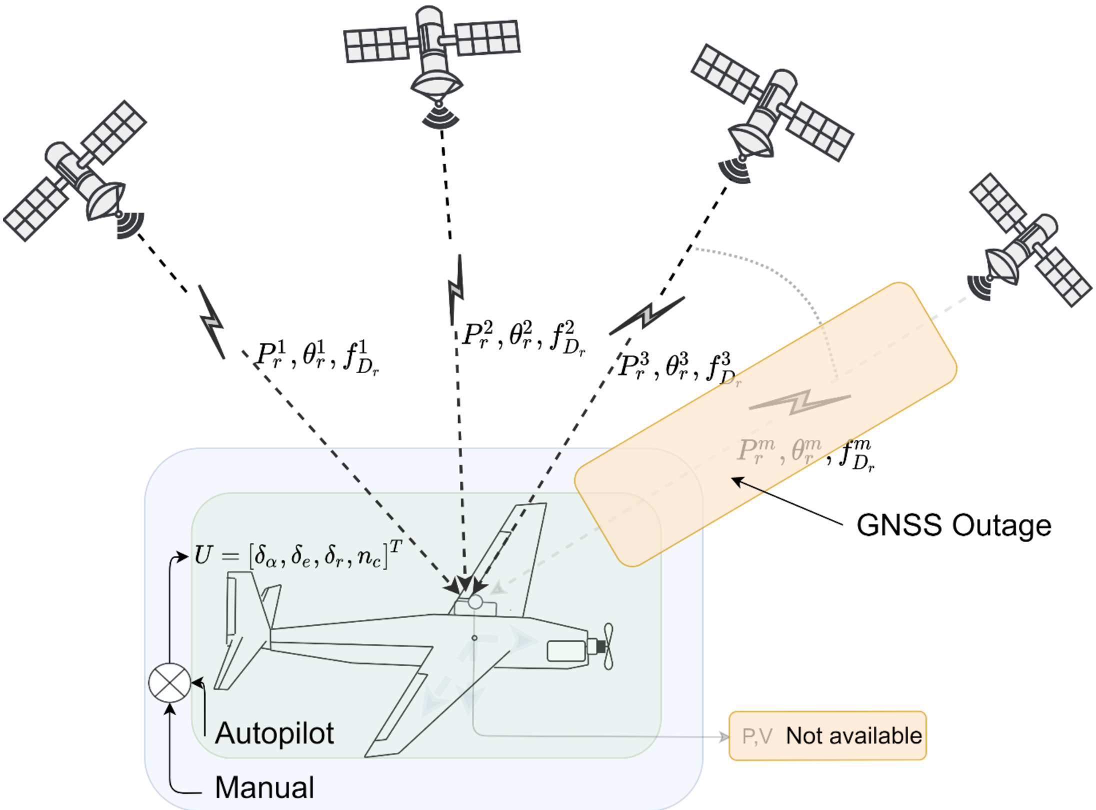


Figure 5.1. A GNSS outage scenario.

A loosely coupled INS/GNSS integration architecture is popular in most low-cost UAV applications due to its simple implementation and relatively low computational load. In low-cost applications, the quality of the inertial sensors used is relatively low. In case of a GNSS outage, the noise in these sensors will cause rapid drift in the navigation solution in a short time, and in case of an IMU failure, the navigation solution can be disabled altogether. Further, the use of direct IMU measurements to drive the navigation solution can become unreliable in the presence of significant thermal loading unless thermal models are used to eliminate stochastic variations with temperature (El-Diasty and Pagiatakis, 2009). Accurate IMU error modelling requires considerable time and effort and, in some cases, special equipment. Moreover, INS-dependent solutions are generally affected by secondary effects such as coning and sculling as a result of vibrations on the host platform, which, if not compensated, can cause significant drift in the navigation solution (Groves, 2008; Vissière *et al.*, 2008; Vasconcelos *et al.*, 2010). A model-based solution is unaffected by the platform's vibrations making the architecture considerably robust. Multi-process models, in which the final navigation solution depends on the INS, have similar limitations to INS-based integration architectures.

This chapter proposes an integration architecture that overcomes the highlighted limitations and provides an alternative integration architecture suitable for low-cost UAV applications. The chapter is organised as follows: Section 5.2 presents the proposed concept. Section 5.3 presents the filtering methodology. Section 5.4 presents the GNSS measurement simulator used to derive raw GNSS observables. Section 5.5 presents the IMU model used in the simulation study. The simulation setup is presented in Section 5.6, and simulation results in Section 5.7. A summary is presented in Section 5.8.

The work presented in this chapter has been published in Mwenegoha *et al.* (2019) and Mwenegoha *et al.* (2020).

5.2. Proposed Concept

An innovative, tightly coupled vehicle dynamic model-based integration architecture (TCVDM) capable of taking full advantage of available raw observables (pseudoranges and Doppler frequencies) from a GNSS receiver is presented and analysed to address the challenges discussed. The architecture uses measurements from a low-cost MEMS-grade IMU alongside raw GNSS observables fused using an EKF to aid the navigation solution. The state vector includes navigation states, IMU error terms, wind velocity components, VDM parameters and the receiver clock bias and drift terms.

Figure 5.2 shows the proposed architecture. Control inputs, which include the control surface deflections and the commanded propeller speed, are used to propagate the navigation states using the rigid body equations of motion for a fixed-wing UAV. An additional input to the VDM is the wind velocity vector. Most fixed-wing UAVs are equipped with an air data system, but the proposed architecture makes it possible to estimate wind velocity components within the navigation filter itself. Since the VDM is used as the main process model, no additional sensors other than an IMU and a GNSS receiver are required. An IMU, unlike a VDM, is usually affected by platform vibrations and thermal effects. On the other hand, GNSS signals can experience severe external interference, or a receiver can be affected by platform dynamics leading to a GNSS outage. The use of a VDM as the main process model ensures a continuous navigation solution regardless of the underlying conditions unless there is a hardware failure of the navigation system. The state vector is augmented to include IMU errors and GNSS receiver clock errors so that they can be estimated in the navigation filter. As previously mentioned (in Section 2.4 of Chapter 2), a VDM requires careful consideration of its structure because it depends on the host platform type. Therefore, having an accurate model or a set of model parameters is essential for successful VDM-based navigation. The proposed approach allows for the online estimation of these parameters. This significantly reduces the effort required in system identification and allows for some variation of the model parameters. This is essential because it would allow changing some aspects of the aircraft, such as the payload or the propeller, without a new system identification routine.

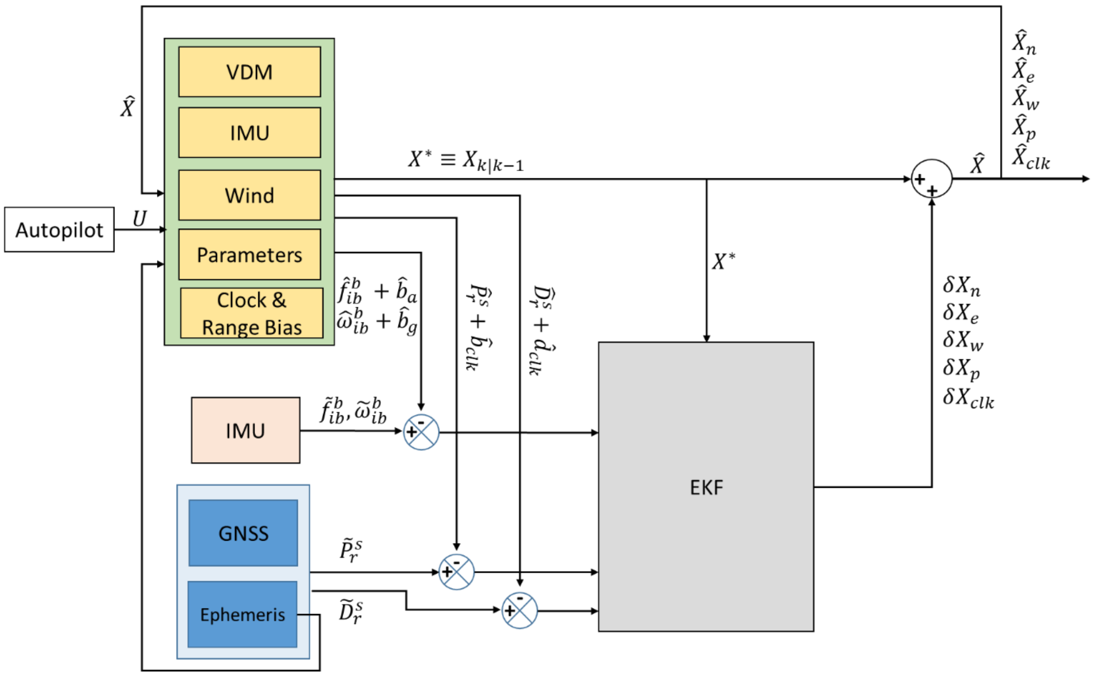


Figure 5.2. Tightly coupled VDM-based integration architecture.

5.3. Filtering Methodology

An EKF is used to estimate the corrections to the navigation states using measurements from an IMU and a GNSS receiver. The filter has a distinctive predictor-corrector structure that is summarised in Table 5.1.

An EKF requires the linearisation of the process model ($F = \partial \dot{X} / \partial X$) and observation model ($H = \partial Z / \partial X$) to determine the appropriate jacobians used in state propagation and update. The following sections will present the process and observation models used.

Table 5.1. EKF propagation and update

Propagation	Update
$[\hat{x}_{k-1\|k-1}, P_{k-1\|k-1}]$	$\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k\left(z_k - h(\hat{x}_{k\|k-1})\right)$
$\hat{x}_{k\|k-1} = \hat{x}_{k-1\|k-1} + \int_{t-\tau_s}^{t} f(\hat{x}, u_d, t')dt'$	$P_{k\|k} = (I - K_k H_k)P_{k\|k-1}(I - K_k H_k)^T + K_k R_k K_k^T$
$P_{k\|k-1} = \Phi_{k-1}\, P_{k-1\|k-1}\Phi_{k-1}^T + Q_{k-1}$	
$\Phi_{k-1} = \sum_{m=0}^{\infty} \frac{F_{k-1}^m \tau_s^m}{m!}$	$K = \frac{P_{k\|k-1}H_k^T}{H_k P_{k\|k-1} H_k^T + R_k}$
$\Phi_{\mathrm{k-1}} \approx I + F_{k-1}\tau_s$	$H = \partial z / \partial x$

5.3.1. Process Models

The navigation states are propagated using the rigid-body equations of motion presented in Chapter 3. The navigation solution is sought in the local navigation frame. This frame has been extensively explained in Chapter 3. The navigation states include:

$$X_n = \left[\mu, \lambda, h, v_{eb,N}^n, v_{eb,E}^n, v_{eb,D}^n, q_0, q_1, q_2, q_3, \omega_x, \omega_y, \omega_z, n\right]^T \tag{5.1}$$

While the quaternion $q_b^n = [q_0, q_1, q_2, q_3]^T$ is used to represent the aircraft's attitude, the attitude error is captured by the rotation vector, ϕ^b. Once the attitude error has been estimated in the filter, it is then used to update the quaternion at each measurement update ($q_{b\ k|k}^n \leftarrow q\{\hat{\phi}_{k|k}^b\} \otimes q_{b\ k|k-1}^n$). This ensures that the expected value of the *a priori* estimate of the rotation vector is always zero (Pittelkau, 2003). In Chapter 3, the relationship between the quaternion error and the rotation vector was given by: $2\delta\dot{q} = [0; \delta\dot{\phi}^b]$. This method allows the architecture to be used in practice because it avoids the singularity exhibited by three dimensional attitude representations such as Euler angles. For the interested reader, a complete review is given by Pittelkau (2003) and Solà (2017).

A random walk process is used to model the IMU errors. A random constant process is superposed in this model by setting the initial uncertainty of the IMU errors to match the uncertainty of the turn-on-bias. The IMU error vector is given by:

$$X_e = \left[b_{ax}, b_{ay}, b_{az}, b_{gx}, b_{gy}, b_{gz}\right]^T \tag{5.2}$$

A random walk process is also used to model wind velocity components. This model captures smooth transitions in wind speed sufficiently. The wind velocity vector is given by:

$$X_w = [w_N, w_E, w_D]^T \tag{5.3}$$

In a typical flight, VDM parameters will be fixed. However, the accuracy to which all the parameters are known varies based on the estimation and system identification routine used. Therefore, a random walk process model with small driving noise is used to model the VDM parameters. The VDM parameters vector is given by:

$$X_p = \begin{bmatrix} CF_{T_1}, & CF_{T_2}, & CF_{T_3}, & CF_{X1}, & CF_{X\alpha}, & \dots \\ \dots CF_{X\alpha 2}, & CF_{X\beta 2}, & CF_{Z1}, & CF_{Z_\alpha}, & CF_{Y1}, & \dots \\ \dots CM_{X_{\delta_\alpha}}, & CM_{X_\beta}, & CM_{X_{\bar{\omega}_x}}, & CM_{X_{\bar{\omega}_z}}, & CM_{Y1}, & \dots \\ \dots CM_{Y_\alpha}, & CM_{Y_{\delta_e}}, & CM_{Y_{\bar{\omega}_y}}, & CM_{Z_{\delta_r}}, & CM_{Z_{\bar{\omega}_z}}, & \dots \\ \dots CM_{Z_\beta}, & \tau_n & & & & \end{bmatrix}^T \tag{5.4}$$

The mass moment of inertia terms are assumed to be known *a priori* and therefore not estimated in this setup. In Chapter 4, it was shown that errors in the mass moment of inertia terms (up to 10%) do not significantly influence the navigation performance of a model-based approach, especially during a GNSS outage.

A two-state random process is used to model the receiver clock errors. The receiver clock error vector is given by:

$$X_{clk} = [b_{clk}, d_{clk}]^T \tag{5.5}$$

where: b_{clk} is the receiver clock bias from the system time (m),
d_{clk} is the receiver clock drift (m/s).

A good representation of the presented process models is shown in Figure 5.3.

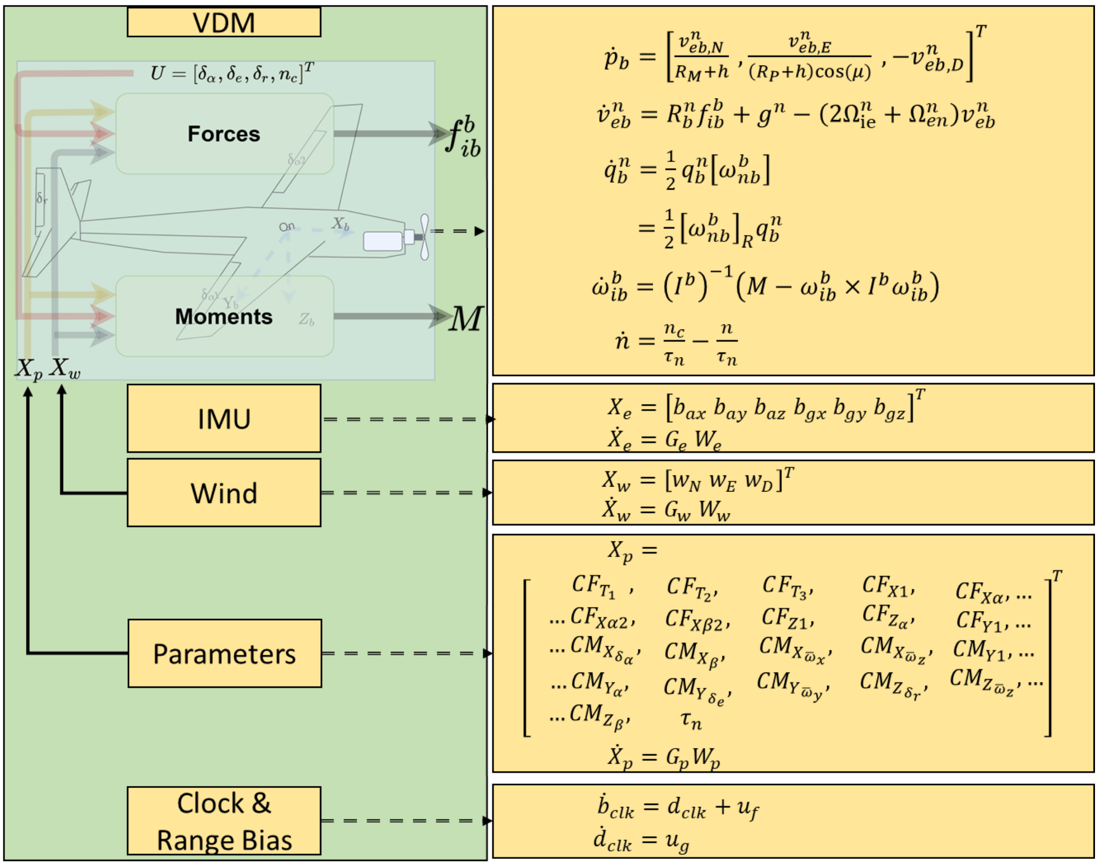


Figure 5.3. Process models for the TCVDM architecture.

The general form of the linearised process model is given by:

$$\begin{bmatrix} \dot{X}_n \\ \dot{X}_e \\ \dot{X}_w \\ \dot{X}_p \end{bmatrix} = \begin{bmatrix} F_{nn} & F_{ne} & F_{nw} & F_{np} \\ F_{en} & F_{ee} & F_{ew} & F_{ep} \\ F_{wn} & F_{we} & F_{ww} & F_{wp} \\ F_{pn} & F_{pe} & F_{pw} & F_{pp} \end{bmatrix} \begin{bmatrix} X_n \\ X_e \\ X_w \\ X_p \end{bmatrix} + \begin{bmatrix} G_n & 0 & 0 & 0 \\ 0 & G_e & 0 & 0 \\ 0 & 0 & G_w & 0 \\ 0 & 0 & 0 & G_p \end{bmatrix} \begin{bmatrix} W_n \\ W_e \\ W_w \\ W_p \end{bmatrix} \tag{5.6}$$

Based on the presented process models, the noise shaping matrices are presented in Table 5.2. It is assumed that the sub-state vectors are uncorrelated (Khaghani and Skaloud, 2016a). Therefore, the off-diagonal blocks in the G matrix are all zero. For the navigation states, uncertainty terms apply only on linear and rotational accelerations as well as the propeller rotational acceleration.

Table 5.2. Noise shaping matrices for the states in the TCVDM architecture.

Navigation states	Other states
$G_n = \begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \end{bmatrix}$	$G_e = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix}$ $G_w = \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}$ $G_p = [I]_{22x22}$ $G_{clk} = \begin{bmatrix} 1&0 \\ 0&1 \end{bmatrix}$

5.3.2. Observation Models

The measurement vector (Z_k) consists of IMU measurements $(\tilde{f}_{ib}^b, \tilde{\omega}_{ib}^b)$ and raw GNSS observables, including pseudoranges and the Doppler frequencies $(\tilde{P}_r^s, \tilde{D}_r^s)$. Here, the measurements are represented using a measurement function (h_m) such that:

$$Z_k = h_m(x_k) + w_k \tag{5.7}$$

where: x_k is the true state vector at the current time index k,
w_k is the measurement error assumed to be white noise.

Defining $E[\bullet]$ as the expectation operator, the measurement covariance R_k is given by:

$$R_k = E[w_k\, w_k^T] \tag{5.8}$$

Therefore, the observation model for the IMU is given by:

$$Z_{IMU} = \begin{bmatrix} \tilde{f}_{ib}^b \\ \tilde{\omega}_{ib}^b \end{bmatrix} = \begin{bmatrix} f_{ib}^b + X_e([1\ 2\ 3]) \\ \omega_{ib}^b + X_e([4\ 5\ 6]) \end{bmatrix} + w_i \tag{5.9}$$

where w_i is the measurement white noise vector. The measurement covariance matrix is defined using the simulated error statistics, and the details will be given in the following sections. The observation model for the GNSS observables (for each satellite) is given by:

$$Z_{GNSS}^{s} = \begin{bmatrix} \tilde{P}_r^s \\ \widetilde{D}_r^s \end{bmatrix} \tag{5.10}$$

$$\begin{bmatrix} \tilde{P}_r^s \\ \widetilde{D}_r^s \end{bmatrix} = \begin{bmatrix} \rho_r^s + X_{clk}(1) + I_r^s + T_r^s + M_P^s \\ -(\frac{f_i}{c}([v_{es}^e - v_{er}^e]^T e_r^s + X_{clk}(2) + \dot{I}_r^s + \dot{T}_r^s + \dot{M}_P^s) \end{bmatrix} + w_g$$

In Equation (5.10), I_r^s and T_r^s represent the ionosphere and troposphere propagation errors, respectively. These are partially corrected, but residual errors remain in the pseudoranges. It is also assumed that the observations contain errors due to multipath effects (M_P^s). The residual ionosphere and troposphere propagation errors as well as errors due to multipath are not estimated in the filter. It is also assumed that the satellite clock corrections have been applied. Details for the correction models are given in the next section. v_{es}^e and v_{er}^e represent the satellite and receiver velocity vectors in the ECEF frame. The geometric range from the receiver to the satellite, ρ_r^s, is given by :

$$\rho_r^s = \left||r_{es}^e - r_{er}^e|\right| + \frac{\omega_{ie}}{c}(y_{es}^e\, x_{er}^e - x_{es}^e\, y_{er}^e) \tag{5.11}$$

where: $r_{es}^e = [x_{es}^e, y_{es}^e, z_{es}^e]^T$ is the satellite position vector in the ECEF frame,
$r_{er}^e = [x_{er}^e, y_{er}^e, z_{er}^e]^T$ is the receiver position vector in the ECEF frame,
c is the speed of light in free space,
$\frac{\omega_{ie}}{c}(y_{es}^e\, x_{er}^e - x_{es}^e\, y_{er}^e)$ is called the Sagnac correction term, which accounts for the increased range as a result of the rotation of the Earth when using ECEF frame formulation.

The line-of-sight vector, e_r^s, is given by:

$$e_r^s = \frac{r_{es}^e - r_{er}^e}{\left||r_{es}^e - r_{er}^e|\right|} \tag{5.12}$$

Since the navigation solution has been formulated in the local NED coordinate frame, a transformation matrix from the NED to the ECEF frame is required. This is given by:

$$R_n^e = \begin{bmatrix} -\cos(\lambda)\sin(\mu) & -\sin(\lambda) & -\cos(\lambda)\cos(\mu) \\ -\sin(\lambda)\sin(\mu) & \cos(\lambda) & -\sin(\lambda)\cos(\mu) \\ \cos(\mu) & 0 & -\sin(\mu) \end{bmatrix} \tag{5.13}$$

This section has developed the process and observation models used in the navigation filter. In practice, the IMU measurements and raw GNSS observables (pseudoranges and Doppler frequencies) can be obtained directly from the IMU and GNSS receiver. In a simulation study, models of these sensors have to be used, as shown in Figure 5.4.

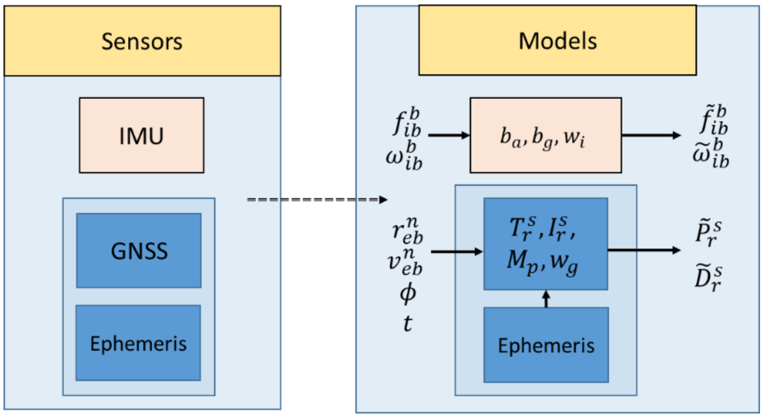


Figure 5.4. Measurements and error models for the IMU and GNSS receiver.

Therefore, the following sections will describe the modelling effort of the GNSS and IMU measurements. Emphasis will be given to the GNSS measurement simulator developed during this research.

5.4. GNSS Measurement Simulator

This section describes the development and testing of a GNSS measurement simulator used to derive raw GNSS observables. The raw observables are used to evaluate the performance of the proposed TCVDM architecture. Motivations behind the development of the simulator are described, and the limitations of the error models are explained in the ensuing.

5.4.1. Motivation

The motivations behind the development of a software-based GNSS measurement simulator are explained below:

- The use of a software-based measurement simulator allows comprehensive, repeatable, and cost-effective multi-system multi-constellation GNSS testing.
- The availability of a precise reference trajectory also makes using simulated measurements preferable. Even though this applies to both hardware and software simulators, the trajectory and measurement generation is easily customisable in a software-based measurement simulator, and the two can be directly integrated. Further, it is easy to set up a Monte Carlo simulation study by defining the error characteristics for the different models adopted in the simulator. This will make testing and evaluating a navigation scheme easy under different conditions.
- Multiple GNSS receivers can easily be simulated, which can be useful to study other applications in a controlled environment, such as GNSS attitude and relative positioning.
- Usually, GNSS hardware simulators comprise heavy and expensive units used to simulate GNSS signals arriving on a receiver. Most hardware-based simulators require expensive licenses and upgrades to keep up to date,

whilst the maintenance costs for a software-based measurement simulator are low.

- Better control of the receiver dynamics. The level of control of the receiver dynamics in a hardware simulator is fairly limited. Pinchin (2011) showed that double differenced code residuals were affected by large biases during turns or when the aircraft experienced rapid accelerations. As a result, the receiver lost lock. It was argued that the operation of the receiver tracking loops was the cause of the increased residuals to cope with the level of dynamics experienced by the receiver. Since the methods used within a receiver are proprietary, it was difficult to identify the exact cause. This makes a software-based GNSS simulator an attractive alternative to a hardware-based one since the receiver dynamics can be added or removed as desired by the user.

5.4.2. Description

This section provides a detailed description of the GNSS measurement simulator used to derive the raw observables.

Figure 5.5 shows the displayed GNSS constellation when running the measurement simulator developed to simulate raw observables (pseudorange, Doppler frequency, and carrier-phase measurements) output by a GNSS receiver.

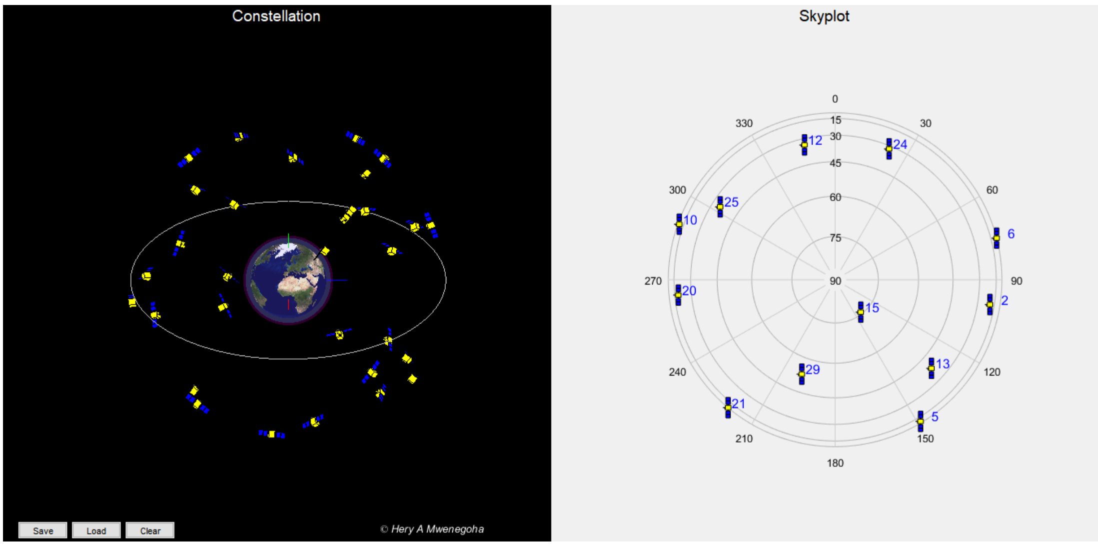


Figure 5.5. GNSS constellation displayed when running the measurement simulator.

The user trajectory is input to the simulator to generate a series of pseudorange and Doppler frequency measurements. Hourly ephemeris products archived by the Crustal Dynamics Data Information System (CDDIS) are used to derive satellite orbits and the user ephemeris structure object (Noll, 2010).

A UAV flight trajectory is generated in Simulink with the 6DOF aircraft model presented in Chapter 3. The output contains a time series of the aircraft's motion variables, including the position and velocity vectors. The trajectory is generated at 100 Hz, but the output of the GNSS measurement simulator can be adjusted to a specific output rate (for instance, 1-10 Hz). As a result, the input trajectory will be sub-sampled.

The available outputs from the GNSS measurement simulator include:

- Timestamp information – GPS week; seconds of the week (SOW); the number of leap seconds; day of the year (DOY).
- GNSS measurements – Pseudoranges; Doppler frequencies; carrier phases; carrier power to noise density ratio (C/N_0).
- Ionospheric delay information – Perturbed coefficients for the Klobuchar model used to simulate the first-order ionospheric delay; time-series of simulated ionospheric delays for each satellite in view.
- Tropospheric delay information – Time series of the tropospheric delay affecting the GNSS measurements for each satellite in view.
- Clock effects – Time series of the receiver clock offset and drift.

The outputs are packed into a Matlab structure object for each measurement epoch and for each receiver modelled. This structure object contains, within it, another structure object for each receiver channel.

The process of generating GNSS measurements is shown in Figure 5.6.

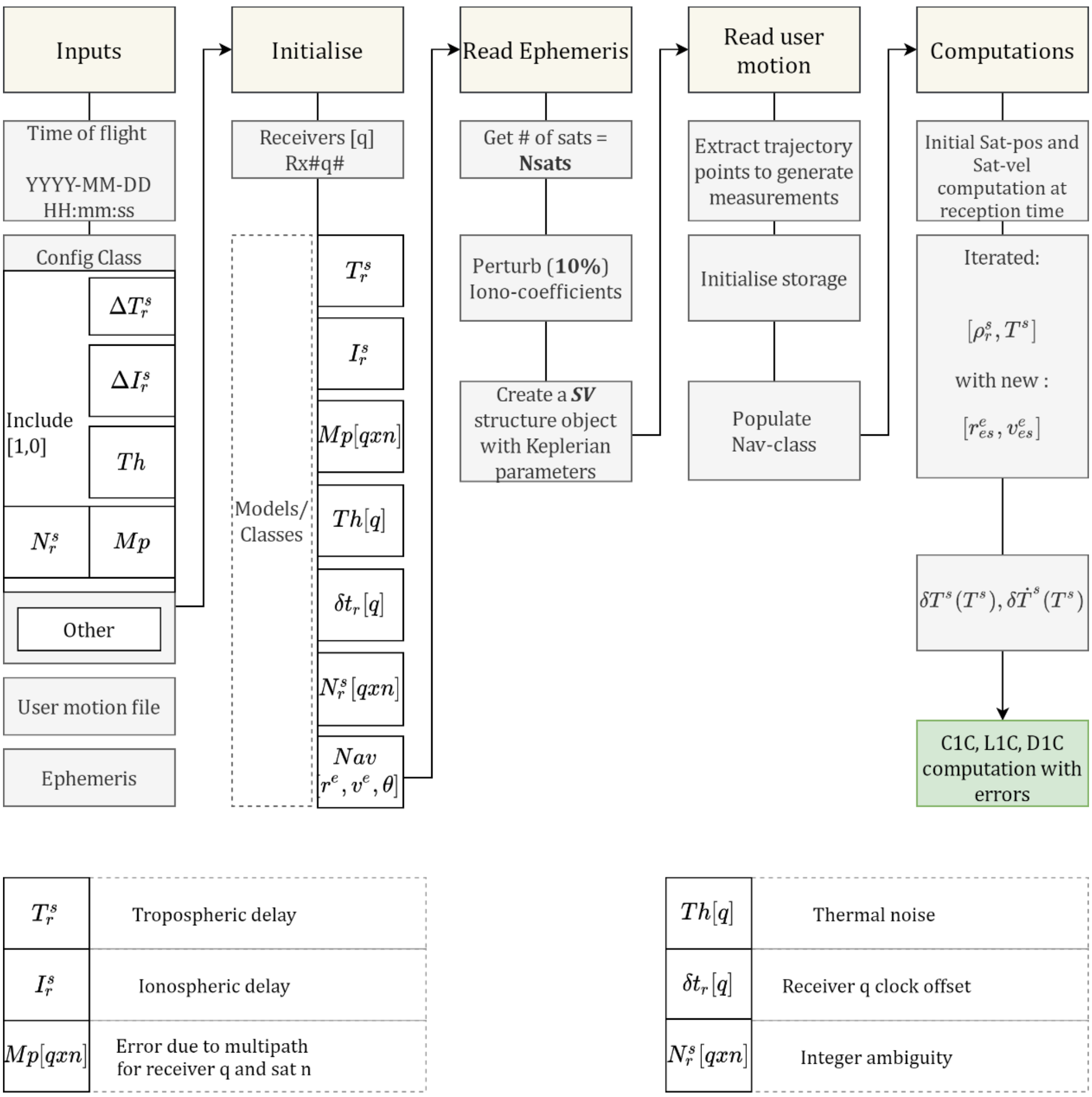


Figure 5.6. GNSS measurement simulator flow diagram. C1C, L1C, and D1C represent the computed pseudorange, carrier phase and Doppler frequency measurement.

The workflow is as follows:

- Initialise: After loading the inputs and defining the error models to include in the GNSS measurements, different classes for the delays and the ambiguities are initialised. The classes define the different errors that affect GNSS measurements.
- Read Ephemeris: A GNSS navigation message file in the RINEX (Receiver Independent Exchange) format is read, and the ionospheric coefficients are extracted. These coefficients will be used in the computation of the ionospheric delay for different measurements before being perturbed for user processing. A Matlab structure object is created for the Keplerian parameters for each space vehicle.
- Read user motion: During the initialisation phase, a navigation class is also initialised. This class contains position, velocity and attitude information for a specific receiver. Here, the navigation class is populated with information from the user motion file based on the desired output rate.
- Computations: This is the main loop. The initial satellite position and velocity vectors are computed at the time of signal reception. This will allow the computation of the initial geometric ranges, which are then used to iteratively calculate the signal transmission time alongside refining the estimates of the satellite position and velocity vectors. After this, the satellite clock offset and drift are calculated, followed by the computation of the GNSS measurements with associated errors.

In the simulator, pseudorange (P_r^s) and Doppler frequency (D_r^s) measurements are given by:

$$P_r^s = \rho_r^s + c\big(dt_r(t_r) - dT_s(T_s)\big) + I_r^s + T_r^s + M_P + \epsilon(\rho) \tag{5.14}$$

$$D_r^s = -\frac{f_i}{c}\left([v_{es}^e(T_s) - v_{er}^e(t_r)]^T e_r^s + c\left(\frac{\partial t_r(t_r)}{\partial t} - \frac{\partial T_s(T_s)}{\partial t}\right) + \dot{I}_r^s + \dot{T}_r^s\right) + \epsilon(f_D) \tag{5.15}$$

where: dt_r is the receiver clock offset at reception time in seconds (t_r),
dT_s is the satellite clock offsets at transmission time in seconds (T_s),
I_r^s is the ionospheric delay in metres,
T_r^s is the tropospheric delay in metres,
M_p is the error due to multipath in metres,
$\epsilon(\rho)$ is the random thermal noise in range measurements in metres,
$\epsilon(f_D)$ includes multipath effects and random noise in the doppler measurements in hertz.

The error sources are classified into three classes, transmission sources, propagation sources and reception sources, as shown in Figure 5.7. The transmission sources are largely due to the use of the broadcast orbit and clock products. These are loosely modelled in this research using a small random constant bias in the measurements (for each satellite $1\text{m} - 1\sigma$) (Kim and Kim, 2015). Propagation sources include effects of the ionosphere and troposphere, and these are discussed in detail in the ensuing. Reception error sources include multipath, thermal noise and the receiver clock, which are also discussed in detail in this research. Other sources of error, such as the antenna phase centre offset and variation, inter-frequency biases, and g-dependent oscillator errors are not considered in the simulation.

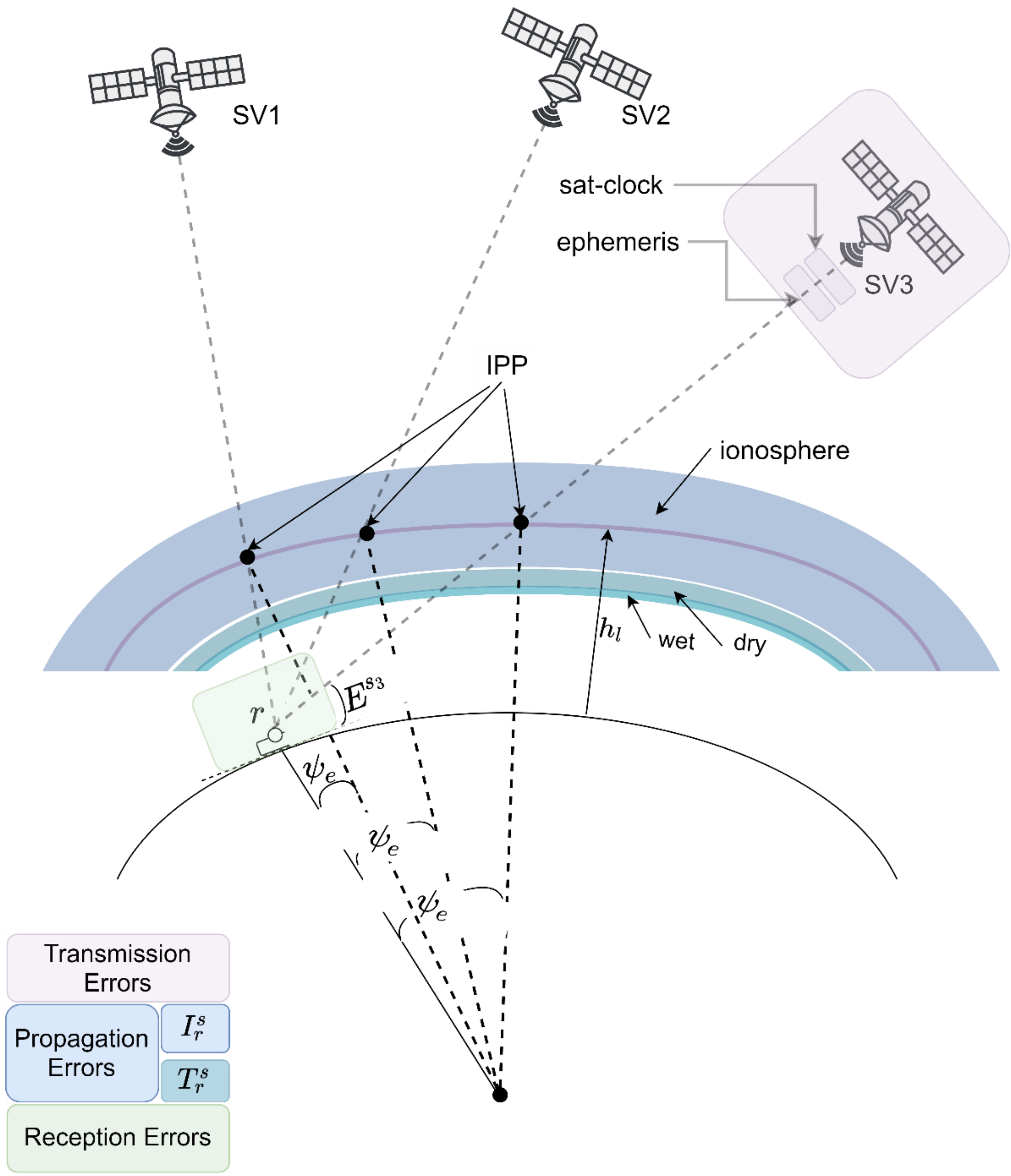


Figure 5.7. Different errors affecting raw GNSS observables considered in the simulator.

5.4.3. Ionospheric Model

The ionosphere is a dispersive medium and is a significant error source of the GNSS error budget. It is located primarily in the region of the atmosphere between 70 km and 1000 km above the Earth's surface (Kaplan and Hegarty, 2006). Sun rays ionise portions of gas molecules in this layer creating free electrons that influence the propagation of electromagnetic waves. The electron density along the path length is usually referred to as the total electron content (TEC). This can be used to estimate the ionospheric delay affecting both the carrier-phase (negative) and pseudorange measurement (positive). This is formulated as:

$$I_r^s = \pm \frac{40.3}{f^2} STEC \tag{5.16}$$

In Equation (5.16), $STEC$ is the slant total electron content as a result of multiplying the vertical TEC by a slant factor. The electron content is a function of the time of day, user location, satellite elevation angle, season, ionising flux, magnetic activity and sunspot cycle (Kaplan and Hegarty, 2006).

Since the ionosphere is a dispersive medium, a significant amount of the ionosphere induced range error can be removed through a linear combination of dual-frequency observables at the expense of increased tracking noise and multipath errors. However, single-frequency users require additional information to correct this error. For the GPS constellation, ionospheric correction parameters are transmitted as part of the navigation message to drive the ionospheric correction algorithm (ICA) based on the Klobuchar model.

The Klobuchar model is used to approximate the diurnal variation of the ionospheric delay using a cosine function with varying amplitude and period based on the user's geodetic position (Klobuchar, 1987). This model removes about 50% of the RMS ionospheric delay and assumes that the electron content is concentrated in a thin layer 350 km in height (Klobuchar, 1987; Kaplan and Hegarty, 2006). Given the approximate user position and line-of-sight (LOS) vector of the observed satellite, which also defines the elevation and azimuth angles, the delay is computed through the following process:

a. Calculation of the Earth-centred angle (in semicircles) from the elevation angle E^s.

$$\psi_e = \frac{0.0137}{E^s + 0.11} - 0.022 \tag{5.17}$$

b. Computation of the latitude of the ionospheric pierce point (IPP in semicircles) from the user's latitude μ, and the Earth-centred angle ψ_e and azimuth angle Az.

$$\mu_{IPP} = \mu + \psi_e \cos Az \tag{5.18}$$

c. Computation of the longitude of the IPP (in semicircles) from the user's longitude λ, the Earth-centred angle ψ_e, azimuth angle Az, and IPP latitude μ_{IPP}.

$$\lambda_{IPP} = \lambda + \frac{\psi_e \sin Az}{\cos \mu_{IPP}} \tag{5.19}$$

d. Calculation of the geomagnetic latitude (in semicircles) from the IPP latitude and longitude (μ_{IPP}, λ_{IPP}).

$$\mu_{MAG} = \mu_{IPP} + 0.064\cos(\lambda_{IPP} - 1.617) \tag{5.20}$$

e. Calculation of the local time (in seconds) at the IPP from the IPP longitude λ_{IPP} and the local GPS time t_{GPS}.

$$t = 43200\lambda_{IPP} + t_{GPS} \tag{5.21}$$

f. Computation of the vertical ionospheric time delay.

$$I_{v_r}^{s} = 5 \cdot 10^{-9} + \sum_{k=0}^{3} \alpha_k(\mu_{MAG})^k \left(1 - \frac{x^2}{2} + \frac{x^4}{24}\right) \tag{5.22}$$

$$x = \frac{2\pi(t - 50400)}{\sum_{k=0}^{3} \beta_k(\mu_{MAG})^k}$$

where: $\sum_{k=0}^{3} \alpha_k(\mu_{MAG})^k$ is the amplitude of the ionospheric delay in seconds,
$\sum_{k=0}^{3} \beta_k(\mu_{MAG})^k$ is the period of the ionospheric delay also in seconds,
x is the phase of the ionospheric delay in radians.

The coefficients, α_k and β_k, are part of the GPS navigation message. These parameters are perturbed with a standard deviation of 10% and then passed to the user ephemeris structure object. A vertical residual ionospheric delay (common to all satellites) modelled as a first-order Gauss-Markov (GM) process with a standard deviation of 2 m and a time constant of 1800 seconds is added to the computed vertical delay (in the ionosphere error generation).

g. Computation of the slant factor

$$F = 1 + 16(0.53 - E^s)^3 \tag{5.23}$$

h. Computation of the slant ionospheric time delay

$$I_r^s = \begin{cases} \left(I_{v_r}^{s} + GM\right) \cdot F & ; |x| \leq 1.57 \\ (5 \cdot 10^{-9} + GM) \cdot F & ; |x| \geq 1.57 \end{cases} \tag{5.24}$$

Even though only the GPS L1 frequency was considered in this research, the model can be extended to other frequencies and constellations. For a fixed user location, the ionospheric delay is plotted for different satellite elevation and azimuth angles, as shown in Figure 5.8. The ionospheric delay is calculated with a 10% variation in the correction coefficients transmitted in the GPS navigation message plus a common residual delay (Basile, Moore and Hill, 2019).

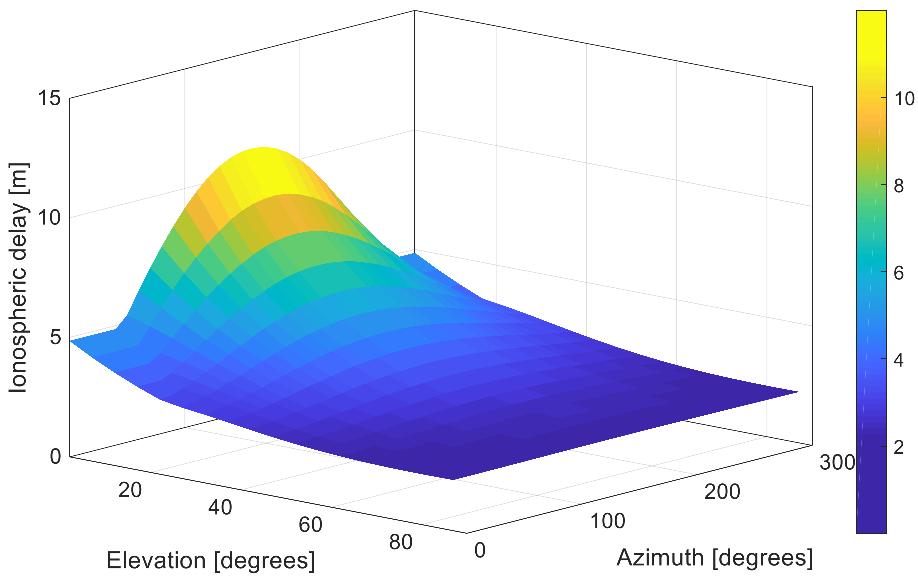


Figure 5.8. Ionospheric delay for a given user location.

As can be seen from Figure 5.8, the Klobuchar model is highly correlated and therefore cannot represent rapid and short-term variations in the ionospheric delay very well. The investigation of the impact of short-term variations of the ionospheric delay on navigation performance is beyond the scope of this research. However, the residual error introduced in the ionospheric delay is used to account for these variations. Short-term variations in the ionosphere-induced range error result from variations in the electron content above or below the daily average captured by the Klobuchar model.

Figure 5.9 shows an ensemble of ten short-term variations in the zenith ionospheric delay modelled using a first-order Gauss-Markov process.

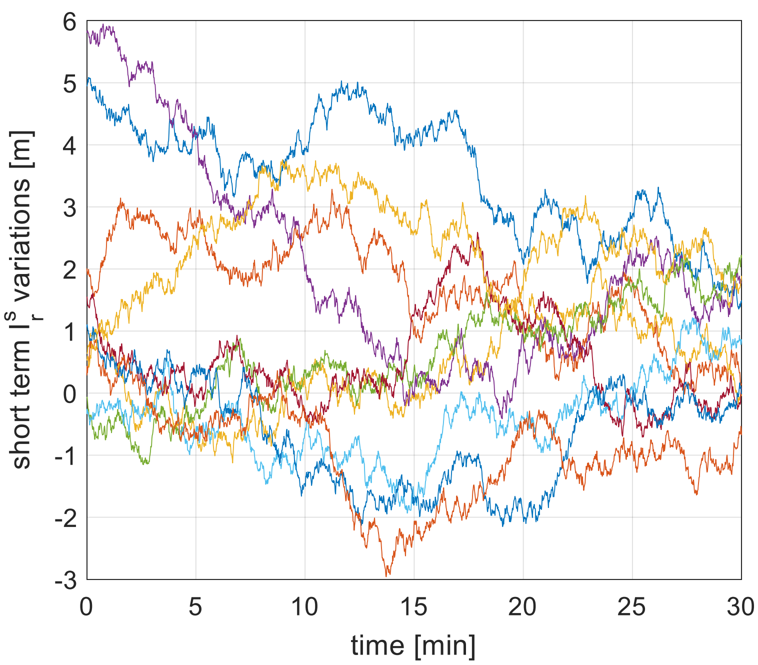


Figure 5.9. The zenith ionospheric delay due to short-term variations in the electron content above(**+**) or below (**-**) the diurnal average.

5.4.4. Tropospheric Model

The troposphere is a non-dispersive medium for frequencies up to 15 GHz (Kaplan and Hegarty, 2006). In this medium, both the carrier and signal information are equally delayed with respect to free-space propagation. The delay is a function of the refractive index, which is a function of the local temperature, pressure and relative humidity. About 90% of the delay arises from dry air and is usually referred to as the dry or hydrostatic delay (Kaplan and Hegarty, 2006). The wet component arises from water vapour and is usually difficult to predict due to uncertainties in atmospheric distribution.

Different models to estimate the tropospheric delay exist, and a complete review of all the models is beyond the scope of this research. A review of some common models is given in Kaplan and Hegarty (2006) for the interested reader. One accurate method of estimating the tropospheric delay was developed at the University of New Brunswick (LaMance, Collins and Langley, 1996). The model is referred to as the UNB3 model and provides look-up tables for average and seasonal variation of the meteorological parameters, which can be used to estimate the zenith total tropospheric delay. These meteorological parameters include pressure (P), temperature (T), water vapour pressure (e) at mean sea level, and temperature and water vapour lapse rates (β and η). These are interpolated based on the specification of the user's latitude. The UNB3 model computes the tropospheric delay using the Saastamoinen models and the Niell mapping functions. Some modifications to the UNB3 model, such as using different mapping functions, has made it favourable for other satellite-based augmentation systems such as the European Geostationary Navigation Overlay Service (EGNOS) (Penna, Dodson and Chen, 2001).

In this research, a simplified UNB3 model that utilises the Black and Eisner (1984) mapping function is used to model the troposphere. This model has been referred to as the EGNOS tropospheric correction model by Penna, Dodson and Chen (2001). The average values of the five meteorological parameters based on the user's latitude are given in Table 5.3.

Table 5.3. Average values of five meteorological parameters used in the EGNOS model.

Latitude [°]	P_0 [mbar]	T_0 [K]	e_0 [mbar]	β_0 [mK/m]	η_0
≤ 15	1013.25	299.65	26.31	6.30	2.77
30	1017.25	294.15	21.79	6.05	3.15
45	1015.75	283.15	11.66	5.58	2.57
60	1011.75	272.15	6.78	5.39	1.81
≥ 75	1013.00	263.65	4.11	4.53	1.55

The seasonal variation of the meteorological parameters is given in Table 5.4.

Table 5.4. Seasonal variation of the five meteorological parameters used in the EGNOS model.

Latitude [°]	ΔP [mbar]	ΔT [K]	Δe [mbar]	$\Delta\beta$ [mK/m]	$\Delta\eta$
≤ 15	0.00	0.00	0.00	0.00	0.00
30	-3.75	7.00	8.85	0.25	0.33
45	-2.25	11.00	7.24	0.32	0.46
60	-1.75	15.00	5.36	0.81	0.74
≥ 75	-0.50	14.50	3.39	0.62	0.30

A linear interpolation scheme is used to compute the meteorological parameters (ζ) based on the receiver's latitude. Following linear interpolation, each meteorological parameter is computed using:

$$\zeta(\mu, D) = \zeta_0(\mu) - \Delta\zeta(\mu) \times \cos\left(\frac{2\pi(D - D_{min})}{365.25}\right) \tag{5.25}$$

where: μ is the receiver's latitude,
D is the day-of-year (starting with 1st of January),
D_{min} is 28 for northern latitudes and 211 for southern latitudes,
ζ_0 and $\Delta\zeta$ represent the average and seasonal variation for each meteorological parameter at the receiver's latitude, respectively.

The zenith total delay (ZTD) is computed and mapped appropriately based on the elevation of the satellite. A residual zenith delay, modelled as a first-order Gauss-Markov process with a standard deviation of 0.2 m and a time constant of 1800 seconds (Basile *et al.*, 2018; Basile, Moore and Hill, 2019), is applied to all satellites following appropriate mapping. The tropospheric delay is given by:

$$T_r^s = (ZTD_d + ZTD_w + \epsilon_{cmn}) \times MF(E^s) \tag{5.26}$$

where: ZTD_d is the zenith dry (hydrostatic) delay,
ZTD_w is the zenith wet delay,
$MF(E^s)$ is the mapping function for a given satellite elevation angle (E^s).

The zenith dry and wet delays are computed as:

$$ZTD_d = ZTD_{d,0}\left[1 - \frac{\beta H}{T}\right]^{\frac{g}{R_d\beta}} \tag{5.27}$$

$$ZTD_w = ZTD_{w,0}\left[1 - \frac{\beta H}{T}\right]^{\frac{(\eta+1)g}{R_d\beta}-1} \tag{5.28}$$

where: g is equal to 9.80665 $\mathrm{m/s^2}$,
H is the height of the receiver above mean sea level,
R_d equals 287.054 J/kg/K,
$ZTD_{d,0}$ and $ZTD_{w,0}$ represent the zenith dry and wet delay at mean sea level.

The zenith dry and wet delays at mean sea level are computed using:

$$ZTD_{d,0} = \frac{10^{-6}k_1 R_d P}{g_m} \tag{5.29}$$

$$ZTD_{w,0} = \frac{10^{-6}k_2 R_d}{g_m(\eta + 1) - \beta R_d} \times \frac{e}{T} \tag{5.30}$$

where: k_1 is equal to 77.604 K/mbar,
k_2 equals 382000 $\mathrm{K}^2/\mathrm{mbar}$,
g_m equals 9.784 $\mathrm{m/s}^2$.

The Black and Eisner mapping function (Kaplan and Hegarty, 2006) is used to map the zenith delays based on the receiver-to-satellite elevation. This is given by:

$$MF(E^s) = \frac{1.001}{\sqrt{0.002001 + \sin^2(E^s)}}\gamma \tag{5.31}$$

$$\gamma = 1 - 0.015\max(0, 4 - E^s)^2$$

In Equation (5.31), γ is applied to adjust for elevation angles below 4°.

Figure 5.10 shows the tropospheric delay based on the EGNOS model for different satellite elevation angles and receiver altitude above mean sea level. The EGNOS model is also highly correlated, and therefore short-term variations in the tropospheric delay are not reflected. The common residual vertical delay introduces short-term variations not captured by the EGNOS model, albeit in time rather than space. An ensemble of ten realisations of this first-order process is presented in Figure 5.11.

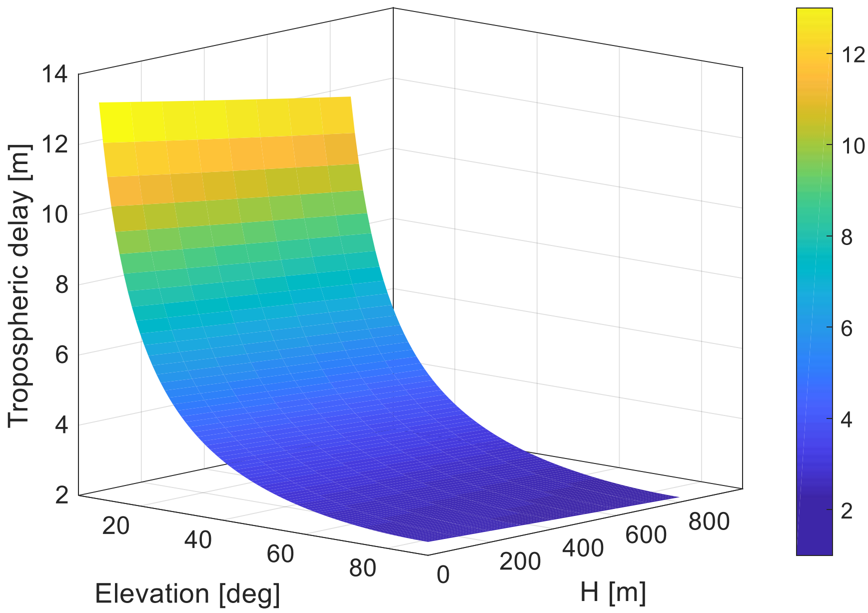


Figure 5.10. Tropospheric delay based on the EGNOS model.

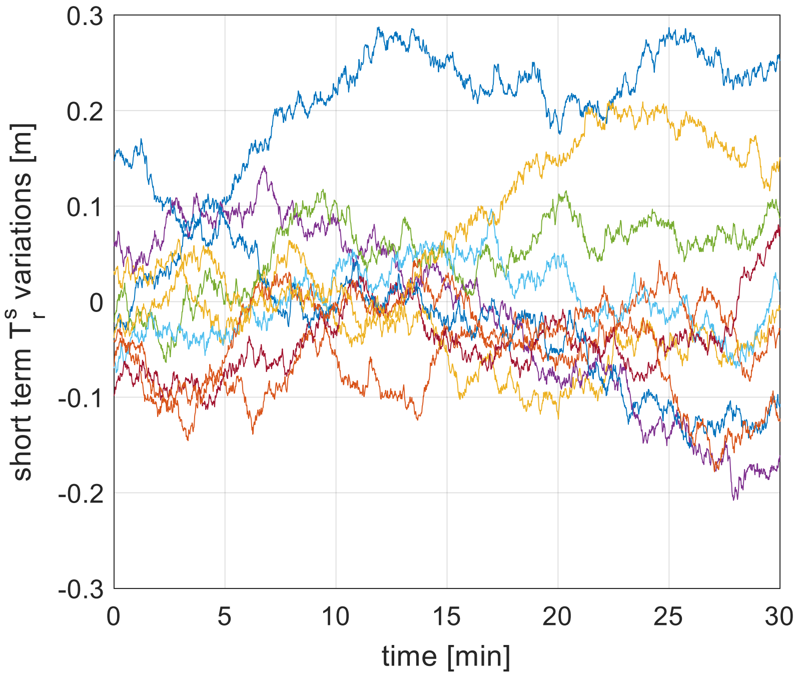


Figure 5.11. The residual tropospheric delay.

5.4.5. Multipath

Multipath interference results from the reception of signals via multiple paths, which may or may not include the direct path. The absence of the direct path is known as non-line-of-sight (NLOS) reception. Multipath errors vary significantly in magnitude based on the receiver operating environment, satellite elevation angle, antenna gain pattern, and signal characteristics.

Simsky *et al.* (2008) analysed 1.5 years of GIOVE-A satellite signals and presented the multipath performance of different signals compared to the GPS-L1 C/A. The data was recorded on a rooftop of a building using the GETR receiver, which was custom-built by Septentrio for the reception of GIOVE signals (Simsky *et al.*, 2008). The antenna was mounted on an elevated support structure which was higher than the other objects on the rooftop. The results indicated that the error due to multipath was largely dependent on the satellite elevation angle during static periods. However, during dynamic periods the difference in multipath suppression for the different signals was less pronounced (with generally smaller code multipath errors). The results have been extracted and curve-fitted. Three multipath groups are classified based on the performance of the GIOVE-A signals vis-à-vis GPS-L1 C/A signal. A high multipath group is defined based on the performance of the GPS L1 C/A signal; a medium multipath group is defined based on the E6BC signal, and a low multipath group based on the performance of the E5AltBOC. The three classes are shown in Figure 5.12. This curve fitting facilitates the extraction of coefficients that can be used to define the level of pseudorange noise resulting from multipath.

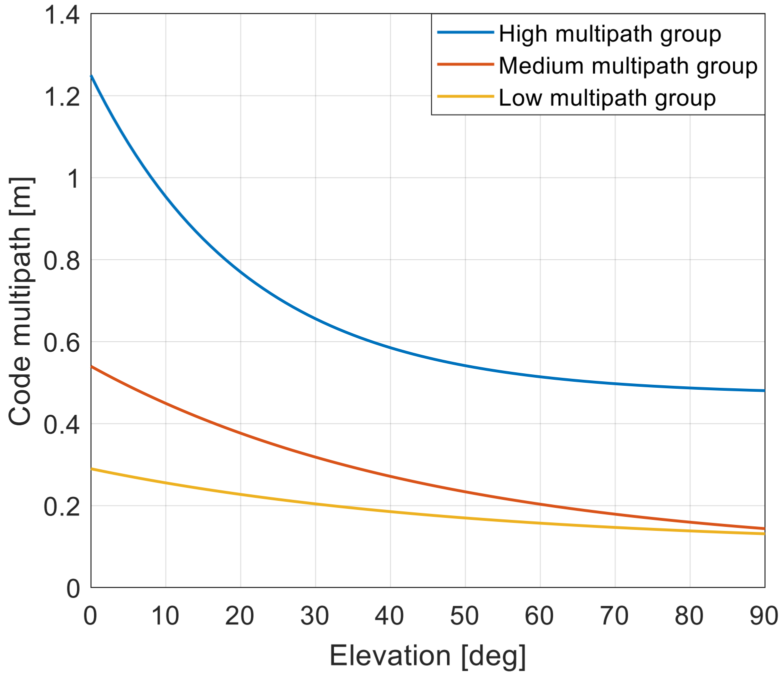


Figure 5.12. Different multipath groups based on the performance of the GIOVE-A signals tracked by a Septentrio GETR receiver.

It is not difficult to see that the error due to multipath has an exponential relationship with the satellite elevation angle. Therefore, the standard deviation of the multipath error is assumed to be dependent on the satellite elevation angle and is given by:

$$\sigma_w = c_0 + c_1 \cdot e^{\frac{E^S}{c_2}} \tag{5.32}$$

where: c_0, c_1 and c_2 represent the coefficients for the given multipath group.

Table 5.5 shows the estimated coefficients for the three multipath groups presented in Figure 5.12.

Table 5.5. Different multipath group exponential law fitting coefficients.

Coefficient	Low	Medium	High
c_0 [m]	0.10	0.08	0.47
c_1 [m]	0.19	0.46	0.78
c_2 [deg]	50	45.60	20.91

It is important to note that the tabulated results are relevant in classifying multipath in high-end receivers. Matera *et al.* (2019) characterised pseudorange multipath errors using data collected on a low-cost GNSS receiver (u-blox M8T) in an urban environment. They showed that the standard deviation for the high multipath group could be an order of magnitude higher than for a high-end receiver, especially for low-elevation satellites. However, the results from their study contained significant NLOS signals, which resulted in non-symmetric probability density functions with large means and standard deviations. In this thesis, the presence of NLOS signals is not considered because fixed-wing UAVs are generally operated in open sky environments (since they cannot hover, and therefore flying in confined spaces is difficult). However, this assumption does not hold for other types of UAVs, such as VTOL aircraft, which can be operated in different environments, including dense urban settings.

Khanafseh *et al.* (2018) analysed ionospheric-corrected code-minus-carrier data and showed that a first-order Gauss-Markov process could be used to model multipath in an automotive setting. Using dual-frequency NovAtel antennas, results for GPS L1 pseudorange indicated that multipath errors had a non-zero mean ranging from 1 cm to 11 cm depending on the environment (open sky, satellite elevation, nearby objects). Using folded cumulative distributions about the median, the standard deviation of the multipath error was found to be in the range from 75 cm to 156 cm. The time constant for the multipath error was estimated from the intercept of the autocorrelation function of the multipath error and the $exp(-1)$ line. It was found that, for a static receiver the time constant ranged from 40 seconds to 150 seconds and for a kinematic receiver it was between 2 seconds and 65 seconds. On the other hand, carrier-phase results indicated that the maximum standard deviation was 3 cm with very little difference in the mean error between satellites with high and low elevation angles. Even though the results reported by Khanafseh *et al.* (2018) are for GPS (L1, L2) and GLONASS, the trend is very similar to the results reported by Simsky *et al.* (2008).

A typical fixed-wing UAV flight will be comprised of six main segments: take-off, climb, cruise, loiter, descent (approach), and landing. Like other airborne receivers, the receiver on the aircraft will be subject to multipath from the airframe but also from other sources depending on the flight segment and the operating environment. For instance, a UAV may experience increased multipath-induced errors during take-off and landing due to rapid changes in the geometry between the aircraft, the satellites, and any multipath sources such as close buildings, nearby objects, and even the ground (Murphy and Snow, 1997). During the cruise segment, a UAV is typically in a straight and level, unaccelerated flight (SLUF) configuration, resulting in less-rapid multipath variation (increased correlation time), unlike the loiter segment marked by turns, speed and altitude changes resulting in more rapid variation. Various multipath-induced error profiles can occur during a typical flight, and a model that works well for one segment might not work for other segments. Therefore, in the simulator, multipath is modelled using a first-order Gauss-Markov process, similar to the modelling effort in Khanafseh *et al.* (2018), with an elevation-dependent standard deviation in the range of 0.5 m to 1.18 m. The time constant τ for each satellite is assumed to be normally distributed with a mean of 35 s and a standard deviation of 10 s, similar to the modelling effort in Basile, Moore and Hill (2019).

$$M_p(t_k) = M_p(t_{k-1}) \cdot e^{-\frac{\Delta t}{\tau}} + w(t_k) \tag{5.33}$$

where: $M_p(t_k)$ is the multipath error at the epoch t_k,
$w(t_k)$ is the driving noise term,
Δt is the propagation interval (s),

τ is the correlation time constant (s).

The values for the coefficients used for the driving noise term are based on the high multipath group.

5.4.6. Thermal Noise

To simulate thermal noise affecting the raw observables (pseudoranges and Doppler frequencies), white noise with a standard deviation varying with the carrier power to noise density ratio (C/N_0) is applied to the measurements. For the pseudorange measurements, this is given by:

$$\epsilon(\rho) = N(0, \sigma_e^2(C/N_{0}{}_r^s)) \tag{5.34}$$

The standard deviation depends on the incoming signal strength and is given by:

$$\sigma_e = b_0 + b_1 \cdot e^{-\frac{C/N_{0}{}_r^s - b_2}{b_3}} \tag{5.35}$$

The coefficients for the exponential model of the standard deviation are determined by fitting the results presented by Richardson, Hill and Moore (2016). The authors analysed the Hatch filter residuals from multi-constellation datasets from reference stations within the Veripos control network. The obtained coefficients are given in Table 5.6. Even though not directly utilised in processing, the simulator outputs carrier-phase measurements. The thermal noise affecting these measurements is assumed to have a constant standard deviation of 1 mm. similarly, the Doppler frequency measurements are assumed to exhibit random noise with a constant standard deviation of 1 Hz.

Table 5.6. Coefficients for the standard deviation of the pseudorange noise for GPS (L1, L2, and L5) and Galileo (E1 and E5, which includes E5a and E5b which can be tracked individually).

Signal	b_0 [m]	b_1 [m]	b_2 [dB-Hz]	b_3 [dB-Hz]
L1	0.05	1.05	28	8
L2	0.05	1.35	28	8
L5	0.02	0.65	28	8
E1	0.02	0.55	28	9
E5b	0.02	0.40	28	9
E5a	0.02	0.25	28	9
E5	0.00	0.15	28	9

Figure 5.13 shows the pseudorange noise standard deviation for six GNSS signals. The Legacy GPS pseudoranges, on L1 and L2, have much larger noise than the Galileo signals. The modernised GPS signal on L5 seems to provide pseudoranges with comparable noise to Galileo. The pseudorange error on Galileo E5 is the smallest, largely due to its very large bandwidth of 51.15 MHz (Kaplan and Hegarty, 2017).

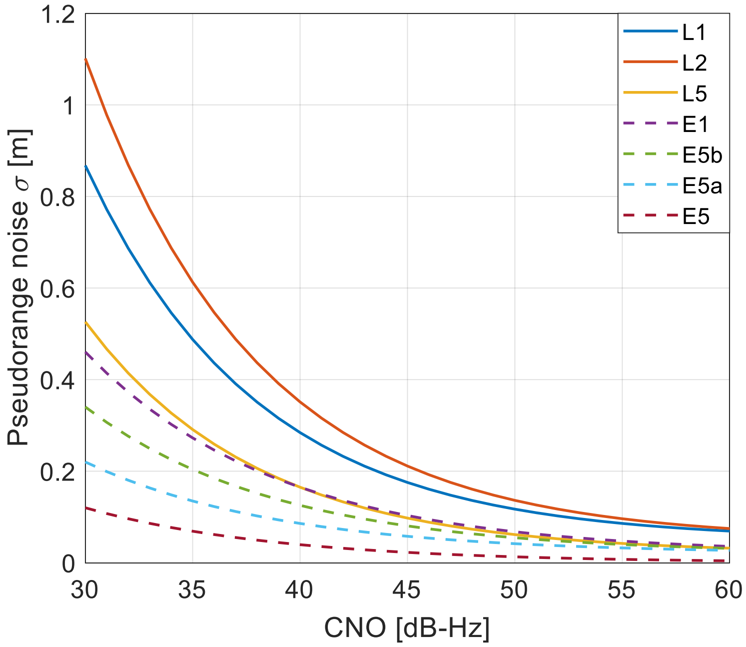


Figure 5.13. Multiconstellation pseudorange noise based on hatch filter residuals.

In this thesis, only the GPS constellation is used and therefore, only the L1 coefficients are used in the simulator. The signal strength values, C/N_0, are computed in the simulator based on Kaplan and Hegarty (2006).

$$C/N_0 = P_r + G_a - N_0 \tag{5.36}$$

where: P_r is the received signal power from a satellite at the antenna input (dBW),
N_0 is the thermal noise power component in a 1-Hz bandwidth (dBW),
G_a is the antenna gain toward a satellite (dBic).

The received minimum power levels, P_r, for Block IIA, IIR, IIR-M, IIF and III satellites are given in Table 5.7.

Table 5.7. Received Minimum RF signal strength for different GPS blocks (20.46 MHz Bandwidth) (GPS Directorate, 2019).

SV Blocks	Channel	Power: C/A or L2C [dBW]
IIA/IIR	L1	-158.5
	L2	-164.5
IIR-M/IIF	L1	-158.5
	L2	-160.0
III	L1	-158.5
	L2	-158.5

Thermal noise power, N_0, is computed using:

$$N_0 = 10\log_{10}(k \cdot (T_{ant} + T_{amp})) \tag{5.37}$$

where: T_{ant} is the antenna equivalent noise temperature assumed to be 100 K (Kaplan and Hegarty, 2006),
T_{amp} is the amplifier noise temperature (K),
k is the Boltzmann's constant equal to 1.38×10^{-23} (J/K).

When considering signal and noise paths through the front-end, one needs to consider the noise figure, N_f, of various components. The noise figure provides an estimate of the amount of noise added by an active component such as a low noise amplifier or even a passive component. The noise figure is usually given by:

$$N_f = \frac{SNR_{in}}{SNR_{out}} \tag{5.38}$$

where: SNR_{in} is the signal-to-noise power ratio into the component (dB),
SNR_{out} is the signal-to-noise power ratio after the component (dB).

The amplifier noise temperature can be related to its noise figure using:

$$T_{amp} = 290\left(10^{\frac{N_f}{10}} - 1\right) \tag{5.39}$$

where N_f is the amplifier noise figure at 290K equal to 4.3 dB (Kaplan and Hegarty, 2006). The amplifier noise temperature is estimated to be 490.5 K, and the thermal noise power is computed to be -201 dBW. The $CN0$ observations are assumed to be affected by white noise with a standard deviation of 1 dB. The $CN0$ observations for a particular satellite will vary depending on the satellite elevation angle due to differences in path loss and the satellite and receiver antenna gain pattern. The antenna gain variation with satellite elevation angle for the simulated receiver is shown in Figure 5.14. In the figure, the CNO observations (realisations) are shown for the simulated receiver alongside observations for a 25 mm x 25 mm patch antenna (u-blox ANN-MS) and 80 mm x 40 mm chip antenna (u-blox, 2019).

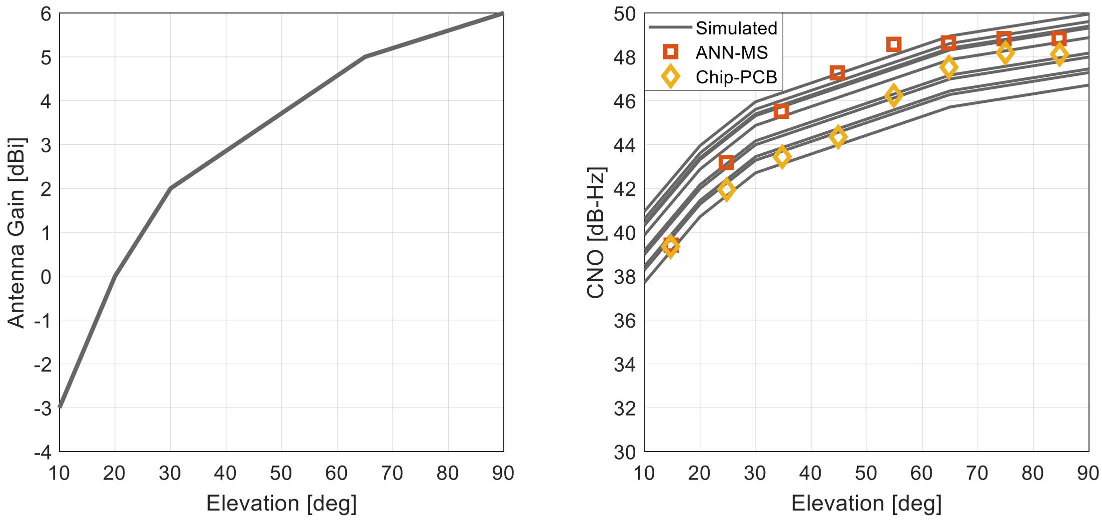


Figure 5.14. Simulated antenna gain (left) and C/N_0 observations with elevation angle (right).

5.4.7. Receiver Clock

The GNSS receiver clock introduces a common ranging error that affects measurements made to all satellites. The error is generally time-varying. The error is the same for simultaneous satellite measurements. With enough measurements, the error can be estimated and removed. Therefore, this error is generally not included as a source of positioning error. However, during a GNSS outage, where the receiver is tracking less than four satellites, this error can significantly influence the position error.

In this research, the receiver clock error has been modelled using a two-state random process model (Tawk *et al.*, 2014). The two-state model represents variations in both the oscillator frequency and phase, as shown in Figure 5.15.

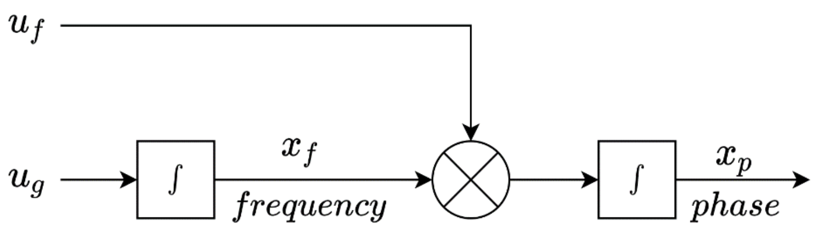


Figure 5.15. The receiver clock model.

This two-state receiver clock model is formulated as:

$$\dot{x}_p = x_f + u_f \tag{5.40}$$

$$\dot{x}_f = u_g \tag{5.41}$$

where: x_p is the receiver clock phase,
x_f is the receiver clock frequency,
u_f, u_g are the independent white noise components.

Brown and Hwang (2012) presented the spectral density components for the Allan deviation for various timing standards, which have been presented in Table 5.8.

Table 5.8. The spectral densities for different timing standards.

Timing standard	$\boldsymbol{S_f}$ [s]	$\boldsymbol{S_g}$ [s^{-1}]
TCXO	1×10^{-19}	4×10^{-19}
OCXO	1×10^{-25}	1×10^{-23}
Rubidium	1×10^{-22}	2×10^{-29}

5.4.8. Other Errors

All other errors, including the antenna phase centre offset (PCO), phase centre variation (PCV), inter-frequency biases, and clock g-dependent errors, which can become significant in applications involving very high-dynamics or high-vibrations (Groves, 2013), have been ignored in the simulator.

5.5. IMU Model

As described in Section 5.3, the proposed architecture uses raw GNSS observables and IMU measurements to estimate corrections for the navigation states. The mechanism to generate raw GNSS observables has been extensively described in the previous section, and therefore in this section, the IMU model is also described. Together, the GNSS measurement simulator and the IMU model simulate a GNSS receiver and an IMU fitted on an aircraft. The error model for IMU measurements includes a turn-on bias component, a dynamic bias component and white noise. For the specific force measurements, this model is given by:

$$\tilde{f}_{ib}^{b} = f_{ib}^{b} + b_{as} + b_{ad} + w_{i_a} \tag{5.42}$$

where: f_{ib}^{b} is the true specific force vector from the user motion file,
b_{as} is the accelerometer turn-on bias vector,
b_{ad} is the accelerometer dynamic bias vector,
w_{i_a} is the Gaussian white noise vector.

The accelerometer dynamic bias is modelled using a first-order Gauss-Markov process.

The rotation rate measurements are given by:

$$\widetilde{\omega}_{ib}^{b} = \omega_{ib}^{b} + b_{gs} + b_{gd} + w_{i_g} \tag{5.43}$$

where: ω_{ib}^{b} is the true rotation rate vector from the user motion file,
b_{gs} is the gyroscope turn-on bias vector,
b_{gd} is the gyroscope dynamic bias vector,
w_{i_g} is the Gaussian white noise vector for the gyroscope measurements.

The gyroscope dynamic bias is also modelled using a first-order Gauss-Markov process, similar to the accelerometer dynamic bias. The IMU model ignores any cross-coupling effects, scale-factors and g-dependent biases because their effects can be crudely approximated by increasing the random walk in the bias term and the noise vector, especially in a low-cost IMU. Table 5.9 shows the error characteristics of the IMU model. These error characteristics are not used in the filter to reflect a situation close to reality that the error characteristics cannot be truly known. The error characteristics are similar to those presented in Tawk *et al.* (2014) and are in general agreement with IMU used in the testing campaign presented in the next chapter.

Table 5.9. IMU error characteristics.

Property	Accelerometer	Gyroscope
Random bias (σ)	40 mg	1000 °/hr
White noise (PSD)	0.5 mg/$\sqrt{\text{Hz}}$	126°/hr/$\sqrt{\text{Hz}}$
First-order Gauss–Markov	0.05 mg	20 °/hr
Correlation Time (τ)	200 s	200 s
Sampling Frequency	100 Hz	100 Hz

5.6. Simulation Setup

This section presents the trajectory used in the simulation alongside the GNSS outage scenario investigated. The navigation filter setup is also described in this section. This includes the assignment of the initial uncertainties, process noise and measurement noise (for the IMU and GNSS measurements). One hundred Monte Carlo runs are used to investigate the performance of the proposed architecture.

5.6.1. Trajectory and GNSS Outage

The 3D trajectory used to investigate the performance of the proposed scheme is presented in Figure 5.16. The blue line shows the actual flight profile and the red line shows a sample realisation.

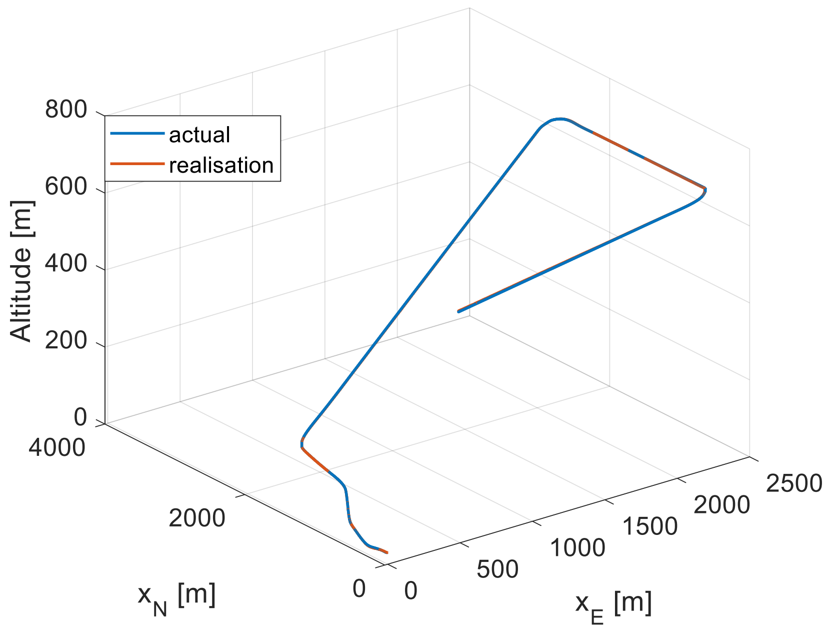


Figure 5.16. 3D Flight Profile.

The flight profile includes a take-off segment which the autopilot system completes at an altitude of 200 m, a climb segment completed at 700 m, a cruise segment and an approach segment (descent). The entire flight lasted 340 seconds. The generated user motion file from the trajectory was used to generate raw GNSS observables (pseudoranges and Doppler frequencies) and IMU measurements using the GNSS measurement simulator and IMU model described in the previous sections. A partial GNSS outage was induced 200 seconds into the flight, as can be seen in Figure 5.17. The GNSS outage is induced by masking low elevation satellites. This scenario can happen in a typical UAV flight due to high levels of in-band GNSS interference from amateur radio (operating in the 23 cm band), spurious emissions from terrestrial radio systems and GNSS Jammers (as explained in the previous chapter). Jammers are a particular threat to UAVs since they are designed to limit GNSS reception. Wilde *et al.* (2016) showed that a simple L1 chirp jammer can cause large errors (over 100 m) in the navigation solution output by a GNSS receiver and can even cause an extended GNSS outage where the receiver does not output a navigation solution. It is shown that the jammer's effect could last over an extended range, and it is more pronounced in open sky conditions, especially when the UAV is flying at higher altitudes.

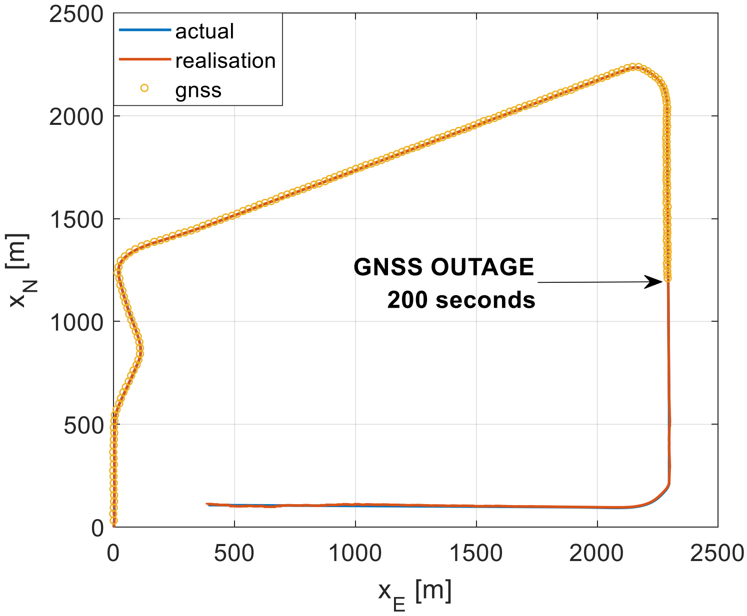


Figure 5.17. 2D Flight Profile with GNSS outage.

The authors show that there is a reduction in the carrier power to noise density ratio (C/N_0), which is more pronounced for lower elevation satellites. On the other hand, an AsteRx4 receiver showed only about 2 dB in C/N_0 reduction for it's highest elevation satellite due to an adaptive filtering strategy achieving a stand-alone positioning error of less than 0.5 m (Wilde *et al.*, 2016). For this reason, this thesis considers only high elevation satellites being tracked by a receiver during the partial GNSS outage. However, the reduction in C/N_0 for the high elevation satellites is not considered in this work. Two cases are investigated with three and two satellites tracked by a receiver during the outage as shown in Figure 5.18.

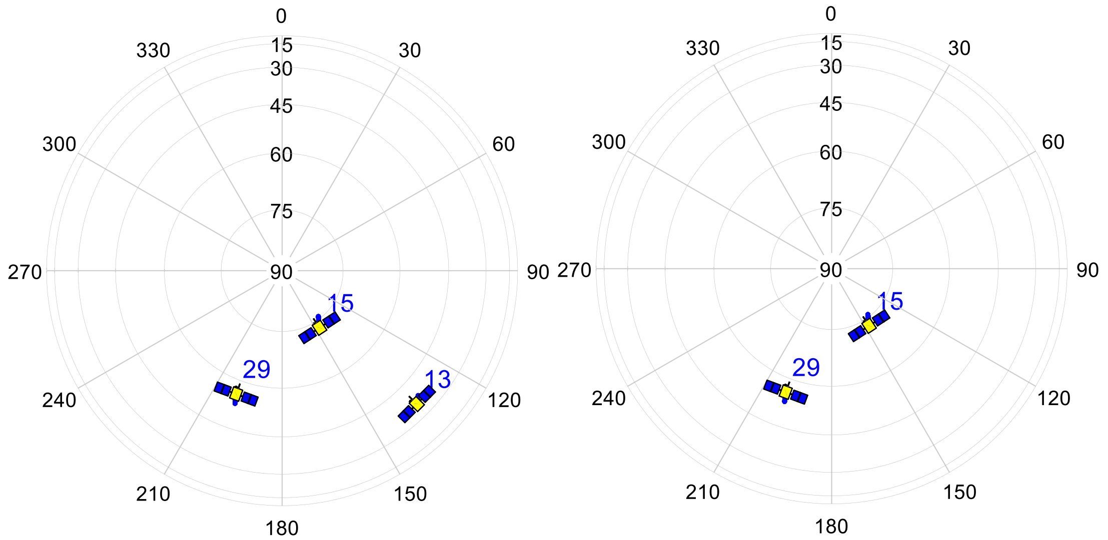


Figure 5.18. Satellites visible during the induced GNSS outage. The left plot shows three high elevation satellites and the right plot shows the remaining high elevation satellites after masking PRN 13.

In the GNSS measurement simulator, it was assumed that the flight took place in GPS week 2042, on the 61st day of the year (DOY) and 568800 seconds into the GPS week.

5.6.2. Initial Uncertainties

The navigation filter requires the initial uncertainties, process noise covariance and measurement noise covariance to be defined. The standard deviation of the initial uncertainty of the states is presented in Table 5.10. The initial error considered for the states is such that $\delta x \sim N(0, \sigma^2)$ and the filter was not sensitive to minor scaling (in the range of 1 to 2) of the initial error.

Table 5.10. Initial uncertainties for the states.

State	Standard deviation (σ)
Position	$[2, 2, 3]$ m
Velocity	$[1, 0.5, 0.5]$ m/s
Attitude	$[3.5°, 3.5°, 5°]$
Rotation rates	1.5 °/s
Propeller speed	15 rad/s
Accelerometer biases	[40 ,40, 40] mg
Wind velocity	[1.5, 1.5, 1.5] m/s
Gyroscope biases	[1000, 1000, 1000] deg/h
VDM parameters	10%
Clock offset	10^4 m
Clock drift	10 m/s

5.6.3. Process Noise

In the EKF, the state covariance matrix is propagated by:

$$P_{k|k-1} = \Phi_{k-1} \, P_{k-1|k-1} \Phi_{k-1}^T + Q_{k-1} \tag{5.44}$$

In Equation (5.44), Q_{k-1} represents the process noise covariance matrix and is usually calculated as:

$$Q_{k-1} = E\left[\int_{t-\tau_s}^{t}\int_{t-\tau_s}^{t} \exp\left(F_{k-1}\cdot(t-t')\right) G_{k-1}\, W_s(t') W_s^T(t'')\, G_{k-1}^T \exp\left(F_{k-1}\cdot(t-t'')\right)^T dt'dt''\right] \tag{5.45}$$

The equation above can be difficult to evaluate, especially when the dynamic matrix is time-varying, as is the case with the proposed scheme. Therefore, the equation is usually simplified in the computation of the process noise covariance by ignoring the time propagation (0.01 s in this thesis) of the system noise over an iteration of the filter (Groves, 2013). However, in some cases, this simplification can lead to suboptimal performance. It is simple enough to show the actual process noise for the receiver model because its dynamic matrix is constant. Therefore, the GNSS receiver clock process noise covariance is given by:

$$Q_{clk} = \begin{bmatrix} \frac{S_g \tau_s^3}{3} + S_f \tau_s & \frac{S_g \tau_s^2}{2} \\ \frac{S_g \tau^2}{2} & S_g \tau_s \end{bmatrix} \tag{5.46}$$

In Equation (5.46), $S_f = 0.009 \text{ m}^2/\text{s}$ and $S_g = 0.0355 \text{ m}^2/\text{s}^3$ represent the spectral amplitudes for the two receiver Gaussian white noise sources, u_f and u_g. The standard deviations of the main diagonal terms of the tuned discrete process noise covariance are presented in Table 5.11. The values for the receiver clock offset and drift are not listed in the table since their spectral amplitudes have already been given.

Table 5.11. The standard deviation of the diagonal terms of the process noise covariance matrix.

State	**Standard deviation (1σ)**
Position	10^{-4} m
Velocity	10^{-4} m/s
Attitude	10^{-4} rad
Rotation rates	10^{-4} rad/s
Propeller speed	10^{-4} rad/s
Accelerometer Bias	2×10^{-5} m/s^2
Gyroscope Bias	2×10^{-6} rad/s
Wind	5×10^{-3} m/s
Model parameters	0.15% of True Values

And the general form of the process noise covariance matrix is given by:

$$Q_k = \begin{bmatrix} Q_{x(1:13)} & 0 & 0 & 0 & 0 \\ 0 & Q_{X_e} & 0 & 0 & 0 \\ 0 & 0 & Q_{X_w} & 0 & 0 \\ 0 & 0 & 0 & Q_{X_p} & 0 \\ 0 & 0 & 0 & 0 & Q_{X_{clk}} \end{bmatrix} \tag{5.47}$$

5.6.4. Measurement Noise

The variance values for the IMU measurement noise covariance matrix are considered within the range of the error characteristics given in Table 5.9. Since the IMU biases are estimated in the filter, their values are not included in the covariance matrix.

For the pseudorange observables, an elevation-dependent covariance matrix is used to match the characteristics of the presented models. This model is based on the model presented in Pinchin (2011).

$$\sigma_s^2 = R_\sigma \left(a_\sigma^2 + \frac{b_\sigma^2}{\sin(E^s)} \right) + \sigma_{sclock}^2 + \sigma_{iono}^2 + \sigma_{tropo}^2 \tag{5.48}$$

where: a_σ and b_σ are set to 0.03 m and 0.04 m, respectively,
R_σ is the code to carrier error ratio set to 100,
E^s is the satellite elevation in radians,
σ_{sclock} is the standard the satellite clock error set to 1.0 m.

The empirical parameters a_σ, b_σ, and R_σ are determined by fitting the elevation-dependent model to the combined standard deviation of the simulated multipath-induced range error and the receiver thermal noise presented in Section 5.4.5 and Section 5.4.6, respectively. In the filter, the values are slightly scaled (in the range from 1 to 2) in each run to reflect a situation close to reality that the true error characteristics can not be truly known. Following the correction of the ionospheric delay using the Klobuchar model, the standard deviation of the residual ionospheric delay is given by:

$$\sigma_{iono} = 0.5\ I_r^s \tag{5.49}$$

where I_r^s is the computed ionospheric delay between the receiver (r) and satellite (s). The tropospheric delay is corrected using the Saastamoinen model (Martellucci and Prieto-Cerdeira, 2009). The standard deviation of the residual tropospheric delay is given by:

$$\sigma_{tropo} = \frac{0.2}{\sin(E^s) + 0.1} \tag{5.50}$$

Doppler frequency measurements are assumed to be affected by white noise with a standard deviation of 0.75 Hz (Takasu and Yasuda, 2013). Even though the value is in close agreement with the standard deviation of the simulated Doppler frequency error presented in Section 5.4.6, it is slightly scaled in each run for the same reason that the true error characteristics can not be truly known.

5.7. Simulation Results

This section presents the simulation results based on the setup described in the previous section. Comparisons are made to a standard tightly coupled INS/GNSS integration architecture (TCINS) described in Section E.2 of Appendix E and a loosely coupled VDM-based integration architecture (LCVDM) described in Section 4.4. The TCINS and LCVDM schemes used the same IMU error characteristics presented in the previous section. The LCVDM scheme used position measurements obtained directly from the simulation, with a random noise component with a standard deviation of 1 m added to the measurements.

5.7.1. Position

Figure 5.19 shows the position error estimation performance for the TCVDM, TCINS and LCVDM architectures with three satellites visible during the GNSS outage. In the figure, the position errors are plotted against the predicted uncertainties (1σ). The final position error for the TCVDM scheme is 18.39 metres, while for the TCINS scheme, it is 29.14 metres. This is an improvement by a factor of about 1.6 for the TCVDM scheme over the TCINS scheme owing to the mitigation provided by the dynamic model. The TCVDM scheme also shows an improvement by a factor of 4.7 over the LCVDM scheme. The significant improvement in position error estimation by the TCVDM scheme over the LCVDM scheme is due to the use of raw GNSS observables (pseudoranges and Doppler frequencies) that continue to be available even when tracking less than four satellites.

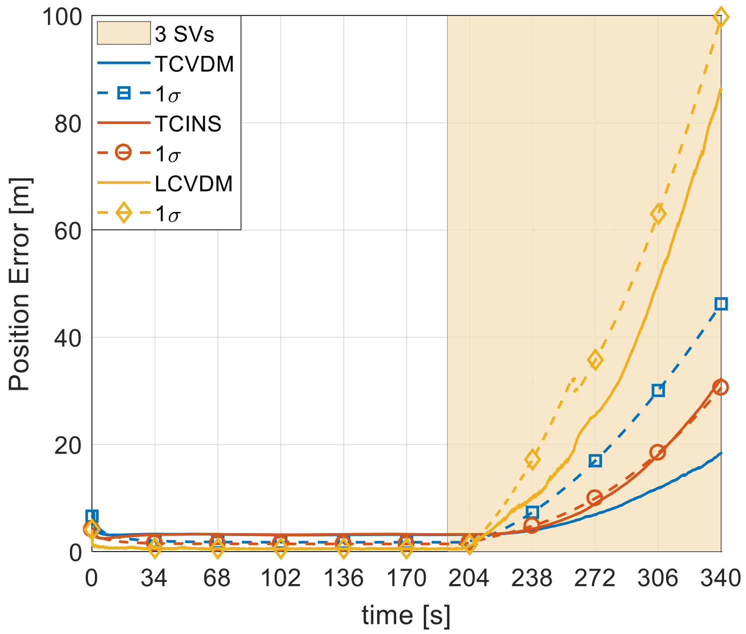


Figure 5.19. Position error with three satellites visible during the outage.

With two satellites visible during the outage, the position error at the end of the flight for the TCINS scheme reaches 357 metres, an order of magnitude larger than the TCVDM scheme, as can be seen in Figure 5.20.

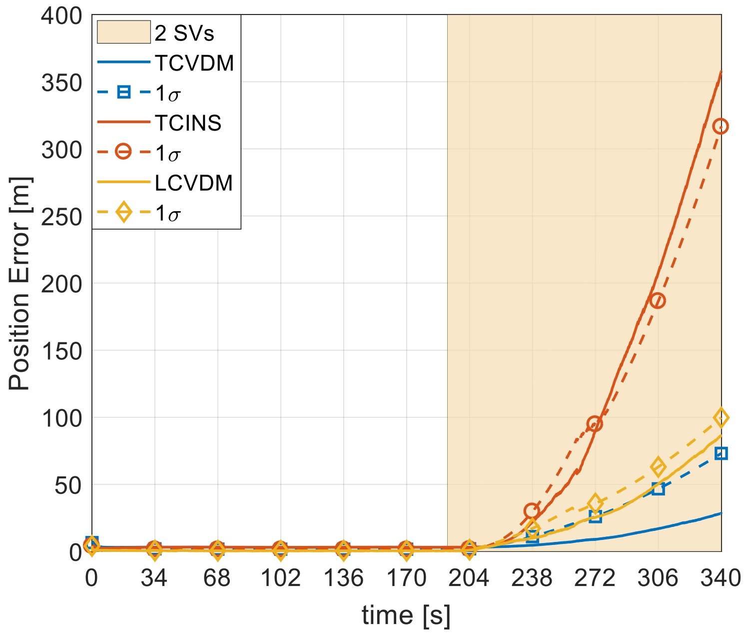


Figure 5.20. Position error with two satellites visible during the outage.

It is also important to mention that the filter seems slightly optimistic in height estimation leading to an overall optimistic nature in the 3D position error. This is attributed mostly to the residual range biases that are not directly estimated within the filter, making the overall error slightly larger. The position error for the

TCVDM scheme with two satellites visible during the outage increased by 54.5% as opposed to the error with three satellites visible. This is still an improvement by a factor of 3 as opposed to the LCVDM scheme.

5.7.2. Velocity

Figure 5.21 shows the velocity error estimation performance for the TCVDM, TCINS and LCVDM architectures with three satellites visible during the GNSS outage. In the figure, the velocity errors are plotted against the predicted uncertainties (1σ). The velocity estimation performance in the east channel for the TCVDM scheme is similar to the TCINS scheme. However, the final velocity error for the TCINS scheme is greater than the error for the TCVDM scheme by a factor of 1.45 in the north channel and by a factor of 1.8 in the down channel. Around 260 seconds, the north and east velocity estimation errors for the LCVDM scheme changed significantly. This also occurs after the outage is induced (200 seconds into the flight). This seems to indicate that most of the aircraft's velocity error is due to the cross-track wind component. For instance, around 260 seconds, the aircraft turns and heads west with mostly a southerly wind leading to a significant increase in the velocity error in the north channel, reaching 1.4 m/s at the end of the outage. At the same time, this error only reaches 0.2 m/s for the TCVDM scheme, an improvement by a factor of 7 compared to the LCVDM scheme.

With two satellites visible during the outage, the TCVDM scheme showed very gradual growth in velocity errors reaching only 0.4m/s, 0.24m/s and 0.22 m/s in the north, east and down directions, as can be seen in Figure 5.22. This is an order of magnitude better in the north and east channels as opposed to the TCINS scheme.

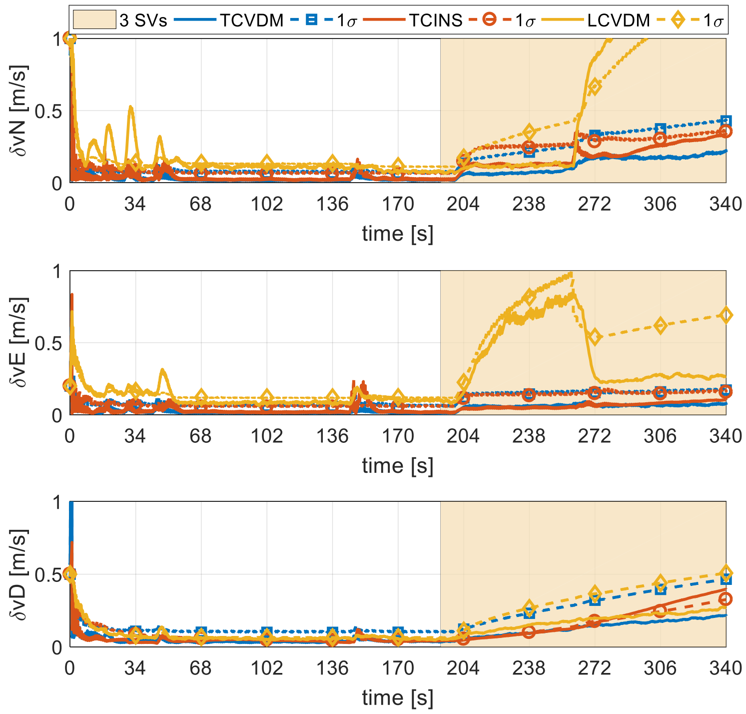


Figure 5.21. Velocity estimation errors with three satellites visible during the outage.

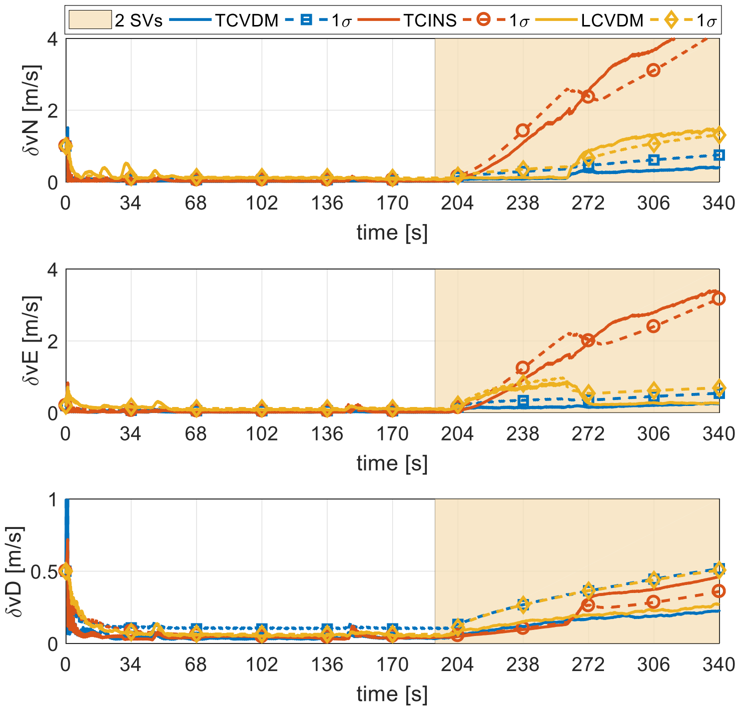


Figure 5.22. Velocity estimation errors with two satellites visible during the outage.

A close inspection of the correlation matrix for the TCINS scheme, presented in Figure 5.23, shows that the horizontal velocity components are correlated significantly with their respective position components. This is expected, and the TCVDM correlation plot from a sample realisation shows a similar pattern in Figure 5.24. However, inspecting this matrix reveals that the estimation of wind velocity helps to reduce rapid error growth in the TCVDM architecture due to the correlation between the wind velocity terms and the aircraft's velocity components.

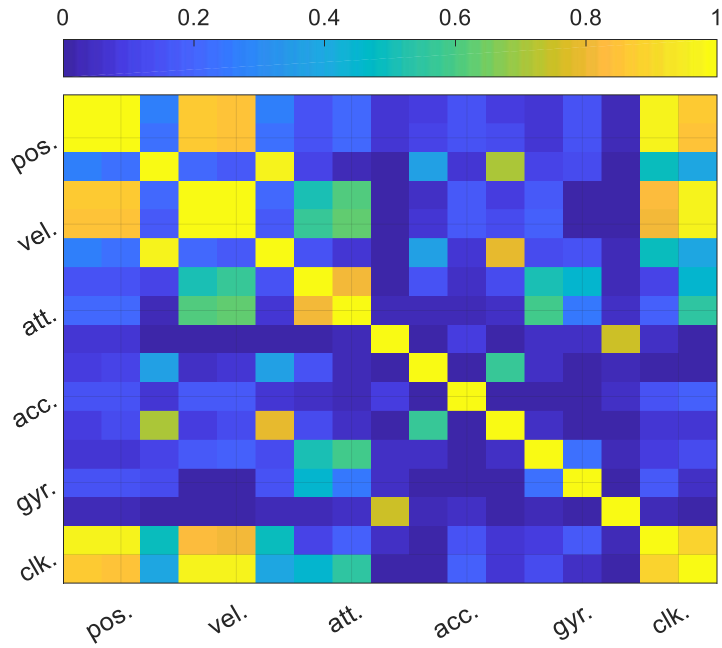


Figure 5.23. Correlation plot for the INS-based scheme at the end of the flight with only two satellites visible during the GNSS outage.

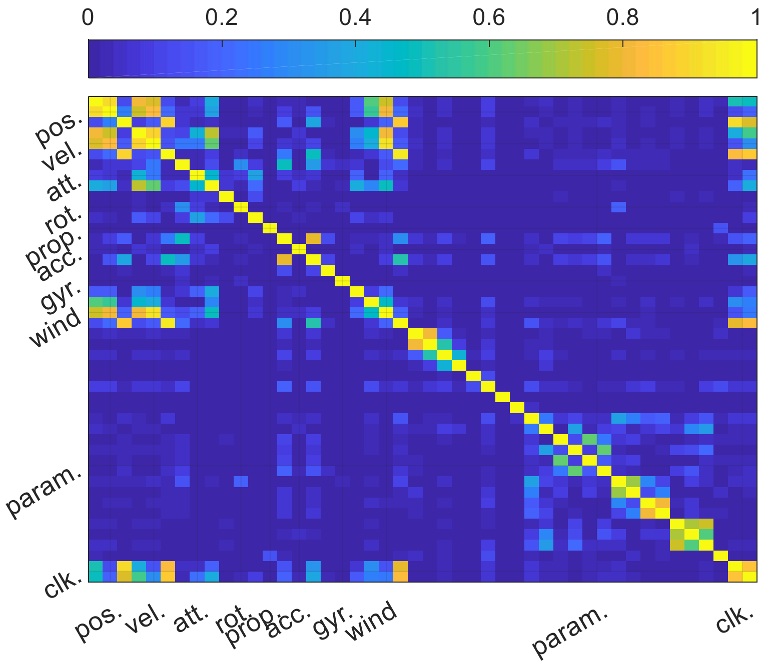


Figure 5.24. Correlation plot for the TCVDM architecture at the end of the flight following a GNSS outage with two satellites visible during the outage.

Figure 5.25 shows the correlation coefficient between the aircraft's velocity vector and the wind velocity vector.

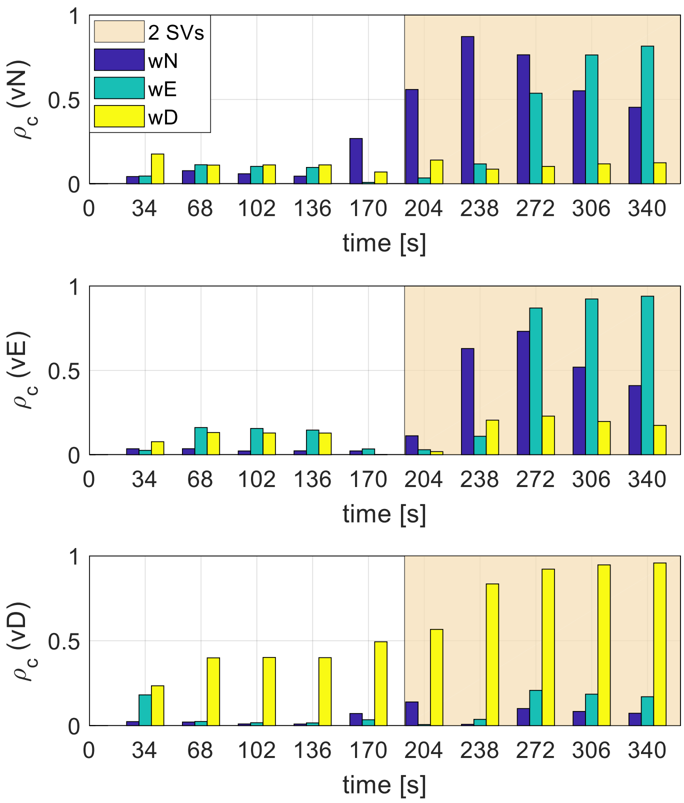


Figure 5.25. A realisation of the correlation coefficient (ρ_c) between the aircraft's velocity vectors and wind velocity for the TCVDM architecture with two satellites visible during the GNSS outage.

Evidently, the aircraft's horizontal velocity components are correlated with the horizontal wind velocity components. In case an air data system is not available, or wind velocity is not estimated within the filter, as is the case with a standard INS-based scheme, errors will accumulate rapidly. Therefore the direct mechanism to estimate wind velocity in the TCVDM and LCVDM architectures helps mitigate this rapid error growth in the velocity vector.

5.7.3. Attitude

Figure 5.26 shows the attitude error estimation results and the predicted uncertainties (1σ) for the TCVDM, TCINS and LCVDM schemes with three satellites in view during the outage. Most attitude errors for the TCVDM architecture and LCVDM scheme are resolved well within the first 100 seconds of GNSS availability. The final pitch and yaw angle estimation errors for the TCVDM architecture seem slightly higher than for the TCINS scheme. The pitch angle estimation error for the LCVDM scheme is very similar to its TCVDM counterpart, while the roll and yaw angle estimation errors for the LCVDM scheme are greater by a factor of 2.5 and 5 compared to their TCVDM counterparts.

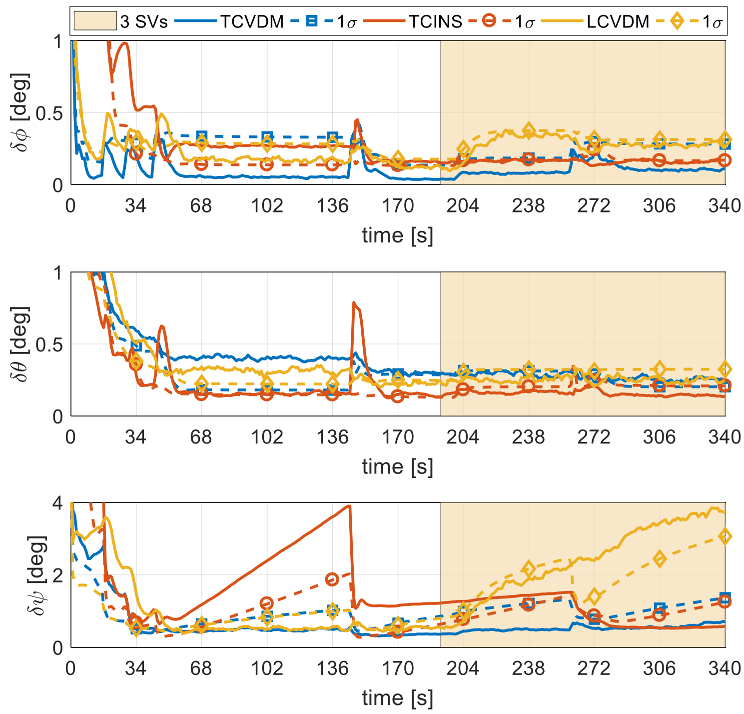


Figure 5.26. Attitude estimation errors with three satellites visible during the GNSS outage.

Figure 5.27 shows the RMS of attitude estimation errors and the predicted uncertainties (1σ) for the TCVDM, TCINS and LCVDM schemes with two satellites visible during the GNSS outage.

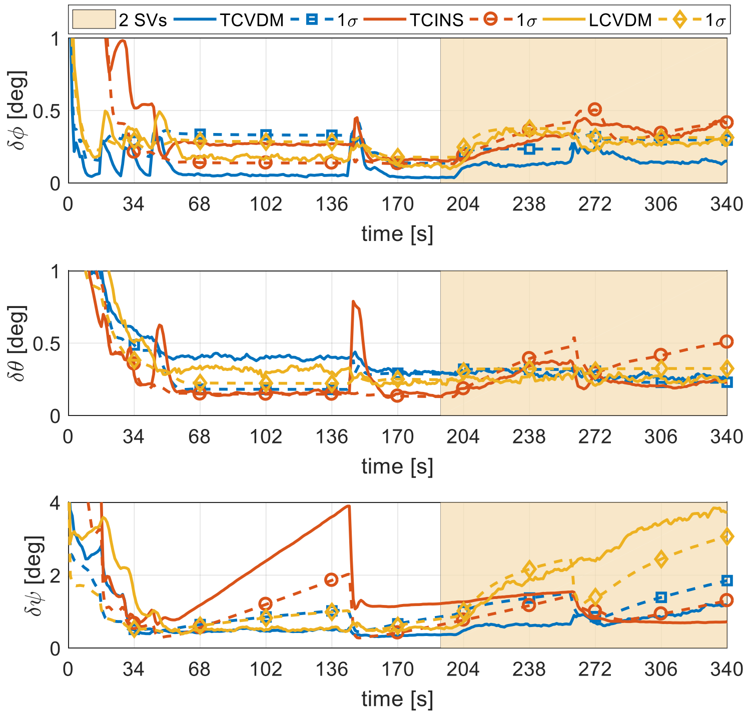


Figure 5.27. Attitude estimation errors with two satellites visible during the GNSS outage.

Generally, the final attitude estimation errors increase slightly compared to the estimation errors with three satellites in view. The final roll angle estimation error for the TCVDM scheme increases by a factor of 1.25 to 0.15° while for the TCINS scheme, it increases by a factor of 2.5. On the other hand, the final pitch angle estimation error for the TCVDM scheme stays the same while it increases by a factor of 2 for the TCINS scheme. The final yaw angle estimation error increases by 57% for the TCVDM scheme and by 26% for the TCINS scheme. Roll and pitch angle estimation errors increase rapidly for the TCINS scheme and only gradually for the TCVDM scheme with a decrease in the number of satellites in view during the outage. This gradual growth of attitude estimation errors for the TCVDM scheme is due to the extra mitigation provided by the dynamic model of the aircraft. It is important to mention that, even though the growth in attitude errors is gradual with decreasing number of satellites in view, the final yaw angle estimation error for the TCVDM architecture is still larger than for the TCINS scheme by almost 60%. In the TCVDM scheme, the yaw angle seems to be correlated with the horizontal wind velocity components, which helps explain the gradual accumulation of this error during the straight and level segment following a turn (between 272 and 340 seconds). This is further indicated by the gradual accumulation of this error by the LCVDM scheme reaching 3.7° at the end of the flight.

5.7.4. IMU Errors

Figure 5.28 and Figure 5.29 show the RMS of accelerometer and gyroscope bias estimation errors, respectively, with two satellites visible during the outage for all 100 runs. The filter estimates about 90% of the initial errors well within the first

40 seconds of GNSS availability, and the estimation continues to improve even during the GNSS outage.

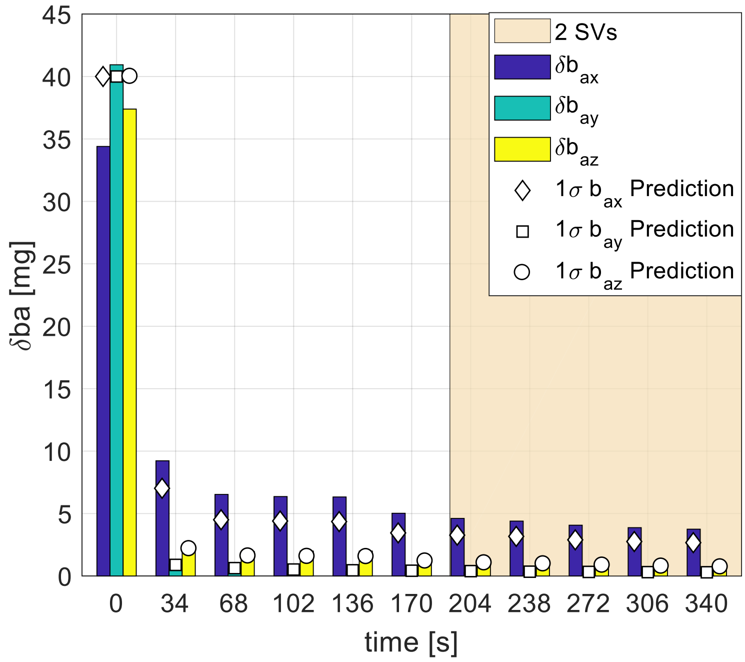


Figure 5.28. Accelerometer bias estimation errors for the TCVDM scheme with two satellites visible during the GNSS outage.

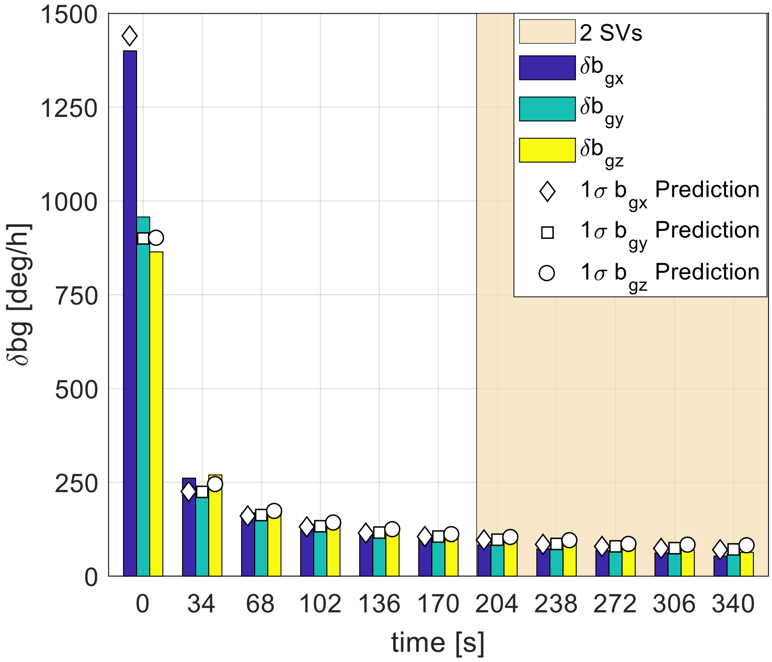


Figure 5.29. Gyroscope bias estimation errors for the TCVDM scheme with two satellites visible during the GNSS outage.

Further, the filter's predicted confidence values (1σ) seem to be consistent with the empirical RMS error due to the correctness of the filter setup. The x-axis accelerometer bias is slightly delayed in its estimation. However, its estimation continues to improve even during the GNSS outage. The continued estimation of both accelerometer and gyroscope biases even during the GNSS outage shows that the filter is able to keep track of these errors and is attributed to the use of the VDM and direct IMU measurements.

5.7.5. Wind Velocity

Figure 5.30 shows the RMS of wind speed estimation errors and the predicted uncertainties (1σ) for the TCVDM scheme with three and two satellites (TCVDM-3 and TCVDM-2) visible during a GNSS outage compared to the LCVDM scheme.

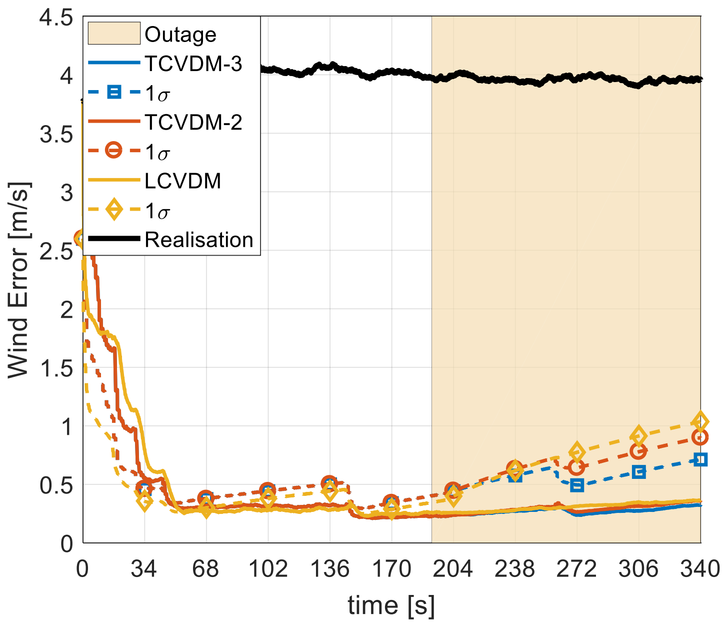


Figure 5.30. Wind velocity estimation errors.

The error in the estimation of wind speed seems to increase with decreasing number of visible satellites during the outage. However, there is only a 10% difference between the error estimated with three satellites in view to the error estimated with two satellites in view during the outage. Turning seems to improve the observability of wind errors slightly, as shown in Figure 5.30 around 260 seconds. However, a straight and level flight following a turn seems to reduce the filter's confidence in wind estimation, as can be seen from 272 seconds to the end of the flight.

5.7.6. VDM Parameters

Figure 5.31 shows the RMS mean error in the estimation of VDM parameters for all 100 runs. The estimation of VDM parameters does not seem to be affected by the decrease in satellites visible during the GNSS outage, thanks to the available IMU measurements. The filter seems to resolve only 40% of the initial VDM parameter uncertainty due to correlation within groups of the parameters. However, for an initial uncertainty of 10%, the performance enhancement is sufficient for navigation due to the significant improvement in navigation accuracy.

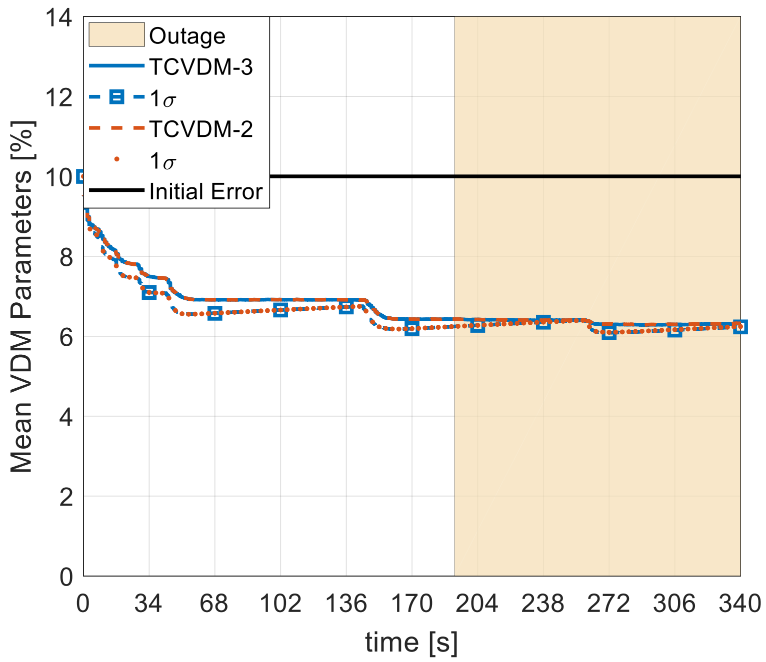


Figure 5.31. VDM parameters estimation errors and predicted uncertainties (1σ).

5.7.7. Receiver Clock

The RMS of the receiver clock bias and drift errors and their predicted uncertainties (1σ) are presented in Figure 5.32 and Figure 5.33, with three and two satellites visible during the GNSS outage.

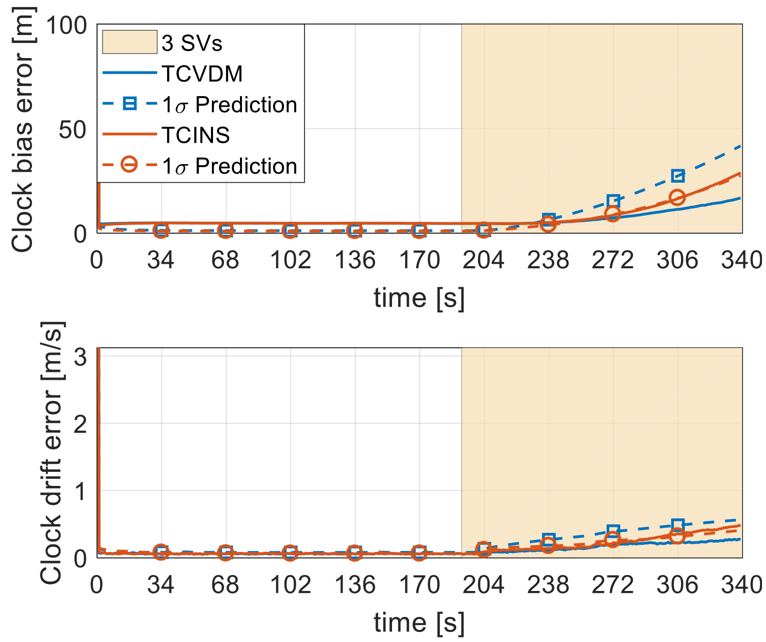


Figure 5.32. Receiver clock bias and drift estimation errors with three satellites visible during the GNSS outage.

The error in the clock bias estimated by the TCVDM architecture increases gradually, reaching only 17 meters with two satellites in view during the GNSS outage. This is only 5% higher than with three satellites in view and an improvement by a factor of 5 compared to the TCINS scheme. With two satellites visible during the outage, the final error in the clock drift estimated by the TCVDM scheme is six times better than that estimated by the TCINS scheme. The improved performance of the navigation states of the TCVDM scheme helped reduce rapid growth in the clock bias and drift errors during the outage, unlike the TCINS scheme. The predicted confidence values (1σ) of the clock bias for both schemes seemed optimistic during GNSS availability due to other range biases not estimated within the filter leading to increased error in the clock bias and position states.

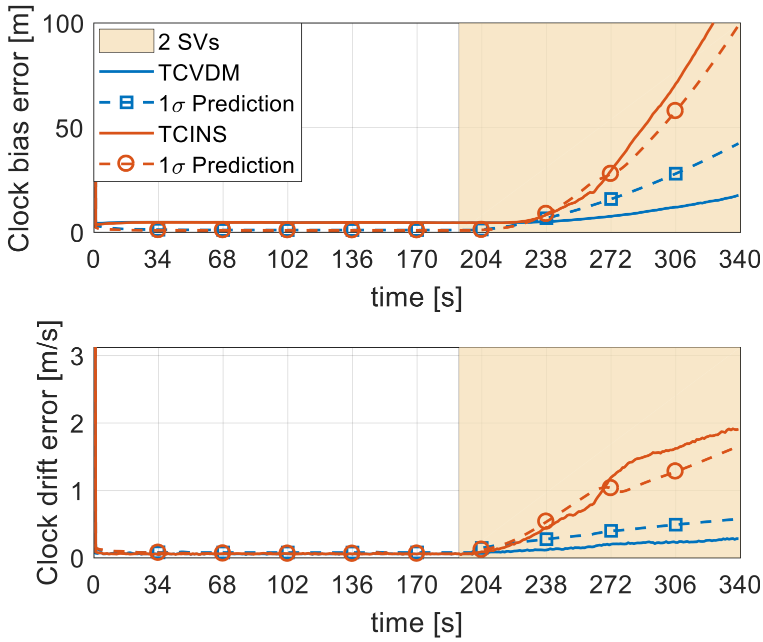


Figure 5.33. Receiver clock bias and drift estimation errors with two satellites visible during the GNSS outage.

5.7.8. Uncertainty Evolution

The ratio of uncertainties, at different times, on the states allows a discussion of their observability. The validity of this discussion stems from the close agreement between the empirical errors and the predicted confidence values in the covariance matrix for most navigation states and other auxiliary states.

Figure 5.34 shows the ratio of uncertainties on the states during the GNSS availability period for the TCVDM scheme. The ratio is given by the uncertainties at the end of the GNSS availability period (t=200s) to the uncertainties at initialisation (t = 0s). Alternating colours are used to represent the sub-state vectors that belong to different groups. For instance, the first three black columns represent the uncertainties on the position states and the following three blue columns represent the uncertainties on the velocity states. Generally, the uncertainties on all navigation states and auxiliary states such as IMU errors, wind velocity states, and the GNSS receiver clock errors decrease significantly during the GNSS availability period. The uncertainties on most VDM parameters also decrease during this period, with some showing significant reduction than others. For instance, the uncertainties on static thrust coefficient (CF_{T_1}) lift curve slope (CF_{Z_α}), pitching moment coefficient at the aerodynamic centre and pitch control derivative ($CM_{Y1}, CM_{Y_{\delta e}}$) decrease greatly as opposed to other VDM parameters. However, the uncertainties on the second-order thrust coefficient and most drag coefficients decrease only slightly. This significant decrease in the uncertainties on the parameters highlighted indicates that they are more observable than other VDM parameters, perhaps due to the significant impact they have on position accuracy (see sensitivity analysis in Section B.2 of Appendix B of this thesis).

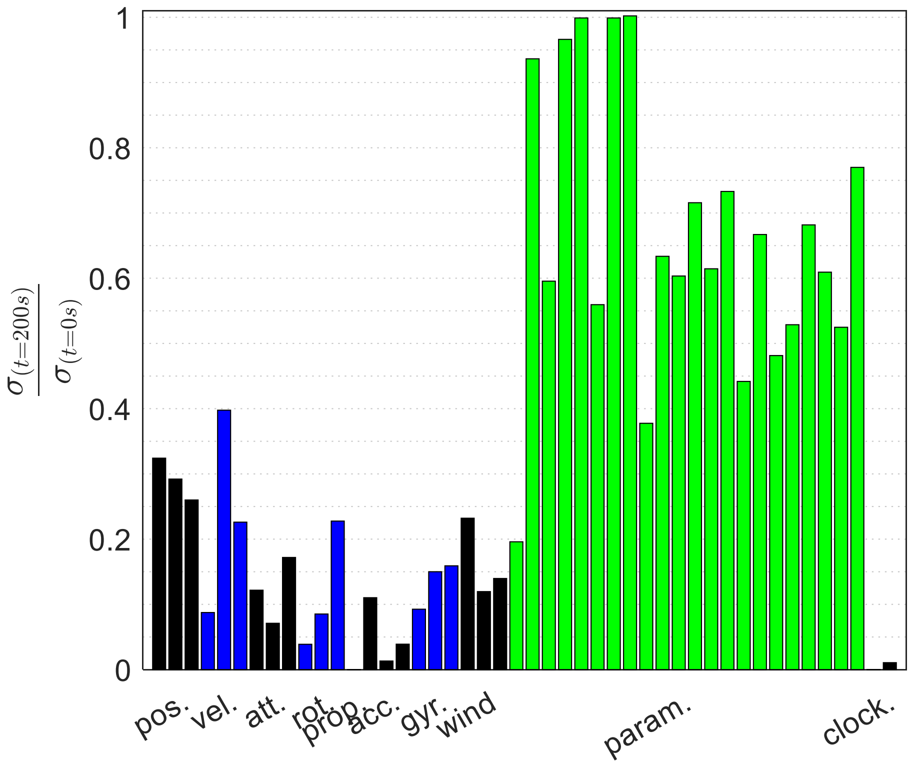


Figure 5.34. The ratio of uncertainties for the TCVDM scheme during GNSS availability.

Figure 5.35 shows the ratio of the uncertainties with three satellites in view during the GNSS outage. The ratio is given by the uncertainties at the end of the GNSS outage (t=340s) to the uncertainties at the beginning of the outage (t=200s). The uncertainty on position states increases by at least a factor of 16, and for the velocity states, it increases by at least a factor of 2. For the attitude states, the ratio decreases to 0.8 for roll angle and to 0.95 for pitch angle, while it increases to 1.6 for yaw angle. The uncertainty on IMU errors decreases during the outage, which indicates that they continued to be observable even during this period. The uncertainty on some VDM parameters decreases during the outage, indicating that the parameters continued to be observable during this period. The uncertainties on the roll control derivative ($CM_{X_{\delta a}}$) and the roll damping derivative ($CM_{X_{\bar{\omega}_x}}$) increase slightly during the outage. The uncertainty on the receiver clock bias increases by a factor of 37, and on the receiver clock drift, it increases by a factor of 3.

In comparison, Figure 5.36 shows the ratio of uncertainties during the GNSS outage for the LCVDM scheme. During this period, the uncertainties on most navigation states increase, with the uncertainty on position states increasing by at least a factor of 140. Similarly, the uncertainty on the velocity states increases by at least a factor of 5.7. The uncertainties on most VDM parameters for the LCVDM scheme are similar to the uncertainties for the TCVDM scheme with either two or three satellites in view during the GNSS outage.

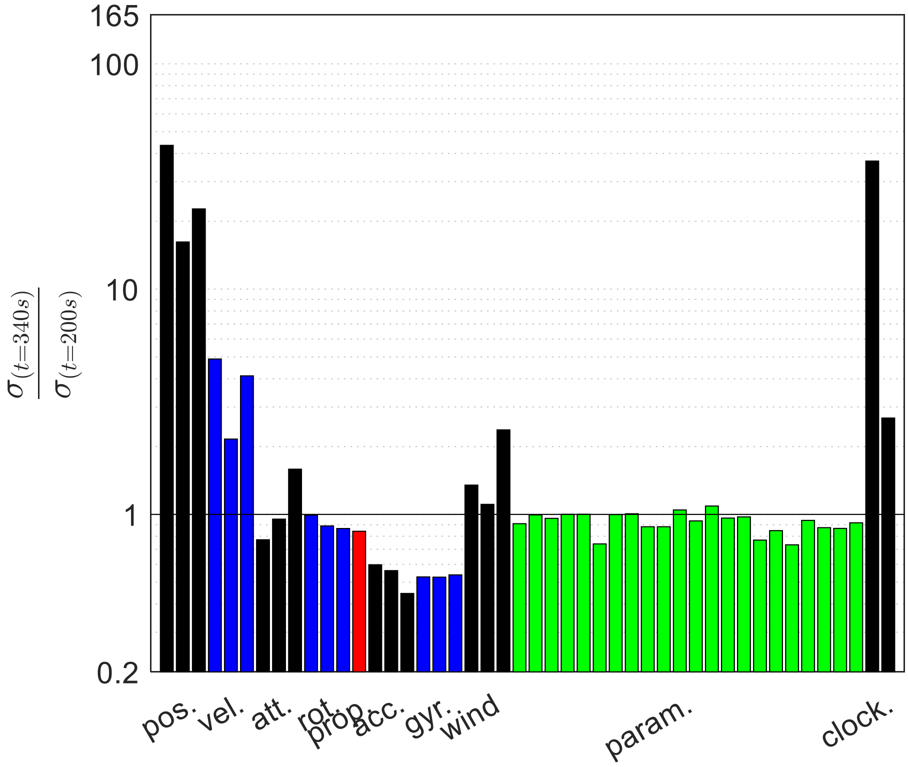


Figure 5.35. The ratio of uncertainties for the TCVDM scheme with three satellites during the outage.

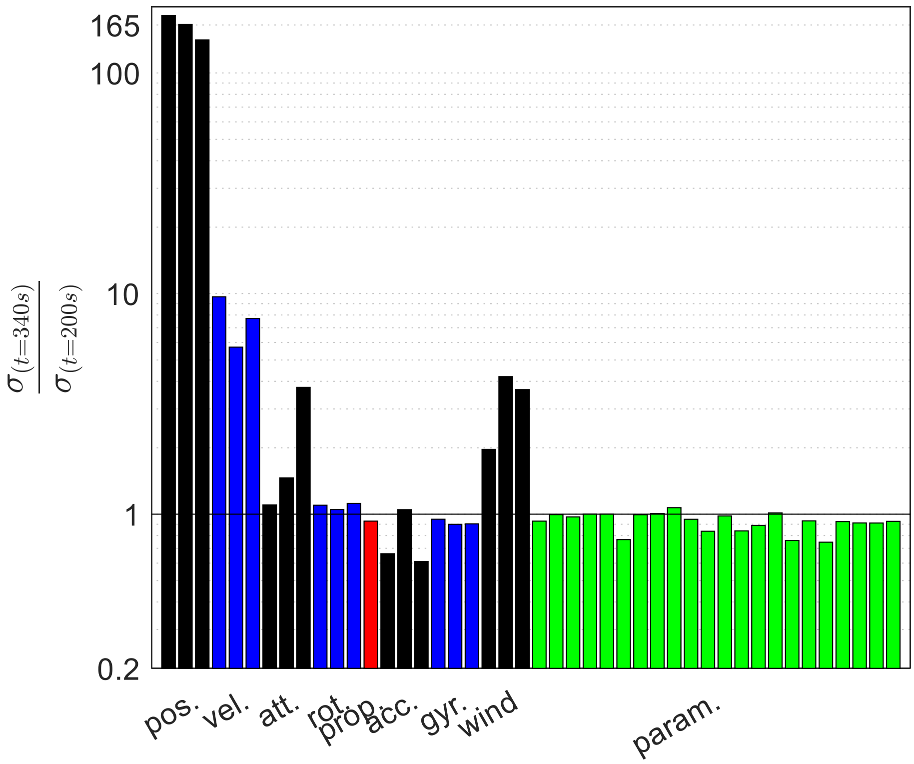


Figure 5.36. The ratio of uncertainties for the LCVDM scheme during the GNSS outage.

The performance with two satellites during the outage is similar to the performance with three satellites, with increased uncertainties on some navigation states.

5.7.9. Correlation

Figure 5.37 and Figure 5.38 show a realisation of the correlation matrix for all TCVDM states before the outage (100 seconds into the flight) and at the end of the GNSS outage with only two satellites visible.

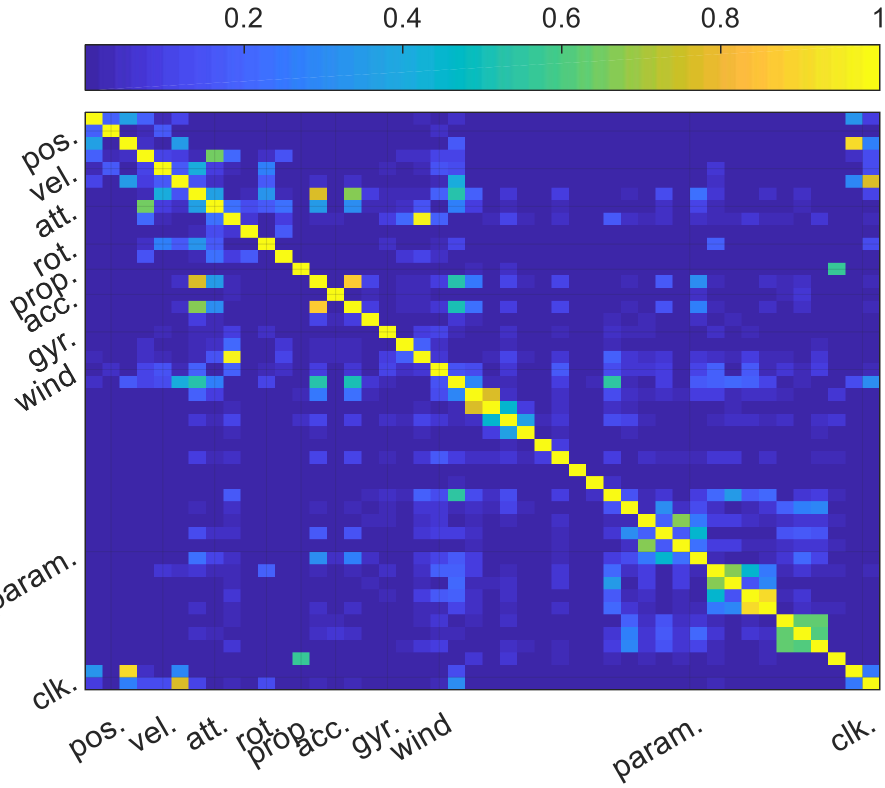


Figure 5.37. Correlation matrix for the TCVDM scheme one hundred seconds into the flight.

During GNSS availability, the clock bias seemed to be significantly correlated with the down component of the position vector. The range biases not estimated in the filter seem to influence the down component of the position vector alongside the receiver clock bias. This helps explain the optimistic nature in the estimation of the position error and the clock offset during GNSS availability.

During the outage, the clock bias and drift terms seemed to be significantly correlated with the down components of position, velocity and wind vectors, which helped mitigate rapid error growth during this period. VDM parameters showed significant correlation within groups and some correlation with other navigation states. The correlation with other navigation states is essential for the overall VDM parameter observability. The observability of VDM parameters is generally trajectory dependent, but even for a modest flight profile, 40% of the initial uncertainty can be resolved.

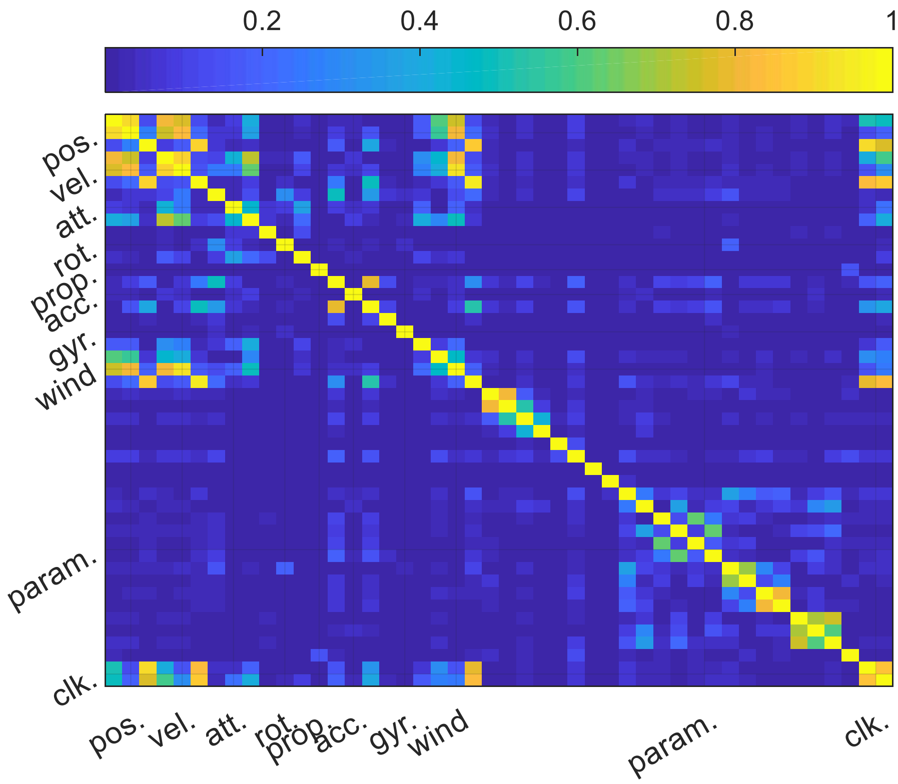


Figure 5.38. Correlation matrix for the TCVDM architecture at the end of the GNSS outage with only two satellites visible.

5.8. Summary

In this chapter, an innovative, tightly coupled vehicle dynamic model-based integration architecture (TCVDM) capable of taking full advantage of available raw GNSS observables (pseudoranges and Doppler frequencies) during a GNSS outage has been presented and analysed. A specific case to a fixed-wing UAV has been investigated, which, alongside the raw observables, uses measurements from a low-cost MEMS-grade IMU to aid the navigation solution.

A GNSS measurement simulator used to derive raw GNSS observables used in the fusion filter is presented and analysed. The reasons for developing and using a software-based measurement simulator are highlighted, and different inputs and settings to the simulator are explained. One advantage of a software-based simulator is the ability to decouple the receiver dynamics from the raw observables. Different error models used in deriving the raw GNSS observables are presented, and some of their limitations are discussed. A summary of the error models used in the simulator is presented in Section B.3 of Appendix B. Following the input of a user motion file, the simulator outputs raw GNSS observables used in the fusion filter.

A Monte Carlo simulation study is used to evaluate the performance of the proposed scheme. The key question being addressed was postulated in the previous chapter. It reads:

- *Can a VDM-based approach gain improved performance from raw GNSS observables available even when tracking less than four satellites?*

Simulation results of the proposed architecture are presented and analysed, along with comparisons to a tightly coupled INS/GNSS integration architecture as well as a loosely coupled VDM scheme (TCINS and LCVDM). Simulation results revealed that the proposed architecture could improve position estimation by one order of magnitude with two satellites visible during an extended GNSS outage lasting over two minutes as opposed to a TCINS. Further, it was found that for a modest trajectory, the proposed architecture only captures about 40% of the initial uncertainty in the VDM parameters due to the significant amount of correlation within groups of the parameters. Other auxiliary states such as wind, IMU errors and clock errors were well estimated even with only two satellites visible. The online estimation of wind velocity also seemed to improve the estimation performance of the aircraft's velocity states due to the significant amount of correlation between the states.

6 Flight Test Measurements

6.1. Introduction

This chapter presents the flight test setup and test results of the proposed tightly coupled VDM-based (TCVDM) integration architecture.

The chapter is organised as follows: The sensors and systems used on the aircraft are described in Section 6.2 and Section 6.3, respectively. The aircraft characterisation routine is explained in Section 6.4 and Section 6.5. Section 6.6 describes the test flight to gather data for testing the architecture. Section 6.7 presents the derivation of the reference navigation solution, and the results are presented in Section 6.9 of this chapter.

The work presented in this chapter has been published in Mwenegoha *et al.* (2020).

6.2. Platform Description

An off-the-shelf fixed-wing UAV (Riot V2) was modified and fitted with a custom flight control system (FCS) to validate the performance of the proposed integration architecture. More specifically, the on-board setup consisted of:

- A MEMS-grade IMU - The NXP 9DOF IMU consisting of the FXAS21002 3-axis gyroscope and the FXOS8700 3-axis accelerometer and magnetometer was used on the flight control system (NXP Semiconductors, 2015, 2017). The IMU was sampled at 100Hz and used to measure the specific force and rotation rates on the UAV. These were then used in guidance and control of the aircraft and were also logged at 20 Hz to test the developed scheme. The IMU was configured to raise an interrupt whenever data was ready for processing.
- A barometer – The Bosch Sensortec BMP388 sampled at 25Hz was used to provide height measurements used in the flight control system (Bosch, 2018). Data from the barometer was also logged but was not used to test the performance of the developed architecture.
- GNSS receiver – Three multi-constellation, u-blox NEO-M8T receivers with an output rate of 4Hz were used on the platform with data from the modules used in post-processing to validate the proposed architecture (u-blox, 2020). Each receiver was installed at a specific location on the aircraft (see Figure 6.24). The coordinates of each receiver relative to the centre of gravity of the aircraft in the body-fixed frame were known and used to derive a GNSS attitude solution used to assess the performance of the developed architecture.
- Flight control system – The ATmega2560, loaded with custom flight control firmware, was used for guidance, navigation, and control of the aircraft. The unit combines 256KB flash memory, 8KB SRAM and 4KB EEPROM (Microchip Technology, 2020b). The unit achieves a throughput of 16MIPS at 16MHz.

- A Datalogger – The Openlog datalogger based on an ATmega328 running at 16MHz was used for logging data from the IMU, BMP388, GNSS receivers and control inputs at 20Hz (Microchip Technology, 2020a).

6.3. Flight Control System and Ground Control Station

A printed circuit board (shield) was designed in Eagle Autodesk for the purpose of housing the IMU, barometer, and other sensors in the aircraft. The shield was installed on the Arduino Mega 2560. Figure 6.1 shows the sensor shield.

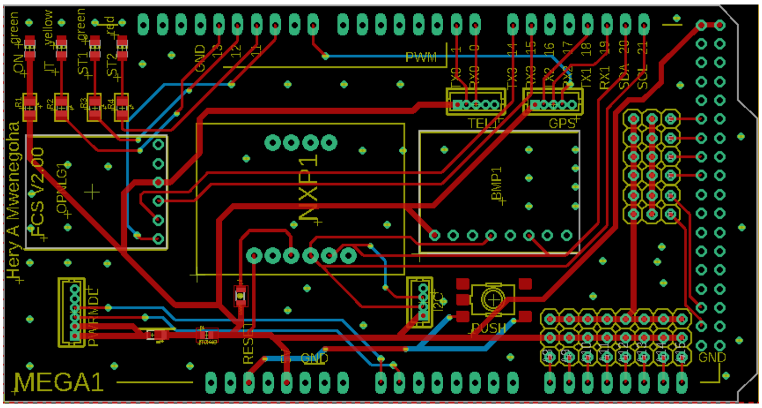


Figure 6.1. Sensor shield for the IMU, barometer and other sensors.

The completed flight control system board with the IMU, barometer and data logging module is shown in Figure 6.2. A custom, open-source flight control firmware[1] was loaded onto the board and used for guidance, navigation and control.

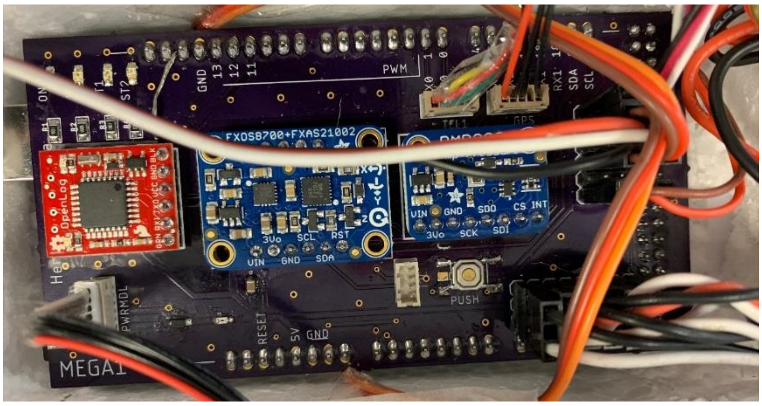


Figure 6.2. Flight control system board.

A custom, open-source ground control software[2] was used to communicate with the aircraft via a radio link. The ground control software was used to program the mission profile, change the aircraft's autopilot settings whenever necessary and log incoming telemetry from the aircraft. Figure 6.3 shows the custom ground control software running on a laptop with a 2.5 GHz Core i5-7200 CPU and 8 GB of RAM.

[1] https://github.com/HeryMwenegoha/PolarisAir
[2] https://github.com/HeryMwenegoha/PolarisGround

Figure 6.3. Ground control software.

6.4. Aerodynamic Model

The TCDVM requires an accurate set of model parameters for it to work effectively well. Aircraft characterisation usually involves laborious calibration routines requiring significant time, effort and cost. In this research, a geometry-based routine has been used to estimate the aerodynamic properties of the experimental aircraft due to its simplicity and the availability of open-source tools. Geometry-based techniques, and more generally empirical techniques, are fast and cost-effective, ideal for small teams and low-cost applications. However, the obtained aerodynamic coefficients from these techniques can be different from the actual coefficients of the aircraft due to the limitations and simplifications in the empirical models. To increase our confidence in the obtained aerodynamic coefficients, the aircraft is further characterised using wind tunnel testing and full-scale oscillation tests.

6.4.1. Wind Tunnel Testing

The aerodynamic characteristics of the aircraft are obtained using the AF100 open-circuit subsonic wind tunnel with a 305 mm x 305 mm x 600 mm closed test section, as shown in Figure 6.4 (left). This is a low-speed wind tunnel with a maximum operating speed of 36 m/s. The tunnel includes the AFA3 three-component force balance, which contains load cells used to measure the aerodynamic forces, lift (up to 100 N), drag (up to 50 N), as well as pitching moment (up to 2.5 Nm), exerted on a model. The aerodynamically designed effuser (cone) linearly accelerates the air entering the tunnel. The air passes through a grill before entering the diffuser and variable axial fan. The control and instrumentation unit includes manometers connected to a pitot-static tube in the test section to show pressure.

The aircraft used in the investigation had a wingspan of 1.4 m. Due to the limited size of the test section, the aircraft was scaled by a factor of three. To minimise the effects of tunnel walls on the airflow, a rule of thumb is to have the maximum span of the aircraft or its model be less than 80% of the tunnel width (Barlow, Rae and Pope, 1999). Since only the aerodynamic coefficients are of interest, only one half of the scaled model of the aircraft was used. This allowed fitting the half-scaled model in the test section, as shown in Figure 6.4 (right). To account for any scaling between the model used in wind tunnel testing and the aircraft used during flight tests, the wind tunnel was operated at a speed of 36 m/s. This was equivalent to 12 m/s on the full-scale aircraft used during flight tests. By matching the Reynolds number of the full-scale aircraft with that of the half-scaled model, the aerodynamic forces on both platforms will be the same provided that the fluid, its temperature and free-stream pressure remains the same.

Figure 6.4. AF100 subsonic wind tunnel (left) and the scaled model in the test section (right).

The scaled model was 3D printed, sanded and painted to reduce the surface roughness. Only the static aerodynamic coefficients were obtained from wind tunnel testing. For a detailed description of the errors encountered in wind tunnel testing, such as solid and wake blockages, the reader is directed to Mwenegoha and Jabbal (2013).

6.4.2. Full-Scale Oscillation Testing

The moment of inertia of the full-scale UAV is determined using the compound and bifilar pendulum setup shown in Figure 6.5. A compound pendulum setup is used to characterise the moment of inertia about the longitudinal axis and lateral axis (I_{xx}, I_{yy}) while the bifilar setup is used to characterise the moment of inertia about the normal axis (I_{zz}).

Figure 6.5. Compound pendulum (left) and bifilar torsion pendulum setup (right).

By oscillating the aircraft about the longitudinal or lateral axis, one can obtain its moment of inertia measured at the point of rotation around the same axis. A simplified differential equation for moments can be used using small-angle approximations, which is the equation for a harmonic oscillator. By applying the same principle to an assemblage consisting of a support frame and an aircraft, successive measurements of the period of oscillations are recorded, and the compound moment of inertia is estimated by (Junos, Mohd Suhadis and Zihad, 2014):

$$I_{xx,yy\ assemblage} = \frac{T^2}{4\pi^2} mgl \tag{6.1}$$

The moment of inertia of the aircraft about a specific axis is obtained by subtracting the moments of inertia of the support frame and the extra moment due to displacement of its centre of gravity.

$$I_{xx,yy\ aircraft} = \frac{T_T^2}{4\pi^2} m_T g l_T - \frac{T_{su}^2}{4\pi^2} m_{su} g l_{su} - m l_a^2 \tag{6.2}$$

where: T_T, T_{su} represent the period of oscillation for the assemblage and support frame, respectively (in seconds),
m_T, m_{su}, m represent the mass of the assemblage, support frame, and aircraft, respectively (in kilograms),
l_T, l_{su}, l_a is the distance from the pivot point to the centre of gravity of the assemblage, support frame and aircraft, respectively (in metres).

The yaw moment of inertia, I_{zz}, is determined using the bifilar torsion pendulum setup by subtracting the moment of inertia of the support frame from the moment of inertia of the assembly given by (Junos, Mohd Suhadis and Zihad, 2014):

$$I_{zz\ aircraft} = \frac{T_T^2}{16\pi^2} m_T g \frac{a^2}{L} - \frac{T_{su}^2}{16\pi^2} m_{su} g \frac{a^2}{L} \tag{6.3}$$

In this research, fifty oscillations are used to characterise the moments of inertia. In theory, the greater the number of oscillations, the lower the error due to the operator, but a damping factor makes the oscillations more difficult to perceive in practice.

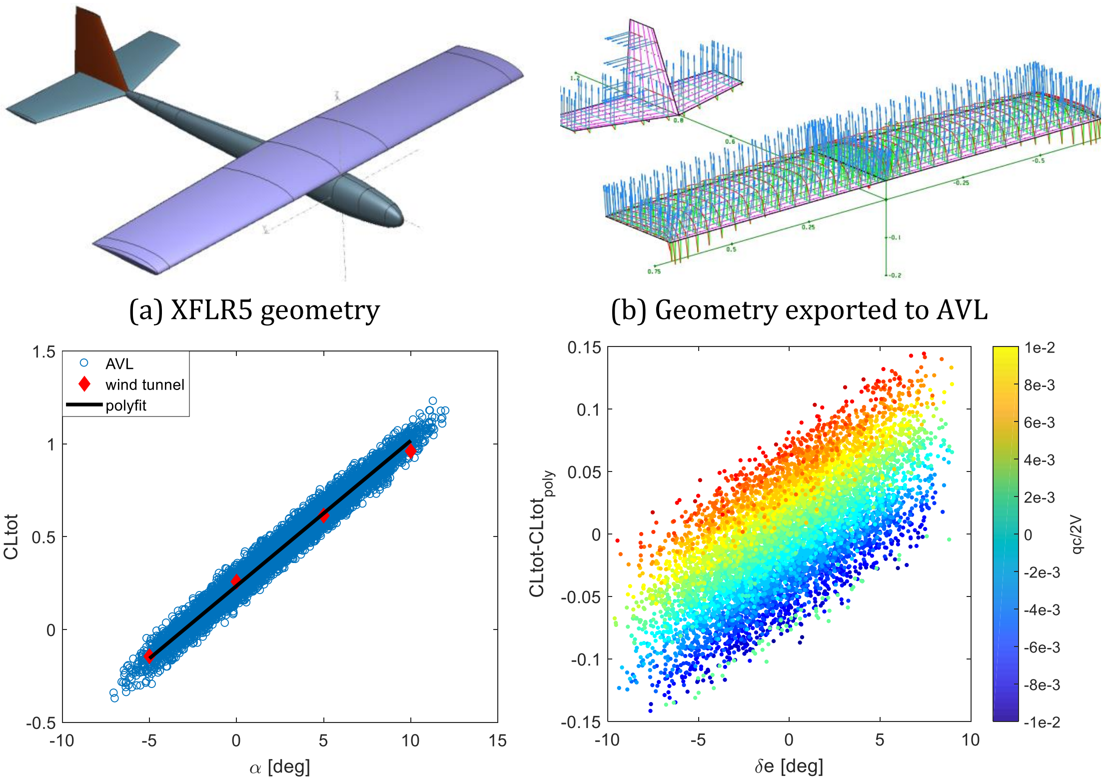
6.4.3. Geometry Based Approach

Methods that fall within this category use empirical and theoretical models and the aircraft geometry to characterise the aircraft. This method is simple, fast, and ideal for low-cost applications. The aerodynamic parameters are estimated using an open-source potential flow solver, Athena Vortex Lattice[3] (AVL), which provides values within 20% of the actual parameters (Klöckner, 2013). AVL uses the vortex lattice method to estimate the aerodynamic coefficients of the aircraft. The wing and other surfaces are modelled as a set of thin lifting panels. Each panel contains a single horse-shoe vortex with a bound vortex located at the panel

[3] http://web.mit.edu/drela/Public/web/avl/

quarter-chord position and two trailing vortex lines shed from each end. A zero flow condition is defined normal to the surface, and the velocity is assumed to contain a component of the free stream velocity and an induced component. The induced component is a function of the strengths of all the vortex panels on the surface.

To obtain the aerodynamic coefficients, first, the geometry is defined using a freely available aerodynamic analysis tool, XLFR5[4] and then, exported to AVL. Monte Carlo simulations are used to investigate the effects of different input variables $(\alpha, \beta, \bar{\omega}_x, \bar{\omega}_y, \bar{\omega}_z, \delta_\alpha, \delta_e, \delta_r)$ on the aerodynamic coefficients. Figure 6.6 shows the workflow used to estimate the coefficients.



(a) XFLR5 geometry (b) Geometry exported to AVL

(c) Polynomial fitting (d) Reduced polynomial fitting

Figure 6.6. Aircraft characterisation workflow.

The XFLR5 geometry (a) is exported to AVL (b) without the fuselage for aerodynamic analysis. Eight input variables are used in the potential flow solver to generate solutions (c). In Figure 6.6 (c), the total lift coefficient is plotted against the angle of attack. The residuals of the polynomial fitting (d) are used to determine appropriate monomials. In Figure 6.6 (d), the residual lift coefficient is plotted against the aileron deflection and the normalised pitch rate (which shows a weaker dependency).

Details of the aircraft geometry and mass properties are given in Table 6.1. It should be noted that the moment of inertia terms are also obtained from XFLR5 following the geometry definition and mass input. The average error of these

[4] http://www.xflr5.tech/xflr5.htm

terms is found to be within 7% of reference values available from full-scale oscillation tests. The aircraft characterisation process using a combination of AVL and XFLR5 provides reasonable initial estimates that can be supplemented with wind tunnel data and full-scale oscillation tests if available.

Table 6.1. Aircraft properties.

Property	Description	Value	
m	Aircraft mass	2.17 kg	
S	Wing area	0.36 m^2	
b	Wingspan	1.40 m	
$\bar{c}$	Mean aerodynamic chord	0.26 m	
D	Propeller Diameter	0.30 m	
		XFLR5	Oscillation Tests
I_{xx}	Roll moment of inertia	0.12 kgm^2	0.139 ± 0.012 kgm^2
I_{yy}	Pitch moment of inertia	0.13 kgm^2	0.124 ± 0.010 kgm^2
I_{zz}	Yaw moment of inertia	0.24 kgm^2	0.274 ± 0.019 kgm^2

6.4.4. Drag Force Coefficient

The workflow presented in Figure 6.6 was used to characterise the aerodynamic properties of the aircraft. Figure 6.7 shows the drag coefficient plotted against the angle of attack and sideslip angle. In the figure, the blue points indicate the AVL solution, and the mesh shows the drag coefficient model fitted with the range of the data from the AVL solution with a coefficient of determination R^2 of 0.9706.

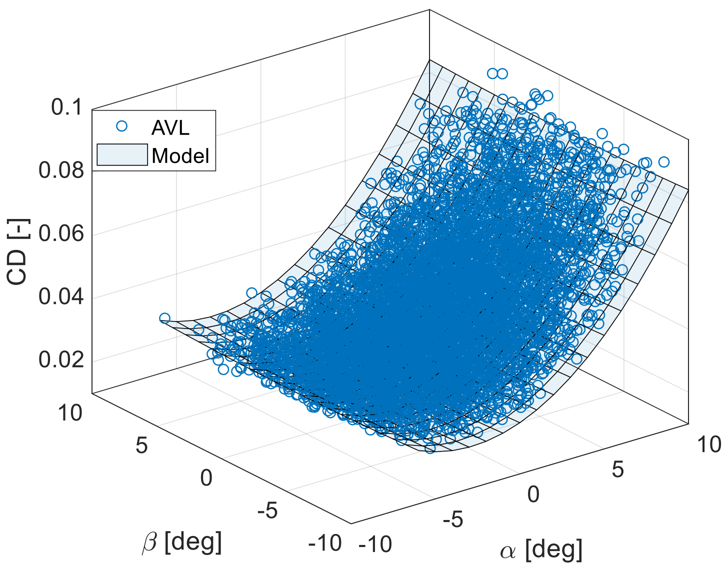


Figure 6.7. Drag force coefficient plotted against both angle of attack and sideslip angle.

These results reveal how well the AVL solution fits the drag coefficient model given in Equation (3.35). However, compared to available wind tunnel results, the AVL solution seems to have a significant offset from wind tunnel data, as shown in Figure 6.8. This is attributed to the missing drag contributions in the potential flow solution in AVL. The available wind tunnel results are not used in the quantitative assessment of the flight test results due to some limitations in wind tunnel testing. For instance, the wind tunnel results were obtained on a scaled model in a clean configuration (e.g. no landing gear). However, the wind tunnel data is used as a simple guide to the AVL solution through a qualitative assessment.

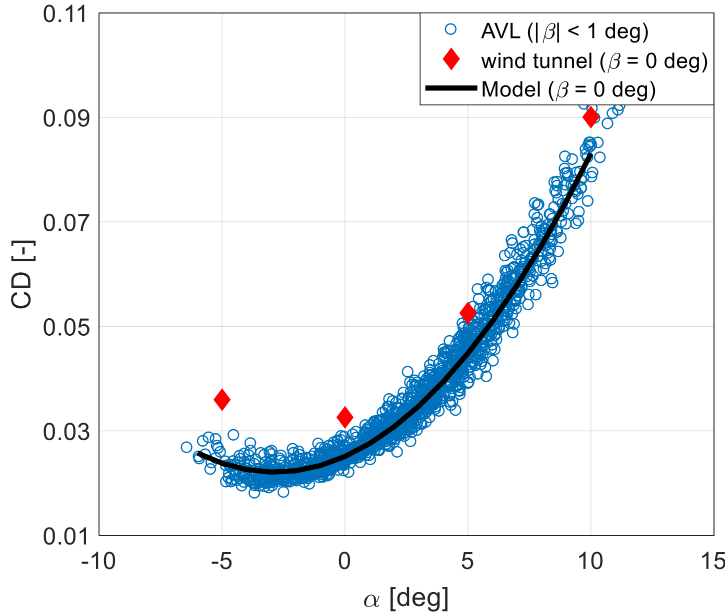


Figure 6.8. Drag force coefficient with angle of attack.

The drag coefficient term seems to show little variation with the sideslip angle, as shown in Figure 6.9. The AVL solution seems to underestimate the variation of drag coefficient with sideslip angle even though the pattern is similar to the wind tunnel results. Several factors could contribute to this, most notably are the missing contributions from the potential flow solution and the lack of a fuselage in the geometry. The model used to fit the AVL solution only considers two input variables (α, β) out of the eight variables used in simulation. It is possible that the missing contributions from other input variables have a significant impact on the drag coefficient even though this argument alone does not account for the difference between the wind tunnel data and the AVL solution.

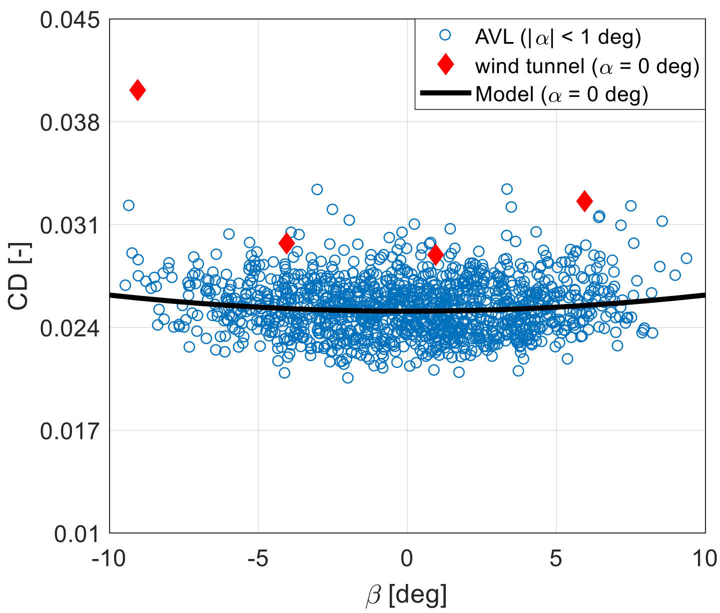


Figure 6.9. Drag force coefficient with sideslip angle.

6.4.5. Lift Force Coefficient

Figure 6.10 shows the lift coefficient plotted against the angle of attack and sideslip angle. In the figure, the blue points indicate the AVL solution, and the mesh shows the lift coefficient model fitted with the range of the data ($R^2 = 0.9724$). Results reveal how well the AVL solution fits the lift coefficient model given in Equation (3.33). The AVL solution shows good agreement with the limited wind tunnel data, as shown in Figure 6.11.

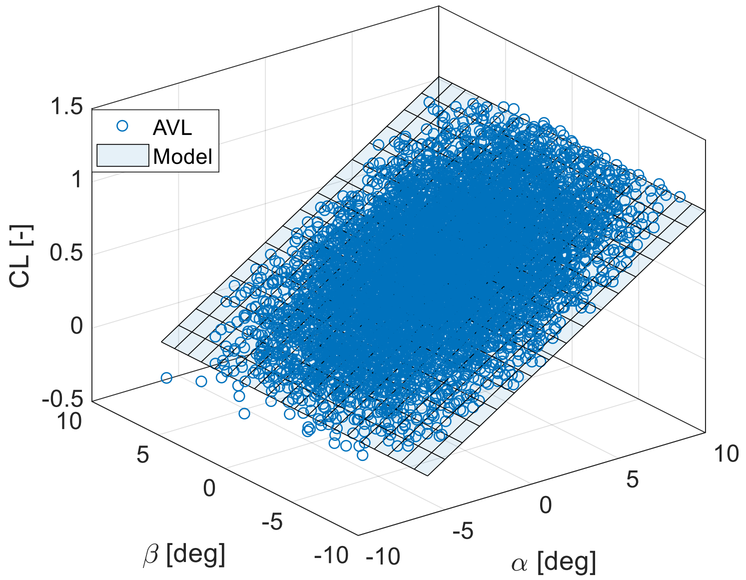


Figure 6.10. Lift force coefficient plotted against both angle of attack and sideslip angle.

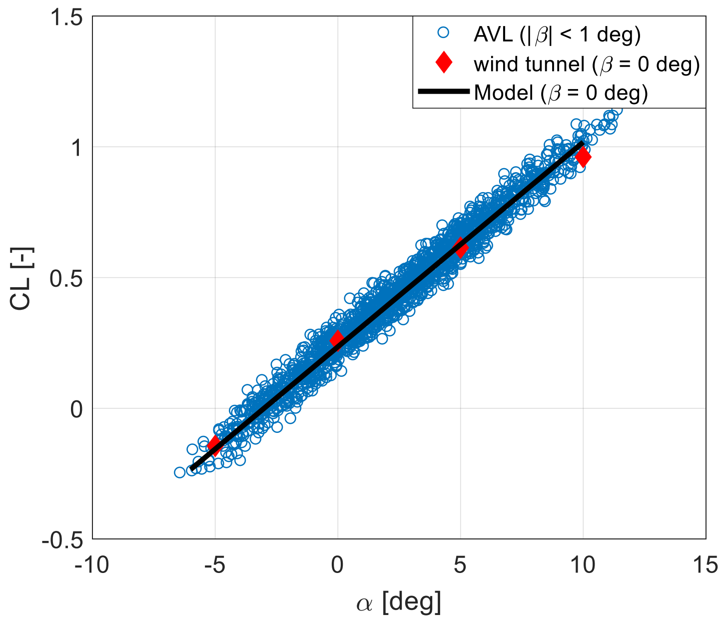


Figure 6.11. Lift force coefficient with angle of attack for the AVL solution, wind tunnel data and the fitted model.

6.4.6. Lateral Force Coefficient

Figure 6.12 shows the lateral coefficient plotted against the angle of attack and sideslip angle. Results show that the lateral force coefficient model given by Equation (3.34) fits the AVL solution well, even though the coefficient of determination (R^2) was only 0.625 for this model. The limited static wind tunnel data shows that the AVL solution underestimates the lateral force coefficient, as shown in Figure 6.13. This could be due to the lack of the fuselage in the AVL solution, leading to underestimating this term.

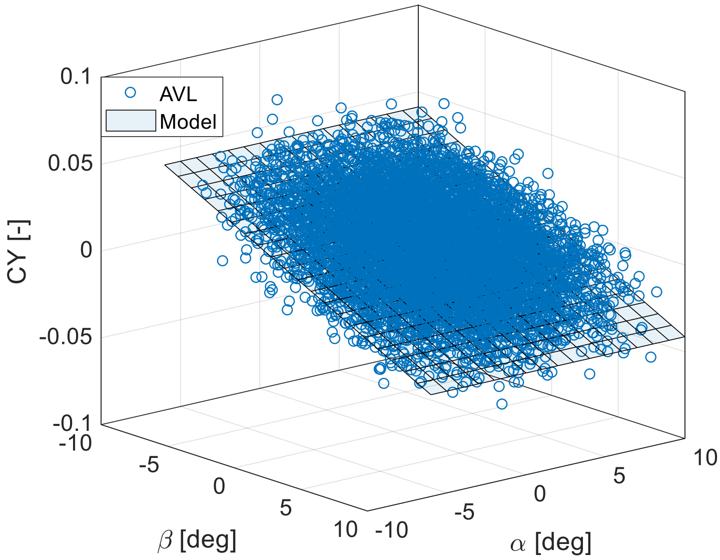


Figure 6.12. Lateral force coefficient plotted against both angle of attack and sideslip angle.

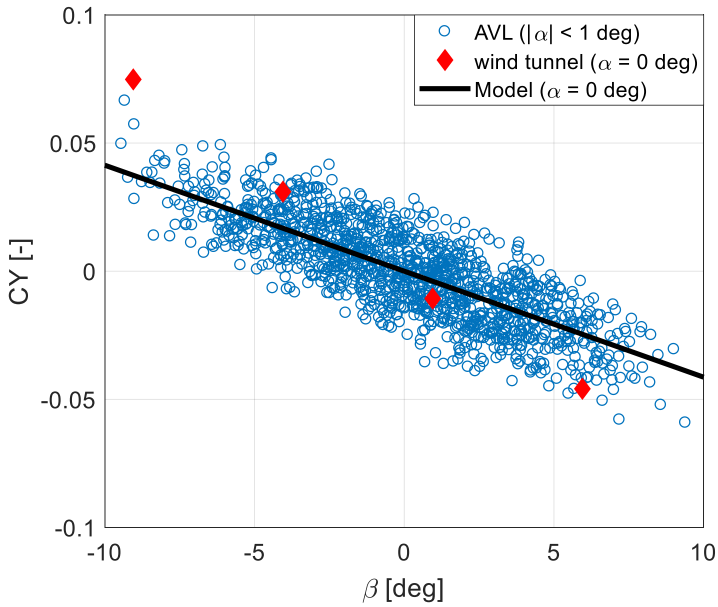


Figure 6.13. Lateral force coefficient with sideslip angle for the AVL solution, wind tunnel data and the fitted model.

6.4.7. Roll Moment Coefficient

Figure 6.14 shows the rolling moment coefficient for the AVL solution and fitted model given by Equation (3.36) with a coefficient of determination (R^2) of 0.9863.

Wind tunnel data for the rolling moment coefficient was not available. Therefore, a qualitative comparison cannot be made. However, the roll stability shown in the plot can be attributed to the slight dihedral on the aircraft.

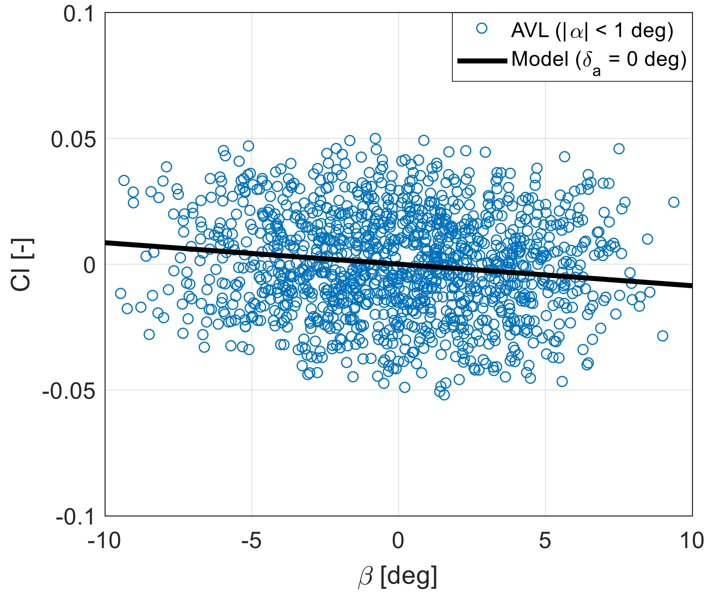


Figure 6.14. Rolling moment coefficient with sideslip angle for the AVL solution and the fitted model ($R^2 = 0.9863$).

6.4.8. Pitch Moment Coefficient

The static pitching moment coefficient results show that the AVL solution and the fitted model given by Equation (3.37) slightly overestimated the pitching moment slope compared to available wind tunnel results. However, the trend is very similar, both indicating that the aircraft has inherent longitudinal stability. Wind tunnel results seem to indicate a lower trim angle (α with $C_m = 0$) than the AVL solution, as shown in Figure 6.15.

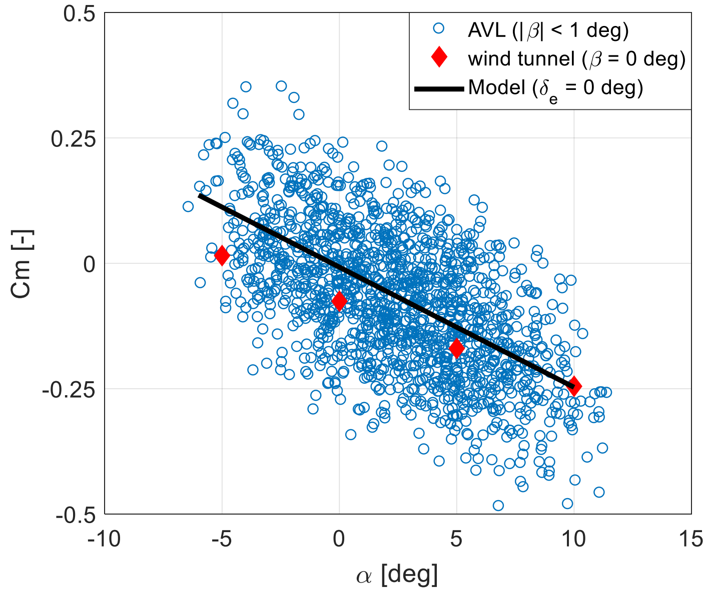


Figure 6.15. Pitching moment coefficient with angle of attack for the AVL solution, wind tunnel data and the fitted model ($R^2 = 0.9939$).

The lift coefficient will be less than zero for the wind tunnel results at the indicated trim angle. However, for the AVL solution, the estimated lift coefficient at its trim angle is positive. The centre of gravity for the AVL solution closely matched that of the actual aircraft. It is possible that the scaled model used in wind tunnel testing was slightly nose-heavy, causing the trim angle to be lower.

6.4.9. Yawing Moment Coefficient

Figure 6.16 shows the yawing moment coefficient results for the AVL solution, the fitted model given by Equation (3.38) and static wind tunnel testing results.

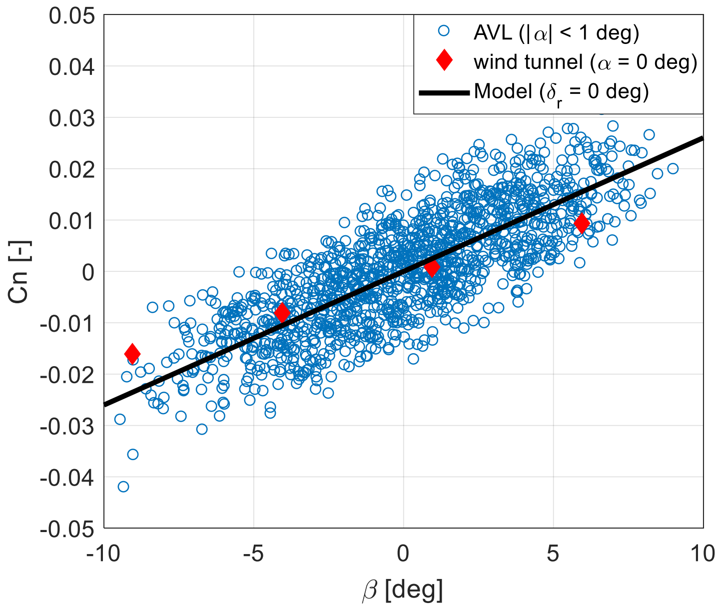


Figure 6.16. Yawing moment coefficient with sideslip angle for the AVL solution, wind tunnel data and the fitted model ($R^2 = 0.9930$).

Both the AVL solution and wind tunnel data show that the aircraft has directional stability. However, the AVL solution seems to slightly overestimate the yawing moment coefficient compared to wind tunnel results.

The estimated aerodynamic coefficients are presented in Section C.2 of Appendix C of this thesis.

6.5. Propulsion Model

This section characterises the propulsion model used on the aircraft and provides information on the variation in performance of each component in different conditions. The section determines the values for the thrust coefficient terms (for the VDM) and derives the commanded RPM using a combination of the pulse width modulated (PWM) signal from the flight control system, current through the motor and voltage measurements.

6.5.1. Electronic Speed Controller

The propulsion system on most fixed-wing UAVs consists of a brushless DC (BLDC) motor with a propeller. The BLDC motor is driven by an electronic speed controller (ESC), which transforms the input DC voltage from the battery into three-phase electricity, as shown in Figure 6.17. A PWM signal is used to adjust the speed of the motor and propeller. Sendobry (2014) showed that the efficiency of the ESC was a function of the PWM signal and current.

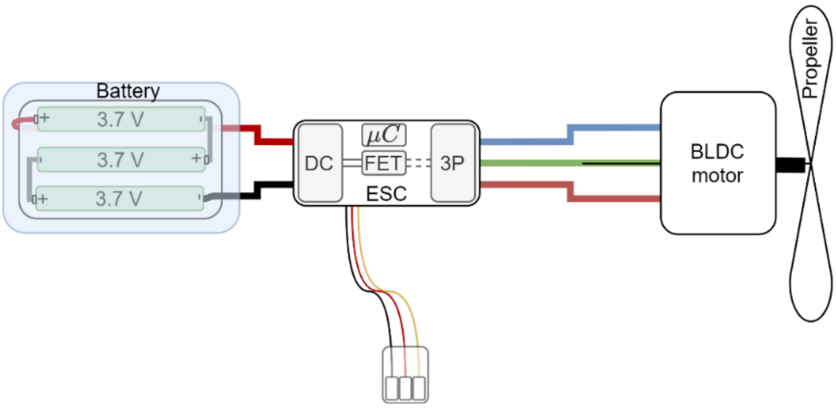
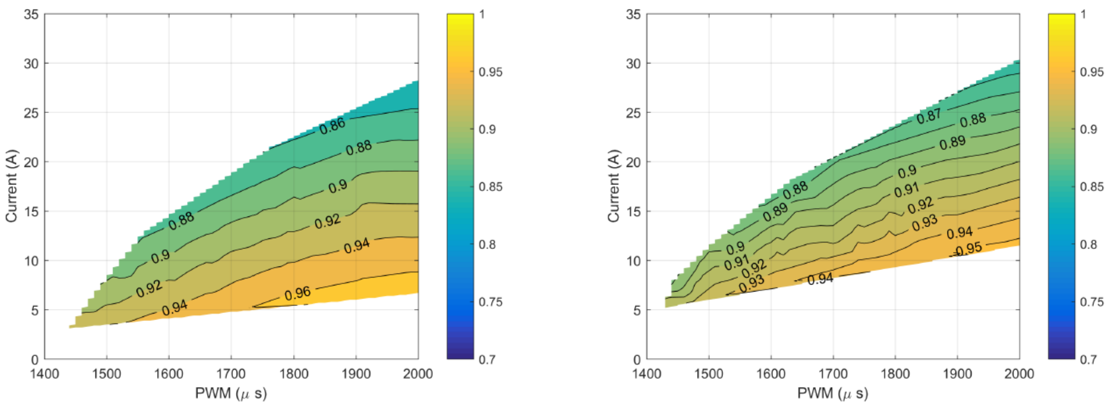


Figure 6.17. Propulsion model used in UAV.

Gong, Macneill and Verstraete (2018) conducted a series of tests on Rimfire .55 480 Kv motor driven by a Castle Phoenix 120 HV ESC over a range of RPMs and demonstrated how the efficiency of the ESC varied at different voltages. Their results can be seen in Figure 6.18. They later modelled the ESC using a bi-linear equation fitted to the efficiency data as a function of the PWM signal and current given by:

$$\eta_{esc} = \alpha_{esc} \cdot PWM + \beta_{esc} \cdot I + \gamma_{esc} \tag{6.4}$$

where: PWM is the input pulse-width-modulated signal (μs),
I is the input current to the ESC (A),
α_{esc} (μs^{-1}), β_{esc} (A^{-1}), γ_{esc} are model parameters.



(a) 13 V (b) 15 V

Figure 6.18. A Castle Phoenix Edge HV 120 ESC Efficiency with a Rimfire .55 Motor 480 Kv motor. The black lines with numbers indicate the efficiency of the ESC (Gong, Macneill and Verstraete, 2018).

The results were similar to the ones obtained by Sendobry (2014). The values of model parameters ($\alpha_{esc}, \beta_{esc}, \gamma_{esc}$) need to be determined at different voltages because they vary with the input voltage. Furthermore, low-cost BLDC and ESC derived from hobby equipment can have considerable variation in efficiency and performance between manufacturers. So the determined parameters will only be useful for the specific setup.

In this research, a Dynamic 60 A electronic speed controller is used and connected to a 3S Overlander battery (11.1 V) with a capacity of 2200 mAh and a 35C constant discharge rate.

6.5.2. Brushless DC Motor

The efficiency of a BLDC motor is given by the ratio of the output mechanical power to the input electrical power.

$$\eta_{motor} = \frac{\tau \cdot \omega}{E \cdot I} \tag{6.5}$$

where: E is the input voltage,
ω is the rotation speed (rad/s),
τ is the motor torque (Nm).

In practice, the efficiency of a BLDC motor and, by extension, an ESC can be measured with a 3-phase power analyser and a dynamometer. However, the equipment can be expensive, and the process time-consuming. For a quick and simple estimate of motor efficiency, different models can be used with varying degrees of complexity. A popular model is the three-constant model (Gong, Macneill and Verstraete, 2018). The three model parameters, $I_0, k_V,$ and R_m, represent the no-load current (A), the voltage constant (RPM/V) and the internal motor resistance (Ω), respectively. These are usually obtained from manufacturers. However, these parameters also tend to vary with the input voltage (Gong, Macneill and Verstraete, 2018). Another simple approach is to assume the BLDC motor has 100 % efficiency even though this will not be true in practice.

In this research, a Tornado Thumper 4240/10 890 kV BLDC motor is used and is shown in Figure 6.19. This motor weighs 140g, has a length of 60 mm and a width of 42.5 mm. It has a continuous power rating of 540 W and a peak power rating of 650 W. The recommended propeller size range is 10 inches x 6 inches – 14 inches x 8 inches (diameter x pitch). Further, it is assumed that the BLDC motor efficiency is constant within a small voltage range. This simplifies the overall system efficiency of the ESC and BLDC motor and allows the motor efficiency to be estimated with the parameters given in Equation (6.4).

Figure 6.19. Tornado Thumper 4240/10 890 Kv V2 motor (*gliders*, 2014).

6.5.3. Propeller

Most small, mass-market UAVs use fixed pitch propellers. Generalised estimations are usually used by model pilots based on the ratio of the mean pitch of the propeller to its diameter (H/D) to estimate the thrust coefficient. This ratio can also be derived from static thrust measurements. The pitch to diameter ratio can be determined from static thrust measurements by matching the static thrust coefficient to defined H/D curves.

The efficiency of the propeller is given by the ratio of the power supplied to the useful power output.

$$\eta_{prop} = \frac{P_{out}}{P_{in}} \tag{6.6}$$

The useful output power is given by:

$$P_{out} = F_T \, V_0 \tag{6.7}$$

where: V_0 is the forward velocity (m/s).

The power supplied to the propeller is given by:

$$P_{in} = \rho n^3 D^5 C_P \tag{6.8}$$

where: C_P is the power coefficient.

Using Equation (3.31) for the thrust force (F_T), the propeller efficiency then simplifies to:

$$\eta_{prop} = J \cdot \frac{C_{F_T}}{C_P} \tag{6.9}$$

Both the thrust and power coefficients are functions of the forward speed, propeller rotation rate, air density, Reynolds number and the tip Mach number (Balmer, 2015).

In this research, an Advanced Precision Composites (APC) thin electric 12-inch x 6-inch propeller is used on the aircraft. Data for this propeller can easily be obtained from the manufacturer and used to estimate the thrust coefficient. Data from the manufacturer is compared to experimental data to provide a qualitative and quantitative estimate of the accuracy. The experimental results from Brandt and Selig (2011) for an APC thin electric 11-inch x 8-inch propeller have been extracted and compared to the data from the manufacturer even though this propeller is not an exact match to the propeller used on the aircraft. Essentially this is not a problem because the results of this comparison are only used to assess the accuracy of the simulated data qualitatively. Figure 6.20 shows the variation of thrust coefficient with the advance ratio at 3000 RPM and 6000 RPM.

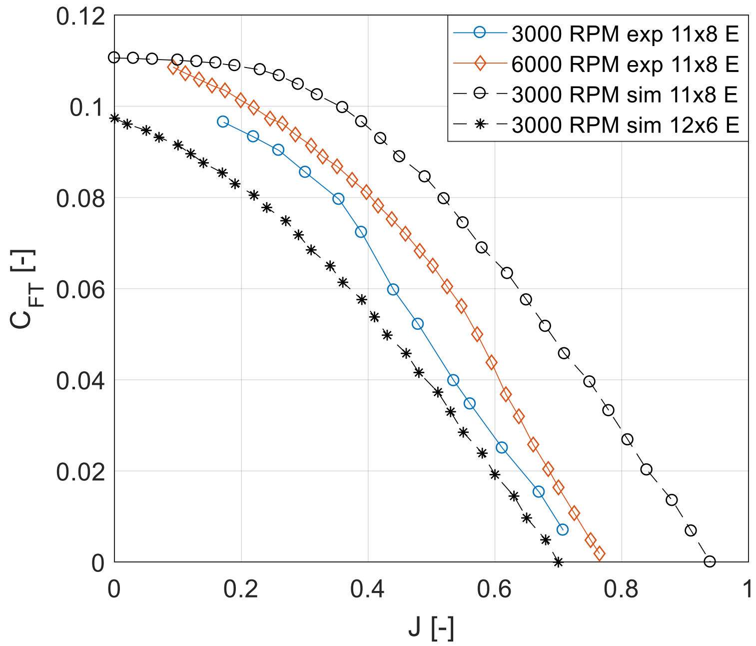


Figure 6.20. Thrust coefficient variation with simulated and experimental data.

Experimental results show that the thrust coefficient has a quadratic relationship with the advance ratio, which fits the model. The agreement of the experimental results with data from the manufacturer for the static thrust coefficient is within 3%. Experimental results also show that the first-order and second-order thrust coefficient terms vary close to the maximum efficiency region (around $J = 0.5$). This variation is not reflected in the simulated data from the manufacturer for both the APC thin electric 11-inch x 8-inch and 12-inch x 6-inch propellers, as shown in Figure 6.21. Based on the APC thin electric 11-inch x 8-inch propeller, the error in the first-order and second-order terms in the simulated data is more than 25% compared to the experimental results.

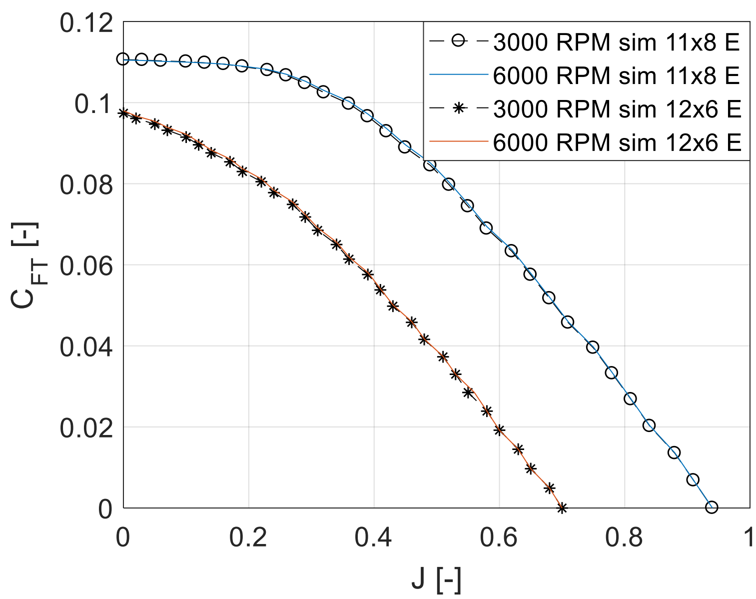


Figure 6.21. Thrust coefficient variation for the APC 11x8 E and 12x6 E.

6.5.4. Commanded RPM

The variation of the ESC efficiency with the PWM signal and current enables the commanded RPM to be determined. This is essential because a low-cost propulsion unit will usually not include a motor/propeller speed sensor. Therefore, the speed of the propeller needs to be inferred from other sensors onboard the aircraft and the commanded PWM signal. Most UAVs will include a power module that usually outputs current and voltage measurements. The commanded propeller speed can be inferred from a combination of these measurements, the PWM signal, and the forward velocity as given in the component efficiencies. This estimation will be suboptimal, especially if it does not consider the variation of different coefficients with voltage, torque load and propeller speed. Using ground-based measurements from a tachometer (measuring the propeller speed) and a wattmeter, the measured propeller speed can be compared to the estimated speed. Figure 6.22 shows the comparison of the measured and estimated speed.

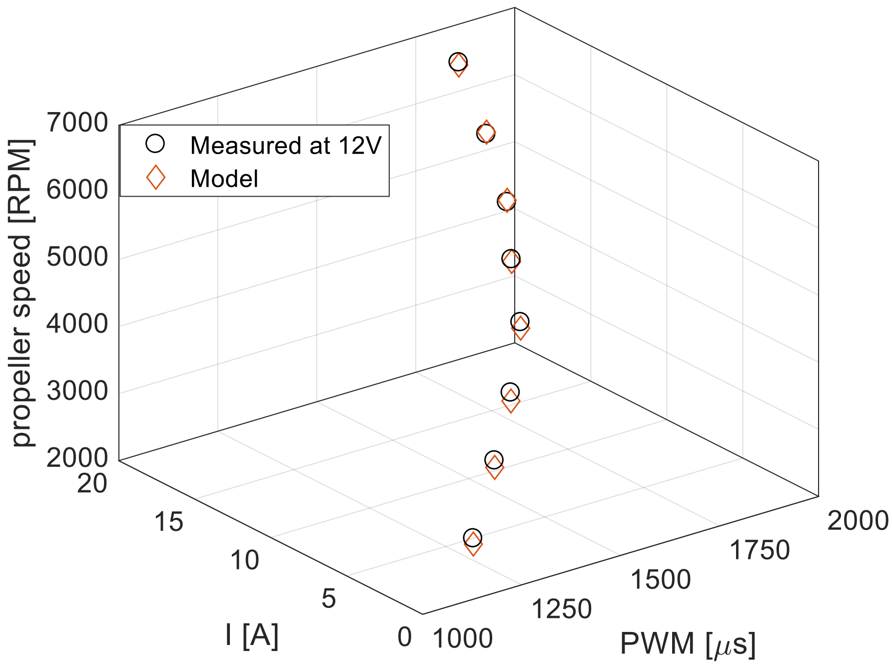


Figure 6.22. A comparison of measured and estimated propeller speed ($R^2 = 0.946$).

The model follows the trend very well, and the difference between the measured and estimated speed at RPMs lower than 4000 is less than 50 RPM.

6.6. Test Flight

The test flight was conducted on the 12th of September 2019 around 1500hrs at Hucknall Model Flying Club, Nottinghamshire, UK (53.048459° N, 1.291661° W). A LeicaGS10 unit, shown in Figure 6.23, was used as the ground reference (base) GNSS receiver to derive a post-processed kinematic (PPK) position solution. The data on the base receiver was logged at 4 Hz.

Figure 6.23. A LeicaGS10 unit used as the base receiver.

Three u-blox NEO-M8T GNSS receivers (GM, G1 and G2) were used on the aircraft to provide three independent position solutions and two baseline solutions ($b1$ and $b2$) for precise attitude determination, as can be seen in Figure 6.24.

Figure 6.24. The Riot V2 with three NEO-M8T GNSS modules.

The flight consisted of six segments, take-off, climb, loiter, autonomous navigation, descent, and land. A human pilot flew the UAV for the first three segments and the last two segments. The first three segments made up the first 200 seconds of the flight. Figure 6.25 shows the height profile during the flight test and the individual segments that have been grouped and colour-coded. The profile has been partitioned into three groups. The first group includes take-off, climb and loiter (T/O-CLB-LOT). The second group includes autonomous navigation (AUTO), and finally, the third group includes descent and landing (DESC-LAND).

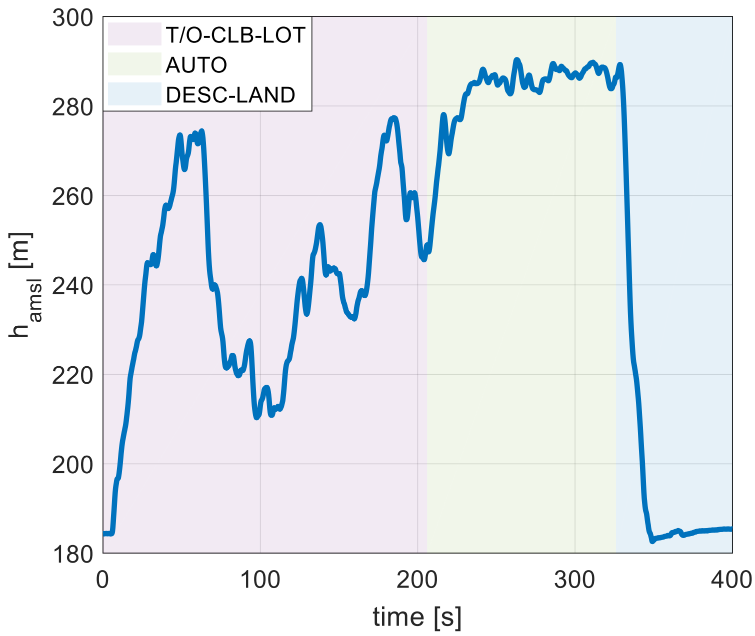


Figure 6.25. The height profile for the flight test.

In loiter mode, the pilot performed a series of manoeuvres such as S-turns, deep-dives, steep climbs to excite different modes. In a VDM-based scheme, manoeuvres that excite different modes are important because they allow the IMU errors and VDM parameters to be observable, as explained in Chapters 4 and 5. After this segment, the autopilot was engaged, and the aircraft flew a pre-programmed mission for 120 seconds. The entire flight lasted 400 seconds. Figure 6.26 shows a partial 2D position plot of the UAV during the flight test. The plot shows the autonomous navigation segment in green and the descent and landing segment in blue.

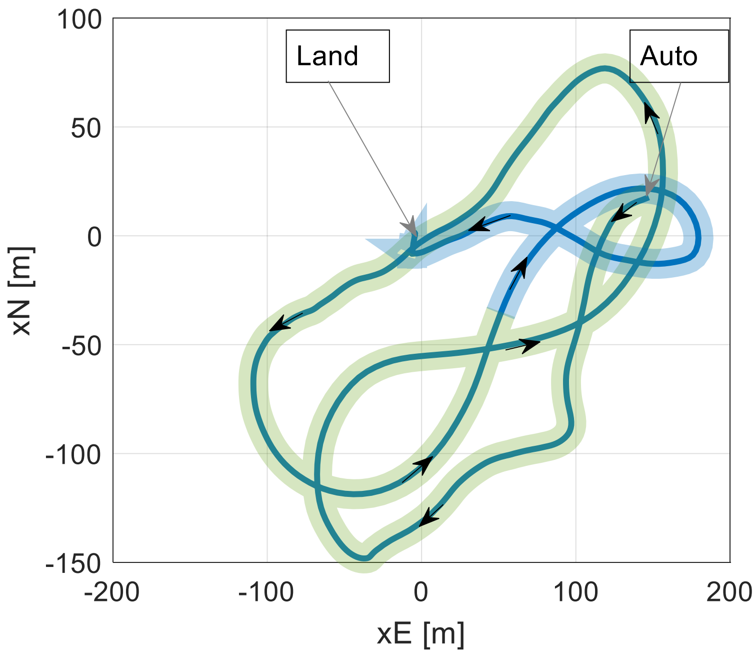


Figure 6.26. A partial 2D position.

During the flight, IMU measurements and control inputs were logged at 20Hz on the FCS logger, whilst GNSS data was logged at 4Hz on independent data loggers for each module. With an elevation masking angle of 15 degrees, the GPS satellites visible during the flight are shown in Figure 6.27 alongside the receiver's estimate of the position dilution of precision (PDOP).

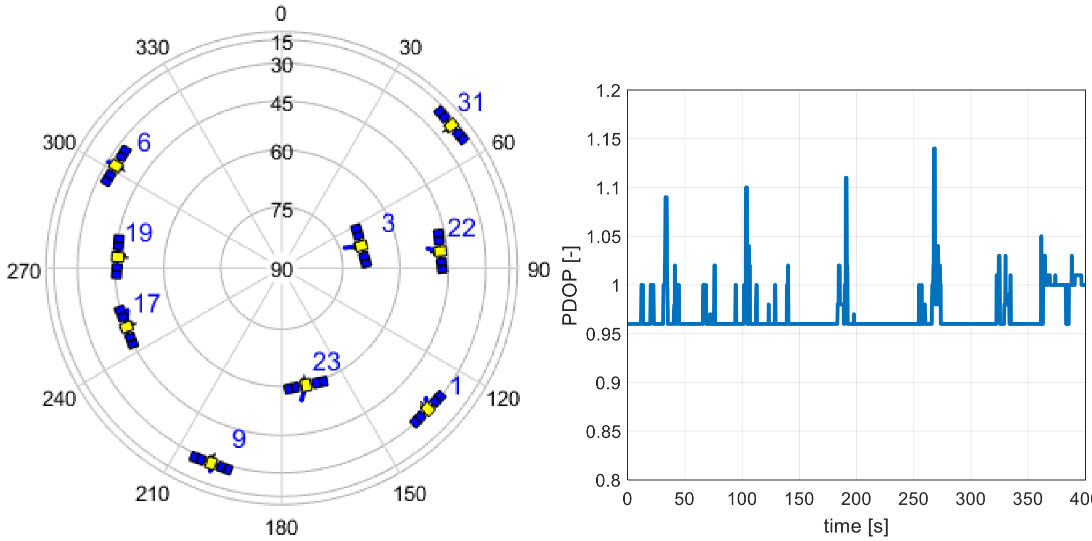


Figure 6.27. A skyplot showing the GPS satellites visible during the flight (left); the PDOP when tracking GPS and GLONASS satellites (right).

6.7. Post-Processing

This section describes the post-processing of IMU and GNSS data. The first section describes the characterisation of the IMU noise, and the second section describes the derivation of the reference position, velocity and attitude solution using the logged data.

6.7.1. IMU Noise

Different noise sources affecting the IMU need to be characterised before the measurements can be used in an integration architecture. The modelling of inertial sensors is a challenging task, and in most practical cases, it is performed by tuning the integration architecture using available specifications. Usually, the manufacturer provides laboratory calibrated values, which could significantly differ from values seen during operations. Therefore, it is important to determine the IMU stochastic properties based on the operating conditions. In this research, the Allan variance has been used to identify and extract noise parameters for stochastic modelling. Optimal performance requires identifying the noise parameters at different temperature points and operating conditions (El-Diasty and Pagiatakis, 2009). However, this requires significant time, effort and cost. Some explanation of the technique is given in Section C.3 of Appendix C, and the reader is also directed to referenced text for a detailed review of the technique (El-Sheimy, Hou and Niu, 2008).

To determine the Allan standard deviation of the three gyroscopes and accelerometers within the NXP-9DOF IMU, static data was gathered from the IMU at a sampling frequency of 100 Hz for one hour. The Allan standard deviations of the three gyroscopes and accelerometers within the NXP-9DOF IMU are shown in Figure 6.28 and Figure 6.29, respectively. The errors in estimating the Allan standard deviations as a result of operating on clusters of different lengths are also shown in Figure 6.28 and Figure 6.29. White noise (in the region where the slope is -1/2) seems to dominate most of the short clusters, while bias instability (in the region where the slope is 0) and rate random walk (in the region where the slope is +1/2) seem to dominate long cluster times. The *x*-axis gyroscope and *z*-axis accelerometer seem to exhibit more white noise.

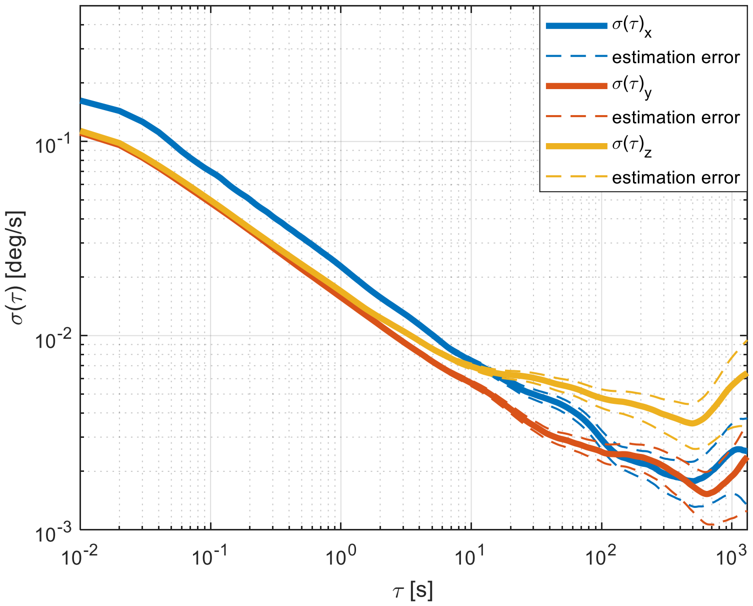


Figure 6.28. Allan standard deviation plot for the gyroscopes sampled at 100Hz for one hour.

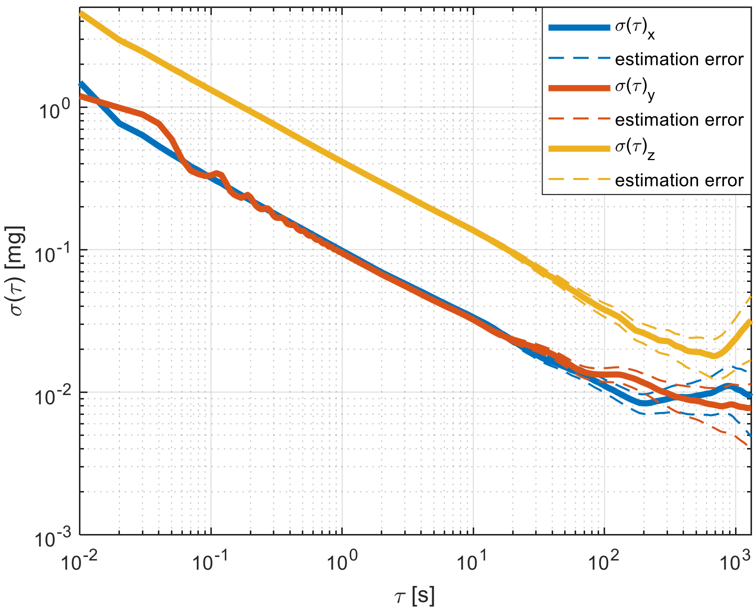


Figure 6.29. Allan standard deviation plot for the accelerometers sampled at 100 Hz for one hour.

The identified stochastic properties are summarised in Table 6.2 and Table 6.3. It should be noted that some coefficients for the accelerometers in the x-axis and y-axis could not be easily identified, and their values have not been filled. It is possible that the values could be identified with a longer duration, but this was not investigated. Instead, the values for the accelerometer in the z-axis were used as the limiting case.

Table 6.2. Allan variance coefficients for the three gyroscopes.

Description	**Coefficient**	**Gyro-x**	**Gyro-y**	**Gyro-z**
Angle random walk	N($°/\mathrm{hr}/\sqrt{\mathrm{Hz}}$)	81.684	57.42	61.74
Bias Instability	B ($°/\mathrm{hr}$)	9.49	8.2896	19.1928
Rate random walk	K ($°/\mathrm{hr}/\sqrt{\mathrm{s}}$)	0.4662	0.3625	1.1088

Table 6.3. Allan variance coefficients for the three accelerometers.

Description	**Coefficient**	**Accel-x**	**Accel-y**	**Accel-z**
Velocity random walk	N ($\mathrm{mg}/\sqrt{\mathrm{Hz}}$)	0.0981	0.0946	0.4172
Bias Instability	B (mg)	-	-	0.0269
Acceleration random walk	K ($\mathrm{mg}/\sqrt{\mathrm{s}}$)	-	-	0.0012

6.7.2. Reference Solution

A post-processed kinematic position solution using data collected from three u-blox NEO-M8T receivers was used with a loosely coupled integration architecture to provide a reference navigation solution to assess the performance of the proposed scheme. This solution is derived from double differenced carrier-phase observables. For relatively short baselines between a roving receiver (r) and a base receiver (b), as is the case with the UAV used in this research, the double differenced carrier-phase observations are formulated as:

$$\lambda\phi_{rb}^{\mathrm{jk}} = \rho_{rb}^{jk} + T_{rb}^{jk} - I_{rb}^{jk} + \lambda N_{rb}^{jk} - \epsilon(\phi_{rb}^{jk}) \tag{6.10}$$

where: λN_{rb}^{jk} is the integer ambiguity term (in metres),
ρ_{rb}^{jk} is the double differenced geometric range (in metres),
T_{rb}^{jk} and I_{rb}^{jk} represent the double differenced tropospheric and ionospheric delay, which for short baselines (less than 1 km) are highly correlated, and therefore, their differences are negligible (Giorgi and Teunissen, 2012).

The equation can be represented in a linearised functional model given by:

$$y = \lambda\phi_{rb}^{\mathrm{jk}} - \lambda\hat{\phi}_{rb}^{\mathrm{jk}} = -\left(e_r^j - e_r^k\right)^T \cdot \Delta r + \lambda N_{rb}^{jk} \tag{6.11}$$

where: Δr is the baseline vector (in metres),
$\hat{\phi}_{rb}^{\mathrm{jk}}$ is the estimated double differenced carrier-phase (in cycles).

The equation is then solved in a least-squares sense to obtain a float baseline solution. It is possible to formulate the integer ambiguity term using the single difference (between receivers) carrier-phase bias terms to avoid bothersome hand-over handling of reference satellites. This reads:

$$\lambda N_{rb}^{jk} = \lambda(B_{rb}^{j} - B_{rb}^{k}) \tag{6.12}$$

and the single differenced carrier-phase bias to a satellite k is given by:

$$B_{rb}^{k} = \left(\phi_{r,0} - \phi_0^k + N_r^k\right) - (\phi_{b,0} - \phi_0^k + N_b^k) \tag{6.13}$$

where: $\phi_{r,0}$ and $\phi_{b,0}$ represent the initial phase of the receiver (r, b) generated carrier signal (replica) at an initial time t_0,
ϕ_0^k is the initial phase of the satellite transmitted carrier signal at an initial time t_0.

To resolve the integer ambiguity terms (N) from the float carrier-phase biases, an integer least square (ILS) problem is formulated as:

$$\breve{N} = \min_{N}\left(\left(\widehat{N} - N\right)^T Q_{\widehat{N}}^{-1}\left(\widehat{N} - N\right)\right) \text{ with } N \epsilon Z \tag{6.14}$$

where $Q_{\widehat{N}}$ represents the covariance matrix of the float ambiguity terms. Once the solution $\breve{N}$ has been obtained, the residual $(\widehat{N} - \breve{N})$ is used to adjust the float baseline solution $\Delta\hat{r}$ to get the fixed baseline solution $\Delta\check{r}$.

$$\Delta\check{r} = \Delta\hat{r} - Q_{\Delta\hat{r}\widehat{N}} Q_{\widehat{N}}^{-1}(\widehat{N} - \breve{N}) \tag{6.15}$$

and the variance-covariance matrix for the baseline vector is adjusted accordingly.

$$Q_{\Delta\check{r}} = Q_{\Delta\hat{r}} - Q_{\Delta\hat{r}\widehat{N}} Q_{\widehat{N}}^{-1} Q_{\Delta\hat{r}\widehat{N}} \tag{6.16}$$

The Least-squares Ambiguity Decorrelation Adjustment (LAMBDA), is a well-known efficient search strategy that shrinks the integer search space and performs a skilful search procedure in the transformed space (Teunissen, 1994). A validation test is performed on the integer vector solution. Usually, a simple ratio test is used. This is defined as the ratio of the weighted sum of the squared residuals of the second-best solution to one by the best solution. The availability of a baseline constraint can also be used to validate the returned integer vector solution.

In this research, the freely available tool, RTKLIB, has been used to post-process raw GNSS observables from three independent NEO-M8T receivers on the aircraft (Takasu and Yasuda, 2013). The three receivers are used to generate independent baseline solutions with respect to a LeicaGS10 GNSS module. Further, two baseline solutions are generated using the master receiver (GM) as the base receiver. The configuration settings for the three NEO-M8T receivers are given in Section C.4 of Appendix C. RTKLIB uses an extended Kalman filter in kinematic mode to estimate the baseline position, velocity and single difference carrier-phase biases. The single difference carrier-phase biases are then combined to form the double differenced bias terms. RTKLIB uses the LAMBDA search strategy to fix the integer vector solution and update other states. The fixed integer vectors can be tightly coupled in the filter through the fix-and-hold setting. This adds a pseudo-measurement vector in the filter with the fixed integer ambiguity terms.

Figure 6.30 shows the tail baseline setup using two of the three GNSS receivers on the aircraft. Knowledge of the baseline length can improve the search strategy for the integer vector solution (Pinchin, 2011). However, in this research, the baseline length was used to validate the fixed solution without modifying the search strategy. This is because RTKLIB was able to return a fixed solution 95% of the time for all three GNSS receivers for the entire duration of the flight. However, the results may contain false positives, and the baseline constraint is used to reduce the rate of false positives in the data.

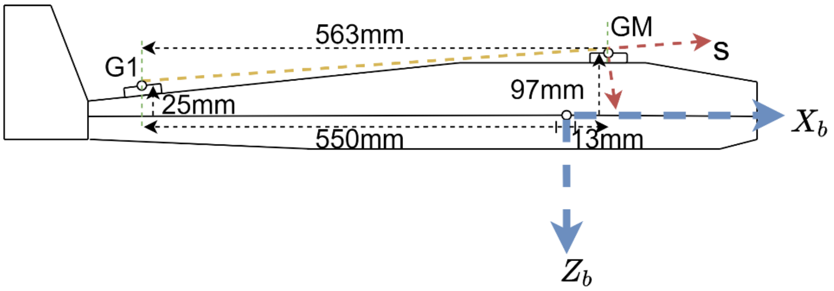


Figure 6.30. Tail baseline (b_1) using the master GNSS receiver (GM) and the receiver on the aircraft tail (GM).

Figure 6.31 shows the estimated tail baseline length plotted with the measured baseline length for the duration of the flight.

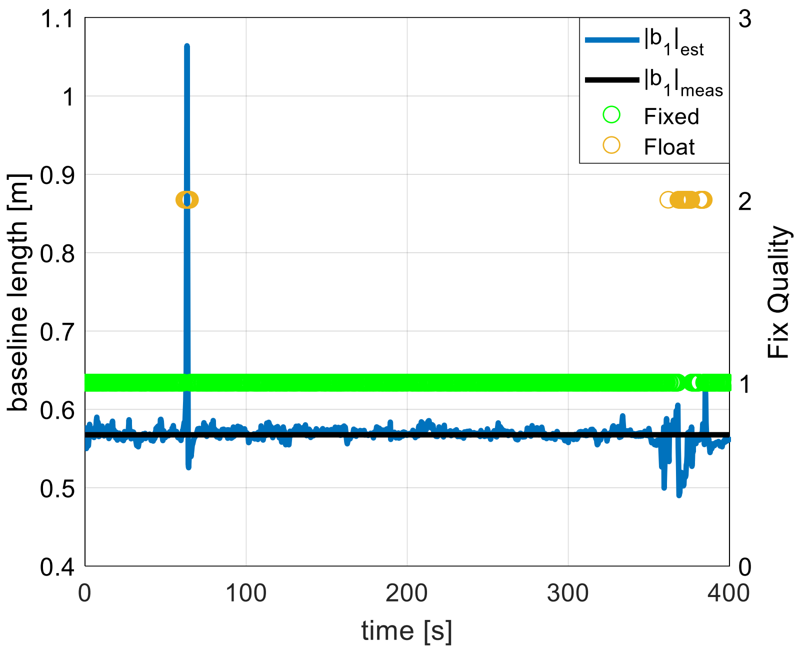


Figure 6.31. Aircraft tail baseline length estimation using RTKLIB. In the figure, the Fix Quality is 1 for a fixed solution and 2 for a float solution.

Evidently, the GNSS baseline length seems to be a good estimate of the measured baseline length for the entire duration of the flight except for the steep descent segment around 50 seconds into the flight and after landing, where the aircraft was at this point close to different objects on the ground. Figure 6.32 shows the corresponding baseline estimation performance for b_2 using the GNSS receiver at the wingtip (G2) and the master GNSS receiver (GM). This solution seems to have slightly more noise than the tail baseline length estimation. Assuming the *a priori* measurement of the baseline lengths is a perfect measurement, the RMS of the difference between the measured and estimated fixed solution for the tail baseline length solution is around 8.3 mm while it is around 9.4 mm for the wing tip baseline length.

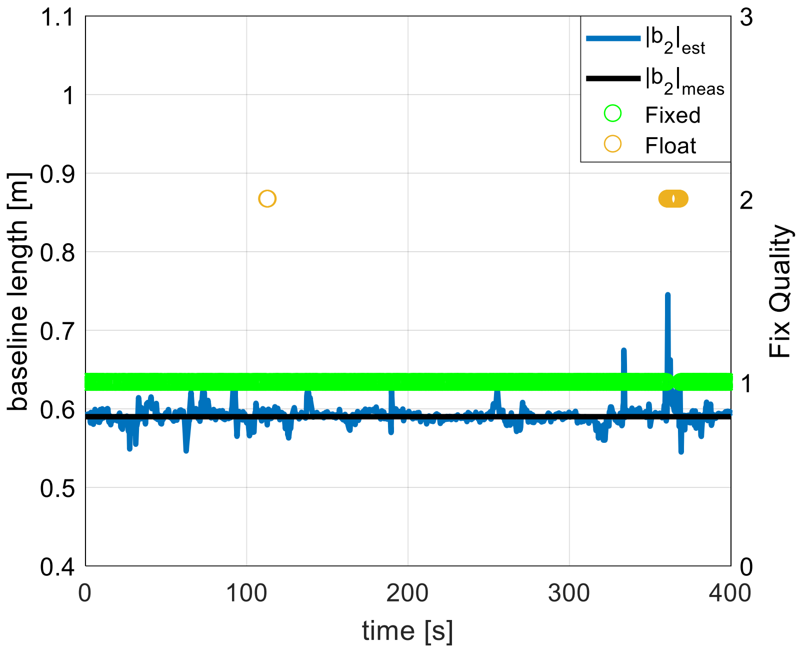


Figure 6.32. Aircraft wingtip baseline length estimation using RTKLIB. In the figure, the Fix Quality is 1 for a fixed solution and 2 for a float solution.

The increased dynamics around the roll axis might be the reason for the increased noise for the wing tip baseline. Pinchin (2011) showed that, for a static case with a relatively short baseline, around 1 m, the precision of the baseline components is around 5.4 mm for a fixed solution. This resulted in the precision for pitch and yaw angle to be around 0.23 degrees and 0.11 degrees, respectively. It was suggested that the difference between the fixed baseline length and the *a priori* measurement of the baseline length is normally distributed, provided that the two are also normally distributed. This reads:

$$\Delta|b| \sim N(\mu_{|\Delta b|}, \sigma^2_{|\Delta b|}) \tag{6.17}$$

It is possible to assign a threshold probability above which the fixed baseline length is taken to be the same as the *a priori* measurement of the baseline length and below which the fixed solution is rejected. It was shown that with a threshold of 98%, the false positive (candidate ambiguity vector passes the ratio test but is in fact false) rate could be reduced to 0.5% using the standard LAMBDA and the baseline constraint.

In this research, baseline components are used to estimate the attitude of the aircraft. Some discussion on the accuracy of the estimates alongside a comparison of different baseline solutions is given in Section C.5 of Appendix C of this thesis. The difference between the tail baseline component b_1 and the local component projected in the NED frame ($R_b^n F^b$ where F^b is the local component) is shown in Figure 6.33. The figure also shows the estimated yaw angle for the two baselines, and the standard deviation of their difference is around one degree ($1\sigma \approx 1°$).

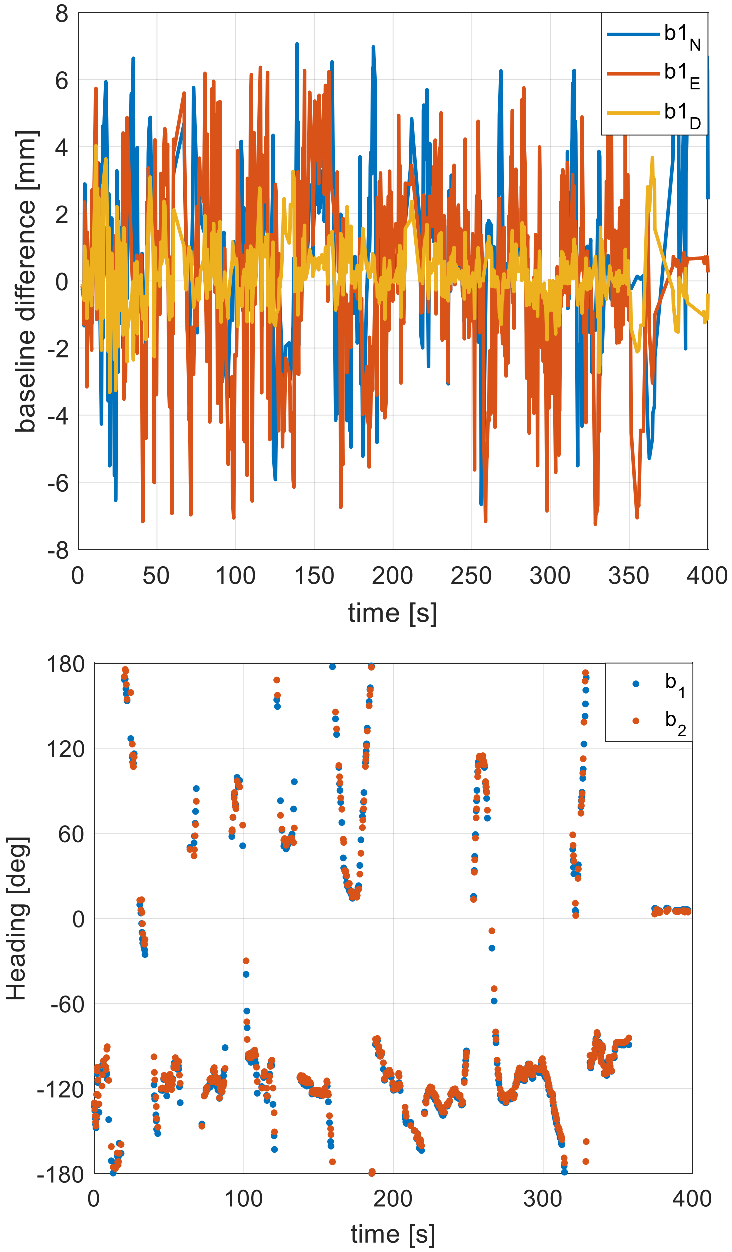
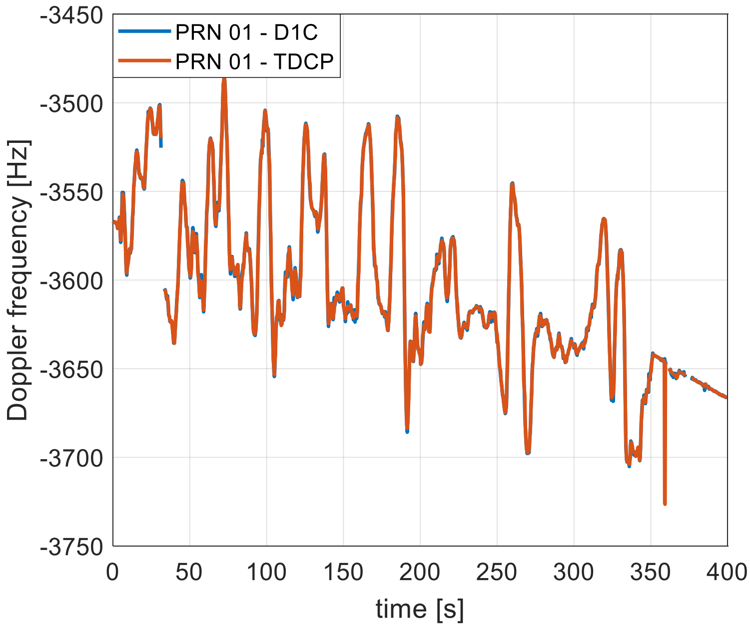


Figure 6.33. Baseline difference (top) and yaw angle estimation (bottom).

Time differenced carrier-phase observations (TDCP) are used to estimate the Doppler frequency shift between the aircraft and the satellites. This is, in turn, used to derive the aircraft's velocity components. Figure 6.34 shows the Doppler frequency derived from time differenced carrier-phase observations for PRN 01 alongside the doppler frequency output by the receiver for this satellite. The figure also shows the down component of the derived velocity estimates alongside the velocity output by the receiver. The TDCP velocity estimates contain less noise as opposed to the velocity estimates output by the receiver. The standard deviation of the difference between the two velocity estimates is around 0.25 m/s.

Following the characterisation of the IMU and the derivation of a post-processed kinematic position solution as well as a GNSS attitude solution, a reference navigation solution was then derived using a standard INS/GNSS integration architecture (see Section E.3 of Appendix E).



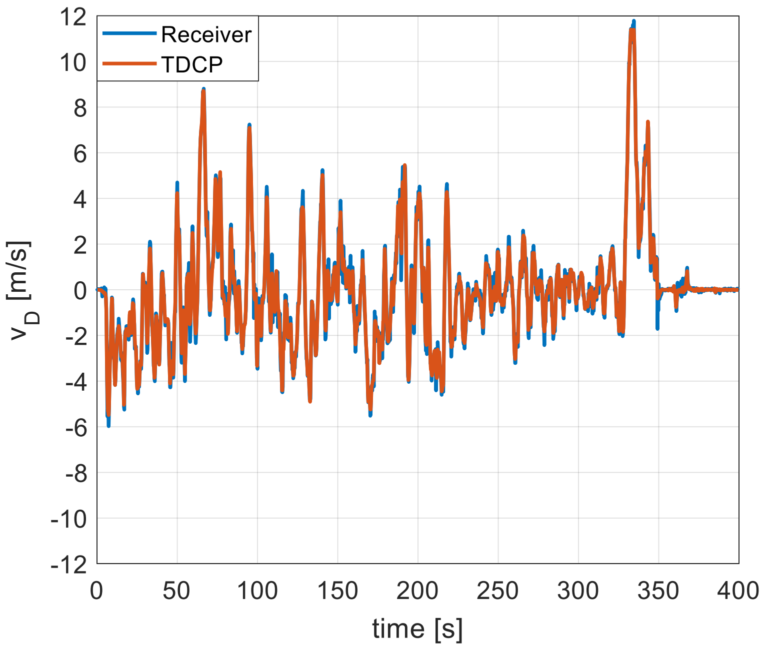


Figure 6.34. Doppler frequency (top) and down component of velocity (bottom).

6.8. Implementation

Following the characterisation of the aerodynamic model and propulsion model and the availability of a reference navigation solution, the TCVDM scheme presented in Chapter 5 can finally be tested using the recorded flight data. The scheme is presented again in Figure 6.35 to aid the reader in understanding how the different parts are used in the implementation.

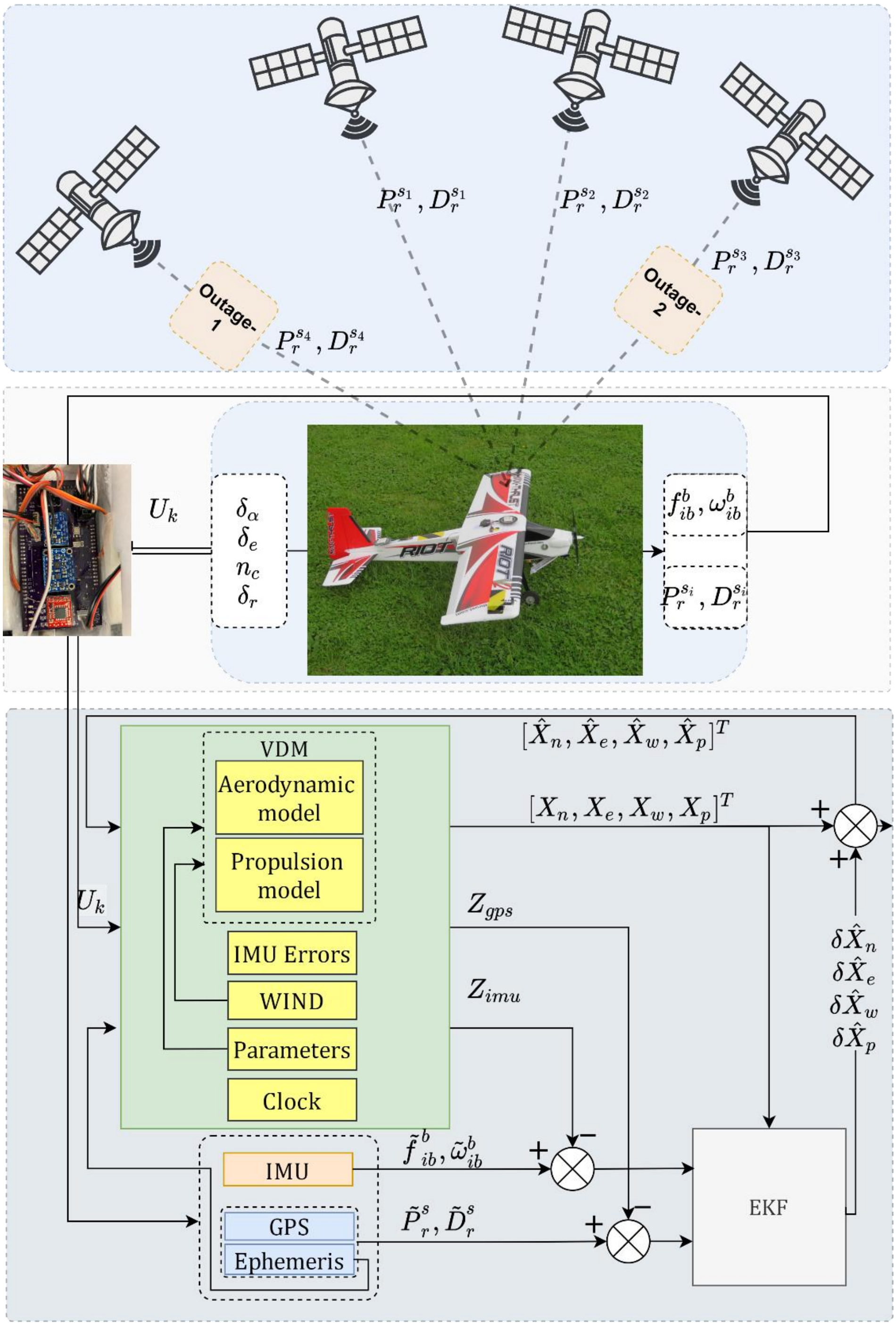
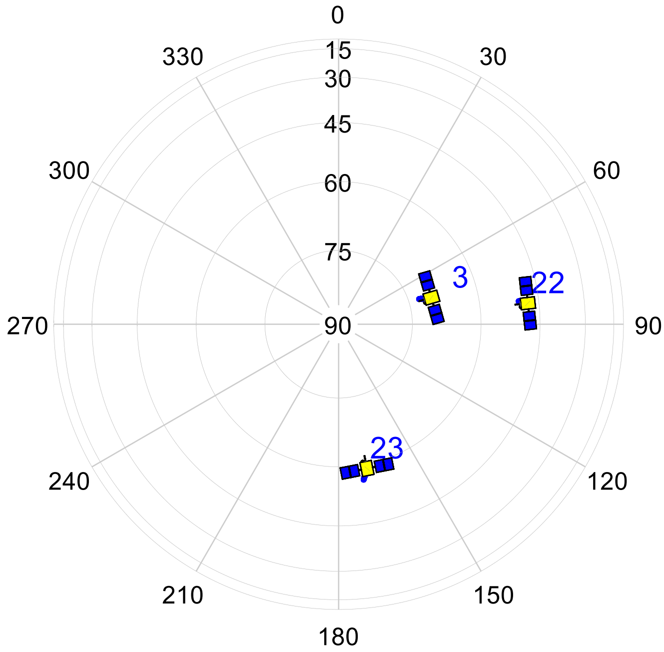
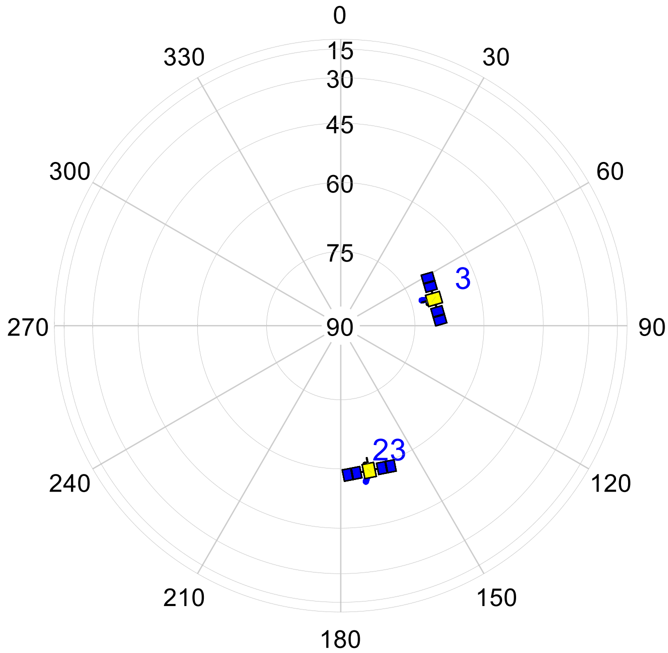


Figure 6.35. The implementation of the TCVDM scheme using real flight data.

The top section of the figure shows the two GNSS outage scenarios investigated. The middle section shows the UAV and the FCS used during the test flight, and the bottom section of the figure shows the TCVDM scheme using the derived aerodynamic and propulsion models. Figure 6.36 shows the number of satellites visible during the two GNSS outage scenarios investigated. A GNSS outage was induced 246 seconds into the flight and lasted for 100 seconds. During this time, the number of satellites visible was reduced by masking low elevation satellites. Similar to Chapter 5, it is important to note that only the GPS constellation was used to test the VDM integration architecture.



(a) First scenario [Mask angle 47°] – Three satellites



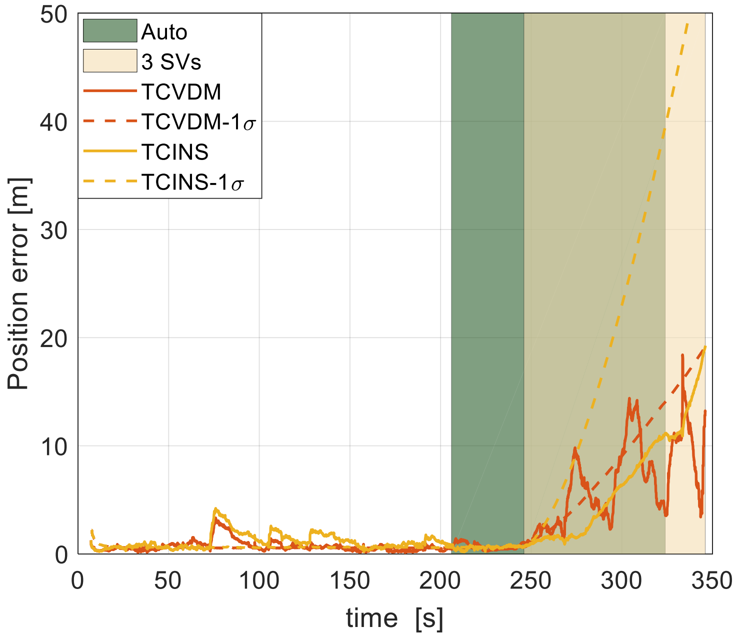
(b) Second scenario [Mask angle 53°] – Two satellites

Figure 6.36. Skyplots showing the remaining satellites after inducing a GNSS outage.

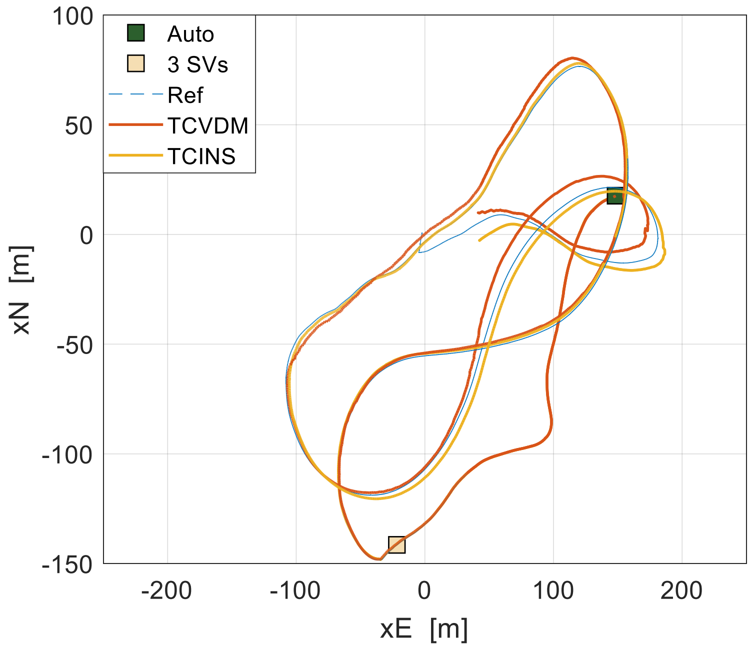
6.9. UAV Flight Results

6.9.1. Position

Figure 6.37 (a) shows the 3D position error and (b) the 2D position plots with three satellites visible during the GNSS outage. The 2D position plot only shows part of the trajectory starting from the point the autopilot was engaged (indicated by the green marker). The 2D position plot also shows the point the GNSS outage was induced and the number of satellites visible for the remainder of the flight.



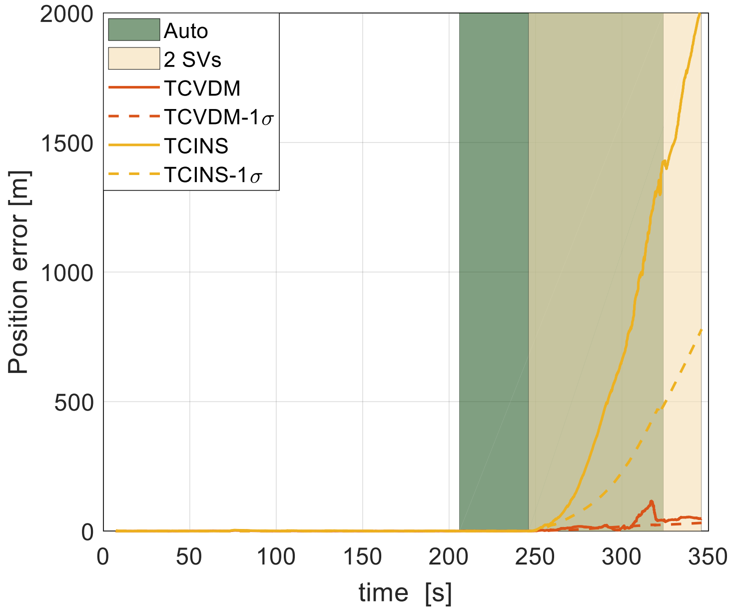
(a) 3D Position error



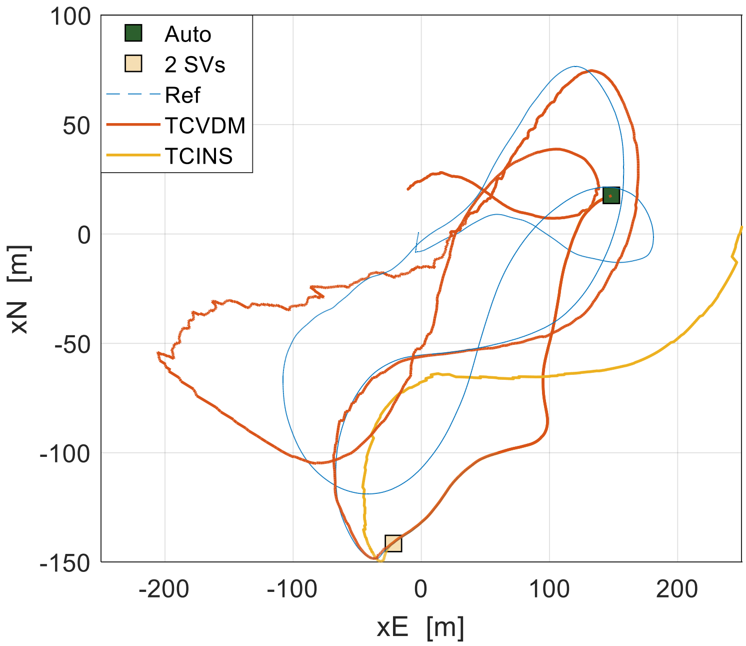
(b) 2D Position plot

Figure 6.37. Position estimation results with three satellites in view during the GNSS outage.

The final position error for the TCVDM scheme was very close to the TCINS scheme reaching only 13 m for the TCVDM and 19 m for the TCINS. However, it is important to point out that the developed TCVDM scheme showed improved estimation during turns depicted by the sharp decrease in overall position error around 280 seconds, 310 seconds and 330 seconds. In contrast, the TCINS scheme experienced gradual growth of position error during the outage. The 2D position plot shows that both the TCVDM and TCINS schemes followed the reference trajectory well. Figure 6.38 shows the 3D position error with two satellites visible during the GNSS outage.



(a) 3D Position error



(b) 2D Position plot

Figure 6.38. Position estimation results with two satellites in view during the GNSS outage.

Position error estimation results for the first 200 seconds of the flight are similar to the results shown in Figure 6.37. The final position error for the developed TCVDM scheme reached 47 metres, an improvement by a factor of 43 as opposed to the TCINS scheme. The 2D position plot also shows how well the VDM-based scheme was able to track the reference position solution as opposed to the INS-based scheme even with just two satellites visible, owing to the mitigation provided by the dynamic model.

6.9.2. Velocity

Figure 6.39 shows the velocity estimation performance for the TCVDM scheme alongside the TCINS architecture with three satellites visible during the GNSS outage. Generally, the performance of the TCINS scheme was better than the TCVDM scheme by a factor of 2 in the north and east channel based on the RMS of the velocity errors. However, with two satellites visible during the GNSS outage, the TCVDM scheme showed an improvement in velocity estimation by a factor of 7 as opposed to the TCINS scheme across all channels, as can be seen in Figure 6.40. Between 275 and 325 seconds, the aircraft flew mostly straight and level and in a south-westerly direction. During this time, the TCINS scheme's estimated velocity degraded rapidly, while the TCVDM scheme's estimated velocity degraded gradually, as can be seen in Figure 6.40. The RMS of the east velocity error was 8.26 m/s for the TCINS and only 1.11 m/s for the TCVDM.

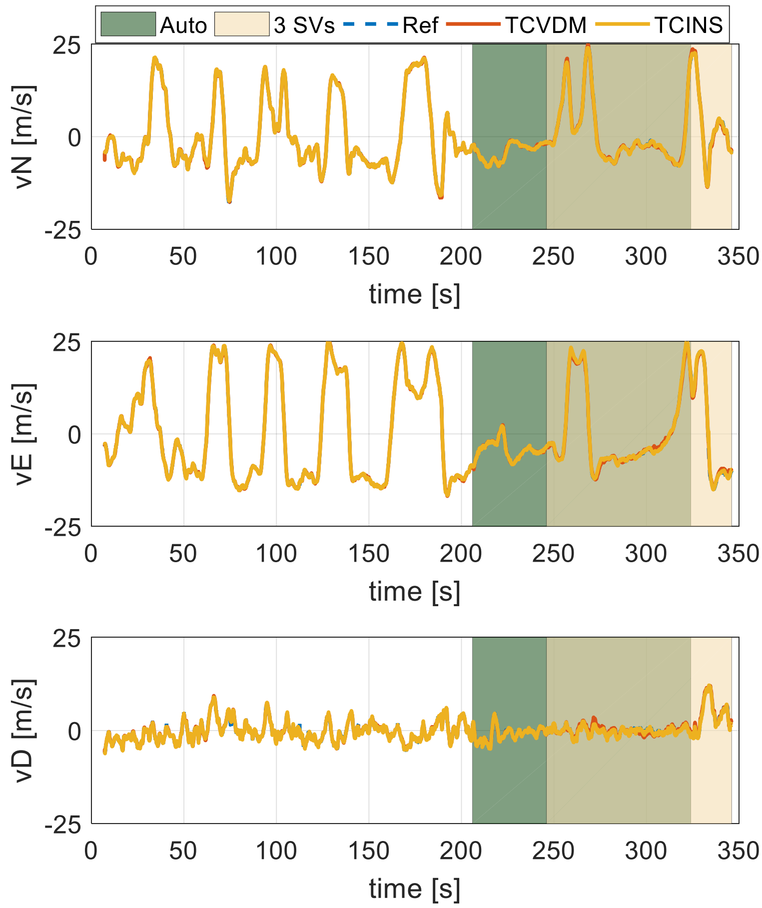


Figure 6.39. Velocity estimation performance with three satellites visible during the GNSS outage.

In Chapter 5, simulation results revealed that the degradation in velocity estimation is largely correlated with the cross-track wind velocity component. The RMS of the north component of the velocity error for the TCINS scheme was 2.80 m/s and only 0.41 m/s for the TCVDM scheme. It is important to point out that there was no direct measurement of wind speed and wind direction on the day of the flight, and therefore the cross-track and along-track wind velocity components are unknown. Since the TCVDM includes the mechanism to directly estimate wind velocity even without an air data system, it inherently has improved aircraft velocity estimation performance as opposed to the TCINS scheme.

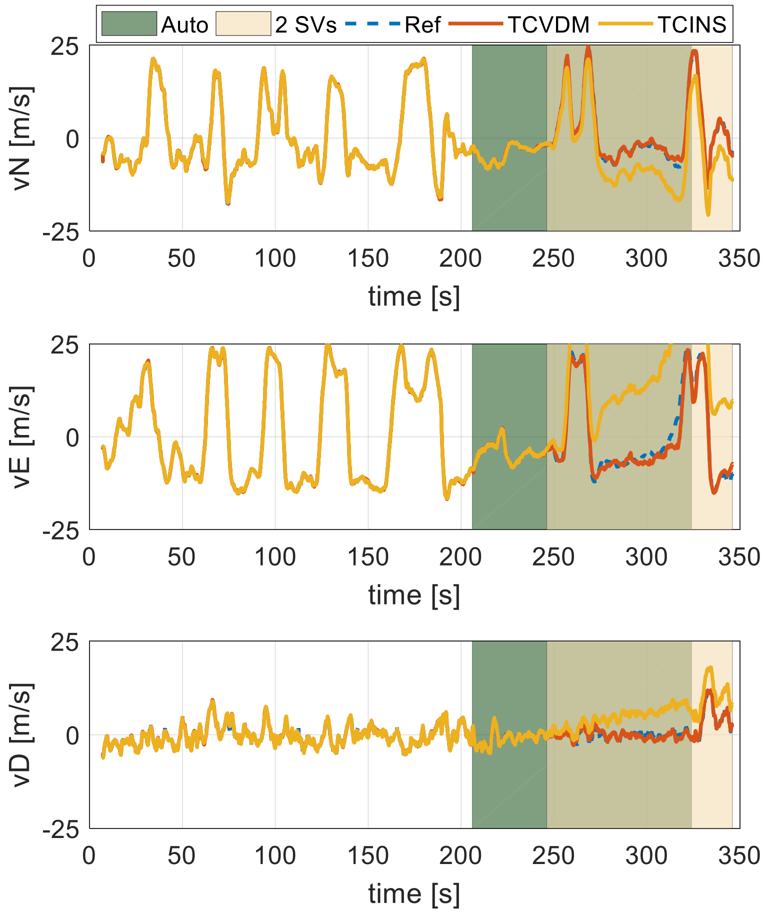


Figure 6.40. Velocity estimation performance with two satellites visible during the GNSS outage.

6.9.3. Attitude

Figure 6.41 and Figure 6.42 show the attitude estimation performance for the TCVDM scheme alongside the TCINS architecture with three and two satellites visible during the GNSS outage. With three satellites visible during the outage, the roll and pitch angle estimation performance (in terms of the RMS of the errors) of the TCINS scheme was better than the TCVDM scheme by at least a factor of 2. The TCINS scheme also showed improved yaw estimation by a factor of 4 as opposed to the TCVDM scheme.

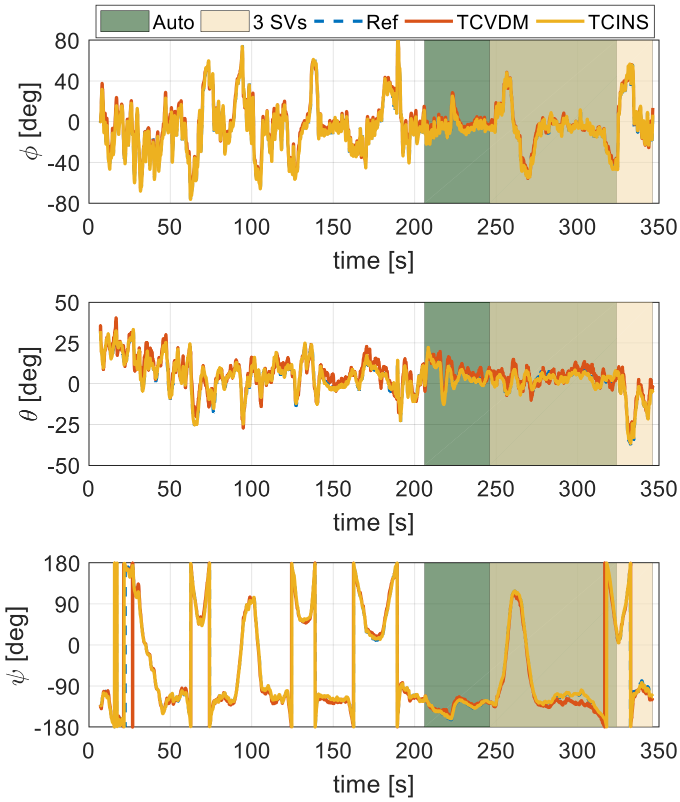


Figure 6.41. Attitude estimation performance with three satellites visible during the GNSS outage.

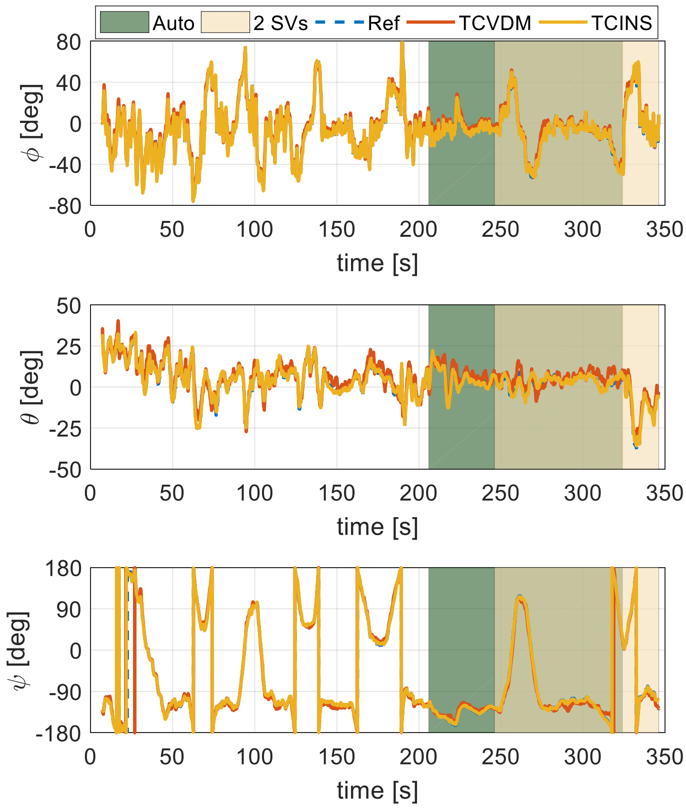


Figure 6.42. Attitude estimation performance with two satellites visible during the GNSS outage.

With two satellites visible during the GNSS outage, the RMS of roll angle errors for the TCVDM scheme increased by 8% from the case with three satellites visible during the outage. The RMS of pitch angle errors for the TCVDM scheme was similar to the case with three satellites visible during the outage but increased by 6% for the yaw angle. Further, the estimated yaw angle in the TCVDM scheme showed increased drift between 275 seconds and 325 seconds when the aircraft was flying mostly straight and level in a south-westerly direction. The significantly poor performance in attitude estimation by the TCVDM scheme compared with the TCINS scheme is attributed to the large uncertainties in the torque coefficients due to the limitations of the simple estimation routine used.

6.10. Summary

In this chapter, the performance of the developed tightly coupled vehicle dynamic model-based integration architecture (TCVDM) was evaluated using flight data gathered from a small, commercial-off-the-shelf UAV. A sensor shield was designed for the ATmega2560 board and loaded with custom firmware used for guidance and control of the aircraft. The board contained a low-cost IMU (NXP 9DOF), a barometer (BMP388) and a data logging module. The board was then mounted onto the aircraft alongside three u-blox NEO-M8T GNSS receivers. The three receivers were used to derive a reference position and attitude solution used to assess the performance of the TCVDM scheme. The small UAV's aerodynamic and propulsion model had to be characterised to test the TCVDM architecture. A geometry-based technique, using freely available packages including Athena Vortex Lattice (AVL) and XFLR5, was used to obtain the aerodynamic coefficients of the UAV and its mass moments of inertia. The technique was supplemented by wind tunnel testing and full-scale oscillation tests. The propulsion model was derived using ground measurements as well as data available from the propeller manufacturer. The reference solution was derived from an INS/GNSS integration architecture using an EKF. Post-processed kinematic position, time differenced carrier-phase derived velocity and GNSS attitude solution were used as measurements in the architecture.

A GNSS outage was induced by artificially removing satellites below a specific elevation angle to investigate the navigation performance of the TCVDM scheme. The entire flight lasted 400 seconds, and the GNSS outage lasted 100 seconds. Two scenarios were investigated, with three and two satellites visible during the GNSS outage. The performance of the TCVDM scheme was compared to the performance of a TCINS scheme. With two satellites visible during the outage, the TCVDM scheme showed a significant improvement in position estimation. The final position error for the TCVDM scheme reached 47 metres, an improvement by more than one order of magnitude as opposed to a TCINS scheme. Similarly, velocity estimation performance improved by a factor of 7 for the TCVDM scheme across all channels compared to the TCINS scheme. The attitude estimation performance for the TCVDM scheme was worse as opposed to the TCINS scheme, especially in the estimation of yaw angle attributed to the large uncertainty in the torque coefficients due to the limitations of the estimation routine.

7 Conclusions & Future Work

7.1. Conclusion

The main research question addressed in this thesis reads:

“To what extent can knowledge of the vehicle dynamic model and associated control inputs be used with low-cost MEMS-grade inertial sensors and mass-market GNSS receivers to reduce drift in the navigation solution during a GNSS outage?”.

Most UAVs use an inertial navigation system integrated with a global navigation satellite system (INS/GNSS). During a GNSS outage (or when tracking less than four satellites), the errors in the inertial navigation solution will grow unboundedly. In low-cost applications where the quality of the inertial sensors is relatively low, the navigation solution errors grow rapidly, rendering the navigation solution useless in a few seconds. The position error can even reach two kilometres in one minute. A GNSS outage can be caused by operating in an urban canyon setting where the reception of satellite signals is difficult, rapid dynamics and intentional or unintentional signal interference. In some applications, additional aiding sensors such as cameras and range finders have been used to mitigate rapid drift in the navigation solution. However, this approach adds weight and extra cost to the overall system. In other cases, advanced error models have been used for the inertial sensors found in an inertial measurement unit (IMU); however, this requires expensive equipment to characterise the errors and introduces additional software complexities.

In recent years, the use of a vehicle dynamic model (VDM) has emerged as a possible alternative to inertial coasting. The approach preserves the autonomy of the navigation system and avoids adding extra weight and cost, ideal for low-cost applications.

The aim of this research was to investigate and test a VDM navigation architecture suitable for a fixed-wing UAV fitted with low-cost MEMS-grade inertial sensors (< £5,000) and a GNSS receiver during periods of extended GNSS outage. In fulfilling this aim, a series of objectives were formulated. The main constraint applied to this research was in the use of a battery-powered fixed-wing UAV. This platform has more range and endurance than the rotary-wing vertical take-off and landing (VTOL) type, such as quadcopters and helicopters.

To meet the research objectives, different VDM integration schemes were reviewed, and approaches that give the most robust navigation solution during GNSS outages were identified. A six-degree-of-freedom (6DOF) fixed-wing aircraft model was developed in Matlab/Simulink. The model was used to generate the GNSS and IMU data used to investigate the performance of different VDM integration schemes. Limitations of the current state-of-the-art VDM navigation techniques were identified alongside approaches to mitigate rapid error growth during GNSS outages without adding extra weight and cost to a fixed-wing UAV.

Following this, a novel integration architecture that implements the approaches and overcomes these limitations was developed. The architecture utilised the VDM as the main process model, and raw GNSS observables and IMU measurements were used to aid the solution. The performance of the developed architecture was investigated using Monte Carlo simulation using the available dataset. A small, commercial-off-the-shelf, fixed-wing UAV was modified and characterised and used for practical testing of the developed architecture. Real flight data gathered from a test flight using the small UAV fitted with a MEMS-grade IMU and GNSS receivers was used to validate the results obtained in the simulation study.

Simulation results showed that the position error estimated by the developed architecture improved by one to two orders of magnitude compared to the error estimated using an INS/GNSS integration scheme during GNSS outages lasting over two minutes. Further, the use of the VDM allowed the estimation of wind velocity components, which contributed to improved velocity error estimation and, in turn, position error estimation. This was achieved without an air data system. Further, errors in the VDM parameters were estimated in the filter, which improved its robustness against variations in these parameters.

Practical testing of the developed architecture using data gathered from the test flight validated the conclusions drawn from the simulation studies. Position error estimation performance for the developed architecture improved by more than one order of magnitude compared to an INS/GNSS scheme during extended GNSS outages. Despite the significant improvement in position accuracy, poor attitude estimation results revealed the importance of a good VDM parameter estimation routine that provides accurate initial VDM parameters.

Unlike the current state-of-the-art VDM navigation schemes, the novel integration architecture operates in a tightly coupled configuration, using raw GNSS observables (pseudoranges and Doppler frequencies) and IMU measurements to estimate corrections to the navigation solution. The use of raw GNSS observables in the architecture has allowed the navigation solution to degrade gracefully during GNSS outages. The developed architecture extends the current state-of-the-art by implementing the recent model-based approach, which uses the VDM as the main process model. Even though the approach was developed and tested on a fixed-wing UAV, it is generally applicable to other platforms such as quadrotors, helicopters, hybrid VTOL aircraft, ground vehicles and underwater vehicles following appropriate modelling of their dynamics.

The developed architecture can work alongside the conventional INS/GNSS integration architecture in applications/areas prone to GNSS outages such as in urban canyon settings, search and rescue operations in mountainous areas, inspection of tunnels, and even in areas with GNSS signal interference.

Further details related to the specific conclusions drawn from the objectives of this research are provided as follows:

❖ *Objective 1: To investigate the navigation performance and quantify limitations of the current state-of-the-art VDM navigation schemes during GNSS outages.*

In Chapter 4, an improved VDM-based integration architecture operating in a loosely coupled configuration (using position and velocity from a GNSS receiver) was presented and analysed. The approach used the VDM as the main process model to propagate the navigation solution whilst IMU and GNSS measurements were fused in a navigation filter to estimate corrections for the states. The architecture used the unscented Kalman filter (UKF) and included the mechanism to estimate the mass moment of inertia terms. The approach is referred to as the UKF/VDM architecture in this thesis. The performance of the proposed concept was compared to the state-of-the-art VDM-based architecture that uses an extended Kalman filter (EKF) and does not include the mechanism to estimate the moments of inertia. The state-of-the-art VDM-based scheme is referred to as the EKF/VDM integration architecture in this thesis. A Monte Carlo simulation study was used to investigate the performance of the UKF/VDM scheme compared to the EKF/VDM scheme.

Results indicated that the choice of the navigation filter, either the UKF or EKF, did not significantly influence the estimation performance of the navigation states. Errors in the mass moment of inertia terms only caused marginal errors in the navigation states. However, they were found to influence the estimation performance of the torque coefficients. The difference between the final root-mean-square (RMS) mean estimation error for all VDM parameters and the predicted confidence value (1σ) was only 19% for the UKF/VDM scheme and 48.5% for the EKF/VDM scheme after a GNSS outage that lasted over two minutes. Further, it was found that sharp turns with rapid speed changes at constant altitude improved the estimation performance of the lateral force coefficient as well as the rolling and yawing moment coefficients. Generally, compound manoeuvres, even during a GNSS outage, were found to improve the estimation performance for most VDM parameters. However, the inclusion of the mass moment of inertia terms or the choice of the filter did not reduce the accumulation of navigation errors during a GNSS outage.

Further, the navigation solution errors for a VDM-based integration architecture were investigated for different GNSS outage intervals with varying roll rates during turns. It was found that the position error increased proportionally with the roll rate for an extended GNSS outage lasting over a minute (60 seconds). However, for a GNSS outage lasting less than one minute, the roll rate during a turn did not influence the position error estimation performance. Similarly, attitude errors were not significantly influenced by GNSS outages lasting up to 60 seconds, with more extended outages (90 seconds) mainly influencing the yaw angle error. The impact was more pronounced when the aircraft turned during the outage.

- *Objective 2: To propose a novel integration algorithm that reduces drift in the navigation solution during an extended GNSS outage lasting over one minute without adding extra weight and cost to small UAVs.*

In Chapter 5, a novel, tightly coupled VDM-based (TCVDM) integration architecture was presented and analysed.

Following the review of different VDM navigation schemes in Chapter 2, this thesis identified that most VDM navigation schemes use a loosely coupled configuration. This resulted in significant drift in the navigation solution drift during extended GNSS outages, confirmed by the investigation in Chapter 4. Therefore, unlike other model-based schemes, the proposed concept used raw GNSS observables (pseudoranges and Doppler frequencies) and IMU measurements fused using an EKF to estimate corrections to the navigation solution even when tracking less than four satellites. The use of raw GNSS observables was identified as a possible alternative to VDM coasting during a partial GNSS outage and could reduce error growth during this period.

The thesis identified further limitations in most model-based schemes, such as the need for an accurate structure and set of model parameters for the host platform and the need to account for external disturbances, such as wind. Therefore, other than the navigation states, the proposed approach estimated wind velocity components, IMU errors, VDM parameters and the receiver clock errors. The review in Chapter 2 revealed that the inclusion of VDM parameters and estimation of wind velocity components improves the performance of the filter.

- *Objective 3: To undertake simulated data testing and practical testing of the proposed integration algorithm.*

A software-based GNSS measurement simulator was developed to generate raw GNSS observables to test the proposed integration architecture. A Monte Carlo simulation study was used to evaluate the performance of the proposed scheme. The error in observations, initialisation, and VDM parameters changed randomly in each realisation while the trajectory and the wind profile were the same. The navigation performance of the architecture was compared to a tightly coupled inertial navigation system (TCINS) and a loosely coupled model-based architecture (LCVDM).

Results from the Monte Carlo simulation study revealed that the developed architecture improves position accuracy by one order of magnitude with two satellites visible during an extended GNSS outage whilst offering similar roll and pitch angle accuracy compared to the TCINS scheme. It was also found that:

- Yaw angle estimation performance for the TCVDM scheme was significantly worse than the TCINS scheme, with a difference of about 60% at the end of the outage.
- For a modest trajectory, the proposed architecture only captured about 40% of the initial uncertainty in the VDM parameters due to significant correlation within groups of the parameters.

- Other auxiliary states such as wind velocity errors, IMU errors and clock errors were well estimated even with only two satellites in view during the outage.
- The error in the estimation of wind speed increased with decreasing number of satellites visible during the outage. However, the difference between the case with three and two satellites is less than 10%.
- Turning during a GNSS outage improved wind speed estimation for the TCVDM scheme while the wind speed error estimated by the LCVDM scheme increased gradually, even during turns.
- With two satellites in view during the outage, the clock bias error estimated by the TCVDM architecture increased gradually to 17 metres. This was only 5% higher than the error estimated with three satellites in view and an improvement by a factor of 5 compared to the error estimated by the TCINS scheme.

The simulation results were validated with real flight data gathered using a small fixed-wing UAV fitted with low-cost inertial sensors and a GNSS receiver. The small UAV was characterised using a geometry-based technique with Athena Vortex Lattice (AVL), supplemented by wind tunnel testing and full-scale oscillation tests. A custom flight control system (FCS) was used on the UAV for guidance, navigation and control. The FCS was also used for logging IMU measurements and control inputs. A custom ground control software (GCS) was used to pre-program the mission profile and change the settings of the FCS. The GCS was also used for logging incoming telemetry (for redundancy) via a radio link. A GNSS outage was induced by precluding low elevation satellite observations.

Flight results showed significant performance enhancement in position and velocity error estimation. It was found that:

- With two satellites visible during the GNSS outage, the RMS estimation errors for the velocity components in the TCVDM scheme improved by a factor of 7 across all channels compared to the TCINS scheme.
- The final position error estimated by the TCVDM architecture improved by a factor of 43 compared to the TCINS scheme.
- The attitude estimation performance for the TCVDM scheme was significantly worse than the TCINS scheme due to large uncertainties in some of the model parameters.

7.2. Limitations

The position and velocity error estimation performance for the proposed architecture in an experimental setting is auspicious and has shown that the scheme can be used during extended GNSS outages to provide an improved navigation solution. However, it is important to highlight some challenges and potential issues that, if addressed, can improve attitude performance altogether.

- The initial parameters used were determined from a Monte Carlo simulation study in AVL using the geometry of the aircraft. The resolution of some of the parameters, especially the moment derivatives, was poor, and this might have significantly contributed to the poor attitude estimation performance during the outage. Because the architecture only resolves a small amount of the initial VDM parameter uncertainty, it is important to have a reasonably good estimate of these parameters.

- Secondary effects such as actuator dynamics and delays in the actuator signals were not considered in this investigation. Actuator dynamics would have improved the model's fidelity with an additional penalty of extra states for each control surface. A tightly coupled architecture is usually sensitive to synchronisation errors, and therefore delays in the actuator signal might have contributed to the degraded performance.
- Such a scheme can only be used after the aircraft has taken off and before it lands; otherwise, some states could be biased.
- The quality of the IMU plays an important role in attitude estimation, especially when the uncertainty in the model parameters is large. Therefore, effects such as large vibrations and thermal loading could indirectly influence the performance of the architecture during an outage.

7.3. Future Work

The use of VDM navigation schemes is still an active area of research that still needs practical testing. To continue this research and make further advancements, recommendations for future work are made in this section.

The architecture developed in this thesis needs further practical testing on small UAVs fitted with low-cost sensors. Some areas that further practical testing could be useful include:

- To investigate the impact of actuator dynamics on navigation performance, especially during a GNSS outage.
- To investigate wind estimation performance without an air data system, especially during a GNSS outage.
- To investigate the impact on the navigation solution of using different higher grade MEMS inertial sensors (temperature compensated, improved bias stability).
- To investigate the impact of additional measurements typically available from sensors in a UAV such as Lidar, magnetometers, barometers, and airspeed sensors.

Experimental results indicate the importance of an accurate set of model parameters. Wind tunnel testing and complex computational routines such as computational fluid dynamics (CFD) can be time-consuming and not ideal for small teams and low-cost applications. Therefore, it is important to investigate suitable VDM parameter estimation routines that are generally low-cost, less time-consuming, and, ideally, take advantage of typical sensors found in most UAVs. For instance, recent advancements in machine learning techniques to characterise the aerodynamic model of small UAVs seems to offer an attractive solution.

Even though the architecture is developed for a fixed-wing UAV, it is generally understood that the principles could be extended to other platforms such as rotary-wing VTOL type following appropriate modelling of the dynamics. Therefore, the architecture's performance could be investigated in a different platform, such as a quadrotor and even hybrid VTOL aircraft, with different dynamics to a fixed-wing UAV.

Even though the proposed architecture has only been tested in a single frequency setting, the algorithm could be tested in a multifrequency setting. Multifrequency testing would allow eliminating the ionospheric error by combining measurements from different frequencies leading to an improved navigation solution. Further, the algorithm could also be used in a multi-constellation setting, taking advantage of improved signals with lower noise and improved multipath performance.

Lastly, the reception of non-line-of-sight (NLOS) signals is a big problem when operating in dense urban environments. This is especially useful for VTOL aircraft such as quadrotors operated in different environments, including dense urban environments. NLOS reception is not easily mitigated by improved signals or receiver design. It is usually dealt with by carrier power to noise density ratio thresholding, outlier detection and 3D mapping, none of which are completely reliable. The proposed TCVDM scheme could potentially be used to aid the detection of outliers due to NLOS reception.

Appendices

A. Autopilot

A.1. L1 Guidance Logic

This section gives a brief overview of the guidance logic used in the simulations. The guidance logic combines a nonlinear lateral guidance control law with a simple adaptive path planning algorithm (Park, Deyst and How, 2004). Figure A.1 shows the nonlinear guidance logic. In the figure, η is the track capture angle, R is the radius of curvature, L_1 is the reference point distance, O is the current aircraft location, P is the reference point and V_n is the ground speed.

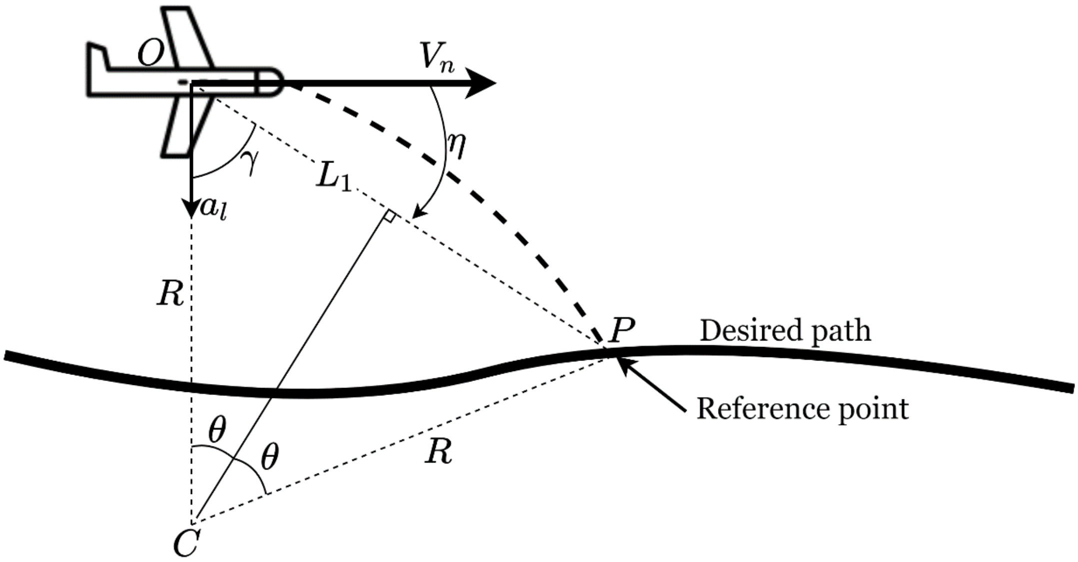


Figure A.1. A nonlinear guidance logic.

The nonlinear guidance logic generates a lateral acceleration command based on the cross-track error, as shown in Figure A.1. This commanded lateral acceleration is given by:-

$$a_l = \frac{V_n^2}{R} \tag{A.1}$$

$$R = \frac{L_1}{2\sin\theta} \equiv \frac{L_1}{2\cos\gamma}$$

since

$$\gamma = \frac{\pi}{2} - n \tag{A.2}$$

therefore

$$R = \frac{L_1}{2\sin\eta} \tag{A.3}$$

$$a_l = \frac{2\,V_n^2}{L_1}\sin\eta$$

The bank angle of an aircraft during a turn is given by:

$$\tan\phi = \frac{V_n^2}{R\,g}\ ;\ \phi_c \approx \frac{a_l}{g} \tag{A.4}$$

The aircraft tends to align its velocity vector with the direction of L_1. The track capture angle determines the direction of the commanded acceleration, as shown in Figure A.2.

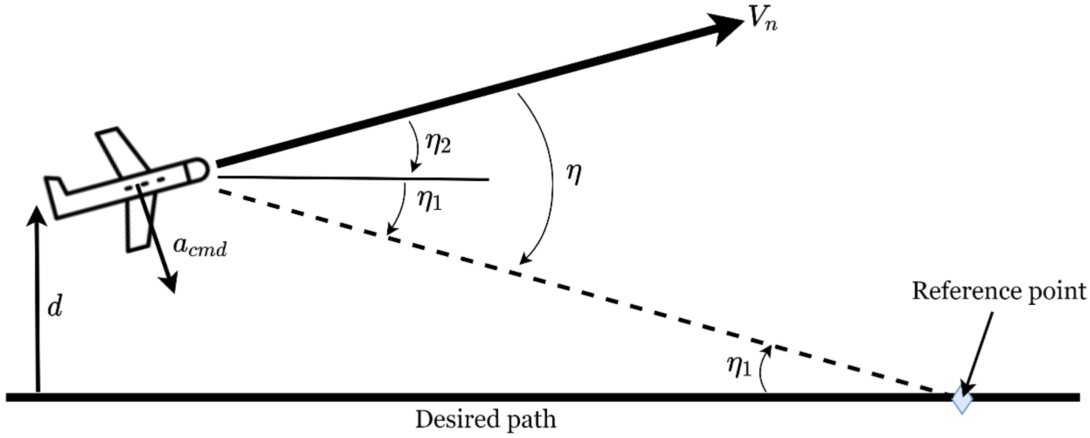


Figure A.2. Linear approximation of the guidance logic; d is the cross-track error for a straight path.

For a small track capture angle some approximations can be made. These are given by:

$$\sin\eta \approx \eta = \eta_1 + \eta_2 \tag{A.5}$$

$$\eta_1 \approx \frac{d}{L_1}; \eta_2 = \frac{\dot{d}}{V_n}$$

Therefore the commanded lateral acceleration is approximated as:

$$a_l \approx \frac{2V_n^2}{L_1}\left(\frac{d}{L_1} + \frac{\dot{d}}{V_n}\right) \equiv 2\frac{V_n}{L_1}\left(\dot{d} + \frac{V_n}{L_1}d\right) \tag{A.6}$$

A small angle approximation on η_2 and further Linearisation (for the case of following a straight line) results to:-

$$a_l = -\ddot{d} = 2\frac{V_n}{L_1}\left(\dot{d} + \frac{V_n}{L_1}d\right) \tag{A.7}$$

$$\ddot{d} + 2\frac{V_n}{L_1}\left(\dot{d} + \frac{V_n}{L_1}d\right) = 0$$

$$\ddot{d} + 2\zeta\omega_n\dot{d} + \omega_n^2 d = 0$$

$$\zeta = \frac{1}{\sqrt{2}}\ ; \omega_n = \sqrt{2}V/L_1$$

With a predefinition of the damping ratio and the time constant, the guidance logic is effectively used for lateral control to guide the aircraft from one waypoint to the other.

A.2. TECS Controller

This section gives a brief overview of the altitude and airspeed controller used in the simulations. The total energy control system (TECS) controller controls altitude and airspeed through the total energy rate and energy distribution rate in a manner that decouples their dynamic responses (Balmer, 2015; Argyle and Beard, 2016). The total energy for an aircraft assumed as a point mass system is given as the sum of potential energy and kinetic energy.

$$E_T = mg(h - h_0) + \frac{1}{2}mV^2 \tag{A.8}$$

where: m is the aircraft mass (kg),
h is the height of the aircraft (m),
V is the airspeed (m/s).

The total energy rate is given by:

$$\dot{E}_T = mg\dot{h} + mV\dot{V} \tag{A.9}$$

where: $\dot{h}$ is the rate of change of height (m/s),
$\dot{V}$ is the rate of change of airspeed (m/s^2).

The energy distribution is given as the difference between the potential and kinetic energy.

$$E_D = mgh - \frac{1}{2}mV^2 \tag{A.10}$$

whereas the energy distribution rate is given by:

$$\dot{E}_D = mg\dot{h} - mV\dot{V} \tag{A.11}$$

The underlying assumptions are that the aircraft is treated as a point mass, and the only way to add energy is through thrust, whilst drag is the only way energy is removed. The elevator is assumed to control the distribution between kinetic and potential energy (speed and altitude). The angle of attack 'α' is assumed small and flight path angle 'γ' is assumed not to influence drag. The commanded thrust is given by:

$$T^c = T_D + \Delta T \tag{A.12}$$

The trim thrust T_D counteracts drag (which is assumed not to vary significantly from the value at trim), and the extra thrust ΔT is needed to meet both energy and energy rate demands. The energy rate is equivalent to the excess power (the difference between the power available and the power required). Dividing the energy rate equation with mgV leads to:

$$\frac{\dot{h}}{V} + \frac{\dot{V}}{g} = \frac{T - D}{mg} \tag{A.13}$$

With $\dot{h}/V = \sin\gamma$, where γ is the flight path angle which is assumed to be small for this case and leads to the following equation:

$$\frac{\dot{V}}{g} + \gamma = \frac{T - D}{mg} \tag{A.14}$$

As stated earlier, the drag value at trim is assumed not to vary significantly and therefore counteracted with the trim thrust setting. Therefore, short-term thrust requirements can be expressed as:

$$\frac{\delta T}{mg} = \frac{\dot{V}}{g} + \gamma \tag{A.15}$$

The commanded throttle and pitch can then be given in terms of the energy rate and energy distribution given by (Balmer, 2015):

$$\frac{T_{des}}{mg} = \frac{K_{TI}}{s}\left(\gamma_{des} - \gamma + \frac{\dot{V}_{des}}{g} - \frac{\dot{V}}{g}\right) - K_{TP}\left(\gamma + \frac{\dot{V}}{g}\right) \tag{A.16}$$

$$\theta_{des} = \frac{K_{EI}}{s}\left(\gamma_{des} - \gamma - \frac{\dot{V}_{des}}{g} + \frac{\dot{V}}{g}\right) - K_{EP}\left(\gamma - \frac{\dot{V}}{g}\right) \tag{A.17}$$

And the desired height rate and airspeed rate are computed as (Argyle and Beard, 2016):

$$\dot{h}_d = k_h(h_{des} - h) \tag{A.18}$$

$$\dot{V}_d = k_v(V_{des} - V) \tag{A.19}$$

And the entire TECS logic is shown in Figure A.3 and Figure A.4.

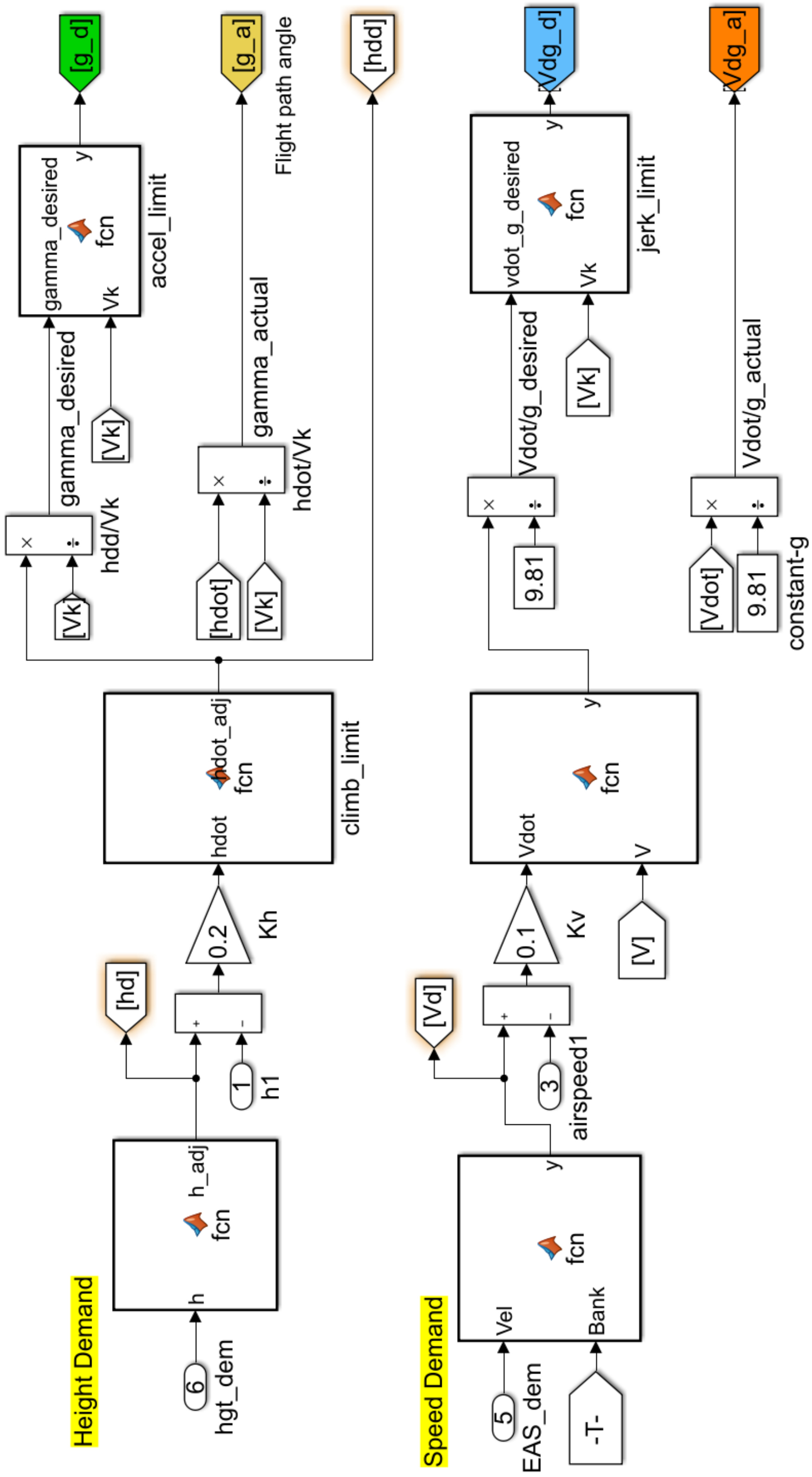


Figure A.3. TECS controller: height and speed demand.

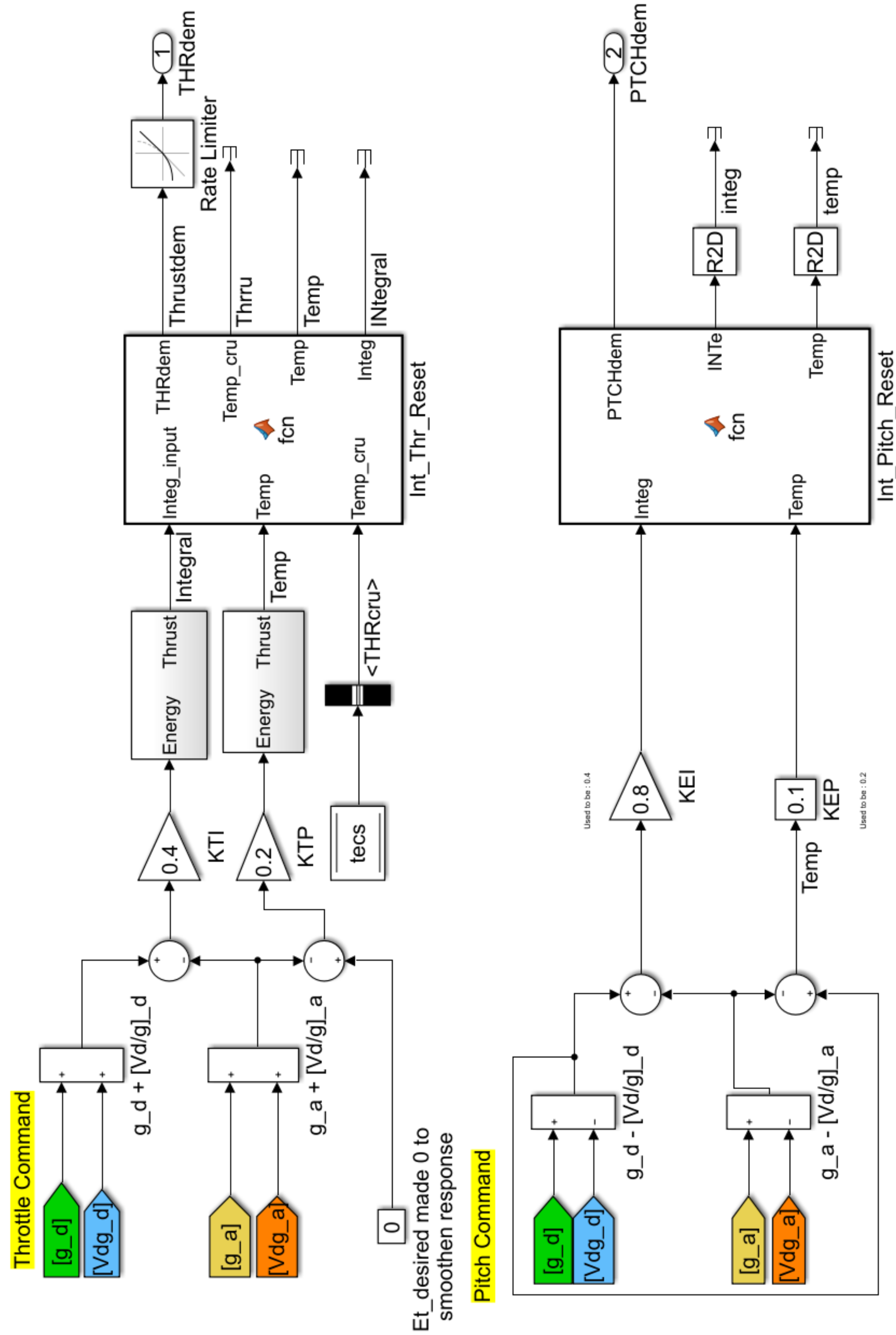


Figure A.4. TECS controller: throttle and pitch commands.

B. Simulations

B.1. Number of Simulations

The purpose of this section is to give a brief overview of the justification of the number of simulations used to evaluate the navigation performance of a VDM-based integration architecture. The minimum number of simulations was determined by evaluating the level of precision (the difference between the sample mean and population mean) with an associated confidence level. Here, a loosely coupled VDM-based integration architecture was used to evaluate the number of simulations required. The GNSS receiver and IMU error characteristics used are given in Table B.1.

Table B.1. IMU error characteristics.

Sensor	Type	Value
Accelerometer	Random bias (σ)	10 mg
	White noise (PSD)	100 μg/$\sqrt{\text{Hz}}$
	GM-Process	0.05 mg
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
Gyroscope	Random bias (σ)	1000 °/hr
	White noise (PSD)	21.6 °/hr/$\sqrt{\text{Hz}}$
	GM-Process	20 °/hr
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
GNSS receiver	White noise (σ)	1 m
	Sampling Frequency	1 Hz

The standard deviation of the initial uncertainty (P_0) considered for the different states is given in Table B.2.

Table B.2. Initial uncertainty [P_0].

State	Standard deviation (σ)
Position	1 m
Velocity	[1, 0.5, 0.5] m/s
Attitude	[3.5°, 3.5°, 5°]
Rotation rates	1.5 °/s
Propeller speed	15 rad/s
Model parameters	10%
Moment of Inertia	10%

The standard deviations used in the process noise covariance matrix (Q_k) for different states are given in Table B.3.

Table B.3. Process Noise.

State	Standard deviation (σ)
Position	10^{-6} m
Velocity	0.008 m/s
Attitude	10^{-6} rad
Rotation rates	10^{-4} rad/s
Propeller speed	10^{-4} rad/s
Accelerometer Bias	2×10^{-5} m/s^2
Gyroscope Bias	2×10^{-6} rad/s
Wind	10^{-3} m/s
Model parameters	0.015% of True Values
Moment of Inertia	0.015% of True Values

Using sampled-based statistics, the combined mean is given by (Ramprasadh and Arya, 2011):

$$\bar{X}_c = \frac{1}{\sum_m N_m}\left(\sum_m N_m \bar{X}_m\right) \tag{B.1}$$

And the combined standard deviation is given by:

$$\sigma_c = \frac{N_m - 1}{\sum_m N_m - 1}\left(\sum_m \sigma_m^2 + \frac{\sum_m N_m - N_m}{N_m - 1} \times var\left(\vec{\bar{X}}_m\right)\right) \tag{B.2}$$

The central limit theorem states that the distribution of the sample means approximates a normal distribution as the sample size gets larger, regardless of the type of distribution of the population data (Brown and Hwang, 2012). This exhibits a phenomenon where the average of the sample means and standard deviations equal the population mean and standard deviation. The statistic associated with a specific confidence interval can be written in the probabilistic form in terms of the level of precision (the difference between the sample mean and population mean).

$$P(-\frac{\phi}{s/\sqrt{n}} < \frac{\bar{x} - \mu}{s/\sqrt{n}} < \frac{\phi}{s/\sqrt{n}}) = confidence\ level \tag{B.3}$$

Given the associated confidence level, the minimum number of simulations required for achieving a certain level of precision can be determined. Evaluating the sample statistics for different runs revealed that 60 Monte Carlo runs resulted in a precision of less than 1 metre in all the estimated position states with a 95% confidence level, as can be seen in Figure B.1. With 60 runs, the precision for velocity states was less than 0.1 m/s, while for the attitude states, it was well below 0.25 degrees and below 0.1 deg/sec for the rotation rate states. The combined standard deviation from different sample runs for each state is used to evaluate the state's precision. Therefore, 100 Monte Carlo runs seemed reasonable in attaining a precision of less than 1 metre in the estimation of position states whilst guaranteeing similarly lower precision for all the other states.

Table B.4. The precision with 60 runs.

State	**Precision : $\bar{X} - \mu$**
Position: $[x_N, x_E, x_D]^T$	< 1 m
Velocity: $[u, v, w]^T$	$<$ 0.1 m/s
Attitude: $[\phi, \theta, \psi]^T$	$<$ 0.25°
Rates : $[\omega_x, \omega_y, \omega_z]$	$<$ 0.1 °$/s$

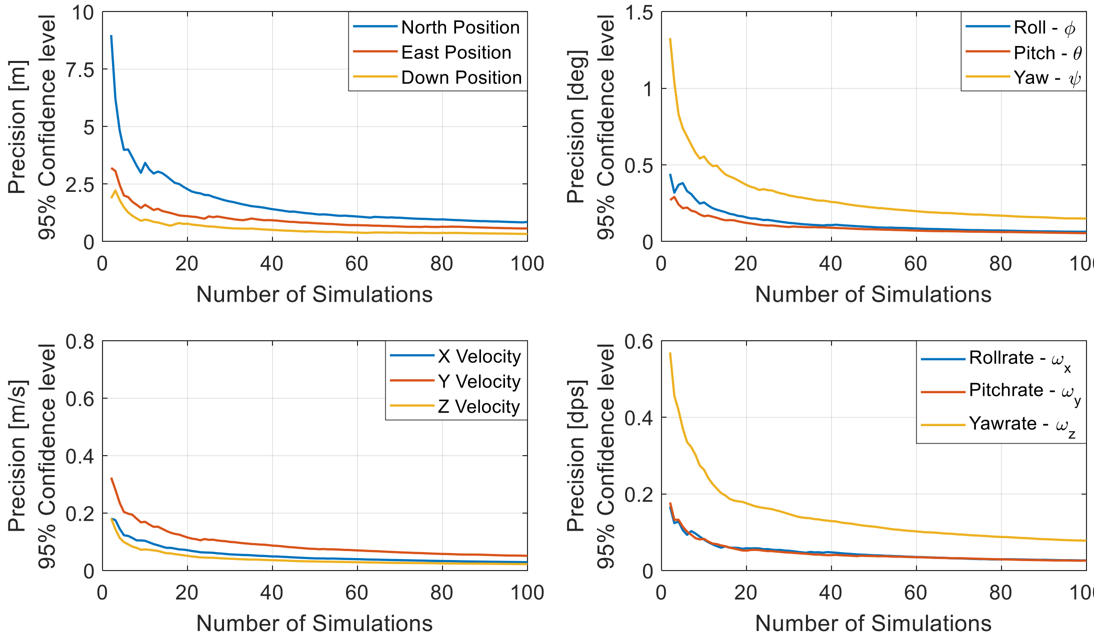


Figure B.1. The precision with a 95% confidence level for different number of simulations.

B.2. Sensitivity Analysis

The purpose of this section is to assess the sensitivity of a VDM-based navigation scheme to random variations (10%) of the VDM parameters. The 10% variation was applied with a random sign $(+/-)$ for the parameter under investigation. Here, a loosely coupled VDM-based scheme using an EKF is used for the sensitivity analysis. The simulation used nominal values, while the navigation filter used perturbed parameters. The state vector includes VDM navigation states, IMU errors and wind velocity states. VDM parameter errors were not estimated within the navigation filter. A GNSS outage was induced 200 seconds into the flight and lasted for over two minutes.

Table B.5 shows the main translational coefficients with a significant impact on the navigation performance. The table shows the maximum and final value state errors. The static thrust coefficient seems to have the largest impact on all navigation states and the wind velocity states. The impact of the lateral force coefficient CF_Y was similar to the drag coefficient CF_X.

Table B.5. VDM-based navigation errors due to translational coefficients.

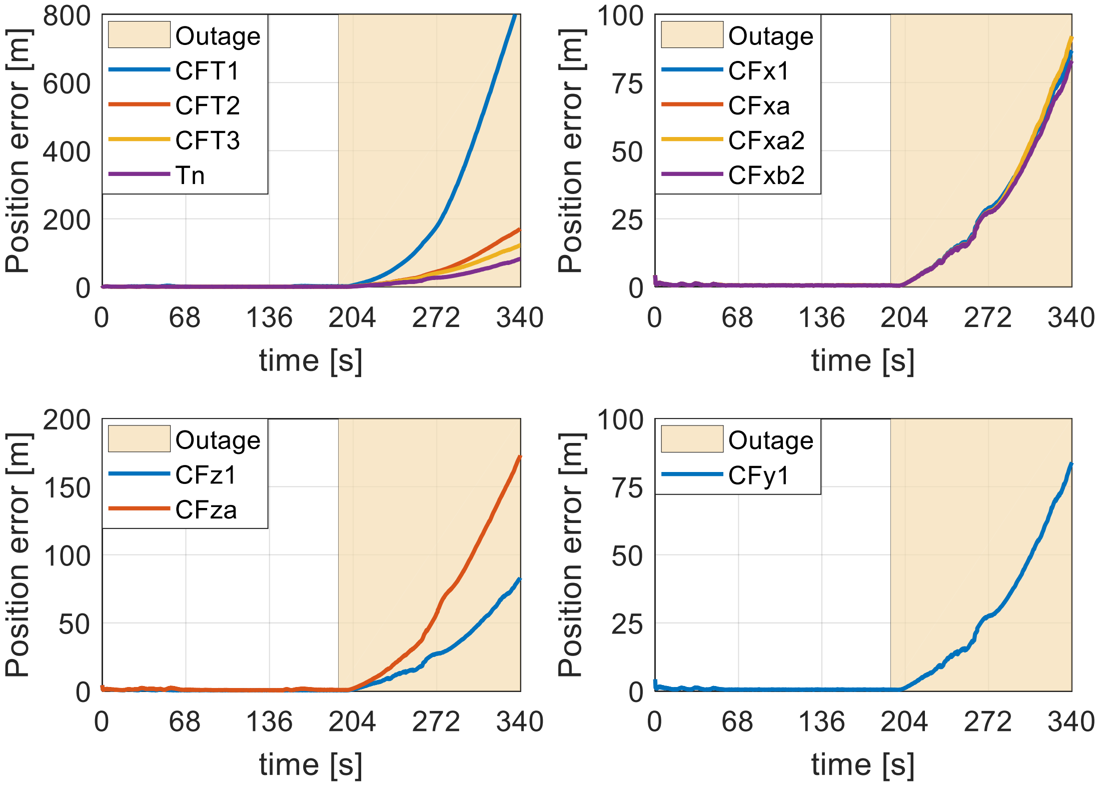
Property	CF_{T_1}		CF_{Z_α}		CF_Y	
	Max	Final	Max	Final	Max	Final
Position error [m]	847.0	847.0	172.0	172.0	80.0	80.0
Velocity [m/s]	10.0	10.0	2.53	2.53	1.4	1.4
Roll error [deg]	7.4	0.6	6.0	0.6	0.7	0.4
Pitch error [deg]	7.5	4.0	4.7	3.0	0.4	0.1
Yaw error [deg]	24.0	24.0	11.0	10.0	6.5	6.5
Wind error [m/s]	9.5	9.5	2.5	2.5	0.4	0.4

Table B.6 shows the main torque coefficients with a significant impact on the navigation performance of a VDM-based scheme. The pitching moment coefficient at the aerodynamic centre seems to have the most impact on the navigation states.

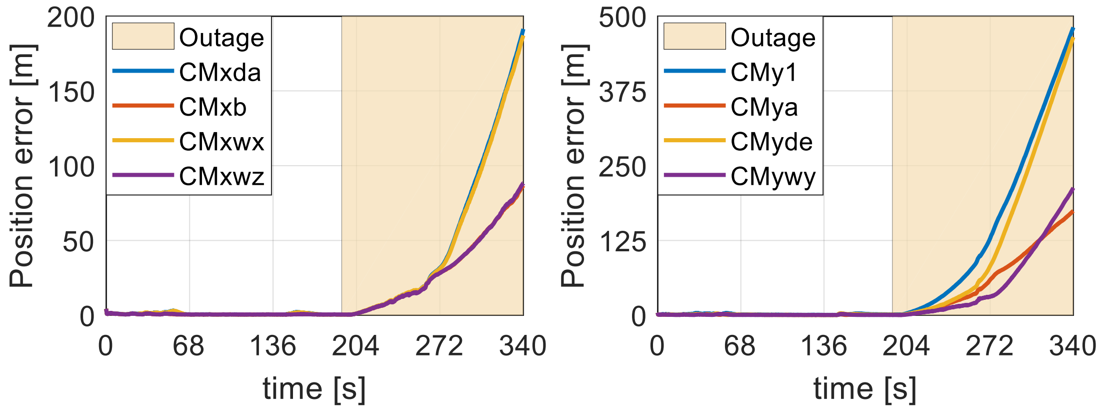
Table B.6. VDM-based navigation errors due to moment coefficients.

Property	$CM_{X_{\delta\alpha}}$		CM_{Y1}		CM_{Z_β}	
	Max	Final	Max	Final	Max	Final
Position error [m]	191.4	191.4	481.1	481.1	86.94	86.94
Velocity [m/s]	3.6	3.4	5.8	5.8	1.5	1.5
Roll error [deg]	6.8	0.4	9.8	0.5	1.3	0.5
Pitch error [deg]	4.1	0.2	8.5	4.7	0.7	0.1
Yaw error [deg]	15.9	14.4	13.0	13.0	6.5	6.5
Wind error [m/s]	1.0	1.0	5.3	5.3	0.4	0.4

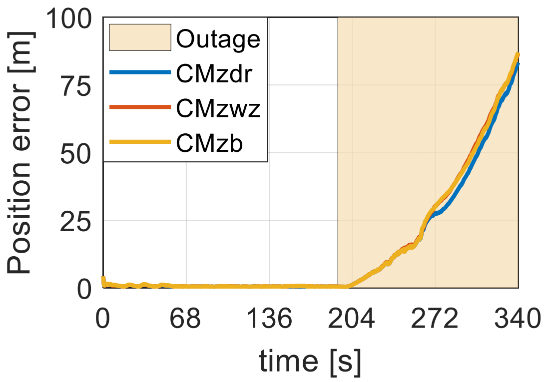
Figure B.2 and Figure B.3 show the position and velocity estimation errors for the VDM parameters under investigation. Figure B.4, Figure B.5, and Figure B.6 show the estimation errors for the roll, pitch, and yaw angles, respectively. Figure B.7 shows the wind speed estimation errors.



a. Thrust, drag, lift and lateral force coefficients

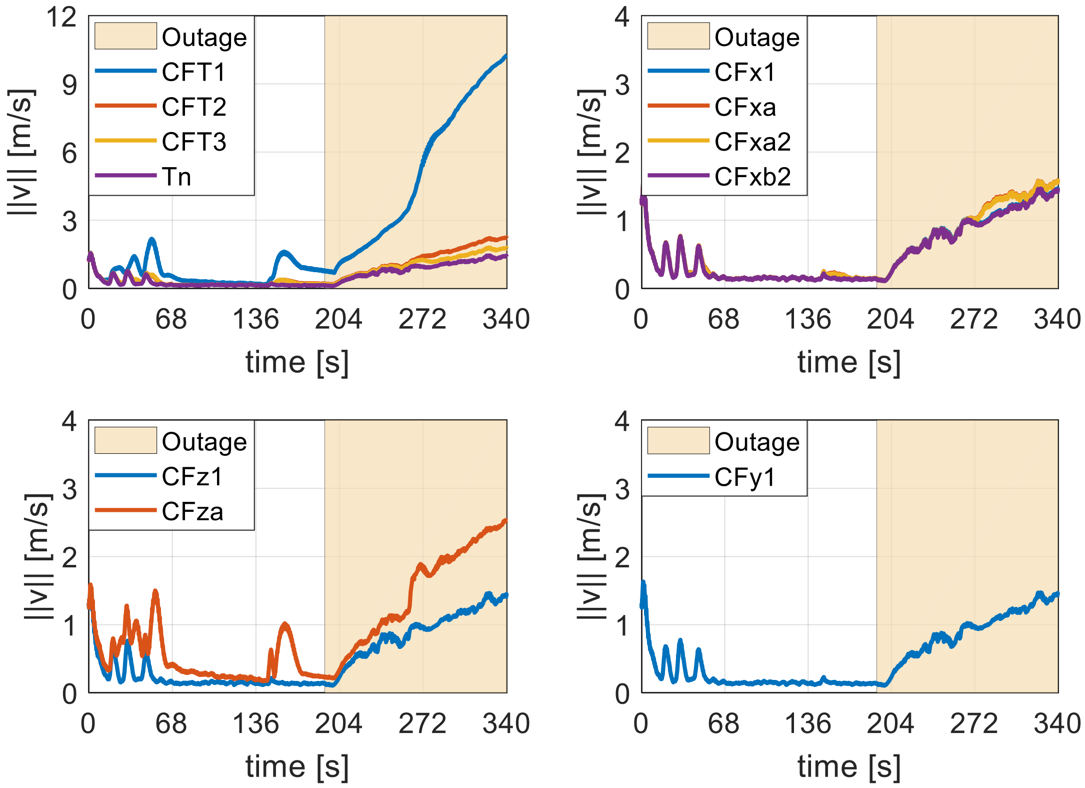


b. Rolling and pitching moment coefficients

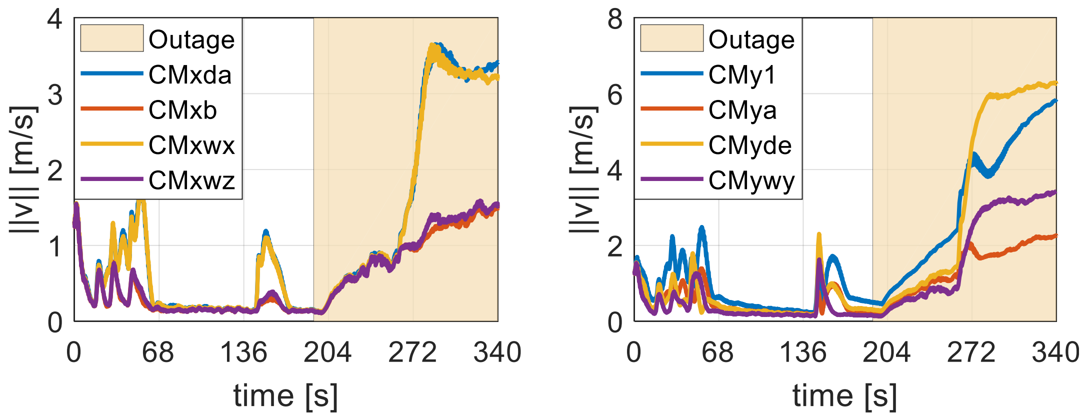


c. Yawing moment coefficients

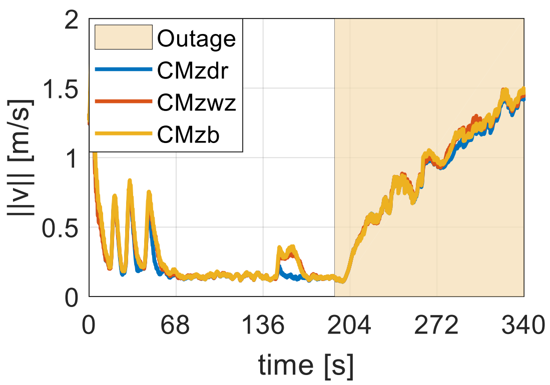
Figure B.2. The impact of VDM parameters on position estimation errors.



a. Thrust, drag, lift and lateral force coefficients

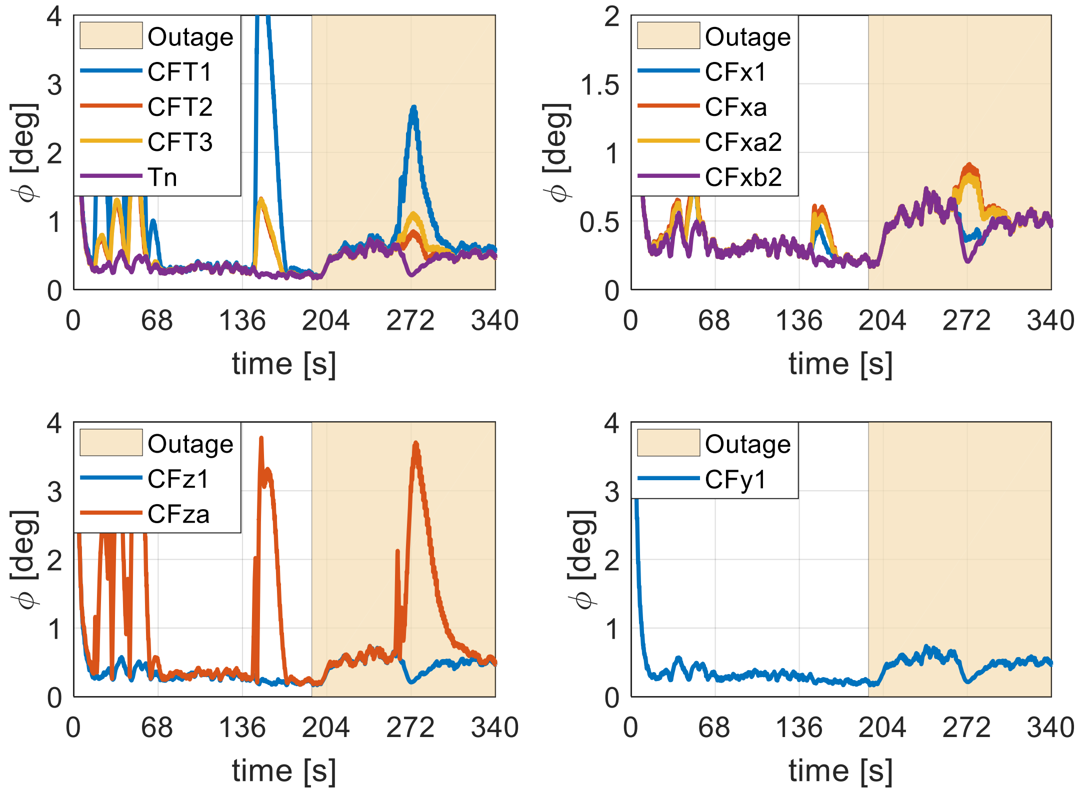


b. Rolling and pitching moment coefficients

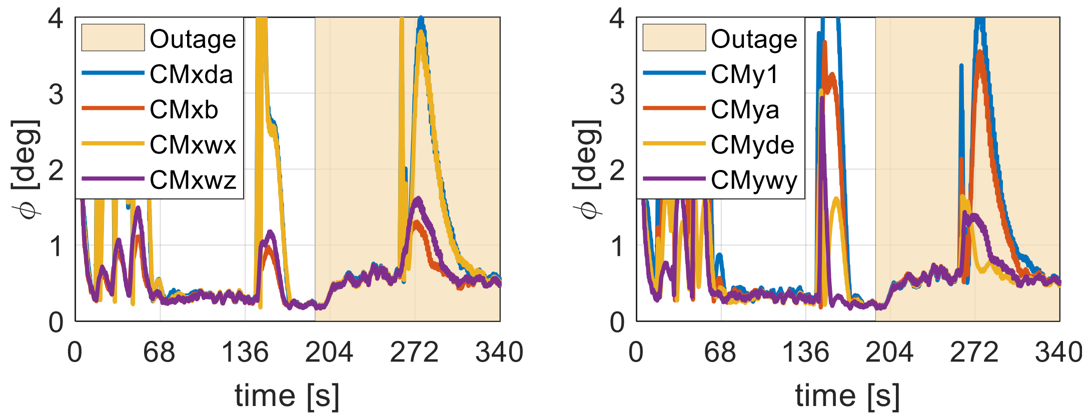


c. Yawing moment coefficients

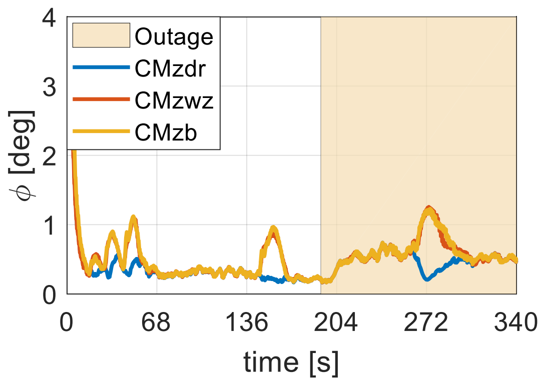
Figure B.3. The impact of VDM parameters on velocity estimation errors.



a. Thrust, drag, lift and lateral force coefficients

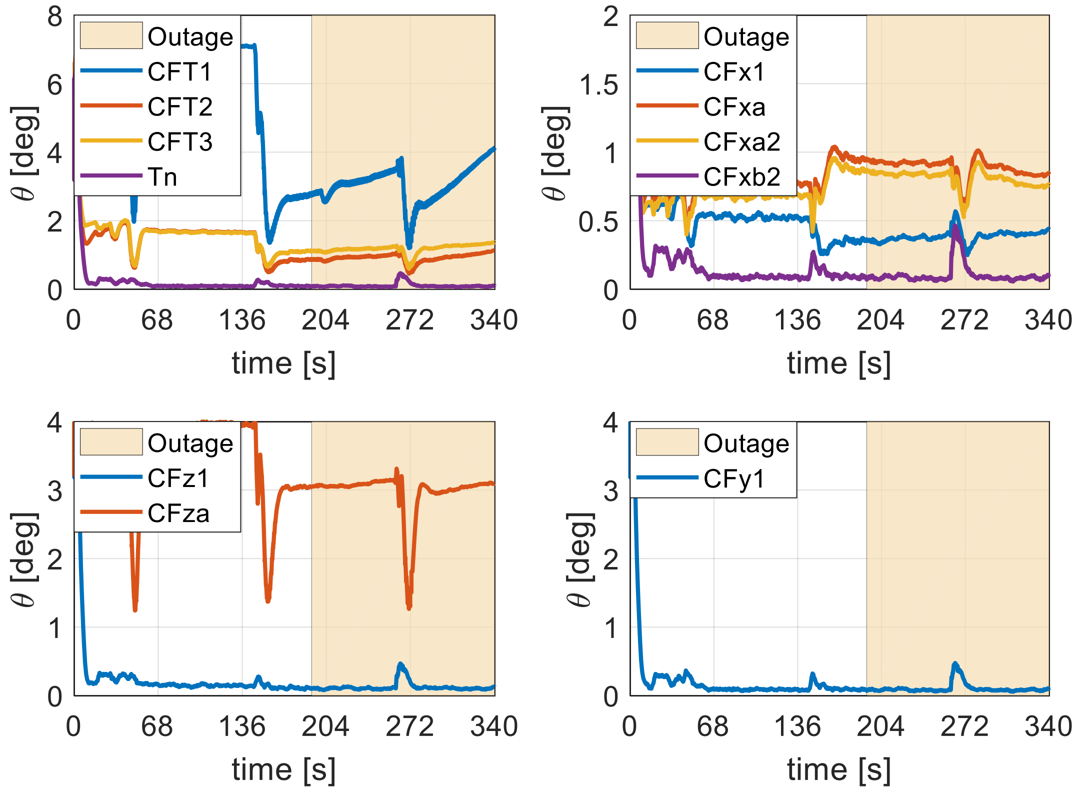


b. Rolling and pitching moment coefficients

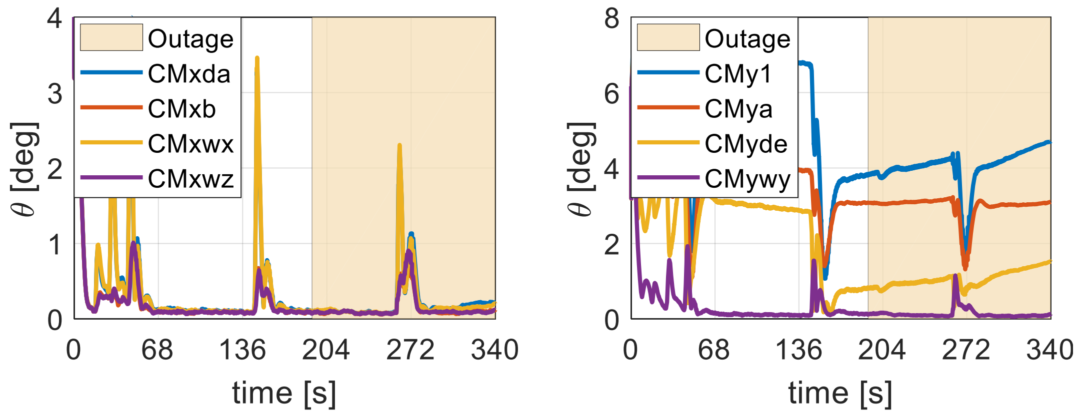


c. Yawing moment coefficients

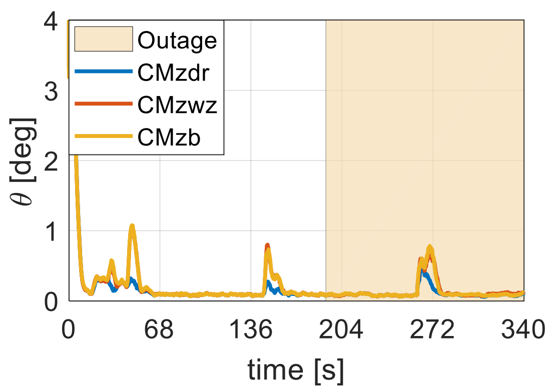
Figure B.4. The impact of VDM parameters on roll angle estimation errors.



a. Thrust, drag, lift and lateral force coefficients

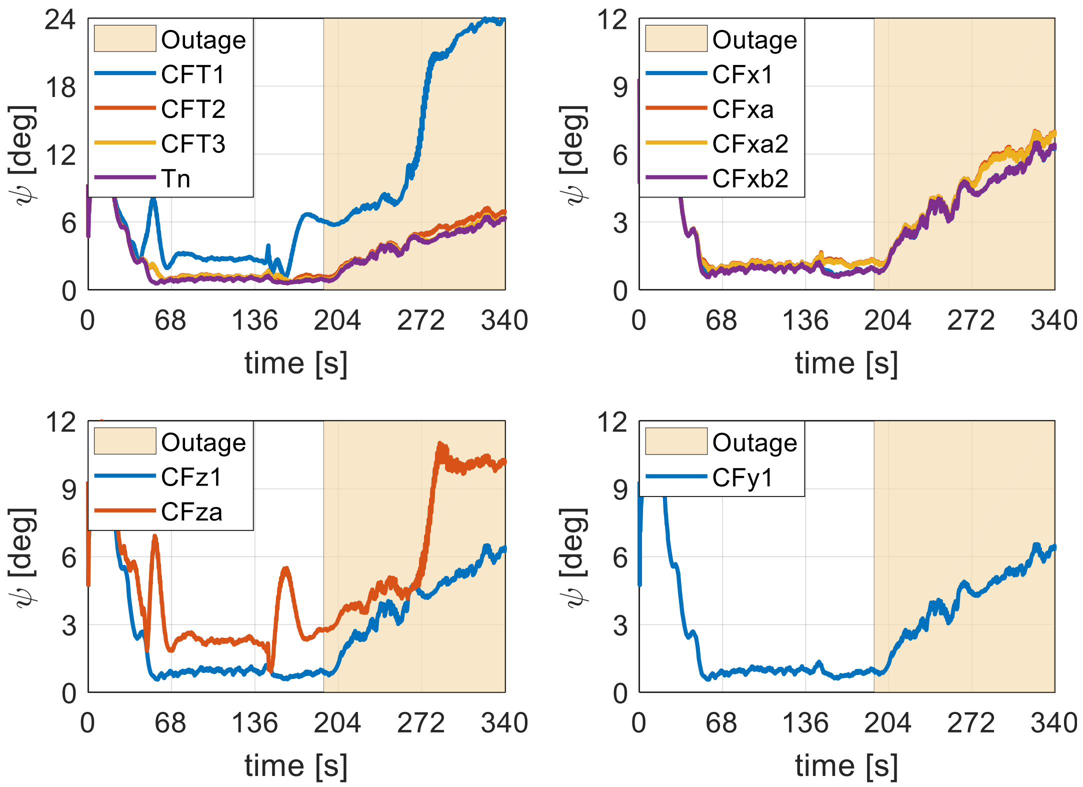


b. Rolling and pitching moment coefficients

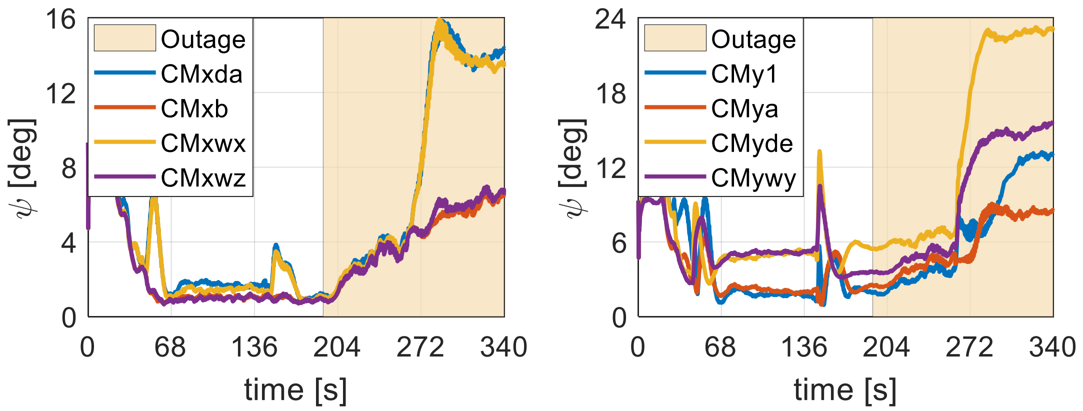


c. Yawing moment coefficients

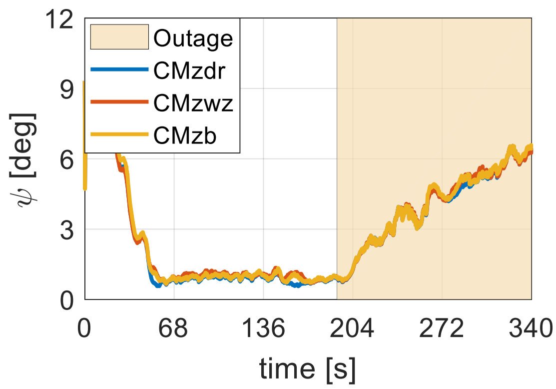
Figure B.5. The impact of VDM parameters on pitch angle estimation errors.



a. Thrust, drag, lift and lateral force coefficients

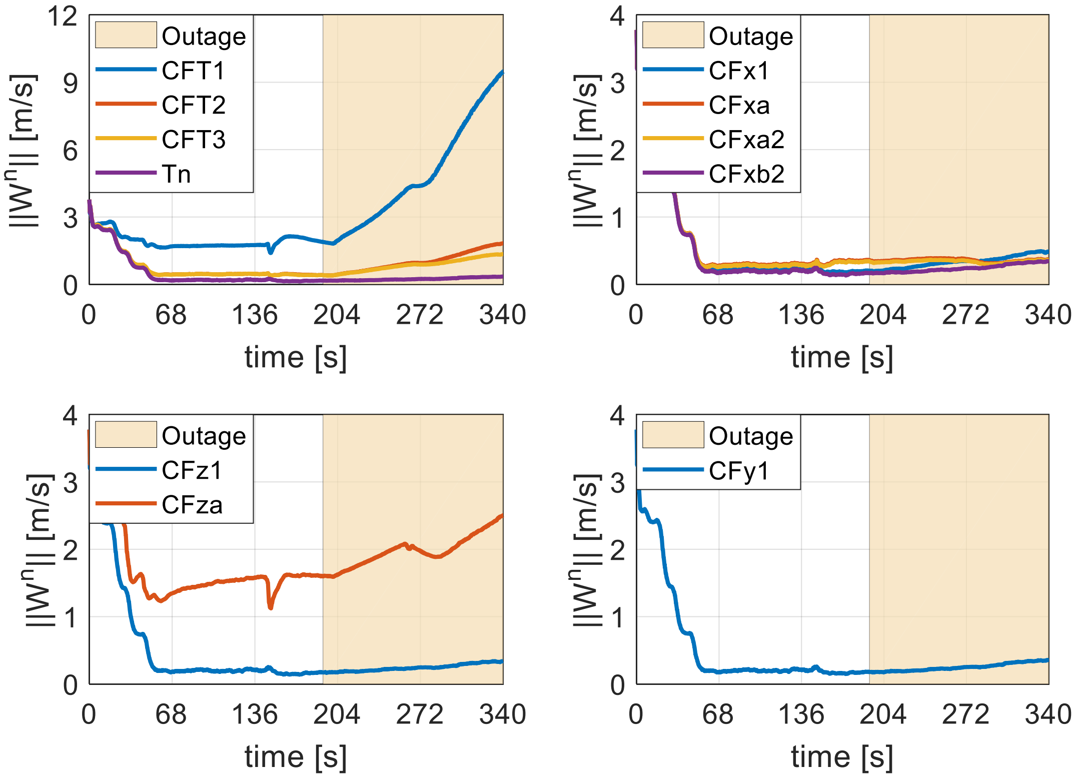


b. Rolling and pitching moment coefficients

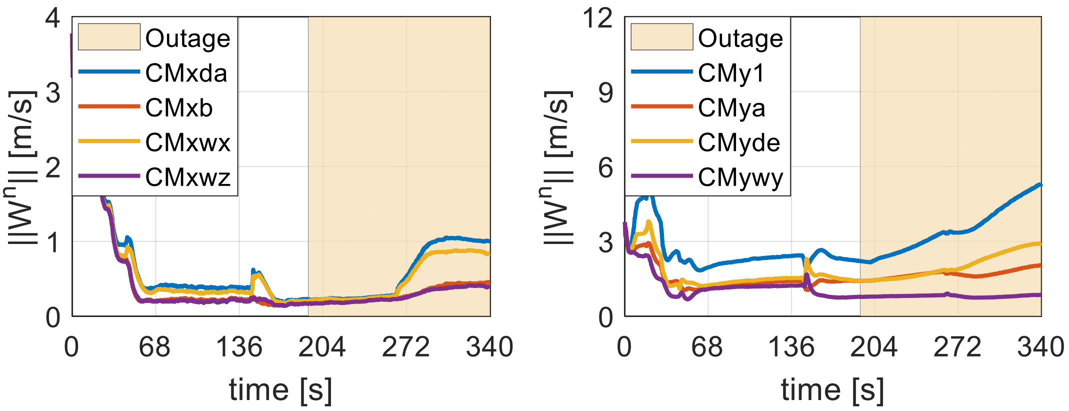


c. Yawing moment coefficients

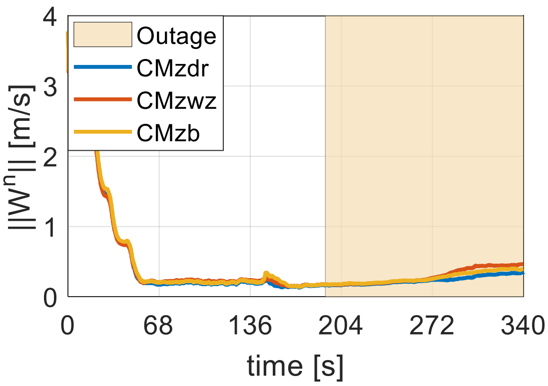
Figure B.6. The impact of VDM parameters on yaw angle estimation errors.



a. Thrust, drag, lift and lateral force coefficients



b. Rolling and pitching moment coefficients



c. Yawing moment coefficients

Figure B.7. The impact of VDM parameters on wind magnitude estimation errors.

B.3. GNSS Measurement Simulator Error Models

This section presents a summary of the error models used in the GNSS measurement simulator. These error models are presented in Table B.7.

Table B.7. GNSS simulator error models.

Ionospheric residual		
First-order Gauss-Markov (σ_{GM})		2 m
Correlation Time (τ)		1800 s
Tropospheric residual		
First-order GM (σ_{GM})		0.2 m
Correlation Time (τ)		1800 s
Multipath		
GM-driving noise	c_0	0.47 m
$\sigma_{WN} = c_0 + c_1 e^{-\frac{Elev}{c_2}}$	c_1	0.78 m
	c_2	20.92°
Correlation Time (τ)		5 - 65 s
Thermal noise		
$\epsilon(\rho) \sim N(0, \sigma_e^2(CN0_r^s)$	b_0	0.05 m
$\sigma_e^2 = c_0 + c_1 . e^{\frac{CN0_r^S - c_2}{c_3}}$	b_1	1.05 m
	b_2	28.0 dB-Hz
	b_3	8.0 dB-Hz
GNSS receiver clock		
Clock offset (σ)		10 km
Clock drift (σ)		20 m/s
Clock drift PSD		$0.1884 \text{ m/s}^{\frac{3}{2}}$
Sampling Frequency		1 Hz

B.4. LCVDM Extended Results

This section presents extended results for the loosely coupled VDM approach using an extended Kalman filter presented in Chapter 4 and the tightly coupled VDM architecture presented in Chapter 5.

The RMS of position errors alongside 100 realisations for the loosely coupled approach is shown in Figure B.8.

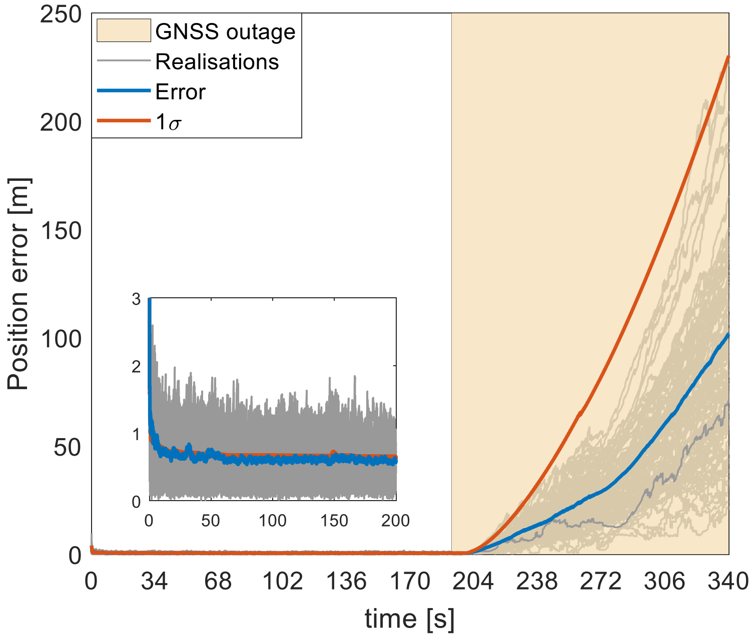


Figure B.8. Position error.

Figure B.9 shows the GNSS position innovations alongside the innovation variances for one of the simulations. In the figure, v_k represents the innovation and S_k represents the innovation covariance matrix.

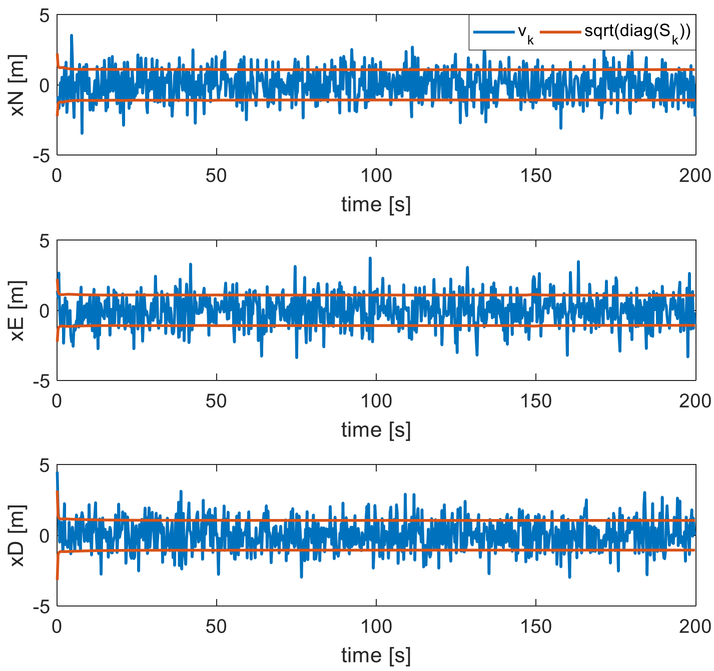


Figure B.9. Position measurement innovations for one realisation out of 100 simulations.

Figure B.10 shows the IMU specific force innovations alongside the innovation variances.

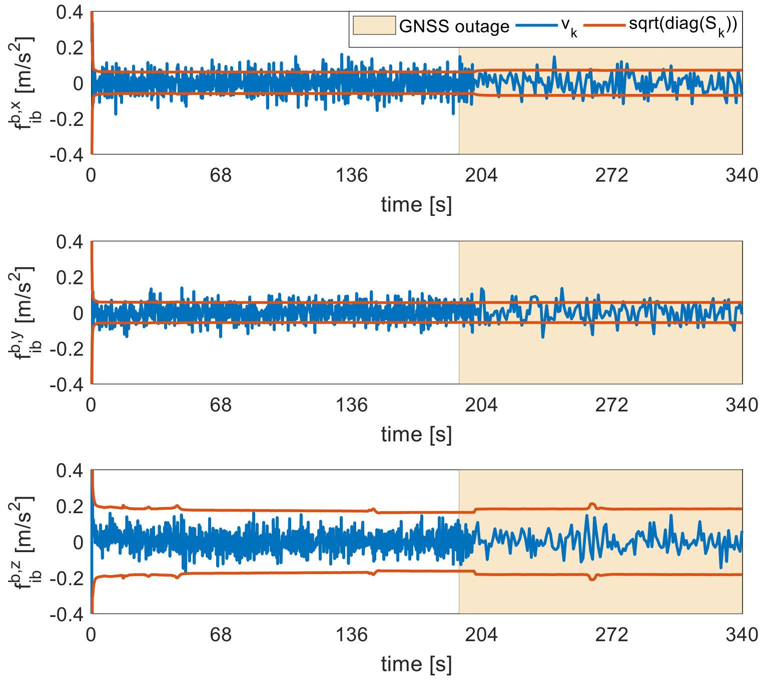


Figure B.10. IMU specific force innovations for one realisation out of 100 simulations.

Figure B.11 shows the IMU rotation rate innovations alongside the innovation variances.

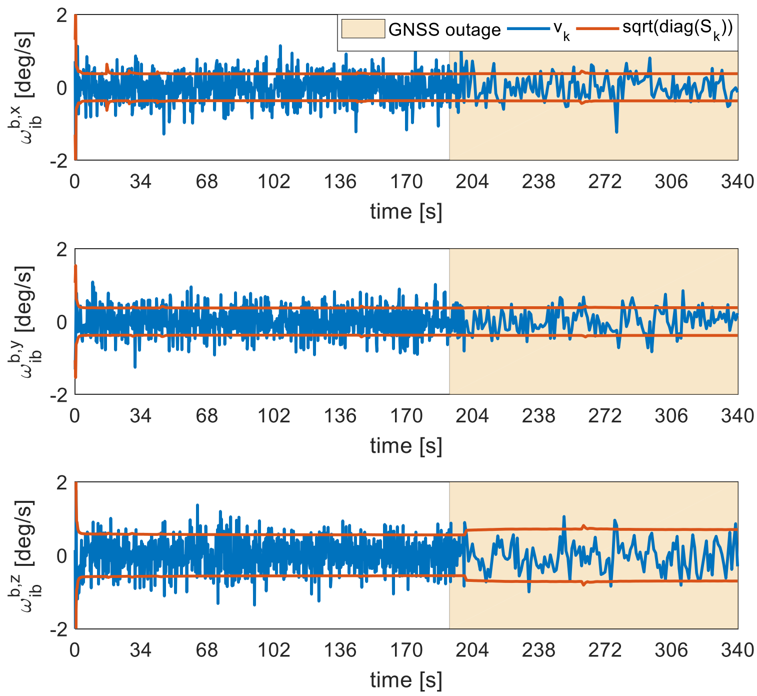


Figure B.11. IMU rotation rate innovations for one realisation out of 100 simulations.

B.5. TCVDM Extended Results

Figure B.12 shows the pseudorange innovations for one realisation out of the 100 simulations for the tightly coupled VDM approach. The plot shows the innovations (v_k) and associated variances (S_k) with three satellites visible during the GNSS outage.

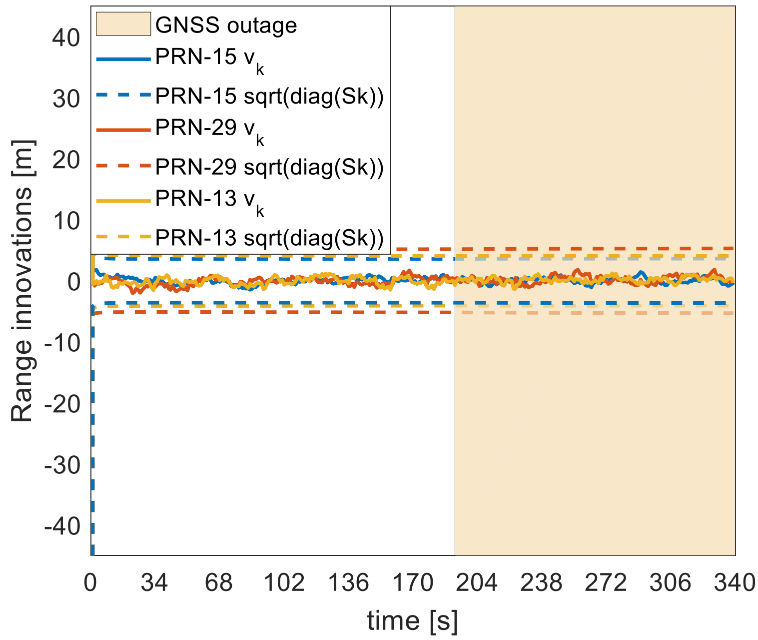


Figure B.12. Pseudorange innovations for one realisation out of 100 simulations.

Figure B.13 shows the Doppler innovations and associated variances for three satellites visible during the GNSS outage. The legend for this figure is similar to the legend in Figure B.12.

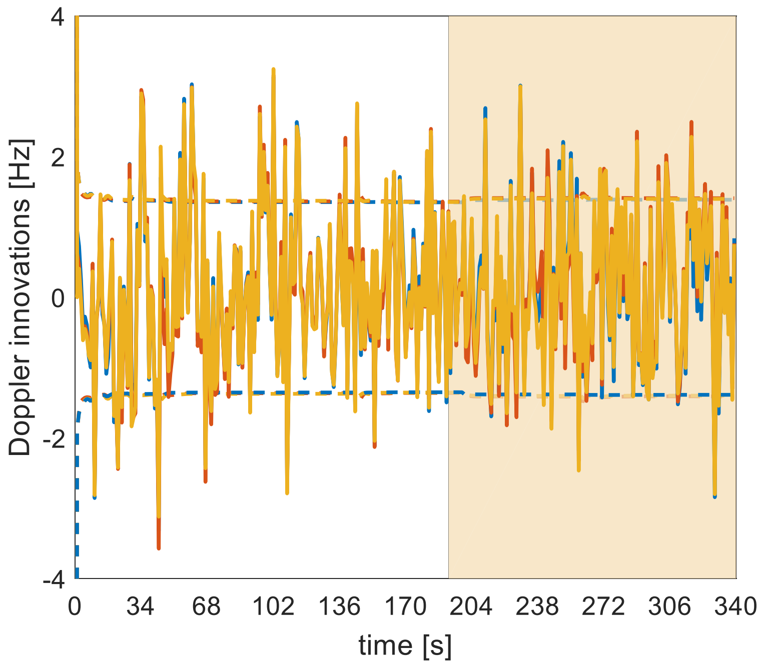


Figure B.13. Doppler innovations for one realisation out of 100 simulations.

Figure B.14 shows the specific force innovations and associated variances with three satellites visible during the GNSS outage.

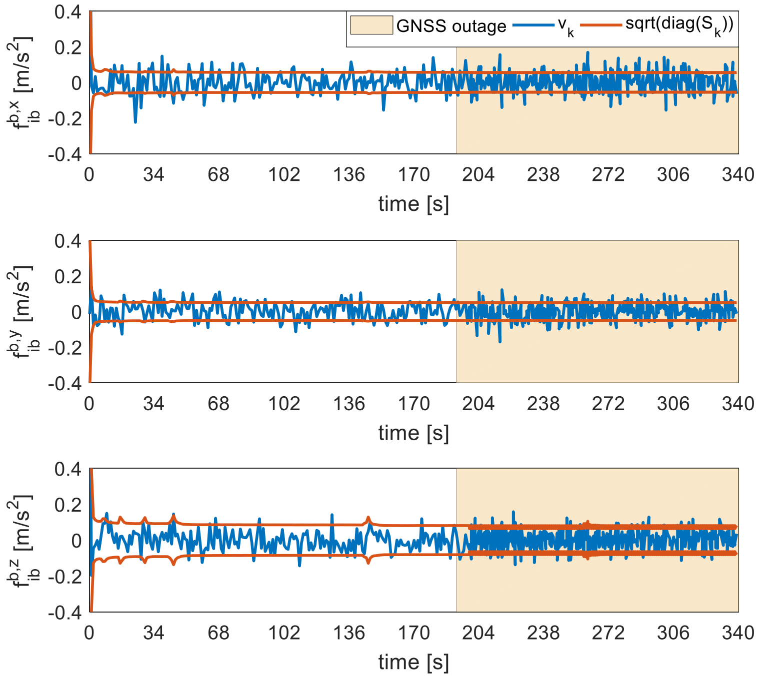


Figure B.14. IMU specific force innovations for one realisation out of 100 simulations.

Figure B.15 shows the rotation rate innovations and associated variances with three satellites visible during the GNSS outage.

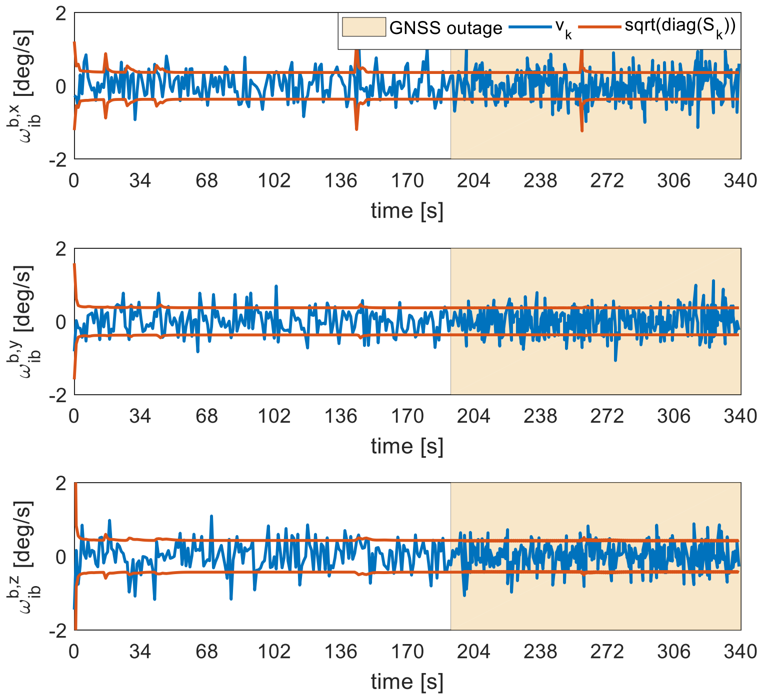


Figure B.15. IMU rotation rate innovations for one realisation out of 100 simulations.

Figure B.16, Figure B.17, and Figure B.18 show the RMS of position estimation errors, velocity estimation errors and attitude estimation errors with one satellite (PRN 15) in view during the GNSS outage. Generally, the navigation performance of the TCVDM scheme with one satellite in view is very similar to the LCVDM scheme. There is marginal improvement in position estimation and even marginal improvement for all other navigation states.

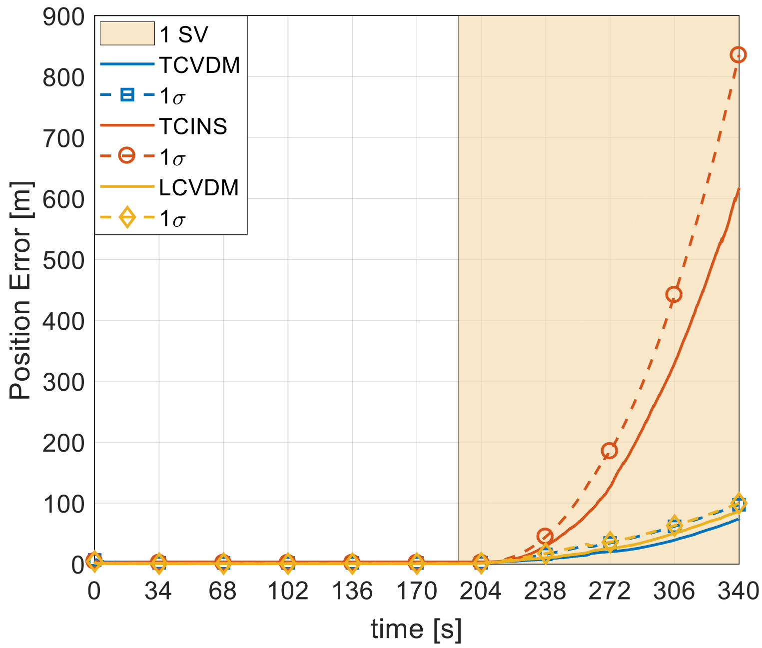


Figure B.16. Position error with one satellite in view (PRN 15) during the GNSS outage.

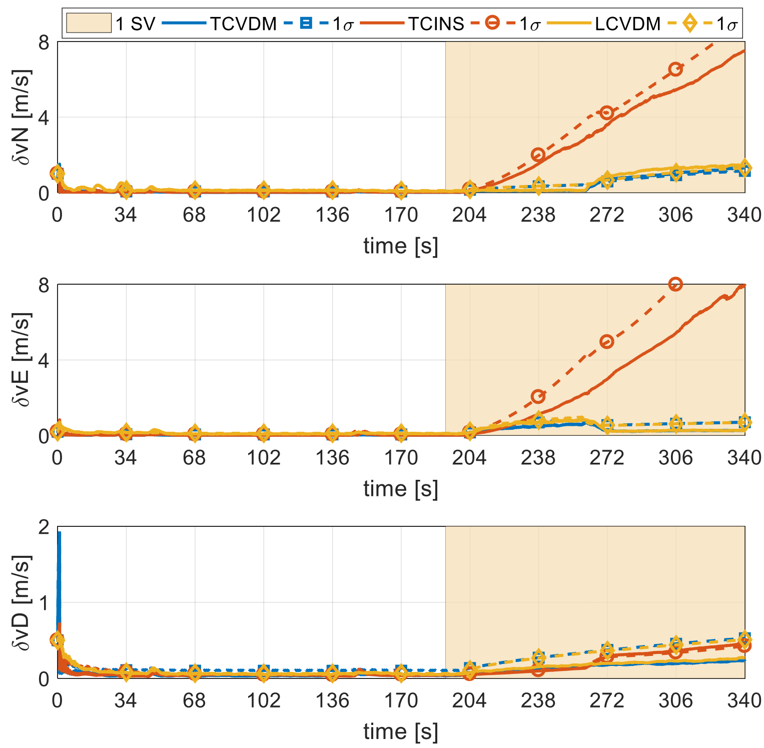


Figure B.17. Velocity errors with one satellite in view (PRN 15) during the GNSS outage.

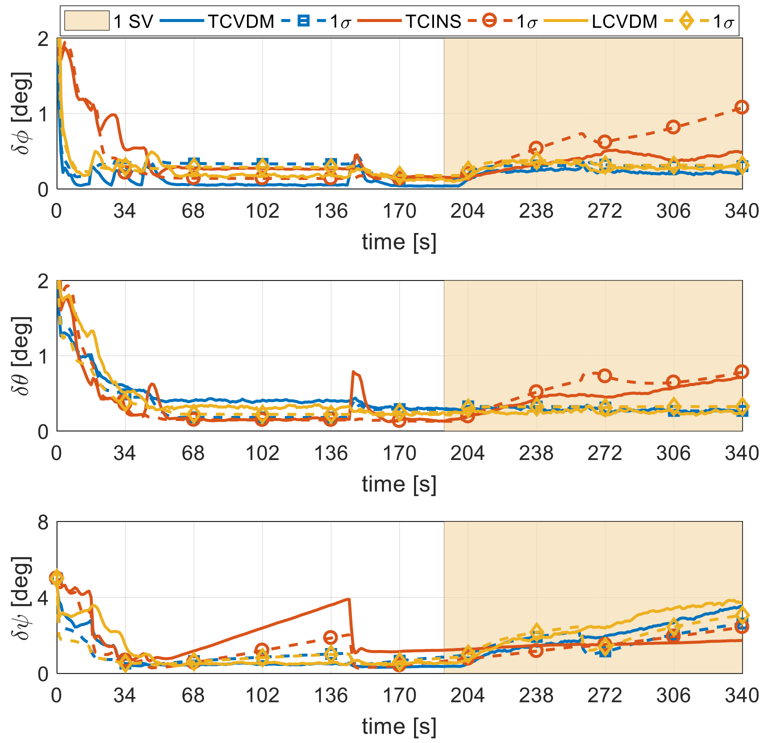


Figure B.18. Attitude errors with one satellite in view (PRN 15) during the GNSS outage.

The final errors for the navigation states are presented in Table B.8.

Table B.8. The final RMS of estimation errors for the navigation states with one satellite in view during the outage.

Property	Final error for the TCVDM	Final error for the LCVDM	Final error for the TCINS
Position [m]	74.01	86.33	616.1
North velocity [m/s]	1.40	1.40	7.538
East velocity [m/s]	0.26	0.26	7.971
Down velocity [m/s]	0.25	0.27	0.460
Roll [deg]	0.31	0.31	0.490
Pitch [deg]	0.27	0.26	0.710
Yaw [deg]	3.52	3.73	1.740

C. Experiment

C.1. BLDC and Propellers

This section presents a brief overview of the three constant BLDC model. The power required to drive a propeller is given by:

$$P_{in} = C_{pow}\rho n^3 D^5 \tag{C.1}$$

Or in terms of the propeller torque Q:

$$P_{in} = 2\pi n Q \tag{C.2}$$

Figure C.1 shows this cubic relationship between the propeller speed and power. Accurate measurement of the propeller speed is a challenging task for small UAVs. In most cases, the propeller speed can be inferred from power measurements, and this information can be used to derive the thrust force. However, this does not consider the losses inside the ESC and the efficiency of the brushless motor.

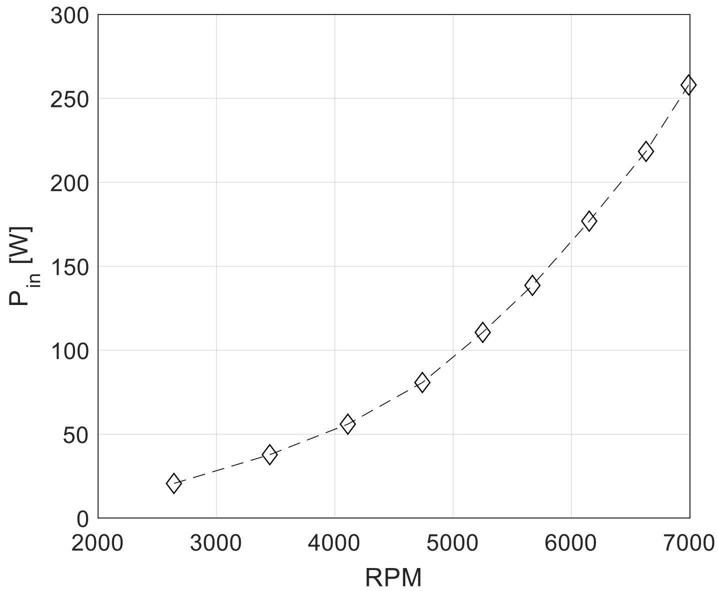


Figure C.1. Power required to drive a propeller.

The electric current in the three-constant BLDC model is given by:

$$I = I_0 + \frac{2\pi \cdot k_V}{60} Q \tag{C.3}$$

where: I_0 is the no-load current (A),
k_V is the voltage constant (RPM/V).

The input voltage is given by:

$$E = E_i + I \cdot R_m \tag{C.4}$$

$$= \omega / k_V + I \cdot R_m \tag{C.5}$$

where: E_i is back emf (V),
R_m is internal motor resistance (Ω).

C.2. Estimated Coefficients

The aerodynamic and propulsion model values estimated using AVL and ground measurements are presented in Table C.1.

Table C.1. Reference values for the aerodynamic and propulsion models.

Property	**Value**	Units	**Property**	**Value**	Units
CF_{T_1}	0.098	$[-]$	CM_{X_β}	-0.050	[rad^{-1}]
CF_{T_2}	-0.120	$[-]$	$CM_{X_{\bar{\omega}_x}}$	-0.400	$[-]$
CF_{T_3}	-0.480	$[-]$	$CM_{X_{\bar{\omega}_z}}$	0.116	$[-]$
CF_{X_1}	-0.024	$[-]$	CM_{Y1}	-0.007	$[-]$
CF_{X_α}	-0.121	[rad^{-1}]	CM_{Y_α}	-1.371	[rad^{-1}]
$CF_{X_{\alpha 2}}$	-1.225	[rad^{-2}]	$CM_{Y_{\delta_e}}$	0.300	$[-]$
$CF_{X_{\beta 2}}$	-0.696	[rad^{-2}]	$CM_{Y_{\bar{\omega}_y}}$	-15.570	$[-]$
CF_{Z_1}	-0.235	$[-]$	$CM_{Z_{\delta_r}}$	0.018	$[-]$
CF_{Z_α}	-4.481	[rad^{-1}]	$CM_{Z_{\bar{\omega}_z}}$	-0.193	$[-]$
CF_{Y1}	-0.096	[rad^{-1}]	CM_{Z_β}	0.149	[rad^{-1}]
$CM_{X_{\delta_\alpha}}$	0.055	$[-]$	τ_n	0.200	s

Control Inputs

$$\delta_\alpha = 0.0017 \cdot PWM - 2.65$$

$$\delta_e = 0.0017 \cdot PWM - 2.52$$

$$\delta_r = 0.0018 \cdot PWM - 2.53$$

Note: PWM $[\mu s]$

ESC efficiency

$$\eta\_esc = \alpha_{esc} \cdot PWM + \beta_{esc} \cdot I + \gamma_{esc}$$

$$\alpha_{esc} = 0.0012$$

$$\beta_{esc} = -0.0287$$

$$\gamma_{esc} = -0.7797$$

Note: ESC efficiency at roughly 12 V

C.3. Allan Variance

This section gives a brief overview of the Allan variance technique used to characterise different sources of noise exhibited by inertial sensors. Given a set of N consecutive data points with a sample time t_0, a group of n consecutive data points can be formed such that $n < N/2$. The cluster average from the instantaneous output rate is given by:

$$\bar{\Omega}_k(T) = \frac{1}{T}\int_{t_k}^{t_k+T} \Omega(t)dt \tag{C.6}$$

where: $\bar{\Omega}_k(T)$ is the cluster average for the specific cluster time T starting from the kth data point,
$\Omega(t)$ is the instantaneous output.

And the difference between two adjacent clusters is given by:

$$\xi_{k+1,k} = \bar{\Omega}_{next}(T) - \bar{\Omega}_k(T) \tag{C.7}$$

The ensemble of this difference for the cluster time is a set of random variables, and the quantity of interest is the variance over all clusters of the same size. Therefore, the Allan variance is given by:

$$\sigma^2(T) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} [\bar{\Omega}_{next}(T) - \bar{\Omega}_k(T)]^2 \tag{C.8}$$

The different random processes affecting inertial sensors can be easily derived from the Allan variance using the unique relationship between it and the power spectral density (PSD) of the intrinsic random process. This assumes that the specific random process $\Omega(T)$ is stationary in time. This relationship is given by (El-Sheimy, Hou and Niu, 2008):

$$\sigma^2(T) = 4\int_0^\infty df \cdot S_\Omega(f) \cdot \frac{\sin^4(\pi f T)}{(\pi f T)^2} \tag{C.9}$$

By defining the PSD of the random processes affecting inertial sensors and evaluating the integral, it is possible to determine the value of the PSD using the log-log plot of the measurable $\sigma^2(T)$ and the cluster time. This is summarised in Table C.2. The definitions of the PSD's of the random processes alongside some derivations can be found in El-sheimy, Hou and Niu (2015).

The percentage error in estimating the Allan variance is dependent on the number of independent clusters used to evaluate it. This is given by:

$$\sigma(\delta) = \frac{1}{\sqrt{2\left(\frac{N}{n}-1\right)}} \tag{C.10}$$

Table C.2. The relationship between the Allan-variance and associated noise PSD.

Noise	PSD	Allan variance	Slope
Quantization noise	$S_\Omega(f) \approx (2\pi f)^2 T_s Q_z^2$	$\sigma^2(T) = \frac{3Q_z^2}{T^2}$	-1
Angle (Velocity) random walk	$S_\Omega(f) = N_0^2$	$\sigma^2(T) = \frac{N_0^2}{T}$	-1/2
Bias Instability	$S_\Omega(f) = \left(\frac{B^2}{2\pi}\right)\frac{1}{f}$	$\sigma^2(T) \approx \frac{2B^2}{\pi}\ln 2$	0
Rate random walk	$S_\Omega(f) = \left(\frac{K}{2\pi}\right)^2 \frac{1}{f^2}$	$\sigma^2(T) = \frac{K^2 T}{3}$	+1/2
Drift rate ramp	$S_\Omega(f) = \frac{R^2}{(2\pi f)^3}$	$\sigma^2(T) = \frac{R^2 T^2}{2}$	+1

C.4. RTKLIB Settings

The purpose of this section is to present the RTKLIB configuration settings for the u-blox NEO-M8T receivers.

```
# rtkpost options (2020/04/14 22:38:33, v.demo5 b33c)
pos1-posmode       =kinematic  # (0:single,1:dgps,2:kinematic,3:static,4:static-start,5:movingbase,6:fixed,7:ppp-kine,8:ppp-static,9:ppp-fixed)
pos1-frequency     =l1         # (1:l1,2:l1+l2,3:l1+l2+l5,4:l1+l2+l5+l6)
pos1-soltype       =combined   # (0:forward,1:backward,2:combined)
pos1-elmask        =25         # (deg)
pos1-snrmask_r     =off        # (0:off,1:on)
pos1-snrmask_b     =off        # (0:off,1:on)
pos1-snrmask_L1=38,38,38,38,38,38,38,38,38
pos1-snrmask_L2=0,0,0,0,0,0,0,0,0
pos1-snrmask_L5=0,0,0,0,0,0,0,0,0
pos1-dynamics      =on         # (0:off,1:on)
pos1-tidecorr      =off        # (0:off,1:on,2:otl)
pos1-ionoopt       =brdc       # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec,6:qzs-brdc,7:qzs-lex,8:stec)
pos1-tropopt       =saas       # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad,5:ztd)
pos1-sateph        =brdc       # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+ssrcom)
pos1-posopt1       =off        # (0:off,1:on)
out-fieldsep       =
out-outsingle      =off        # (0:off,1:on)
out-maxsolstd      =0          # (m)
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal   # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all        # (0:all,1:single)
out-nmeaintv1      =0          # (s)
out-nmeaintv2      =0          # (s)
out-outstat        =residual   # (0:off,1:state,2:residual)
stats-weightmode   =elevation  # (0:elevation,1:snr)
stats-eratio1      =300
stats-eratio2      =300
stats-eratio5      =300
stats-errphase     =0.003      # (m)
stats-errphaseel   =0.003      # (m)
stats-errphasebl   =0          # (m/10km)
stats-errdoppler   =1          # (Hz)
stats-snrmax       =52         # (dB.Hz)
stats-stdbias      =30         # (m)
stats-stdiono      =0.03       # (m)
```

```
pos1-posopt2      =off       # (0:off,1:on)
pos1-posopt3      =off       #
(0:off,1:on,2:precise)
pos1-posopt4      =off       # (0:off,1:on)
pos1-posopt5      =off       # (0:off,1:on)
pos1-posopt6      =off       # (0:off,1:on)
pos1-exclsats     =          # (prn ...)
pos1-navsys       =15        #
(1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
pos2-armode       =fix-and-hold #
(0:off,1:continuous,2:instantaneous,3:fix-and-
hold)
pos2-gloarmode   =fix-and-hold #
(0:off,1:on,2:autocal,3:fix-and-hold)
pos2-bdsarmode =off       # (0:off,1:on)
pos2-arfilter      =on        # (0:off,1:on)
pos2-arthres       =3
pos2-arthres1      =0.1
pos2-arthres2      =0
pos2-arthres3      =1e-09
pos2-arthres4      =1e-05
pos2-varholdamb    =0.1       # (cyc^2)
pos2-gainholdamb   =0.01
pos2-arlockcnt     =120
pos2-minfixsats    =4
pos2-minholdsats   =5
pos2-mindropsats   =10
pos2-rcvstds       =off       # (0:off,1:on)
pos2-arelmask      =15        # (deg)
pos2-arminfix      =100
pos2-armaxiter     =1
pos2-elmaskhold    =15        # (deg)
pos2-aroutcnt      =30
pos2-maxage        =0.3       # (s)
pos2-syncsol       =off       # (0:off,1:on)
pos2-slipthres     =0.05      # (m)
pos2-rejionno      =1000      # (m)
pos2-rejgdop       =30
pos2-niter         =3
pos2-baselen       =0         # (m)
pos2-basesig       =0         # (m)
out-solformat      =xyz       #
(0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on        # (0:off,1:on)
out-outopt         =on        # (0:off,1:on)
out-outvel         =off       # (0:off,1:on)
out-timesys        =gpst      # (0:gpst,1:utc,2:jst)
out-timeform       =hms       # (0:tow,1:hms)
out-timendec       =3
out-degform        =deg       # (0:deg,1:dms)
```

```
stats-stdtrop      =0.3       # (m)
stats-prnaccelh    =3         # (m/s^2)
stats-prnaccelv    =2         # (m/s^2)
stats-prnbias      =0.0001    # (m)
stats-prniono      =0.001     # (m)
stats-prntrop      =0.0001    # (m)
stats-prnpos       =0         # (m)
stats-clkstab      =5e-12     # (s/s)
ant1-postype       =rinexhead  #
(0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:
rtcm,6:raw)
ant1-pos1          =0         # (deg|m)
ant1-pos2          =0         # (deg|m)
ant1-pos3          =0         # (m|m)
ant1-anttype       =
ant1-antdele       =0         # (m)
ant1-antdeln       =0         # (m)
ant1-antdelu       =0         # (m)
ant2-postype       =llh       #
(0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:
rtcm,6:raw)
ant2-pos1          =53.048905065 # (deg|m)
ant2-pos2          =-1.291562956 # (deg|m)
ant2-pos3          =189.154600000009 #
(m|m)
ant2-anttype       =
ant2-antdele       =0         # (m)
ant2-antdeln       =0         # (m)
ant2-antdelu       =0         # (m)
ant2-maxaveep      =1
ant2-initrst       =on        # (0:off,1:on)
misc-timeinterp    =off       # (0:off,1:on)
misc-sbasatsel     =0         # (0:all)
misc-rnxopt1       =
misc-rnxopt2       =
misc-pppopt        =
file-satantfile    =
file-rcvantfile    =
file-staposfile    =
file-geoidfile     =
file-ionofile      =
file-dcbfile       =
file-eopfile       =
file-blqfile       =
file-tempdir       =
file-geexefile     =
file-solstatfile   =
file-tracefile     =
```

C.5. Comparisons Between Two Tail Baseline Solutions

Two tail baseline solutions are computed using two independent reference receivers. The first solution ($||b_1||$ and $||b_2||$) uses the master GNSS receiver on the aircraft (GM) alongside the tail and wingtip receivers to compute this solution. The second solution ($||\text{G1} - \text{GM}||$ and $||\text{G2} - \text{GM}||$) uses the Leica GS10 receiver on the ground as the reference and computes independent solutions to the master receiver (GM), tail receiver (G1), and wingtip receiver (G2) on the aircraft. Both the G1 and GM receivers had a fixed solution 99% of the time, while the G2 receiver had a fixed solution 96% of the time. The baseline solution is then computed as the difference between the tail/wingtip and master solutions. The two baseline solutions are then compared, and the results are shown in Figure C.*2*.

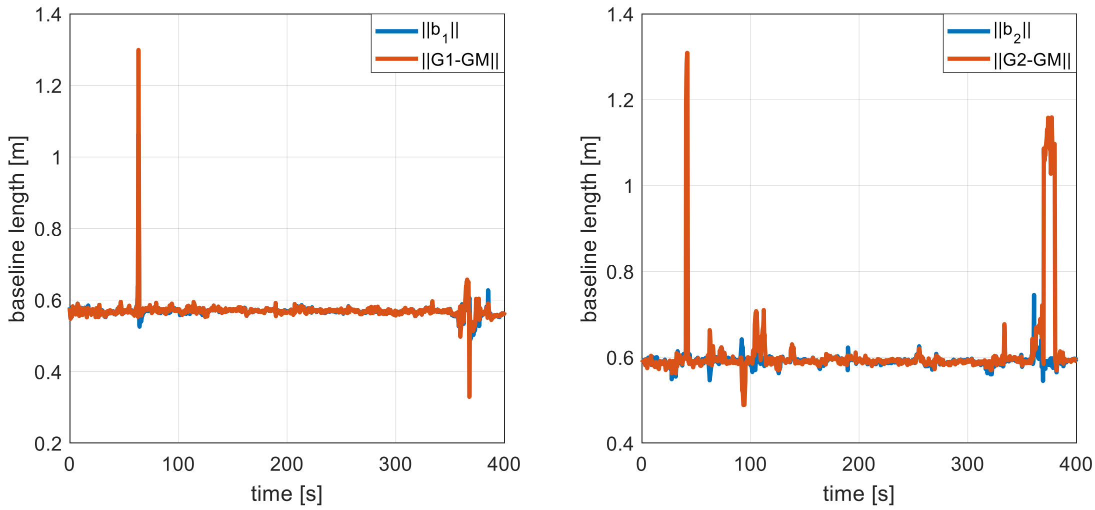


Figure C.2. GNSS tail and wingtip baseline lengths comparison.

The two solutions are very similar but with increased noise when using the Leica GS10 receiver. This might be attributed to the increased decorrelation of the errors between the ground-based receiver and the ones on the aircraft, high rate multipath and also different antenna on the aircraft and ground-based receiver. The wingtip baseline solution showed increased noise as opposed to the tail baseline solution attributed to the increased level of dynamics about the roll axis and other secondary effects such as flexure.

The tail and wingtip baseline components in the body frame (in metres) are given by:

$$F^b = \begin{bmatrix} -0.563 & -0.009 \\ 0.0 & -0.591 \\ 0.072 & 0.0 \end{bmatrix} \tag{C.11}$$

The estimation of attitude components is assessed using the difference between the baseline components in the NED frame with the local body components projected in the NED frame ($R_b^n F^{bi}$ where i is the column). Figure C.3 shows this difference for the tail and wingtip baseline components.

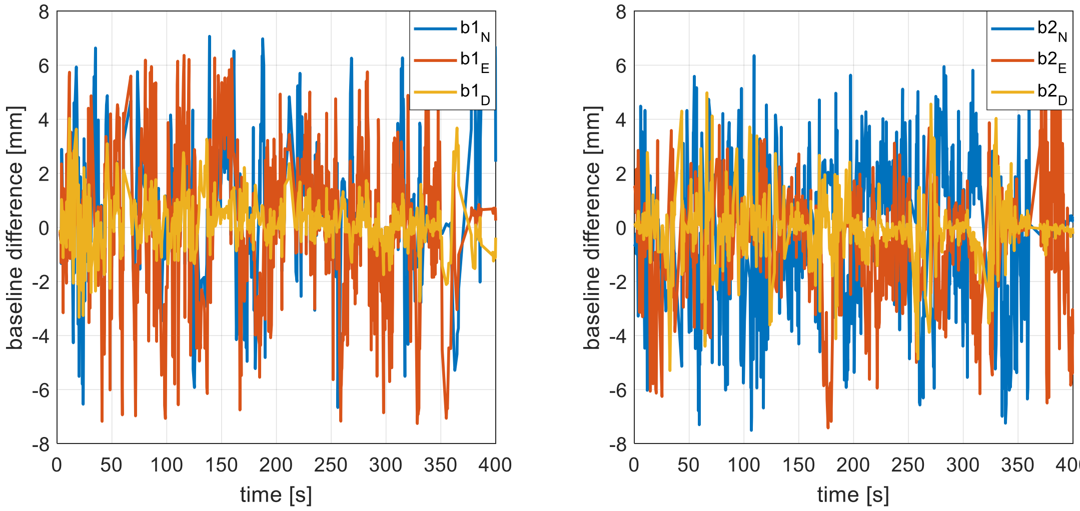


Figure C.3. Baseline components difference during the test flight.

The standard deviation for the tail baseline is around 2.2 mm for the north component, 2.7mm for the east component, and 0.9 mm for the down component. And the standard deviation for the wingtip baseline is around 2.7 mm for the north component, 2.1 mm for the east component, and 1.1 mm for the down component. Further, the two baselines on the aircraft, b_1 and b_2, can be compared using independently estimated aircraft yaw angles. Ideally, once resolved to the aircraft body frame, the two baselines should produce identical yaw results. In practice, the estimates will be different. Figure C.4 shows the heading angle estimation using the two baseline components. Not all epochs have been estimated as a result of not having a fixed solution.

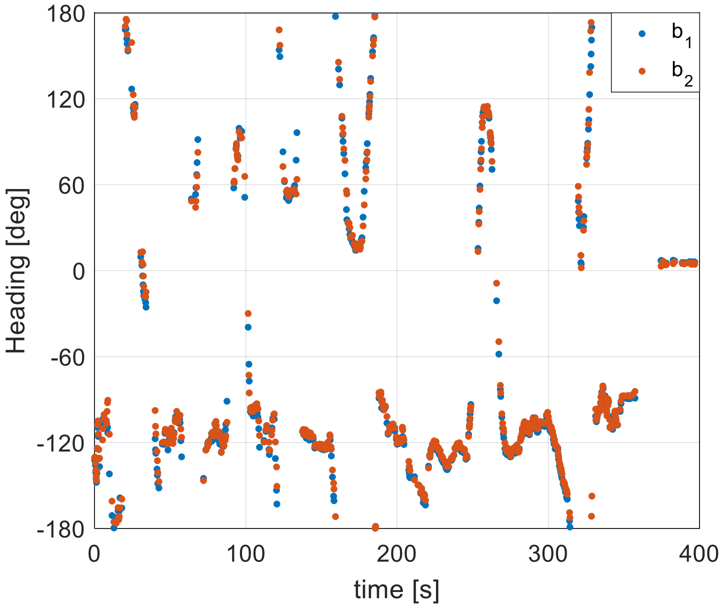


Figure C.4. Heading estimation using the two baseline vectors independently.

The difference between the estimated yaw angles using the two baselines is shown in Figure C.5. The standard deviation is around 1 degree for the results.

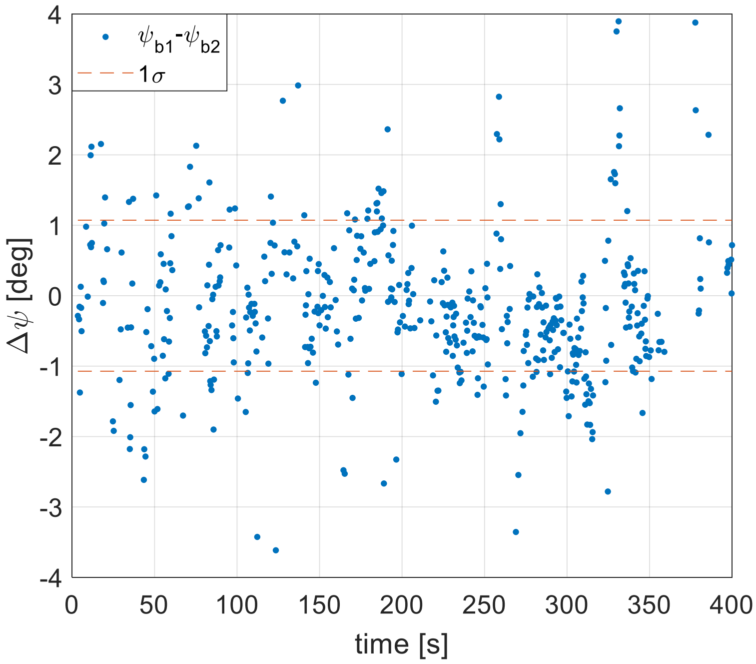


Figure C.5. The difference in yaw angle between the two baselines.

D. Derivations

D.1. Linear Dynamics

In linear dynamics, we usually consider a force acting to accelerate a body. Newton's second law states that the rate of change of momentum of a body is directly proportional to the force applied and occurs in the same direction as the applied force. Considering an inertial frame i, this can be written as:

$$\sum_j F_j = \left[\frac{d}{dt}(mv)\right]_i \tag{D.1}$$

In Equation (D.1), all the components are considered to be in the inertial frame. The time derivative of an arbitrary vector resolved in a different frame other than its reference frame also depends on the relative rotation between the frames.

$$v_{ib}^b = R_i^b v_{ib}^i \tag{D.2}$$

To simplify our notation since we are only working with two frames: $v_{ib}^i = v^i$ and $v_{ib}^b = v^b$. The time derivative of the inertial velocity in the body frame is given by:

$$\dot{v}^b = R_i^b[\dot{v}^i] - \Omega_{ib}^b v^b \tag{D.3}$$

This can also be represented as:

$$\dot{v}^b = R_i^b\left[\dot{v}^i\right] - \omega_{ib}^b \times v^b \tag{D.4}$$

Assuming that mass does not change over a short period, we can recall our previous equation as:

$$\frac{1}{m}\left[\sum_j F_j\right] = \left[\frac{d}{dt}(v)\right]_i \tag{D.5}$$

Noting that:

$$\left[\dot{v}^i\right] = \frac{1}{m}\left[\sum_j F_j\right] = \left[\frac{d}{dt}(v)\right]_i \tag{D.6}$$

Therefore Equation (D.4) can be written as:

$$\dot{v}^b = \frac{1}{m}\left[\sum_j F_j^b\right] - \omega_{ib}^b \times v^b \tag{D.7}$$

where:

$$\frac{1}{m}\left[\sum_j F_j^b\right] = \frac{1}{m}\left[mg^b + F_T^b + F_{aero}^b\right] \tag{D.8}$$

Substituting Equation (D.8) into Equation (D.7) and noting that $v^b = [u, v, w]^T$ and $\omega_{ib}^b = [p, q, r]^T$ we get:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \frac{1}{m}\left[mg^b + F_T^b + F_{aero}^b\right] - \begin{bmatrix}0 & -r & q\\ r & 0 & -p\\ -q & p & 0\end{bmatrix} . \begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{D.9}$$

D.2. Rotational Dynamics

In rotational dynamics, we usually consider a torque or a moment acting to rotate a body. Torque is analogue of the concept of force used in linear dynamics. Torque is defined as the product of force and its perpendicular distance from the point of application to the axis of rotation.

$$\sum_j M = \left[\frac{d}{dt}(I\omega)\right]_i \tag{D.10}$$

The product $I\omega$ is the angular momentum (H) analogue of linear momentum (p). The torque/moment acting on an aircraft is generated by the deflection of the primary control surfaces ($\delta_\alpha, \delta_e, \delta_r$). The term I is the mass moment of inertia matrix of the aircraft. This gives the body mass distribution around the origin. Therefore it is convenient to express the moments and the angular momentum in the aircraft's body-fixed frame similar to the force. Therefore the rate of change of angular momentum in the body frame can be expressed as:

$$\dot{H}^b = \sum_j M^b - \omega_{ib}^b \times H^b \tag{D.11}$$

This can also be written as:

$$(I\dot{\omega_{ib}^b}) = \sum_j M^b - \omega_{ib}^b \times I\omega_{ib}^b \tag{D.12}$$

Assuming the mass moment of inertia matrix is constant i.e. $\dot{I} = 0$. The equation then becomes:

$$I\dot{\omega}_{ib}^b = \sum_j M^b - \omega_{ib}^b \times I\omega_{ib}^b \tag{D.13}$$

where

$$\sum_j M^b = M_{aero}^b + M_{prop}^b + M_{gyroscopic}^b \tag{D.14}$$

where:

$$M_{aero}^b = [M_X, M_Y, M_Z]^T \tag{D.15}$$

$$M_{prop}^b = \rho n^2 D^5 CM_p \tag{D.16}$$

$$M_{gyroscopic}^b = \begin{bmatrix} 0 \\ -j_m \cdot r \cdot (n_{cw} \cdot 2\pi) \\ j_m \cdot q \cdot (n_{cw} \cdot 2\pi) \end{bmatrix} \tag{D.17}$$

where j_m is the inertia of the rotating component, n_{cw} is the clockwise propeller speed given in rev/s, r and q hold their usual meaning. It is also assumed that the motor-propeller assembly is mounted along the longitudinal axis of a fixed-wing aircraft and that the propeller is spinning clockwise when viewed from the rear.

E. INS/GNSS Architectures

This section describes the loosely coupled and tightly coupled INS/GNSS integration architectures used in performance comparison with the architectures used in Chapters 4, 5 and 6.

E.1. Loosely Coupled INS/GNSS

Figure E.*1* shows the implemented loosely coupled INS/GNSS integration architecture. The architecture is implemented using an error state extended Kalman filter comprising 15 states. Other than the navigation states, the state vector also includes IMU error terms.

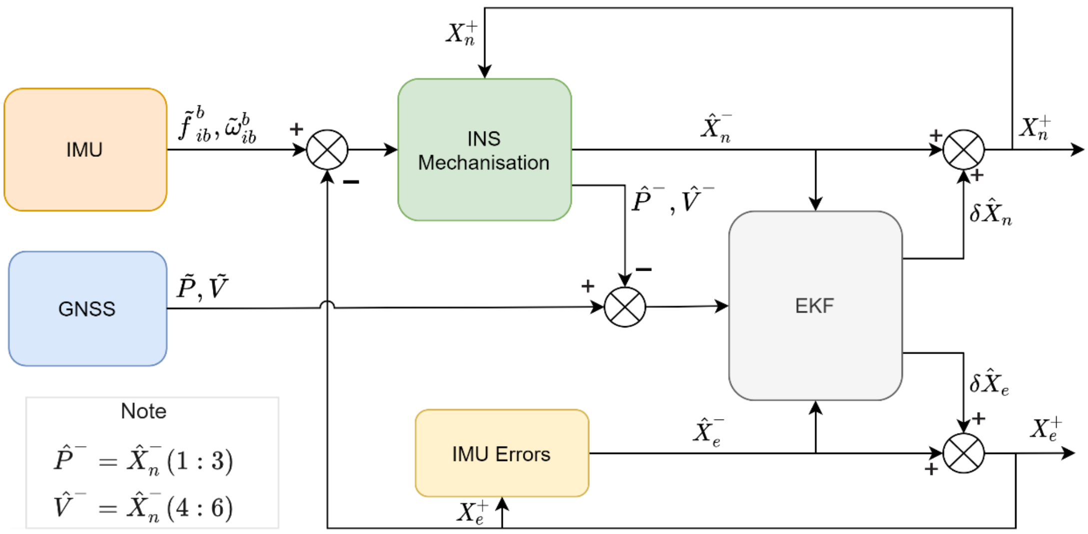


Figure E.1. The implemented loosely coupled INS/GNSS architecture.

The total state vector (X) is given by:

$$X = [x_N, x_E, x_D, v_x^b, v_y^b, v_z^b, \phi, \theta, \psi, b_{ax}, b_{ay}, b_{az}, b_{gx}, b_{gy}, b_{gz}]^T \tag{E.1}$$

The process models for the navigation states (in the total state form) are given in the following equations:

$$\dot{x}^n = R_b^n v^b \tag{E.2}$$

$$\dot{v}^b = f_{ib}^b + (R_b^n)^T g^n - \omega_{ib}^b \times v^b \tag{E.3}$$

$$\dot{\phi}_{nb} = R_w \, \omega_{ib}^b \tag{E.4}$$

In the filter, the IMU errors are modelled using a random-walk process, while in the simulation environment, the IMU errors are modelled using a first-order Gauss-Markov process. The random-walk process used in the filter is given by:

$$\dot{b}_{a,i|g,i} = w_{rnd} \quad \text{(E.5)}$$
$$\text{with } i = [x, y, z]$$

where $b_{a,i|g,i}$ represents the accelerometer and gyroscope biases and w_{rnd} represents the driving white noise.

The measurement vector consists of GNSS receiver measurements given by:

$$Z_{GNSS} = \begin{bmatrix} x_N \\ x_E \\ x_D \end{bmatrix} + w_g \quad \text{(E.6)}$$

where w_g represents the residual error for the GNSS measurements modelled as Gaussian white noise.

The process and observation models are linearised to obtain the dynamic matrix ($F_{ij} = \partial \dot{X}_i / \partial X_j$) and observation matrix ($H = \partial Z / \partial X$) used in the EKF covariance propagation and measurement update steps.

Table E.*1* shows the modelled stochastic properties of the IMU and GNSS receiver used in the simulation environment.

Table E.1. Stochastic properties for IMU and GNSS receiver.

Sensor	**Type**	**Value**
Accelerometer	Random bias (σ)	10 mg
	White noise (PSD)	100 μg/√Hz
	GM-Process	0.05 mg
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
Gyroscope	Random bias (σ)	1000 °/hr
	White noise (PSD)	21.6 °/hr/√Hz
	GM-Process	20 °/hr
	Correlation time (τ)	200 s
	Sampling Frequency	100 Hz
GNSS Receiver	White noise (σ)	5 m
	Sampling Frequency	1 Hz

Table E.*2* shows the standard deviation of the initial errors used in the filter.

Table E.2. The standard deviation of the initial errors for the states

State	**Standard deviation (1σ)**
Position	[1.0,1.0,1.0] m
Velocity	[1.0, 0.5, 0.5] m/s
Attitude	[3.5°, 3.5°, 5.0°]
Accelerometer bias	[10, 10, 10] mg
Gyroscope bias	[1000,1000,1000] deg/hr

And Table E.3 shows the standard deviation of the tuned process noise used in the filter.

Table E.3. Process noise.

State	Standard deviation (1σ)
Position	10^{-6} m
Velocity	10^{-3} m/s
Attitude	10^{-4} rad
Accelerometer Bias	2×10^{-4} m/s^2
Gyroscope Bias	2×10^{-6} rad/s

E.2. Tightly Coupled INS/GNSS

Figure E.2 shows the implemented tightly coupled INS/GNSS integration architecture. The architecture is implemented using an error state extended Kalman filter comprising of 17 states. Other than the navigation states and the IMU error terms, the state vector also includes the GNSS receiver clock errors terms (the receiver clock offset and drift).

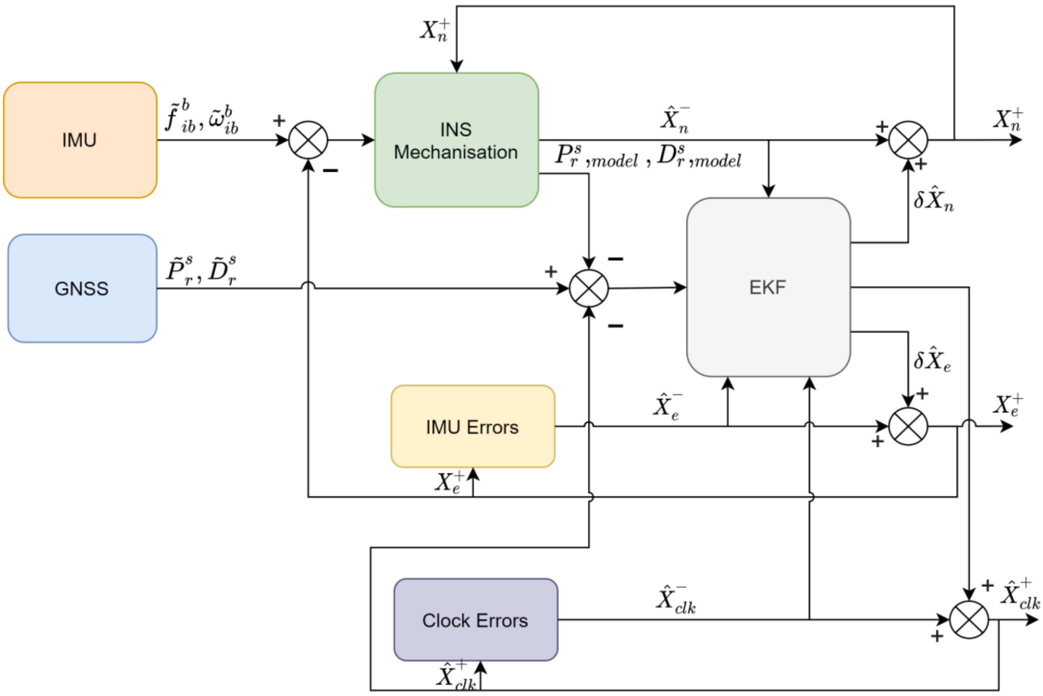


Figure E.2. The implemented tightly coupled INS/GNSS architecture.

The total state vector (X) is given by:

$$X = \begin{bmatrix} \mu, \lambda, h, v^n_{eb,N}, v^n_{eb,E}, v^n_{eb,D}, q_0, q_1, q_2, \dots \\ \dots q_3, b_{ax}, b_{ay}, b_{az}, b_{gx}, b_{gy}, b_{gz}, b_{clk}, d_{clk} \end{bmatrix}^T \tag{E.7}$$

The total state vector has 18 elements, but the error state vector has 17 elements because the aircraft's orientation error is represented using the rotation vector instead of the quaternion. The relationship between the quaternion error and the rotation vector is given by: $2\delta\dot{q} = \delta\dot{\phi}^{b}$.

The process models for the navigation states (in the total state form) are given by:

$$\dot{p}_b = \left[\frac{v_{eb,N}^{n}}{R_M + h}, \frac{v_{eb,E}^{n}}{(R_P + h)\cos(\mu)}, -v_{eb,D}^{n}\right]^T \tag{E.8}$$

$$\dot{v}_{eb}^{n} = R_b^n f_{ib}^{b} + g^n - (2\Omega_{\mathrm{ie}}^{n} + \Omega_{en}^{n})v_{eb}^{n} \tag{E.9}$$

$$\begin{aligned}\dot{q}_b^n &= \frac{1}{2} q_b^n \otimes \left[\omega_{nb}^{b}\right] \\ &= \frac{1}{2}\left[\omega_{nb}^{b}\right]_R q_b^n\end{aligned} \tag{E.10}$$

In the filter, the IMU errors are modelled using a random-walk process. In the simulation environment, the IMU errors are modelled using a first-order Gauss-Markov process. The random-walk process used in the filter is given by:

$$\begin{gathered}\dot{b}_{a,i|g,i} = w_{rnd} \\ \text{with } i = [x, y, z]\end{gathered} \tag{E.11}$$

where $b_{a,i|g,i}$ represents the accelerometer and gyroscope biases and w_{rnd} represents the driving white noise. A two-state random process is used to model the receiver clock errors in both the filter and the simulation environment, albeit with different values used for each setup. The two-state random process is given by:

$$\begin{gathered}\dot{b}_{clk} = d_{clk} + w_{uf} \\ \dot{d}_{clk} = w_{ug}\end{gathered} \tag{E.12}$$

where b_{clk} is the receiver clock bias from system time (m), d_{clk} is the receiver clock drift (m/s) and w_{uf} and w_{ug} represent the driving white noise terms.

The observation vector consists of raw GNSS observables (pseudoranges and Doppler frequencies) from the GNSS receiver. These are given by:

$$Z_{GNSS} = \begin{bmatrix}\tilde{P}_r^s \\ \tilde{D}_r^s\end{bmatrix} = \begin{bmatrix}\rho_r^s + b_{clk} \\ -(\frac{f_i}{c}([v_{es}^{e} - v_{er}^{e}]^T e_r^s + d_{clk})\end{bmatrix} + w_g \tag{E.13}$$

It is assumed that the ionospheric and tropospheric effects have been partially corrected using the Klobuchar and Saastamoinen models, respectively. It is also assumed that the satellite clock corrections have been applied. v_{es}^{e} and v_{er}^{e} represent the satellite and receiver velocity vectors in the ECEF frame. All the other terms have been defined in Chapter 5.

The process and observation models are linearised to obtain the dynamic matrix ($F_{ij} = \partial\dot{X}_i/\partial X_j$) and observation matrix ($H = \partial Z/\partial X$) used in the EKF covariance propagation and measurement update steps. It is important to note that the quaternion is expressed in terms of the rotation vector for the linearisation step, as explained in Chapter 5.

Table E.4 shows the IMU error characteristics used in the simulation environment.

Table E.4. IMU error characteristics.

Property	**Accelerometer**	**Gyroscope**
Random bias (σ)	40 mg	1000 °/hr
White noise (PSD)	0.5 mg/$\sqrt{\text{Hz}}$	126°/hr/$\sqrt{\text{Hz}}$
First-order Gauss–Markov	0.05 mg	20 °/hr
Correlation Time (τ)	200 s	200 s
Sampling Frequency	100 Hz	100 Hz

The models used for the raw GNSS observables in the simulation environment of the tightly coupled INS/GNSS scheme are the same as those presented in Chapter 5. Similarly, the measurement uncertainty used in the filter is the same as that defined in Chapter 5 for both the pseudoranges and Doppler frequencies.

Table E.5 shows the standard deviation of the initial errors used in the filter.

Table E.5. Initial uncertainties for the states.

State	**Standard deviation (σ)**
Position	[2, 2, 3] m
Velocity	[1.0, 0.5, 0.5] m/s
Attitude	[3.5°, 3.5°, 5°]
Accelerometer bias	[40, 40, 40] mg
Gyroscope bias	[1000,1000,1000] deg/h
Clock offset	10^4 m
Clock drift	10 m/s

Table E.6 shows the standard deviation of the diagonal terms of the tuned process noise covariance matrix.

Table E.6. The process noise used in the filter.

State	**Standard deviation (σ)**
Position	10^{-4} m
Velocity	10^{-4} m/s
Attitude	10^{-4} rad
Accelerometer bias	2×10^{-5} m/s^2
Gyroscope bias	2×10^{-6} rad/s
Receiver clock	[0.01 m, 0.02 m/s]

E.3. Reference INS/GNSS Architecture

Figure E.3 shows the setup used to derive the reference navigation solution used to assess the performance of the TCVDM scheme using real flight data.

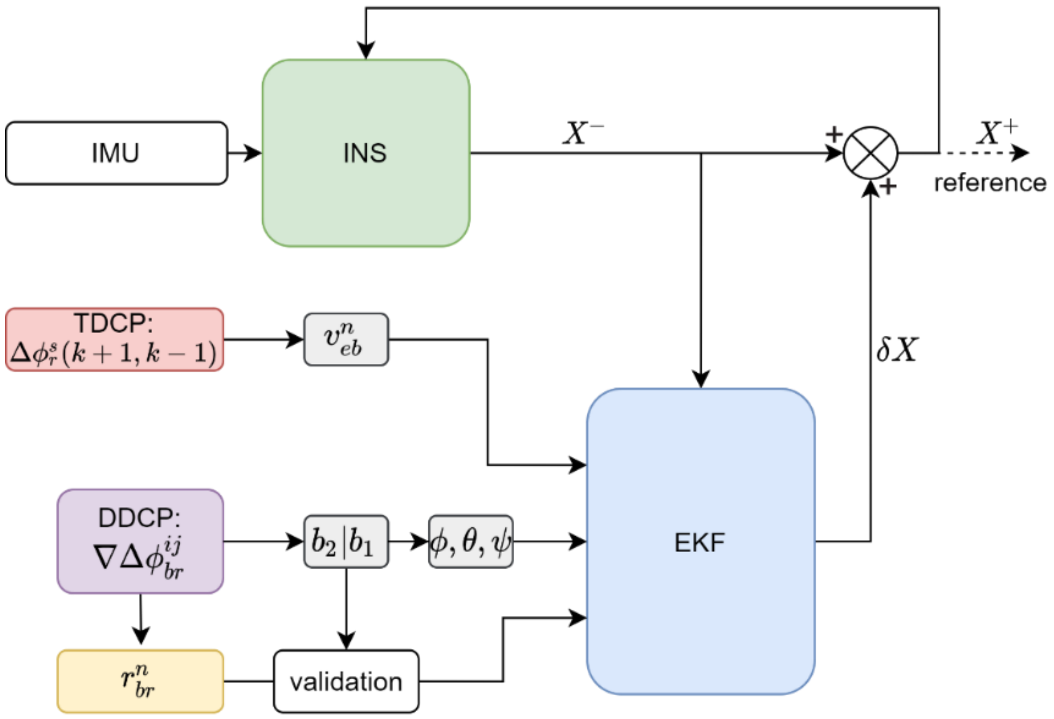


Figure E.3. Reference solution setup.

The architecture is essentially a loosely coupled INS/GNSS integration scheme, similar to the one presented in Section E.1.However, the setup uses quaternions to represent the aircraft's orientation, similar to the setup described in Section E.2. The process models for the navigation states and IMU errors estimated in the filter are the same as the ones described in Section E.2.

The observation vector used in the filter consisted of:

a. Post-processed kinematic (PPK) position: derived using the LeicaGS10 receiver and the NEO-M8T receiver on the aircraft (GM), as described in Section 6.7.2. The standard deviations of the measurement noise are in agreement with values presented in Section 6.7.2.
b. Velocity: derived from time-differenced carrier phase observations (TDCP) (Freda *et al.*, 2015) from the NEO-M8T receiver on the aircraft (GM). The standard deviations of the velocity measurements were computed during static periods (when the aircraft was not moving). The values used (1σ) were 0.065 m/s for the north and east velocity and 0.09 m/s for the down velocity.
c. GNSS attitude: derived from the two baseline solutions b_1 and b_2 as described in Section 6.7.2. The standard deviations of the values used for the measurement covariance matrix in the filter agreed with the standard deviation of the derived yaw angle presented in Section 6.7.2.

The initial states used in the filter were initialised directly from the measurements. Therefore, the standard deviations of the initial state covariance matrix are similar to the measurement uncertainties, as can be seen in Table E.7.

Table E.7. Initial uncertainties for the states.

State	**Standard deviation (σ)**
Position	$[0.1, 0.1, 0.12]$ m
Velocity	$[0.1, 0.1, 0.1]$ m/s
Attitude	$[0.8°, 0.8°, 1.0°]$
Accelerometer bias	[40, 40, 40] mg
Gyroscope bias	[1000, 1000, 1000] deg/h

The standard deviation of the tuned process noise is also shown in Table E.8.

Table E.8. The process noise used in the filter.

State	**Standard deviation (σ)**
Position	10^{-4} m
Velocity	10^{-2} m/s
Attitude	10^{-3} rad
Accelerometer bias	2×10^{-5} m/s^2
Gyroscope bias	2×10^{-6} rad/s

References

Argyle, M. E. and Beard, R. W. (2016) 'Nonlinear Total Energy Control for the Longitudinal dynamics of an aircraft', *Proceedings of the American Control Conference*, 2016–July(4), pp. 6741–6746.

Babu, R. and Wang, J. (2009) 'Ultra-tight GPS/INS/PL integration: A system concept and performance analysis', *GPS Solutions*, 13, pp. 75–82.

Balmer, G. R. (2015) *Modelling and Control of a Fixed-wing UAV for Landings on Mobile Landing Platforms*. KTH Royal Institute of Technology.

Barbour, N. M. (2011) *Inertial Navigation Sensors*. Cambridge, MA 02139, USA.

Barlow, J. B., Rae, W. H. and Pope, A. (1999) *Low-Speed Wind Tunnel Testing*. 3rd edn, *Physics Today*. 3rd edn. New York: John Wiley & Sons. ISBN 0471557749.

Basile, F. *et al.* (2018) 'Multi-frequency precise point positioning using GPS and Galileo data with smoothed ionospheric corrections', *2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings*, pp. 1388–1398.

Basile, F., Moore, T. and Hill, C. (2019) 'Analysis on the potential performance of GPS and galileo precise point positioning using simulated real-time products', *Journal of Navigation*, 72(1), pp. 19–33.

Beard, R. W. and McLain, T. W. (2013) *Small Unmanned Aircraft: Theory and Practice*, *Princeton University Press*. Princeton, New Jersey: Princeton University Press. ISBN 9780691149219 (ISBN).

Black, H. D. and Eisner, A. (1984) 'Correcting Satellite Doppler Data for Tropospheric Effects', *Journal of Geophysical Research*, 89(D2), pp. 2616–2626.

Borre, K. *et al.* (2007) *A software-defined GPS and Galileo receiver: a single-frequency approach*. 1st edn. Birkhäuser. ISBN 9780817637781.

Bosch (2018) *BMP388 - Digital pressure sensor Datasheet*. Available at: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP388-DS001.pdf (Accessed: 27 October 2019).

Brandt, J. B. and Selig, M. S. (2011) 'Propeller Performance Data at Low Reynolds Numbers', in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Orlando, Florida: AIAA.

Brossard, M. (2020) 'Denoising IMU Gyroscopes with Deep Learning for Open-Loop Attitude Estimation', *IEEE Robotics and Automation Letters*, 5(3), pp. 4796–4803.

Brown, R. G. and Hwang, P. Y. . (2012) *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises*. 4th edn. Hoboken, NJ: John Wiley & Sons. ISBN 9780470609699.

Bryson, M. and Sukkarieh, S. (2004) 'Vehicle Model Aided Inertial Navigation for a UAV using Low-cost Sensors', in *Australian Conference on Robotics and Automation*. Canberra: ARAA.

CAA (2016) *Air Navigation Order 2016 | UK Civil Aviation Authority*. https://www.caa.co.uk/News/Air-Navigation-Order-2016/.

Capuano, V. (2016) *GNSS-Based Navigation for Lunar Missions*. École Polytechnique Fédérale de Lausanne.

China Satellite Navigation Office (2017) *BeiDou Navigation Satellite System Signal In Space Interface Control Document*. Available at: http://www.beidou.gov.cn/xt/gfxz/201712/P020171218337008148266.pdf (Accessed: 28 August 2018).

Cork, L. (2014) *Aircraft Dynamic Navigation for Unmanned Aerial Vehicles*. Queensland University of Technology.

Crocoll, P. *et al.* (2013) 'Unified Model Technique for Inertial Navigation', *NAVIGATION*, 60(3), pp. 179–193.

Crocoll, P. and Trommer, G. F. (2014) 'Quadrotor Inertial Navigation Aided by a Vehicle Dynamics Model with In-Flight Parameter Estimation', in *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*. Tampa, Florida, pp. 1784–1795.

Dadkhah, N., Mettler, B. and Gebre-egziabher, D. (2008) 'A Model-Aided AHRS for Micro Aerial Vehicle Application', in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*. Savannah, GA, pp. 545–553.

Ducard, G. (2007) *Fault-tolerant flight control and guidance systems for a small unmanned aerial vehicle*. ETH zurich.

Dunn, M. J. (2018) *Interface Specification IS-GPS-200J*. Available at: https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf (Accessed: 28 August 2018).

El-Diasty, M. and Pagiatakis, S. (2009) 'A Rigorous Temperature-Dependent Stochastic Modelling and Testing for MEMS-Based Inertial Sensor Errors', *Sensors*, 9(11), pp. 8473–8489.

El-Sheimy, N., Hou, H. and Niu, X. (2008) 'Analysis and Modeling of Inertial Sensors Using Allan Variance', *IEEE Transactions on Instrumentation and Measurement*, 57(1), pp. 140–149.

European GNSS Agency (2019) *GSA GNSS Market Report*, *Cleanroom Technology*.

European Union (2016) *European GNSS (Galileo) Open Service Signal-In-Space Interface Control Document*. Available at: https://www.gsc-europa.eu/system/files/galileo_documents/Galileo-OS-SIS-ICD.pdf (Accessed: 28 August 2018).

Falco, G., Pini, M. and Marucco, G. (2017) 'Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios', *Sensors*, 17(2).

Freda, P. *et al.* (2015) 'Time-differenced carrier phases technique for precise GNSS velocity estimation', *GPS Solutions*, 19(2), pp. 335–341.

Gelb, A. *et al.* (1974) *Applied optimal estimation*. Cambridge, MA, London: MIT Press. ISBN 0262200279.

George, M. and Sukkarieh, S. (2005) 'Tightly Coupled INS / GPS with Bias Estimation for UAV Applications', in *Proceedings of the Australiasian Conference on Robotics and Automation (ACRA)*. Sydney, Australia: ARAA.

Giorgi, G. and Teunissen, P. J. G. (2012) 'GNSS Carrier Phase-Based Attitude Determination', in *Recent Advances in Aircraft Technology*.

gliders (2014). http://www.gliders.uk.com/.

Gong, A., Macneill, R. and Verstraete, D. (2018) 'Performance Testing and Modeling of a Brushless DC Motor, Electronic Speed Controller and Propeller for a Small UAV', in *2018 Joint Propulsion Conference*. Cincinnati, Ohio: AIAA.

GPS Directorate (2019) *NAVSTAR GPS Space Segment/Navigation User Segment Interfaces (IS-GPS-200K)*. Available at: https://www.gps.gov/technical/icwg/IS-GPS-200K.pdf.

Groves, P. D. (2008) *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston: Artech House. ISBN 978-1-58053-255-6.

Groves, P. D. (2013) *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd edn. Artech House. ISBN 9781608070053.

Hide, C. (2003) *Integration of GPS and low cost INS measurements*. University of Nottingham.

Hofmann-Wellenhof, B., Lichtenegger, H. and Wasle, E. (2008) *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. Vienna: SpringerWienNewYork. ISBN 978-3-211-73012-6.

Honeywell (2018) *HG9900 Inertial Measurement Unit*. Available at: https://aerospace.honeywell.com/content/dam/aerobt/en/documents/learn/products/sensors/brochures/N61-1638-000-000-hg9900inertialmeasurementunit-bro.pdf.

ICAO (1993) *Manual of the ICAO Standard Atmosphere: extended to 80 kilometres (262,500 feet)*. 3rd edn. Montreal, Quebec: ICAO.

Jiménez López, J. and Mulero-Pázmány, M. (2019) ‘Drones for Conservation in Protected Areas: Present and Future’, *Drones*, 3(1).

Julier, S. J. and Durrant-whyte, H. F. (2003) ‘On The Role of Process Models in Autonomous Land Vehicle Navigation Systems’, *IEEE Transactions on Robotics and Automation*, 19(1), pp. 1–14.

Julier, S. J. and Uhlmann, J. K. (1997) ‘New extension of the Kalman filter to nonlinear systems’, in *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*. SPIE.

Junos, M. H., Mohd Suhadis, N. and Zihad, M. M. (2014) ‘Experimental Determination of the Moment of Inertias of USM e-UAV’, *Applied Mechanics and Materials*, 465–466, pp. 368–372.

Kalman, R. E. (1960) ‘A New Approach to Linear Filtering and Prediction Problems’, *Journal of Basic Engineering*, 82(Series D), pp. 35–45.

Kaplan, E. D. and Hegarty, C. J. (2006) *Understanding GPS: Principles and Applications*. 2nd edn. Norwood, MA: Artech House. ISBN 1580538940.

Kaplan, E. D. and Hegarty, C. J. (2017) *Understanding GPS/GNSS: Principles and Applications*. 3rd edn. Artech House. ISBN 9781630810580.

Khaghani, M. and Skaloud, J. (2016a) ‘Autonomous Vehicle Dynamic Model-Based Navigation for Small UAVs’, *NAVIGATION*, 63(3), pp. 345–358.

Khaghani, M. and Skaloud, J. (2016b) ‘Evaluation of Wind Effects on UAV Autonomous Navigation of Small Uavs Based on Vehicle Dynamic Model’, in *Proceedings of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*. Portland, Oregon: ION, pp. 1432–1440.

Khaghani, M. and Skaloud, J. (2018a) ‘Assessment of VDM-based autonomous navigation of a UAV under operational conditions’, *Robotics and Autonomous Systems*, 106, pp. 152–164.

Khaghani, M. and Skaloud, J. (2018b) ‘VDM-based UAV Attitude Determination in Absence of IMU Data’, in *European Navigation Conference, ENC 2018*. IEEE, pp. 84–90.

Khan, W. and Nahon, M. (2016) ‘Modeling Dynamics of Agile Fixed-Wing UAVs for Real-Time Applications’, in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. Arlington, VA USA, pp. 1303–1312.

Khanafseh, S. *et al.* (2018) ‘GNSS multipath error modeling for automotive applications’, in *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*, pp. 1573–1589.

Kim, J. and Sukkarieh, S. (2003) ‘A Baro-Altimeter Augmented INS/GPS Navigation System for an Uninhabited Aerial Vehicle’, in *The 6th International Symposium on Satellite Navigation Technology Including Mobile*

Positioning & Location Services.

Kim, M. and Kim, J. (2015) 'A Long-Term Analysis of the GPS Broadcast Orbit and Clock Error Variations', *Procedia Engineering*, 99, pp. 654–658.

Klobuchar, J. A. (1987) 'Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users', *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(3), pp. 325–331.

Klöckner, A. (2013) 'Geometry Based Flight Dynamics Modelling of Unmanned Airplanes', in *AIAA Modeling and Simulation Technologies Conference*. Boston, MA: AIAA 2013-5154.

Koifman, M. and Bar-Itzhack, I. Y. (1999) 'Inertial navigation system aided by aircraft dynamics', *IEEE Transactions on Control Systems Technology*, 7(4), pp. 487–493.

Koifman, M. and Merhav, S. J. (1991) 'Autonomously aided strapdown attitude reference system', *Journal of Guidance, Control, and Dynamics*, 14(6), pp. 1164–1172.

Konrad, T. *et al.* (2017) 'State Estimation for a Multirotor using Tight-Coupling of GNSS and Inertial Navigation', *IFAC-PapersOnLine*, 50(1), pp. 11683–11688.

Kuipers, J. B. (1999) 'Quaternions and Rotation Sequences', in *Proceedings of the International Conference on Geometry, Integrability and Quantization*. Varna Bulgaria, pp. 127–143.

LaMance, J., Collins, P. and Langley, R. (1996) 'Limiting Factors in Tropospheric Propagation Delay Error Modelling for GPS Airborne Navigation', in *Proceedings of the 52nd Annual Meeting of The Institute of Navigation*. Cambridge, MA, pp. 519–528.

Lau, T. K., Liu, Y. H. and Lin, K. W. (2013) 'Inertial-based localization for unmanned helicopters against GNSS outage', *IEEE Transactions on Aerospace and Electronic Systems*, 49(3), pp. 1932–1949.

Laupré, G. and Skaloud, J. (2020) 'On the Self-Calibration of Aerodynamic Coefficients in Vehicle Dynamic Model-Based Navigation', *Drones*, 4(3), pp. 1–15.

Lim, C. *et al.* (2019) 'Dynamic performance evaluation of various GNSS receivers and positioning modes with only one flight test', *Electronics*, 8(12).

Madison, R. *et al.* (2007) 'Vision-Aided Navigation for Small UAVs in GPS-Challenged Environments', in *AIAA Infotech at Aerospace 2007 Conference and Exhibit*. California: AIAA 2007-2986.

Martellucci, A. and Prieto-Cerdeira, R. (2009) 'Review of tropospheric, ionospheric and multipath data and models for global navigation satellite systems', in *Proceedings of the 3rd European Conference on Antennas and Propagation, EuCAP 2009*. Berlin: IEEE, pp. 3697–3702.

Matera, E. R. *et al.* (2019) 'Characterization of Line-of-sight and Non-line-of-sight Pseudorange Multipath Errors in Urban Environment for GPS and Galileo', in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*. Reston, Virginia, pp. 177–196.

Microchip Technology (2020a) *ATmega48A/PA/88A/PA/168A/PA/328/P Datasheet*. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf.

Microchip Technology (2020b) *ATmega640/V-1280/V-1281/V-2560/V-2561/V*. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf.

Moorhouse, D. J. and Woodcock, R. J. (1982) *Background Information and User Guide for MIL-F-8785C, Military Specification - Flying Qualities of Piloted Airplanes*.

Mueller, K., Crocoll, P. and Trommer, G. F. (2016) 'Model-Aided Navigation with Wind Estimation for Robust Quadrotor Navigation', in *Proceedings of the 2016 International Technical Meeting of The Institute of Navigation*. Monterey, California, pp. 689–696.

Murphy, T. and Snow, R. (1997) 'GPS Multipath on Large Commercial Air Transport Airframes', *NAVIGATION*, 43(4), pp. 397–406.

Mwenegoha, H. *et al.* (2019) 'Model-Based Autonomous Navigation with Moment of Inertia Estimation for Unmanned Aerial Vehicles', *Sensors*, 19(11).

Mwenegoha, H. A. *et al.* (2019) 'Enhanced Fixed Wing UAV Navigation in Extended GNSS Outages using a Vehicle Dynamics Model and Raw GNSS Observables', in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. Miami, Florida, pp. 2552–2565.

Mwenegoha, H. A. *et al.* (2020) 'A Model-based Tightly Coupled Architecture for Low-Cost Unmanned Aerial Vehicles for Real-Time Applications', *IEEE Access*.

Mwenegoha, H. A. and Jabbal, M. (2013) 'Investigation of Passive Flow Control Techniques to Enhance the Stall Characteristics of a Microlight Aircraft', *International Journal of Flow Control*, 5(3), pp. 215–242.

Niu, X. *et al.* (2014) 'Using Allan variance to analyze the error characteristics of GNSS positioning', *GPS Solutions*, 18(2), pp. 231–242.

Noll, C. E. (2010) 'The crustal dynamics data information system: A resource to support scientific analysis using space geodesy', *Advances in Space Research*, 45(12), pp. 1421–1440.

NovAtel (2011) *OEMStar ™ Installation and Operation User Manual*, *OM-20000126*.

Nusbaum, U. and Klein, I. (2017) ‘Control Theoretic Approach to Gyro-Free Inertial Navigation Systems’, *IEEE Aerospace and Electronic Systems Magazine*, 32(8), pp. 38–45.

NXP Semiconductors (2015) *FXAS21002C: 3-Axis Digital Angular Rate Gyroscope Data sheet*. Available at: https://www.nxp.com/docs/en/data-sheet/FXAS21002.pdf (Accessed: 27 October 2019).

NXP Semiconductors (2017) *FXOS8700CQ: 6-axis sensor with integrated linear accelerometer and magnetometer Data sheet*. Available at: https://www.nxp.com/files-static/sensors/doc/data_sheet/FXOS8700CQ.pdf (Accessed: 27 October 2019).

Papadimitratos, P. and Jovanovic, A. (2008) ‘Protection and Fundamental Vulnerability of GNSS’, in *2008 IEEE International Workshop on Satellite and Space Communications*. IEEE, pp. 167–171.

Park, S., Deyst, J. and How, J. (2004) ‘A New Nonlinear Guidance Logic for Trajectory Tracking’, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Rhode Island: AIAA 2004-4900.

Penna, N., Dodson, A. and Chen, W. (2001) ‘Assessment of EGNOS Tropospheric Correction Model’, *Journal of Navigation*, 54(1), pp. 37–55.

Pinchin, J. (2011) *GNSS Based Attitude Determination for Small Unmanned Aerial Vehicles*. University of Canterbury.

Pittelkau, M. E. (2003) ‘Rotation Vector in Attitude Estimation’, *Journal of Guidance, Control, and Dynamics*, 26(6), pp. 855–860.

Quinchia, A. G. *et al.* (2013) ‘A Comparison between Different Error Modeling of MEMS Applied to GPS/INS Integrated Systems’, *Sensors*, 13(8), pp. 9549–9588.

Ramprasadh, C. and Arya, H. (2011) ‘A monte carlo analysis of sensor fusion algorithm for the estimation of aerodynamic angles in a mini aerial vehicle’, *International Journal of Micro Air Vehicles*, 3(1), pp. 35–47.

Richardson, T., Hill, C. and Moore, T. (2016) ‘Analysis of Multi - Constellation GNSS Signal Quality’, in *Proceedings of the 2016 International Technical Meeting of The Institute of Navigation*. Monterey, California: ION, pp. 631–638.

Russian Space Systems, J. (2016) *GLONASS Interface Control Document: General Description of Code Division Multiple Access Signal System*. Available at: http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD-GLONASS-CDMA-General.-Edition-1.0-2016.pdf (Accessed: 28 August 2018).

Saeed, A. S. *et al.* (2015) ‘A review on the platform design, dynamic modeling and control of hybrid UAVs’, in *International Conference on Unmanned Aircraft Systems (ICUAS)*. Colorado, USA: IEEE, pp. 806–815.

Sendobry, A. (2011) 'A Model Based Navigation Architecture for Small Unmanned Aerial Vehicles', in *Proceedings of the European Navigation Conference (ENC)*. London, United Kingdom: Royal Institute of Navigation (RIN).

Sendobry, A. (2014) *Control System Theoretic Approach to Model Based Navigation*. Technische Universität Darmstadt.

Simsky, A. *et al.* (2008) 'Experimental Results for the Multipath Performance of Galileo Signals Transmitted by GIOVE-A Satellite', *International Journal of Navigation and Observation*.

Solà, J. (2017) *Quaternion Kinematics for the error-state Kalman filter*. arXiv:1711.02508 [cs.RO].

Squire, W. and Trapp, G. (1998) 'Using Complex Variables to Estimate Derivatives of Real Functions', *SIAM Review*, 40(1), pp. 110–112.

Takasu, T. and Yasuda, A. (2013) *RTKLIB ver. 2.4.2 Manual*. Available at: http://www.rtklib.com/rtklib.htm.

Tawk, Y. (2013) *Advanced Algorithmic and Architecture Designs for Future Satellite Navigation Receivers*. École Polytechnique Fédérale de Lausanne.

Tawk, Y. *et al.* (2014) 'Implementation and Performance of a GPS/INS Tightly Coupled Assisted PLL Architecture Using MEMS Inertial Sensors', *Sensors*, 14(2), pp. 3768–3796.

Teunissen, P. J. G. (1994) 'A New Method for Fast Carrier Phase Ambiguity Estimation', in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS'94*. Las Vegas, NV, USA: IEEE, pp. 562–573.

Titterton, D. H. and Weston, J. L. (2004) *Strapdown Inertial Navigation Technology*. 2nd edn. Stevenage, UK: IEE. ISBN 0 86341 358 7.

u-blox (2019) *GNSS antennas: RF design considerations for u-blox GNSS receivers*. Available at: www.u-blox.com.

u-blox (2020) *NEO/LEA-M8T: u-blox M8 concurrent GNSS timing modules Datasheet*. Available at: www.u-blox.com.

Vasconcelos, J. F. *et al.* (2010) 'Embedded UAV model and LASER aiding techniques for inertial navigation systems', *Control Engineering Practice*, 18(3), pp. 262–278.

Vissière, D. *et al.* (2008) 'Experimental autonomous flight of a small-scaled helicopter using accurate dynamics model and low-cost sensors', *IFAC Proceedings Volumes*, 41(2), pp. 14642–14650.

Wan, E. A. and Merwe, R. van der (2000) 'The Unscented Kalman Filter for Nonlinear Estimation', in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, Alberta, Canada: IEEE, pp. 153–158.

Wang, J. *et al.* (2004) 'Integration of GPS/INS/Vision Sensors to Navigate Unmanned Aerial Vehicles', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII(Part B1), pp. 963–970.

Wilde, W. De *et al.* (2016) 'GNSS Interference in Unmanned Aerial Systems', in *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*. Portland, Oregon, pp. 1465–1476.

Xsens (2018) *MTi 100-series Datasheet*. Available at: www.xsens.com.

Youn, W. *et al.* (2020) 'Model-Aided State Estimation of HALE UAV With Synthetic AOA / SSA for Analytical Redundancy', *IEEE Sensors Journal*, 20(14), pp. 7929–7940.

Yu, S. *et al.* (2016) 'Technical Analysis of VTOL UAV', *Journal of Computer and Communications*, 4(15), pp. 92–97.

Zahran, S. *et al.* (2018) 'Hybrid Machine Learning VDM for UAVs in GNSS-denied Environment', *NAVIGATION*, 65(3), pp. 477–492.

Zanetti, R., DeMars, K. J. and Bishop, R. H. (2009) 'On Underweighting Nonlinear Measurements', *Advances in the Astronautical Sciences*, 135(2), pp. 1085–1096.