

Sensor Fusion INS/GNSS based on Fuzzy Logic Adaptive Error-State Kalman Filter and Unscented Kalman Filter

by

Gabriel Giannini de Cunto

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Aerospace Engineering

Carleton University
Ottawa, Ontario

© 2020

Gabriel Giannini de Cunto

Abstract

In this work, a Fuzzy Logic Adaptive Control (FLAC) is used to correct an Error-State Kalman Filter (ESKF) and an Unscented Kalman Filter (UKF) in a loosely coupled INS/GNSS system. The FLAC is used to prevent the Kalman Filter (KF) to diverge or to reach to a high bound solution when the Inertial Measurement Unit (IMU) presents a dominant $1/f$ flicker noise. First, the ESKF and UKF implementation were tuned to achieve the optimal solution when the IMU has only white noise. Secondly, a $1/f$ flicker noise was applied to the IMU, making both Kalman Filters implementation achieve a suboptimal solution. And thirdly, a FLAC was used to correct both ESKF and UKF when coloured noise is present. The results evidence the influence of coloured noise in the system, which makes both Kalman Filter implementations reach to a large error bound solution. After analyzing the Kalman Filter behaviour with coloured noise, a novel FLAC methodology was defined. The FLAC combines the observation of both the residuals and the states error covariance and apply the correction using the exponential weighted parameter when the error covariance presents a higher than expected value, and a process noise injection when the residuals are broader than expected. The application of the proposed FLAC methodology figures out as the best solution to deal with the coloured noise, leading to a final solution that improves the navigation accuracy for all the states, preserving the stability of the error covariance matrix. Finally, the results for ESKF are compared against the results for the UKF. It was showed that, although both Kalman filter implementations bring equivalent outcomes, the UKF is slightly less sensitive to disturbances.

Acknowledgements

I would like first to thank my supervisor, Professor Jurek Sasiadek. He allowed me to explore and research an exciting and important topic, giving guidance where it was needed.

I also want to give thanks to Rear Admiral Ferreira Marques and Captain Rodney for their support and solicitude with me.

I also want to give thanks to God, to my family, to my wife Larisse, my son Rafael and my little Laura. They all helped to motivate me. They gave me support and strength when I needed it.

Table of Contents

Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	ix
List of Tables	xvi
1. Introduction	17
1.1. Motivation	17
1.2. Problem Statement	18
1.3. Previous Work.....	19
1.4. Thesis Objectives	23
1.5. Contributions.....	24
1.6. Organization.....	24
2. Background Information.....	26
2.1. Coordinate frame and Earth Model.....	26
2.1.1. Body Frame.....	27
2.1.2. Local Navigation Frame	27
2.1.3. Earth-Centered Earth-Fixed (ECEF) Frame	28
2.1.4. Earth and Gravity Models.....	29
2.1.4.1. Earth Ellipsoid Model	29

2.1.4.2.	Earth Rotation	30
2.1.4.3.	Centrifugal and Coriolis forces	30
2.1.4.4.	Gravitation and Gravity.....	31
2.1.5.	Frame Transformations	32
2.1.5.1.	Transformation from Body to Local NED coordinate frame	32
2.1.5.2.	Transformation from Local NED to ECEF coordinate frame.....	33
2.1.6.	Quaternion.....	35
2.2.	Inertial Navigation.....	36
2.2.1.	The INS dynamic model	37
2.2.2.	Gyroscope and accelerometer raw data obtained by inverse kinematics....	39
2.2.3.	Error sources	40
2.2.3.1.	IMU Noise Model	43
2.2.3.2.	IMU with Coloured noise.....	44
2.3.	GNSS.....	45
2.3.1.	GNSS principles.....	45
2.3.2.	Error sources	47
2.4.	INS/GNSS Integration model.....	48
2.5.	Kalman Filter.....	49
2.5.1.	The Error-State model.....	50
2.5.2.	Error-State Kalman Filter model	53

2.5.3.	Unscented Kalman Filter model	56
2.6.	Kalman Filter Divergence	59
2.6.1.	Exponential Weighted Kalman Filter	61
2.6.1.1.	Exponential Weighted KF	61
2.6.1.2.	Exponential Weighted UKF	62
2.6.2.	Fictitious Process Noise Injection.....	63
2.7.	Fuzzy Adaptive Kalman Filter	64
2.7.1.	Fuzzy Logic	64
2.7.2.	Adaptive Kalman Filter	65
2.7.3.	Fuzzy Adaptive Kalman Filter Architecture	66
2.7.3.1.	Criteria for chosen the FL input.....	66
2.7.3.2.	Membership Functions.....	68
2.7.3.3.	Criteria for base rules definition	69
2.7.3.4.	Criteria for apply the FL output	69
3.	Methodology.....	71
3.1.	INS Model	72
3.1.1.	Navigation profile generation	73
3.1.2.	Gyroscope and accelerometer raw data generation	76
3.1.3.	INS dynamic model	77
3.2.	INS/GNSS integration.....	77

3.2.1.	GNSS raw data generation.....	77
3.2.2.	IMU Noise Model.....	78
3.2.3.	Loosely Coupled INS/GNSS system	79
3.2.3.1.	System initialization.....	80
3.2.3.2.	System and Measurement Covariance Matrices	82
3.3.	Fuzzy Logic.....	83
3.3.1.	Fuzzy Logic Implementation	84
4.	Simulation Results and Discussions	88
4.1.	INS Simulation.....	88
4.2.	Noisy IMU and GNSS Data.....	91
4.3.	ESKF based method.....	95
4.3.1.	White and Coloured Noise ESKF Solution	95
4.3.2.	White and Coloured Noise ESKF states	99
4.4.	UKF Solution	105
4.4.1.	White and Coloured Noise UKF Solution	106
4.4.2.	White and Coloured Noise UKF states.....	109
4.5.	Fuzzy Logic.....	114
4.5.1.	Fuzzy Logic behavior	115
4.5.2.	Fuzzy Logic Adaptive ESKF	116
4.5.3.	Fuzzy Logic Adaptive UKF.....	125

4.6.	ESKF and UKF comparison.....	130
4.6.1.	System with white noise	130
4.6.2.	System with coloured noise	133
4.6.3.	System with coloured noise and FL correction.....	136
4.7.	Thesis Limitations	140
5.	Conclusions	142
6.	References	145

Appendices

Appendix A	- Exponential Weighted UKF derivation	149
Appendix B	- Study on the exponential weighted parameter	151
Appendix C	- Study on the Fuzzy Logic approach.....	154
Appendix C.1	Identifying the Fuzzi Logic Inputs	154
Appendix C.2	Appling the Corrector Factor.....	157
Appendix D	- ESKF and UKF comparison	163
Appendix D.1	White Noise	164
Appendix D.1.1	True-ground error.....	164
Appendix D.1.2	Error Covariance	166
Appendix D.1.3	Residuals	168
Appendix D.2	Coloured Noise	170
Appendix D.2.1	True ground error	170

Appendix D.2.2	Error Covariance	172
Appendix D.2.3	Residuals	174
Appendix D.3	Fuzzy Logic	176
Appendix D.3.1	True grund error:.....	176
Appendix D.3.2	Error Covariance	178
Appendix D.3.3	Residuals	180
Appendix E	- Motion and IMU raw data generation MATLAB Code	182
Appendix E.1	Navigation profile generation.....	182
Appendix E.2	NED to ECEF transformation	184
Appendix E.3	Accelerometer and Gyroscope Raw data generation	185
Appendix F	- INS, ESKF, and UKF MATLAB Code	187
Appendix F.1	INS model.....	187
Appendix F.2	ESKF model	190
Appendix F.3	UKF model	194
Appendix G	- GNSS/INS integration Simulink model.....	200
Appendix G.1	IMU noise model	200
Appendix G.2	INS/GNSS integration model	201
Appendix G.3	System Initialization	202
Appendix G.4	Fuzzy Logic Implementation	203

List of Figures

Figure 2-1 - Comparison between body (red) and local (blue) frames.....	28
Figure 2-2 - Comparison between local (yellow) and ECEF (blue) frames	29
Figure 2-3 - Basic diagram of an INS.....	37
Figure 2-4 - INS dynamic model	38
Figure 2-5 - Biases and scale factors instability for accelerometers considering certain applications and technologies used in the manufacture of sensors [39].	42
Figure 2-6 - Biases and scale factors instability for gyroscopes considering certain applications and technologies used in the manufacture of sensors [39].	42
Figure 2-7 – Gyroscope simulated with white noise (left) and with coloured noise (right)	45
Figure 2-8 - Loosely coupled INS/GNSS system using KF	49
Figure 2-9 - Optimal, Apparent divergence, and True divergence for error covariance ..	60
Figure 2-10 - Different shapes of Membership Function: Gaussian (left), Triangular (right), and Trapezoidal (down)	68
Figure 3-1 - General view of the proposed methodology	72
Figure 3-2 - General diagram for the INS Model	73
Figure 3-3 - Position, velocity and attitude profiles in NED frame for the first navigation profile.....	74
Figure 3-4 - Position, velocity, and attitude profiles in NED frame for the second navigation profile.....	75
Figure 3-5 - Loosely coupled INS/GNSS system with the use of Error Feedback Extended Kalman Filter	80
Figure 3-6 - FLAS applied in the Loosely Coupled GNSS/INS System.....	85
Figure 4-1 - Error between the ground-true and the INS solution for the navigation with no force applied.....	89

Figure 4-2 - Error between the ground-true navigation and the INS navigation solution for the realistic navigation generated.....	90
Figure 4-3 - Noise-free (left) and noise (right) y-axis accelerometer data	91
Figure 4-4 - Noise-free (left) and noisy (right) y-axis gyroscope data	91
Figure 4-5 - White and coloured noise variance over the time for the gyroscope considering the firsts 20s (left) and the full motion of 3600s (right)	92
Figure 4-6 – Attitude, Position, and Velocity errors for the INS-alone solution for a noisy IMU.....	93
Figure 4-7 - Position and Velocity errors for the GNSS-alone solution for a noise GNSS	94
Figure 4-8 - Attitude error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise	96
Figure 4-9 - Position error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise	96
Figure 4-10 - Velocity error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise	96
Figure 4-11 – Gyroscope bias correction given by the ESKF with white (left) and coloured (right) noise.....	98
Figure 4-12 - Accelerometer bias correction given by the ESKF with white (left) and coloured (right) noise.....	98
Figure 4-13 – Variance for Attitude, Position, and Velocity states considering the ESKF with white (left) and coloured (right) noise	100
Figure 4-14 - Variance for the gyroscope and accelerometer bias states considering the ESKF with white (left) and coloured (right) noise	101
Figure 4-15 – KF residuals for Attitude, Position, and Velocity considering the ESKF with white (left) and coloured (right) noise	102
Figure 4-16 - KF residuals for gyroscope and accelerometer bias considering the ESKF with white (left) and coloured (right) noise	103

Figure 4-17 – Kalman Gain for Position and Velocity considering the ESKF with white (left) and coloured (right) noise	104
Figure 4-18 - Attitude error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise	106
Figure 4-19 - Position error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise	106
Figure 4-20 - Velocity error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise	107
Figure 4-21 – Gyroscope bias correction given by the UKF with white (left) and coloured (right) noise.....	108
Figure 4-22 - Accelerometer bias correction given by the UKF with white (left) and coloured (right) noise.....	108
Figure 4-23 – Variance for Attitude, Position, and Velocity states considering the UKF with white (left) and coloured (right) noise	110
Figure 4-24 - Variance for the gyroscope and accelerometer bias state considering the UKF with white (left) and coloured (right) noise	111
Figure 4-25 – KF residuals for Attitude, Position, and Velocity considering the UKF with white (left) and coloured (right) noise	112
Figure 4-26 - KF residuals for gyroscope and accelerometer bias considering the UKF with white (left) and coloured (right) noise	113
Figure 4-27 - Kalman Gain for Position and Velocity considering the UKF with white (left) and coloured (right) noise	114
Figure 4-28 - Fuzzy Logic output α and β for position and velocity states	115
Figure 4-29 – Position error using ESKF without FL correction (left) and with FL correction (right).....	116
Figure 4-30 – Velocity error using ESKF without FL correction (left) and with FL correction (right).....	117
Figure 4-31 – Attitude error using ESKF without FL correction (left) and with FL correction (right).....	117

Figure 4-32 – Position error using ESKF without FL correction (left) and with FL correction (right)	118
Figure 4-33 – Velocity error using ESKF without FL correction (left) and with FL correction (right)	119
Figure 4-34 – Attitude error using ESKF without FL correction (left) and with FL correction (right)	119
Figure 4-35 – Variance for the attitude, position, and velocity states considering the system without FL (left) and with FL (right) correction.....	120
Figure 4-36 – ESKF residuals for Attitude, Position, and Velocity considering the ESKF without FL correction (left) and with FL correction (right)	122
Figure 4-37 - ESKF residuals for gyroscope and accelerometer bias considering the ESKF without FL correction (left) and with FL correction (right)	123
Figure 4-38 – Kalman Gain for Position and Velocity considering the ESKF without FL correction (left) and with FL correction (right)	124
Figure 4-39 –Position error using UKF without FL correction (left) and with FL correction (right)	125
Figure 4-40 – Velocity error using UKF without FL correction (left) and with FL correction (right)	126
Figure 4-41 – Attitude error using UKF without FL correction (left) and with FL correction (right)	126
Figure 4-42 – Variance for the attitude, position, and velocity states considering the system without FL (left) and with FL (right) correction.....	127
Figure 4-43 – UKF residuals for Attitude, Position, and Velocity considering the UKF without FL correction (left) and with FL correction (right)	128
Figure 4-44 – Kalman Gain for Position and Velocity considering the UKF without FL correction (left) and with FL correction (right)	129
Figure 4-45 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude	131
Figure 4-46 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and U velocity (right).....	131

Figure 4-47 - Accelerometer bias (left) and Gyroscope bias (right) determination	131
Figure 4-48 – ESKF and UKF variance (left) and the residuals (right) for the pitch error state	132
Figure 4-49 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude	133
Figure 4-50 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and zoom in the peak occurred around 2300 seconds (right)	134
Figure 4-51 - Error between the true-ground and the INS/GNSS integration for the U velocity (left) and zoom in the peak occurred around 2300 seconds (right)	134
Figure 4-52 - Accelerometer bias (left) and Gyroscope bias (right) determination	134
Figure 4-53 – ESKF and UKF variance (left) and the residuals (right) for the pitch error state	135
Figure 4-54 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude	136
Figure 4-55 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and zoom in the peaks occurred around 2300 seconds (right).....	137
Figure 4-56 - Error between the true-ground and the INS/GNSS integration for the U velocity (left) and zoom in the peaks occurred around 2700 seconds (right).....	137
Figure 4-57 - Accelerometer bias (left) and Gyroscope bias (right) determination	137
Figure 4-58 – ESKF and UKF error variance (left) and the residuals (right) for the pitch error state	139
Figure B 1- KF without correction, with FL weighted single alpha correction, and with FL weighted matrix alpha correction.....	153
Figure C 1 - Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the Attitude pitch	154
Figure C 2 - - Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the position Z	155
Figure C 3- Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the Velocity U.....	155

Figure D 1 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)	164
Figure D 2 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)	165
Figure D 3 - Attitude, position, and velocity error covariance using ESKF (left) and UKF (right)	166
Figure D 4 – Gyroscope and accelerometer bias error covariance using ESKF (left) and UKF (right)	167
Figure D 5 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)	168
Figure D 6 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)	169
Figure D 7 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)	170
Figure D 8 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)	171
Figure D 9 - Attitude, position, and velocity error covariance using ESKF (left) and UKF (right)	172
Figure D 10 – Gyroscope and accelerometer bias error covariance using ESKF (left) and UKF (right)	173
Figure D 11 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)	174
Figure D 12 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)	175
Figure D 13 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)	176
Figure D 14 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)	177
Figure D 15 - Attitude, position, and velocity error covariance using ESKF (left) and UKF (right)	178

Figure D 16 – Gyroscope and accelerometer bias error covariance using ESKF (left) and UKF (right) 179

Figure D 17 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)..... 180

Figure D 18 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)..... 181

List of Tables

Table 1 – List of Fuzzy Logic Adaptive KF approach	22
Table 2 - Unscented Transformation Sample Weights	56
Table 3 – Tactical-grade gyroscope error	78
Table 4 - Tactical-grade accelerometer error	78
Table 5 – Membership functions for input variables	86
Table 6 – Membership functions for output variables	87
Table C 1 - Results for applying a constant exponential weighted parameter Alpha.....	158
Table C 2 - Results for applying a constant Beta value	160

1. Introduction

1.1. Motivation

Inertial navigation systems (INS) are used to indicate the real position, velocity, and attitude of a body performing three-dimensional navigation. The INS operates continuously, showing low short-term noise, being invulnerable to jamming and interference [1]. However, INS will suffer degradation in long-term navigation as the errors presented in the sensor are integrated through the navigation equations.

On the other hand, the Global Navigation Satellite System (GNSS) provides reasonable accuracy for long-term navigation. However, the short-term errors are high, the signals are subject to obstruction and interference, and the standard GNSS does not provide attitude.

The advantages and disadvantages of INS and GNSS are complementary. Therefore, the fusion of these technologies can benefit both, providing a navigation solution with high accuracy in long- and short-term [1]. Usually, this fusion is done by applying one of the many different types of Kalman Filter (KF) methods. The KF uses the stochastic model to estimate, correct, and compensate for errors in the INS model, using the navigation solution and the errors from the INS and GNSS sensors.

Two main assumptions for Kalman filter implementation are that: i) the system is a linear dynamic system and ii) that all noise sources are white. However, in practice, this rarely occurs [2].

For the first assumption, Extended Kalman Filter (EKF), Error-State Kalman Filter (ESKF), and later Unscented Kalman Filter (UKF), have been extensively and successfully used for addressing non-linear systems.

For the second assumption, in some Inertial Measurement Units (IMUs) the noise may change during the time, showing a coloured characteristic, which leads the KF to diverge or to converge to a high bound [3]. This process is usually characterized by a $1/f$ power spectral density - PSD that can be called $1/f$ flicker noise [4]–[6]. To overcome this situation, widened noise or additional states must be considered to overbound the real noise system. However, this can result in accuracy loss and in need of modelling additional states in the KF, which can be a demanding task [1].

An alternative is the use of Adaptive KF approaches in which the assumed process and measurement noise covariance may vary according to the measurement innovations.

J. Z. Sasiadek et al.[3], [7] propose a Fuzzy Adaptive Extended Kalman Filter (FAEKF) for adapting the process and measurement noise covariance matrices, using an exponential data weighting controlled by the FL to adjust the EKF.

Other FL implementation approaches were used by [8], [9], [10], [11] and [12], where a correction factor is added or multiplied to the process noise covariance and/or measurement noise covariance, in which the FL defines the factor value.

Therefore, it is pertinent to study the application of a Fuzzy Logic adaptive Kalman Filter to deal with a situation where a $1/f$ flicker noise is dominant in the IMU.

1.2. Problem Statement

In some IMUs the noise may change during the time, showing a coloured characteristic. This situation leads the KF to reach a sub-optimal solution [3]. To overcome this situation, a Fuzzy Logic Adaptive Kalman Filter is used in an Error Feedback ESKF and UKF systems, applied in a loosely coupled INS/GNSS navigation solution. The objective is to

deal with the KF sub-optimal solution in a situation where a 1/f flicker noise is dominant in the IMU.

1.3. Previous Work

Different types of adaptive KF have been developed and used since the advent of the Kalman Filter in 1960. Usually, they intend to facing the problem where the system noise covariance matrix, Q , and measurement noise covariance matrix, R , can not be determined during the development phase, or to compensate situations where these covariance matrices vary over the time.

Two main adaptive KF approaches have been successfully used for navigation propose [1], the multiple model adaptive estimation (MMAE)[13] and the innovation adaptive estimation (IAE) [14]. Both share the same concept of utilising new statistical information obtained from the innovation to update the system and measurement noise covariance matrix in a systematic approach.

Further classics adaptive approaches comprise Maximum Likelihood Estimation (MLE), Bayesian Estimation, Correlation Method, and Covariance-Matching [14]. The MLE is a technique that defines a set of parameters that influence a likelihood function based on the KF measurements and estimates states; the goal is to maximize the observed likelihood function, making adaptations in these parameters. The Bayesian Estimation is based on obtaining recursive equations for the *a posteriori* probability density of the state and covariance matrices, using conditional probabilities and integration. The Correlation Method estimates the covariance matrices, based on the autocorrelation function of the output or the innovation sequence. Finally, the Covariance-Matching approach aimed to

update the KF in a way that the theoretical covariance match the observed innovations covariance, achieving the optimal KF performance.

Other adaptive approaches include Sage–Husa Adaptive filtering [15], [16], Adaptive Complementary Fusion Filter [17], and approaches based on the Interacting Multiple Model (IMM) [18].

Based on the adaptive techniques described, a countless number of different adaptive Kalman filter have been developed. Some of them are the Robust Adaptive Kalman Filter [19] and the Adaptive Sparse Interpolation Lossless Complementary Filter [20] both based on the adaptive complementary approach; the Centralized Fusion Based on IMM and Adaptive KF [21] and the IMM strong tracking square root cubature KF [22], based on the IMM; the Variational Bayesian adaptive Kalman filter [23],[24]; the maximum likelihood estimation adaptive extended Kalman filter (MLE-AEKF) [25]; and the Weighted Robust Sage-Husa Adaptive Kalman Filter [26].

With the advent of artificial intelligence (AI), Neural Networks (NN) [27]–[29], Fuzzy Logic (FL) [3], [30], and Genetic Algorithms (GA)[31][32] techniques have been used to adapt the covariance matrices to the real noise situation in a more precise and efficient way. The Fuzzy Logic highlight as a widely used technique applied in adaptive controls to deal with nonlinear systems with uncertainties, where the errors can be compensated using a heuristic knowledge of the system. Therefore, no mathematical model of the noise is needed.

Fraser [33] showed that the MLE and FL adaptive KF approaches achieved comparable performance, with the FL presenting half of processing time needed by the MLE and also been a more flexible implementation technique.

The use of a Fuzzy Adaptive Extended Kalman Filter (FAEKF), was initially proposed by J. Z. Sasiadek et al. [3], [7] in 2000, where an exponential data weight controlled by the FL is used to adapt the EKF. The exponential data weighting is a method that prevents the Kalman gain to go to zero with time. If the Kalman gain reaches values near zero, the KF ignores new measurements. Therefore, if the process noise changes during time, the filter will not be able to compensate for it.

In 2003, J. Z. Sasiadek and Q. Wang [34], tested the FAEKF in a system where non-white noise was added to the GPS measurements, showing the possibility to use the FAEKF to deal with coloured noise.

Using the same weighting technique, S. Yazdkhasti, J. Z. Sasiadek and S. Ulrich [30] applied the Fuzzy logic in a UKF and compared it against the FAEKF, finding better results when UKF was used.

Different methodologies to correct and improve the Kalman filter using Fuzzy logic have been proposed by [8], [9], [10], [11] and [12]. All these works use the fuzzy logic to correct the system noise covariance matrix Q , and/or the measurement noise covariance matrix R , by applying a multiplication or addition factor to the Q and/or R matrix. The fuzzy logic determines the value of this factor. The main difference between them lies in the choice of Fuzzy logic inputs - how the fuzzy logic observes the system.

Table 1 shows a comparison between different studies using Fuzzy Logic to adapt the Kalman Filter used in navigation systems. In this table, the parameter α is the FL output.

Table 1 – List of Fuzzy Logic Adaptive KF approach

Publication	Year	KF type used	FL input	How the FL output is applied
[3]	2000	EKF	Error covariance matrix and residuals	$P_k^{\alpha^-} = P_k^- \alpha^{2k}$ $R_k = R \alpha^{-2(k+1)}$ $Q_k = Q \alpha^{-2(k+1)}$
[8]	2001	KF	Difference between the theoretical and actual covariance of the residuals	$R_k = R + \alpha,$ and/or $Q_k = Q + \alpha$
[34]	2003	EKF	Error covariance matrix and residuals	$P_k^{\alpha^-} = P_k^- \alpha^{2k}$ $R_k = R \alpha^{-2(k+1)}$ $Q_k = Q \alpha^{-2(k+1)}$
[9]	2004	KF and EKF	Difference between the theoretical and actual covariance of the residuals	$R_k = R + \alpha,$ and/or $Q_k = Q + \alpha$
[10]	2006	KF	covariance and the mean value of the residual	$R_k = \alpha R,$ $Q_k = \alpha Q$
[35]	2013	UKF	Degree of divergence of the residual	$\hat{P}_k^- = \alpha \hat{P}_k^-$ $P_{zz} = \alpha P_{zz}$ $P_{xz} = \alpha P_{xz}$
[12]	2016	EKF	Bias and oscillation metrics	$R_k = R + \alpha,$ and/or $Q_k = Q + \alpha$
[11]	2016	cubature KF	Degree of divergence of the residual	$Q_k = \alpha Q$
[30]	2016	UKF	Error covariance matrix and residuals	$P_k^{\alpha^-} = P_k^- \alpha^{2k}$ $R_k = R \alpha^{-2(k+1)}$ $Q_k = Q \alpha^{-2(k+1)}$

From Table 1, it is possible to observe that all approaches, except the referenced by [12], use the error covariance and/or the residuals obtained in the KF, as FL inputs. For the methodology used to apply the FL in the KF, it is possible to see that three different methods are used.

The first one is to apply an exponential weighting parameter, as described by [36], where the FL will define the parameter. The second one is to add a correction factor to the Q and/or R, in which the FL defines the factor. And the thirty-one is to multiply a corrector factor to the Q and/or R. The only exception is the methodology used by [35], in which the corrector factor is applied directly to the error covariance matrix.

Although the different approaches, all publications presented in Table 1 reported that the use of Fuzzy logic enhanced the final solution of the KF. Therefore, it can be seen that FAKF methods are indeed capable of improving the navigation accuracy over the traditional KF.

1.4. Thesis Objectives

This work seeks to develop a methodology to deal with a situation where a coupled INS/GNSS navigation solution is presenting sub-optimal behaviour due to the presence of coloured noise in the IMU. The proposal is to use a Fuzzy Logic adaptive Kalman Filter to achieve a better final solution. The implementation is tested in an Error Feedback ESKF and UKF systems, applied in a loosely coupled INS/GNSS navigation solution. The following sub-objectives are foreseen:

- a) Compare the INS/GNSS integration solution with the INS-alone and GNSS-alone solution. Show the better accuracy in the final navigation solution when KF is used.
- b) Tune the KF to achieve the optimal solution when only white noise is present in the system. Reach the best solution possible for the KF implementation.
- c) Make the KF system became suboptimal by applying a coloured noise in the IMU. Show the influence of the non-white noise in an optimal KF.

- d) Analyze the Kalman Filter behaviour with coloured noise and the influence of different correction approach. Define a FLAC application methodology to achieve the best solution for the sub-optimal KF.
- e) Implement the FLAC in the KF system. Correct the KF solution by the use of FLAC.
- f) Analyze the ESKF and UKF behaviour under optimal solution, sub-optimal solution, and FLAC correction. Show the benefits of the FLAC application and the difference between ESKF and UKF.

1.5. Contributions

The contributions of this work to the field of positioning and navigation are as follows:

- Contribute to understanding the effect of non-white noises in a KF system used for navigation solution.
- Introduction to a novel FLAC application methodology for an adaptive KF.
- Contribute to understanding the difference between ESKF and UKF.

1.6. Organization

The remainder of this work is organized into the following parts:

- Part 2: Background on coordinate systems, INS, GNSS positioning, INS/GNSS integration, Kalman Filter, exponential data weighting, fuzzy logic and adaptive KF.
- Part 3: Methodology used for the INS dynamic model, INS/GNSS integration, error and noise model, system initialization, system and measurement covariance matrices, and fuzzy logic for the weighted KF.

- Part 4: Simulation results for the INS model, noisy IMU and GNSS measurements, ESKF solution, UKF solution, Fuzzy logic weighted ESKF, Fuzzy logic weighted UKF, and thesis limitations.
- Part 5: Concludes the research, discusses the results and recommends future work for the expansion of the concept.

2. Background Information

This chapter provides the foundations of navigation and sensor fusion used in this thesis. Section 2.1 describes the coordinate frames used, show the applied model for the Earth's: surface, rotation, and gravity, and presents the equations to perform transformations between different coordinate frames. Section 2.2 focuses on the Inertial Navigation, introduces the main concepts of inertial navigation, shows the dynamic navigation model, the inverse kinematics to obtain the sensor data, the error sources, and the IMU noise model. Section 2.3 describes the GNSS principles, the system architecture, the positioning method, the error sources, and the GNSS noise model. Section 2.4 provides information about the Error-State Kalman Filter and the Unscented Kalman Filter. Section 2.5 introduces the INS/GNSS integration, describes the loosely coupled INS/GNSS system, the integration model, and the system initialization problem. Section 2.6 describes the KF divergence problem, the exponential data weighted and the fictitious process noise injection, showing the equations for the ESKF and UKF. Finally, Section 2.7 introduces the Fuzzy logic and shows an overview of the architecture and fuzzification-defuzzification concepts.

2.1. Coordinate frame and Earth Model

The navigation of an object is described by the determination of its position, orientation, and motion. The navigation must be defined in relation to some reference, which is characterised by an origin and a set of axes. The origin is chosen based on the application, and the sets of axes are chosen to be an orthogonal right-handed basis. Therefore, the navigation determination involves at least two coordinate frames - the object frame and a reference frame.

When inertial sensors are determining the navigation of an object, three frames are usually used: the body frame which is attached to the sensors and moves with the object, the local frame that have the same origin of the body frame but their axes are aligned with the topographic directions and the Earth-Centered Earth-Fixed frame that have their origin in the centre of the Earth, their z-axis aligned with the north pole, and their y-axis with the Greenwich meridian.

Therefore, a set of equations are needed to represent the position, orientation, and motion of an object in a frame. And another set is required to transform the navigation from one frame to another frame.

2.1.1. Body Frame

The body frame is usually attached to the origin and orientated in the same way to the object. The axes for this frame remain fixed with respect to the body of the object. The most common is to adopt the x-axis pointing in the direction of travel (forward direction); z-axis pointing to the down (gravity direction in the initial condition), and y is the right axis, completing the orthogonal set. Usually, the initial position is defined as (0, 0, 0) in the instant before the motion starts.

Considering that the IMU is attached to the body of the object, it is possible to recognise that the sensor of the IMU will sense any linear acceleration or angular velocity in the body coordinate frame.

2.1.2. Local Navigation Frame

The Local Navigation Frame has its origin coincident with the corresponding body frame origin. However, the axes are aligned considering the topographic directions, not the

object. The convention used in this thesis is the North-East-Down (NED) with the x-axis point the north (N), the z-axis points Down (D) in the gravity direction, and the y-axis completes the orthogonal set by pointing east (E). Usually, the initial position is defined as $(0, 0, z)$ in an instant before the motion starts, where z is the height of the object related to the Earth surface at mean sea level.

Figure 2-1 shows the difference between the Local Frame and the Body Frame for an aeroplane doing a manoeuvre.

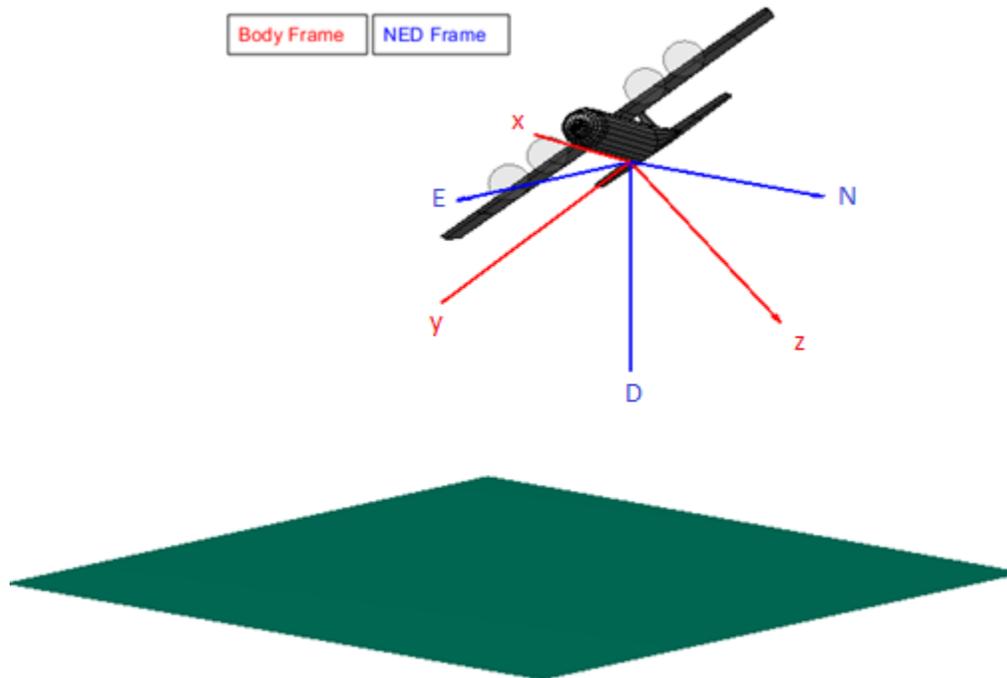


Figure 2-1 - Comparison between body (red) and local (blue) frames

2.1.3. Earth-Centered Earth-Fixed (ECEF) Frame

In the ECEF frame, the origin is the centre of the Earth, which is at the centre of mass. The z-axis points from the centre of the Earth to the north pole, along the Earth's axis of rotation. The x-axis points from the centre of the Earth to the intersection between the

equator and the Greenwich Meridian – zero latitude and zero longitude. And the y-axis points from the centre of the Earth to the intersection of the equator with the 90° longitude, according to the right-handed orthogonal set.

Figure 2-2 shows the difference between the Local Frame and the ECEF frame.

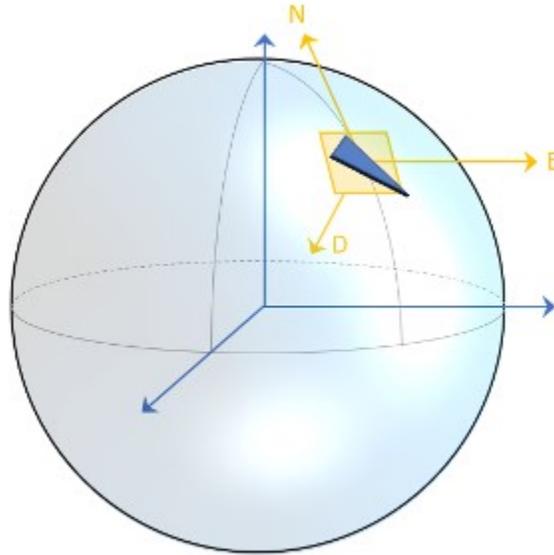


Figure 2-2 - Comparison between local (yellow) and ECEF (blue) frames

2.1.4. Earth and Gravity Models

2.1.4.1. Earth Ellipsoid Model

To use the Local Frame with respect to the ECEF frame, the distance between the centre of the Earth, used by the ECEF frame, and the Earth's surface, used by the Local Frame, must be known. Therefore, the first step is to describe the Earth surface in the ECEF frame. However, the Earth is not a perfect sphere, and modelling the real surface is not practical [1]. Therefore, the Earth is approximated as an ellipse at the mean sea level. One of the most common models used is the WGS84 system [37], which are also used in this work.

In Local Frame, the height is usually defined by the distance between the object and the Earth surface, using a normal direction related to the Earth surface in the ellipsoid model. However, the normal to an ellipsoid rarely intersect the ellipsoid centre, which is the origin for the ECEF frame. Therefore, the transformation between the Local and ECEF frames must take into account the eccentricity and the meridian radius of curvature that varies with the latitude position of the object. More details will be presented in section 2.1.5.2.

2.1.4.2. *Earth Rotation*

In ECEF coordinate systems the Earth rotation movement is described as a clockwise rotation about their z-axis. Therefore, the Earth-rotation vector in the ECEF frame is given by:

$$\omega_{er}^e = \begin{pmatrix} 0 \\ 0 \\ \omega_{ie} \end{pmatrix} \quad (2.1)$$

Where ω_{ie} is the Earth rotation rate, with the WGS 84 value of 7.292115×10^{-5} rad s⁻¹ [37].

2.1.4.3. *Centrifugal and Coriolis forces*

When the ECEF is used to describe the position of an object, even if the object is stationary with respect to this frame, it will be seen rotating at the Earth rotation rate with respect to an Earth Inertial frame (a frame that doesn't rotate together with the Earth). Therefore, looking by the Earth Inertial frame perspective, a centripetal force must be present to describe the object movement.

However, from the ECEF perspective, the same object is stationary, so the object has no acceleration even though the centripetal force is still present. Therefore, from the ECEF

perspective, a force that cancels the centripetal force must be present. This is the centrifugal force and is a pseudo-acceleration that arises from the use of a rotating reference frame.

Another situation that must be considered is when the object is moving north at a constant velocity. Again, with respect to the ECEF, the object has zero acceleration. However, with respect to an Earth Inertial frame, the object movement is curved. Therefore, a retarding acceleration force in a direction perpendicular to the velocity direction must be present. From the ECEF perspective, a second pseudo-force that opposes this retarding force must be present. This is the Coriolis force.

Therefore, these two pseudo-forces can be considered as an acceleration in ECEF, given by [1]:

$$a^e = -\Omega^e \Omega^e r^e - 2\Omega^e \dot{r}^e \quad (2.2)$$

Where Ω^e is the skew-symmetric matrix of the Earth rotation rate, and r^e is the position of an object in the ECEF. The first term of the presented equation is the centrifugal force, and the second term is the Coriolis force.

2.1.4.4. *Gravitation and Gravity*

Before presenting the model for the gravity used in this work, it is essential to define the difference between gravitation and gravity. Gravitation is the fundamental mass attractive force, discovered by Sir Isaac Newton. Gravity is what people and instruments sense when they are not moving in relation to the Earth, and its value will vary according to the reference frame chosen. Therefore, we are interested in knowing the gravity force, which will be given by the following relation considering ECEF:

$$g^e = \gamma^e - \Omega^e \Omega^e r^e \quad (2.3)$$

The first term of the equation is the gravitation acceleration, and the second term is the centrifugal force, already discussed in the previous section. The gravitation varies along the Earth's surface, and a simple model in ECEF is given by [1]:

$$\gamma^e = -\frac{\mu}{|r^e|^3} \left\{ r^e + \frac{3}{2} J_2 \frac{R_0^2}{|r^e|^2} \left\{ \begin{array}{l} \left[1 - 5 \left(\frac{r^e}{|r^e|^2} \right)^2 \right] r_x^e \\ \left[1 - 5 \left(\frac{r^e}{|r^e|^2} \right)^2 \right] r_y^e \\ \left[3 - 5 \left(\frac{r^e}{|r^e|^2} \right)^2 \right] r_z^e \end{array} \right\} \right\} \quad (2.4)$$

Considering the WGS 84 values [37], μ is the Earth gravitational constant of $3.986004418 \times 10^{14} \text{ m}^3 \text{ s}^{-2}$, r_m^e is the distance of the object from the center of the Earth, R_0 is the Equatorial radius of 6378137 m, J_2 is the Earth's second gravitational constant which value is 1.082627×10^{-3} , and r_x^e, r_y^e, r_z^e are the position in ECEF coordinate system.

2.1.5. Frame Transformations

2.1.5.1. Transformation from Body to Local NED coordinate frame

Looking at Figure 2-1, the transformation from Body to Local NED coordinate frames is done by three consecutive rotations, as following: around z-axis by an angle γ , around y-axis by an angle β , and around x-axis by an angle α . The angles γ , β , and α are respectively the Yaw, Pitch and Roll of the motion profile. This transformation can be represented by the following Direct Cosine Matrix (DCM):

$$C_b^L = \begin{bmatrix} c\beta c\gamma & -c\alpha s\gamma + s\alpha s\beta c\gamma & s\alpha s\gamma + c\alpha s\beta c\gamma \\ c\beta s\gamma & c\alpha c\gamma + s\alpha s\beta s\gamma & -s\alpha c\gamma + c\alpha s\beta s\gamma \\ -s\beta & s\alpha c\beta & c\alpha c\beta \end{bmatrix} \quad (2.5)$$

Where c and s are cosine and sine, respectively.

2.1.5.2. Transformation from Local NED to ECEF coordinate frame

To transform the motion from the NED to ECEF coordinate frame, the WGS84 system is used, where the earth shape is approximated as an ellipse [37]. It is convenient to utilize the ECEF position in the dynamic's equations, once the geodetic may cause some numerical problem during the calculations due to the differences between the Latitude and Longitude, presented in radians, and the height, presented in meters [1].

a) Position conversion

First, it is convenient to define the origin in geodetic Latitude (φ), Longitude (λ) and Height (h) position. Therefore, the initial states in ECEF can be obtained by:

$$\begin{aligned} r_{X_0}^e &= (R_N + h_0) \cos(\varphi_0) \cos(\lambda_0) \\ r_{Y_0}^e &= ((R_N + h_0) \cos(\varphi_0) \sin(\lambda_0)) \\ r_{Z_0}^e &= [(R_N(1 - e^2) + h_0] \sin(\varphi_0) \end{aligned} \quad (2.6)$$

Where R_N is the radii of curvature of the Earth and e is the Earth ellipsoid parameter.

Second, an offset vector from the local system origin, rotated from NED to ECEF can be obtained by:

$$\begin{aligned}
r_{dX}^e &= \cos(\lambda_0)(\cos(\varphi_0)D(t) - \sin(\varphi_0)N(t)) - \sin(\lambda_0)E(t) \\
r_{dY}^e &= \sin(\lambda_0)(\cos(\varphi_0)D(t) - \sin(\varphi_0)N(t)) + \cos(\lambda_0)E(t) \\
r_{dZ}^e &= \sin(\varphi_0)D(t) + \cos(\varphi_0)N(t)
\end{aligned} \tag{2.7}$$

Where N , E , and D are the position in the NED frame.

Finally, the position in ECEF frame in time (t) can be obtained by:

$$\begin{aligned}
r_X^e(t) &= r_{X_0}^e + r_{dX}^e \\
r_Y^e(t) &= r_{Y_0}^e + r_{dY}^e \\
r_Z^e(t) &= r_{Z_0}^e + r_{dZ}^e
\end{aligned} \tag{2.8}$$

b) Velocity conversion

Velocity may be transformed from the NED frame to ECEF using the appropriate coordinate transformation matrix, as follows:

$$v_b^e = C_L^e v_b^L \tag{2.9}$$

Where the C_L^e is a sequence of rotation between NED and ECEF frame give by:

$$C_L^e = \begin{bmatrix} -\sin(\varphi)\cos(\lambda) & -\sin(\lambda) & -\cos(\varphi)\cos(\lambda) \\ -\sin(\varphi)\sin(\lambda) & \cos(\lambda) & -\cos(\varphi)\sin(\lambda) \\ \cos(\varphi) & 0 & -\sin(\varphi) \end{bmatrix} \tag{2.10}$$

c) Attitude conversion

Similar to the velocity conversion, the transformation between NED to ECEF attitude may be done using:

$$a_b^e = C_L^e a_b^L \tag{2.11}$$

2.1.6. Quaternion

Using Euler angles to describe the orientation can bring a problem called the singularity, that occurs when the second angle in the sequence of rotations describe by equation 2.5 is around 90 or 270 degrees. In this situation, the other two angles will be aligned, making it not possible to differentiate between both. As an example, if the pitch angle became 90 degrees, the roll and yaw direction will be aligned, and any movement can be both yaw and roll.

Therefore, a mathematical tool called quaternion can be used to represent rotations to avoid Euler angles singularity. A quaternion is a hyper-complex number with four components [38]:

$$q = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \quad (2.12)$$

The attitude in DCM can be converted to and from quaternion using the following relation:

$$C = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_0) & 2(q_1q_3 - q_2q_0) \\ 2(q_1q_2 - q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_0) \\ 2(q_1q_3 + q_2q_0) & 2(q_2q_3 - q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.13)$$

$$\begin{aligned} q_0 &= \frac{1}{2} \sqrt{1 + C_{1.1} + C_{2.2} + C_{3.3}} \\ q_1 &= \frac{C_{3.2} - C_{2.3}}{4 q_0} \\ q_2 &= \frac{C_{3.1} - C_{1.3}}{4 q_0} \\ q_3 &= \frac{C_{1.2} - C_{2.1}}{4 q_0} \end{aligned} \quad (2.14)$$

2.2. Inertial Navigation

As Figure 2-3 shows, an Inertial navigation system (INS) is mainly consisted of an Inertial Measurement Unit (IMU) and a Processing Unit. The IMU aims to measure specific forces acting on an axis using accelerometers, and angular rates through gyroscopes, of a moving body. Processing Unit will receive the sensor measurements from the IMU and apply a dynamic model to determine the change in the previous position, velocity, and attitude.

Through the combination of multiple accelerometers with multiple gyroscopes, usually three of each - one for each frame axis, connected to a processing unit, it is possible to determine the position, velocity, and attitude of a body performing three-dimensional navigation, without any external reference.

There are several types of accelerometers and gyroscopes, which are used in different kinds of applications. Their size, mass, performance, and cost may vary by several orders of magnitude depending on the technological solution needed.

In general, sensors that require greater precision are used in missiles, rockets, ships, and submarines, as they perform long-time navigation, where a small measurement error in the IMU will cause significant deviations in the navigation solution. According to Grove [1], a marine navigation system can cost more than a million dollars and offer a navigation solution that provides a deviation of less than 1.8 km per day of navigation.

The INS suffer degradation in long-term navigation as the errors presented in the sensors are integrated through the navigation equations. Therefore, one solution for this problem is to fuse the INS with another type of sensor, where the disadvantages of the INS can be compensated, bring a better overall navigation solution.

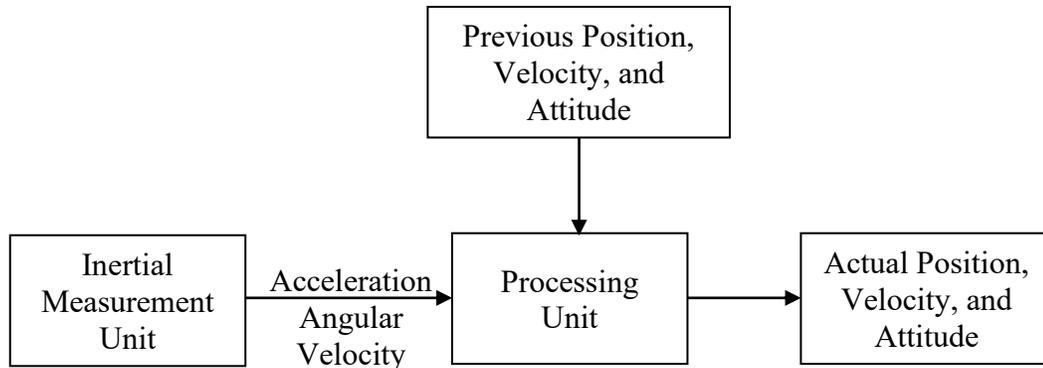


Figure 2-3 - Basic diagram of an INS

2.2.1. The INS dynamic model

The INS dynamic model considers that the initial position, velocity, and orientation of the motion are known. Still, after the navigation started, the only input in the system will be the gyroscope and accelerometer sensor data. This way, a new position, velocity, and orientation will be calculated based on the previous position, velocity, and orientation and the actual sensor measurement. Figure 2-4 shows the steps considered for the INS dynamic model.

By doing that, any noise and bias in the gyroscope and accelerometer will be accumulated over time, drifting the calculated position, velocity, and attitude in relation to the true ones.

The INS dynamic model is used to convert the Euler angles roll, pitch and yaw into quaternion vector, avoiding the singularities that may occur when using the Euler angles for navigation.

It was considered that the Coriolis force is much smaller than gravity. Therefore, the variation of the Coriolis force over the time interval will be neglect.

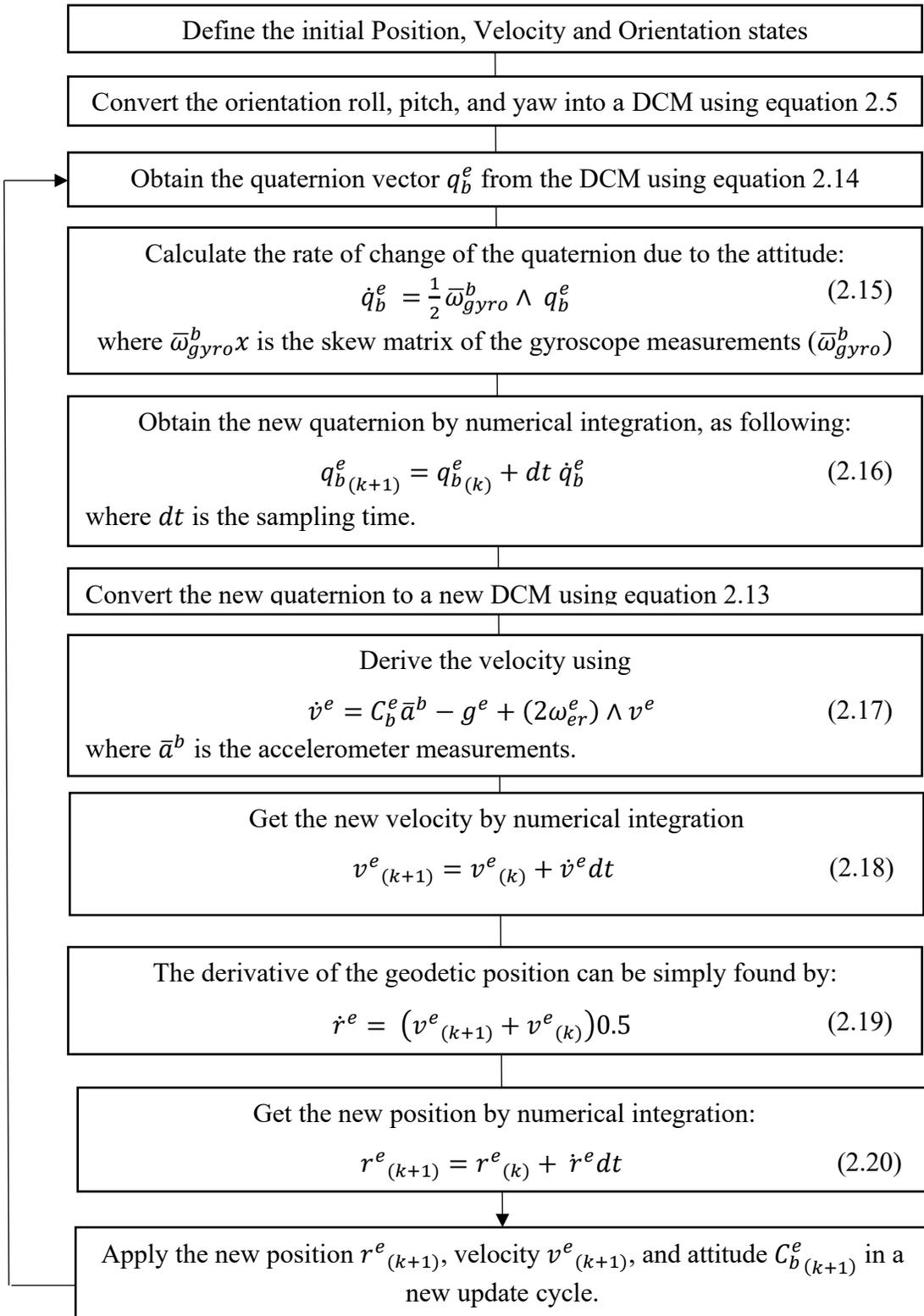


Figure 2-4 - INS dynamic model

2.2.2. Gyroscope and accelerometer raw data obtained by inverse kinematics

The inverse kinematics is a model whose purpose is to generate raw IMU data that can be used to test the INS dynamic model.

Once the velocity profile is known, it is possible to calculate the acceleration in the ECEF frame by simply derivate the velocity by the time, as follows.

$$a^e = \frac{dv^e}{dt} \quad (2.21)$$

The raw accelerometer data is the acceleration a^b represented in the body frame. However, the gravity and the Coriolis force must be compensated. Therefore, the raw accelerometer can be obtained by [1]:

$$a^b = C_e^b(a^e - g^e + (2\omega_{er}^e) \wedge v^e) \quad (2.22)$$

Where $\wedge v^e$ is the cross product between the velocity vector and the Coriolis force ($2\omega_{er}^e$).

The gravity in the ECEF coordinate system can be obtained by equation 2.3, as following [1]:

$$g^e = \gamma^e + \omega_{tr}^e{}^2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} r^e \quad (2.23)$$

Gravity is a composition of two terms. The first is the gravitational acceleration, gives by γ^e , and the second is the centrifugal acceleration due to the Earth's rotation, which is applied to maintain an object stationary with respect to the rotating Earth, as discussed in section 2.1.4.4.

The raw gyroscope data (ω_{gyro}^b) can be obtained by calculating the derivate of the DCMs. This is done by calculating the difference between two consecutive DCMs, divided by the time period.

$$\dot{C}_b^e = \frac{C_b^e(k) - C_b^e(k-1)}{dt} \quad (2.24)$$

In this case, the result will be a skew matrix ($S_g^B x$), as following [1]:

$$S_g^B x = \frac{C_e^b(k-1)C_b^e(k) - I}{dt} \quad (2.25)$$

Where I is the identity matrix.

The raw gyroscope data in the body frame will be taken from specific positions in the $S_g^B x$ matrix, where the data in x, y and z-axis are acquired from the positions $S_g^B(3,2)$, $S_g^B(1,3)$, and $S_g^B(2,1)$, respectively.

The obtained gyroscope value must be compensated by the earth rotate rate and transport rate. Therefore, the final raw gyroscope data will be found by:

$$\omega_{gyro}^b = \omega_{gyro}^b + {}^b_e C (\omega_{tr}^e + \omega_{er}^e) \quad (2.26)$$

2.2.3. Error sources

The deviation between the INS indicated, and the true navigation position, velocity, and attitude occur mainly due to intrinsic errors in the accelerometers and gyroscopes sensor, and due to their final assembly. Such errors are characterized as bias, scale factor, cross-coupling, and random noise, which can be divided into deterministic or random deviations. Deterministic deviations have a predictable characteristic and, therefore, can be compensated by the INS processing unit. To do so, the sensors and the IMU must be characterized to know the existing deviations. Random deviations, on the other hand, have

unpredictable characteristic and are usually caused by the physical limit of sensitivity of the sensors. This deviation is not normally compensable, but their characterization is essential to define whether a sensor is suitable or not for a specific application.

The two main deterministic errors are the bias and the scale factor. The bias is a drift error presenting by the sensor that is independent of the forces, and angular velocities applied, being present in all accelerometers and gyroscopes. The scale factor is the error showing when the sensor output is not proportional to its input, being a reason for the variation in the output with a variation in the input. If these errors are accurately characterized, they can be compensated by the INS.

Usually, to define the suitability of a sensor for an application, the unpredictable errors are considered. It is possible to identify three primary sources for this error type: the noise, the bias instability, and the scale factor instability.

Figure 2-5 and Figure 2-6, taken from [39], show the biases and scale factors instability for accelerometers and gyroscopes, respectively, considering certain applications and technologies used in the manufacture of sensors. Looking at these figures, it is possible to see that, for example, for applications in submarines, biases instability smaller than $50 \mu\text{g}$ for the accelerometer and lower than $0.01 \text{ }^\circ \text{ hr}^{-1}$ for gyroscope are indicated. To achieve this accuracy, it is necessary to use specific sensors that are often not available for purchase due to their strategic importance.

Therefore, to improve the usability of standard IMU, one of the leading solutions is to fuse the INS with another type of sensor, where the disadvantages of the INS can be compensated, bring a better overall navigation solution.

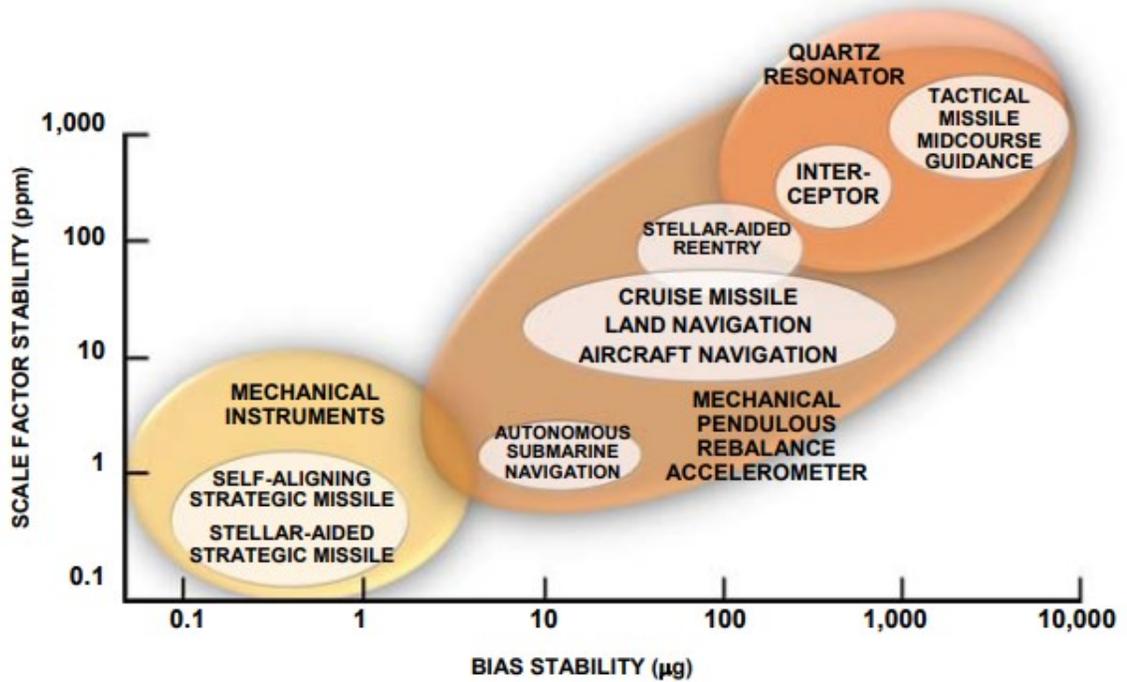


Figure 2-5 - Biases and scale factors instability for accelerometers considering certain applications and technologies used in the manufacture of sensors [39].

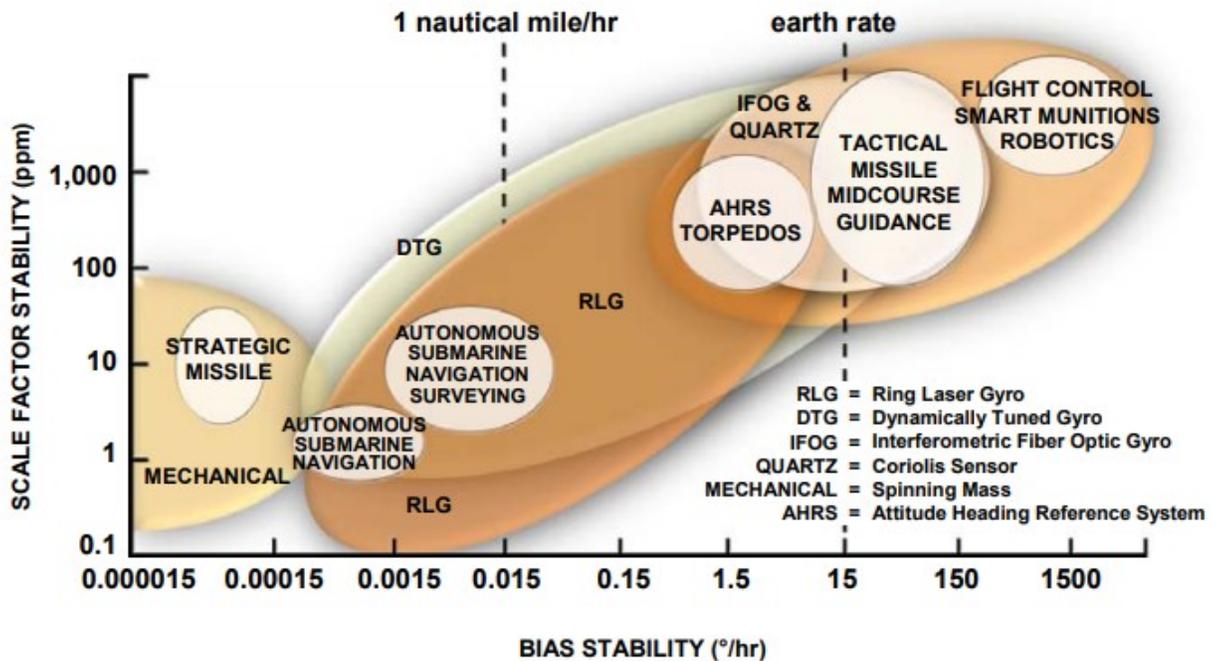


Figure 2-6 - Biases and scale factors instability for gyroscopes considering certain applications and technologies used in the manufacture of sensors [39].

2.2.3.1. IMU Noise Model

A typical IMU sensor measurement can be modelled as follow, to consider noisy in the analyses [1]:

$$\begin{aligned}\bar{\omega}_{gyro}^b &= (1 + M_g)\omega_{gyro}^b + b_g + Ga^b + w_g \\ \bar{a}^b &= (1 + M_a)a^b + b_a + w_a\end{aligned}\tag{2.27}$$

Where $\bar{\omega}_{gyro}^b$ and \bar{a}^b are respectively the noisy gyro and noisy accelerometer measurements, ω_{gyro}^b and a^b are the true measurement, M_g and M_a are the scale factor and cross-coupling error, b_g and b_a are the bias, w_g and w_a are a Gaussian white random noise, and G is the gyro g-dependent biases.

The sensor scale factor and cross-coupling error can be expressed by the following:

$$M_{Sensor} = \begin{pmatrix} S_{Sensor,x} & m_{Sensor,xy} & m_{Sensor,xz} \\ m_{Sensor,yx} & S_{Sensor,y} & m_{Sensor,yz} \\ m_{Sensor,zx} & m_{Sensor,zy} & S_{Sensor,z} \end{pmatrix}\tag{2.28}$$

Where S_{Sensor} and m_{Sensor} are the scale factor and cross-coupling, respectively.

The presented model considers that the unpredictable errors are Gaussian white random noises. This means that the noise will show a probability distribution function (PDF) that can be described by a Gaussian distribution with zero mean. However, the biases and scale factors instability can change the noise characteristic over time. Therefore, the noise will no longer be white, changing its mean value over time.

2.2.3.2. IMU with Coloured noise

A typical IMU sensor model considering noisy in the measurements was presented in equation 2.27 where w_g and w_a are Gaussian white random noise for the gyroscope and accelerometer, respectively. This means that the noise is a time series with power spectral density (PSD) that follow a power law with the form of:

$$S(f) = L(f) \quad (2.29)$$

To simulate a coloured noise, a $1/f$ filter can be applied to the white noise w_g and w_a , providing a power spectral density (PSD) of [40]:

$$S(f) = \frac{L(f)}{|f|} \quad (2.30)$$

This is equivalent to consider that the Raw IMU data have a $1/f$ flicker noise in its measurements.

Figure 2-7 presents a probability density function (PDF) for a gyroscope with white noise and with coloured noise. The coloured noise was created by applying a $1/f$ filter in the data with white noise. When the IMU noise is white, the PDF that describes the signal becomes constant after a short period of time. When the IMU noise is coloured, the PDF that describes the signal will vary over time, presenting a different value for each time instant, becoming broader as time goes by.

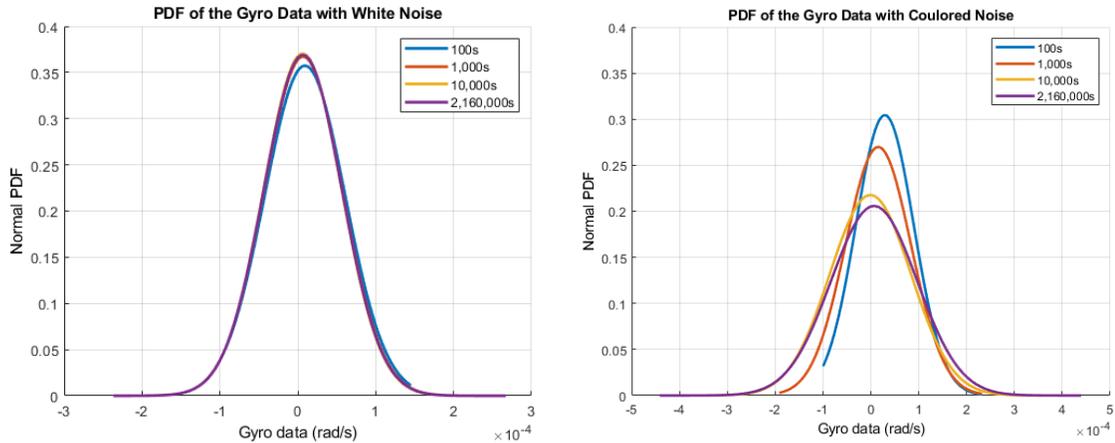


Figure 2-7 – Gyroscope simulated with white noise (left) and with coloured noise (right)

2.3. GNSS

GNSS is a global navigation system based on satellite technology. It provides to the user a 3-D position (and velocity) solution by measure the ranges between a receiver and a few observed satellites. The most used GNSS is the GPS - Global Positioning System was created and is maintained by the U.S. Department of Defence. Another GNSS is the Galileo, developed by the European Union, and the GLONASS, managed by the Russian Space Forces.

The GNSS provides reasonable accuracy for long-term navigation, as the position and velocity are updated continuously by the satellites. However, the short-term errors are high, the signals are subject to obstruction and interference, and the standard GNSS does not provide attitude.

2.3.1. GNSS principles

A GNSS solution is determined using a passive pseudorange in a 3-dimensional space. Determine the positioning by ranging means that the position of an object is determined by measuring the range to other objects with a known position. Pseudorange is a measure of

the distance through measuring the time that the satellite signal spends to reach the receiver's antenna. Passive means that the receiver will just receive and process the information transmitted by the satellite, without sending any data. Therefore, the essential GNSS measurement is the time that takes for the signal to travel from the satellite to the receiver antenna.

The message in the satellite signal contains a time of emission, determined by a precise atomic clock. The receiver records the time of arrival of the signal and identifies how long time it took for the signal travel from the satellite to the receiver. As all electromagnetic radiation travels at the constant speed of light, the receiver can determine the range to the satellite.

Usually, to determine the position in a 3-Dimensional space using the range, the receiver must observe at least three satellites. However, as the range is measured using time, a fourth satellite must be used to correct the time in the receiver.

The pseudorange for the satellite corrected for the receiver clock bias (δt), and the atmosphere errors (ε^m), is given by [41]:

$$\rho_c^m = r^m + c\delta t + \varepsilon^m \quad (2.31)$$

Where c is the speed of light and r^m is true range from the satellite to receiver, that can be obtained in the ECEF system, as follows:

$$r^m = \sqrt{(x - x^m)^2 + (y - y^m)^2 + (z - z^m)^2} \quad (2.32)$$

The Doppler Effect is a phenomenon where the electromagnetic signal presents a frequency shift caused by the relative motion of the emitter and receiver. Based on this, some GPS

receivers also measure the Doppler frequency of the received GPS signal to determine the velocity state. Therefore, in this situation, the GNSS will provide the position and the velocity as a navigation solution.

2.3.2. Error sources

The error sources in the GNSS solution arises from many factors such as the receiver measurement, the atmosphere condition, the elevation of the satellite, the environmental near the receptor, and intentional interference - called Jamming [41].

The receiver measurement errors usually occur by loose of the carrier tracking, which can happen due to receiver noise, High Dynamics, radio frequency interference or multipath interference. Therefore, even in good reception condition, the GNSS navigation solution will have a variance of a few meters.

The atmosphere condition may cause a signal propagation delay due to refraction of the signal, which may occur in the ionosphere and troposphere. These can be partially corrected by using atmosphere models.

The satellite elevation is the angle between the horizon line and the satellite, as observed by the receptor. Low angles will cause large ionosphere and troposphere delays and multipath interference. In this situation, if other satellites are available, most receivers will just ignore satellites that are in low angle elevations.

The environmental near the receptor can be the biggest challenge for the GNSS navigation solution. The signal can be blocked, attenuated, or reflected by tunnels, buildings, mountains, etc. In this case, the GNSS solution will present significant errors, or even it will be not possible to determine the navigation solution.

The GNSS jamming occurs because the GNSS signal is weak compared to other radio signals. Therefore, if a stronger signal that exactly matches the GPS signal structure is generated, the GPS receiver may think that this is a real GPS satellite signal and then will use it for the navigation determination.

There are several methods to avoid or reduce all these problems. Among them is the integration with INS.

2.4. INS/GNSS Integration model

The architecture of an INS/GNSS integrated system can be defined by the way that the corrections are used, the type of the GNSS measurements considered, and the integration algorithm applied. Therefore, many different types of integration can be used and combined. However, three architectures are usually found in the literature: loosely coupled, tightly coupled, and deeply coupled.

The loosely coupled architecture, presented by Figure 2-8 use as measurement inputs the: INS - position, velocity and attitude, and the GNSS - position and velocity. In this architecture, the corrections are done in the position and velocity solution.

The tightly coupled INS/GNSS architecture differs from the loosely coupled as it uses the GNSS pseudo-range and pseudorange-rate as measurements inputs. This is called range-domain integration.

The deeply coupled INS/GNSS architecture combines INS/GNSS integration and GNSS signal tracking into a single estimation algorithm. This is called tracking-domain integration.

There are two advantages in using the loosely coupled architecture: simplicity and redundancy [1]. Both advantages come from the fact that the measurements of the GNSS are navigation states. Therefore, any INS and any GNSS equipment can be used, and an integrated navigation solution is available together with both stand-alone GNSS and stand-alone INS navigation solutions, allowing reach to a navigation solution even when one of the sensors are not available.

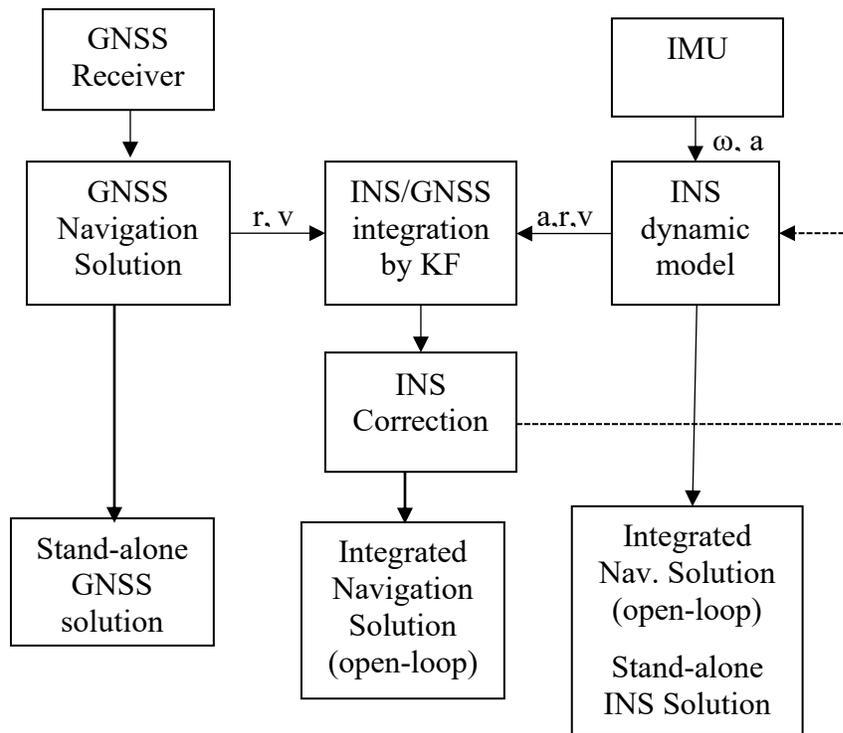


Figure 2-8 - Loosely coupled INS/GNSS system using KF

2.5. Kalman Filter

The Kalman filter is the only practical finite-dimensional solution to the real-time optimal estimation problem for stochastic systems [2]. It was invented by R. E. Kalman in 1960 [42] and has been studied and improved by numerous authors since then. Its mathematical model has been used in a wide range of essential applications for estimating the current conditions of dynamic systems when unpredictable disturbances are present. The Kalman

filter uses deterministic and statistical properties of the system parameters and new measurements to obtain the optimal estimates for the system.

Just after R. E. Kalman published his work in 1960, one of the persons in charge of the Apollo project, Stanley F. Schmidt, recognized its potential applicability in the estimation and control of the trajectory of the Apollo spacecraft. The use in the Apollo project is known as the first full implementation of the Kalman filter [2]. Schmidt, using the work from Kalman, developed a nonlinear solution for the KF, called Extended Kalman Filtering (EKF), which has been used ever since for many real-time nonlinear applications.

Later, in 1995, Jeffrey Uhlmann et al. [43], [44] developed a nonlinear adaptation of the Kalman filter called Unscented Kalman Filter (UKF). The UKF has about the same complexity and stability as the EKF but with potentially greater robustness against nonlinear effects.

2.5.1. The Error-State model

In the error-state model, the Kalman Filter is used to indicate the error between the predicted and the measurement values. The advantage is that when the error-state are considered, the dynamic will be represented by small, linear, and suitable for linear-Gaussian filtering signals. On the other side, when true states are considered, significant non-linear signals will represent the dynamic of the system.

Furthermore, the equations for a Total State EKF are the same as for a closed-loop ESKF model [1]. Therefore, the same behaviour is expected for both approaches. The difference between the two models remains on the implementation.

In the Error-State Kalman Filter used in an INS/GNSS integration, a high-frequency IMU provides the measurement data that are processed in order to describe the true navigation-states. This step does not consider the errors, noise, and other model imperfections occurred in the system. Consequently, the errors will be accumulated, and the navigation solution will drift over time. Therefore, these errors must be estimated and corrected to provide an accurate solution. The KF does this. Using measurements from GNSS, the system can predict and correct the errors in the IMU. After estimated, the predicted errors are feedback to the INS solution aim to correct all states. This correction provides a posterior Gaussian estimate of the error-state, which can also be used to adjust the system. Using this solution, high-frequency sensors, like gyroscopes and accelerometers, can be fused with low-frequency sensors, as GNSS and Vision.

Furthermore, according to Madyastha et al. [45], the advantages of using ESKF over the Total State EKF are:

- a) The ESKF is always operating close to the origin, far from possible parameter singularities, like gimbal locks, providing a guarantee that the linearization is always valid.
- b) The error-state always has small values, meaning that all second-order products are negligible. This makes the computation of partial derivatives easier and faster. Some terms may even be considered as constant or equal to a state magnitude.
- c) It is possible to apply the KF corrections (which are the only means to observe the errors states) at a lower rate.

The Error-State Kalman Filter can be used to estimating the error in the attitude, velocity, and position, and also the bias for the accelerometer and gyroscope. In this case, the state vector can be described as following, in an ECEF frame [46]:

$$x^e = \begin{bmatrix} \delta \hat{A}^e \\ \delta \hat{r}^e \\ \delta \hat{v}^e \\ \widehat{ba}^e \\ \widehat{bg}^e \end{bmatrix} \quad (2.33)$$

Where, $\delta \hat{A}^e$, $\delta \hat{r}^e$, and $\delta \hat{v}^e$ are the attitude, position and velocity errors, respectively. And \widehat{ba}^e and \widehat{bg}^e are the accelerometer and gyroscope biases, respectively.

Therefore, the following relations provides the system dynamics:

The attitude error derivative can be given by:

$$\delta \dot{A}^e = C_b^e \delta \omega_{gyro}^b - \Omega^e \delta A^e \quad (2.34)$$

Where $\delta \omega_{gyro}^b$ is the error in the gyroscope.

If it is considered that the gyroscope bias can represent this error, equation 2.34 becomes:

$$\delta \dot{A}^e = C_b^e b g^e - \Omega^e \delta A^e \quad (2.35)$$

The rate of change of the velocity error in ECEF can be given by:

$$\dot{v}^e = C_b^e \bar{a}^b - g^e + (2\omega_{er}^e) \wedge v^e \quad (\text{Repeat. Eq. 2.17})$$

$$\delta \dot{v}^e = -C_b^e \bar{a}^b \wedge \delta A^e - 2 \Omega^e \delta v^e - F_{23}^e + C_b^e b g^e \quad (2.36)$$

Where

$$F_{23}^e = -\frac{2\gamma^e r_m^e T}{r_{es(L)}^e r_m^e}$$

$r_{eS}^e(L)$ is the geocentric radius at the surface of the Earth. Equation 2.36 considers that the centrifugal term is small and can be neglected. More information can be found on the reference [1].

Finally, the time derivative of the position error is given by:

$$\delta \dot{r}^e = \delta v^e \quad (2.37)$$

Therefore, the matrix that represents the dynamic of the system for the states given by equation 2.33, will be given considering equations 2.35, 2.36 and 2.37, as following:

$$F^e = \begin{bmatrix} -\Omega_{ie}^e & 0_{3x3} & 0_{3x3} & 0_{3x3} & C_b^e \\ 0_{3x3} & 0_{3x3} & I_{3x3} & 0_{3x3} & 0_{3x3} \\ -(C_b^e a^b)\Lambda & F_{23}^e & 2\Omega_{ie}^e & C_b^e & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \quad (2.38)$$

This relation is the transition matrix used in the KF for continuous-time.

2.5.2. Error-State Kalman Filter model

The full ESKF solution in the ECEF frame, considering the discrete-time algorithm, for an error-states implementation, can be described in three phases with a total of eleven steps, as following [1]:

The first phase - System Propagation:

- 1) Determine the first-order transition matrix considering discrete-time (from equation 2.38):

$$\Phi = \begin{bmatrix} I_{3x3} - \Omega_{ie}^e dt & 0_{3x3} & 0_{3x3} & 0_{3x3} & C_b^e dt \\ 0_{3x3} & I_{3x3} & I_{3x3} dt & 0_{3x3} & 0_{3x3} \\ -(C_b^e a^b) \Lambda dt & F_{23}^e dt & I_{3x3} - 2\Omega_{ie}^e dt & C_b^e dt & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & I_{3x3} \end{bmatrix} \quad (2.39)$$

Where dt is the sampling time.

- 2) Determine an approximate system noise covariance matrix using:

$$Q_{INS} = \begin{bmatrix} S_{gr} I_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & S_{ar} I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & S_{gbd} I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & S_{gbd} I_{3x3} \end{bmatrix} dt \quad (2.40)$$

Where S_{gr} and S_{ar} are the gyro and accelerometer random noise PSD respectively; and the S_{gbd} and S_{abd} are the gyro and accelerometer bias variation PSD respectively.

- 3) Propagate the estimated states, by:

$$\hat{x}_k^- = \begin{bmatrix} \delta \hat{A}^e \\ \delta \hat{r}^e \\ \delta \hat{v}^e \\ \widehat{ba}^e \\ \widehat{bg}^e \end{bmatrix}_k = \Phi_{k-1} \begin{bmatrix} \delta \hat{A}^e \\ \delta \hat{r}^e \\ \delta \hat{v}^e \\ \widehat{ba}^e \\ \widehat{bg}^e \end{bmatrix}_{k-1}^+ \quad (2.41)$$

It can be considered that all previous states are zero due to closed-loop correction.

Therefore, this step can be omitted.

- 4) Propagate the state estimation error covariance matrix, using:

$$P_k^- \approx \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + Q_{INS_{k-1}} \quad (2.42)$$

The second phase – Measurements Update:

- 5) Set-up the measurement matrix, considering that the distance between the IMU and the GNSS antenna is minimal and can be neglected:

$$H_k = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (2.43)$$

- 6) Determine the measurement noise covariance matrix using:

$$R_k = E(w_m w_m^T) \quad (2.44)$$

Where w_m is the GNSS measurement variance of the position and velocity. If it is assumed that all components of GNSS position and velocity are independent and have equal variance, the R_k can be estimated by:

$$R_k = \begin{bmatrix} w_p 3 \times 3 & 0_{3 \times 3} \\ 0_{3 \times 3} & w_v 3 \times 3 \end{bmatrix} \quad (2.45)$$

Where w_p and w_v are, respectively, the position and velocity GNSS variance.

- 7) Calculate the Kalman gain matrix by:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (2.46)$$

- 8) The measurement state can be given by the error between the GNSS and INS states, as follow:

$$\delta z_k^{e-} = \begin{bmatrix} r_{INS}^e - r_{GNSS}^e \\ v_{INS}^e - v_{GNSS}^e \end{bmatrix} \quad (2.47)$$

- 9) Update the state estimates using:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \delta z_k^{e-} \quad (2.48)$$

- 10) Update the state estimation error covariance matrix using:

$$P_k^+ = (I - K_k H_k) P_k^- \quad (2.49)$$

Third phase – Closed-loop correction:

11) Correct the attitude, velocity, and position of the INS solution using:

$$\begin{aligned}
 \hat{C}_b^{e+} &= \delta \hat{C}_b^{e- T} \hat{C}_b^{e-} \\
 \hat{v}^{e+} &= \hat{v}^{e-} - \delta \hat{v}^e \\
 \hat{r}^{e+} &= \hat{r}^{e-} - \delta \hat{r}^e
 \end{aligned} \tag{2.50}$$

Where $\delta \hat{C}_b^{e- T}$ can be approximated by

$$\delta \hat{C}_b^{e- T} = (I_{3 \times 3} - [\delta \hat{A}^e \Lambda])$$

2.5.3. Unscented Kalman Filter model

There are several sampling strategies for computing the weighted averages of propagated statistics when applying UKF. Therefore, before defining the procedure used in this work, three different approaches, presented in Table 2, were simulated to determine the best one.

Table 2 - Unscented Transformation Sample Weights

Approach	Sampling Strategy	Sample size	Sample values	Sample Weights
1	Symmetric	2n	$\hat{x}_{k-1}^+ + \sqrt{n} S_{k-1}^+$ $\hat{x}_{k-1}^+ - \sqrt{n} S_{k-1}^+$	$W = 1/(2n)$
2	Symmetric	2n+1	\hat{x}_{k-1}^+ $\hat{x}_{k-1}^+ + \sqrt{n+k} S_{k-1}^+$ $\hat{x}_{k-1}^+ - \sqrt{n+k} S_{k-1}^+$	$W = 1/(2n+k)$
3	Scaled	2n+1	\hat{x}_{k-1}^+ $\hat{x}_{k-1}^+ + \sqrt{n+\lambda} S_{k-1}^+$ $\hat{x}_{k-1}^+ - \sqrt{n+\lambda} S_{k-1}^+$ $\lambda = \alpha^2(n+k) - n$	$W = \lambda/(2n+\lambda)$

- $\alpha, \lambda,$ and k are tuning parameters.

After tuning all parameter, all the three approaches showed similar results, and the only difference was that the approach 2 and 3 showed a light better outcome for the measured states but a slight worst result for the other states (unmeasured) when compared with the approach one. This means that in the considered system, where the UKF has 15 states, when approaches 2 and 3 were used, six states (the position and velocity measured by the GNSS) showed slightly better results than the approach 1. However, the other nine states – 3-dimensional attitude, gyroscope bias, and accelerometer bias – showed marginally worse results.

Therefore, the approach number 1 was chosen to be used in the project due to the simplicity of the system, once no tuning parameter is used, and the better results for the attitude determination. This approach will be described in detail.

The full UKF solution in ECEF frame, considering discrete-time algorithm, for an error-states implementation, using the symmetric sampling strategy with $2n$ of sample size, can be described in three phases with a total of eleven steps as follow [47]:

The first phase - System Propagation:

- 1) Obtain the square root of the error covariance matrix by using Cholesky factorization.

$$P_{k-1}^+ = S_{k-1}^+ S_{k-1}^{+T} \quad (2.51)$$

The eigenvalue–eigenvector Cholesky factor was used.

- 2) Calculate the sigma points.

$$x_{k-1}^{+(i)} = \begin{cases} \hat{x}_{k-1}^+ + \sqrt{n} S_{k-1}^+ & i \leq n, \\ \hat{x}_{k-1}^+ - \sqrt{n} S_{k-1}^+ & n + 1 \leq i \leq 2n. \end{cases} \quad (2.52)$$

With n is the number of states.

- 3) The calculated sigma points are propagated using:

$$x_k^{-(i)} = \hat{x}_{k-1}^+ + f(x_{k-1}^{+(i)}, t_k) dt \quad (2.53)$$

Where $f(x_{k-1}^{+(i)}, t_k)dt$ is the transition matrix in discrete-time given by equation 2.39.

- 4) The state estimate and its error covariance are propagated using:

$$\hat{x}_k^- = \frac{1}{2n} \sum_i^{2n+1} x_k^{-(i)} \quad (2.54)$$

$$\hat{P}_k^- = \frac{1}{2n} \sum_i^{2n+1} (x_k^{-(i)} - \hat{x}_k^-)(x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \quad (2.55)$$

Where Q_{INS} is the same as defined equation 2.40.

The second phase – Measurements Update:

- 5) Generate new sigma points by:

$$x_k^{-(i)} = \begin{cases} \hat{x}_k^- + \sqrt{n}S_k^- & i \leq n, \\ \hat{x}_k^- - \sqrt{n}S_k^- & n + 1 \leq i \leq 2n. \end{cases} \quad (2.56)$$

- 6) Obtained both the sigma points and the mean measurement innovation by:

$$\begin{aligned} \delta z_k^{-(i)} &= z_k - h(\hat{x}_k^{-(i)}, t_k) \\ \delta \hat{z}_k^- &= \frac{1}{2n} \sum_i^{2n+1} \delta z_k^{-(i)} \end{aligned} \quad (2.57)$$

Where $h(\hat{x}_k^{-(i)}, t_k)$ can be defined as $H_k \hat{x}_k^{-(i)}$. The H_k is the same as defined in equation 2.40.

- 7) Calculate the measurement innovations covariance, by:

$$C_k^- = \frac{1}{2n} \sum_i^{2n+1} (\delta z_k^{-(i)} - \delta \hat{z}_k^-)(\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T + R_k \quad (2.58)$$

Where R_k is the same as defined in equation 2.44.

8) Calculate the Kalman Gain:

$$K_k = \left[\frac{1}{2n} \sum_i^{2n+1} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right] (C_k^-)^{-1} \quad (2.59)$$

9) Update the state estimates using:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \delta z_k^- \quad (2.60)$$

10) Update the state estimation error covariance matrix using:

$$P_k^+ = P_k^- - K_k C_k^- K_k^T \quad (2.61)$$

Third phase – Closed-loop correction:

11) Correct attitude, velocity, and position of the INS solution using equations 2.50.

2.6. Kalman Filter Divergence

Filter divergence occurs when the actual error of the system is not bounded in the predicted covariance P [48]. In this situation, two types of divergence can be considered: the apparent and the true. In apparent divergence, the actual error of the system remains bounded but in a larger bound than the predicted error covariance. In this situation, the filter will be suboptimal, leading to a solution that converges to a higher bound.

In the true divergence, the actual error of the system tends to infinity. This occurs when the system is unstable or when unstable states are present and are not modelled. Figure 2-9 presents the difference between an Optimal, Apparent divergence, and True divergence error covariance.

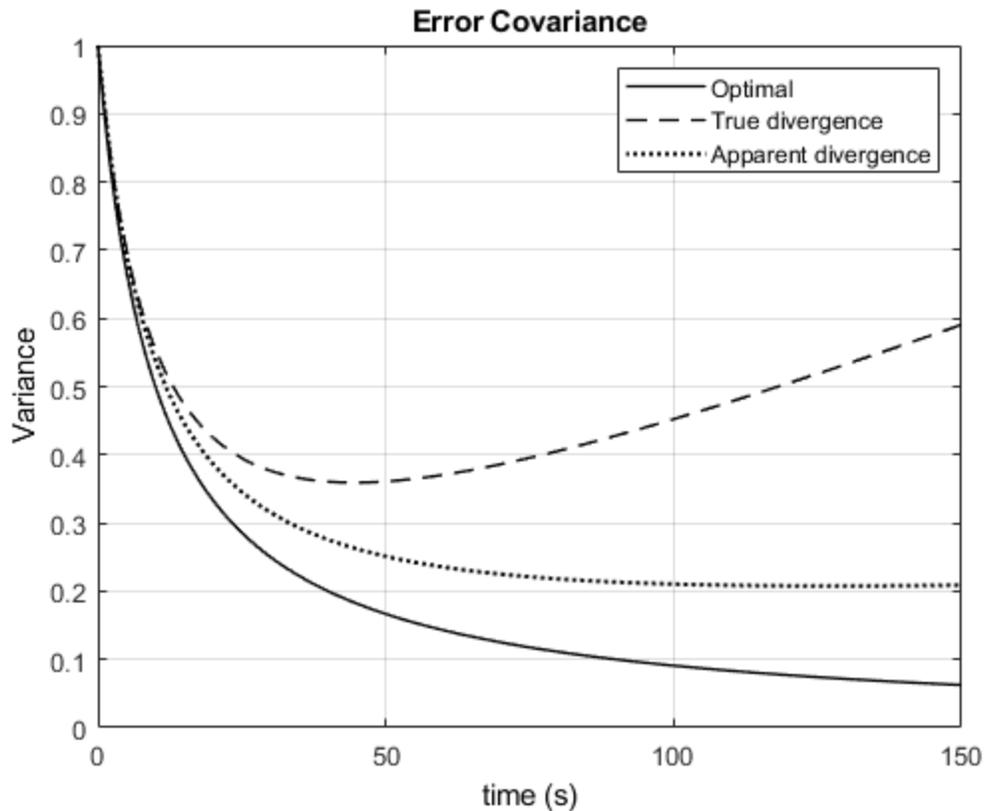


Figure 2-9 - Optimal, Apparent divergence, and True divergence for error covariance

In Kalman Filter, the predicted error covariance matrix and the Kalman Gain goes to values near zero as more measurements are made (large k). This means that the KF estimates become more reliable by the time. Therefore, new measurements will be less and less effective in the system.

However, if the actual error of the system changes over the time or if some unmodeled state is present, the KF will not be able to correct the system for large k , and the solution of the filter will diverge from the true values.

The KF assume that the system noise covariance matrix Q_{INS} are represented by a known Gaussian white noise w_g and w_a . If the process has only white noise, the measurement innovations δz_k^{e-} should also be a Gaussian zero-mean white noise. However, in reals

IMUs, the noise may present some coloured characteristic, which can cause the filter to have an apparent divergence or to achieve a suboptimal solution.

Acc. to [48], two methods for preventing divergence are available: the exponential data weighting and the fictitious process noise injection.

2.6.1. Exponential Weighted Kalman Filter

To prevent divergence of the KF for large k , the exponential data weighting can be used [36]. This method avoids the Kalman Gain to go to zero by applying an exponential weighting in the Kalman gain and the covariance matrix. Therefore, the filter will always consider new measurements.

2.6.1.1. Exponential Weighted KF

In the weighted KF, the error covariance matrix and the system noise covariance matrix can be written as weighted covariance, as follow:

$$\begin{aligned} R_k &= R \alpha^{-2(k+1)} \\ Q_k &= Q \alpha^{-2(k+1)} \end{aligned} \tag{2.62}$$

Where α is the weighted parameter. Considering $P_k^\alpha = P_k^- \alpha^{2k}$ and applying equation 2.62 to the ESKF model presented in section 2.5.2, the weighted ESKF may be given as follows (the complete derivation can be found in [49]):

$$\begin{aligned} P_k^\alpha &= (\alpha)^2 \Phi_{k-1} P_{k-1}^\alpha \Phi_{k-1}^T + Q_{INS_{k-1}} \\ K_k &= P_k^\alpha H_k^T \left(H_k P_k^\alpha H_k^T + \frac{R_k}{(\alpha)^2} \right)^{-1} \end{aligned} \tag{2.63}$$

In these equations, the parameter α is a constant value always ≥ 1 . It is easy to see that when $\alpha = 1$ the KF will work as a normal KF. When $\alpha > 1$, the Kalman Gain and the error

covariance matrix will be kept away from zero by the alpha parameter. The higher the alpha parameter, the greater the new measurements are considered by the filter.

Because the considered system is Loosely coupled INS/GNSS system, which takes advances on 3-dimensional position and velocity, it is proposed an original representation of the weighted parameter α as a matrix to allow weighting the position (α_p) and velocity (α_v) individually as followed:

$$\alpha_k = \begin{bmatrix} \alpha_{p_{3 \times 3}} & 0_{3 \times 3} \\ 0_{3 \times 3} & \alpha_{v_{3 \times 3}} \end{bmatrix} \quad (2.64)$$

Therefore, the weighted Kalman gain and the weighted error covariance matrix can be written as follows:

$$\begin{aligned} P_k^\alpha &\approx (I + H_k^T \alpha H_k)^2 \Phi_{k-1} P_{k-1}^\alpha \Phi_{k-1}^T + Q_{INS_{k-1}} \\ K_k &= P_k^\alpha H_k^T \left(H_k P_k^\alpha H_k^T + \frac{R_k}{(I + \alpha)^2} \right)^{-1} \end{aligned} \quad (2.65)$$

Where I is the identity matrix, in this case, $\alpha \geq 0$. Consequently, when $\alpha=0$, no weighing is applied in the system. For $\alpha > 0$, the system will give higher weighting for the recent measurement.

Appendix B provides a study on the use of a constant alpha and matrix alpha weighted parameter.

2.6.1.2. Exponential Weighted UKF

Similar to the weighted ESKF presented in section 2.6.1.1, the Weighted UKF can be obtained by applying the equations 2.62 to the UKF model shown in section 2.5.3.

However, in the case of the UKF, the state estimation error covariance matrix, the measurement innovations covariance, and the covariance between the state estimation and the measurement innovations, they all must be considered as weighted covariances.

Therefore, the weighted UKF, for the time-invariant model, can be given by:

$$\begin{aligned}
P_k^\alpha &= \alpha^2 \frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \\
K_k &= \frac{\left[\frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right]}{\frac{1}{2n} \sum_i^{2n} (\delta z_k^{-(i)} - \delta \hat{z}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T + \alpha^{-2} R_k}
\end{aligned} \tag{2.66}$$

Appendix A provides the full derivation for equations 2.66.

If the parameter α is a matrix, given by equation 2.64, the weighted UKF can be written as follows:

$$\begin{aligned}
P_k^\alpha &= (I + H_k^T \alpha H_k)^2 \frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \\
K_k &= \frac{\left[\frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right]}{\frac{1}{2n} \sum_i^{2n} (\delta z_k^{-(i)} - \delta \hat{z}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T + (I + \alpha)^2 R_k}
\end{aligned} \tag{2.67}$$

Where I is the identity matrix and $\alpha \geq 0$.

2.6.2. Fictitious Process Noise Injection

Fictitious process noise injection is a methodology that prevents the KF divergence by ensuring that in the model, all states are sufficiently excited by noise. By equations 2.42, 2.46 and 2.49, it is possible to get an approximation of the Kalman gain, when the predicted error covariance goes to zero:

$$K_k \approx \frac{Q_k}{Q_k + R_k} \tag{2.68}$$

This equation shows that to excite the system to KF consider new measurements, the Q_k must be increased to a large value by injection of fictitious process noise. Therefore,

another option for applying a correction parameter is in a way that will enlarge Q_k . However, if the value of Q_k is too large, it will result in a suboptimal behaviour of the filter. Therefore, the injection must be correctly chosen in a way that the error remains bounded. The noise injection can be done by adding a correction value to the process covariance matrix as following:

$$Q_k = Q_k + \beta Q_k \quad (2.69)$$

Where β is tuned to ensure that the process noise covariance matrix bounds the actual process noise.

2.7. Fuzzy Adaptive Kalman Filter

As stated before, the KF solution considers that the system noise is white noise, meaning that the noise is a Gaussian zero-mean random noise. However, in a real system, this assumption is not always true. The presence of coloured noise will make the system diverge from the optimal solution. Therefore, adaptive KF can be used to face this problem.

However, before presenting the Fuzzy Adaptive KF used in this project, a brief review of fuzzy logic and adaptive Kalman Filter theory is given in the following sections.

2.7.1. Fuzzy Logic

The fuzzy logic is used to making decisions with estimated values when the system is incomplete or with uncertainties. In contrast to the classical logic theory, where the states are defined as 0 or 1, the fuzzy logic theory uses degrees of membership that varies from 0 to 1. Also, frequently non-numeric values are used to facilitate the expression of rules and states.

The process for use Fuzzy logic can be described in three steps: a) fuzzification - fuzzify all input values into fuzzy membership functions, b) Fuzzy logic rule base - execute rules to compute the fuzzy output functions, and c) defuzzification - de-fuzzify the fuzzy output functions to get numerical values as output [50].

Fuzzification is the process where the inputs, numerical or crisp, are converted to fuzzy. By identify specific characteristics in the input values, it is possible to convert it to fuzzy values using membership functions. Therefore, this process may involve designate membership values for possible inputs.

The Fuzzy logic rule base interpreted the fuzzy inputs through a set of rules. This process aimed to understand the behaviour of the state of the system to provide the best solution.

Finally, the defuzzification transforms the fuzzy value in crisp outputs values, that can be read and used by the system.

2.7.2. Adaptive Kalman Filter

The measurement noise covariance matrix R and system noise covariance matrix Q is usually determined by the specifications of the sensor and the characteristics of the system. Also, it can be more accurately defined during the development phase of the system.

However, the R and Q may not be fully known in advance, or the optimum Kalman Filter tuning may vary over time due to some sensor characteristic or due to the changes in the expected trajectory geometry or dynamics profile. For both cases, an adaptive Kalman Filter may be used to correct the measurement and system noise covariances during the system operation.

Two main adaptive approaches, developed in the 70s, have been successfully used: the multiple model adaptive estimation (MMAE)[13] and the innovation adaptive estimation (IAE) [14]. They both share the same concept of using new statistical information obtained from the innovation to update the system and measurement noise covariance matrix in a systematic approach [25].

With the advent of artificial intelligence (AI), Neural Networks (NN) [27], [28], Fuzzy Logic (FL) [3], [30], and Genetic Algorithms (GA)[31][32] techniques have been used to adapt the covariances matrices to the real situation in a more precisely and efficiently way. The Fuzzy Logic highlights as a common technique used in adaptive controls to deal with nonlinear systems with uncertainties, which the errors can be modelled using a heuristic knowledge of the system, therefore, no mathematical model of the noise is needed.

2.7.3. Fuzzy Adaptive Kalman Filter Architecture

Usually, the Fuzzy Adaptive Kalman Filter (FAKF) consists in a standard KF with a feedback adaptation control, that adjust the covariances matrix Q, R, and/or the Kalman gain through the use of Fuzzy logic. Therefore, the differences between the implementations remain in three aspects: a) the chosen of the state and the definition of its membership functions that will be used by the FL, b) the definition of the base rules used by the FL, and c) how the output of the FL is applied to correct the Kalman filter.

2.7.3.1. Criteria for choosing the FL input

The states that will be observed by the FL must be chosen in a way that identifies when the KF is not optimal and/or when the system is diverging. Looking through the equations for the ESKF and UFK, it is possible to see that both KF approach will give the estimated states (\hat{x}_k^+) and the error covariance matrix (P_k^+).

The estimated states vector is given by the equation 2.41, and provide the error estimates for attitude, velocity, position, accelerometer bias and gyroscope bias. It gives the error of the INS system, that will be corrected as a feedback signal. In this case, it can be considered that if the predicted state presents a value bigger than the initial uncertain, given by the characteristics of the sensor, the system is sub-optimal.

Therefore, it can be considered that the system is not optimal or diverging when:

$$\hat{x}_k^+ \geq \sigma_{sensor} \quad (2.70)$$

where σ_{sensor} is the standard deviation of the sensor converted to the state of \hat{x}_k^+ .

However, this value can be too broad, once the KF can be not optimal with values much smaller than the expected by equation 2.70. Therefore, a study of the system performance is needed to determine the values that can identify sub-optimal behaviour.

To define if the system is not optimal or diverging by observing the error covariance matrix (P_k^+), two situations must be considered: the apparent divergence and the true divergence, as discussed in section 2.6. Figure 2-9 showed the difference in the error covariance values between an optimal, a true divergence and an apparent divergence KF.

When the system is truly diverging, it is easy to identify it by analyzing the rate of change of the error covariance matrix. However, it is not trivial to determine when the system is facing an apparent diverging. To do so, the expected behaviour of the system must be known. This can be done by fully characterizing the system during optimal operation, or by creating a parallel KF that has in the input the expected noise signal artificially created.

2.7.3.2. Membership Functions

After defined the inputs to be observed, they must be mapped into their respective fuzzy sets. The fuzzy sets describe how much an input value meets the criteria specified for the Fuzzy logic. This is done in terms of membership functions, which will map the input in a universe from zero to one, considering different shapes, chosen in a way that best describes the distribution of the data. Although there are some techniques to assign membership function to fuzzy sets, the intuition is usually used.

Figure 2-10 shows different shapes of membership functions that are usually chosen. However, the shape is not so important as it is the placement of the curves on the universe, the number of curves, and the curves overlapping [51].

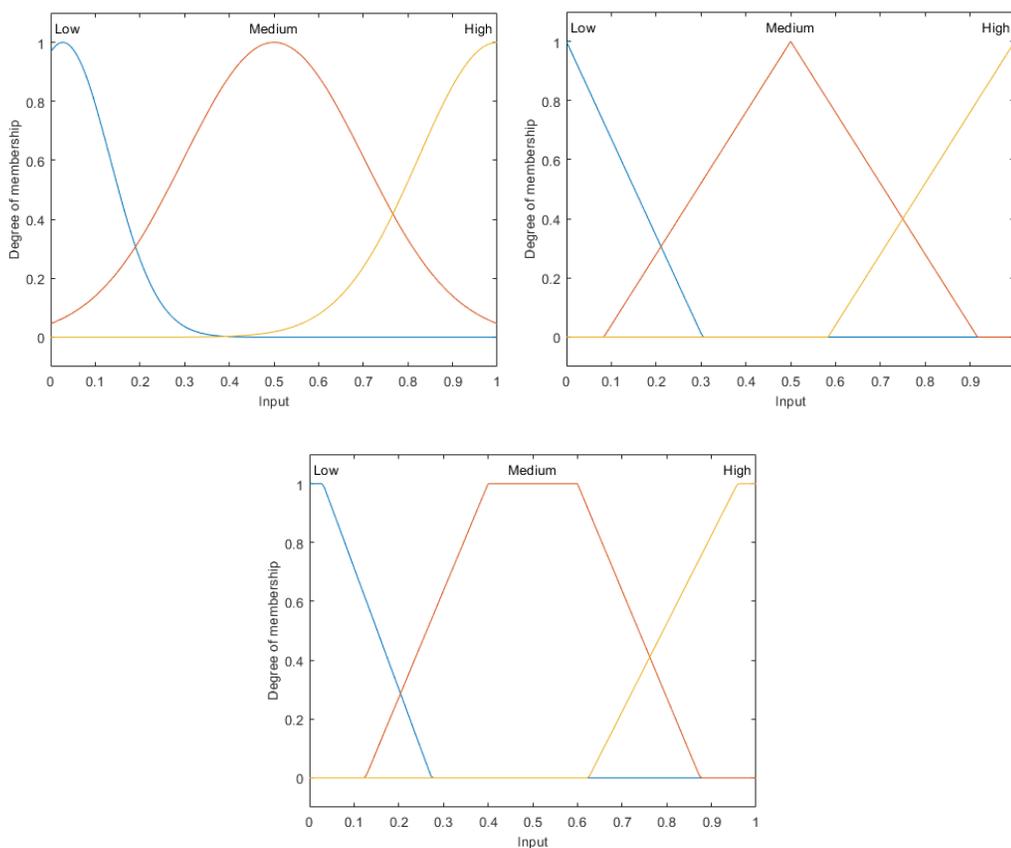


Figure 2-10 - Different shapes of Membership Function: Gaussian (left), Triangular (right), and Trapezoidal (down)

2.7.3.3. Criteria for base rules definition

The Fuzzy based rule uses a deductive form described as the IF-THEN rule base. It expresses an inference such that if one fact is known, it is possible to infer a conclusion.

Therefore, the generical base rule for a Fuzzy Logic system can be expressed as:

$$\text{IF premise (antecedent), THEN conclusion (consequence)} \quad (2.71)$$

Usually, it is necessary to obtain an overall conclusion from more than one premises. This process is known as aggregation, which can be done by a conjunctive or disjunctive system of rules. In conjunctive systems, the rules are connected by “AND” logical conjunction; in this case, the system must be jointly satisfied. In disjunctive systems, the rules are connected by “OR” logical conjunction; in this case, the system must be individually satisfied.

If the KF output \hat{x}_k^+ and P_k^+ are used as premises, the base rule will involve aggregation.

The type of aggregation used will be defined by analyzing the system and determine if the KF is not optimal when both premises are satisfied or when any of the two premises are satisfied.

2.7.3.4. Criteria for apply the FL output

If we consider that the divergences in the KF come from unmodeled noise or changes in the measurement (R) and system (Q) covariance noises, the use of FL output to correct these noise models is the straightforward solution. However, if the predicted error covariance, and hence the Kalman gain, are near zero, changes in the R and Q that does not bring the Kalman gain to a higher value, will cause a small effect in the system.

Acc. to Lewis et. al [48], there are two methodologies to prevent KF divergence: Fictitious process noise injection and exponential data weighting. The two methodologies were discussed in section 2.6.

For the exponential data weighting, the FL output can be used to regulate the weighting parameter - alpha. And, for the process noise injection, the FL output can be used to enlarge the system noise covariance matrix.

The approach used to correct the KF will be defined by analyzing the behaviour of the system when each of the two methodologies is used.

3. Methodology

This chapter provides the methodology used in this project and will be separated as follow:

Section 3.1 describes the INS model, and it is divided into three topics, as following: a) generation of the navigation profile with attitude, position, and velocity; b) raw gyroscope and accelerometer data creation, by the use of inverse kinematics; and c) determination of the navigation profile by using an INS dynamic model, having as input the raw gyroscope, the raw accelerometer, and a defined initial position, velocity, and attitude.

Section 3.2 shows the INS/GNSS integration using ESKF and UKF, and is divided in a) generation of the raw GNSS data; b) white noise and coloured noise addition to the raw IMU and GNSS; c) INS/GNSS integration using ESKF and UKF, including the system initialization and the definition of the system and measurement covariance matrix.

Finally, section 3.3 describes the development of the Fuzzy logic adaptive KF.

A general view of the proposed system is presented in Figure 3-1.

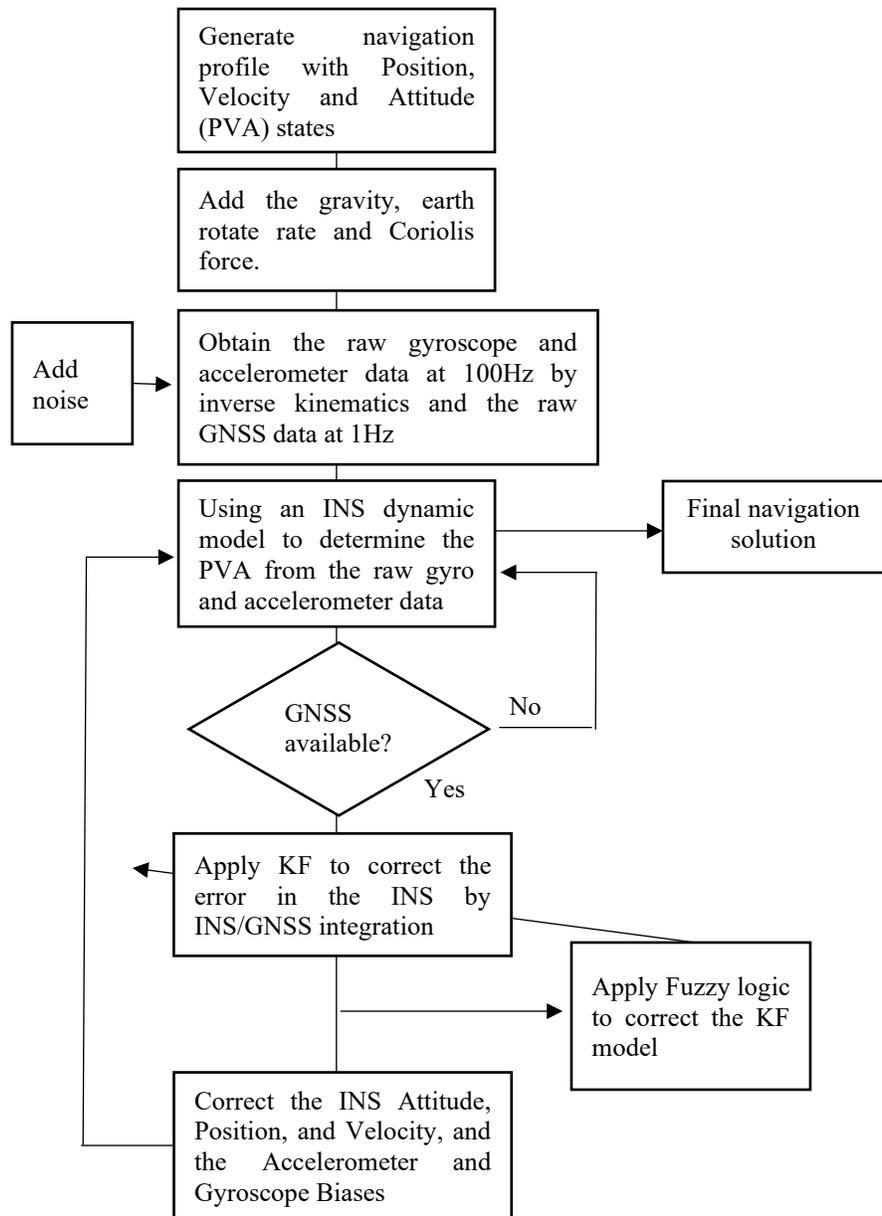


Figure 3-1 - General view of the proposed methodology

3.1. INS Model

Using MATLAB software with the aerospace toolbox, a motion profile of an object performing 3D navigation can be created. This motion profile may provide the Attitude, Position, and Velocity states, making the generation of the raw IMU data possible. With the raw IMU data, which is the 3-dimensional accelerometer and gyroscope sensor data, it is

possible to apply an INS dynamic model to compare the ground-true motion and the obtained motion aimed to validate de model INS. Figure 3-2 presents the general diagram for these steps.

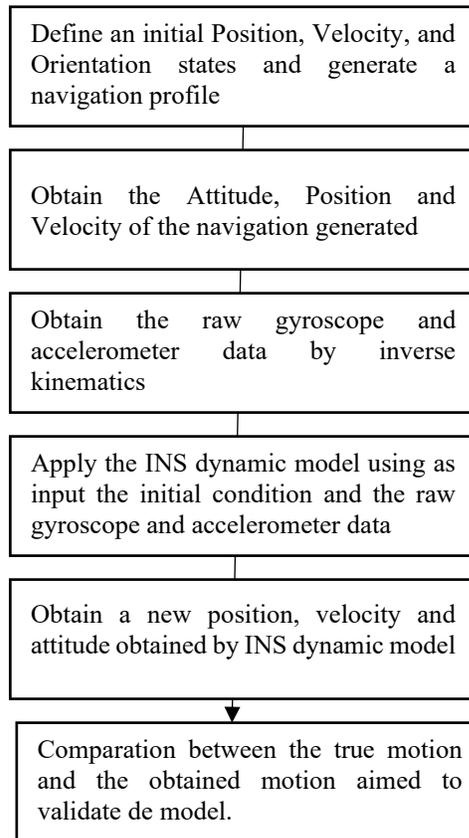


Figure 3-2 - General diagram for the INS Model

3.1.1. Navigation profile generation

Two navigation profiles of an object performing 3D navigation were created. The first one considered that no force is applied in the object for a navigation of 1000 seconds, so the accelerometer and gyroscope measurements should be equal to zero, maintain the initial attitude and velocity throw all time. This navigation profile was simulated aimed to check

the error between the true navigation and the calculated when no dynamic movement is present.

The initial attitude was defined as 5° in roll, -5° in pitch, and 0° in yaw. The initial velocity was defined as 1m/s in north, -1m/s the east and 0 m/s in down. The position, velocity and attitude profiles in the NED frame are presented in Figure 3-3.

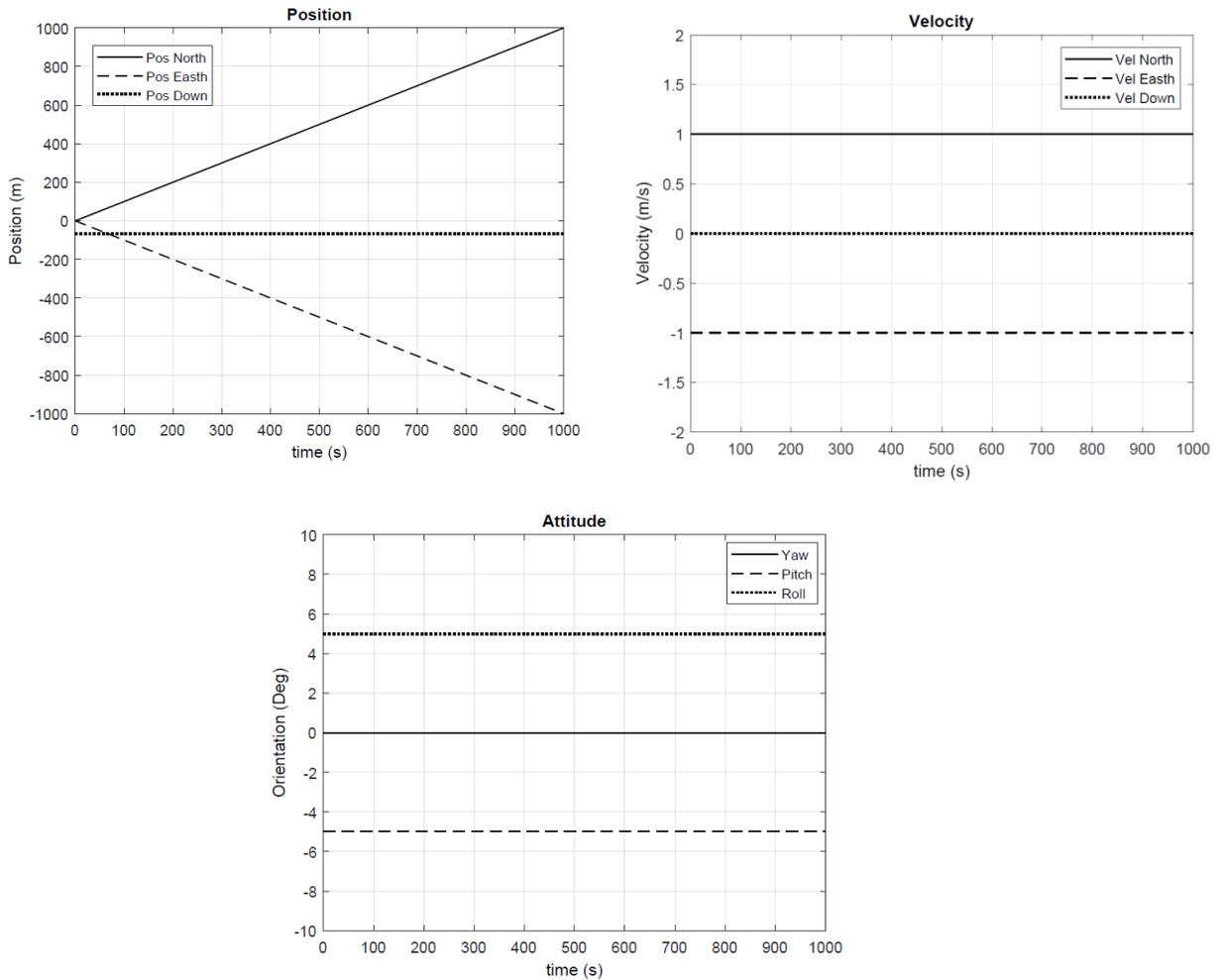


Figure 3-3 - Position, velocity and attitude profiles in NED frame for the first navigation profile.

For the second navigation profile, the movement was generated by the Matlab's function "kinematicTrajectory", considering a navigation of 3600 seconds with a radius of 15,000 m, a climb rate of 0.5 m/s, and an initial yaw and pitch of 90 and 10 degrees respectively [52]. The navigation was done by defining a constant angular velocity and a constant acceleration so that the movement describes a spiralling circular trajectory. The gravity, Earth rotation rate and transport rate were considered. The position, velocity and attitude profiles in the NED frame are presented in Figure 3-4.

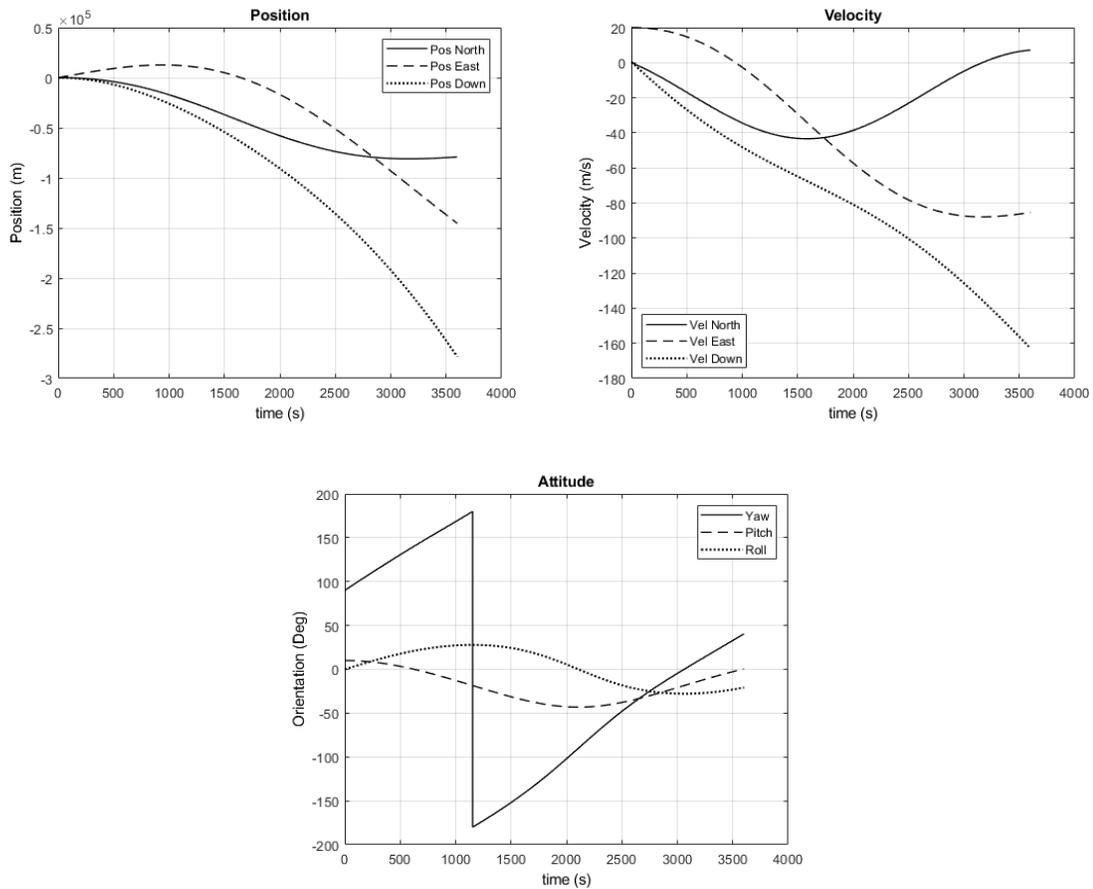


Figure 3-4 - Position, velocity, and attitude profiles in NED frame for the second navigation profile.

After obtained the navigation profile in the NED coordinate frame, an initial state in the ECEF frame was chosen, and the profile was transformed from NED to ECEF coordinate system following the methodology described in section 2.1.5.2. The initial position was defined as:

- a) Initial Latitude: 45.38° ;
- b) Initial Longitude: -75.70° ; and
- c) Initial altitude: 66 m (in relation to the ocean level);

The code used to generate the navigation profile and to perform the NED to ECEF transformation are available in Appendix E.2.

3.1.2. Gyroscope and accelerometer raw data generation

Acc. to Grove [1], the typical IMU sampling time is 100 Hz, so this value was considered in the model.

The raw gyroscope and accelerometer data were generated by applying the equations presented in section 2.2.2 in the generated data described previously. The raw gyroscope data was determined by calculating the rotation difference between two consecutive DCMs and then divide by sampling period. The raw accelerometer data was obtained by performing the derivative of the velocity in the local frame. Both the gyroscope and accelerometer data were compensated by the gravity, Coriolis effect, Earth rotation rate, and transport rate.

The code used to generate the raw gyroscope and accelerometer data is available in Appendix E.3.

3.1.3. *INS dynamic model*

The INS dynamic model uses the raw IMU data and an initial position, velocity, and attitude, to generate a 3D navigation profile with the position, velocity, and attitude. This process will be done using the approach and equations presented in section 2.2 and considering the same implementation rate as the IMU, which is 100Hz.

The code used to generate the navigation profile, using the raw gyroscope and accelerometer data, is available in Appendix F.1.

3.2. INS/GNSS integration

The INS/GNSS integration was done by the use of the software MATLAB and SIMULINK. The MATLAB was used to create the functions that describe the INS and the KF models. The SIMULINK was used to link the models, to define the inputs, and to show the outputs.

3.2.1. *GNSS raw data generation*

In this project, the raw GNSS data will be considered as the position and velocity generated in the ECEF coordinate frame, as described in section 3.1.1., with a Gaussian white noise added to simulate the GNSS uncertainties. Acc. to Grove [1], the typical GNSS sampling time is 1 Hz, value considered in the model.

For the GNSS noise model, it was considered that the estimated pseudo-range and pseudo-range rate errors will cause a standard deviation of $\sigma_p = 5m$ in the position and of $\sigma_v = 0.1m/s$ in the velocity. Therefore, a Gaussian white noise signal, considering these

variances, were added in the ground true position and velocity generated, to simulate the noisy GNSS raw data.

3.2.2. IMU Noise Model

In the present work, it will be considered a tactical-grade IMU, which the error characteristics presented in Table 3 for the gyroscope and in Table 4 for the accelerometer. The errors values were defined based on the commercial IMU Honeywell HG1700 [53]–[55].

Table 3 – Tactical-grade gyroscope error

Gyroscope error	Factory unit	Used value unit
Bias	$10^\circ/hr$	$5 \times 10^{-5} \text{ rad/s}$
Random-noise SD	$0.01^\circ/\sqrt{hr}$	$5 \times 10^{-6} \text{ rad}/\sqrt{s}$
Scale Factor	400 ppm	4×10^{-4}
Cross-Coupling	300 ppm	3×10^{-4}
G-dependent biases	$1^\circ/hr/g$	$0.5 \times 10^{-6} \text{ rad} - \text{sec}/m$

Table 4 - Tactical-grade accelerometer error

Accelerometer error	Factory unit	Used value unit
Bias	1 mg	$1 \times 10^{-2} \text{ m/s}^2$
Random-noise SD	$100 \mu\text{g}/\sqrt{\text{Hz}}$	$1 \times 10^{-3} \text{ m/s}^{1.5}$
Scale Factor	500 ppm	5×10^{-4}
Cross-Coupling	300 ppm	3×10^{-4}

The errors and noises were applied in the simulated IMU, following the model described in section 2.2.3.1 to create a new noisy sensor signal, which will be used in the INS solution.

Firstly, using equation 2.27, a system where w_g and w_a are Gaussian white random noise was created and tuned to provide an optimal KF estimation.

Secondly, a coloured noise was applied to the white noise w_g and w_a using a $1/f$ filter. This provided a $1/f$ flicker noise in the IMU sensors measurements. In this case, no other parameters were changed in the KF, which was tuned considering Gaussian white noise.

The SIMULINK model used to generate the noisy IMU raw data is available in Appendix G.1.

3.2.3. Loosely Coupled INS/GNSS system

To integrate the GNSS with the INS system, an Error Feedback KF was applied, considering a loosely coupled INS/GNSS system. The Error Feedback means that the estimated position errors, velocity errors, attitude errors, and any estimation error in the IMU sensor, such as biases, are feedback from the KF solution to the INS, aimed to correct the INS states.

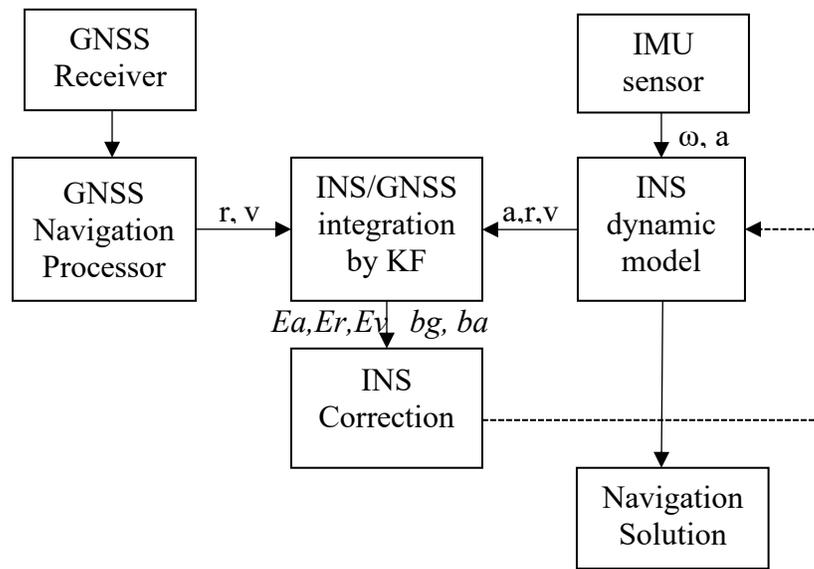


Figure 3-5 - Loosely coupled INS/GNSS system with the use of Error Feedback Extended Kalman Filter

The feedback correction occurs on each Kalman Filter iteration. The loosely coupled designation means that the KF will use both the GNSS position and velocity solution as the measurement inputs to perform the integration. Figure 3-5 shows the architecture considered.

For perform the GNSS/INS integration, two types of KF was considered, the ESKF as described in section 2.5.2, and the UKF as described in section 2.5.3. The MATLAB functions for both KF types are given in Appendix F.2 for the ESKF model and in Appendix F.3 for the UKF model.

The SIMULINK model for the INS/GNSS integrated system is given in Appendix G.2.

3.2.3.1. System initialization

In the initial attitude, position, and velocity, uncertainty must be considered to simulate the system properly. For the position and velocity, the initial uncertainty values can be given

by the GNSS measurements. Therefore, its uncertainties are proportional to the GNSS position and velocity standard deviation, as presented in section 3.2.1.

For defining the initial attitude, two processes are usually applied: a levelling process and a gyrocompass process. The levelling process initializes the roll and pitch attitudes by considering that during the initialization, the INS is stationary, so the accelerometers will measure only the gravity force, which can be considered in the down direction of the Earth's surface. The gyrocompass process initializes the yaw by considering that the only rotational movement that the gyroscope senses are the Earth rotation, which is in the z-direction of an ECEF frame. By measuring this rotation, the yaw can be determined.

Therefore, considering a tactical-grade IMU, it was considered a 1 mrad initial attitude error standard deviation.

The initial error covariance matrix P represents the uncertainties of the initial state for the attitude, position, and velocity, as described above, and the accelerometer and gyroscope biases described in section 3.2.2.

Therefore, the initial P is defined as follows:

$$P_0 = \begin{bmatrix} 1 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \times 10^{-5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \times 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \times 10^{-5} \end{bmatrix}^2$$

The SIMULINK model for the INS/GNSS system initialization is given in Appendix G.3.

3.2.3.2. System and Measurement Covariance Matrices

The ESKF and UKF were tuned to provide an optimal solution when w_g and w_a are Gaussian white random noise. This means that the system and the measurement covariance matrices were tuned to give the smaller navigation error.

The measurement covariance matrix was defined according to equation 2.45, where w_p and w_v are, respectively, the position and velocity GNSS variance. Considering that the noises are independent of each other and that the standard deviation of GNSS position was defined as $\sigma_p = 5m$ and of GNSS velocity as $\sigma_v = 0.1m/s$, the variance is just the square of the standard deviation. Therefore, the measurement covariance matrix for both ESKF and UKF was defined as:

$$R_k = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}^2$$

The system covariance matrix was defined according to equation 2.40, Where S_{gr} and S_{ar} are the gyro and accelerometer random noise PSD respectively; and the S_{gbd} and S_{abd} are the gyro and accelerometer bias variation PSD respectively.

The gyro and accelerometer random noise PSD are the square of the random noise SD, obtained by Table 3 and Table 4, multiply by dt . The gyro and accelerometer bias variation PSD were tuned to be the lower value possible to the KF presents the minimum error in the navigation solution. Therefore, the system covariance matrix values were defined as:

$$a) S_{gr} = (5 \times 10^{-6})^2 \text{ rad}^2 \text{ s}^{-1}$$

$$b) S_{ar} = (1 \times 10^{-3})^2 \text{ m}^2 \text{ s}^{-3}$$

$$c) S_{gbd} = (1 \times 10^{-8})^2 \text{ rad}^2 \text{ s}^{-3}$$

$$d) S_{abd} = (1 \times 10^{-5})^2 \text{ m}^2 \text{ s}^{-5}$$

Once errors such as the cross-coupling, scale factor, and g-dependent were introduced in the system, but not modelled as states in the KF, their effects may be compensated by increasing the gyro and accelerometer bias variation PSD. Therefore, to maintain KF stability, these errors approximations must over bound their impact on the KF states.

3.3. Fuzzy Logic

Before defining the Fuzzy Logic implantation approach, some analyses were done in the system with coloured noise aimed to determine the KF states that will be observed by the FL and the methodology to correct the KF. These analyses are presented in Appendix B and Appendix C.

In Appendix B it is demonstrated that the application of an exponential weighted parameter in a matrix form, which allows weighting the position (α_p) and velocity (α_v) individually while not affecting the other states, brings the best solution for the FL weighted KF implementation.

In Appendix C, it is demonstrated that for the fuzzification, the best KF states do be observed are the error covariance and the residuals. The error covariance observation will indicate if the KF is in steady-state or not, and the residuals will show the error peaks in the determination of the position and velocity states.

The study for choosing the methodology used by the FL to correct the KF are presented in Appendix C.2. It is showed that the exponentially weighted approach achieves better results before the system reaches the steady-state - when the error covariance is high, and the fictitious noise injection approach showed better results when the system is in steady-state.

Based on these analyses, a novel FLAC implementation methodology was defined as a hybrid solution where the Fuzzy Logic observes the error covariance and the residuals from the KF solution, applying an exponentially weighted correction when the error covariance matrix is high, and a process noise injection when the system is in steady-state, arise as the best approach.

3.3.1. Fuzzy Logic Implementation

The fuzzy logic is used to adjust an exponential weighted value α , and a process noise injection β in the KF, as discussed in section 2.6. This means that a Fuzzy logic adaptive system (FLAS) is used to adjust the noise strengths in the KF model. Figure 3-6 shows the architecture considered.

The exponential weighted value α is applied in the KF solution as described in section 2.6.1, and the process noise injection β is used in the KF solution as described in section 2.6.2.

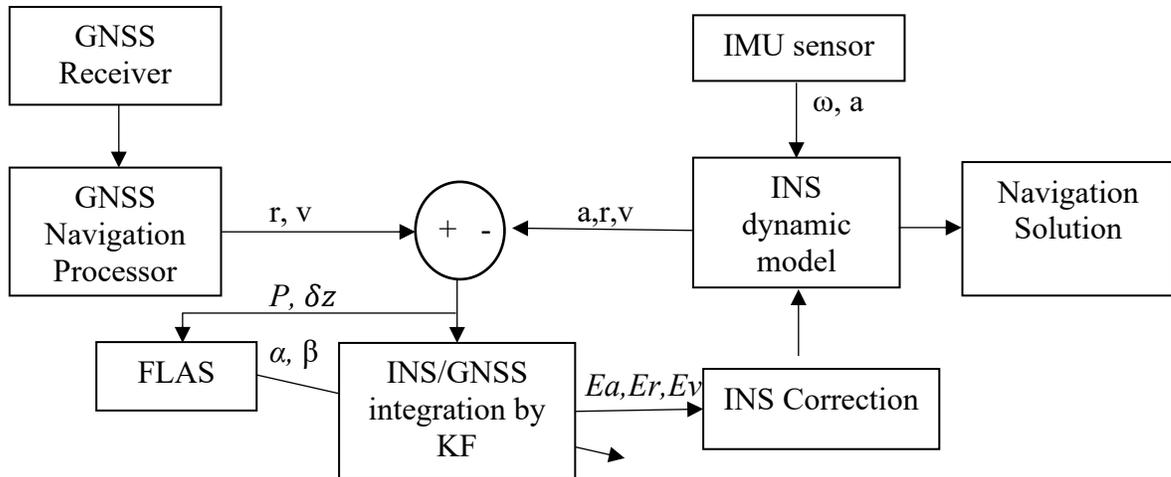


Figure 3-6 - FLAS applied in the Loosely Coupled GNSS/INS System

For the Fuzzy logic, it was considered two inputs, the residuals (δz) and the covariance of residuals matrix (P) using the conjunction rule “AND”, two outputs (α and β), and nine rules as below:

- 1) If P is small and δz is small, then α is zero and β is small;
- 2) If P is small and δz is medium, then α is zero and β is large;
- 3) If P is small and δz is large, then α is zero and β is large;
- 4) If P is medium and δz is small, then α is small and β is small;
- 5) If P is medium and δz is medium, then α is small and β is medium;
- 6) If P is medium and δz is large, then α is medium and β is large;
- 7) If P is large and δz is small, then α is large and β is zero;
- 8) If P is large and δz is medium, then α is medium and β is zero; and
- 9) If P is large and δz is large, then α is small and β is small.

The Fuzzy Logic will be applied individually for each measurement state available, which is 6 in the considered system – 3 for position and 3 for velocity. The membership functions for the input and output variable are present in Table 5 and Table 6, respectively.

For Table 6, the values are the same for the correction in the position and the velocity.

The SIMULINK model for the FL implementation is given in Appendix G.4.

Table 5 – Membership functions for input variables

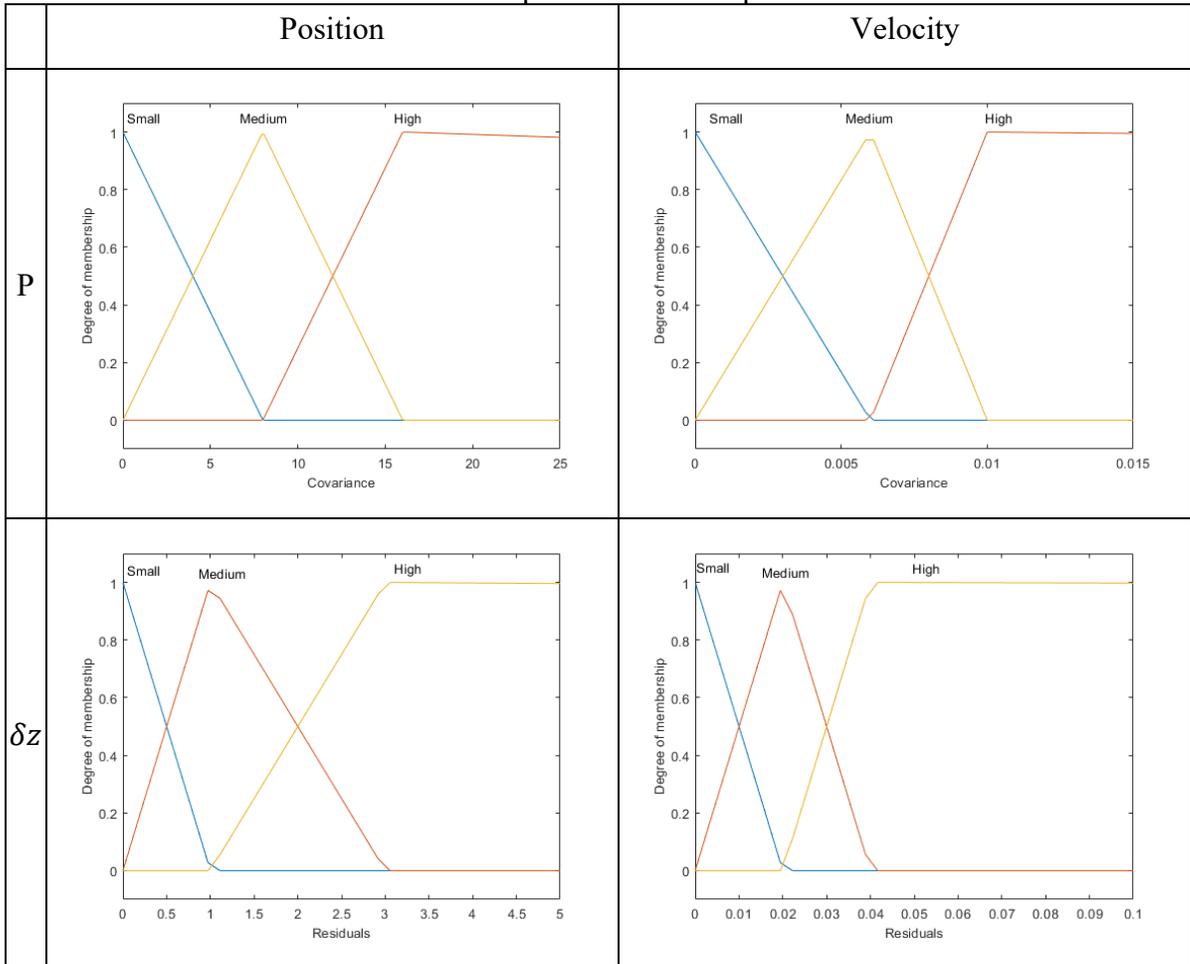
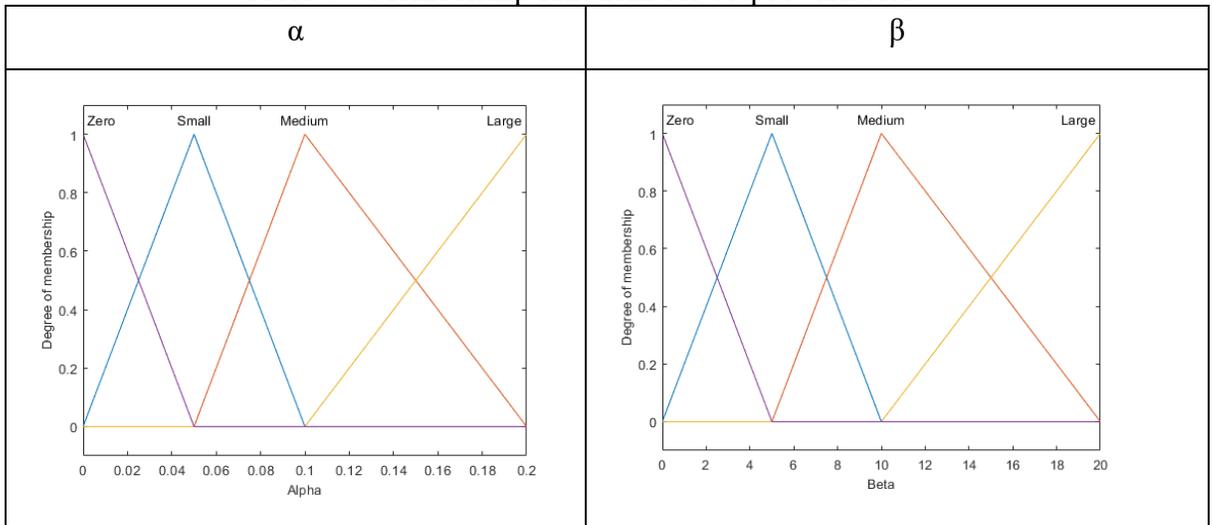


Table 6 – Membership functions for output variables



4. Simulation Results and Discussions

This chapter provides the simulation results and is divided as follows:

Section 4.1 describes the INS stand-alone results for the noise-free IMU, showing the results for two situations. The first one considered that no force is applied in the object for navigation of 1000 seconds. And the second, considering a realistic case of o object performing a navigation over 3600 seconds.

Section 4.2 shows the noise IMU and GNSS, and its effect in the error between the true ground navigation and the INS and GNSS alone solution.

Section 4.3 shows the results for the INS/GNSS integration using ESKF for two situations: system with white noise and system with coloured noise.

Section 4.4 shows the results for the INS/GNSS integration using UKF for two situations: system with white noise and system with coloured noise.

Section 4.5 describes the results for the Fuzzy logic adaptive KF.

And section 4.6 makes a comparison between the ESKF and UKF.

4.1. INS Simulation

To check the INS dynamic model, the navigation profile was first simulated, considering a noise-free IMU. The error measurements between the true navigation and the INS navigation solution are presented in Figure 4-1 for the first navigation generated - considered that no force is applied in the object, and in Figure 4-2 for the second navigation – the navigation profile used for the KF implementation.

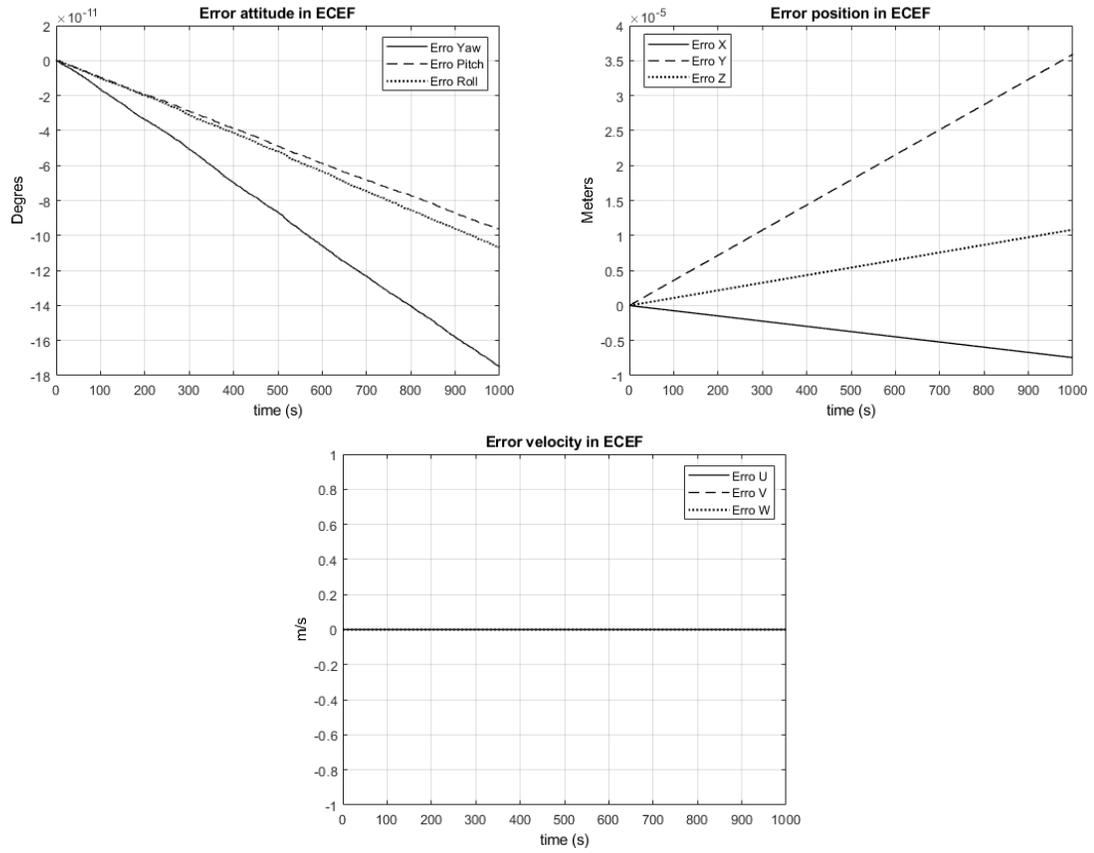


Figure 4-1 - Error between the ground-true and the INS solution for the navigation with no force applied

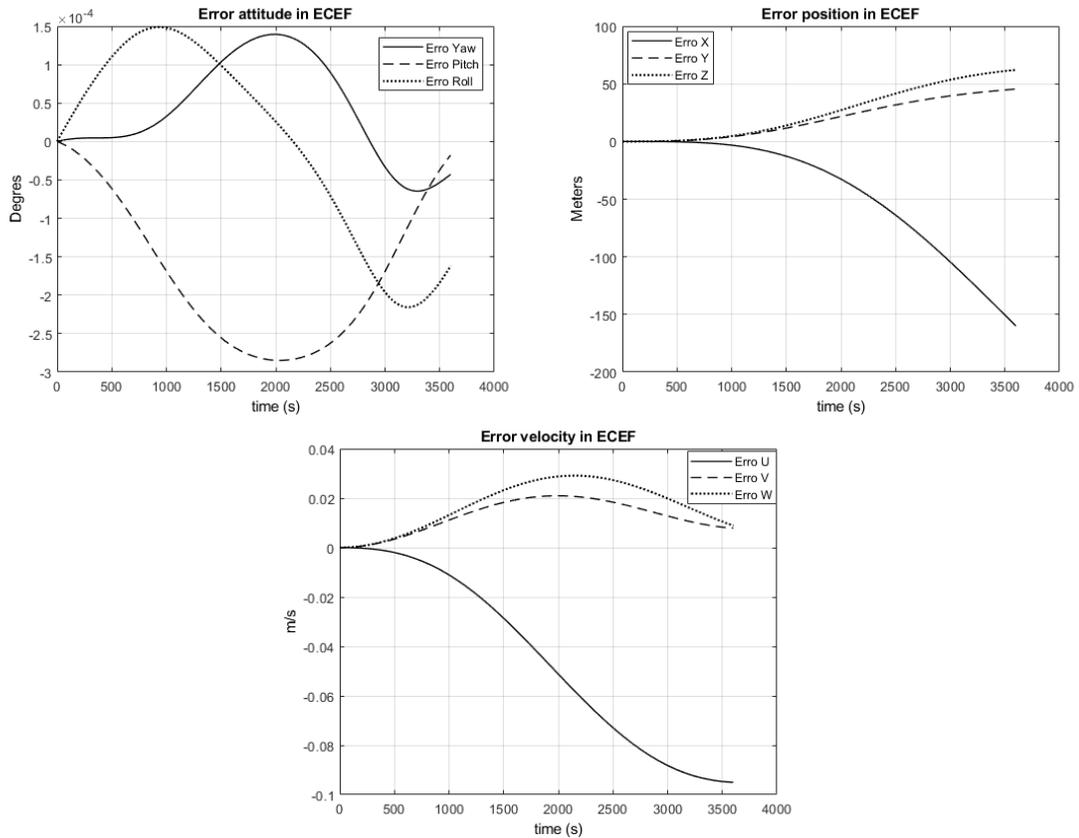


Figure 4-2 - Error between the ground-true navigation and the INS navigation solution for the realistic navigation generated

The results show a good coupling between the ground-true and INS solution for the attitude, velocity, and position with a small drift between the true and the calculated navigation.

The errors between the ground-true and the INS solution are related to the motion profile and errors that are introduced and propagated by the numerical derivations and integrations applied in the INS dynamic model, once discrete-time is considered. Therefore, it is expected that the errors become more significant in a navigation showing a high dynamic profile. This can be verified by comparing the Figure 4-1 and Figure 4-2, in which the navigation profile with no acceleration and angular velocity - first navigation, shows a

smaller error in the attitude, velocity, and position when compared with the second navigation profile, where acceleration and angular velocity are present.

4.2. Noisy IMU and GNSS Data

Following the methodology described in section 3.2.2, noises and bias were added in the IMU. Figure 4-3 presents the noisy and the noise-free y-axis accelerometer, and Figure 4-4 presents the noisy and the noise-free y-axis gyroscope. By these figures, it is possible to see the effect into the sensors signals by the introduction of noises.

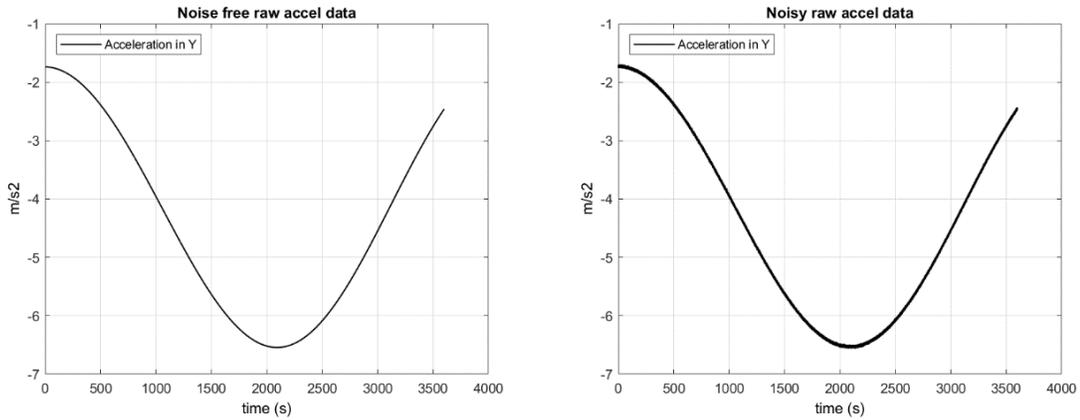


Figure 4-3 - Noise-free (left) and noise (right) y-axis accelerometer data

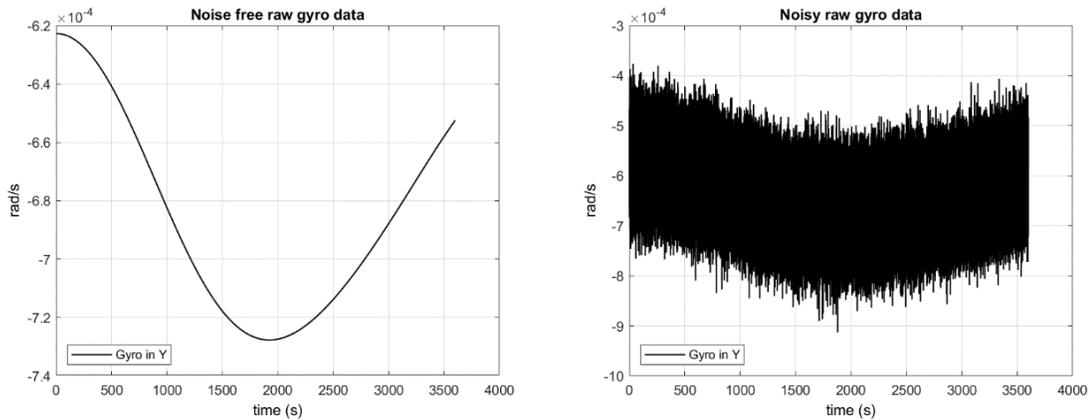


Figure 4-4 - Noise-free (left) and noisy (right) y-axis gyroscope data

Two identical systems were created, in which the only difference between them is that in the first one only Gaussian white noise was added, and, in the second system, coloured noise was considered in the place of the white noise. Figure 4-5 shows the white and coloured noise variance over the time for the y-axis gyroscope considering two situations: the firsts 20s and the full motion of 3600s. By this figure, it is possible to observe that the coloured noise will present a higher variance, which is expected considering the change in the mean from the zero value – white noise, to other values – coloured noise.

It is also possible to see that the variance for the coloured noise will rapidly go to an elevated level. Considering that the IMU is providing measurements at a frequency of 100Hz, it will take around 50 measures (0.5 seconds) for the variance of the coloured noise be higher than for the white noise. Furthermore, it is also noteworthy that the variance of the coloured noise will be changing over the time, and the white noise, on the other hand, will become constant after a brief period of time.

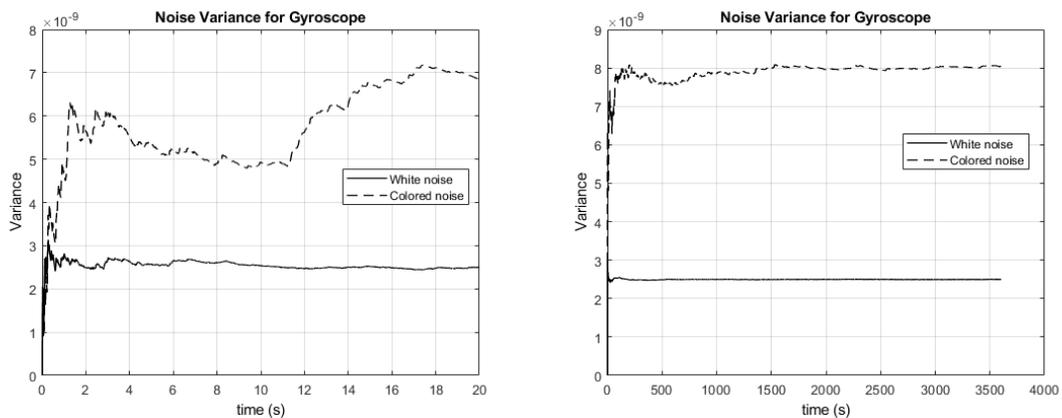


Figure 4-5 - White and coloured noise variance over the time for the gyroscope considering the firsts 20s (left) and the full motion of 3600s (right)

After applying the noise and bias into the IMU, following the methodology described in section 3.2.2, a new INS was simulated aimed to verify the influence of these small errors in the final attitude, position and velocity determination.

Figure 4-6 shows the error between the true-ground navigation and the INS navigation solution obtained by the noisy sensors considering the presence of gaussian white noise.

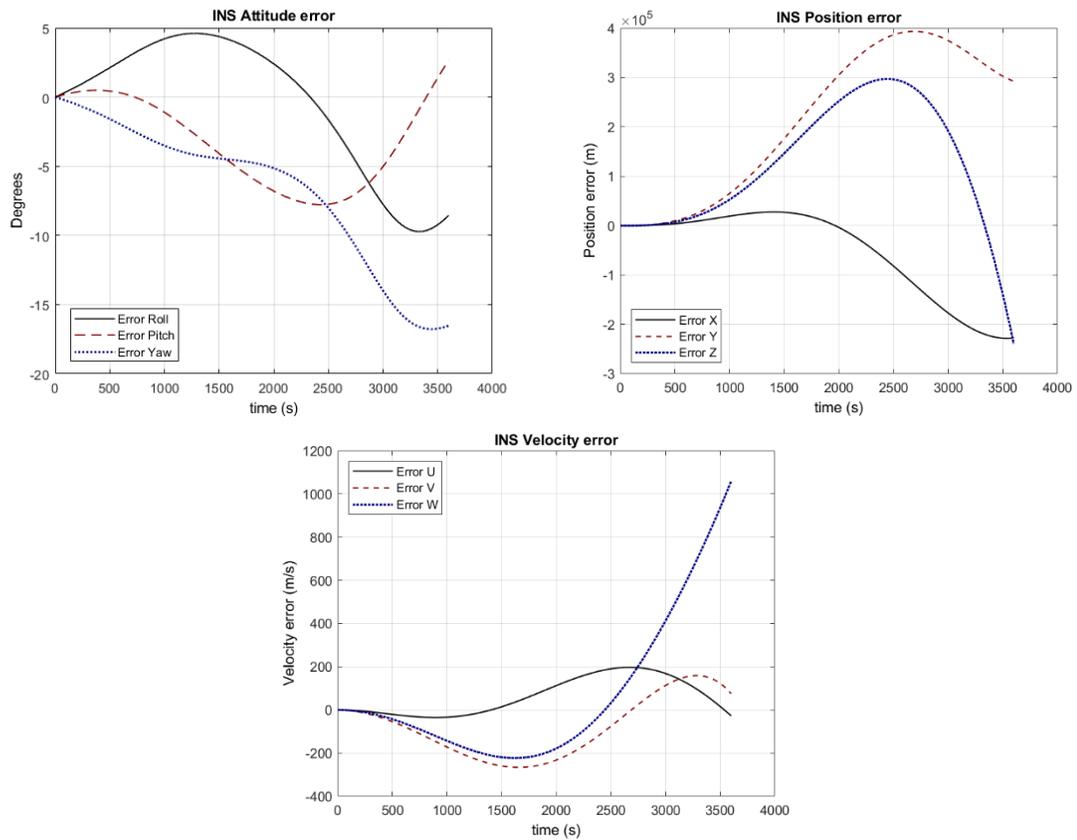


Figure 4-6 – Attitude, Position, and Velocity errors for the INS-alone solution for a noisy IMU

When Figure 4-6 is compared against Figure 4-2, it is possible to see the substantial effect caused in the INS-alone navigation solution by the small errors introduced in the IMU. This occurs because the INS suffer degradation in long-term navigation as the errors

presented in the sensor are accumulated over time and integrated through the navigation equations.

To add noise to the GNSS solution, the methodology described in section 3.2.1 was considered. This led to a final solution where the GNSS position and velocity are always following the ground-true position and velocity with a standard deviation. Figure 4-7 shows the error between the true-ground navigation and the GNSS-alone navigation solution obtained by the noisy sensors.

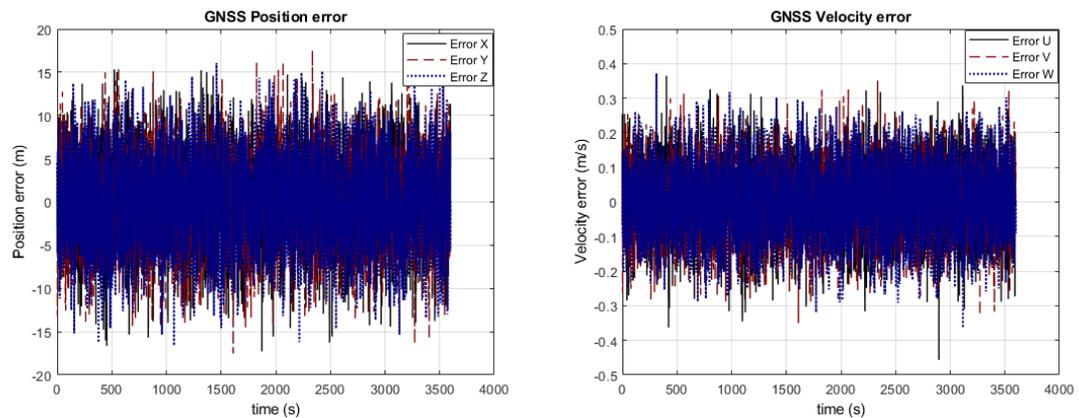


Figure 4-7 - Position and Velocity errors for the GNSS-alone solution for a noise GNSS

The INS show a low short-term noise, however, suffer degradation in long-term navigation. On the other hand, the GNSS provides good accuracy for long-term navigation. However, its short-term errors are high, and the standard GNSS does not provide attitude.

Therefore, the fusion of these technologies can benefit both, providing a navigation solution with high accuracy in long- and short-term.

4.3. ESKF based method

This section is intended to show the ESKF based method. First, the results for the GNSS/INS integration considering both the IMU with white and coloured noise are presented. And secondly, the ESKF states outputs are compared aimed to identify the states that best reflect the divergences caused by the coloured noise introduction.

The ESKF was modelled considering the methodology described in sections 2.5.2 and 3.2.3, using the navigation profile shown in Figure 3-4.

4.3.1. White and Coloured Noise ESKF Solution

Figure 4-8 present the attitude error between the true-ground and the INS/GNSS integration using ESKF with white and coloured noise.

Figure 4-9 present the position error between the true-ground and the INS/GNSS integration using ESKF with white and coloured noise.

Figure 4-10 present the velocity error between the true-ground and the INS/GNSS integration using ESKF with white and coloured noise.

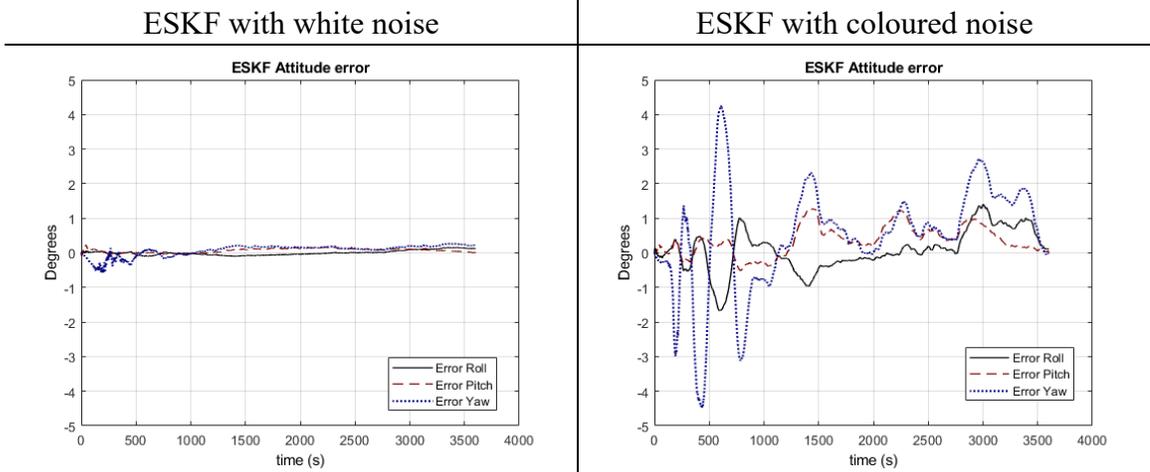


Figure 4-8 - Attitude error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise

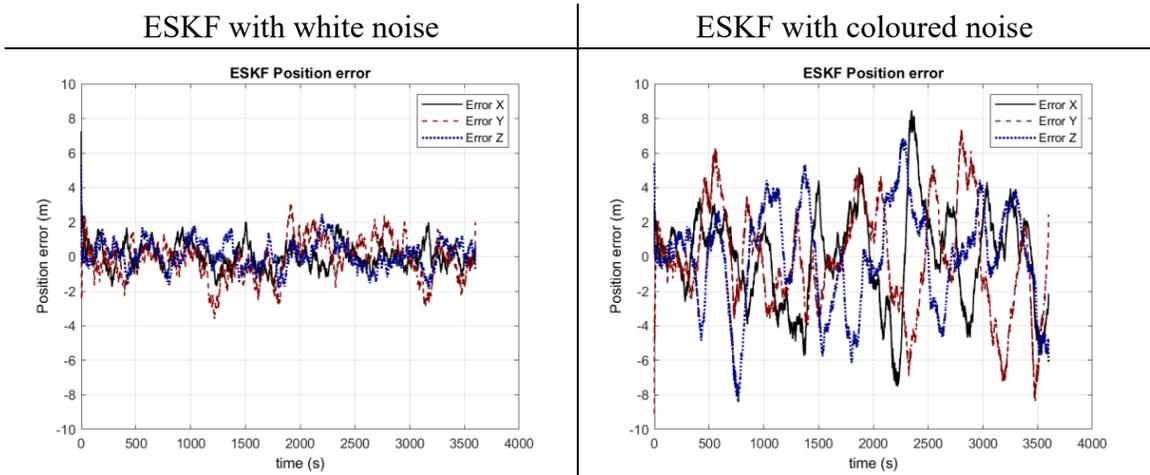


Figure 4-9 - Position error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise

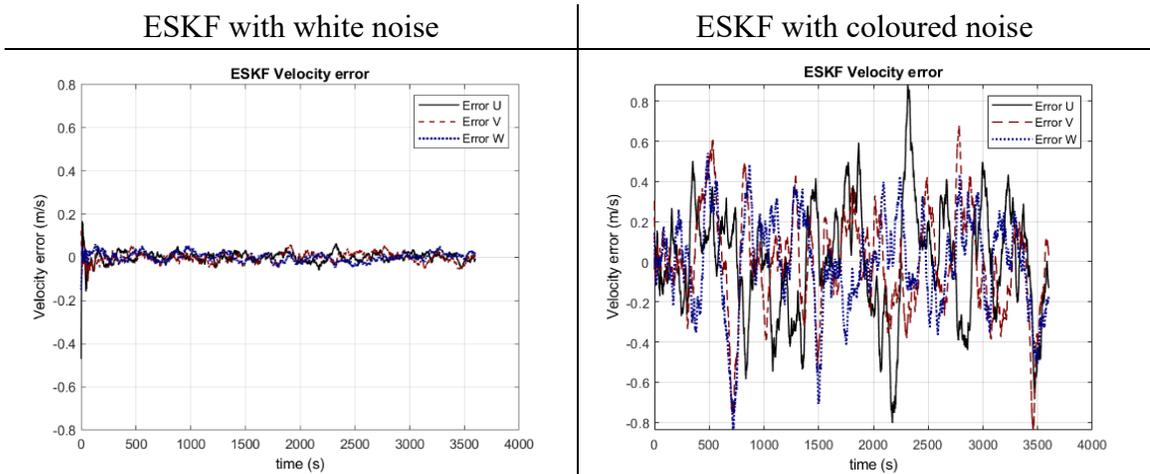


Figure 4-10 - Velocity error between the true-ground and the INS/GNSS integration using ESKF with white (left) and coloured (right) noise

By the results present in Figure 4-8 through Figure 4-10, it is possible to confirm the improvement in the attitude, position, and velocity determination using ESKF rather than the INS or GNSS alone solution, which the results were presented in section 4.2.

For the attitude, although the ESKF solution shows a high error in the beginner, the tendency is that in long navigations, the errors are maintained at a low level. The INS-alone shows a smaller attitude error in the beginner of the navigation. However, the error value demonstrates a tendency to grow. Therefore, the accumulated error in the INS-alone attitude is in the range of 17° after 3600 seconds of navigation. In the ESKF solution, the final attitude error remains between $\pm 0.2^\circ$ for the system with white noise and between $\pm 3^\circ$ for the system with coloured noise after the system achieves the stability.

For the position, the accumulated error in the INS-alone position is in the range of 5×10^5 m after 3600 seconds of navigation; the GNSS-alone solution shows a deviation between ± 15 m; and in the ESKF solution, the final position error remains between ± 3 m for the system with white noise and between ± 8 m for the system with coloured noise.

For the velocity, the accumulated error in the INS-alone are in the range of 1500 m/s after 3600 seconds of navigation; the GNSS-alone solution shows a deviation between ± 0.4 m/s; and in the ESKF solution, the final velocity error is between ± 0.06 m/s for the system with white noise and between ± 0.8 m/s for the system with coloured noise.

The results also show the effect in the ESKF solution when coloured noise is present in the IMU. Although the introduction of this noise brings small changes in the INS-alone solution, the coloured noise makes the ESKF system achieve a solution that converges to

a large bound then the solution reached when only Gaussian white noise is present. In this case, the states that benefited from the ESKF solution are the attitude and the position, once the ESKF solution for the velocity is worst then the GNSS-alone solution.

The bias correction for the gyroscope and accelerometer are presented in Figure 4-11 and Figure 4-12, respectively, using ESKF with white and coloured noise.

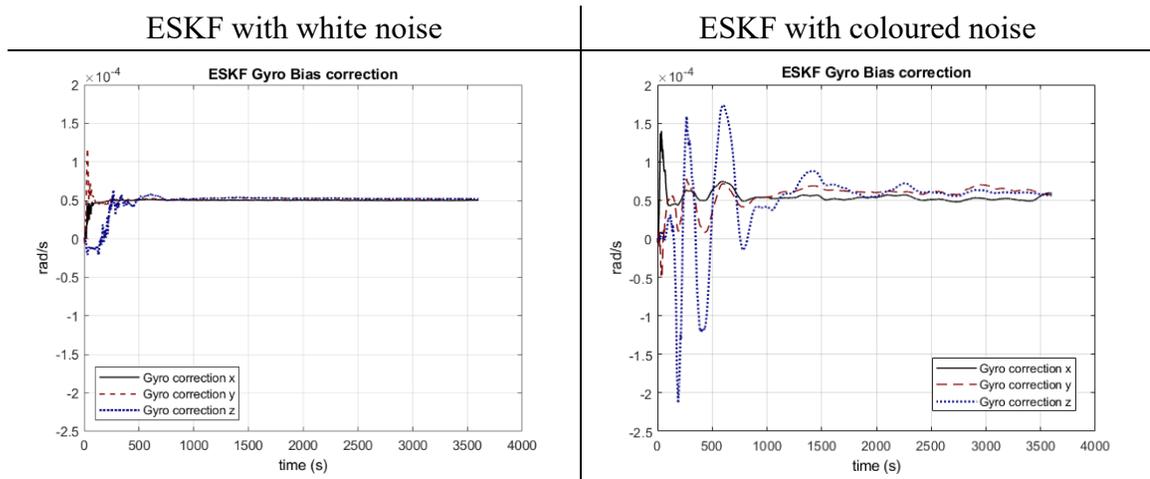


Figure 4-11 – Gyroscope bias correction given by the ESKF with white (left) and coloured (right) noise

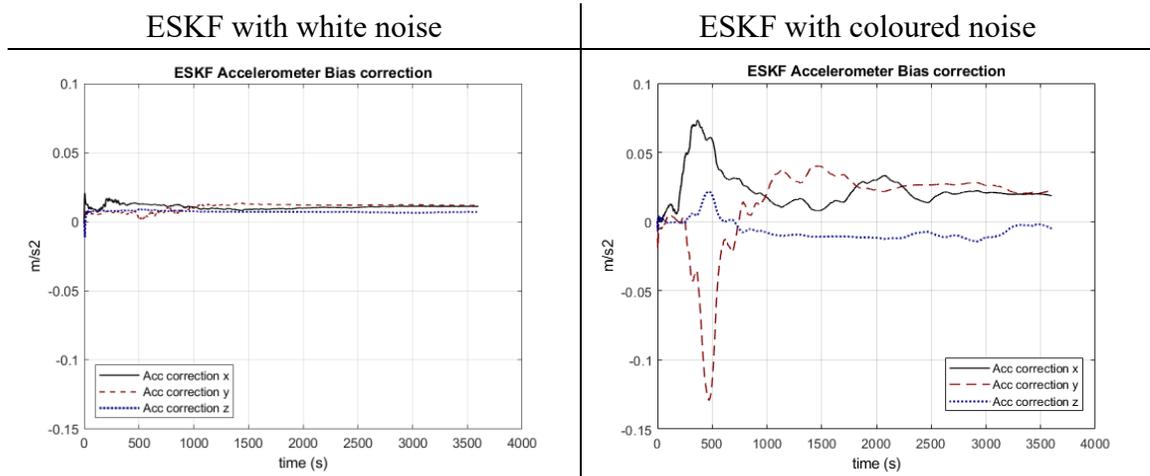


Figure 4-12 - Accelerometer bias correction given by the ESKF with white (left) and coloured (right) noise

Figure 4-11 shows that both ESKF with white and coloured noise could predict correctly the bias of $5 \times 10^{-5} \text{ rad/s}$ introduced in the gyroscope. However, when coloured noise is

present, the system faces a higher instability at the initialization and needs more time to achieve the correct bias value.

For the accelerometer bias correction, it is possible to see that for the ESKF with white noise the system can get close to the correct bias of $1 \times 10^{-2} \text{ m/s}^2$. However, when coloured noise is present, the system faces a higher instability and achieve a wrong bias prediction for the z-axis. According to Grove [1], because of the high effect of the gravity in the accelerometer measurements, its effects can be absorbed by the z-axis accelerometer bias states leading the KF to determine a wrong bias value.

4.3.2. *White and Coloured Noise ESKF states*

The variances of the state estimates are given by the diagonal elements of the error covariance matrix.

Figure 4-13 presents the *a posteriori* variance for Attitude, Position, and Velocity states, using ESKF with white and coloured noise.

Figure 4-14 presents the *a posteriori* variance for the gyroscope and accelerometer bias states, using ESKF with white and coloured noise.

The results show that the variance converges for all states. This indicates that the noises are corrected modelled. Comparing the system with white noise against the system with coloured noise, no apparent divergence is found when the coloured noise is introduced. The only effect observed is a small instability in the firsts 1000 seconds for the velocity and accelerometer bias error covariance, in the system with coloured noise.

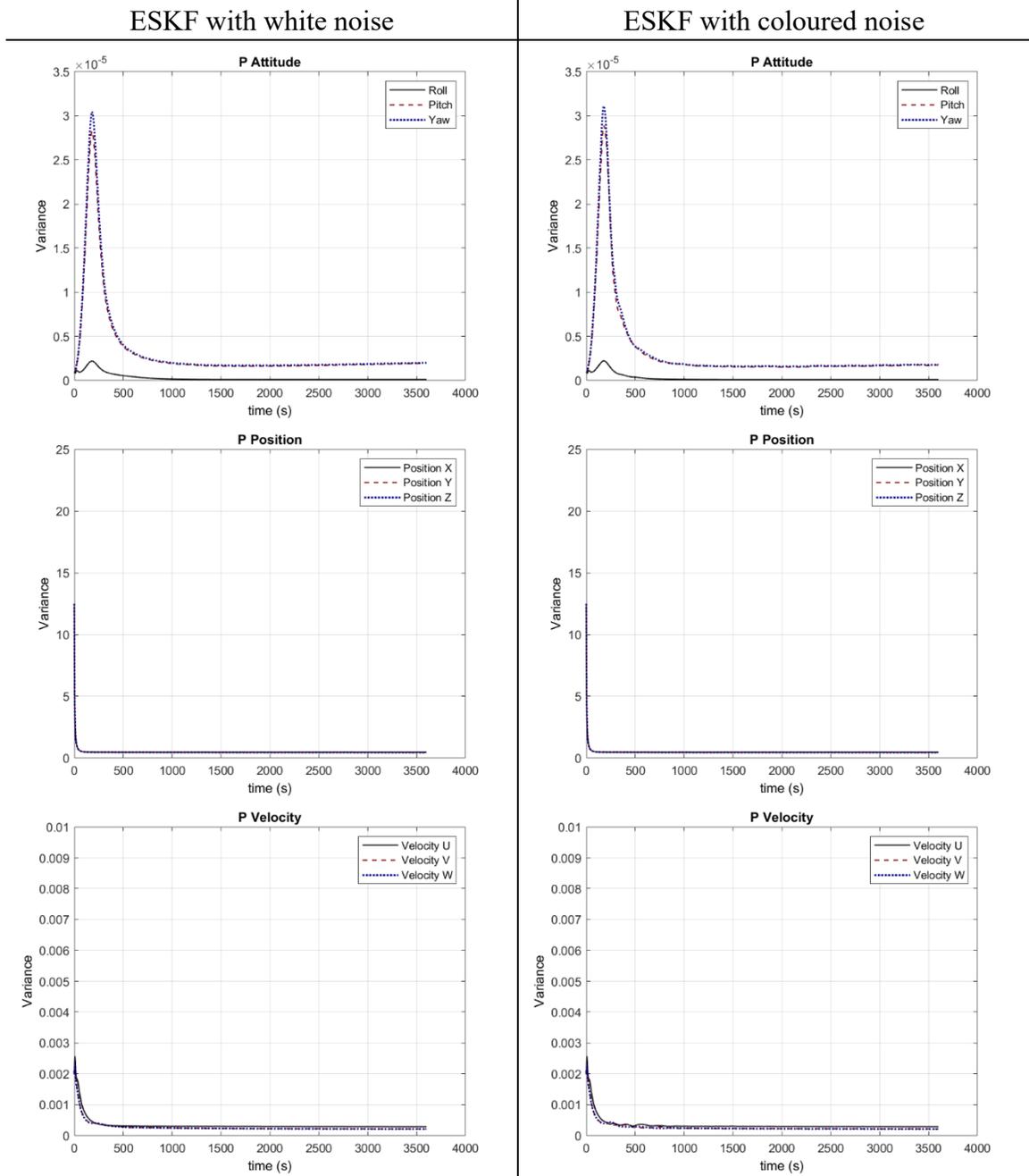


Figure 4-13 – Variance for Attitude, Position, and Velocity states considering the ESKF with white (left) and coloured (right) noise

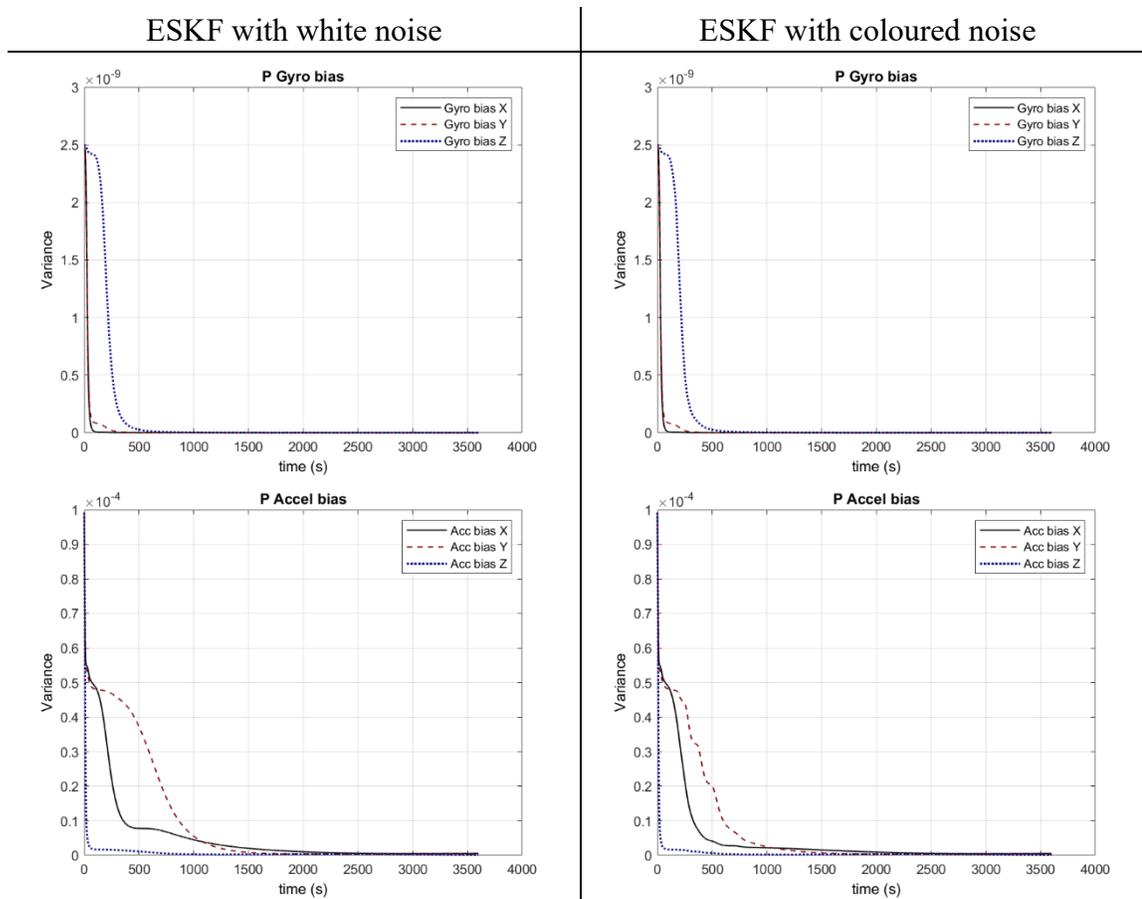


Figure 4-14 - Variance for the gyroscope and accelerometer bias states considering the ESKF with white (left) and coloured (right) noise

Comparing the variances of the states estimates to the true-ground error, presented in section 4.3.1, it is possible to infer that the noise introduced in the system, by the unmodeled coloured noise, is enough to lead the system to navigation solution with a higher error bound. However, it is not enough to cause any kind of KF divergence in the error covariance.

As a coloured noise is a change of the noise PSD over time, probably the system was able to achieve stability before the noise deviates from the zero mean enough to make the system diverge.

Figure 4-15 present the residuals for Attitude, Position, and Velocity, using ESKF with white and coloured noise. Figure 4-16 presents the residuals for the gyroscope and accelerometer bias, using ESKF with white and coloured noise.

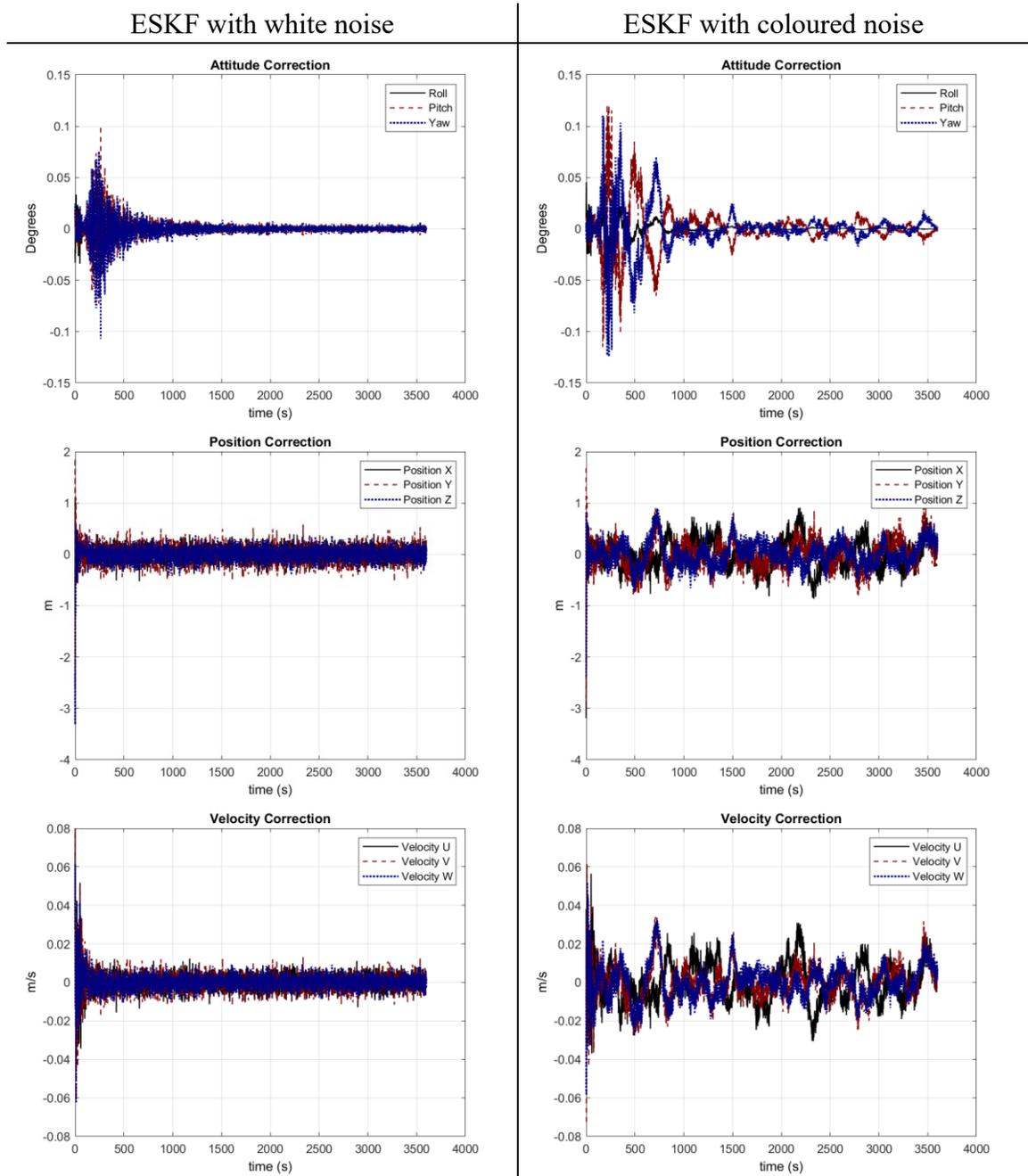


Figure 4-15 – KF residuals for Attitude, Position, and Velocity considering the ESKF with white (left) and coloured (right) noise

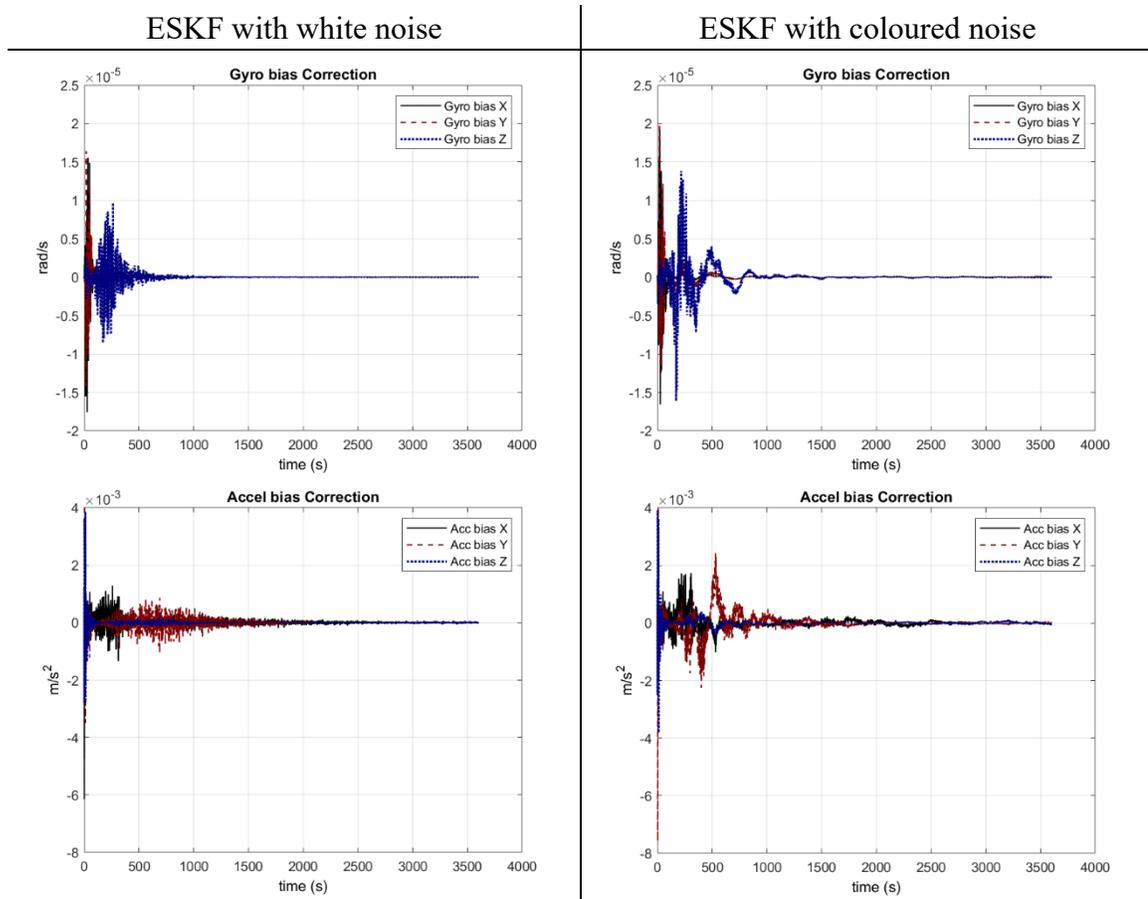


Figure 4-16 - KF residuals for gyroscope and accelerometer bias considering the ESKF with white (left) and coloured (right) noise

The residual represents the errors remaining in the system after the corrections done by the Kalman filter. The state vector residual is given by the difference between the true state vector and the Kalman filter state estimates. In an error-state implementation, the true state vector is always zero due to the feedback corrections. Therefore, the residuals are obtained simply by reversing the sign of the KF state estimates.

The residuals show a significant difference between the system with white and with coloured noises. The residuals values for the system with the coloured noise is in the order of twice bigger than the values obtained for the system with white noise. These results

show that the KF is mistaken by the coloured noise, making the KF correct the INS excessively.

When the Kalman filter is designed to the optimal operation with white noise, a decrease in the error covariance will be accompanied by a reduction in the corresponding state residual. However, when coloured noise is introduced, it can result in an error covariance smaller than the corresponding state residuals or even in a growing residual, causing the KF divergence.

Figure 4-17 presents the Kalman gain for the position and velocity states, considering the ESKF with white and coloured noise.

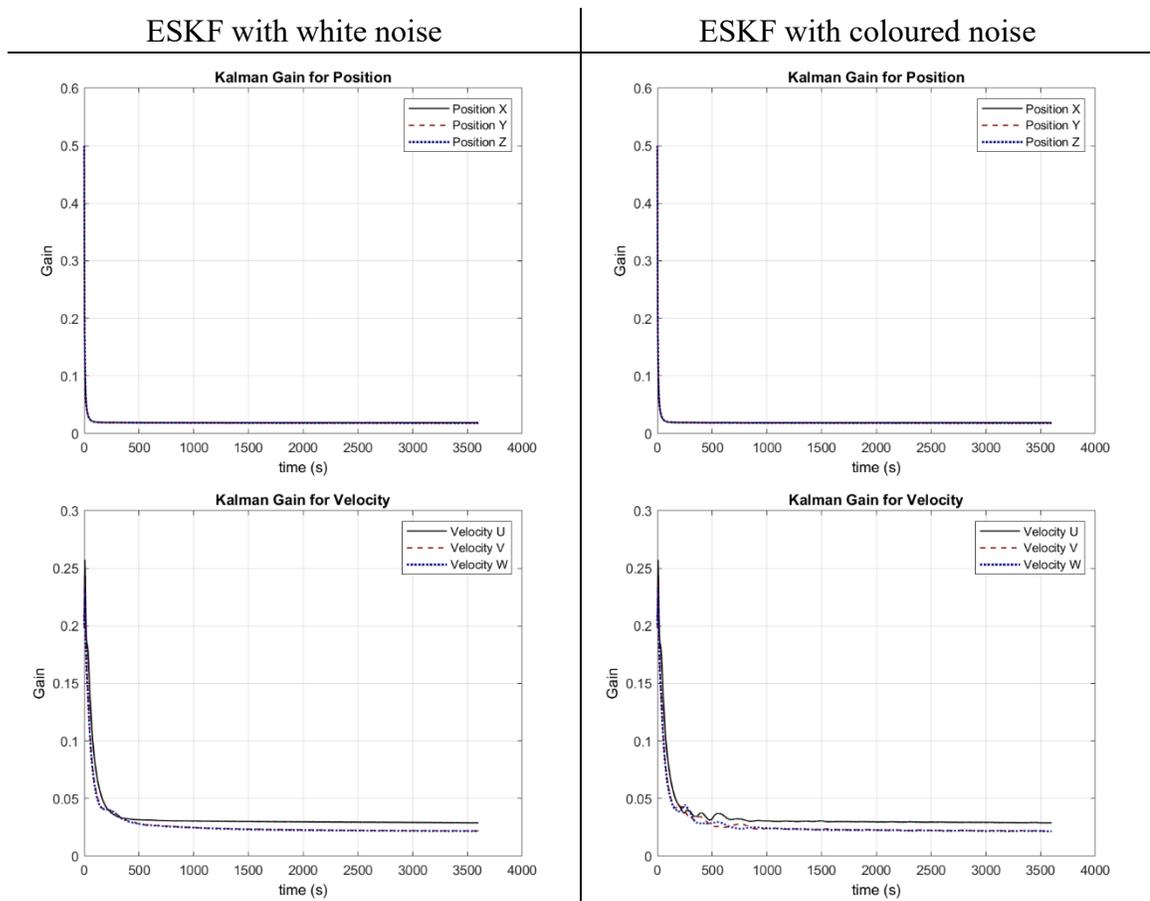


Figure 4-17 – Kalman Gain for Position and Velocity considering the ESKF with white (left) and coloured (right) noise

The Kalman gain results show similar behaviour as the state estimates variance, where the minimum gain, reached when the system achieves the steady-state, are visually the same for the system with white and coloured noise. The only effect observed is a small instability in the firsts 1000 seconds for the Kalman Gain in the velocity state for the system with coloured noise.

Therefore, the residuals were identified as the state that best represents the high error bound showed for the ESKF navigation solution when coloured noise is introduced in the system.

One advantage in using the Error-State Kalman Filter is that there is no need to calculate the covariance of the state estimates to identify when the KF is presenting some instability.

The residuals are already providing the error of the system.

4.4. UKF Solution

Using the same inputs, parameters, and configuration of the ESKF system, a UKF system in place of the ESKF system was simulated. This section is intended to show the UKF solution, and is divided as follows: first, the results for the GNSS/INS integration considering both IMUs with white and coloured noise. And secondly, the UKF states outputs are compared aimed to identify the states that best reflect the effect caused by the coloured noise introduction.

The UKF was modelled considering the methodology described in sections 2.5.3 and 3.2.3, using the navigation profile shown in section 3.1.1.

4.4.1. White and Coloured Noise UKF Solution

Figure 4-18 present the attitude error between the true-ground and the INS/GNSS integration solution using UKF with white and coloured noise. Figure 4-19 present the position error between the true-ground and the INS/GNSS integration solution using UKF with white and coloured noise. Figure 4-20 present the velocity error between the true-ground and the INS/GNSS integration solution using UKF with white and coloured noise.

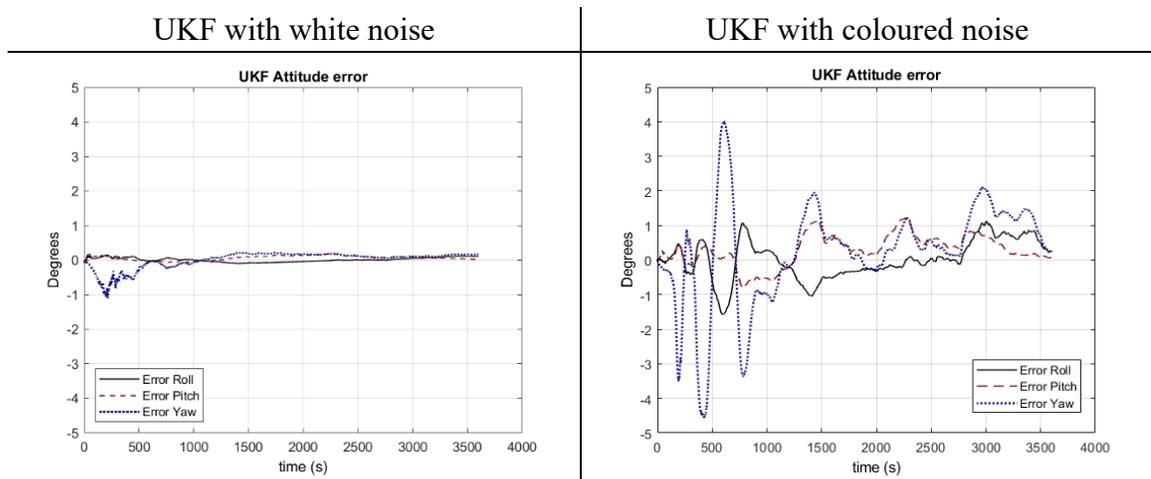


Figure 4-18 - Attitude error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise

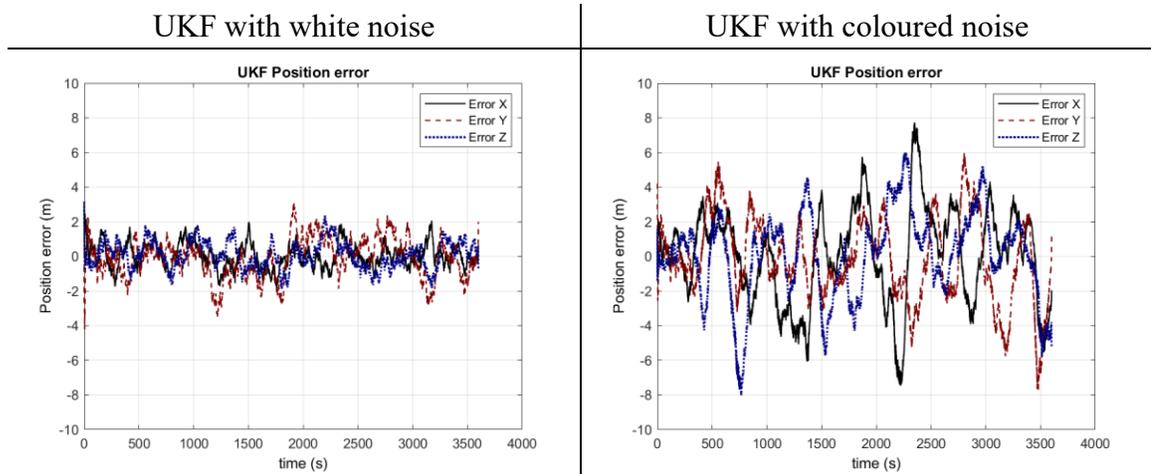


Figure 4-19 - Position error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise

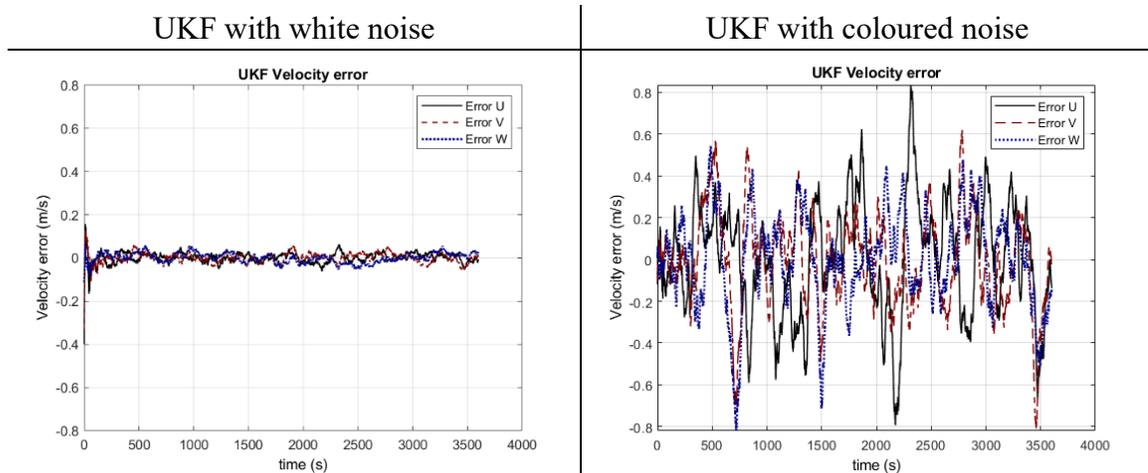


Figure 4-20 - Velocity error between the true-ground and the INS/GNSS integration using UKF with white (left) and coloured (right) noise

The results present in Figure 4-18 through Figure 4-20 shows that the UKF presents similar results to those found using ESKF for the system with white noise, and slightly better outcomes for the system with coloured noise. The attitude error after the system achieve the stability for the UKF solution remains between $\pm 0.2^\circ$ for the white noise, the same value found for the ESKF, and between $\pm 2^\circ$ for the coloured noise (the value found for ESKF was $\pm 3^\circ$).

For the position, the UKF solution shows a final position error that remains between ± 3 m for the white noise and between ± 8 m for the coloured noise.

For the velocity, the ESKF solution shows a final velocity error between ± 0.06 m/s for the white noise and between ± 0.8 m/s for the coloured noise.

As observed in the attitude state, the position and velocity states showed a small improvement when UKF is used over ESKF for a system with coloured noise. Further discussion will be given in section 0.

The bias correction for the gyroscope and accelerometer are presented in Figure 4-21 and Figure 4-22, respectively, using UKF with white and coloured noise.

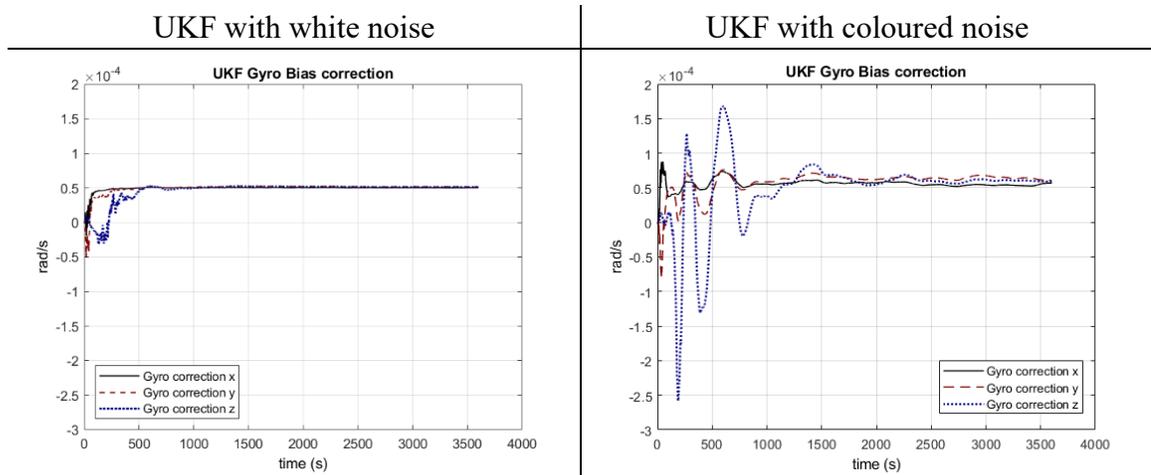


Figure 4-21 – Gyroscope bias correction given by the UKF with white (left) and coloured (right) noise

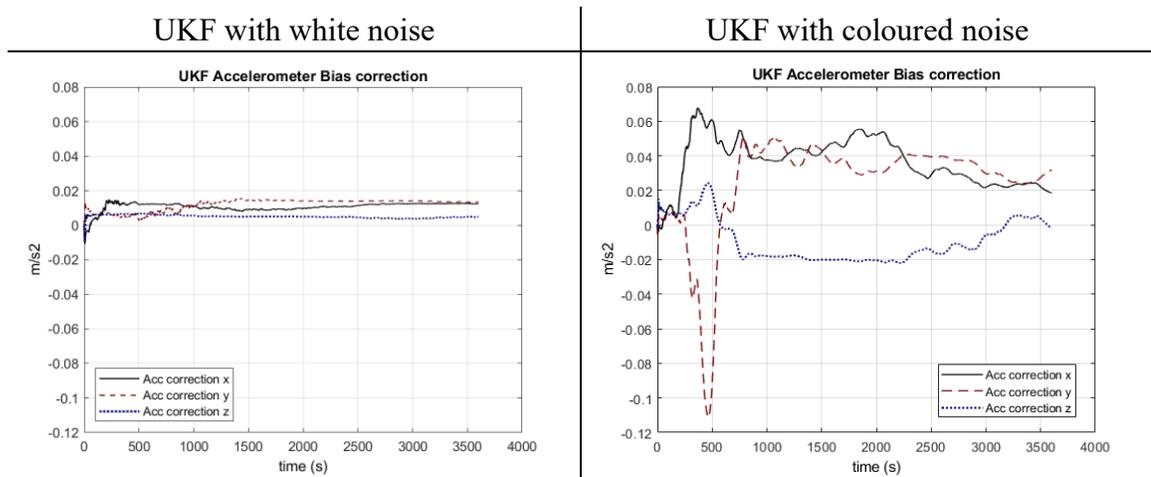


Figure 4-22 - Accelerometer bias correction given by the UKF with white (left) and coloured (right) noise

Similar to the results shown for the ESKF, Figure 4-21 shows that both systems with white and coloured noise could correctly predict the bias of $5 \times 10^{-5} \text{ rad/s}$ using UKF. However, when coloured noise is present, the system faces a higher instability at the beginning and needs more time to achieve the correct bias value.

For the accelerometer bias correction, it is possible to see that the system closely estimates the correct bias of $1 \times 10^{-2} \text{ m/s}^2$. However, for the system with coloured noise, the predicted accelerometer bias achieves a wrong prediction for the z-axis. This effect can be caused by the high impact of gravity in the accelerometer measurements.

4.4.2. White and Coloured Noise UKF states

Figure 4-23 presents the variance for Attitude, Position, and Velocity states, using UKF with white and coloured noise.

Figure 4-24 presents the variance for the gyroscope and accelerometer bias states, using UKF with white and coloured noise.

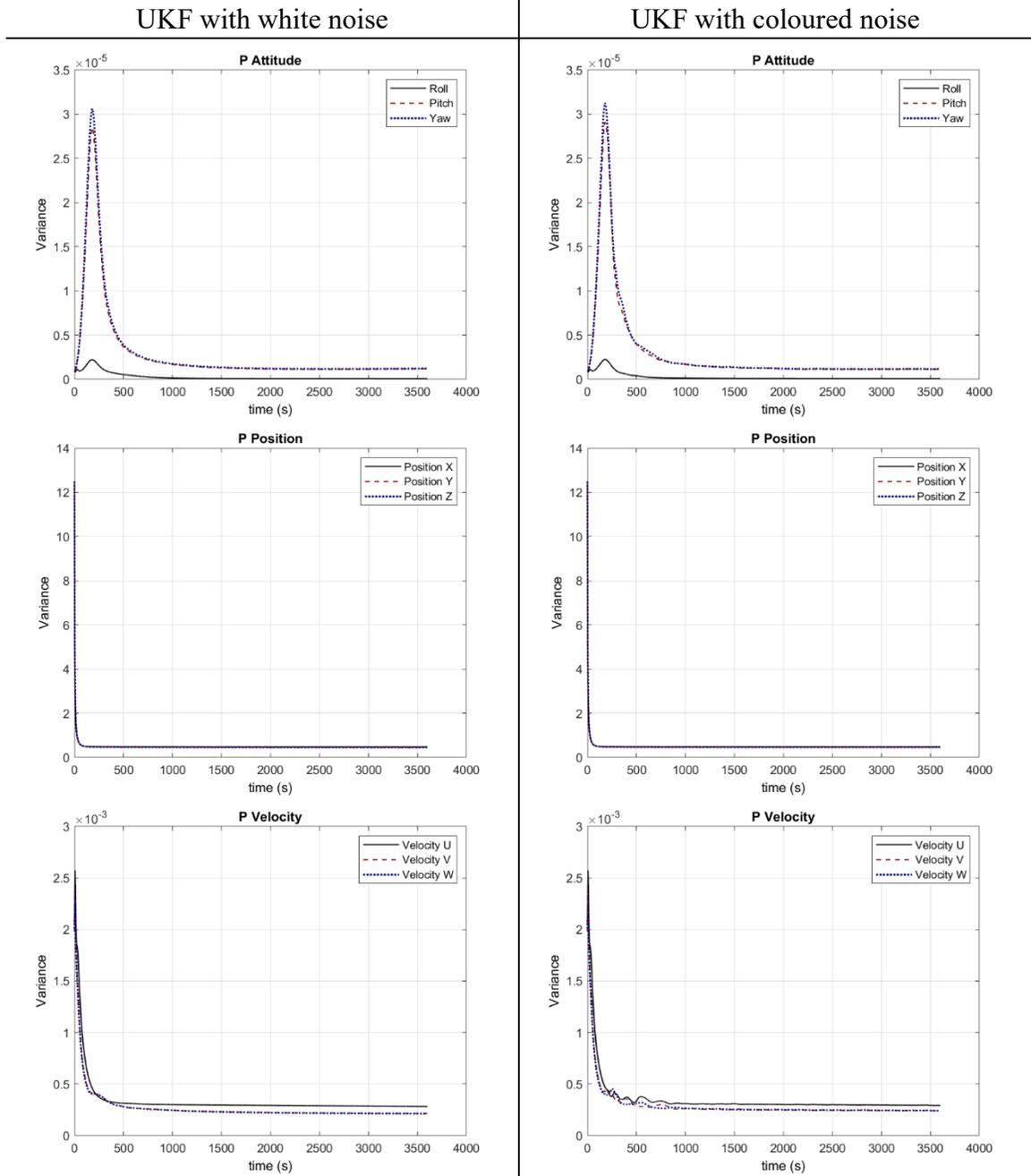


Figure 4-23 – Variance for Attitude, Position, and Velocity states considering the UKF with white (left) and coloured (right) noise

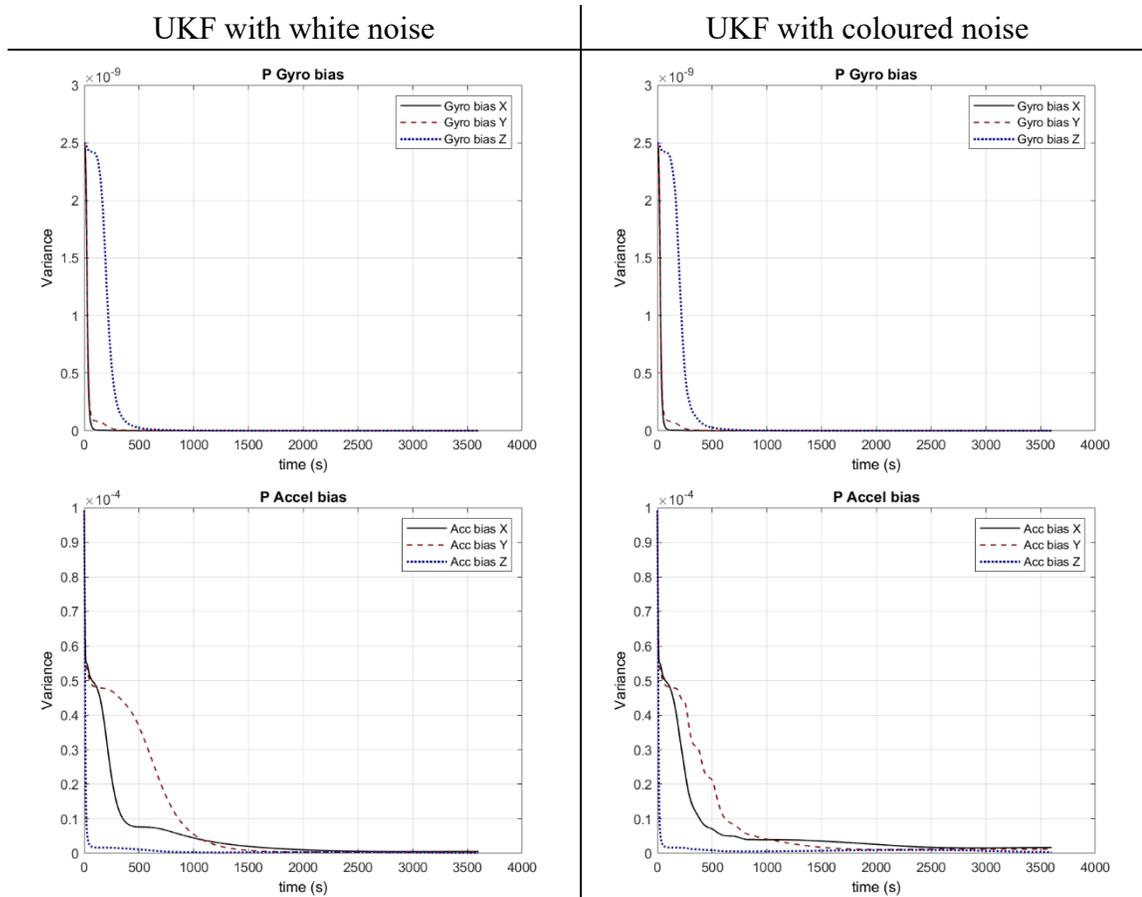


Figure 4-24 - Variance for the gyroscope and accelerometer bias state considering the UKF with white (left) and coloured (right) noise

Figure 4-25 presents the KF residuals for Attitude, Position, and Velocity, using UKF with white and coloured noise.

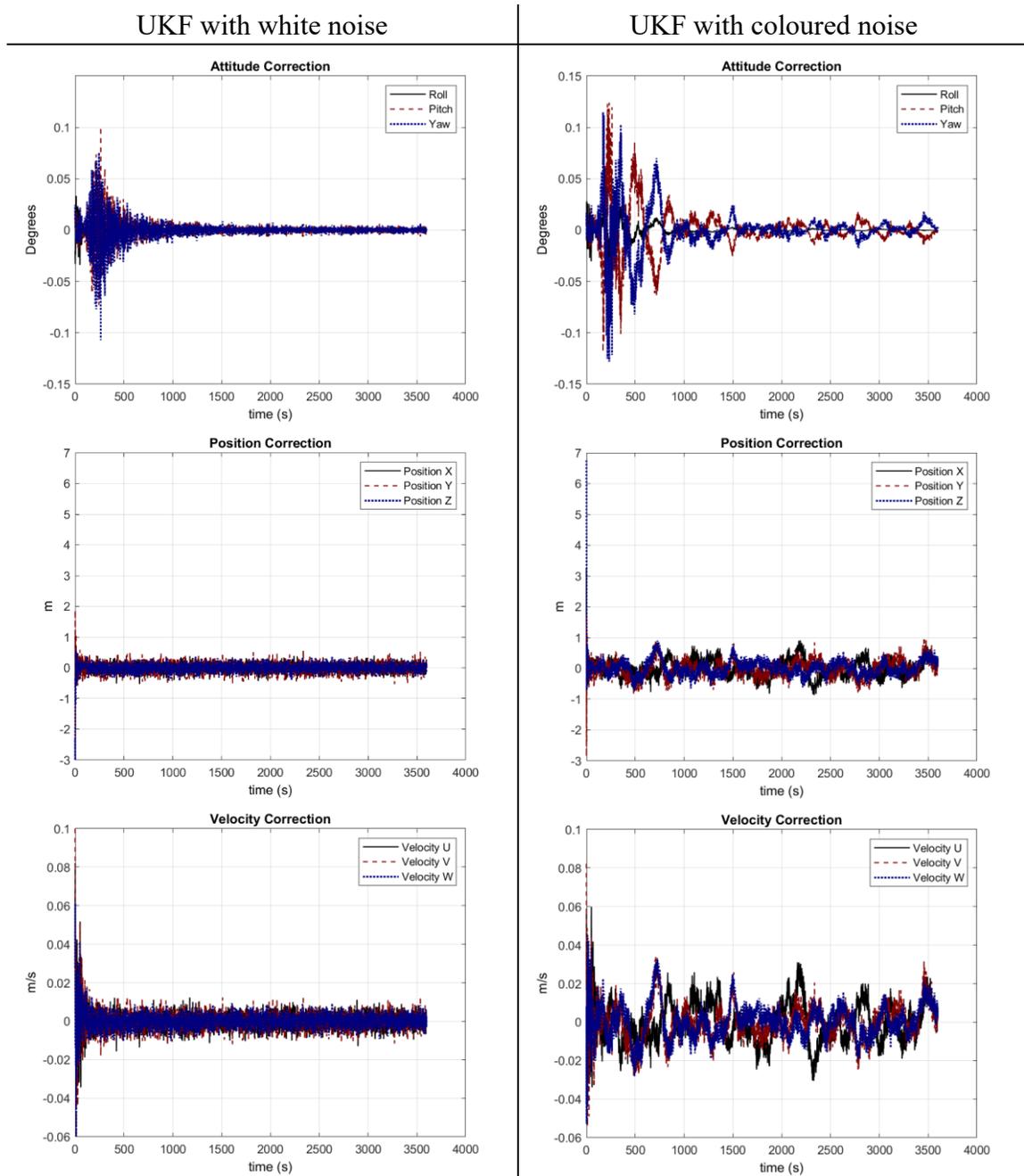


Figure 4-25 – KF residuals for Attitude, Position, and Velocity considering the UKF with white (left) and coloured (right) noise

Figure 4-26 presents the KF residuals for the gyroscope and accelerometer bias, using UKF with white and coloured noise.

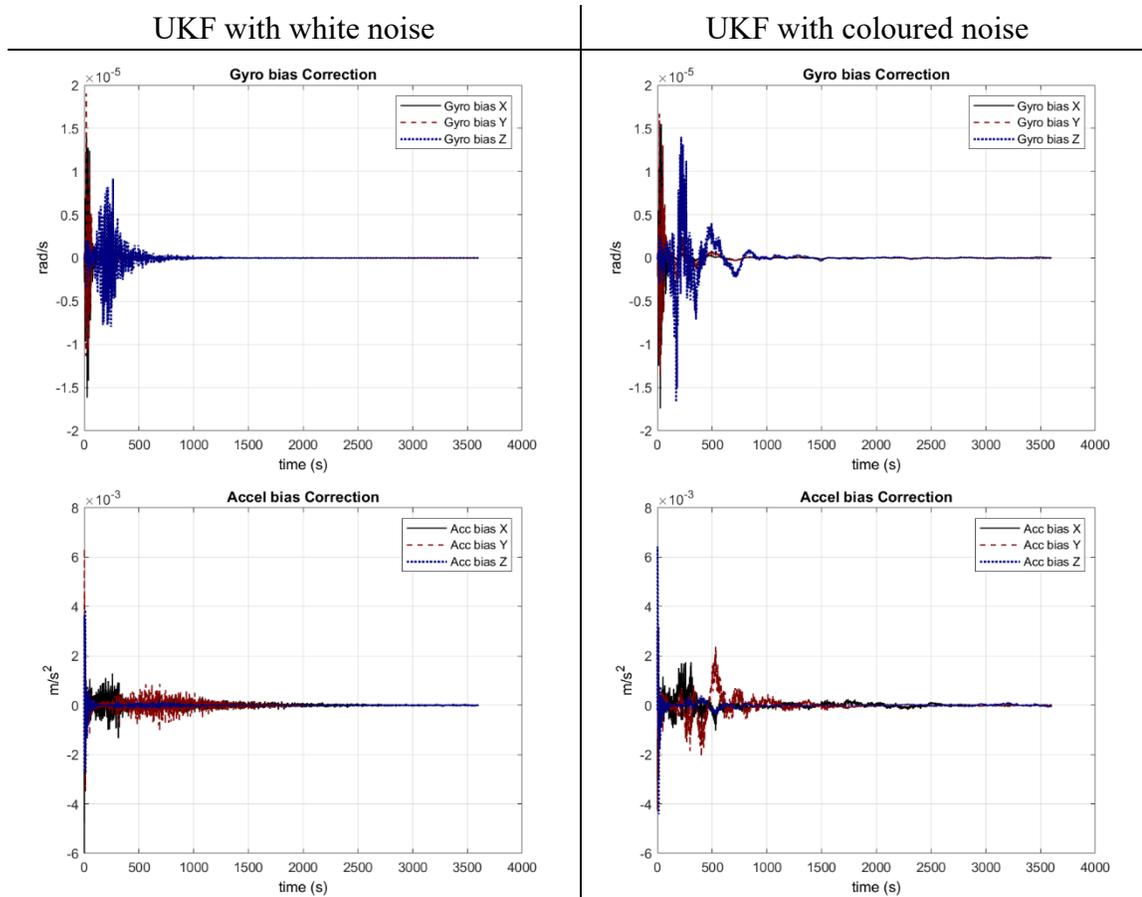


Figure 4-26 - KF residuals for gyroscope and accelerometer bias considering the UKF with white (left) and coloured (right) noise

Figure 4-27 presents the Kalman gain for the position and velocity, considering the UKF with white and coloured noise.

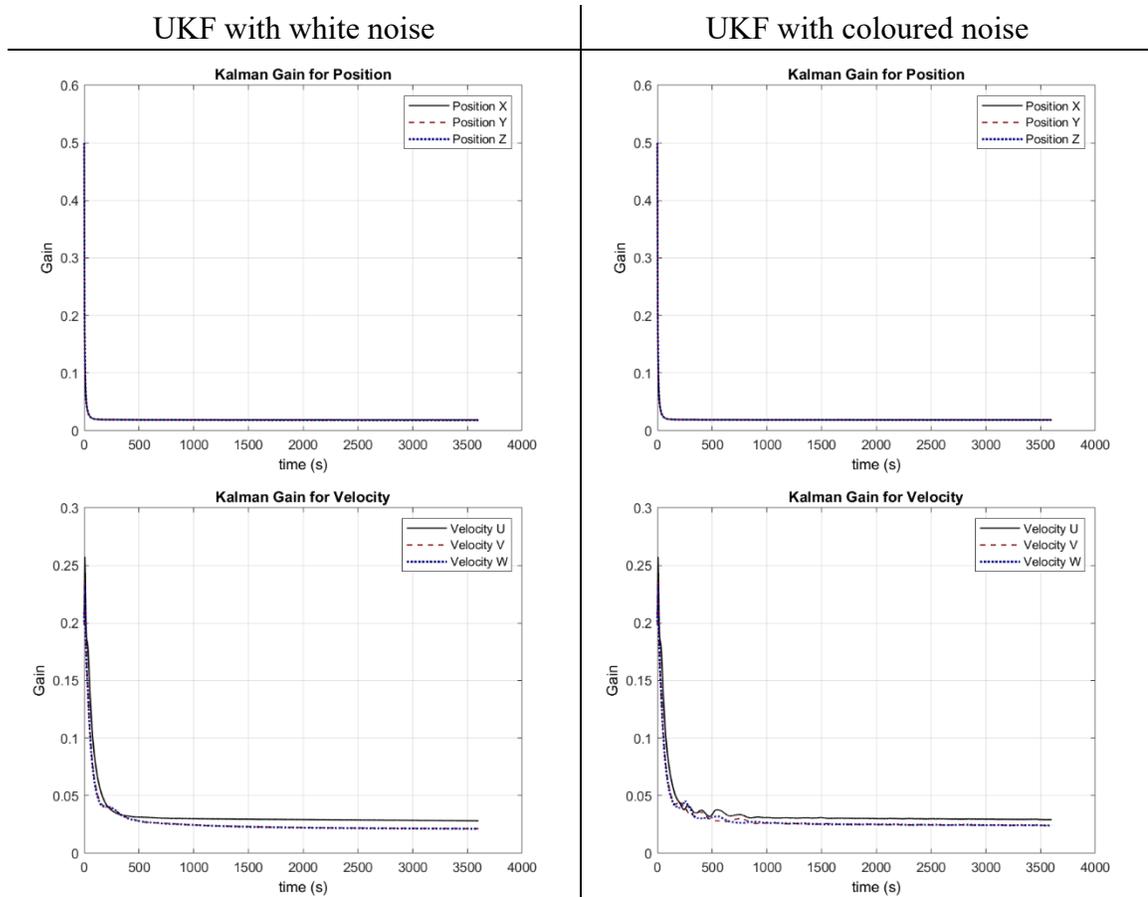


Figure 4-27 - Kalman Gain for Position and Velocity considering the UKF with white (left) and coloured (right) noise

The results for the error covariance of the state estimates, residuals and Kalman gain obtained by the use of UKF are visually identical to the results for the ESKF. Therefore, the same analyses made in section 4.3.2 are valid for the UKF.

A further discussion about the differences between the ESKF and UKF will be provided in section 4.6.

4.5. Fuzzy Logic

This section is intended to show the results when the Fuzzy Logic is implemented in both ESKF and UKF solutions and is divided as follows: first, the fuzzy logic behavior is shown. Secondly, the Fuzzy Logic adaptive ESKF simulation results are presented and compared

against the standard ESKF. And, Thirdly, the Fuzzy Logic adaptive UKF simulation results are presented and compared against the standard UKF.

4.5.1. Fuzzy Logic behavior

Figure 4-28 shows the behaviour of the Fuzzy Logic output α and β when it is applied to the ESKF solution with coloured noise, as discussed in section 4.3.1, following the methodology described in section 3.3

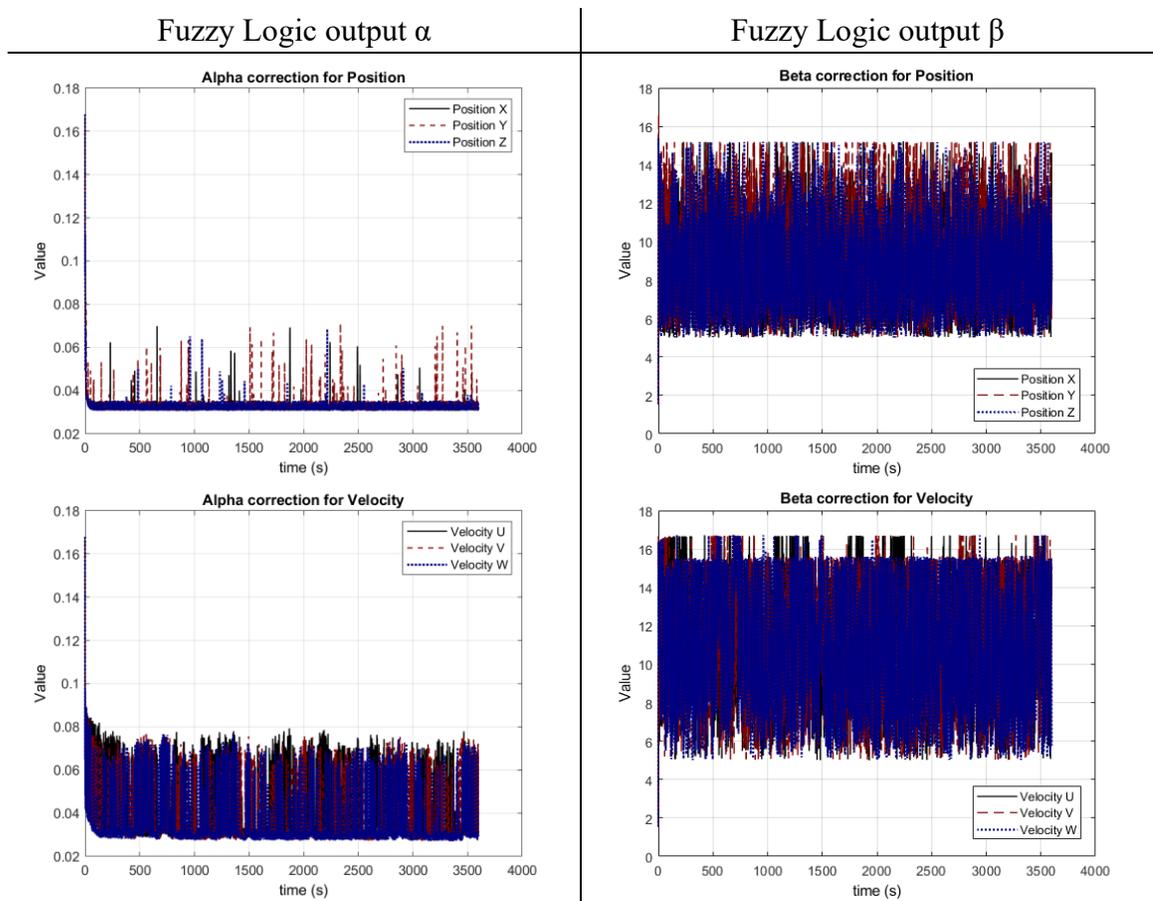


Figure 4-28 - Fuzzy Logic output α and β for position and velocity states

These results show that the Fuzzy Logic is adjusting the KF as expected, as apply an exponential weighted correction α when the error covariance value is high, manly during

the system initialization, and a fictitious process noise injection β when the system is in steady-state.

4.5.2. Fuzzy Logic Adaptive ESKF

Fuzzy Logic Adaptive ESKF applied to a system with white noise

To check if the application of the fuzzy logic will bring any damage to an optimal system, the fuzzy logic adaptive ESKF was applied to the system with white noise. The results are presented for the INS/GNSS integration using ESKF with and without the FL correction. Figure 4-29 shows the error in the position, Figure 4-30 present the velocity error, and Figure 4-31 present the attitude error.

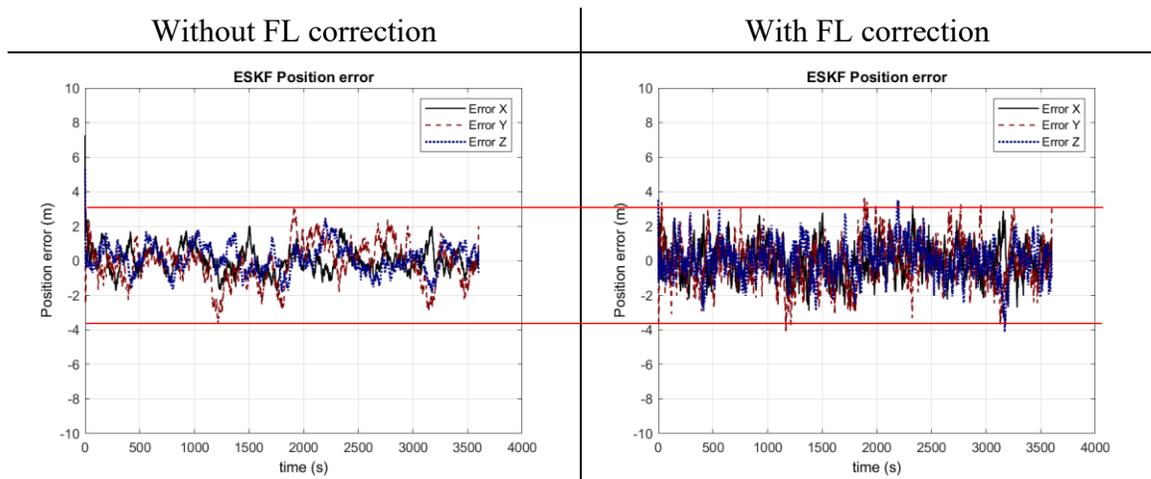


Figure 4-29 – Position error using ESKF without FL correction (left) and with FL correction (right)

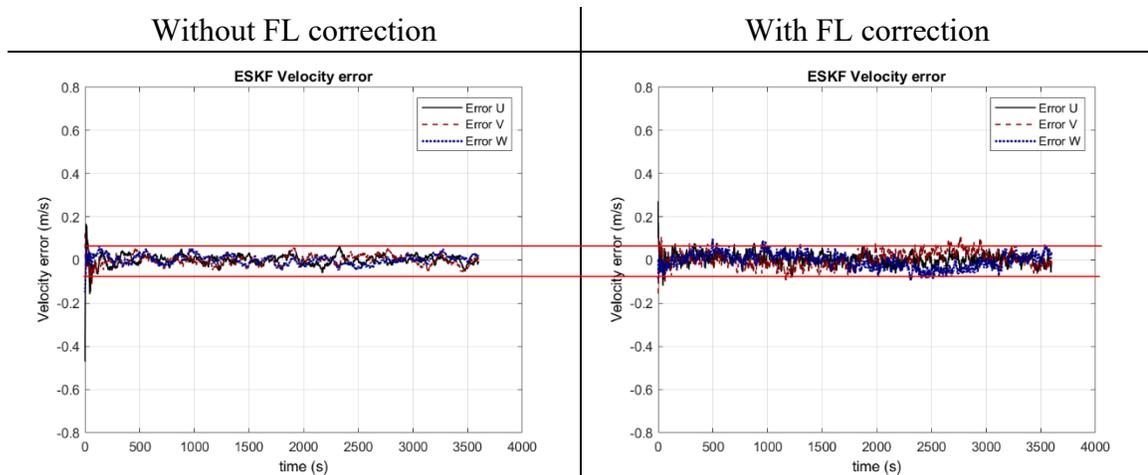


Figure 4-30 – Velocity error using ESKF without FL correction (left) and with FL correction (right)

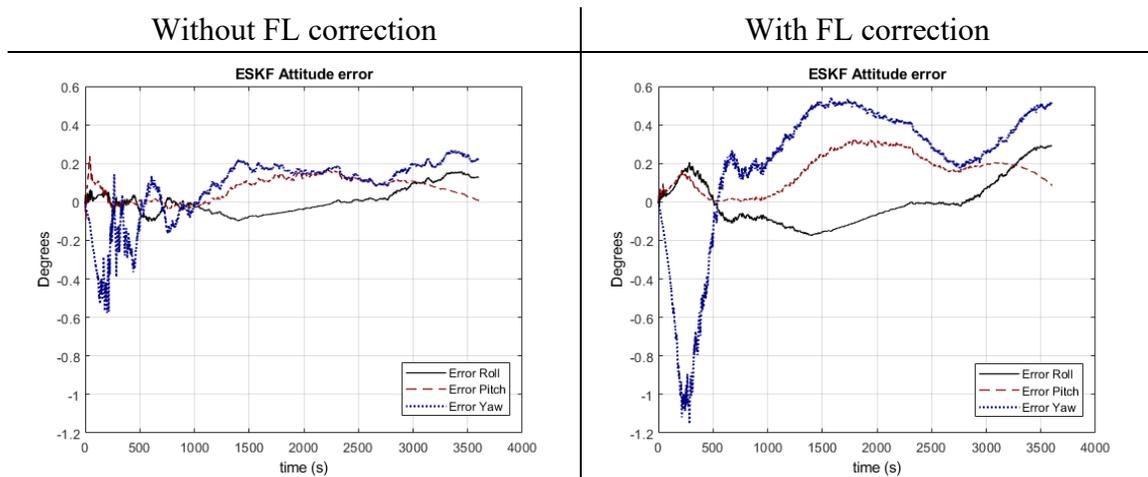


Figure 4-31 – Attitude error using ESKF without FL correction (left) and with FL correction (right)

These results show the effect of applying a Fuzzy logic adaptive ESKF solution in an optimal KF, where only white noise is present. It is possible to observe that, when the FL is used, all states are slightly damaged. For the position and velocity, the application of the FL enlarged the error by around 10%. For the attitude, after the system achieves the stability, the error without the FL was between $\pm 0.2^\circ$ and, with the introduction of FL correction, the error was elevated to $\pm 0.6^\circ$.

Once the ESKF was tuned to present the optimal solution when only white noise is present, it is expected that by the use of FL, which mainly enlarge the process noise covariance value, the final solution would not be optimal.

However, the degradation in the attitude, position, and velocity states caused by the use of FL is relatively small. Therefore, the advantages and harms caused using an FL adaptive approach needs to be considered to define the benefits of its application.

Fuzzy Logic Adaptive ESKF applied to a system with coloured noise

Considering the methodology presented in section 3.3.1, a FL adaptive ESKF was simulated aimed to correct the sub-optimal ESKF solution occurred when coloured noise is present in the IMU. The results are presented for the INS/GNSS integration using ESKF with and without the FL correction. Figure 4-32 shows the error in the position, Figure 4-33 present the velocity error, and Figure 4-34 present the attitude error.

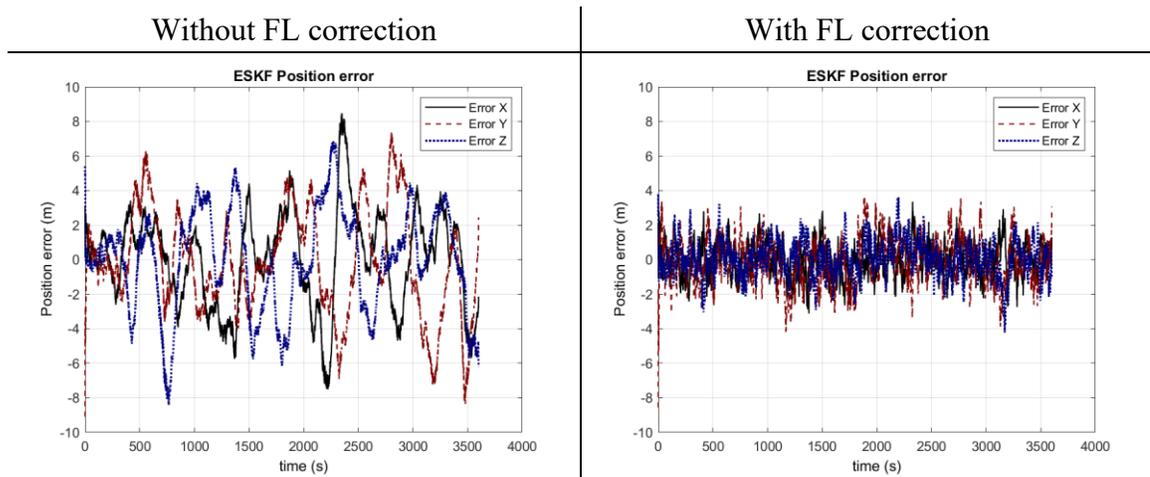


Figure 4-32 – Position error using ESKF without FL correction (left) and with FL correction (right)

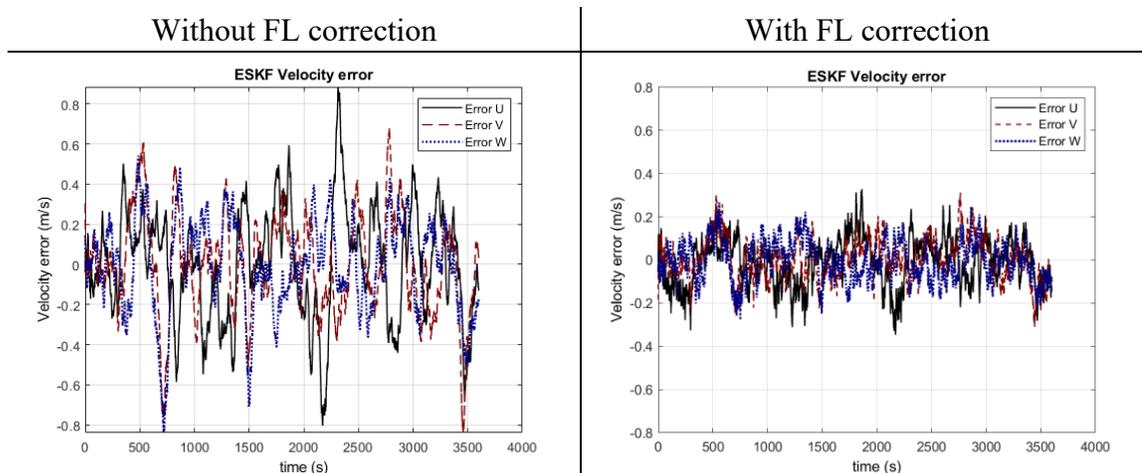


Figure 4-33 – Velocity error using ESKF without FL correction (left) and with FL correction (right)

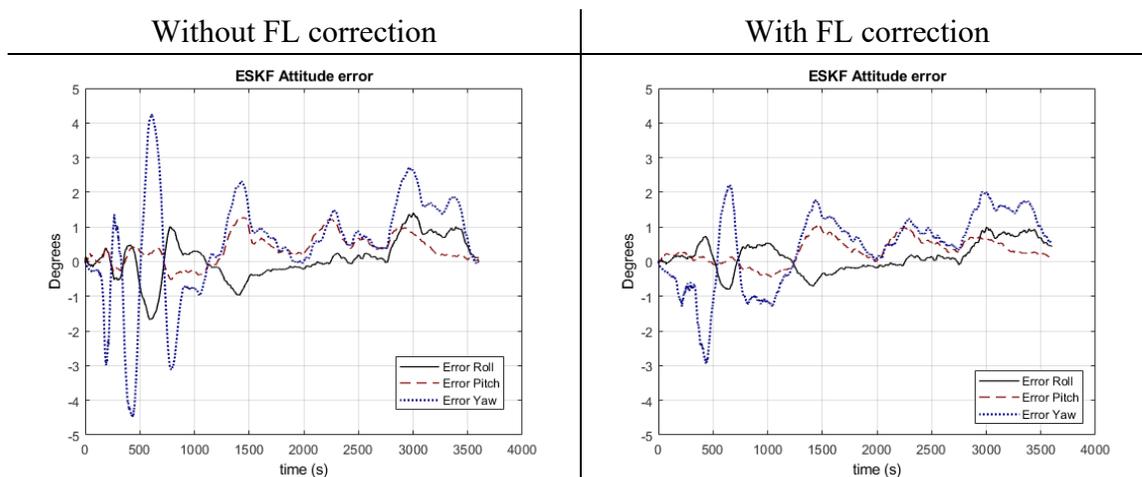


Figure 4-34 – Attitude error using ESKF without FL correction (left) and with FL correction (right)

These results show the effect of applying a Fuzzy logic adaptive ESKF solution in a system with coloured noise in the IMU. It is possible to observe a substantial improvement when the FL is used. In steady-state, the maximum error for the position was decreased from +/- 8 m to +/- 4 m, in the velocity from +/- 0.8 m/s to +/- 0.3 m/s, and in the attitude from +/- 2.8° to +/- 2°. The overall results show that it was possible to reduce the error caused by the coloured noise by approximately half of the value.

To verify the effect of the FL application in the stability of the system, the ESKF states will be presented. Figure 4-35 presents the *a posteriori* variance for the Attitude, Position, and Velocity states, using ESKF with coloured noise, for the system with and without FL.

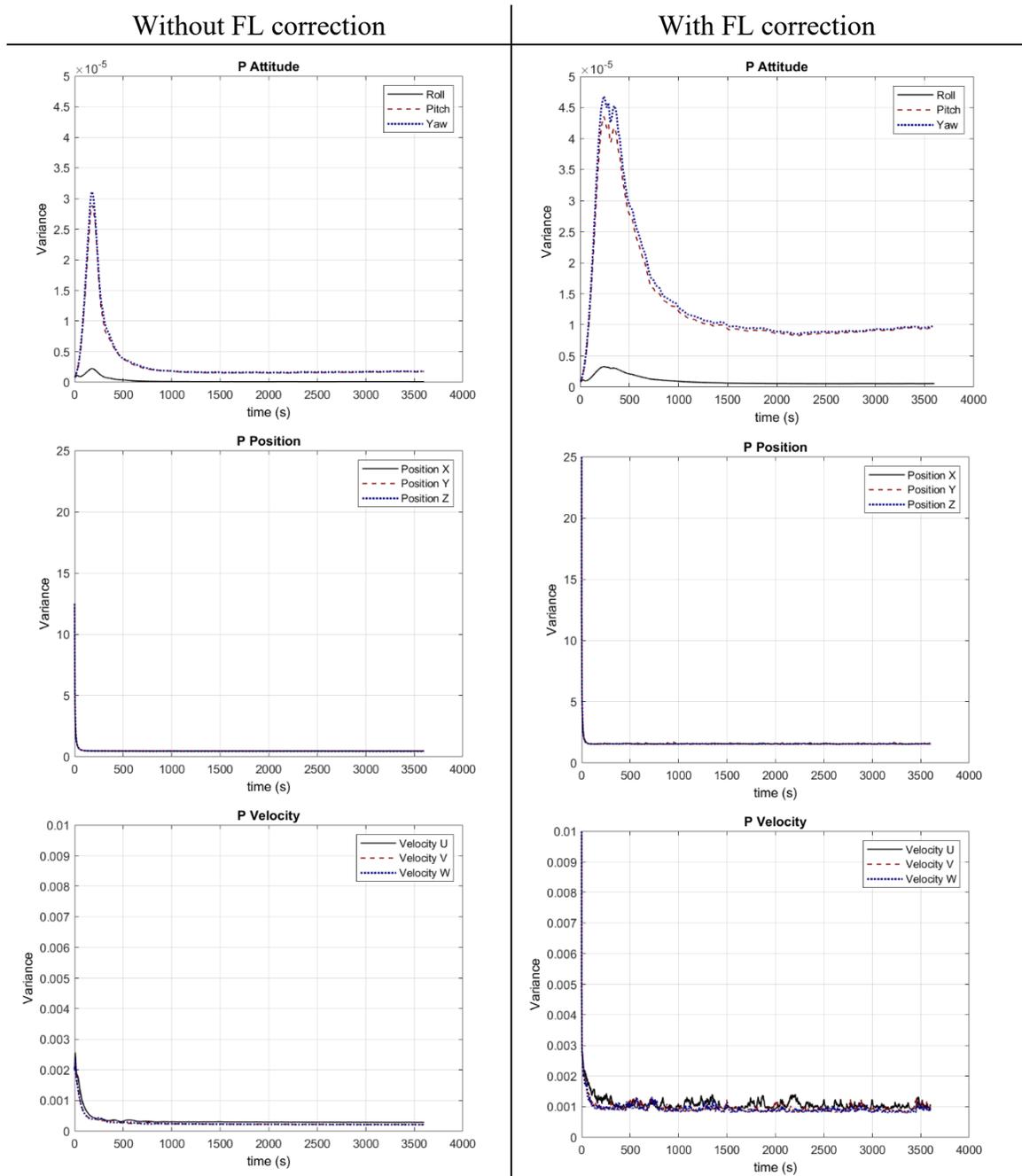


Figure 4-35 – Variance for the attitude, position, and velocity states considering the system without FL (left) and with FL (right) correction

The variance for the attitude, position, and velocity states shows that the application of the FL elevates the error variance value and add some instability when the system achieves the steady-state. This is expected, as the implementation of the FL enlarges mainly the process noise covariance matrix, which causes a growth in the error covariance values. The results also show that no true divergence is observed. Therefore, although the FL application enlarges the error covariance of the system, the stability of the system was maintained.

Figure 4-36 present the residuals for Attitude, Position, and Velocity states, using ESKF with coloured noise for the system without and with FL correction.

Figure 4-37 presents the residuals for the gyroscope and accelerometer bias states, using ESKF with coloured noise for the system without and with FL correction.

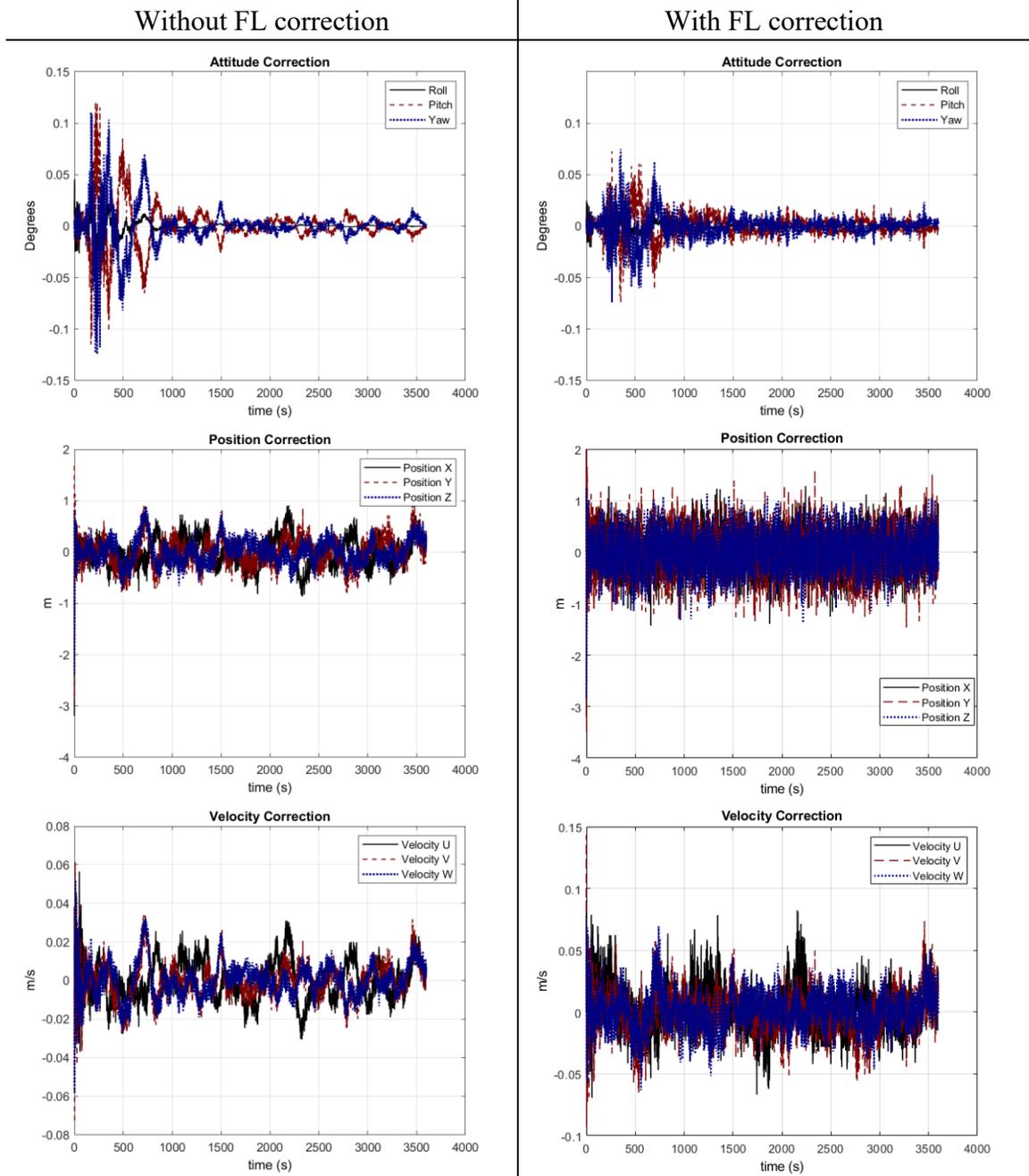


Figure 4-36 – ESKF residuals for Attitude, Position, and Velocity considering the ESKF without FL correction (left) and with FL correction (right)

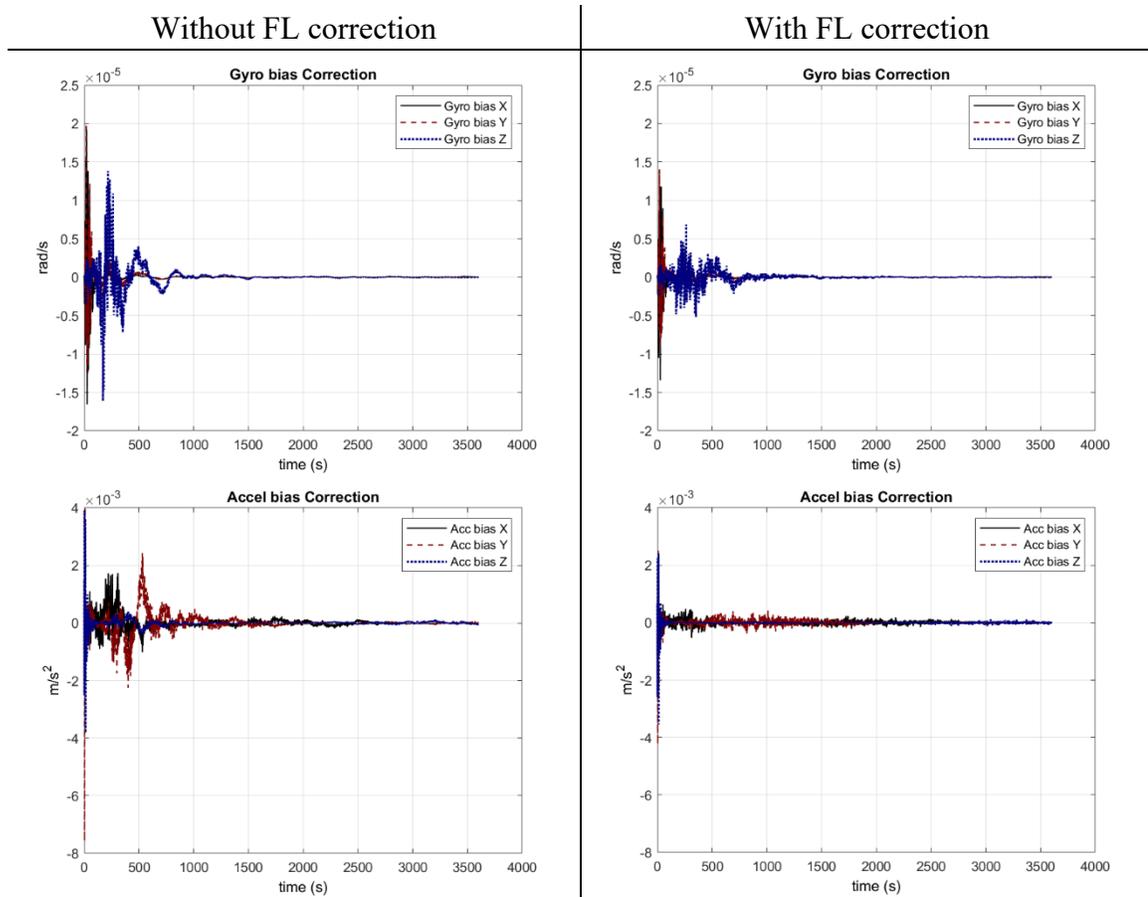


Figure 4-37 - ESKF residuals for gyroscope and accelerometer bias considering the ESKF without FL correction (left) and with FL correction (right)

The residuals show a significant difference between the system with and without FL correction. For the states that are measured by the GNSS - position and velocity, the residuals are slightly increased, showing that the KF is correcting the INS in a more weighted way. This is expected, as the application of the FL enhanced the Kalman gain for these states. However, for the other states – attitude, gyroscope bias, and accelerometer bias -, the residuals show a behaviour that is close to the optimal system. This indicates that the FL application brought the INS correction, for these non-measured states, close to the optimal values.

Figure 4-38 presents the Kalman gain for the position and velocity using ESKF with coloured noise for the system without and with FL correction.

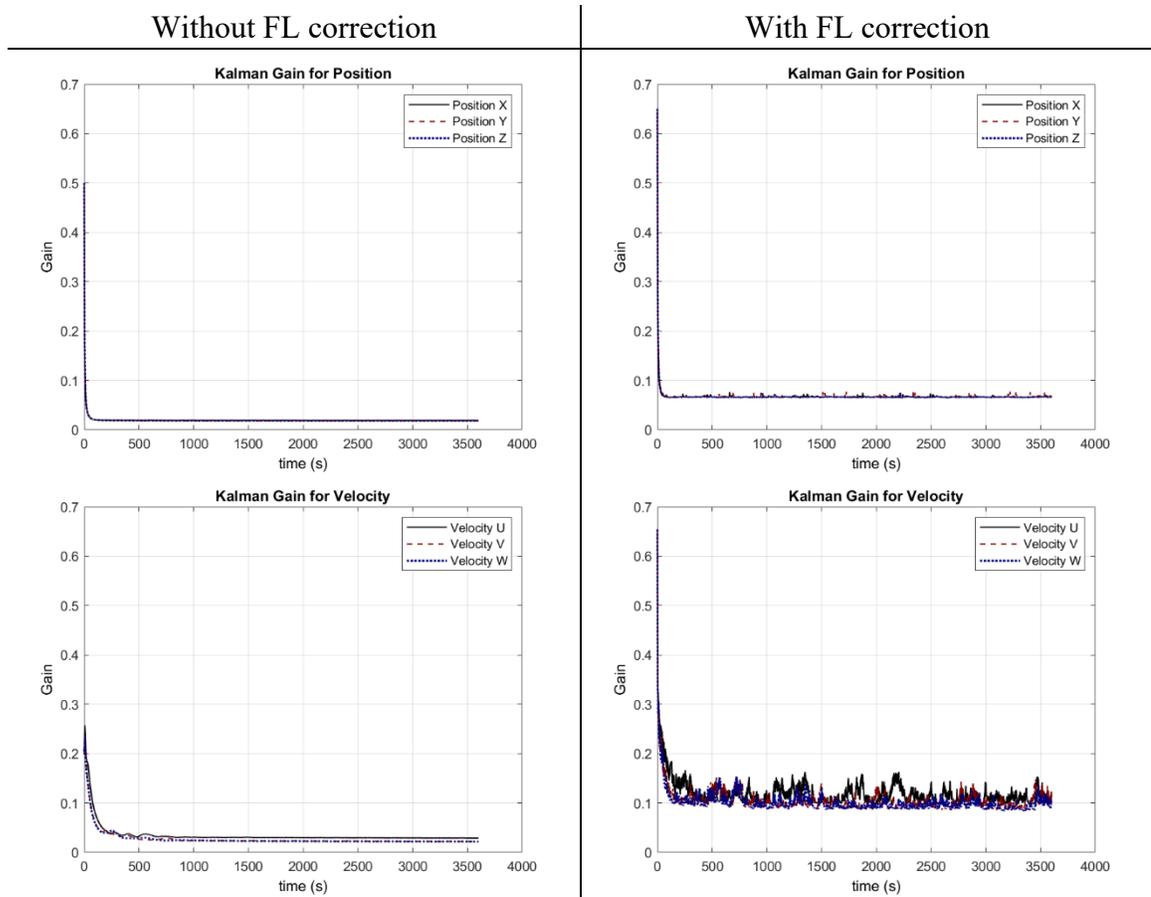


Figure 4-38 – Kalman Gain for Position and Velocity considering the ESKF without FL correction (left) and with FL correction (right)

The Kalman gain results show similar behaviour as the variance for the state estimates, as the FL elevates the Kalman gain value and also add some instability when the system achieves the steady-state. This is expected, as the application of the FL enlarged the error covariance value, which causes an increase in the Kalman gain. This prevents the KF from discounting new measurements for large k , which is precisely the main objective of the FL application.

4.5.3. Fuzzy Logic Adaptive UKF

In the same way, as done for ESKF, as presented in the last section, a Fuzzy Logic Adaptive UKF was simulated to correct the sub-optimal UKF solution achieved when coloured noise is present. The system and the FL application parameters are the same for the ESKF and UKF.

Because the results for the FL application in a system with white noise are visually the same for the ESKF and UKF, the results for the simulation of the FL adaptive UKF in a system with white noise will not be discussed. The reader can refer to the last section and take the ESKF results as the UKF results.

For the system with coloured noise, the overall results are very similar to the results found for the ESKF and will be briefly presented.

The results are presented for the INS/GNSS integration using UKF with and without the FL correction. Figure 4-39 present the position error, Figure 4-40 present the velocity error, and Figure 4-41 present the attitude error.

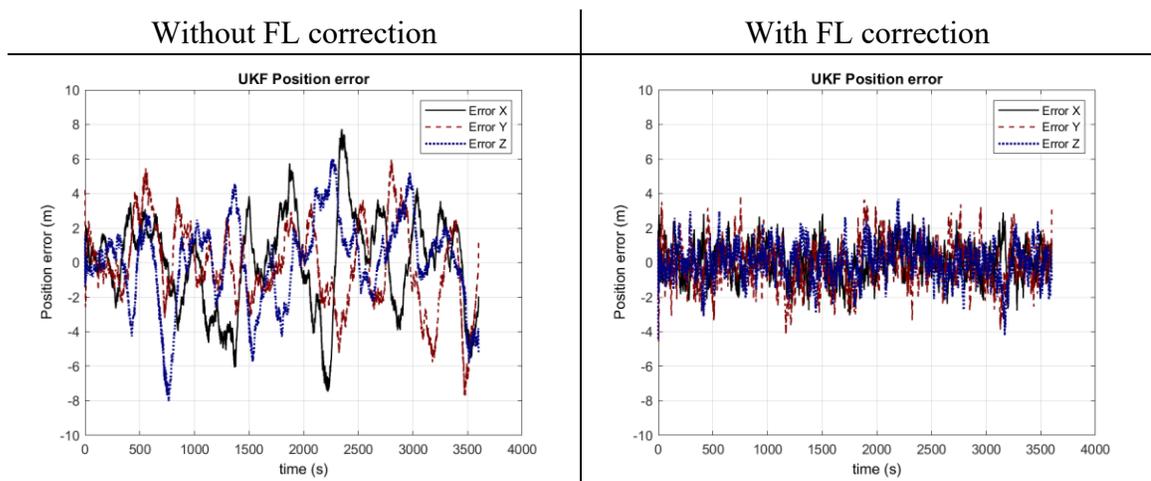


Figure 4-39 –Position error using UKF without FL correction (left) and with FL correction (right)

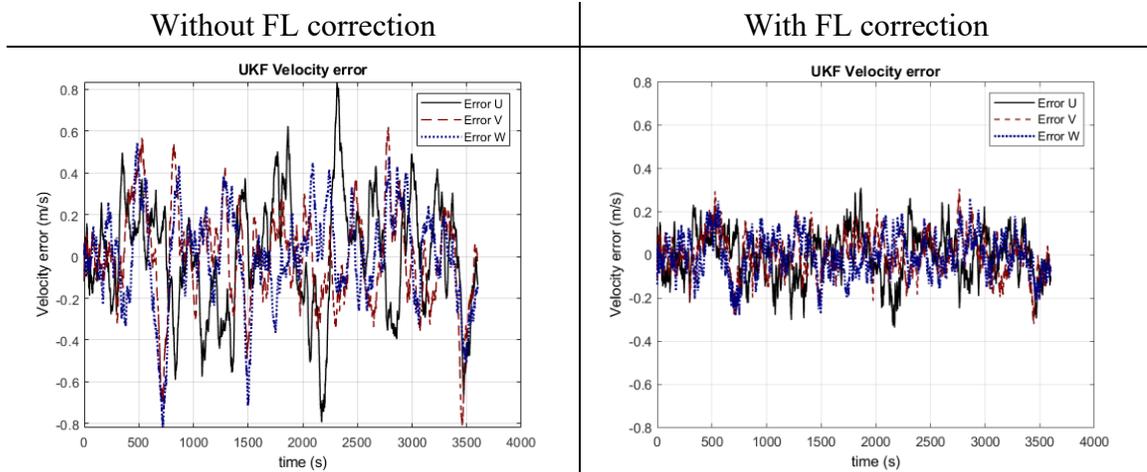


Figure 4-40 – Velocity error using UKF without FL correction (left) and with FL correction (right)

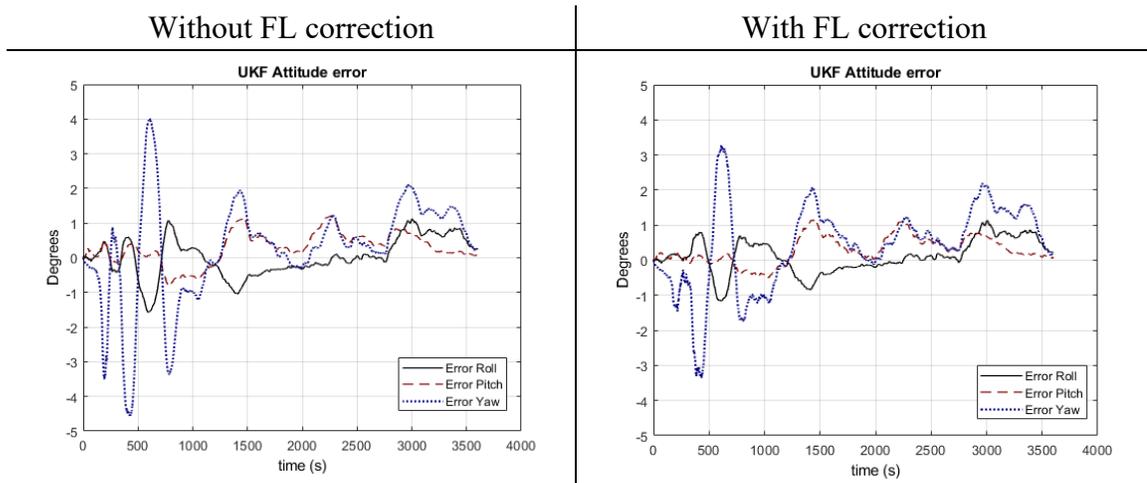


Figure 4-41 – Attitude error using UKF without FL correction (left) and with FL correction (right)

The results show in Figure 4-39 through Figure 4-41 indicates that the UKF present similar results to those found using ESKF. In steady-state, the maximum error for the position was decreased from ± 8 m to ± 4 m, the velocity from ± 0.8 m/s to ± 0.3 m/s, and the attitude from $\pm 2.8^\circ$ to $\pm 2^\circ$. The overall results show that it was possible to reduce the error caused by the coloured noise by approximately half of the value.

Figure 4-42 presents the *a posteriori* variance for the Attitude, Position, and Velocity states, using UKF with coloured noise, for the system with and without the FL application.

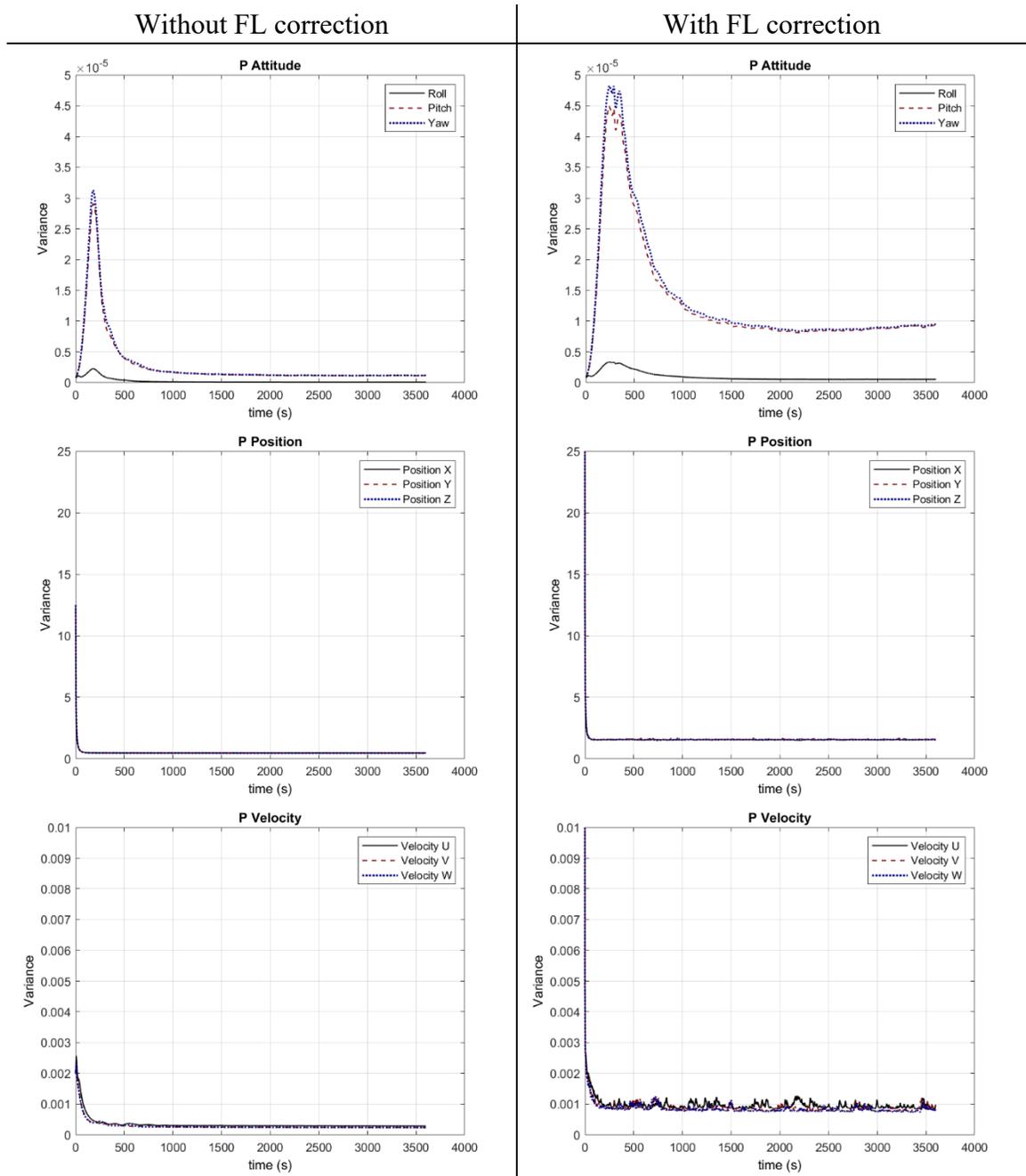


Figure 4-42 – Variance for the attitude, position, and velocity states considering the system without FL (left) and with FL (right) correction

Figure 4-43 present the residuals for Attitude, Position, and Velocity, using UKF with coloured noise for the system without and with FL correction.

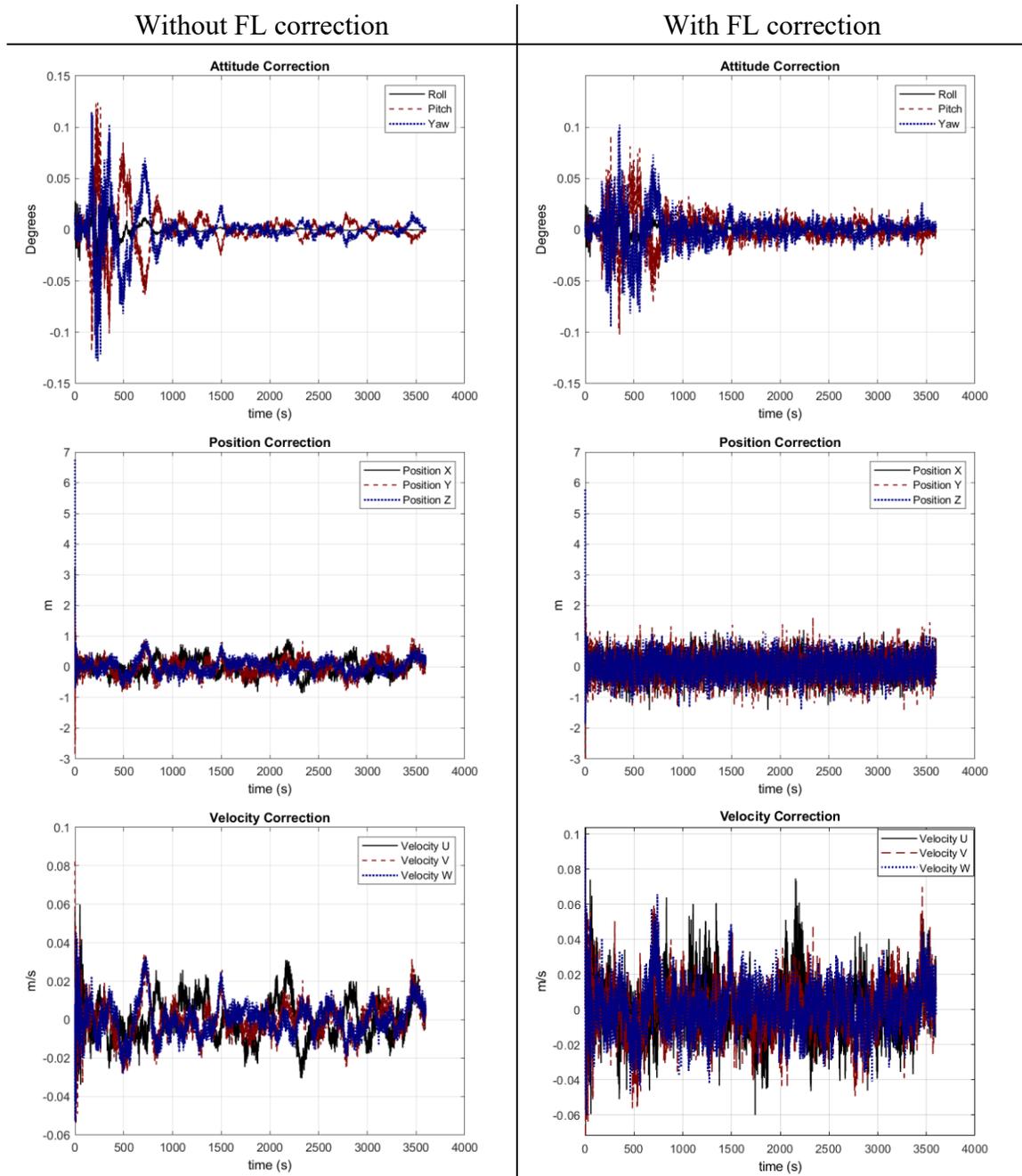


Figure 4-43 – UKF residuals for Attitude, Position, and Velocity considering the UKF without FL correction (left) and with FL correction (right)

Figure 4-44 presents the Kalman gain for the position and velocity using UKF with coloured noise for the system without and with FL correction.

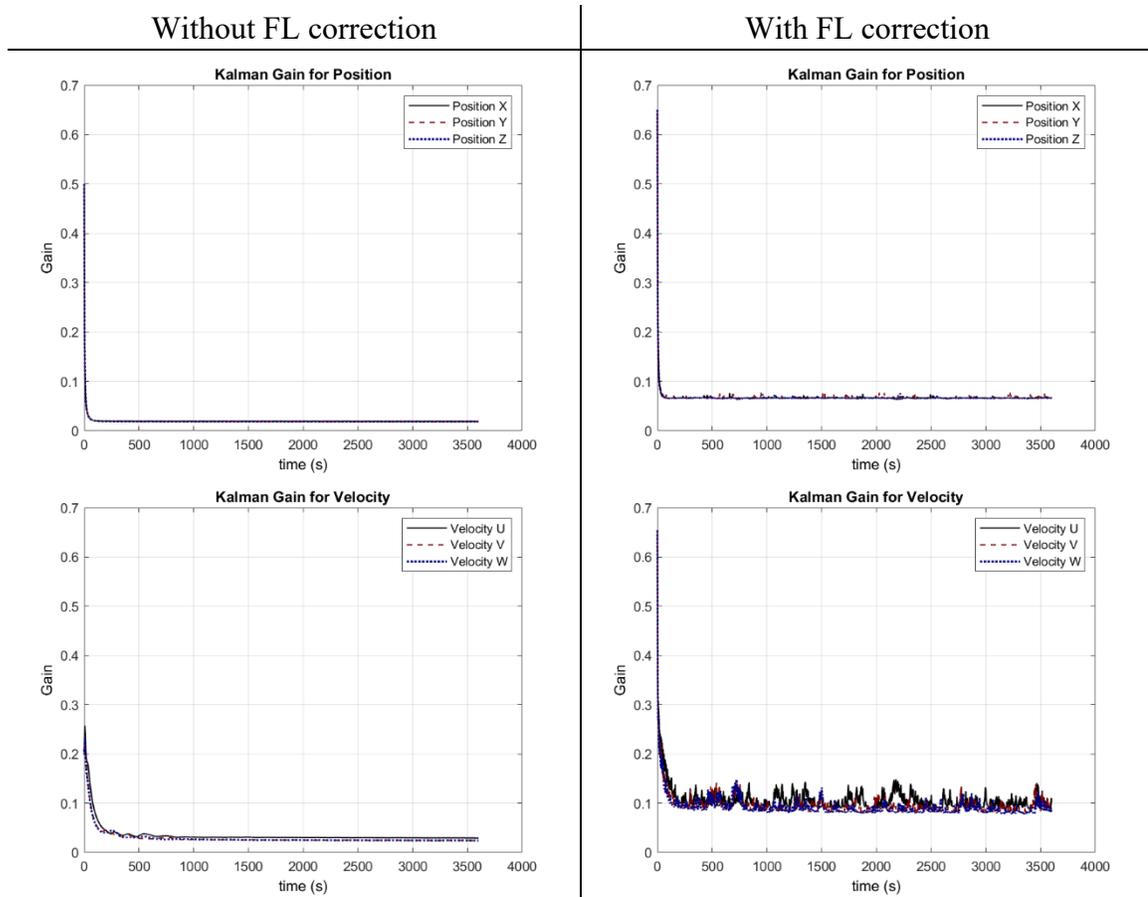


Figure 4-44 – Kalman Gain for Position and Velocity considering the UKF without FL correction (left) and with FL correction (right)

The results for the variance of the state estimates, residuals, and Kalman gain, obtained when UKF is used, are similar to the results obtained for the ESKF. Therefore, the same analyses made in section 4.5.2 are valid for the UKF.

A further discussion about the differences between the ESKF and UKF will be provided in the next section.

4.6. ESKF and UKF comparison

This section is intended to compare the results obtained by the ESKF and UKF. The results are discussed for the system with white noise, with coloured noise, and with the Fuzzy Logic correction for the coloured noise system, considering the true-ground attitude, position, and velocity error, the error covariance values, and the residuals.

This section will show only the main outcomes from the ESKF and UKF simulated results. The results will be presented solely for the dimension that produces the highest deviation. The full results are available in Appendix D. The reader can refer to this Appendix to have an overview of the difference between ESKF and UKF simulations results.

It is essential to highlight that all the inputs, process covariance matrix, measurement covariance matrix, transition matrix, and Fuzzy logic application are the same for both systems. Therefore, the differences observed in the outputs between the ESKF and UKF are solely due to its different methodology.

4.6.1. System with white noise

Considering the system with white noise in the IMU, the results are presented only for the dimension that shows the highest deviation for each state. Figure 4-45 presents the ESKF and UKF comparison for the attitude error. Figure 4-46 presents the comparison for the position and velocity error, and Figure 4-47 present the comparison for the gyroscope bias and accelerometer bias determination.

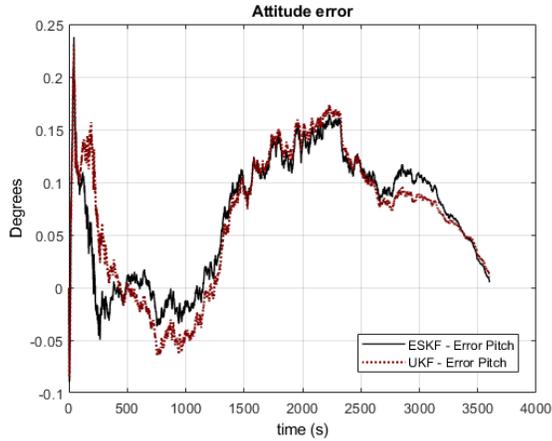


Figure 4-45 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude

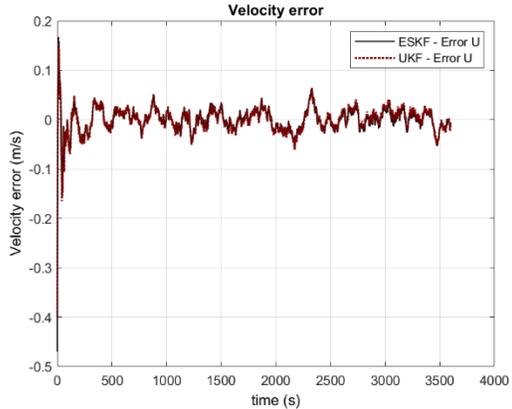
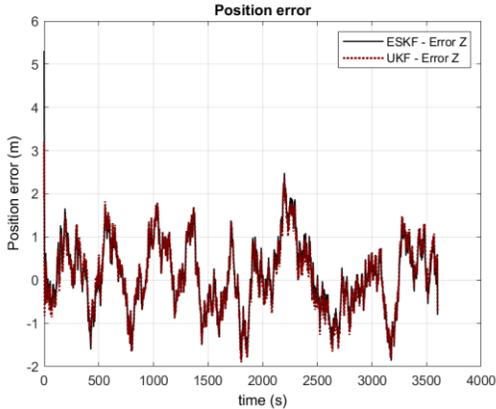


Figure 4-46 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and U velocity (right)

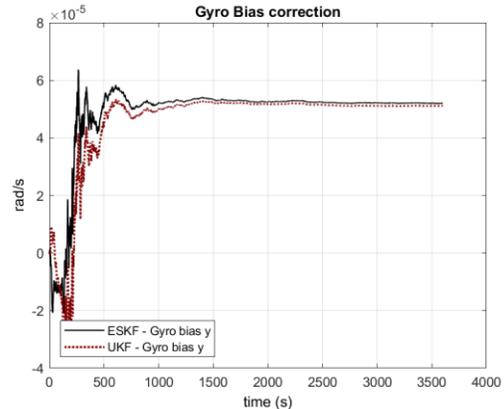
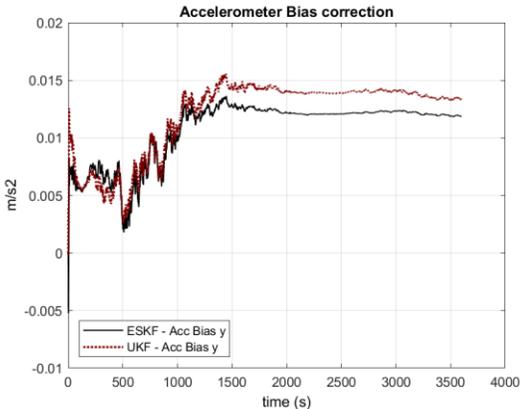


Figure 4-47 - Accelerometer bias (left) and Gyroscope bias (right) determination

It is possible to observe that the states that are measured by the GNSS (position and velocity) present the same results for ESKF and UKF. The other states, given by the KF transition matrix propagation, shows a small difference between the ESKF and UKF.

The results show that the attitude error, also shown in Figure D 1, presents the same overall behaviour for both KF approaches. However, the error is a little broader for the UKF. This same behaviour is also observed for the accelerometer bias determination.

For the variance of the state estimation and the residuals, the only state that shows a slight difference between ESKF and UKF is the attitude. Figure 4-48 shows the error covariance and the residuals for the pitch, where it is possible to observe that the error covariance for the UKF reaches a small value when compared to the ESKF. A small error covariance means that the Kalman gain will also be lower for the UKF. Therefore, as expected, the residuals for the ESKF are slightly bigger.

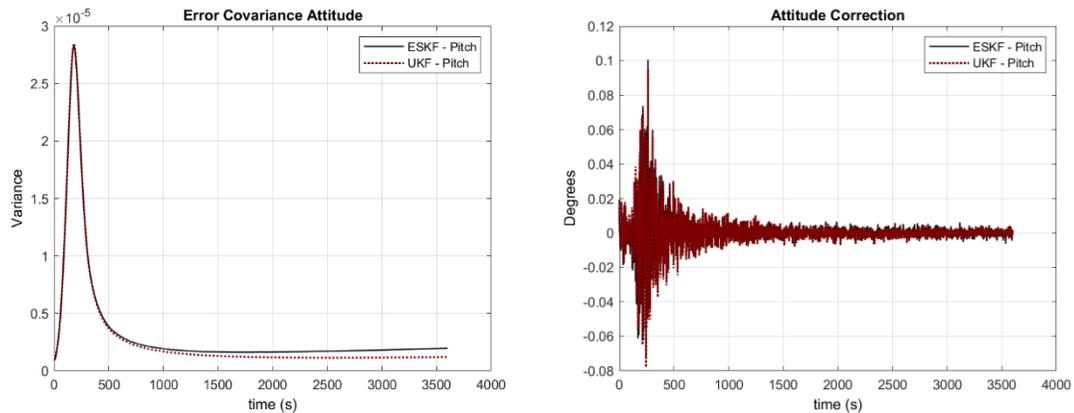


Figure 4-48 – ESKF and UKF variance (left) and the residuals (right) for the pitch error state

The overall behaviour shows that the ESKF is a little better to determine the states that are not measured and show the same performance for the states that are measured.

However, an important note must be considered. The system was tuned to present an optimal behaviour using the ESKF system, and the parameters used for the process and measurement noise covariance matrix were passed on to the UKF. By doing slight changes in the process noise covariance matrix, it was possible to find slightly better results for the UKF than for the ESKF.

4.6.2. System with coloured noise

Considering the system with coloured noise in the IMU, the results are presented only for the dimension that gives the highest deviation for each state. Figure 4-49 shows the ESKF and UKF comparison for the attitude error. Figure 4-50 presents the comparison for the position error. Figure 4-51 presents the comparison for the velocity error. And Figure 4-52 shows the comparison for the gyroscope bias and accelerometer bias determination.

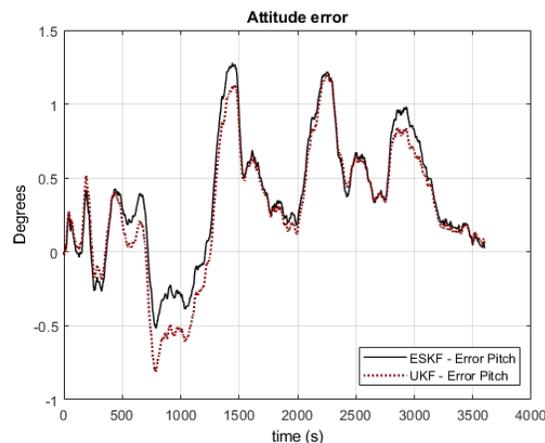


Figure 4-49 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude

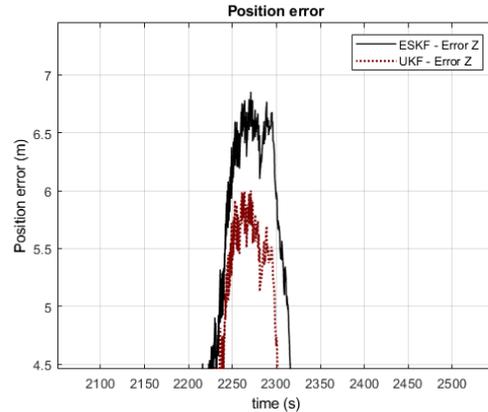
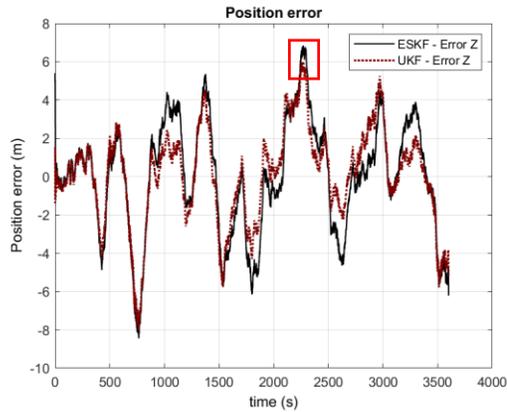


Figure 4-50 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and zoom in the peak occurred around 2300 seconds (right)

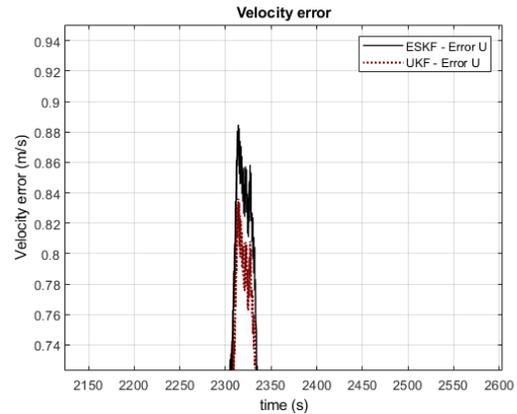
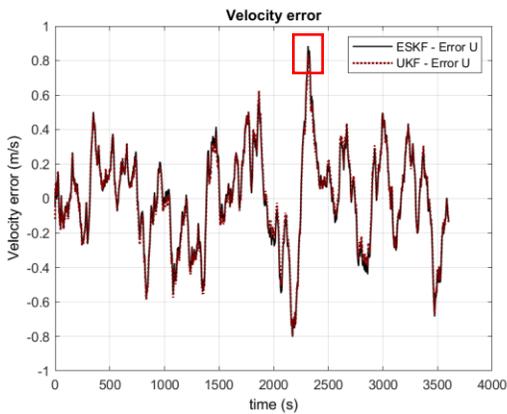


Figure 4-51 - Error between the true-ground and the INS/GNSS integration for the U velocity (left) and zoom in the peak occurred around 2300 seconds (right)

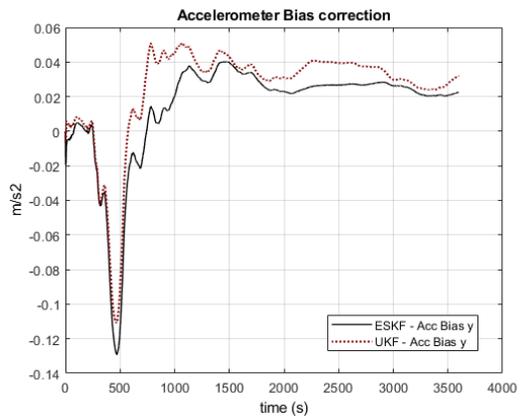
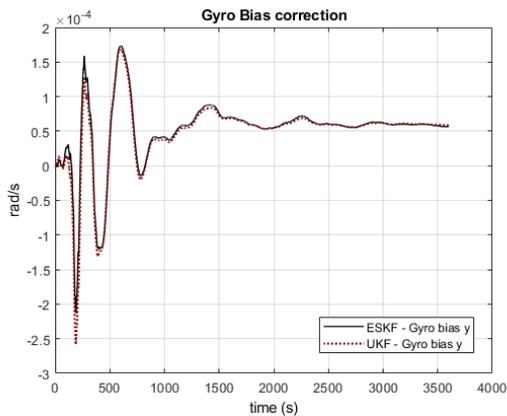


Figure 4-52 - Accelerometer bias (left) and Gyroscope bias (right) determination

It is possible to observe that for all states, the UKF present slightly better results than the results for the ESKF. For the attitude, after the system achieved the steady-state, the UKF shows a smooth behaviour, with small peaks and with the error always remained closer to zero.

For the position and velocity errors, it is also possible to observe that the values of the error peaks were smaller for the UKF, as it is emphasized in the zoom of Figure 4-50 and Figure 4-51.

For the error covariance matrix and the residuals, the states that show a slight difference between ESKF and UKF are the attitude and the accelerometer bias correction. Figure 4-53 shows the error covariance and the residuals for the pitch attitude, where it is possible to observe the same behaviour showed for the white noise system, in which the pitch error variance for the UKF reach a small value when compared to the ESKF. On the other side, the residual for both UKF and ESKF are visually identical.

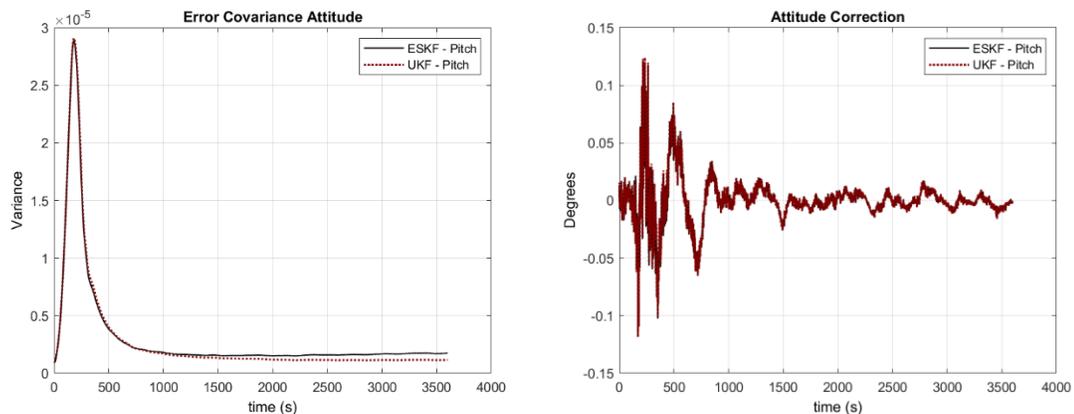


Figure 4-53 – ESKF and UKF variance (left) and the residuals (right) for the pitch error state

The overall behaviour shows that the UKF present slightly better performance than the ESKF for all states when coloured noise is introduced. This indicates that the UKF is less sensitive to the disturbances caused by the coloured noise.

4.6.3. System with coloured noise and FL correction

Considering the system with coloured noise in the IMU and the Fuzzy Logic corrections, as described in section 3.3, the results are presented only for the dimension that gives the highest deviation for each state. Figure 4-54 shows the ESKF and UKF comparison for the attitude error, Figure 4-55 presents the comparison for the position error, Figure 4-56 presents the comparison for the velocity error, and Figure 4-57 present the comparison for the gyroscope bias and accelerometer bias determination.

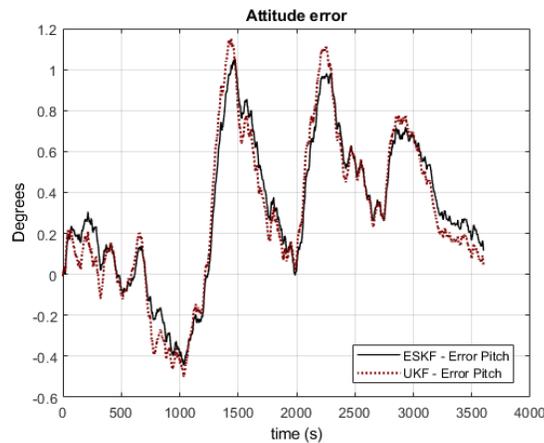


Figure 4-54 - Error between the true-ground and the INS/GNSS integration for the pitch Attitude

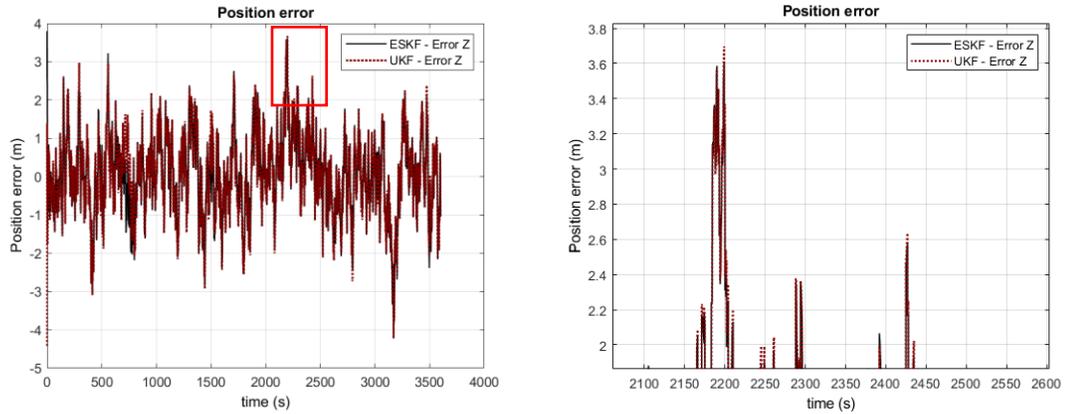


Figure 4-55 - Error between the true-ground and the INS/GNSS integration for the Z position (left) and zoom in the peaks occurred around 2300 seconds (right)

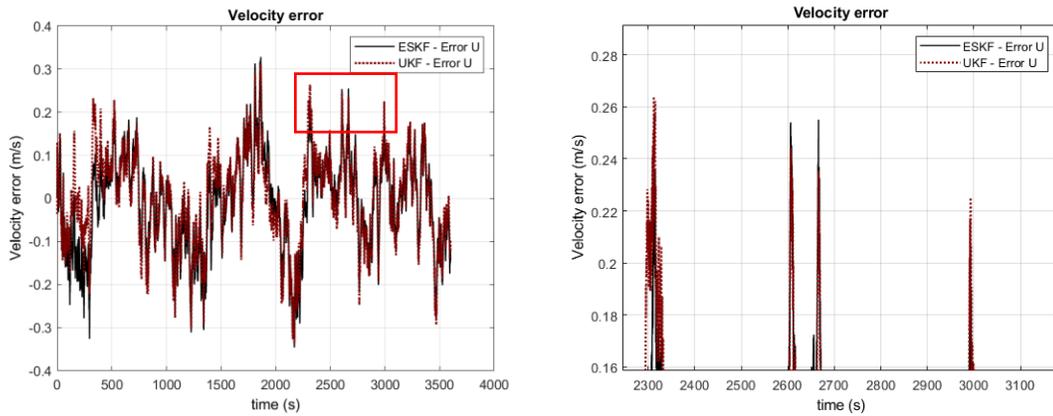


Figure 4-56 - Error between the true-ground and the INS/GNSS integration for the U velocity (left) and zoom in the peaks occurred around 2700 seconds (right)

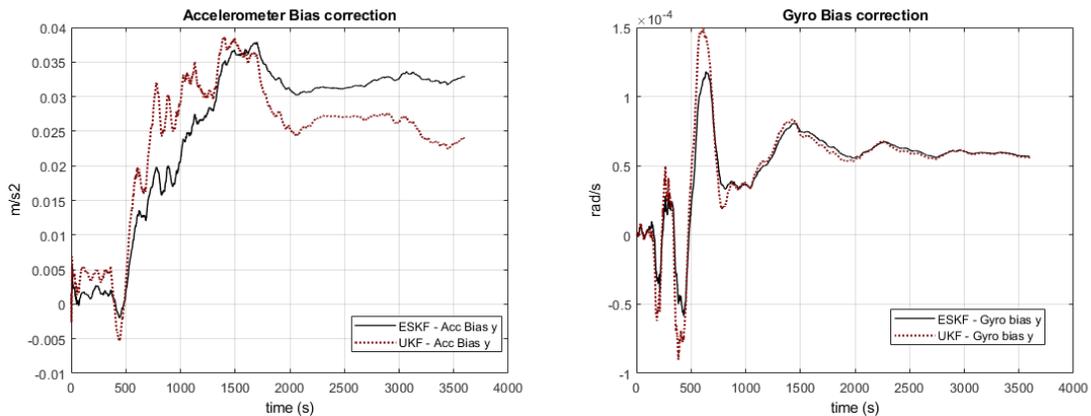


Figure 4-57 - Accelerometer bias (left) and Gyroscope bias (right) determination

The fuzzy logic is applied using the same membership functions, and the same values in both UKF and ESKF systems aimed to correct the large error bound caused by the introduction of the coloured noise. The comparison between these two KF implementations shows that the FL plays almost the same effect in both systems, bringing the true-ground error to a lower bound. By Figure 4-54 through Figure 4-57, it is possible to see that both systems present similar behaviour with just some small differences.

For the attitude, after the system achieved the steady-state, the ESKF shows a smooth behaviour, with small peaks and with the error always remained closer to zero compared to the UKF implementation.

For the position and velocity errors, both UKF and ESKF shows a similar behaviour, in which the error peaks achieved smaller values sometimes for ESKF and other times for the UKF implementation. Therefore, it is not possible to determine which one brings the best solution for the position and velocity states.

The gyroscope and accelerometer bias determination shows that the UKF faces a higher instability in the starting. However, achieve the same final solution for the gyroscope bias determination and a slightly better solution for the accelerometer bias determination. As it is possible to see by Figure D 16, the UKF error covariance for these states takes longer to achieve the steady-state and also reach to a higher value when compared to the ESKF.

Figure 4-58 shows the variance and the residuals for the pitch error state, where it is possible to observe that the error covariance for the UKF present a higher instability at the beginner but reach the same final value when compared to the ESKF. A higher error covariance means that the Kalman gain will also be higher for the UKF. Therefore, as

expected, the residuals for the UKF are slightly more prominent in the beginning of the navigation and virtually the same after the system achieve the steady-state.

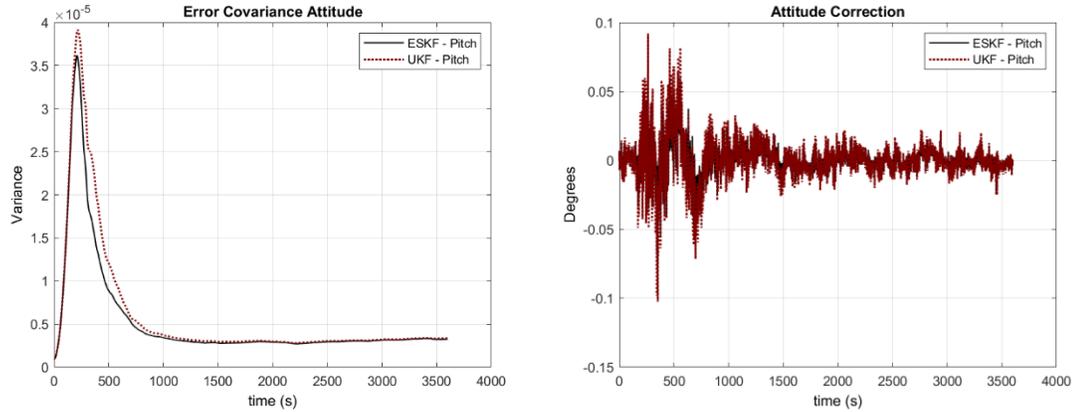


Figure 4-58 – ESKF and UKF error variance (left) and the residuals (right) for the pitch error state

The overall behaviour shows that the ESKF is a little better for correcting the states that are not measured and show the same behaviour for the states that are measured.

These results corroborate to show how the UKF is less sensitive to disturbances when compared to the ESKF. If on the one hand, this characteristic makes the UKF have a better result when coloured noise is present, on the other hand, its stability makes that the Fuzzy Logic correction plays less effect on the UKF solution, considering that the same methodology is used for both ESKF and UKF solutions.

Therefore, for the UKF achieves a better solution, a Fuzzy Logic that corrects the system in a slight weighty way shall be considered.

4.7. Thesis Limitations

The presented thesis addressed the navigation degradation caused by the introduction of coloured noise in a navigation system that uses KF for sensor fusion. Although different scenarios were considered, it is essential to determine the limitations that the proposed methodology has for real applications.

- Dynamics of the system: In the present work, it was considered a navigation profile that is changing slowly during the time, showing a low dynamic characteristic. This was done in order to emphasize the effect of the coloured noise in the system. For a high dynamic system, the Kalman Filter would need to be retuned, and the values for the membership function used in the Fuzzy Logic would also need to be chosen accordingly. However, it is expected that the proposed methodology will still be valid for a highly dynamic system.
- Real-time application: The proposed FLAC methodology uses one Fuzzy Logic implementation for each measurement state in the KF. Although both the Fuzzy Logic and the KF has been extensively used in a real-time application, the Fuzzy Logic application in parallel together with a KF implementation can make the system highly computational demanding, making it difficult for real-time implementation. One alternative is to use the FLAC to characterize the system and to define the best Kalman Gain value for the system.
- Noise characteristics: The real noise in the IMU is challenging to be simulated and will be different for different sensors and different frequencies. Bias instability in sensors like gyroscope can be represented by a power spectrum given by $1/f^\alpha$ where $\alpha \in (0,2)$, though the α values are usually close to 1 [56]. In the case of

accelerometers sensor, the coloured noise can present a power spectrum of $1/f^\alpha$ where $\alpha \in (0,3)$. However, in this work, it was assumed that the $\frac{1}{f}$ flicker noise is dominant for all sensors over the full frequency range in the IMU. If the real noise characteristic differs largely from this assumption, both the KF and the Fuzzy Logic must be retuned to represent the real noise situation.

- **Sensor accuracy:** In the present work it was considered a costumer grade GNSS and a tactical grade IMU. Especially for the GNSS it was considered a standard deviation of $\sigma_p = 5m$ in the position and of $\sigma_v = 0.1m/s$ in the velocity. The accuracy of the final navigation solution may be enhanced by using high precision GNSS system which utilizes Carrier Phase measurements or by utilizing two or more GNSS antennas in the same platform to obtain a partial attitude solution [57].
- **System stability:** Usually, the stability of adaptive control systems should be analytically proved to allow its implementation. Although different methodologies such as Lyapunov's direct and indirect method, describing function, and Popov's method have been used to prove the stability of FLC [58], the use of Fuzzy Logic Adaptive Control has been avoided for applications where no risk should be run. The stability problem was not addressed in the present work.

5. Conclusions

This work developed a methodology to deal with a situation where the navigation solution determined by an Inertial Navigation System (INS) coupled with a Global Navigation Satellite System (GNSS) is presenting sub-optimal behaviour due to the presence of coloured noise in the Inertial Measurement Unit (IMU).

First, it was shown the substantial effect caused in the INS-alone navigation solution by introducing small errors in the IMU. This occurs once the INS suffer degradation in long-term navigation as the errors presented in the sensor are accumulated over time and integrated through the navigation equations.

Secondly, an optimal INS/GNSS integration was developed using Error-States Kalman Filter (ESKF) and Unscented Kalman Filter (UKF), where white noises were the only type of noise presented in the system. It was confirmed the improvement in the attitude, position, and velocity determination using the Kalman Filter (KF) solution rather than the INS or GNSS alone solutions.

Thirdly, it was shown the effect in the KF optimum solution caused by the introduction of coloured noise in the IMU. Although this introduction brings slight changes in the INS-alone solution, the coloured noise makes the KF system achieve a solution that converges to a larger bound, showing a sub-optimal behaviour.

The analysis done with the ESKF system and with coloured noise showed the best approach for the Fuzzy Logic (FL) implementation for the adaptive KF. A novel solution where the FL observes the error covariance and the residuals from the KF states, and apply an exponential weighted correction when the error covariance matrix is higher than expected,

and a process noise injection when the system is in steady-state, and the residuals are higher than expected, showed to be the best approach.

Applying the Fuzzy Logic adaptive ESKF solution in the optimal KF solution with only white noise, it was possible to observe that, when the FL is used, all states are slightly damaged. Once the ESKF was tuned to present an optimal solution when only white noise is present, it is expected that by the use of FL, which mainly enlarge the process noise covariance matrix, the final solution would be degraded.

Further, the Fuzzy logic adaptive ESKF and UKF solution was applied to the system with coloured noise in the IMU. It was possible to observe a substantial improvement when the FL is used. In steady-state, the maximum error for the position was decreased from ± 8 m to ± 4 m, the velocity from ± 0.8 m/s to ± 0.3 m/s, and the attitude from $\pm 2.8^\circ$ to $\pm 2^\circ$. The overall results show that it was possible to reduce the error caused by the coloured noise by approximately half of the value.

The slight degradation in the attitude, position, and velocity states caused by the use of FL in an optimal KF may be accepted when considering the substantial improvement caused by the FL in a sub-optimal KF solution. This is important when considering practical application where the optimum tuning is unknown and/or where it varies over time.

Therefore, the novel proposed FLAC implementation shows a potential to be used in systems where the IMU shows coloured noise, mainly for the IMUs fabricated using MEM technology.

Finally, a comparison between the results obtained by the ESKF and UKF was made for the system with white noise, with coloured noise, and with the Fuzzy Logic correction applied to the coloured noise system.

The comparison showed that both implementation approaches have similar behaviour, with the UKF been slightly less sensitive to disturbances when compared to the ESKF. If on the one hand, this characteristic makes the UKF have a better result when coloured noise is present, on the other hand, its stability makes that the Fuzzy Logic correction plays less effect on the UKF solution, considering that the same methodology is used for both ESKF and UKF solutions.

Therefore, for the UKF achieves a better solution, a Fuzzy Logic that corrects the system in a slight weighted way shall be considered.

6. References

- [1] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Second Edi. 2013.
- [2] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB®: Fourth Edition*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2014.
- [3] J. Z. Sasiadek, "Sensor fusion," *Annu. Rev. Control*, vol. 26 II, no. 2, pp. 203–228, 2002.
- [4] M. Pittelkau, "Attitude determination Kalman filter with a 1/f flicker noise gyro model," in *26th International Technical Meeting of the Satellite Division of the Institute of Navigation*, 2013, pp. 2143–2160.
- [5] M. Kirkko-Jaakkola, J. Collin, and J. Takala, "Bias prediction for MEMS gyroscopes," *IEEE Sens. J.*, vol. 12, no. 6, pp. 2157–2163, 2012.
- [6] J. M. Barrett, "Analyzing and Modeling Low-Cost MEMS IMUs for Use in an Inertial Navigation System," p. 95, 2014.
- [7] J. Z. Sasiadek and Q. Wang, "Fuzzy Adaptive Kalman Filtering for INS/GPS Data Fusion and Accurate Positioning," *IFAC Proc. Vol.*, vol. 34, no. 15, pp. 410–415, 2001.
- [8] P. J. Escamilla-Ambrosio and N. Mort, "Development of a fuzzy logic-based adaptive Kalman filter," *2001 Eur. Control Conf. ECC 2001*, no. 11, pp. 1768–1773, 2001.
- [9] D. Loebis, R. Sutton, J. Chudley, and W. Naeem, "Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system," *Control Eng. Pract.*, vol. 12, no. 12 SPEC. ISS., pp. 1531–1539, 2004.
- [10] M. Ding and Q. Wang, "An integrated navigation system of NGIMU/ GPS using a fuzzy logic adaptive Kalman filter," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 3613 LNAI, pp. 812–821.
- [11] C. H. Tseng, S. F. Lin, and D. J. Jwo, "Fuzzy adaptive cubature kalman filter for integrated navigation systems," *Sensors (Switzerland)*, vol. 16, no. 8, 2016.
- [12] A. L. da Silva and J. J. da Cruz, "Fuzzy adaptive extended Kalman filter for UAV INS/GPS data fusion," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 38, no. 6, pp. 1671–1688, 2016.
- [13] D.T.~Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. Automat. Contr.*, vol. 10, no. 4, pp. 434–439, 1965.
- [14] R. K. Mehra, "Approaches to Adaptive Filtering," *IEEE Trans. Automat. Contr.*, vol. 17, no. 5, pp. 693–698, 1972.
- [15] B. Y. Hu F., Lü T., "Application of improved adaptive Kalman filtering in

- SINS/GPS integrated navigation,” *Comput. Eng. Appl.*, vol. 54, pp. 253–257, 2018.
- [16] Y. Shi, “Adaptive UKF method with applications to target tracking,” *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 37, no. 6. pp. 755–759, 2011.
- [17] R. G. Valenti, I. Dryanovski, and J. Xiao, “Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs,” *Sensors (Switzerland)*, vol. 15, no. 8, pp. 19302–19330, 2015.
- [18] H. A. P. Blom, “An efficient filter for abruptly changing systems,” *CDC. IEEE*, pp. 656–658, 1984.
- [19] Q. Fan *et al.*, “Improved pedestrian dead reckoning based on a robust adaptive Kalman filter for indoor inertial location system,” *Sensors (Switzerland)*, vol. 19, no. 2, 2019.
- [20] Z. Zhe, W. Jian-bin, S. Bo, and T. Guo-feng, “Adaptive Complementary Filtering Algorithm for IMU Based on MEMS,” pp. 5409–5416, 2020.
- [21] J. Qiu *et al.*, “Centralized Fusion Based on Interacting Multiple Model and Adaptive Kalman Filter for Target Tracking in Underwater Acoustic Sensor Networks,” *IEEE Access*, vol. 7, pp. 25948–25958, 2019.
- [22] B. Han, H. Huang, L. Lei, C. Huang, and Z. Zhang, “An Improved IMM Algorithm Based on STSRCKF for Maneuvering Target Tracking,” *IEEE Access*, vol. 7, pp. 57795–57804, 2019.
- [23] S. Sarkka and A. Nummenmaa, “Recursive noise adaptive Kalman filtering by variational Bayesian approximations,” *IEEE Trans. Automat. Contr.*, vol. 54, no. 3, pp. 596–600, 2009.
- [24] N. Davari and A. Gholami, “Variational Bayesian adaptive Kalman filter for asynchronous multirate multi-sensor integrated navigation system,” *Ocean Eng.*, vol. 174, no. May 2018, pp. 108–116, 2019.
- [25] A. H. Mohamed and K. P. Schwarz, “Adaptive Kalman filtering for INS/GPS,” *J. Geod.*, vol. 73, no. 4, pp. 193–203, 1999.
- [26] B. Singh and D. Dahlhaus, “Weighted Robust Sage-Husa Adaptive Kalman Filtering for Angular Velocity Estimation,” *Signal Process. - Algorithms, Archit. Arrange. Appl. Conf. Proceedings, SPA*, vol. 2019–September, pp. 71–76, 2019.
- [27] Y. Ning, J. Wang, H. Han, X. Tan, and T. Liu, “An optimal radial basis function neural network enhanced adaptive Robust Kalman filter for GNSS/INS integrated systems in complex urban areas,” *Sensors (Switzerland)*, vol. 18, no. 9, 2018.
- [28] V. Pesce, S. Silvestrini, and M. Lavagna, “Radial basis function neural network aided adaptive extended Kalman filter for spacecraft relative navigation,” *Aerosp. Sci. Technol.*, vol. 96, p. 105527, 2019.
- [29] B. Zhao, D. Zhu, T. Xi, C. Jia, S. Jiang, and S. Wang, “Convolutional neural network and dual-factor enhanced variational Bayes adaptive Kalman filter based indoor localization with Wi-Fi,” *Comput. Networks*, vol. 162, 2019.

- [30] S. Yazdkhasti, J. Z. Sasiadek, and S. Ulrich, "Performance enhancement for GPS/INS fusion by using a fuzzy adaptive unscented Kalman filter," *2016 21st Int. Conf. Methods Model. Autom. Robot. MMAR 2016*, pp. 1194–1199, 2016.
- [31] M. Gessesse and M. M. Atia, "Multi-sensor attitude and heading reference system using genetically optimized kalman filter," *Midwest Symp. Circuits Syst.*, vol. 2018–Augus, pp. 460–463, 2019.
- [32] W. Khalaf, I. Chouaib, and M. Wainakh, "Novel adaptive UKF for tightly-coupled INS/GPS integration with experimental validation on an UAV," *Gyroscopy Navig.*, vol. 8, no. 4, pp. 259–269, 2017.
- [33] C. T. Fraser, "Adaptive Extended Kalman Filtering Strategies for Autonomous Relative Navigation of Formation Flying Spacecraft," 2018.
- [34] J. Z. Sasiadek and Q. Wang, "Low Cost Automation using INS/GPS Data Fusion for Accurate Positioning," *Robotica*, vol. 21, pp. 255–260, 2003.
- [35] D. J. Jwo, C. F. Yang, C. H. Chuang, and T. Y. Lee, "Performance enhancement for ultra-tight GPS/INS integration using a fuzzy adaptive strong tracking unscented Kalman filter," *Nonlinear Dyn.*, vol. 73, no. 1–2, pp. 377–395, 2013.
- [36] F. L. Lewis, *Optimal estimation : with an introduction to stochastic control theory* . New York: Wiley, 1986.
- [37] United States. National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems.," 1997.
- [38] J. B. Kuipers, *Quaternions and rotation sequences : a primer with applications to orbits, aerospace, and virtual reality* . Princeton, N.J: Princeton University Press, 1999.
- [39] N. Barbour and G. Schmidt, "Inertial sensor technology trends," *IEEE Sens. J.*, vol. 1, no. 4, pp. 332–339, 2001.
- [40] N. J. Kasdin, "Discrete Simulation of Colored Noise and Stochastic Processes and $1/f\alpha$ Power Law Noise Generation," *Proc. IEEE*, vol. 83, no. 5, pp. 802–827, 1995.
- [41] G. Xu, *GPS Theory, Algorithms and Applications* , Second Edi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [42] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Fluids Eng. Trans. ASME*, vol. 82, no. 1, pp. 35–45, 1960.
- [43] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "New approach for filtering nonlinear systems," *Proc. Am. Control Conf.*, vol. 3, no. June, pp. 1628–1632, 1995.
- [44] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Signal Process. Sens. Fusion, Target Recognit. VI*, vol. 3068, p. 182, 1997.
- [45] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation," in *AIAA*

Guidance, Navigation, and Control Conference, 2011.

- [46] Joan Sola, “Quaternion Kinematics for Error-State KF,” 2017. [Online]. Available: <http://www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf>.
- [47] M. Grewal and A. Andrews, “Kalman theory, theory and practice using MATLAB,” pp. 417–418, 2008.
- [48] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation With an Introduction to Stochastic Control Theory*. CRC Press, 2017.
- [49] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. CRC Press, 2008.
- [50] D. S. Hooda, *Fuzzy logic models and fuzzy control an introduction*. Oxford, [England: Alpha Science International Limited, 2017.
- [51] T. J. Ross, *Fuzzy logic with engineering applications*, Fourth edi. Chichester, West Sussex, England: Wiley, 2017.
- [52] Mathworks, “Aerospace Toolbox User’s Guide.” 2019.
- [53] Honeywell, “HG1700 Inertial Measurement Unit,” 2018.
- [54] M. Petovello, “Real -time integration of a tactical-grade IMU and GPS for high - accuracy positioning and navigation,” ProQuest Dissertations Publishing, 2003.
- [55] T. Ford and J. Neumann, “OEM4 Inertial : An Inertial / GPS Navigation System on the OEM4 Receiver,” vol. 3, pp. 150–168, 1998.
- [56] K. Jerath, S. Brennan, and C. Lagoa, “Bridging the gap between sensor noise modeling and sensor characterization,” *Meas. J. Int. Meas. Confed.*, vol. 116, no. September 2017, pp. 350–366, 2018.
- [57] M. Blois, “Interpolation of Experimental Error Statistics for Mission Design of GPS Partial Attitude Determination,” 2019.
- [58] J. M. Giron-Sierra and G. Ortega, “A survey of stability of fuzzy logic control with aerospace applications,” *IFAC Proc. Vol.*, vol. 15, no. 1, pp. 67–78, 2002.

Appendix A - Exponential Weighted UKF derivation

Let set the model covariance matrix as weighted covariance, as follow:

$$\begin{aligned} R_k &= R \alpha^{-2(k+1)} \\ Q_k &= Q \alpha^{-2(k+1)} \end{aligned} \quad (\text{A1})$$

Where α is the weighted parameter. Following the UKF methodology described in section 2.5.3, the error covariance develops by:

$$P_k^- = \frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \alpha^{-2(k+1)} \quad (\text{A2})$$

Or

$$P_k^- \alpha^{-2(k+1)} = \frac{\alpha^{-2(k+1)}}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \quad (\text{A3})$$

Considering $P_k^\alpha = P_k^- \alpha^{2k}$ the weighted covariance matrix is now given as follows:

$$P_k^\alpha = \alpha^2 \frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (x_k^{-(i)} - \hat{x}_k^-)^T + Q_{INS} \quad (\text{A4})$$

With the initial condition $P_0^\alpha = P_0$.

The measurement innovations covariance is given by:

$$P_{zz_k}^- = C_k^- = \frac{1}{2n} \sum_i^{2n+1} (\delta z_k^{-(i)} - \delta \hat{z}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T + R_k \alpha^{-2(k+1)} \quad (\text{A5})$$

And the cross-covariance, by:

$$P_{xz_k}^- = \left[\frac{1}{2n} \sum_i^{2n+1} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right] \quad (\text{A6})$$

Now, considering $P_{zz}^\alpha = P_{zz_k}^- \alpha^{2k}$ and $P_{xz}^\alpha = P_{xz_k}^- \alpha^{2k}$, the Kalman Gain is now given as follows:

$$K_k = \frac{P_{xz_k}^-}{P_{zz_k}^-} = \frac{P_{xz}^\alpha}{P_{zz}^\alpha}$$

$$K_k = \frac{\alpha^{2k} \left[\frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right]}{\alpha^{2k} \left(\frac{1}{2n} \sum_i^{2n} (\delta z_k^{-(i)} - \delta \hat{z}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right) + \frac{\alpha^{2k} R_k}{\alpha^{2(k+1)}}}$$
(A7)

The Kalman Gain is finally given by:

$$K_k = \frac{\left[\frac{1}{2n} \sum_i^{2n} (x_k^{-(i)} - \hat{x}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T \right]}{\frac{1}{2n} \sum_i^{2n} (\delta z_k^{-(i)} - \delta \hat{z}_k^-) (\delta z_k^{-(i)} - \delta \hat{z}_k^-)^T + \alpha^{-2} R_k}$$
(A8)

Appendix B- Study on the exponential weighted parameter

As discussed in section 2.6.1, Lewis [48] proposes the equations for an exponential weighted Kalman Filter.

The proposal of this Appendix is to vary the advantages to consider the weighted parameter α as a matrix rather than a single number, to allow weighting the position (α_p) and velocity (α_v) individually while not affecting the other states (attitude and feedback-errors). The reader can refer to section 2.6.1.1 for more details.

To do so, an ESKF implementation where the IMU presents a 1/f flicker noise was simulated considering three different situations. First, an ESKF system with no correction. Secondly, an ESKF system with a FLAC correction where a single number alpha is used. And finally, an ESKF system with a FLAC correction where a matrix alpha is used.

Figure B 1 presents the obtained results. The first column of Figure B 1 shows the system behaviour when 1/f flicker noise is dominant in the IMU, making the ESKF system reach a solution that converges to a large bound. In this case, the position presents a maximum error of +/- 5 m, the velocity a maximum error of +/- 0.7 m/s, and the attitude a maximum error of +/- 4°.

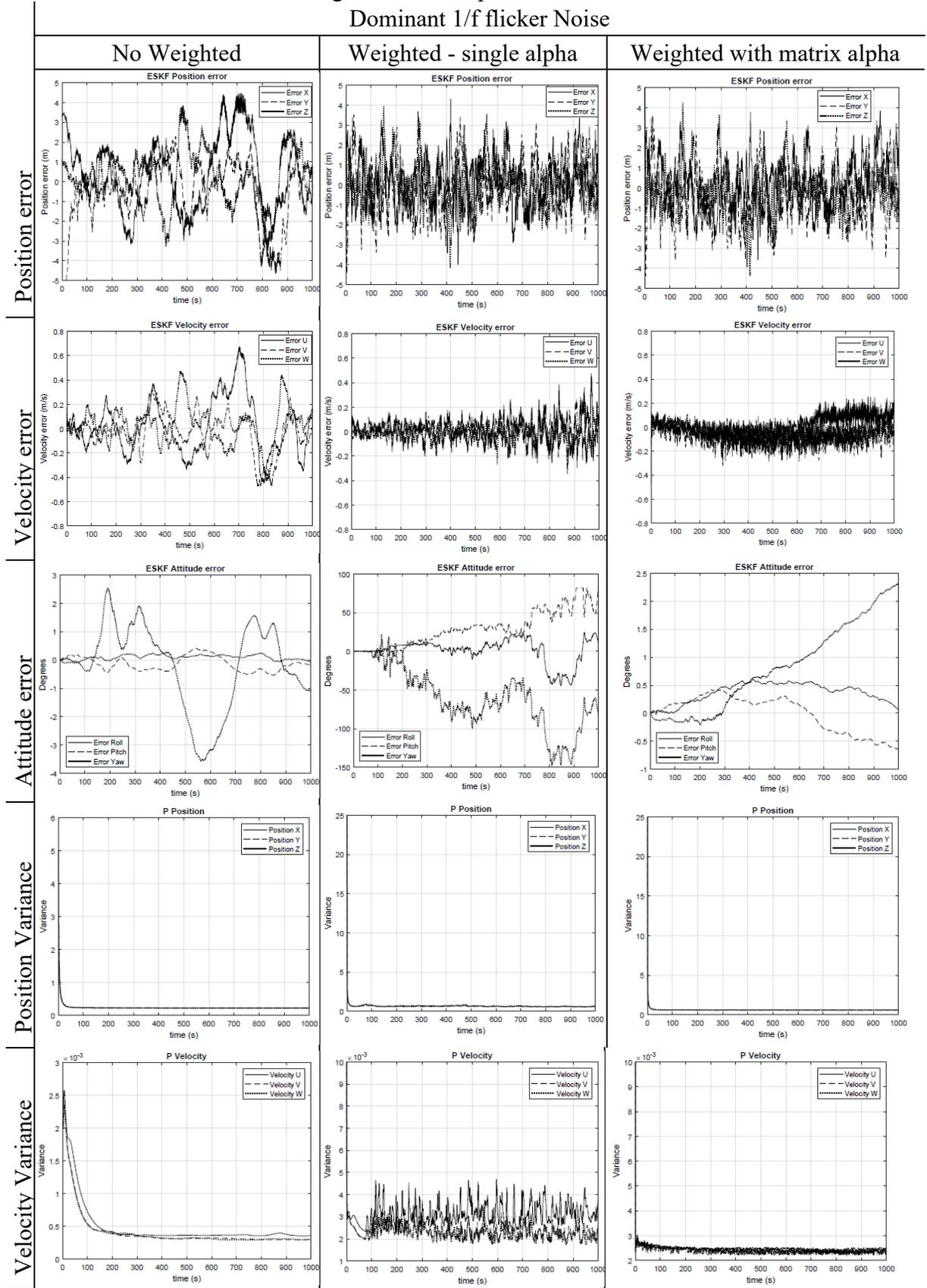
The second and third columns of Figure B 1 show the effect of applying FLAC correction using a constant alpha and a matrix alpha, respectively. For both cases, it is possible to observe an improvement when compared to the non-weighted solution previously described. The position presents a maximum error of +/- 4 m for both alphas and the velocity a maximum error of +/- 0.2 m/s for the matrix alpha and +/- 0.45 m/s for the

constant alpha. However, it is possible to observe a considerable difference in the attitude determination between both methods. For the constant alpha, a high disturbance in the predicted altitude and gyro bias correction is observed. This is caused once the alpha is determined by the residual and the variance for the position state. Therefore, using this weighting factor to adjust other states can lead to a wrong correction. Furthermore, the use of a constant alpha generated a high instability in the error covariance matrix, which is not desirable.

The proposed alpha in matrix form was confirmed as the best solution to deal with the coloured noise using FLAC weighted ESKF. This methodology leads to a final solution that was able to improve the navigation accuracy for all the states, preserving the stability of the error covariance matrix and, consequentially, the stability of the system.

Figure B 1- KF without correction, with FL weighted single alpha correction, and with FL weighted matrix alpha correction

Dominant 1/f flicker Noise



Appendix C - Study on the Fuzzy Logic approach

Appendix C.1 Identifying the Fuzzy Logic Inputs

In section 4.3, the residuals were identified as the Kalman filter states that best represent the high error bound achieved for the ESKF and UKF solutions when coloured noise is introduced in the system.

This appendix aims to identify how the FL should correct the KF solution. To do so, the residuals are compared against the error in the navigation states. The analyses are made for the dimension that presents the highest deviation for the attitude, showed in Figure C 1, for the position, showed in Figure C 2, and for the velocity, showed in Figure C 3. In these figures, the red line represents approximately the moment when the system achieved the steady-state and the red circles the points where the system achieves the higher error bound.

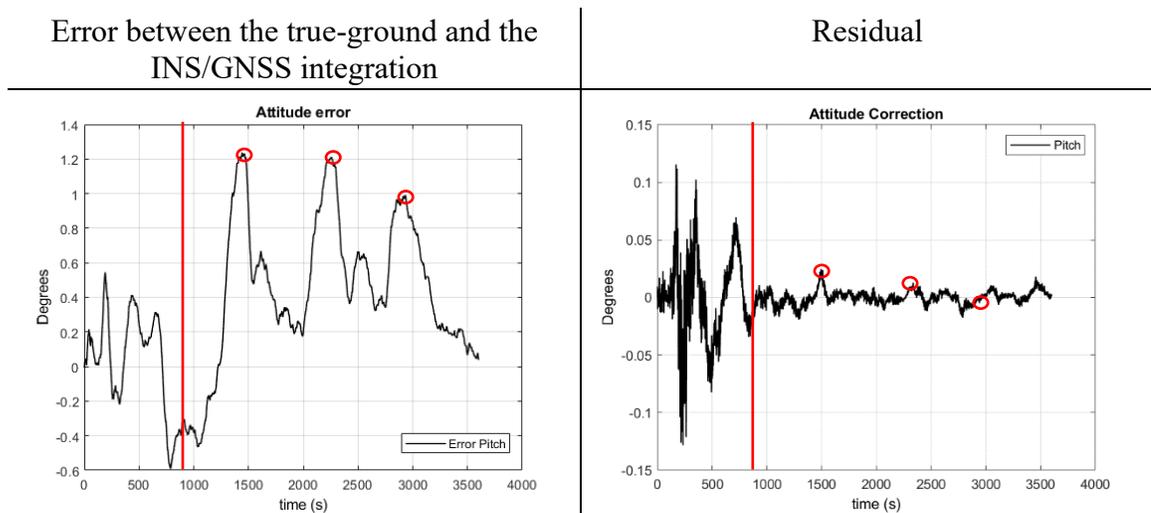


Figure C 1 - Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the Attitude pitch

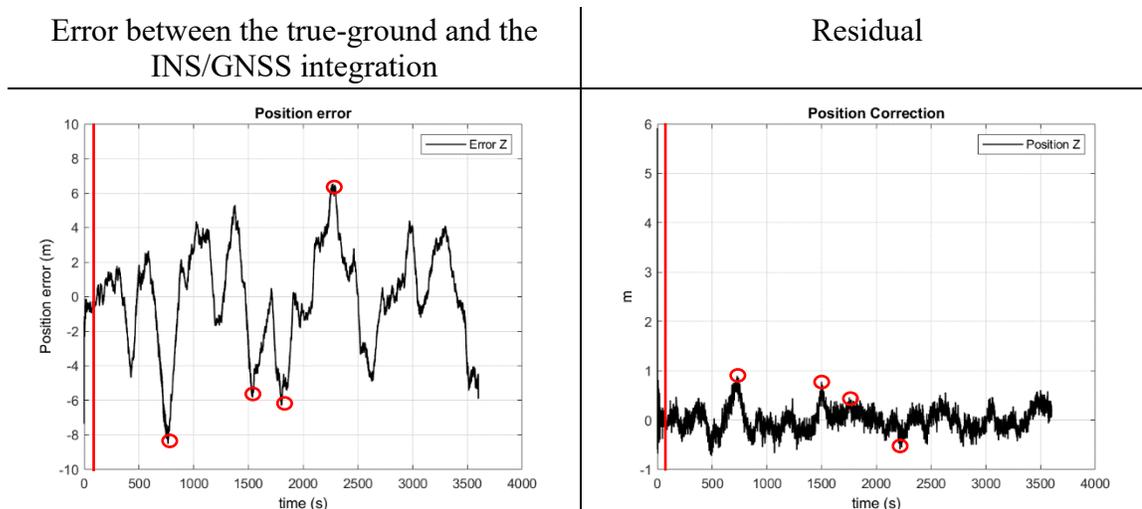


Figure C 2 - - Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the position Z

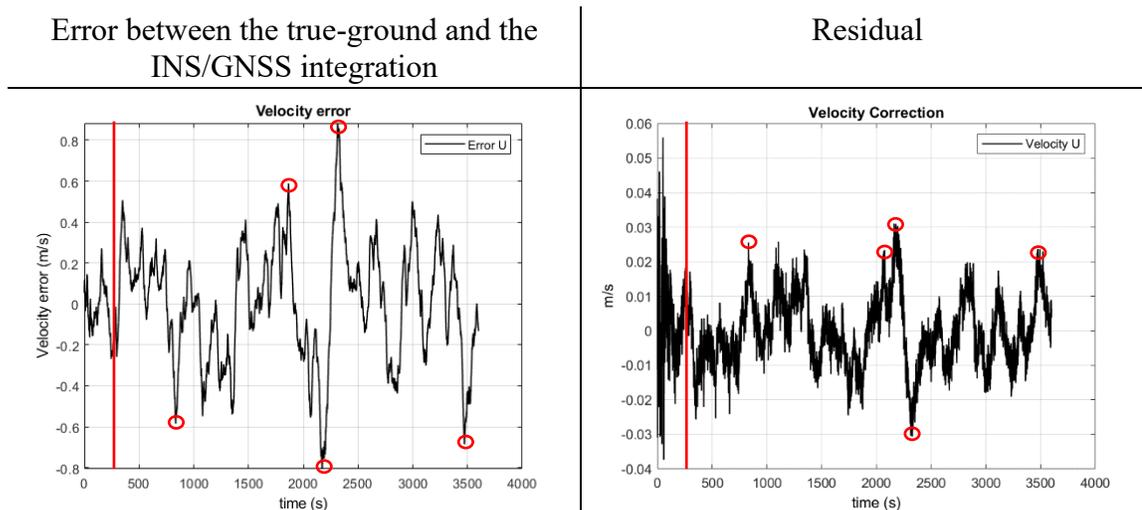


Figure C 3- Error between the true-ground and the INS/GNSS integration (left) and residuals (right) for the Velocity U

Figure C 1 to Figure C 3 shows that before the Kalman Filter achieves the steady-state, the residuals show high values. This is expected, once in the starting the variances of the state estimates have elevated values, allowing the Kalman Filter to make high corrections. This effect is also enhanced because the system is predicting the gyroscope and accelerometer bias at the beginning.

Therefore, the application of the FL must consider two stages, one in the system starts, where the residuals and the variances of the state estimates are elevated, and another one when the filter achieves the steady-state, where the residuals are high. Still, the variances of the state estimates are small. The identification of these two stages can be made by observing the error covariance matrix.

After the system achieves the steady-state, the analyses of the points where the system shows the high error bound indicates that for the attitude, the high errors not necessarily occur when the KF outputs a high residual. Therefore, the attitude was disregarded as an option to be observed by the FL.

About the position and velocity states, it is possible to observe in Figure C 2 and Figure C 3 a coupling between the navigation errors peaks and the residuals peaks. Thus, these two states were chosen to be observed by the FL.

It is worth noting that the states that will be observed by the FL are the measurement state, given by the error between the GNSS and INS.

Therefore, it was defined that the FL will observe the error covariance matrix to verify if the KF is in steady-state or not, and the residuals to identify the error peaks, for the position and velocity states.

Comparing the system with the white noise against the system with coloured noise, the values for the error covariance matrix and residuals for the fuzzification logic were identified as follows:

Error covariance matrix:

- Position: steady state when $P < 0.5$;
- Velocity: steady state when $P < 3 \times 10^{-4}$;

Residuals:

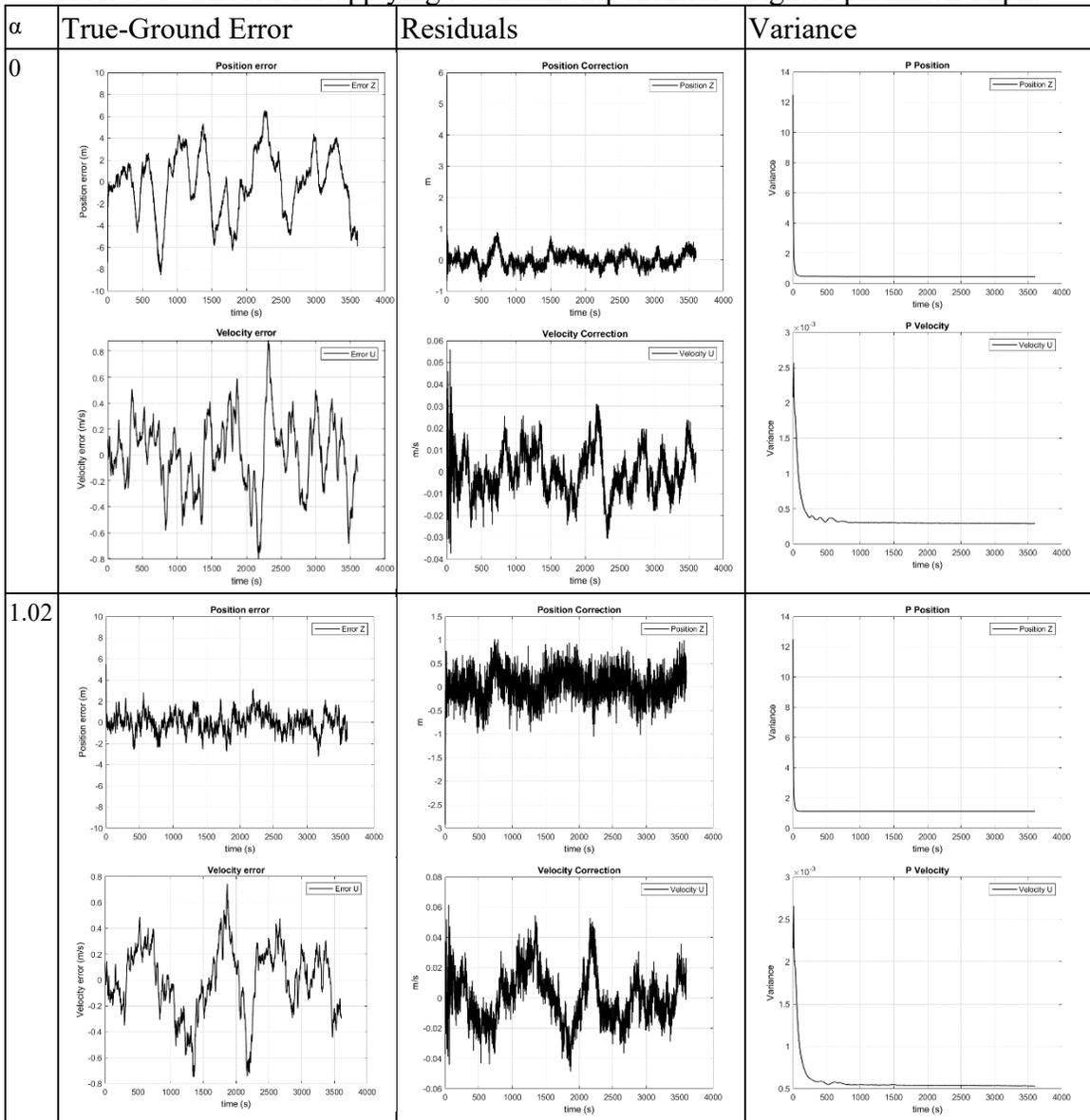
- Position: White noise system: $z < 0.5$. Coloured noise system: $z < 1$;
- Velocity: White noise system: $z < 0.01$. Coloured noise system: $z < 0.03$;

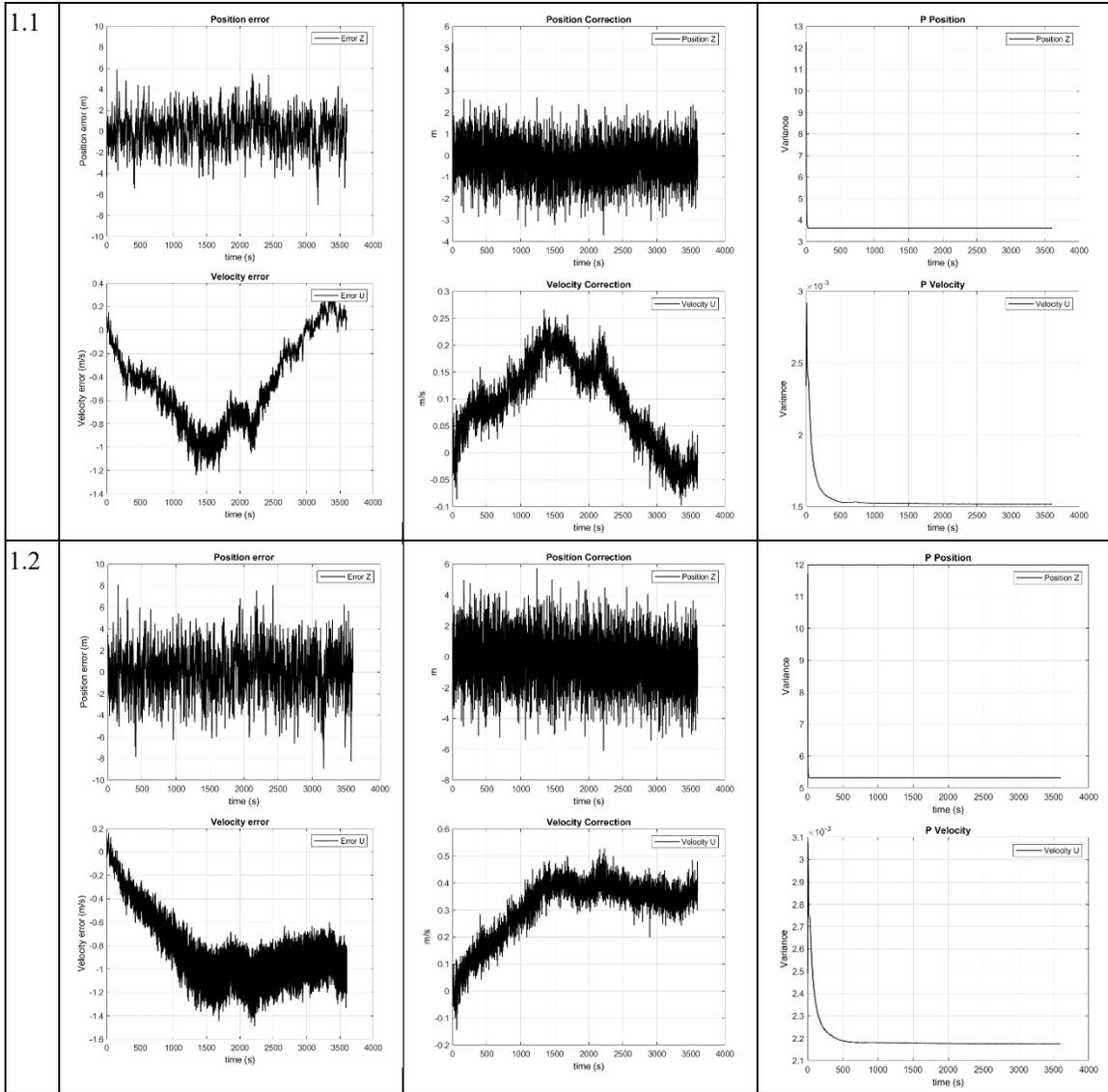
Appendix C.2 Applying the Corrector Factor

This section aims to identify the effect of applying a constant factor intended to correct the KF solution with coloured noise. To do so, two approaches were used, the exponential data weighting, as described in section 2.6.1, and the fictitious process noise injection, as described in section 2.6.2. The analyses were made for the ESKF with coloured noise in the IMU.

For the exponential data weighting, three values for the weighting parameter alpha were chosen based on [7], as 1.02, 1.1 and 1.2. The results are shown in Table C 1 for the velocity U and position Z, considering the true-ground error, residual and variances of the state estimates.

Table C 1 - Results for applying a constant exponential weighted parameter Alpha





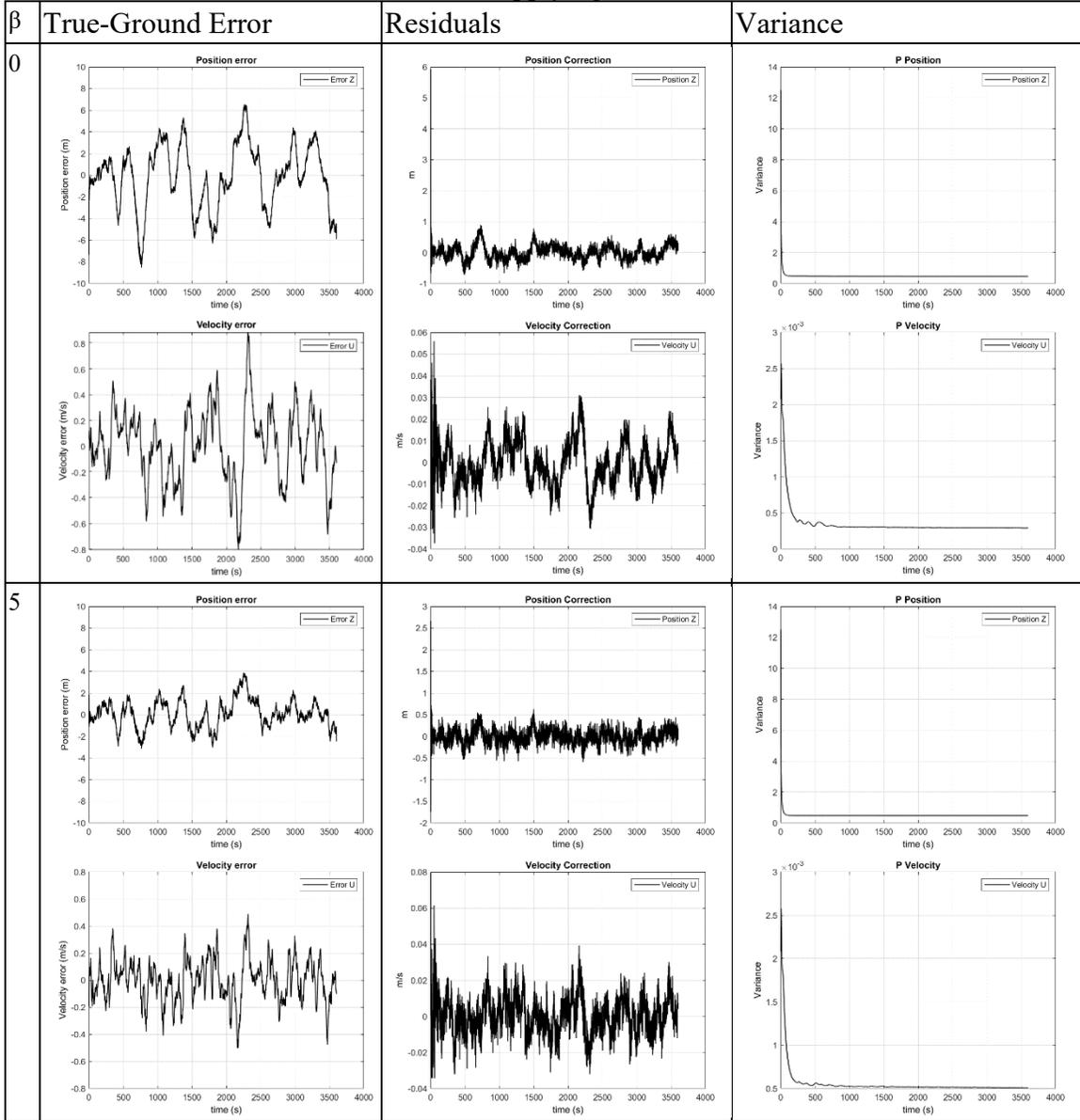
Repeating section 2.6.2, for the fictitious process noise injection, the following relation was used to apply the FL in the KF:

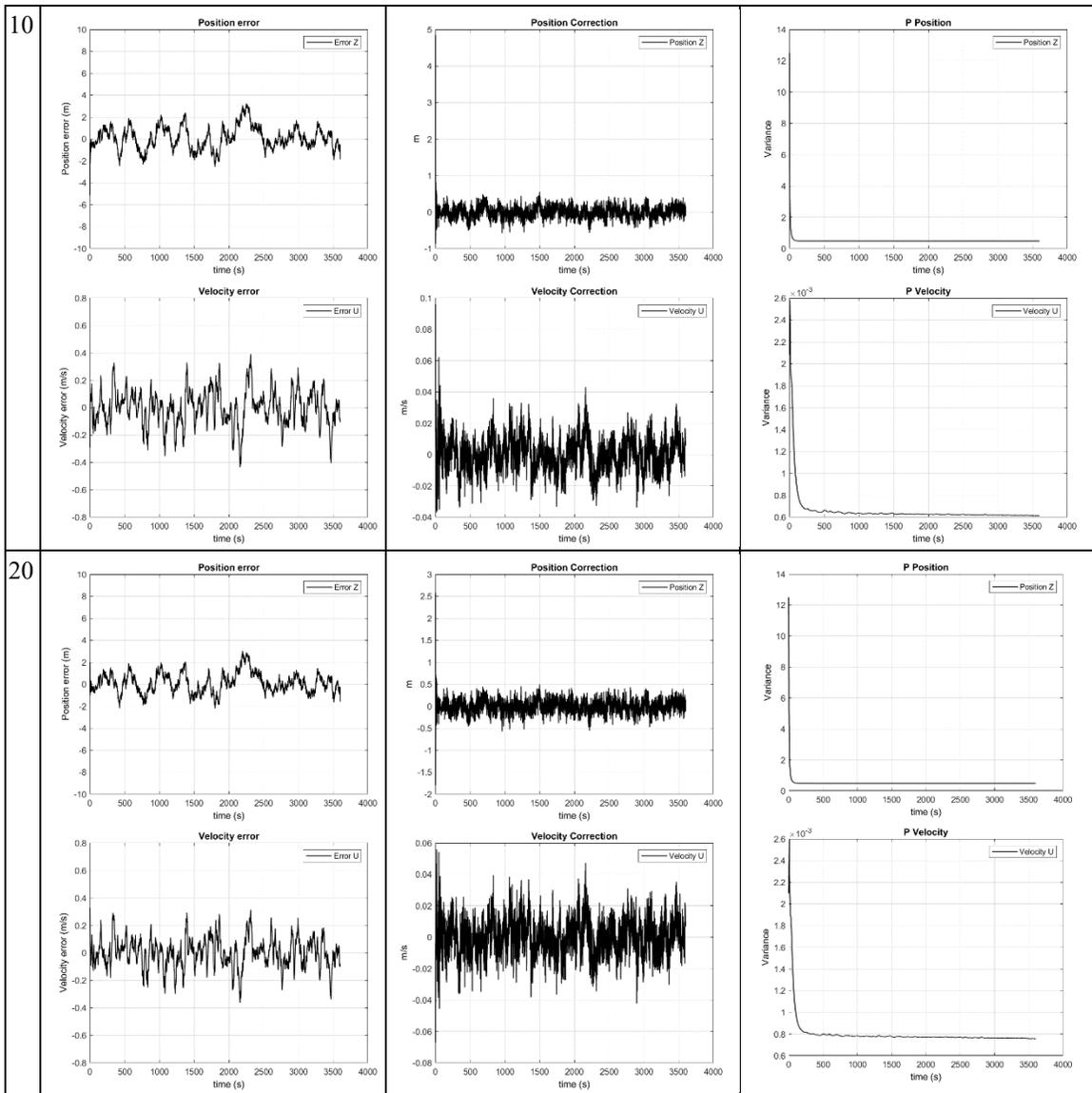
$$Q_k = Q_k + \beta Q_k \tag{2.69}$$

Where β is the FL output from the KF.

Three values for the noise injection parameter beta were chosen based on [11], as 5, 10 and 20. The results are shown in Table C 2 for the velocity U and position Z, considering the true-ground error, residual and variances of the state estimates.

Table C 2 - Results for applying a constant Beta value





The outcomes from Table C 1 and Table C 2 are listed below:

- Both methodologies improved the KF solution, reducing the true-ground error for all states when applied.
- The exponential weighted approach achieves the best solution for the position state, showing the smaller true-ground error and error covariance value for the 1.02 alpha.
- The exponential weighted approach showed a small improvement for the velocity for the 1.02 and 1.1 alpha, and a degraded result for 1.2 alpha.

- The exponential weighted approach decreased the time for the system achieves the steady-state and decreased the initial instability for the velocity error variance. However, the error variance was stabilized in a higher value.
- The injection noise approach improved the position and velocity true-ground error, showing significant improvement for the velocity.
- The injection noise approach achieved a smaller error covariance value when the system was in steady-state.

The overall conclusion is that the exponential weighted approach shows better results before the system achieve the steady-state when the error covariance is high. And the fictitious noise injection approach shows better results when the system is in steady-state.

Therefore, a hybrid solution where the Fuzzy Logic apply an exponential weighted correction when the error covariance matrix is high, and a fictitious noise injection when the system is in steady-state, shows to be the best approach.

Appendix D - ESKF and UKF comparison

This appendix shows the full results for the ESKF and UKF comparison. Discussion about these results can be found in section 0. The results are presented for the system with white noise, with coloured noise and with the Fuzzy Logic correction for the coloured noise system, considering the true-ground error, the variances of the state estimates, and the residuals.

Red lines were added to the maximum and minimum value registered in some of the results to facilitate the visualization of the differences between the ESKF and UKF.

Appendix D.1 White Noise

Appendix D.1.1 True-ground error

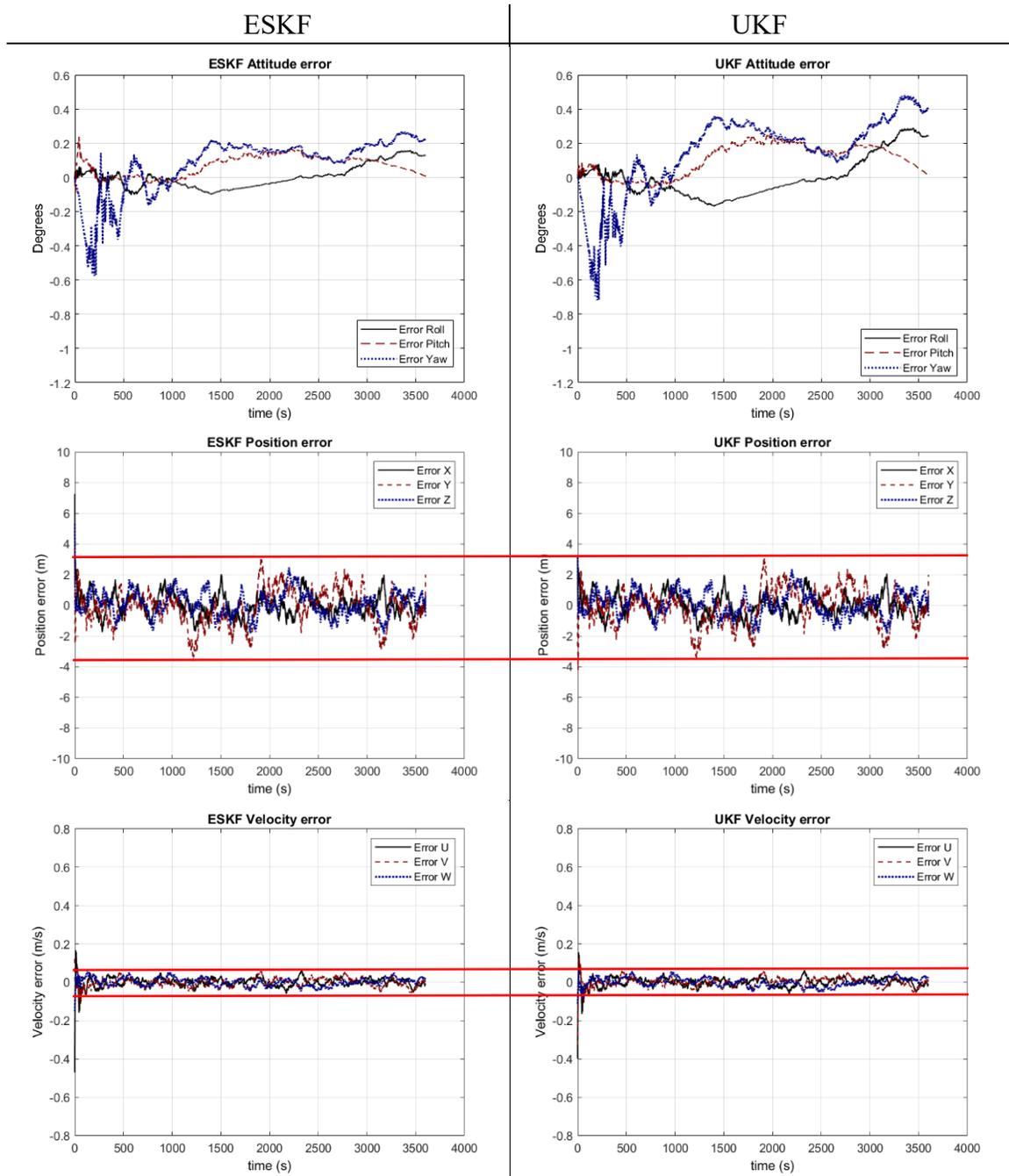


Figure D 1 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)

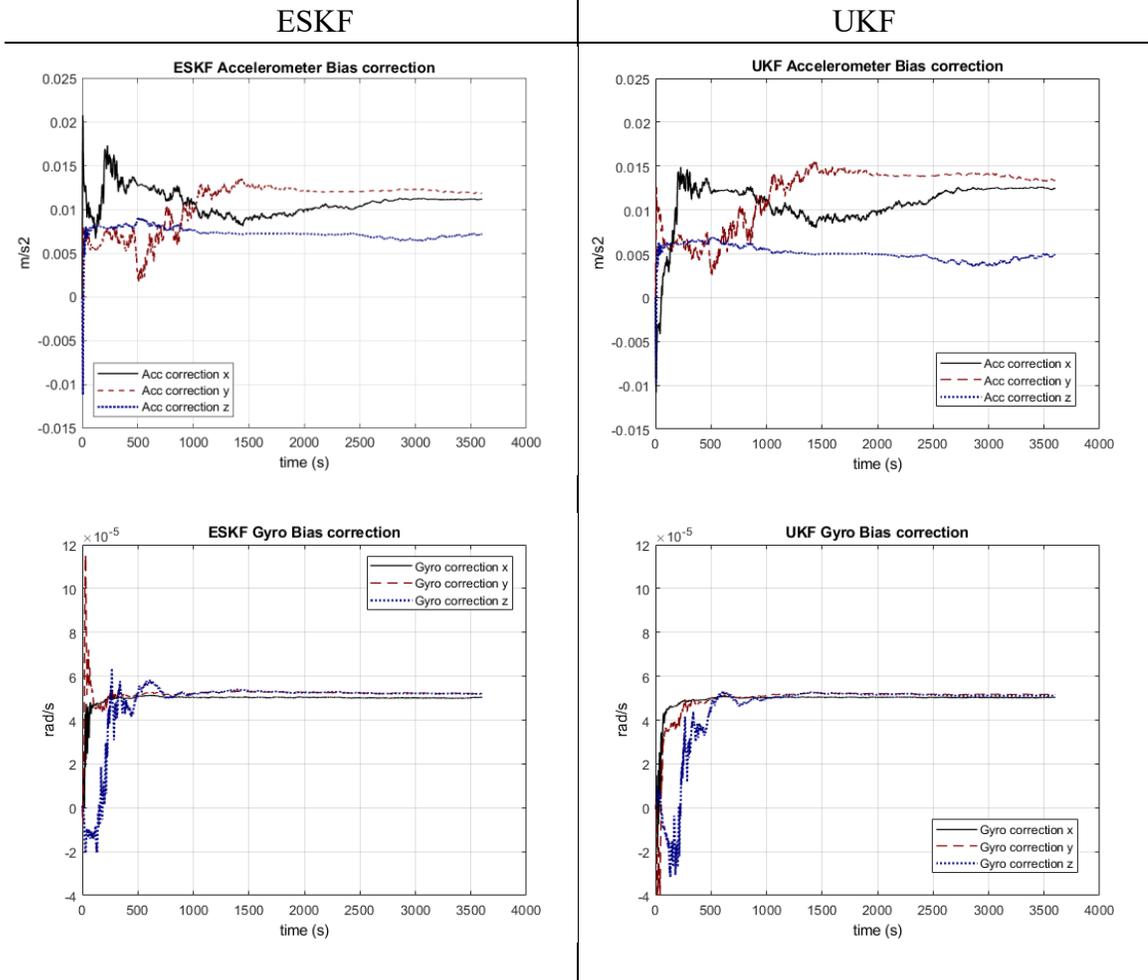


Figure D 2 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)

Appendix D.1.2 Error Covariance

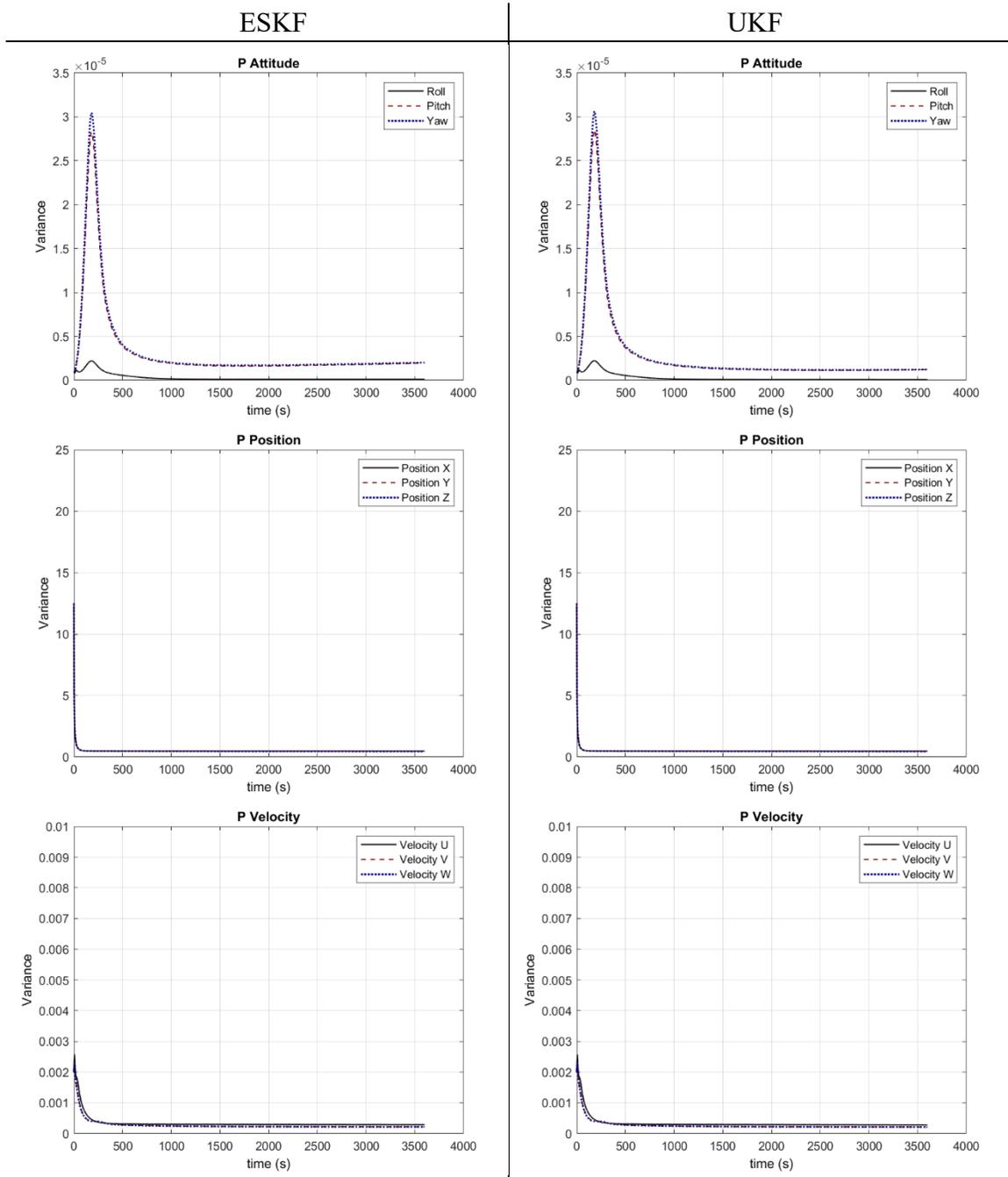


Figure D 3 - Attitude, position, and velocity states variance using ESKF (left) and UKF (right)

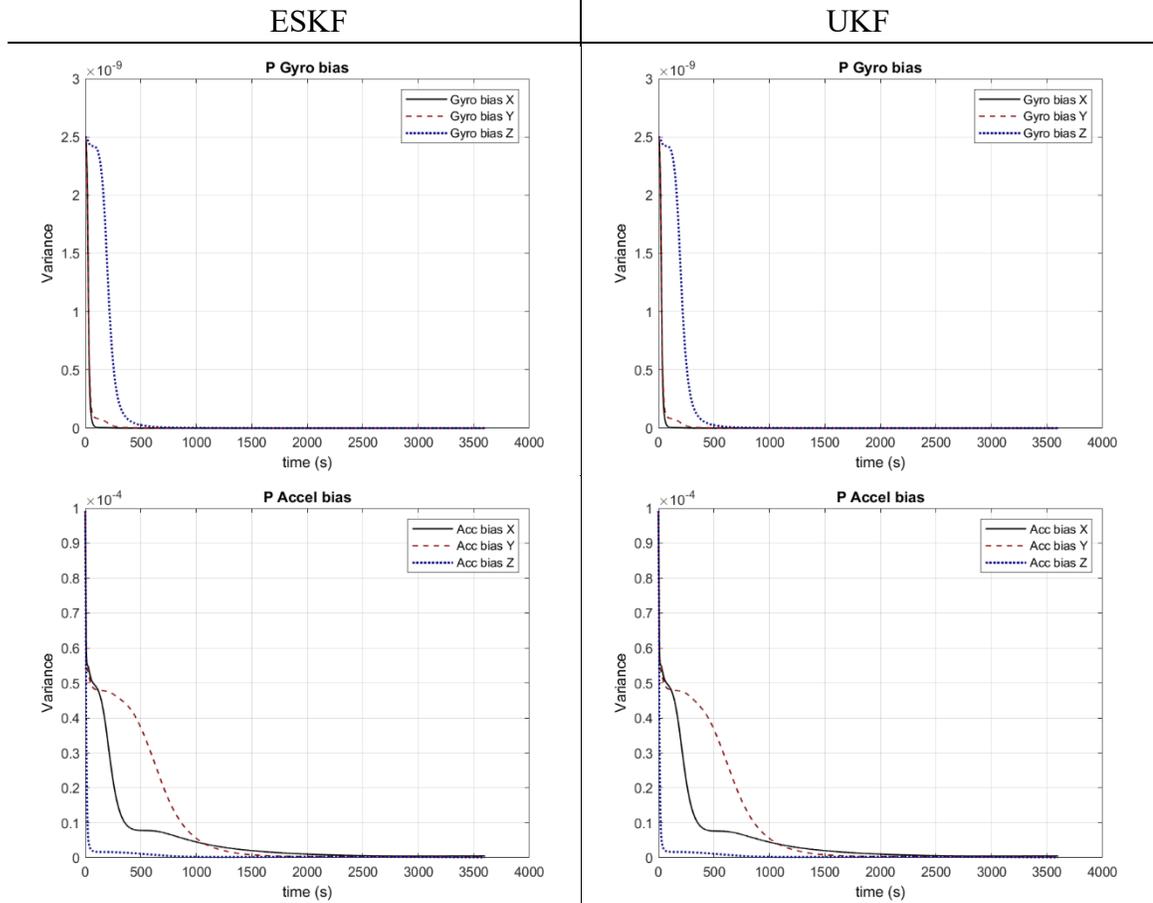


Figure D 4 – Gyroscope and accelerometer bias states variance using ESKF (left) and UKF (right)

Appendix D.1.3 Residuals

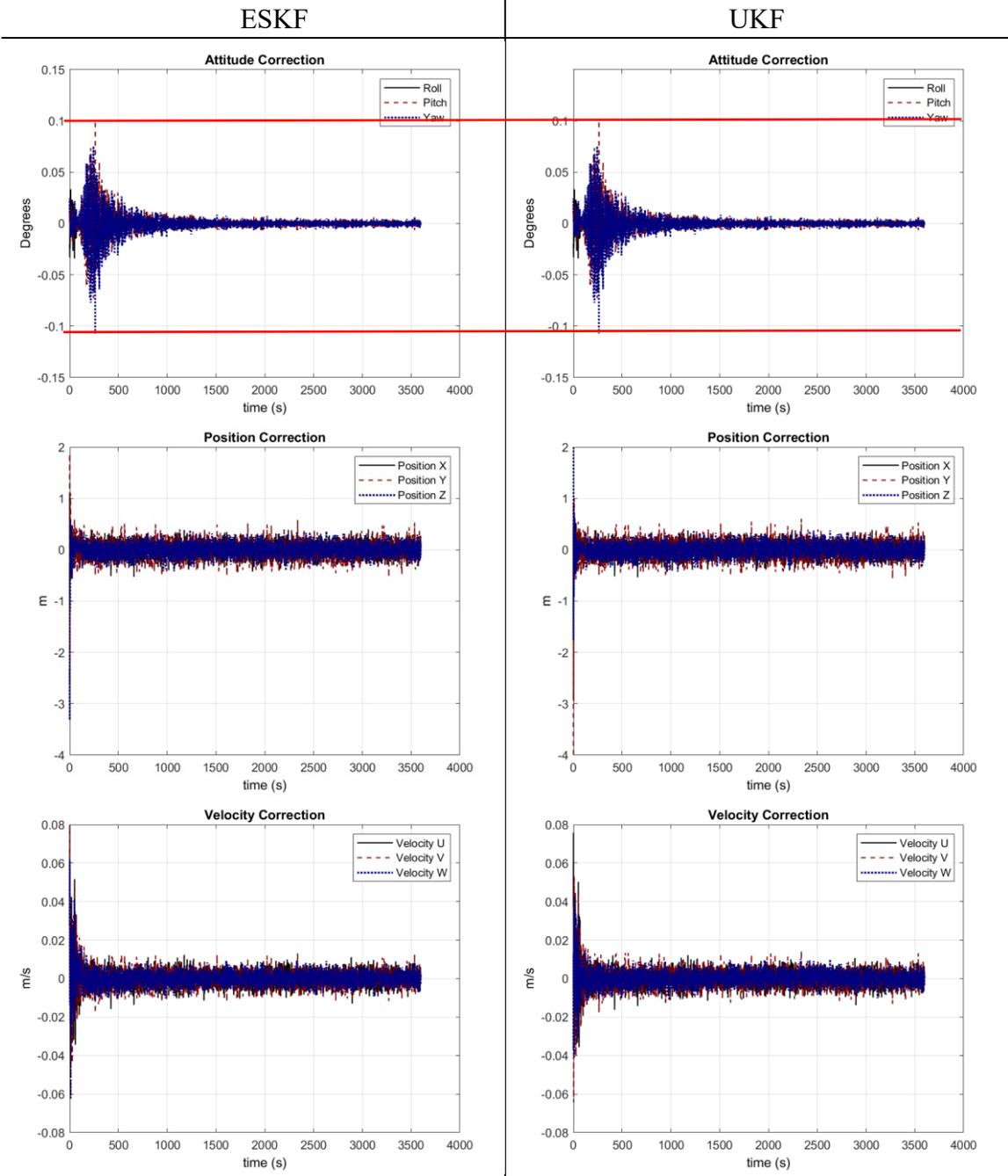


Figure D 5 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)

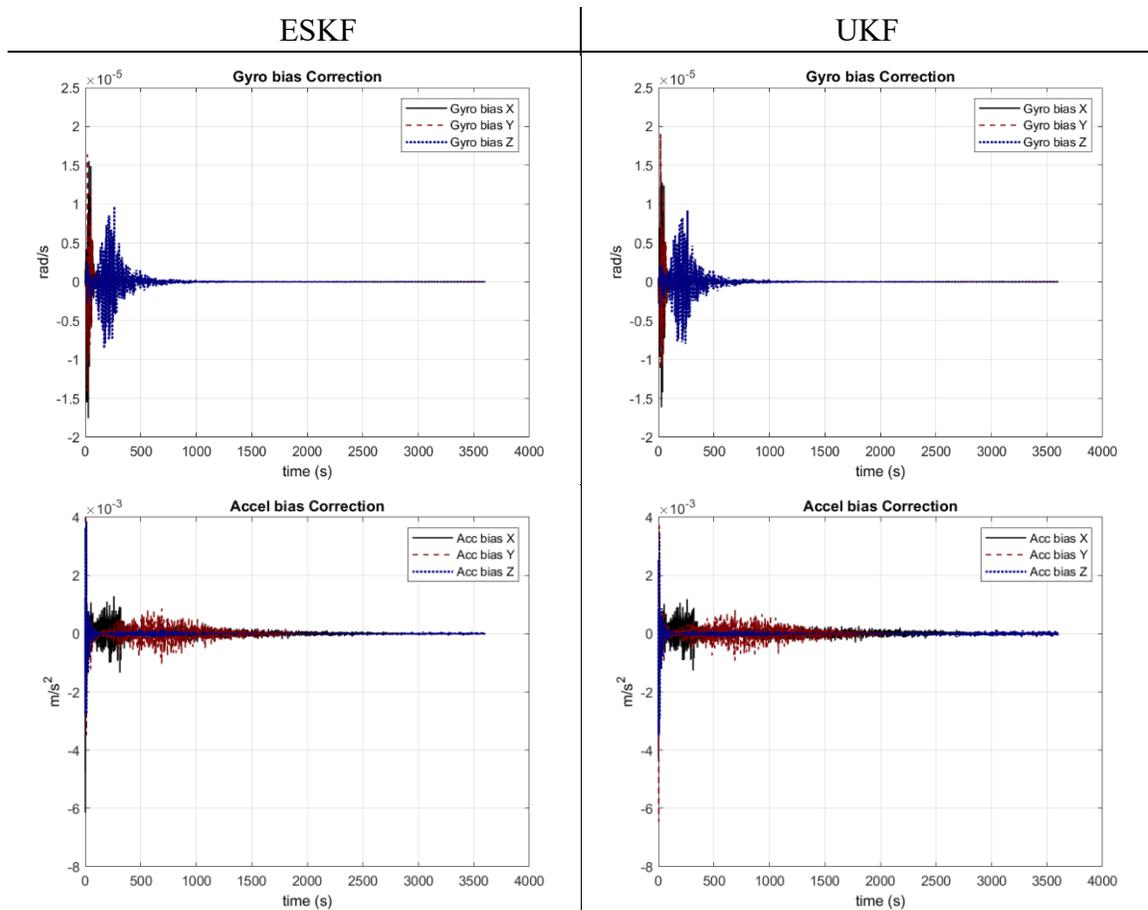


Figure D 6 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)

Appendix D.2 Coloured Noise

Appendix D.2.1 True ground error

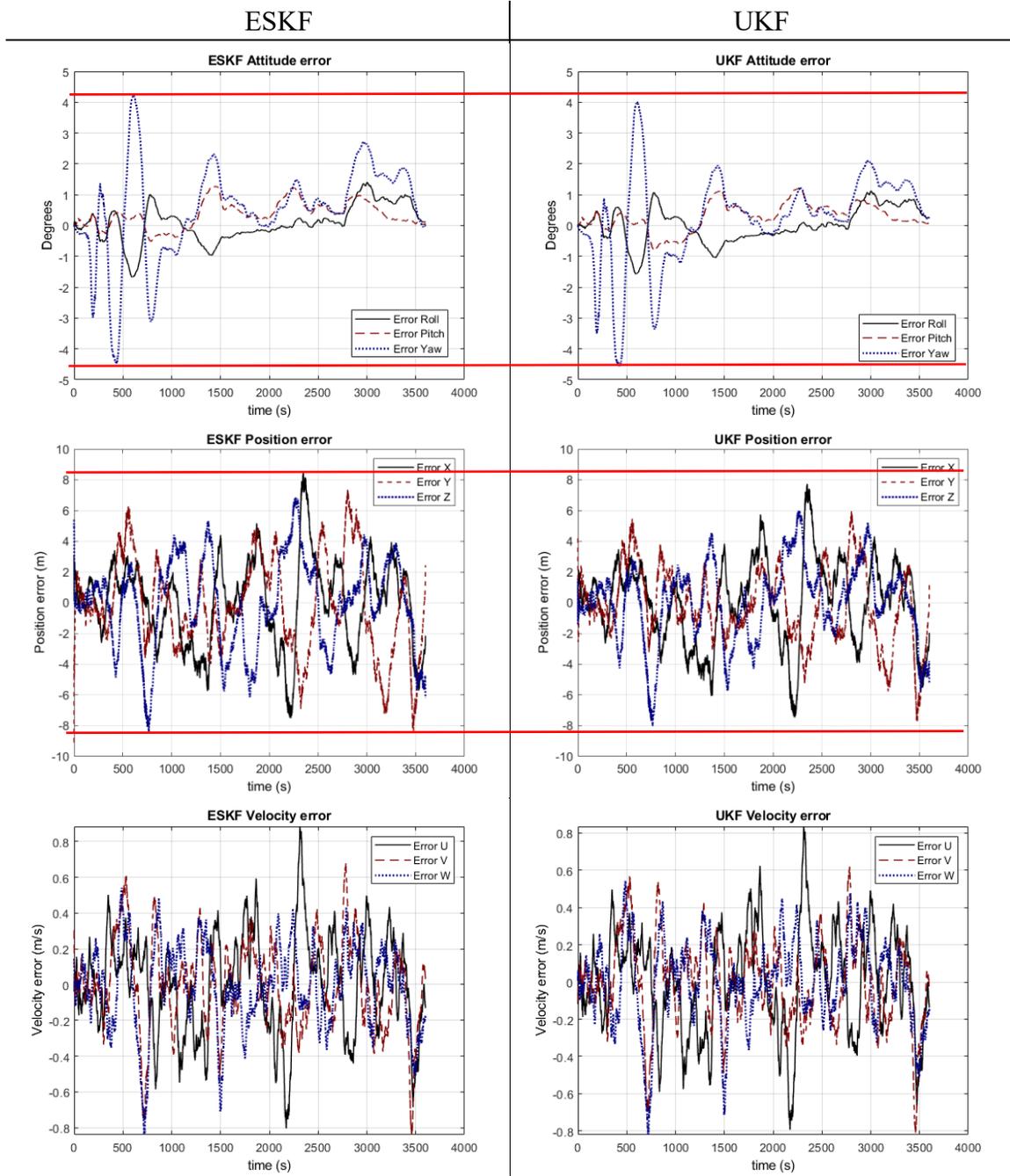


Figure D 7 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)

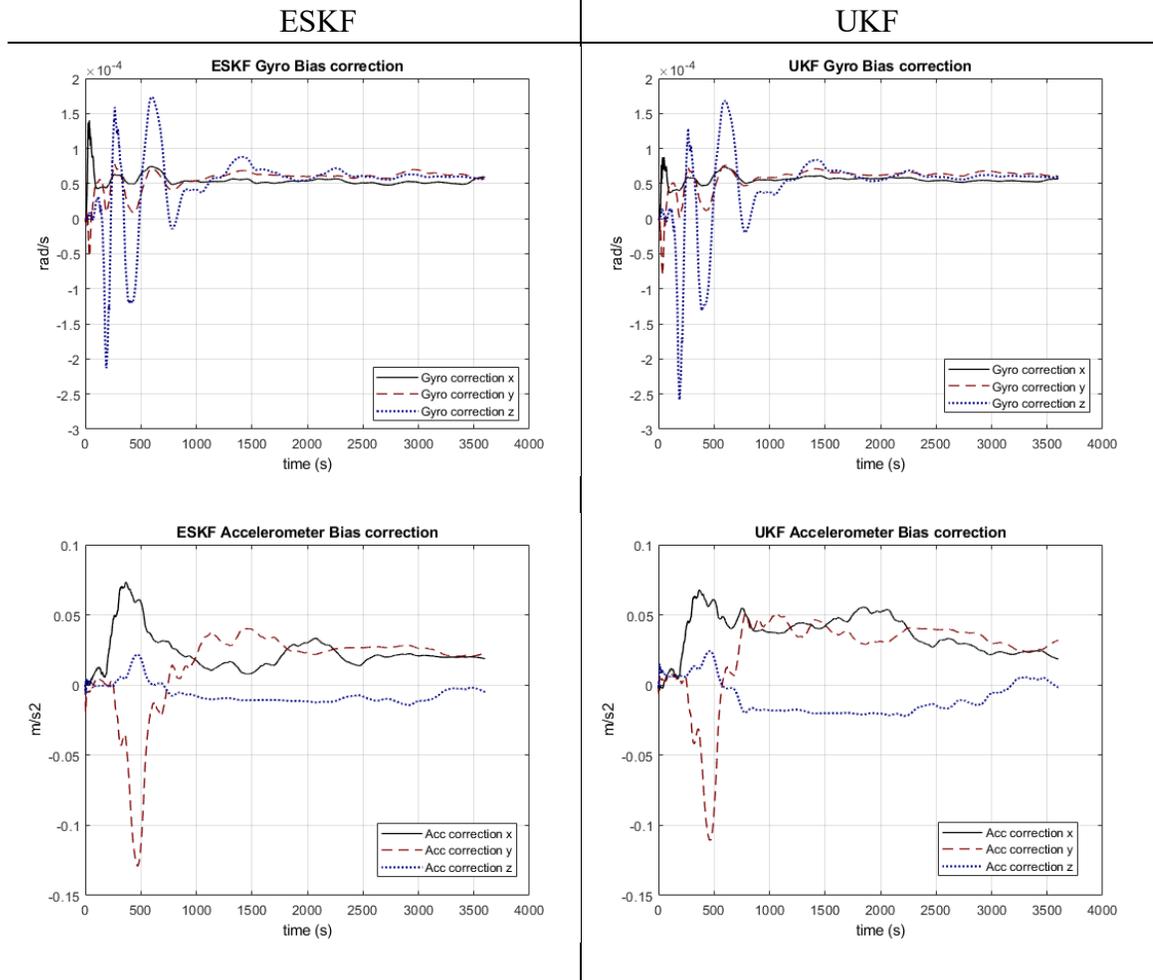


Figure D 8 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)

Appendix D.2.2 Error Covariance

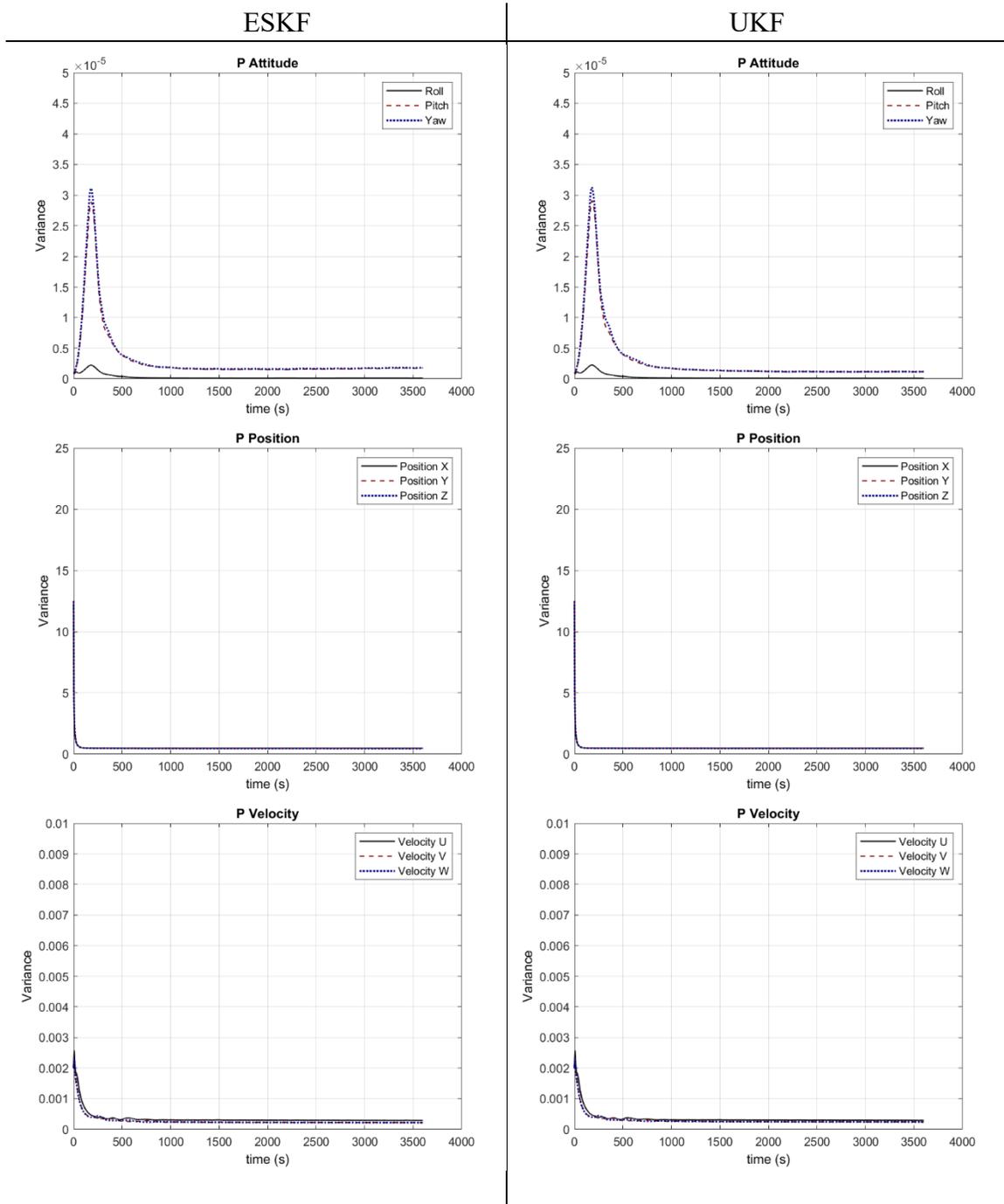


Figure D 9 - Attitude, position, and velocity states variance using ESKF (left) and UKF (right)

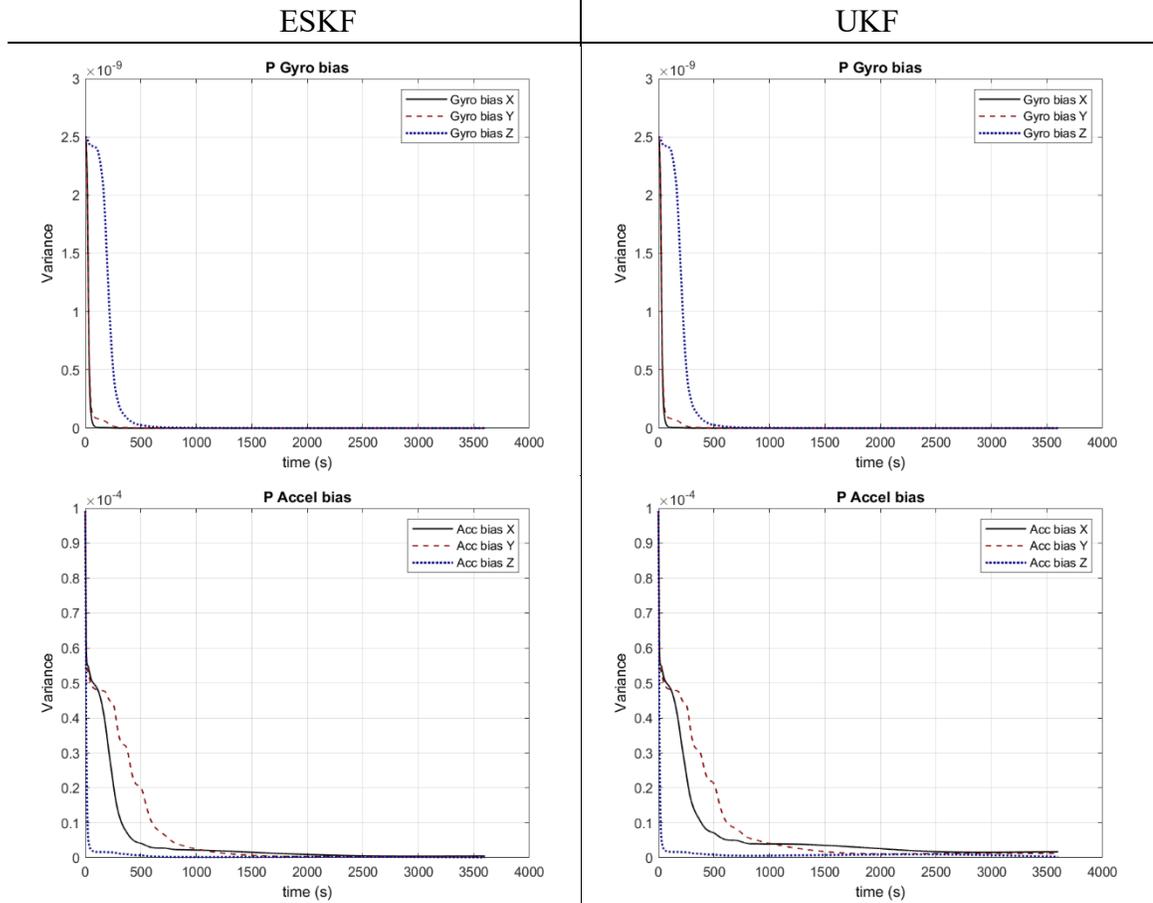


Figure D 10 – Gyroscope and accelerometer bias states variance using ESKF (left) and UKF (right)

Appendix D.2.3 Residuals

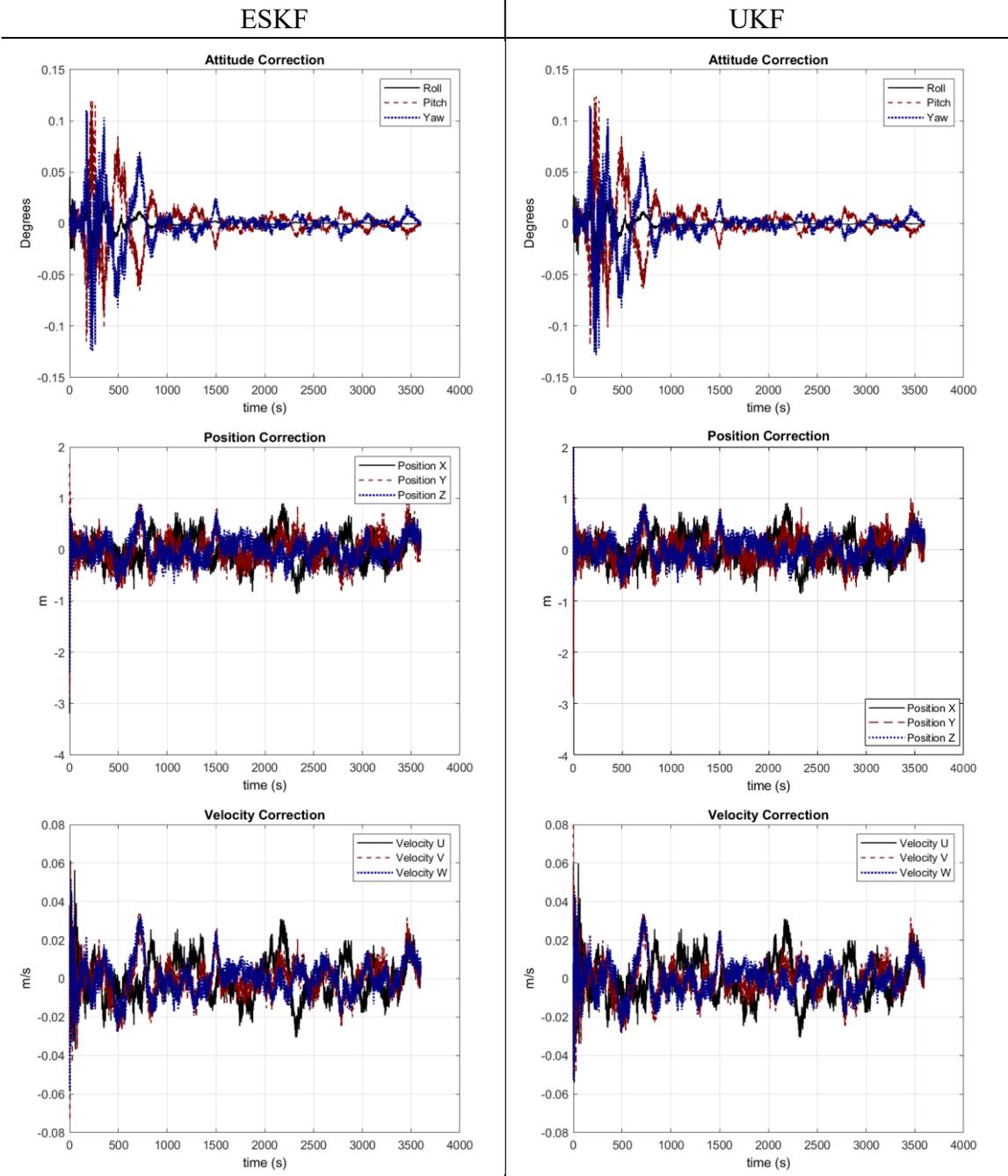


Figure D 11 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)

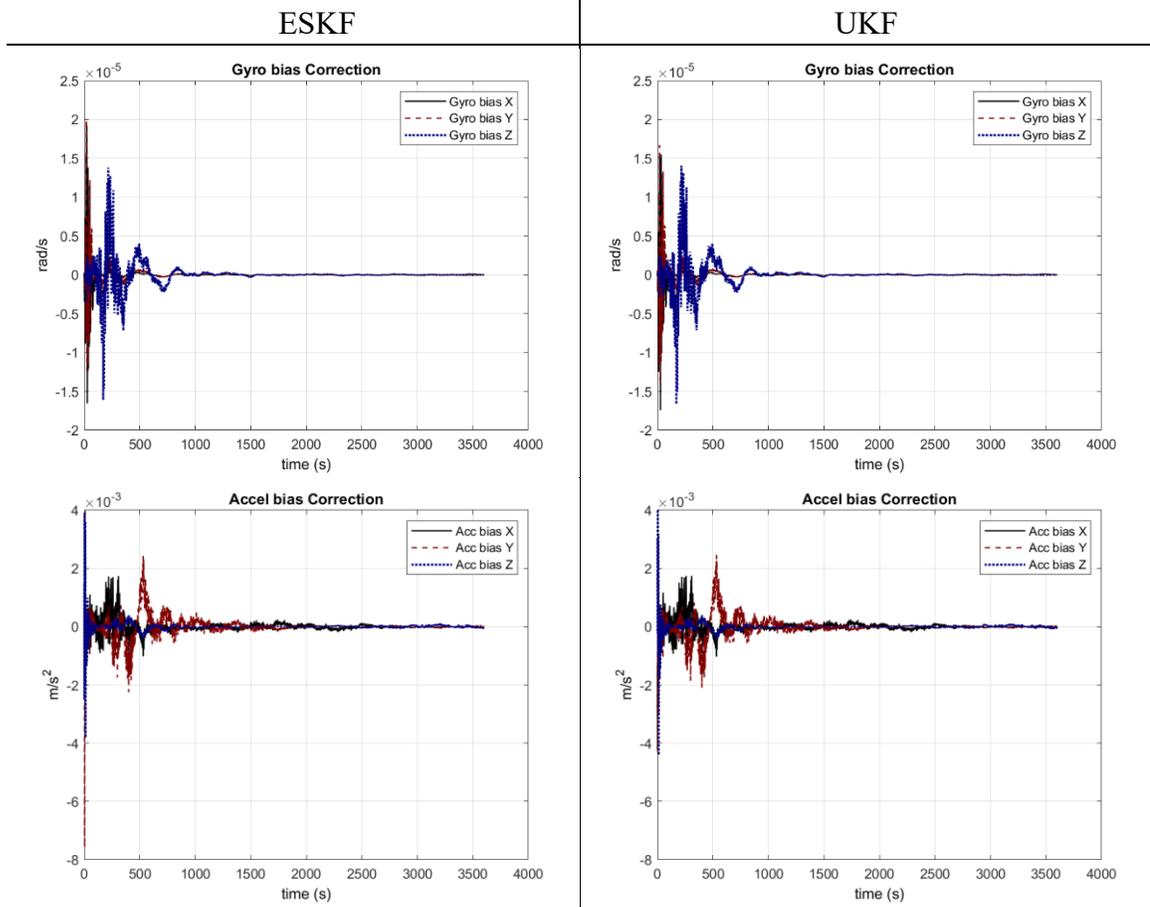


Figure D 12 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)

Appendix D.3 Fuzzy Logic

Appendix D.3.1 True grund error:

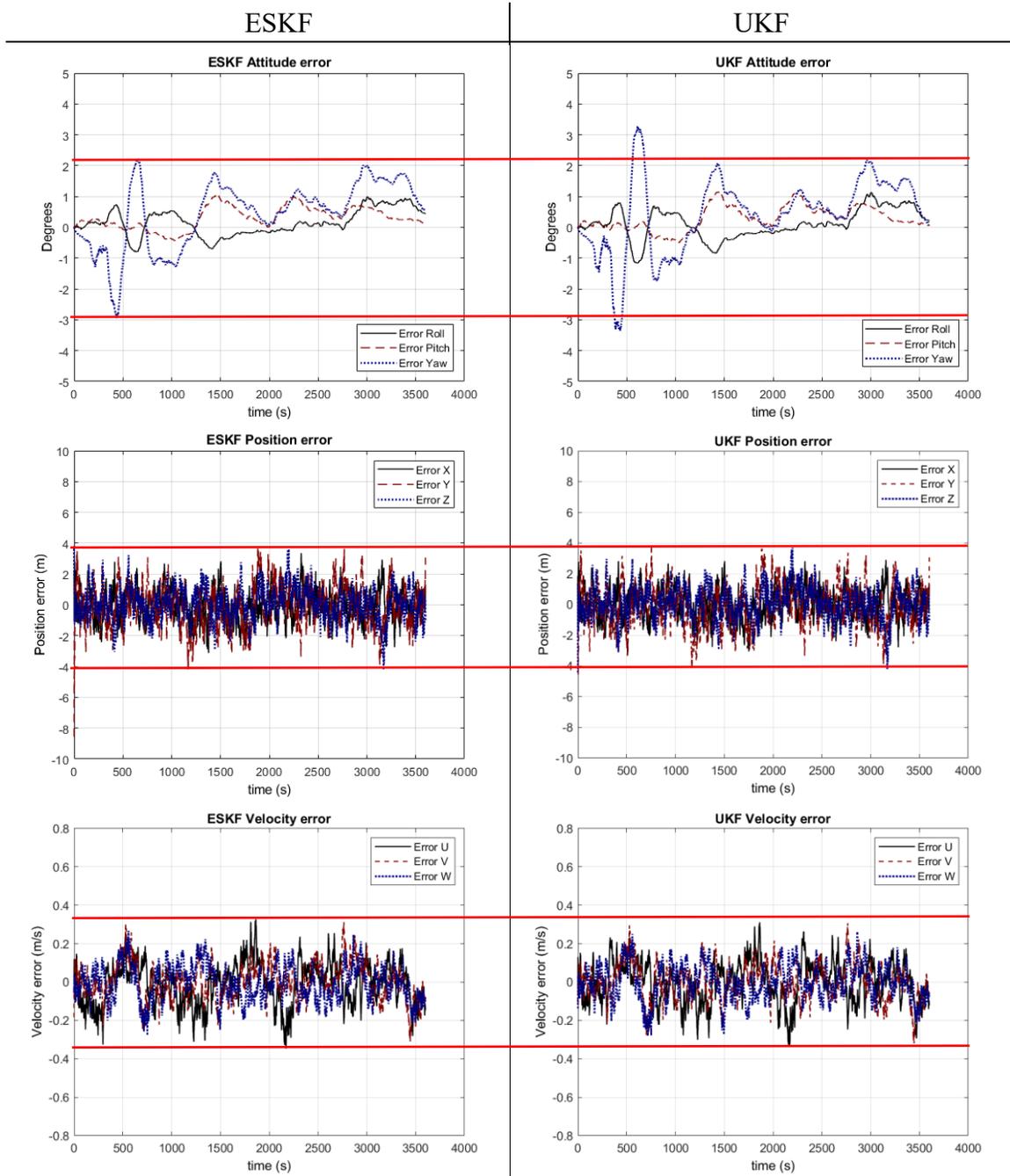


Figure D 13 – Attitude, position, and velocity error between the true-ground and the INS/GNSS integration using ESKF (left) and UKF (right)

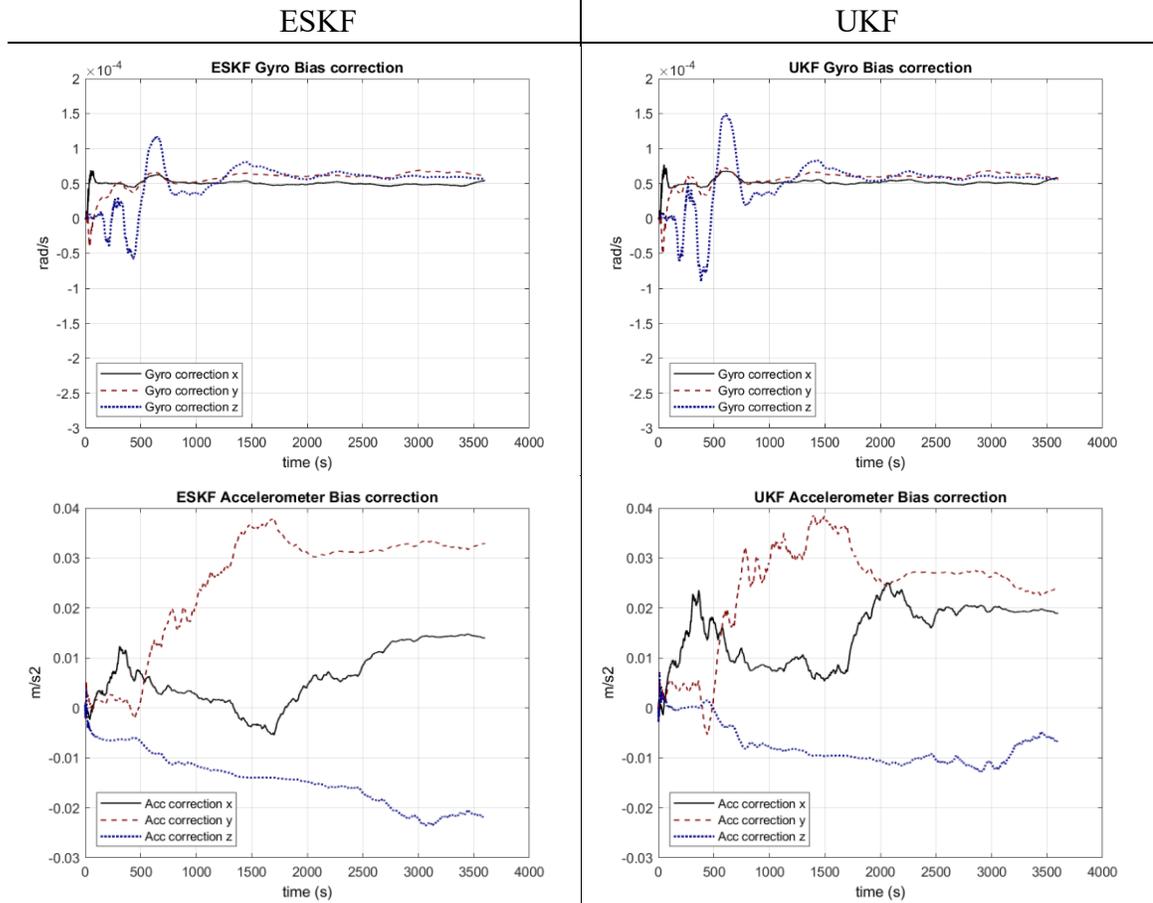


Figure D 14 - Gyroscope and accelerometer bias correction using ESKF (left) and UKF (right)

Appendix D.3.2 Error Covariance

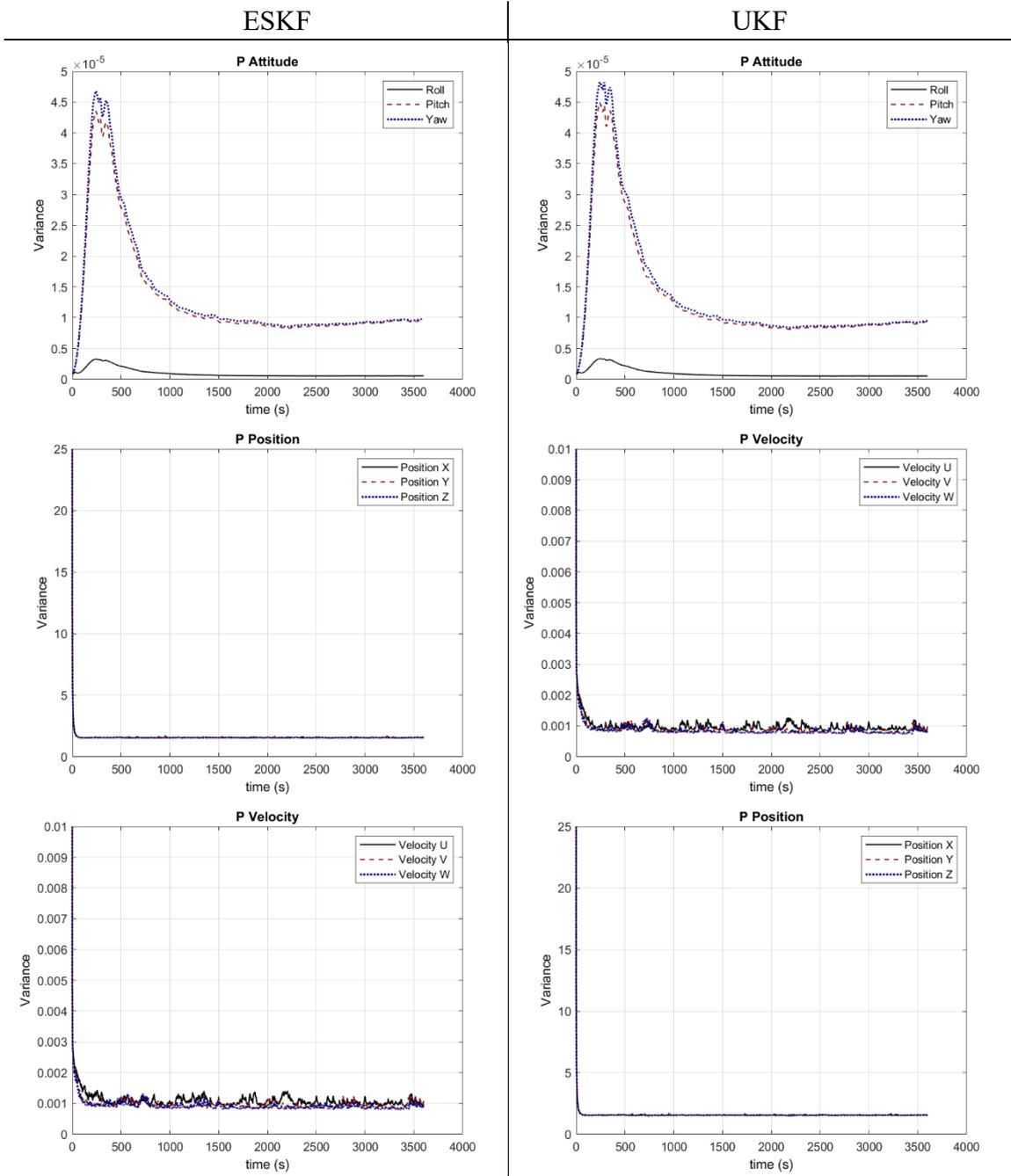


Figure D 15 - Attitude, position, and velocity states variance using ESKF (left) and UKF (right)

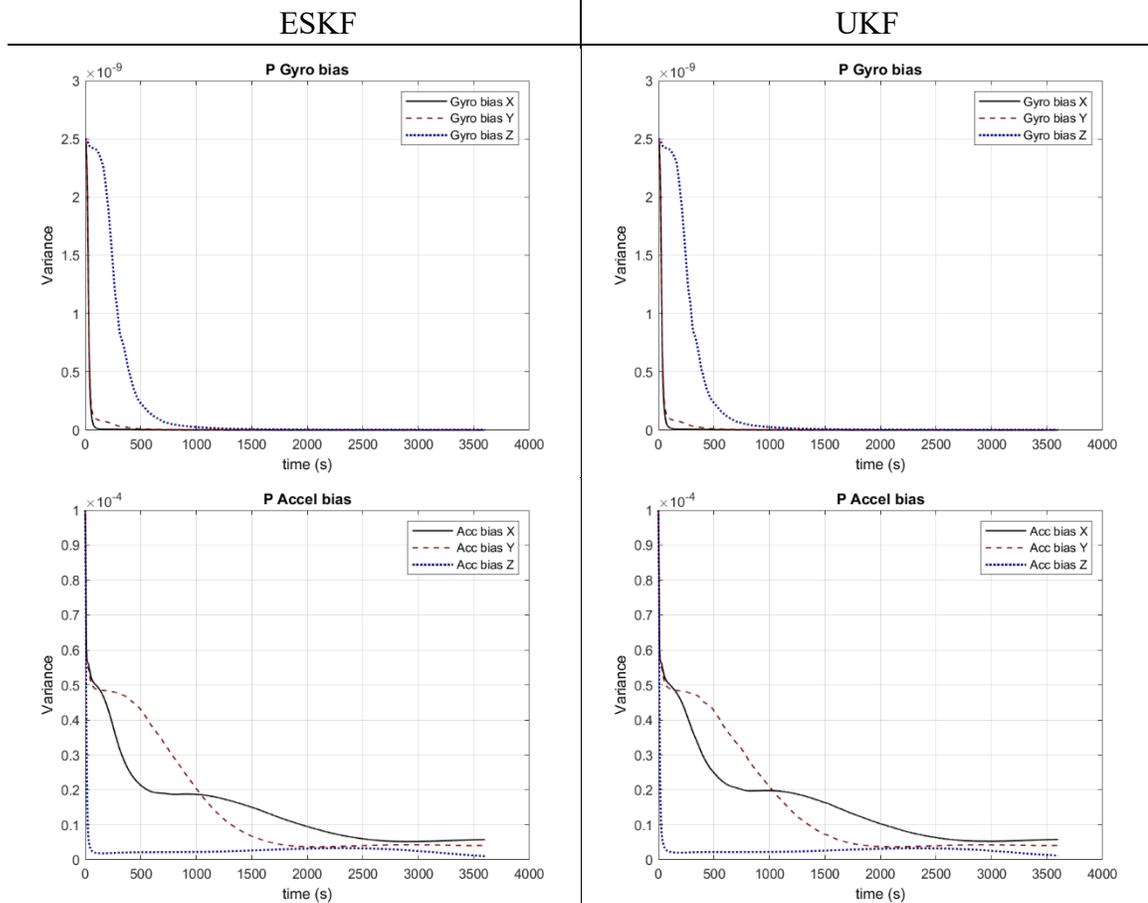


Figure D 16 – Gyroscope and accelerometer bias states variance using ESKF (left) and UKF (right)

Appendix D.3.3 Residuals

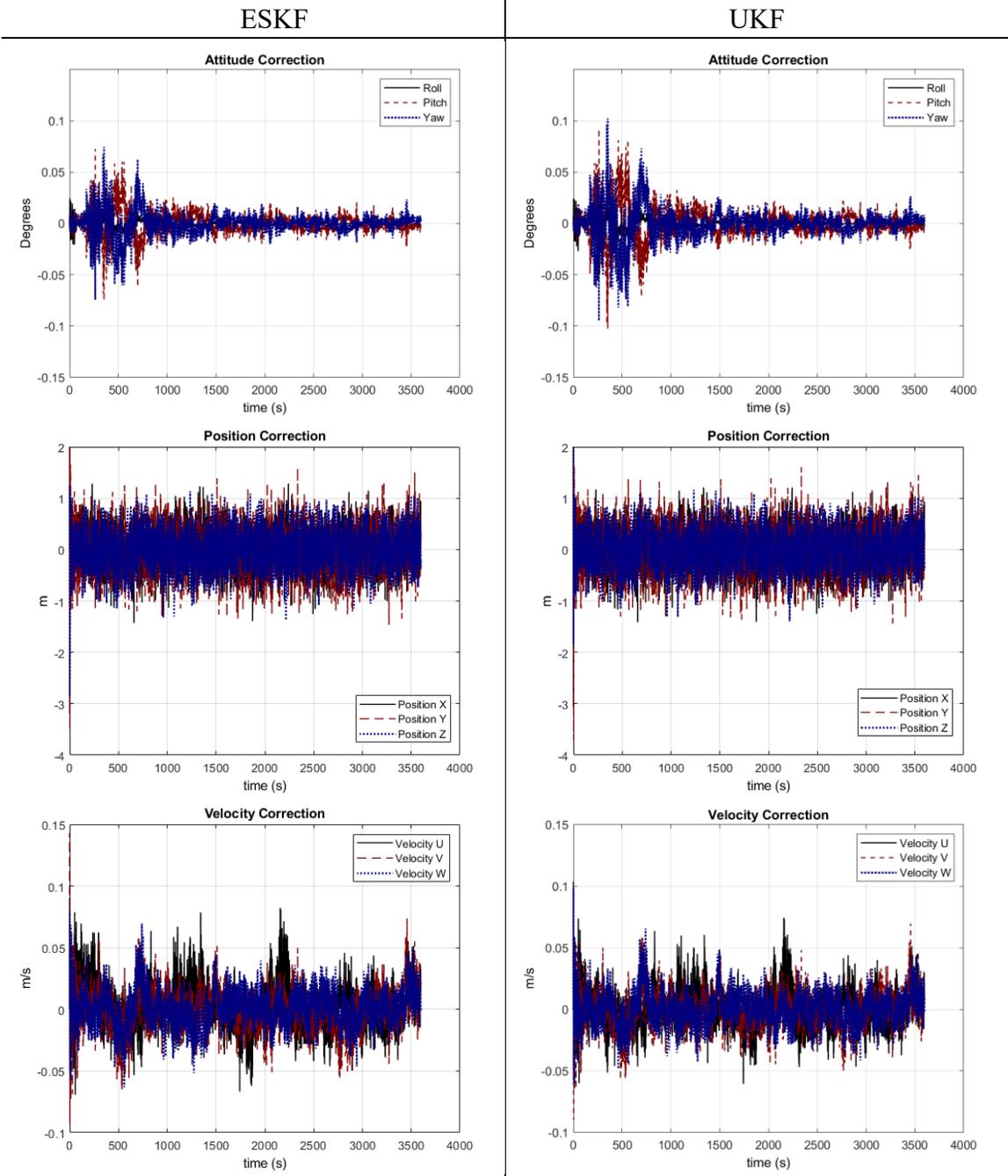


Figure D 17 - KF residuals for Attitude, Position, and Velocity considering the ESKF (left) and UKF (right)

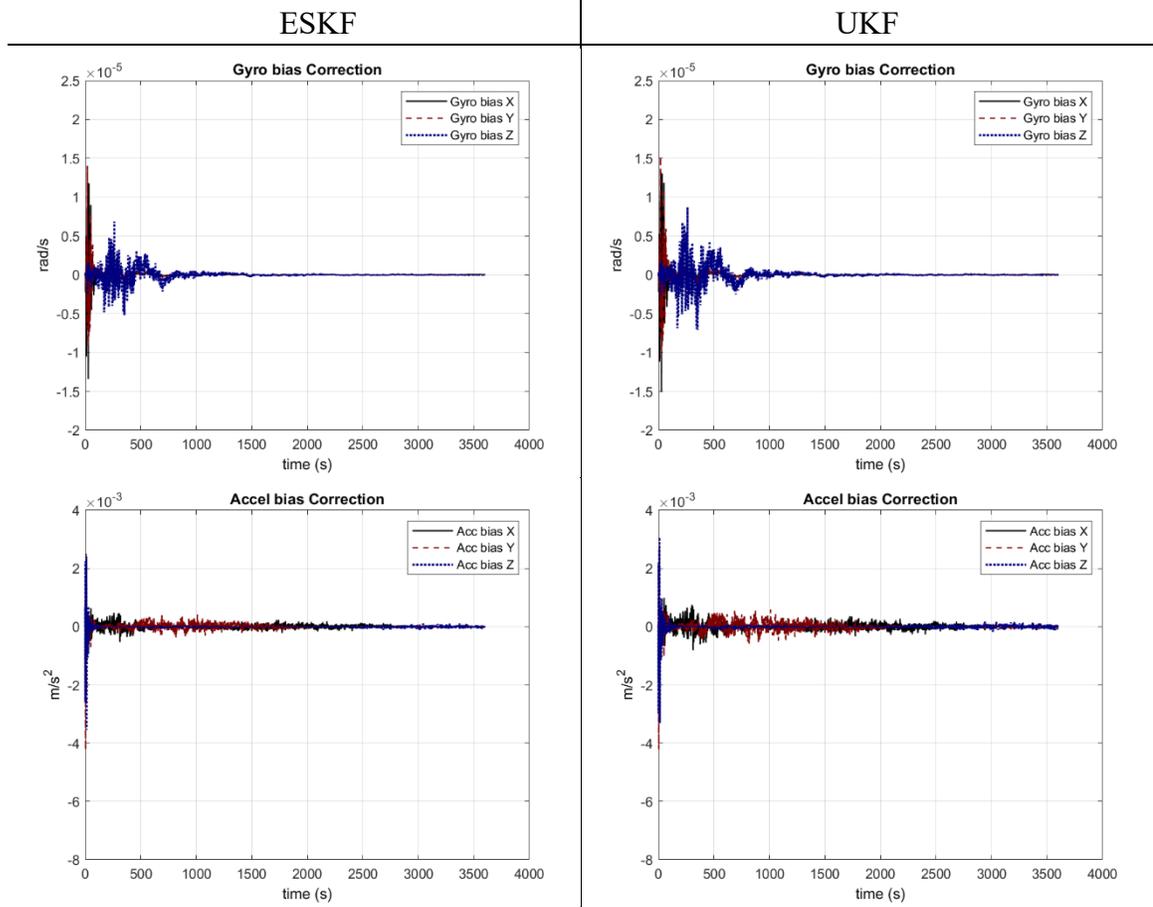


Figure D 18 - KF residuals for gyroscope and accelerometer bias considering the the ESKF (left) and UKF (right)

Appendix E - Motion and IMU raw data generation MATLAB Code

This appendix contains samples of the MATLAB code used to create the motion profile and the raw gyroscope and accelerometer data.

Appendix E.1 Navigation profile generation

```
%%The purpose of this code is to generate a navigation profile
%Gabriel Giannini de Cunto - Rev 04 - 11/02/2020

clear all
%
%-----
%1 Part - Creat Navigation data in NED Frame
%Based on the Matlab documentation available in:
%_https://www.mathworks.com/help/nav/ref/kinematictrajectory-system-
object.html
%-----

duration = 3600; % seconds
fs = 100; % Hz
N = duration*fs; % number of samples
t = ((0:(N-1))/fs)'; %sampling time

radius = 15000; % meters
speed = 20; % meters per second
climbRate = 0.5; % meters per second
initialYaw = 90; % degrees
Pitch = 10; % degrees

initPos = [0, 0, -66];%Initial Position
initVel = [0, speed, climbRate];%Initial Velocity
initOrientation =
quaternion([initialYaw,Pitch,0], 'eulerd', 'zyx', 'frame');%Initial
Orientation

%Generate the trajectory data
trajectory = kinematicTrajectory('SampleRate',fs, ...
    'Velocity',initVel, ...
    'Position',initPos, ...
    'Orientation',initOrientation);

%Specify a constant acceleration and angular velocity in the body
coordinate system. Rotate the body frame to account for the pitch.
accBody = zeros(N,3);
accBody(:,2) = speed^2/radius;
accBody(:,3) = -0.06;

angVelBody = zeros(N,3);
angVelBody(:,1) = (speed/radius)/2;
angVelBody(:,3) = speed/radius;
```

```

pitchRotation = quaternion([0,Pitch,0], 'eulerd', 'zyx', 'frame');
angVelBody = rotateframe(pitchRotation,angVelBody);
accBody = rotateframe(pitchRotation,accBody);

%Call trajectory with the specified acceleration and angular velocity
in the body coordinate system. Plot the position, orientation, and
speed over time.
[position, orientation, velocity] = trajectory(accBody,angVelBody);

%Generate the DCM orientation
eulerAnglesRandians = euler(orientation, 'zyx', 'frame');
dcm = angle2dcm(eulerAnglesRandians(:,1), eulerAnglesRandians(:,2),
eulerAnglesRandians(:,3), 'zyx');

figure;
plot(t,velocity(:,1), '-', 'LineWidth',1, 'Color',[0 0 0])
hold on
plot(t,velocity(:,2), '--', 'LineWidth',1, 'Color',[0 0 0])
hold on
plot(t,velocity(:,3), ':', 'LineWidth',1.5, 'Color',[0 0 0])
hold off
grid on;title('Velocity', 'Color',[0 0 0]);xlabel('time (s)', 'Color',[0
0 0]);ylabel('Velocity (m/s)', 'Color',[0 0 0]);
legend('Vel North', 'Vel East', 'Vel Down');

figure;
plot(t,position(:,1), '-', 'LineWidth',1, 'Color',[0 0 0])
hold on
plot(t,position(:,2), '--', 'LineWidth',1, 'Color',[0 0 0])
hold on
plot(t,position(:,3), ':', 'LineWidth',1.5, 'Color',[0 0 0])
hold off
grid on;title('Position', 'Color',[0 0 0]);xlabel('time (s)', 'Color',[0
0 0]);ylabel('Position (m)', 'Color',[0 0 0]);
legend('Pos North', 'Pos East', 'Pos Down');

figure;
plot(t,rad2deg(eulerAnglesRandians(:,1)), '-', 'LineWidth',1, 'Color',[0 0
0])
hold on
plot(t,rad2deg(eulerAnglesRandians(:,2)), '--', 'LineWidth',1, 'Color',[0
0 0])
hold on
plot(t,rad2deg(eulerAnglesRandians(:,3)), ':', 'LineWidth',1.5, 'Color',[0
0 0])
hold off
grid on;title('Attitude', 'Color',[0 0 0]);xlabel('time (s)', 'Color',[0
0 0]);ylabel('Orientation (Deg)', 'Color',[0 0 0]);
legend('Yaw', 'Pitch', 'Roll');

```



```

        -cos(lat(i,1)) * cos(lon(i,1)), -cos(lat(i,1)) *
sin(lon(i,1)), -sin(lat(i,1))];
    ECEF_DCM(:, :, i) = C_e_n' * dcm(:, :, i);

%Calculate Pos and Vel in ECEF frame.
    [ECEF_Vel(i,1), ECEF_Vel(i,2), ECEF_Vel(i,3)] =
ned2ecefv(velocity(i,1), velocity(i,2), velocity(i,3), Initial_lat, Initial
_lon, 'radians'); %Velocity in ECEF
    [ECEF_Pos(i,1), ECEF_Pos(i,2), ECEF_Pos(i,3)] =
ned2ecef(position(i,1), position(i,2), position(i,3), Initial_lat, Initial
lon, Initial_h, wgs84, 'radians'); %Position in ECEF

end

%Calculate Gravity in ECEF for the navigation profile
for i=1:length(t)
    % Calculate distance from center of the Earth
    mag_r = sqrt(ECEF_Pos(i, :)'' * ECEF_Pos(i, :)');

    % Calculate gravitational acceleration
    z_scale = 5 * (ECEF_Pos(i,3) / mag_r)^2;
    gamma = -mu / mag_r^3 * (ECEF_Pos(i, :)' + 1.5 * J_2 * (a / mag_r)^2
*...
        [(1 - z_scale) * ECEF_Pos(i,1)']; (1 - z_scale) *
ECEF_Pos(i,2)';...
        (3 - z_scale) * ECEF_Pos(i,3)']]);

    % Add centripetal acceleration
    g(i,1:2) = (gamma(1:2) + Earth_rot^2 * ECEF_Pos(i,1:2)')';
    g(i,3) = gamma(3);

end

```

Appendix E.3 Accelerometer and Gyroscope Raw data generation

```

%
%-----
%Inverse kinematic model for calculate the raw gyroscope and
accelerometer
% Gabriel Giannini de Cunto - Rev 04 - 09/02/2020

% Inputs:
% dt          time interval between epochs (s)
% ECEF_DCM    body-to-ECEF-frame coordinate transformation matrix
% =[X, Y, Z];
% ECEF_Vel=   velocity of body frame w.r.t. ECEF frame, resolved
along
%           ECEF-frame axes (m/s)
% ECEF_Pos    Cartesian position of body frame w.r.t. ECEF frame,
resolved
%           along ECEF-frame axes (m)
% Outputs:
% Calc_acc    specific force of body frame w.r.t. ECEF frame,
resolved

```

```

%           along body-frame axes, averaged over time interval
(m/s^2)
%   Calc_Gyro   angular rate of body frame w.r.t. ECEF frame, resolved
%           about body-frame axes, averaged over time interval
(rad/s)

for i=2:length(t)

    %Calculate the acceleration al
    al(1,1) = (ECEF_Vel(i,1)-ECEF_Vel(i-1,1))/dt;
    al(1,2) = (ECEF_Vel(i,2)-ECEF_Vel(i-1,2))/dt;
    al(1,3) = (ECEF_Vel(i,3)-ECEF_Vel(i-1,3))/dt;

    %calculate the accelerometer
    Calc_accel(i-1,:)=ECEF_DCM(:, :, i-1)'*(al(1, :)-g(i-
1, :))+cross(Total_comp_l_g,ECEF_Vel(i-1, :))';

    %Calculate the angular velocity
    Sg=((ECEF_DCM(:, :, i-1)'*ECEF_DCM(:, :, i))-eye(3))/dt;
    wlb= [Sg(3,2) Sg(1,3) Sg(2,1)];

    %Determine the Gyroscope vector compensated by Earth rotation rate
    Calc_gyro(i, :)=wlb+(ECEF_DCM(:, :, i)'*Earth_rot_ECEF)';
end

%Smoothing function
Calc_gyro(1, :) = Calc_gyro(2, :);
Calc_accel(length(t), :) = Calc_accel(length(t)-1, :);
Calc_accel=sgolayfilt(Calc_accel,10,45);
Calc_gyro=sgolayfilt(Calc_gyro,10,45);

%Create the vector for the SIMULINK
[ECEF_Yaw, ECEF_Pitch, ECEF_Roll] = dcm2angle(ECEF_DCM, 'zyx');
ECEF_Yaw=rad2deg(ECEF_Yaw);
ECEF_Pitch=rad2deg(ECEF_Pitch);
ECEF_Roll=rad2deg(ECEF_Roll);
ECEF_Att=[ECEF_Roll ECEF_Pitch ECEF_Yaw];

%
% Save the data
%
SaveData=[t Calc_gyro Calc_accel ECEF_Pos ECEF_Vel ECEF_Att]';

save('Raw_IMU_GNSS.mat', 'SaveData', '-v7.3');

```

Appendix F - INS, ESKF, and UKF MATLAB Code

This appendix contains the MATLAB functions used for the INS model, ESKF model and UKF model.

Appendix F.1 INS model

```
%INS model
%Gabriel Giannini de Cunto - Rev. 11 - 22/12/2019

function [INS_Px_out, INS_Py_out, INS_Pz_out,INS_Vu_out, INS_Vv_out,
INS_Vw_out, INS_DCM_out] = INS_ECEF(INS_Px_in,INS_Py_in,INS_Pz_in,...
    INS_Vu_in, INS_Vv_in, INS_Vw_in,INS_DCM_in,Accel_x,
Accel_y,Accel_z,Gyro_x,Gyro_y,Gyro_z)

% Inputs:
%   INS_P_in   previous position in ECEF (m)
%   INS_V_in   previous velocity in ECEF (m/2)
%   INS_DCM_in previous body-to-ECEF DCM
%   Gyro       angular rate of body frame w.r.t. ECEF (rad/s)
%   Accel      specific force of body frame w.r.t. ECEF (m/s^2)

% Outputs:
%   INS_P_out   New position of body in ECEF (m)
%   INS_V_out   New velocity of body in ECEF (m/s)
%   INS_DCM_out New body-to-ECEF-frame DCM

% parameters
dt=0.0100; %Sampling time - must be the same of Ts
Earth_rot = 7.292115E-5; % Earth rotation rate (rad/s)

% Earth rotation and Coriolis in ECEF.
Earth_rot_ECEF = [0 0 Earth_rot];
Total_comp_l_g= 2*Earth_rot_ECEF;

% Initialize the scalar
INS_Px_out=0;
INS_Py_out=0;
INS_Pz_out=0;
INS_Vu_out=0;
INS_Vv_out=0;
INS_Vw_out=0;
INS_DCM_out=zeros(3,3);
g=zeros(1,3);

%Create the vectors
INS_P_in=[INS_Px_in, INS_Py_in, INS_Pz_in];
INS_V_in=[INS_Vu_in, INS_Vv_in, INS_Vw_in];
Accel=[Accel_x, Accel_y, Accel_z];
Gyro=[Gyro_x, Gyro_y, Gyro_z];
```

```

% Calculate distance from center of the Earth
mag_r = sqrt(INS_P_in' * INS_P_in);

% Calculate gravitational acceleration
a = 6378137; % WGS84 Equatorial radius in meters.
mu = 3.986004418E14; %WGS84 Earth gravitational constant (m^3 s^-2)
J_2 = 1.082627E-3; %WGS84 Earth's second gravitational constant
z_scale = 5 * (INS_P_in(1,3) / mag_r)^2;
gamma = -mu / mag_r^3 * (INS_P_in(1,:)') + 1.5 * J_2 * (a / mag_r)^2 * ...
    [(1 - z_scale) * INS_P_in(1,1)']; (1 - z_scale) * INS_P_in(1,2)'; ...
    (3 - z_scale) * INS_P_in(1,3)');
% Add centripetal acceleration
g(1,1:2) = (gamma(1:2) + Earth_rot^2 * INS_P_in(1,1:2)')';
g(1,3) = gamma(3);

% Begin the INS Dynamics
% Get quaternion vector from DCM
% [q] = convert_dcm_to_quat(INS_DCM_in);
a = 0;
b = 0;
c = 0;
d = 0;
Tr = trace(INS_DCM_in);
Pa = 1 + Tr;
Pb = 1 + 2*INS_DCM_in(1,1)-Tr;
Pc = 1 + 2*INS_DCM_in(2,2)-Tr;
Pd = 1 + 2*INS_DCM_in(3,3)-Tr;
[m,i] = max([Pa Pb Pc Pd]);
switch(i)
    case 1
        a = 0.5*sqrt(Pa);
        b = (INS_DCM_in(3,2)-INS_DCM_in(2,3))/(4*a);
        c = (INS_DCM_in(1,3) - INS_DCM_in(3,1))/(4*a);
        d = (INS_DCM_in(2,1) - INS_DCM_in(1,2))/(4*a);
    case 2
        b = 0.5*sqrt(Pb);
        c = (INS_DCM_in(2,1)+INS_DCM_in(1,2))/(4*b);
        d = (INS_DCM_in(1,3) + INS_DCM_in(3,1))/(4*b);
        a = (INS_DCM_in(3,2) - INS_DCM_in(2,3))/(4*b);
    case 3
        c = 0.5*sqrt(Pc);
        d = (INS_DCM_in(3,2)+INS_DCM_in(2,3))/(4*c);
        a = (INS_DCM_in(1,3) - INS_DCM_in(3,1))/(4*c);
        b = (INS_DCM_in(2,1) + INS_DCM_in(1,2))/(4*c);
    case 4
        d = 0.5*sqrt(Pd);
        a = (INS_DCM_in(2,1)-INS_DCM_in(1,2))/(4*d);
        b = (INS_DCM_in(1,3) + INS_DCM_in(3,1))/(4*d);
        c = (INS_DCM_in(3,2) + INS_DCM_in(2,3))/(4*d);
end

if a <= 0
    a = -a;
    b = -b;

```

```

        c = -c;
        d = -d;
    end
    q = [a;b;c;d];

%Get the earth rotation rate in L frame
Total_comp_b=INS_DCM_in'*Earth_rot_ECEF';

%Correcting gyro
Corr_gyro= Gyro- Total_comp_b';

%Calculates the gyro skew-symmetric matrix
w = [ 0, -Corr_gyro(1,1), -Corr_gyro(1,2), -Corr_gyro(1,3);...
      Corr_gyro(1,1), 0, Corr_gyro(1,3), -Corr_gyro(1,2);...
      Corr_gyro(1,2), -Corr_gyro(1,3), 0, Corr_gyro(1,1);...
      Corr_gyro(1,3), Corr_gyro(1,2), -Corr_gyro(1,1), 0];

%Calculate the qdot
qdot=(1/2)*(w*q);

%Calculate the new q
New_q = q + (dt)*qdot;
qa=New_q(1,1);
qb=New_q(2,1);
qc=New_q(3,1);
qd=New_q(4,1);
%--> Normalize the quaternion
normalization_factor = sqrt(qa^2 + qb^2 + qc^2 + qd^2);
qa = qa/normalization_factor;
qb = qb/normalization_factor;
qc = qc/normalization_factor;
qd = qd/normalization_factor;

%Convert the new q into the new DCM
INS_DCM_out = [qa^2+qb^2-qc^2-qd^2, 2*(qb*qc-qa*qd),
2*(qb*qd+qa*qc);...
2*(qb*qc+qa*qd), qa^2-qb^2+qc^2-qd^2, 2*(qc*qd-qa*qb);...
2*(qb*qd-qa*qc), 2*(qc*qd+qa*qb), qa^2-qb^2-qc^2+qd^2];

%calculate the velocity dot
vldot= (INS_DCM_in*Accel')'+ g -cross(Total_comp_l_g,INS_V_in);

%Calculate the new velocity
INS_V = INS_V_in + (dt)*vldot;

INS_Vu_out=INS_V(1);
INS_Vv_out=INS_V(2);
INS_Vw_out=INS_V(3);

% UPDATE CARTESIAN POSITION
INS_P = (INS_P_in + INS_V_in* dt);
INS_Px_out=INS_P(1);
INS_Py_out=INS_P(2);
INS_Pz_out=INS_P(3);

```

Appendix F.2 ESKF model

```
%Perform the ESKF and update the states
% Gabriel Giannini de Cunto - Rev 11 - 13/12/2019

function [DCM_ECEF_new, KF_x, KF_y, KF_z, KF_Vu, KF_Vv, KF_Vw,...
    KF_bg_x,KF_bg_y,KF_bg_z,KF_ba_x,KF_ba_y,KF_ba_z,P_matrix_new,
    delta_z_new, K_matrix] = INS_GNSS_KF(Clock, GNSS_X, GNSS_Y,
    GNSS_Z,GNSS_Vu, GNSS_Vv, GNSS_Vw,...
    DCM_ECEF, INS_X, INS_Y, INS_Z, INS_Vu, INS_Vv, INS_Vw, bg_x, bg_y,
    bg_z, ba_x, ba_y, ba_z, P_matrix, delta_z, IMU_Lat, Acc_x, Acc_y,
    Acc_z, Alpha, KF_K_matrix)

% Inputs:
% Clock          System clock
% GNSS_X,Y,Z      GNSS ECEF position (m)
% GNSS_u,v,w      GNSS ECEF velocity (m/s)
% DCM_ECEF        prior body to ECEF DCM
% INS_Vu,Vv,Vw    prior INS velocity (m/s)
% INS_X,Y,X       prior INS position (m)
% bg_x,y,z        prior estimated gyroscope biases
% ba_x,y,z        prior estimated accelerometer biases
% P_matrix        Prior KF error covariance matrix
% IMU_Lat         previous latitude
% Acc_x,y,z       Accelerometer measurements
% Alpha          Weight factor when Fuzzy is applied
%
% Outputs:
% DCM_ECEF_new    New body to ECEF DCM
% KF_Vu,Vv,Vw     New velocity (m/s)
% KF_X,Y,Z        New position (m)
% KF_bg_x,y,z     prior estimated gyroscope biases
% KF_ba_x,y,z     prior estimated accelerometer biases
% P_matrix_new    updated Kalman filter error covariance matrix

%Create the vectors/scalars
P_matrix_new=zeros(15);
delta_z_new=zeros(6,1);
DCM_ECEF_new=zeros(3);
KF_x=0;
KF_y=0;
KF_z=0;
K_matrix=zeros(15,6);
KF_Vu=0;
KF_Vv=0;
KF_Vw=0;
KF_bg_x=0;
KF_bg_y=0;
KF_bg_z=0;
KF_ba_x=0;
KF_ba_y=0;
KF_ba_z=0;
GNSS_P= [GNSS_X GNSS_Y GNSS_Z]';
GNSS_V= [GNSS_Vu GNSS_Vv GNSS_Vw]';
```

```

INS_P=[INS_X INS_Y INS_Z]';
INS_P_in=[INS_X INS_Y INS_Z];
INS_V = [INS_Vu INS_Vv INS_Vw]';
IMU_bias=[ba_x ba_y ba_z bg_x bg_y bg_z]';

%Implement the EKF each second.
if ((floor(Clock)-Clock) == 0)

    k=floor(Clock);

    % Earth Constants
    Earth_Rot_Rate = 7.292115E-5; % Earth rotation rate in rad/s
    R_0 = 6378137; %WGS84 Equatorial radius in meters
    e = 0.0818191908425; %WGS84 eccentricity
    dt=1;%Update frequency
    mu = 3.986004418E14; %WGS84 Earth gravitational constant (m^3
s^-2)
    J_2 = 1.082627E-3; %WGS84 Earth's second gravitational constant
    g=zeros(3,1);

    % Skew symmetric matrix of Earth rate
    Skew_Rot_Rat = [ 0, -Earth_Rot_Rate, 0;...
                    Earth_Rot_Rate, 0, -0;...
                    -0, 0, 0];

    %Skew symmetric matrix of DCM_ECEF * Earth_Forces
    s=DCM_ECEF * Acc_Forces;
    F_e_2l=-[ 0, -s(3), s(2);...
             s(3), 0, -s(1);...
             -s(2), s(1), 0];

    %Geocentric radius
    geocentric_radius = R_0 / sqrt(1 - (e * sin(IMU_Lat))^2) *...
        sqrt(cos(IMU_Lat)^2 + (1 - e^2)^2 * sin(IMU_Lat)^2);

    %Gravity in the ECEF frame

    % Calculate distance from center of the Earth
    mag_r = sqrt(INS_P_in'' * INS_P_in');

    % Calculate gravity acceleration in ECEF
    a = 6378137; % WGS84 Equatorial radius in
meters.
    mu = 3.986004418E14; %WGS84 Earth gravitational constant (m^3
s^-2)
    J_2 = 1.082627E-3; %WGS84 Earth's second gravitational constant
    z_scale = 5 * (INS_P_in(1,3) / mag_r)^2;
    gamma = -mu / mag_r^3 * (INS_P_in(1,:)'' + 1.5 * J_2 * (a /
mag_r)^2 *...
        [(1 - z_scale) * INS_P_in(1,1)''; (1 - z_scale) *
INS_P_in(1,2)'';...
        (3 - z_scale) * INS_P_in(1,3)']]);
    g(1:2,1) = (gamma(1:2) + Earth_Rot_Rate^2 * INS_P_in(1,1:2)')';
    g(3,1) = gamma(3);

```

```

%Create the acc vector
Acc_Forces = [Acc_x, Acc_y, Acc_z]';

% Gyro noise PSD (rad^2/s)
gyro_noise_PSD = (5e-6)^2;
% Accelerometer noise PSD (m^2 s^-3)
accel_noise_PSD = (1e-3)^2;
% Accelerometer bias variation PSD (m^2 s^-5)
accel_bias_PSD_x = (1e-5)^2;
accel_bias_PSD_y = (1e-5)^2;
accel_bias_PSD_z = (1e-5)^2;
% Gyro bias variation PSD (rad^2 s^-3)
gyro_bias_PSD_x = (1e-8)^2;
gyro_bias_PSD_y = (1e-8)^2;
gyro_bias_PSD_z = (1e-8)^2;

% Position measurement noise SD per axis (m)
pos_meas_SD = 5;
% Velocity measurement noise SD per axis (m/s)
vel_meas_SD = 0.1;

% 1st phase - system initialization and propagation
% Determine transition matrix
Phi = eye(15);
Phi(1:3,1:3) = eye(3) - Skew_Rot_Rat * dt;
Phi(1:3,13:15) = DCM_ECEF * dt;
Phi(4:6,7:9) = eye(3) * dt;
Phi(7:9,1:3) = -F_e_21*dt;
Phi(7:9,7:9) = eye(3) - 2 * Skew_Rot_Rat * dt;
Phi(7:9,4:6) = -dt * 2 * g / geocentric_radius * INS_P' / sqrt
(INS_P' *INS_P);
Phi(7:9,10:12) = DCM_ECEF * dt;

% Determine approximate system noise covariance matrix
Q_INS = zeros(15);
Q_INS(1:3,1:3) = eye(3) * gyro_noise_PSD * dt;
Q_INS(7:9,7:9) = eye(3) * accel_noise_PSD * dt;
Q_INS(10:12,10:12) = [accel_bias_PSD_x, 0, 0; 0,
accel_bias_PSD_y, 0; 0, 0, accel_bias_PSD_z]* dt;
Q_INS(13:15,13:15) = [gyro_bias_PSD_x, 0, 0; 0,
gyro_bias_PSD_y, 0; 0, 0, gyro_bias_PSD_z]* dt;

% Propagate state estimates considering that all states % are
zero due to closed-loop correction.
x_est_propagated = zeros(15,1);

% Set-up measurement matrix using
H_mat = zeros(6,15);
H_mat(1:3,4:6) = eye(3);
H_mat(4:6,7:9) = eye(3);

```

```

    % Propagate state estimation error covariance matrix
    P_matrix_propagated = ((eye(15)+H_mat'*Alpha*H_mat)^2)*Phi
*(P_matrix + 0.5 * Q_INS) *Phi' + 0.5 * Q_INS;

    % 2nd Phase - measurement update

    % Measurement noise covariance matrix (R) assuming that
    % GNSS noise are not dependent
    R_GNSS = zeros(6,6);
    R_GNSS(1:3,1:3) = eye(3) * pos_meas_SD^2;
    R_GNSS(4:6,4:6) = eye(3) * vel_meas_SD^2;

    % Calculate Kalman gain
    K_matrix = P_matrix_propagated * H_mat' / (H_mat * ...
    P_matrix_propagated * H_mat' + (R_GNSS/((eye(6)+Alpha)^2)));

    % Formulate measurement innovations
    delta_z = zeros(6,1);
    delta_z(1:3,1) = INS_P-GNSS_P;
    delta_z(4:6,1) = INS_V-GNSS_V;
    delta_z_new=delta_z;

    % Update state estimates using
    x_est_new = x_est_propagated + K_matrix * delta_z;

    % Update state estimation error covariance matrix using
    P_matrix_new = (eye(15) - K_matrix * H_mat) *
P_matrix_propagated;

    %3rd phase - closed-loop correction

    %Calculate the skew of x_est_new
    s1=x_est_new(1:3,1);
    Sk_x_est=[    0, -s1(3,1),  s1(2,1);...
    s1(3,1),    0, -s1(1,1);...
    -s1(2,1),  s1(1,1),    0];

    % Correct attitude, velocity, and position using
    DCM_ECEF_new = (eye(3) - Sk_x_est) * DCM_ECEF;
    KF_V_new = INS_V - x_est_new(7:9);
    KF_P_new = INS_P - x_est_new(4:6);

    % Update IMU bias estimates
    IMU_bias_new = IMU_bias + x_est_new(10:15);
    % Update the INS states;
    KF_x=KF_P_new(1);
    KF_y=KF_P_new(2);
    KF_z=KF_P_new(3);
    KF_Vu=KF_V_new(1);
    KF_Vv=KF_V_new(2);
    KF_Vw=KF_V_new(3);
    KF_bg_x=IMU_bias_new(4);
    KF_bg_y=IMU_bias_new(5);
    KF_bg_z=IMU_bias_new(6);

```

```

        KF_ba_x=IMU_bias_new(1);
        KF_ba_y=IMU_bias_new(2);
        KF_ba_z=IMU_bias_new(3);
else
% Update the INS states when KF is not used;
P_matrix_new= P_matrix;
delta_z_new=delta_z;
DCM_ECEF_new=DCM_ECEF;
KF_x=INS_X;
KF_y=INS_Y;
KF_z=INS_Z;
KF_Vu=INS_Vu;
KF_Vv=INS_Vv;
KF_Vw=INS_Vw;
KF_bg_x=bg_x;
KF_bg_y=bg_y;
KF_bg_z=bg_z;
KF_ba_x=ba_x;
KF_ba_y=ba_y;
KF_ba_z=ba_z;
K_matrix=KF_K_matrix;
end

```

Appendix F.3 UKF model

```

%Perform the UKF and update the states
% Gabriel Giannini de Cunto - Rev 12 - 15/03/2020

function [DCM_ECEF_new, KF_x, KF_y, KF_z, KF_Vu, KF_Vv, KF_Vw,...
        KF_bg_x,KF_bg_y,KF_bg_z,KF_ba_x,KF_ba_y,KF_ba_z,P_matrix_new,
x_est_new, K_matrix_new] = INS_GNSS_UKF(Clock, GNSS_X, GNSS_Y,
GNSS_Z,GNSS_Vu, GNSS_Vv, GNSS_Vw,...
        DCM_ECEF, INS_X, INS_Y, INS_Z, INS_Vu, INS_Vv, INS_Vw, bg_x, bg_y,
bg_z, ba_x, ba_y, ba_z, P_matrix,IMU_Lat, Acc_x, Acc_y, Acc_z, x_est,
K_matrix, Alpha)

%Implements one cycle of the loosely coupled INS/GNSS
% Kalman filter plus correction of all INS states

% Inputs:
% GNSS_P          GNSS estimated ECEF user position (m)
% GNSS_V          GNSS estimated ECEF user velocity (m/s)
% DCM_ECEF        prior estimated body to ECEF coordinate
%                 transformation matrix
% INS_V           prior estimated ECEF user velocity (m/s)
% INS_P           prior estimated ECEF user position (m)
% IMU_bias        prior estimated IMU biases (body axes)
% P_matrix        previous Kalman filter error covariance matrix
% Earth_Forces    measured specific force
% IMU_Lat         previous latitude solution
% Alpha           Weight factor
%
% Outputs:
% DCM_ECEF_new    updated estimated body to ECEF coordinate

```

```

%           transformation matrix
%   INS_V_new   updated estimated ECEF user velocity (m/s)
%   INS_P_new   updated estimated ECEF user position (m)
%   IMU_bias_new updated estimated IMU biases
%       Rows 1-3       estimated accelerometer biases (m/s^2)
%       Rows 4-6       estimated gyro biases (rad/s)
%   P_matrix_new updated Kalman filter error covariance matrix

%Create the vectors
P_matrix_new=zeros(15);
DCM_ECEF_new=zeros(3);
K_matrix_new=zeros(15,6);
x_est_new = zeros(15,1);
KF_x=0;
KF_y=0;
KF_z=0;
KF_Vu=0;
KF_Vv=0;
KF_Vw=0;
KF_bg_x=0;
KF_bg_y=0;
KF_bg_z=0;
KF_ba_x=0;
KF_ba_y=0;
KF_ba_z=0;
Ts=0.01;
GNSS_P= [GNSS_X GNSS_Y GNSS_Z]';
GNSS_V= [GNSS_Vu GNSS_Vv GNSS_Vw]';
INS_P=[INS_X INS_Y INS_Z]';
INS_P_in=[INS_X INS_Y INS_Z];
INS_V = [INS_Vu INS_Vv INS_Vw]';
IMU_bias=[ba_x ba_y ba_z bg_x bg_y bg_z]';

%Implement the UKF each second.

if ((floor(Clock)-Clock) == 0)
    k=floor(Clock);

    % Constants
    Earth_Rot_Rate = 7.292115E-5; % Earth rotation rate in rad/s
    R_0 = 6378137; %WGS84 Equatorial radius in meters
    e = 0.0818191908425; %WGS84 eccentricity
    dt=1;%Update frequency
    mu = 3.986004418E14; %WGS84 Earth gravitational constant (m^3
s^-2)
    J_2 = 1.082627E-3; %WGS84 Earth's second gravitational constant
    g=zeros(3,1);

    %Gravity in the ECEF frame

    % Calculate distance from center of the Earth
    mag_r = sqrt(INS_P_in' * INS_P_in);

    % Calculate gravity acceleration in ECEF

```

```

a = 6378137; % WGS84 Equatorial radius in
meters.
mu = 3.986004418E14; %WGS84 Earth gravitational constant (m^3
s^-2)
J_2 = 1.082627E-3; %WGS84 Earth's second gravitational constant
z_scale = 5 * (INS_P_in(1,3) / mag_r)^2;
gamma = -mu / mag_r^3 * (INS_P_in(1,:)') + 1.5 * J_2 * (a /
mag_r)^2 * ...
    [(1 - z_scale) * INS_P_in(1,1)']; (1 - z_scale) *
INS_P_in(1,2)'; ...
    (3 - z_scale) * INS_P_in(1,3)']];
g(1:2,1) = (gamma(1:2) + Earth_Rot_Rate^2 * INS_P_in(1,1:2)')';
g(3,1) = gamma(3);

% Gyro noise PSD (rad^2/s)
gyro_noise_PSD_x = (5e-6)^2;
gyro_noise_PSD_y = (5e-6)^2;
gyro_noise_PSD_z = (5e-6)^2;
% Accelerometer noise PSD (m^2 s^-3)
accel_noise_PSD = (1e-3)^2;

% Accelerometer bias random walk PSD (m^2 s^-5)
accel_bias_PSD = (1e-5)^2;
% Gyro bias random walk PSD (rad^2 s^-3)
gyro_bias_PSD = (1e-8)^2;

% Position measurement noise SD per axis (m)
pos_meas_SD = 5;
% Velocity measurement noise SD per axis (m/s)
vel_meas_SD = 0.1;

%Create the acc vector
Acc_Forces = [Acc_x, Acc_y, Acc_z]';

% Skew symmetric matrix of Earth rate
Skew_Rot_Rat = [ 0, -Earth_Rot_Rate, 0;...
    Earth_Rot_Rate, 0, -0;...
    -0, 0, 0];

%Skew symmetric matrix of DCM_ECEF * Earth_Forces
s=DCM_ECEF * Acc_Forces;
F_e_21=[ 0, -s(3), s(2);...
    s(3), 0, -s(1);...
    -s(2), s(1), 0];

%Geocentric radius
geocentric_radius = R_0 / sqrt(1 - (e * sin(IMU_Lat))^2) * ...
    sqrt(cos(IMU_Lat)^2 + (1 - e^2)^2 * sin(IMU_Lat)^2);

% Determine transition matrix
Phi = eye(15);
Phi(1:3,1:3) = eye(3) - Skew_Rot_Rat * dt;
Phi(1:3,13:15) = DCM_ECEF * dt;
Phi(4:6,7:9) = eye(3) * dt;
Phi(7:9,1:3) = -F_e_21*dt;

```

```

Phi(7:9,7:9) = eye(3) - 2 * Skew_Rot_Rat * dt;
Phi(7:9,4:6) = -dt * 2 * g / geocentric_radius * INS_P' / sqrt
(INS_P' *INS_P);
Phi(7:9,10:12) = DCM_ECEF * dt;

%Determine approximate system noise covariance matrix
Q_INS = zeros(15);
Q_INS(1:3,1:3) = [gyro_noise_PSD_x, 0, 0; 0, gyro_noise_PSD_y,
0; 0, 0, gyro_noise_PSD_z]* dt;
Q_INS(7:9,7:9) = eye(3) * accel_noise_PSD * dt;
Q_INS(10:12,10:12) = eye(3) * accel_bias_PSD * dt;
Q_INS(13:15,13:15) = eye(3) * gyro_bias_PSD * dt;

%Set-up measurement matrix using
H_mat = zeros(6,15);
H_mat(1:3,4:6) = -eye(3);
H_mat(4:6,7:9) = -eye(3);

%Set-up measurement noise covariance matrix (R) assuming that
% GNSS position and velocity are independent and have equal
variance.
R_GNSS = zeros(6,6);
R_GNSS(1:3,1:3) = eye(3) * pos_meas_SD^2;
R_GNSS(4:6,4:6) = eye(3) * vel_meas_SD^2;

%
% -----
% UKF Main
%
% The x_est_propagated propagate state estimates considering
that all states are zero
% due to closed-loop correction.
x_est_propagated = zeros(15,1); %Previous state
n = length(x_est_propagated); % Number of states

%
% 1 - Calculate the lower-triangular Cholesky factor of P.
S0=chol(P_matrix, 'lower');

% %
% Eigenvector Choleksy factor
% [U,Lambda,V] = svd(P_matrix);
% S0 = U*sqrt(Lambda);

% 2 - Sigma points selection
c = sqrt(n);
Sigma_Selec=zeros(n,2*n);
Sigma_Selec(:,1:n) = x_est_propagated*ones(1,n)+c*S0;
Sigma_Selec(:,n+1:2*n) = x_est_propagated*ones(1,n)-c*S0;

% 3 Propagate each sigma point in to the model
Sigma_Propag=zeros(n,2*n);
for i=1:2*n
Sigma_Propag(:,i)= Phi*Sigma_Selec(:,i);
end

% 4 Propagate the states calculation
x_est= mean(Sigma_Propag,2);

```

```

Var_Sig=zeros(15,15,2*n);
for i=1:2*n
Var_Sig(:,:,i)=(Sigma_Propag(:,i)-x_est)*(Sigma_Propag(:,i)-
x_est)';
end
P_est=(mean(Var_Sig,3)+((eye(15)+H_mat'*Alpha*H_mat)^2)*Q_INS);

% 5 Generate new sigma points considering the P_est
%Calculate the lower-triangular Cholesky factor of P.
S1=chol(P_est,'lower');

%      %Eigenvector Choleksy factor
%      [U,Lambda,V] = svd(P_est);
%      S1           = U*sqrt(Lambda);

% Sigma points selection
Sigma_New=zeros(n,2*n);
Sigma_New(:,1:n) = x_est*ones(1,n)+c*S1;
Sigma_New(:,n+1:2*n) = x_est*ones(1,n)-c*S1;

x_est_new0=mean(Sigma_New,2);

% 6 sigma point and mean measurement innovations are calculated
z = zeros(6,1);
z(1:3,1) = INS_P-GNSS_P;
z(4:6,1) = INS_V-GNSS_V;

delta_z_Sigma=zeros(length(z),2*n);
for i=1:2*n
delta_z_Sigma(:,i)= z-H_mat*Sigma_New(:,i);
end

x_est=mean(delta_z_Sigma,2);

%7 Calculate the Covariance of the measurement
Var_delta_z=zeros(6,6,2*n);
for i=1:2*n
Var_delta_z(:,:,i)=(delta_z_Sigma(:,i)-
x_est)*(delta_z_Sigma(:,i)-x_est)';
end
Cov_Measurement=mean(Var_delta_z,3)+(R_GNSS);

%      %8 - Calcule the Kalman Gain
Var_gain=zeros(15,6,2*n);
for i=1:2*n
Var_gain(:,:,i)=(Sigma_New(:,i)-
x_est_new0)*(delta_z_Sigma(:,i)-x_est)';
end
%Kalman_Gain=mean(Var_gain,3)/(Cov_Measurement);
K_matrix=mean(Var_gain,3)/(Cov_Measurement);

%      %9 Update state estimates using
x_est_new = x_est_new0 + K_matrix * x_est;

```

```

%10 Update the state estimation error covariance matrix using
P_matrix_new=P_est-K_matrix*Cov_Measurement*K_matrix';

%CLOSED-LOOP CORRECTION
%Calculate the skew of x_est_new
s1=x_est_new(1:3,1);
Sk_x_est=[ 0, -s1(3,1), s1(2,1);...
          s1(3,1), 0, -s1(1,1);...
          -s1(2,1), s1(1,1), 0];

% Correct attitude, velocity, and position using
DCM_ECEF_new = (eye(3) - Sk_x_est) * DCM_ECEF;

KF_V_new = INS_V - x_est_new(7:9);

KF_P_new = INS_P - x_est_new(4:6);

% Update IMU bias estimates
IMU_bias_new = IMU_bias + x_est_new(10:15);

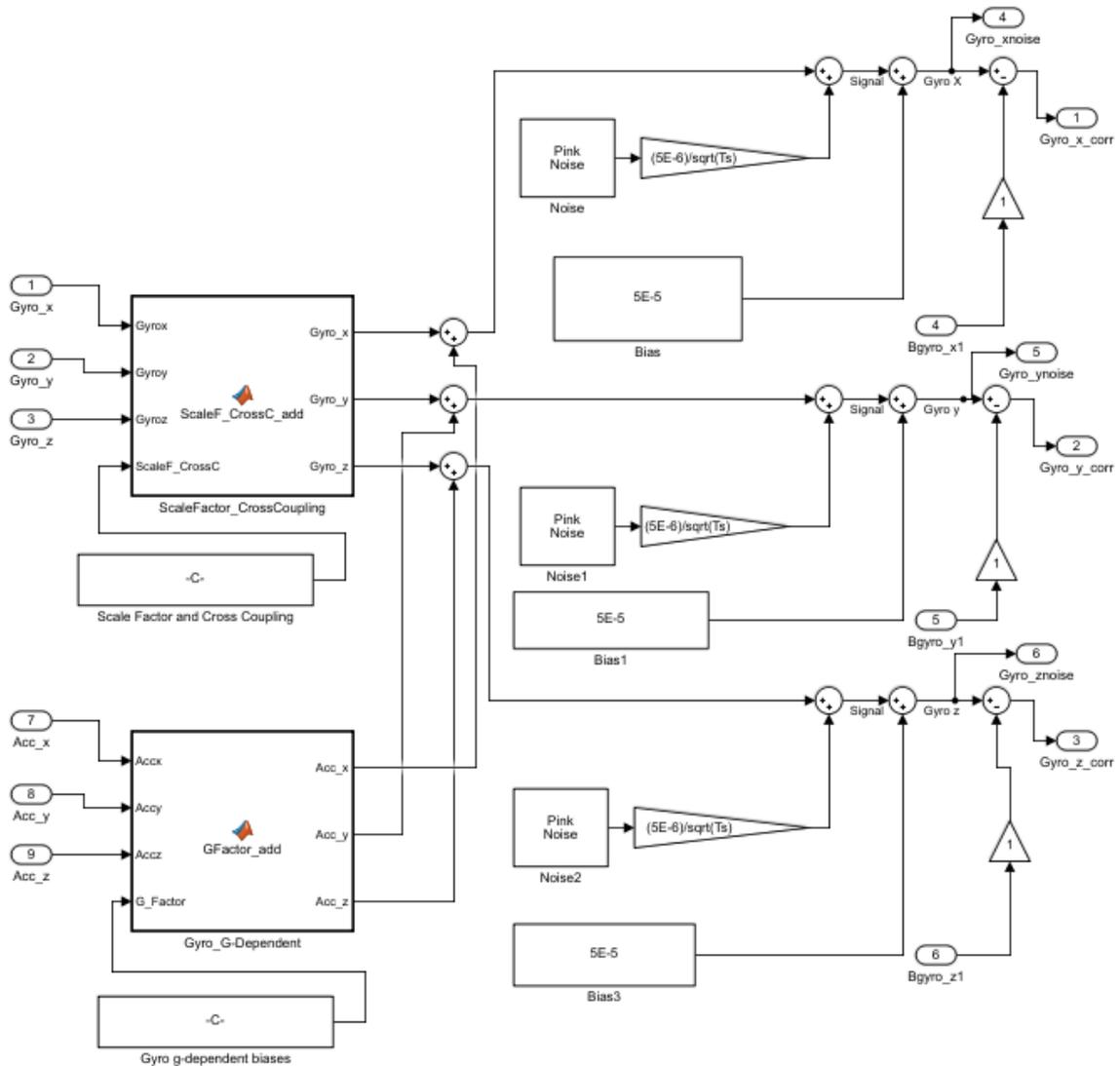
% Update the INS states;
KF_x=KF_P_new(1);
KF_y=KF_P_new(2);
KF_z=KF_P_new(3);
KF_Vu=KF_V_new(1);
KF_Vv=KF_V_new(2);
KF_Vw=KF_V_new(3);
KF_bg_x=IMU_bias_new(4);
KF_bg_y=IMU_bias_new(5);
KF_bg_z=IMU_bias_new(6);
KF_ba_x=IMU_bias_new(1);
KF_ba_y=IMU_bias_new(2);
KF_ba_z=IMU_bias_new(3);
K_matrix_new=K_matrix;
else
% Update the INS states when KF is not used;
P_matrix_new= P_matrix;
x_est_new=x_est;
K_matrix_new=K_matrix;
DCM_ECEF_new=DCM_ECEF;
KF_x=INS_X;
KF_y=INS_Y;
KF_z=INS_Z;
KF_Vu=INS_Vu;
KF_Vv=INS_Vv;
KF_Vw=INS_Vw;
KF_bg_x=bg_x;
KF_bg_y=bg_y;
KF_bg_z=bg_z;
KF_ba_x=ba_x;
KF_ba_y=ba_y;
KF_ba_z=ba_z;
end

```

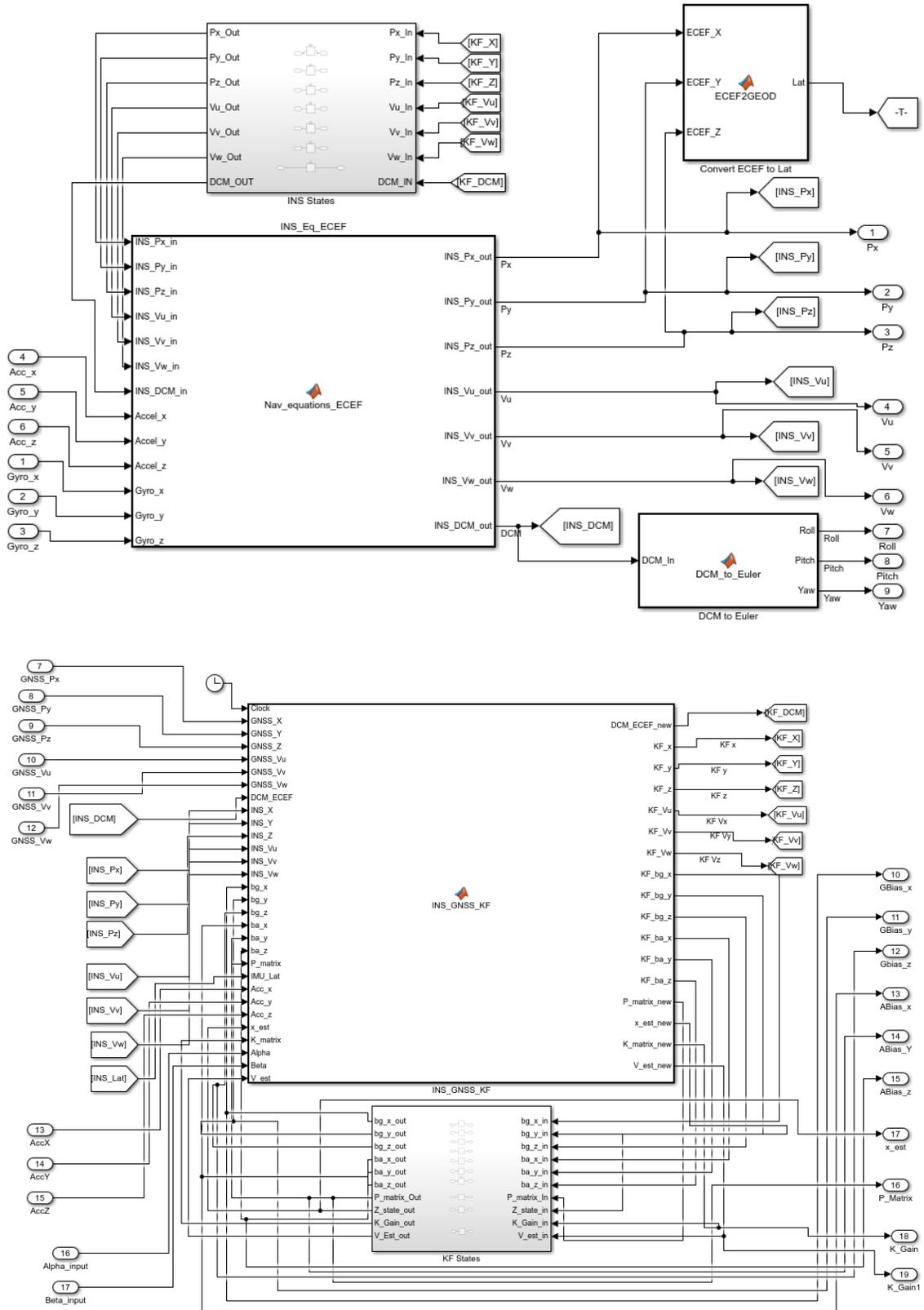
Appendix G - GNSS/INS integration Simulink model

This appendix contains samples of the SIMULINK model used to perform the GNSS/INS integration.

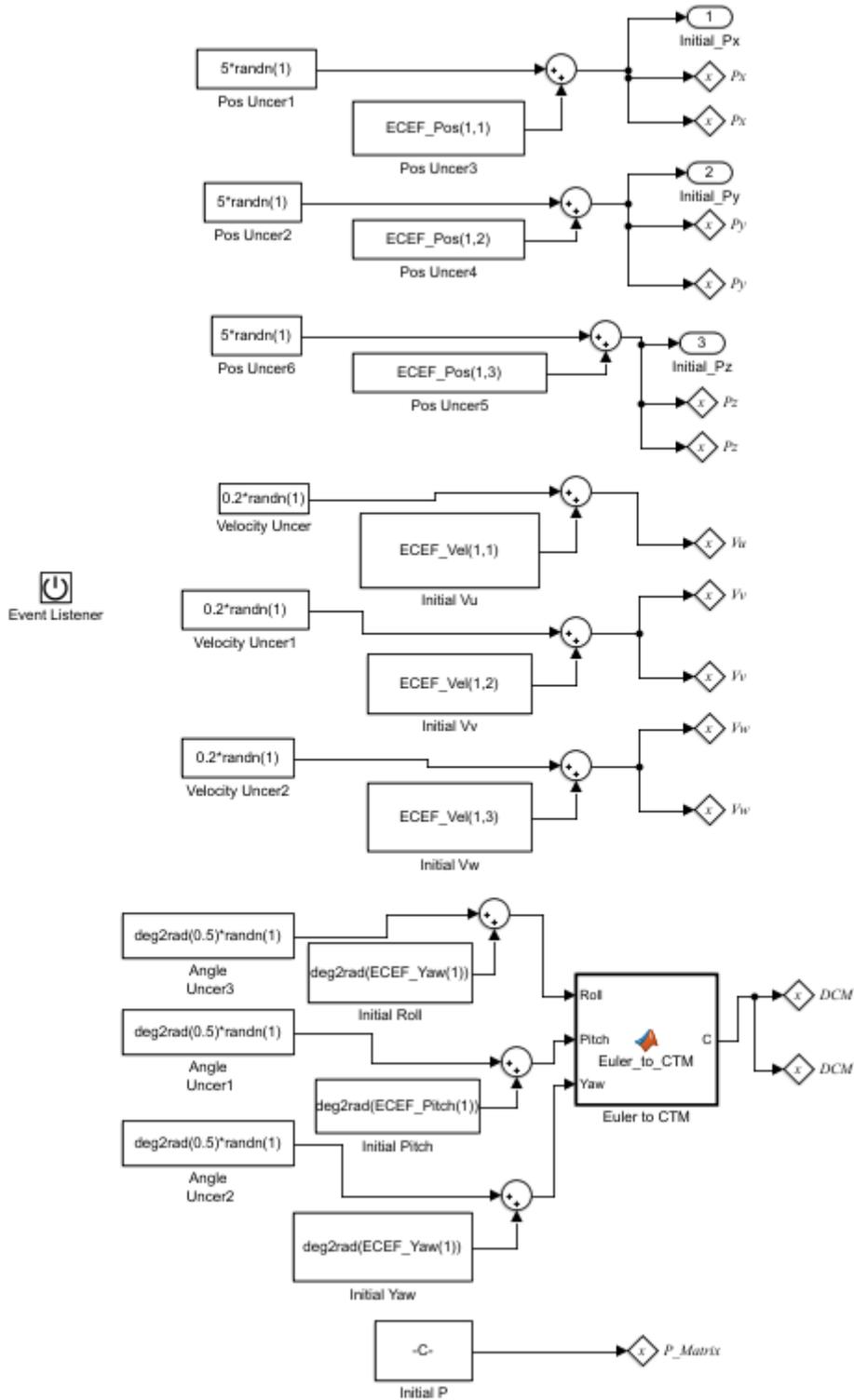
Appendix G.1 IMU noise model



Appendix G.2 INS/GNSS integration model



Appendix G.3 System Initialization



Appendix G.4 Fuzzy Logic Implementation

