MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/02.06.02.41-TDI

ON METHODS FOR IN-FLIGHT ALIGNMENT OF INERTIAL NAVIGATION SYSTEMS

Rian Koja

Master's Dissertation of the Graduate Course in Engineering and Space Tecnoloy, guided by Dr. Waldemar de Castro Leite Filho, approved in February 21, 2019.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/3SMGTAP>

INPE
São José dos Campos
2019

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

BOARD OF PUBLISHING AND PRESERVATION OF INPE INTELLECTUAL PRODUCTION - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):

Chairperson:

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CGCPT)

Members:

Dra. Carina Barros Mello - Coordenação de Laboratórios Associados (COCTE)

Dr. Alisson Dal Lago - Coordenação-Geral de Ciências Espaciais e Atmosféricas (CGCEA)

Dr. Evandro Albiach Branco - Centro de Ciência do Sistema Terrestre (COCST)

Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia e Tecnologia Espacial (CGETE)

Dr. Hermann Johann Heinrich Kux - Coordenação-Geral de Observação da Terra (CGOBT)

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação - (CPG)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SESID)

DIGITAL LIBRARY:

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SESID)

DOCUMENT REVIEW:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SESID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SESID)

ELECTRONIC EDITING:

Ivone Martins - Serviço de Informação e Documentação (SESID)

Murilo Luiz Silva Gino - Serviço de Informação e Documentação (SESID)

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/02.06.02.41-TDI

ON METHODS FOR IN-FLIGHT ALIGNMENT OF INERTIAL NAVIGATION SYSTEMS

Rian Koja

Master's Dissertation of the Graduate Course in Engineering and Space Tecnoloy, guided by Dr. Waldemar de Castro Leite Filho, approved in February 21, 2019.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/3SMGTAP>

INPE
São José dos Campos
2019

Cataloging in Publication Data

Koja, Rian.

K829m On methods for in-flight alignment of inertial navigation systems / Rian Koja. – São José dos Campos : INPE, 2019.

xxiv + 237 p. ; (sid.inpe.br/mtc-m21c/2019/02.06.02.41-TDI)

Dissertation (Master in Engineering and Space Tecnology) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2019.

Guiding : Dr. Waldemar de Castro Leite Filho.

1. Inertial navigation. 2. In-flight alignment. 3. GPS-aided navigation. 4. Covariance propagation. I.Title.

CDU 629.7.072

Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

Aluno (a): *Rian Koja*

Título: "ON METHODS FOR IN-FLIGHT ALIGNMENT OF INERTIAL NAVIGATION SYSTEMS"

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de ***Mestre*** em

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Helio Koiti Kuga

Presidente / INPE / SJCampos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Waldemar de Castro Leite Filho

Orientador(a) / INPE / SJCampos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Ronan Arraes Jardim Chagas

Membro da Banca / INPE / São José dos Campos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Jacques Waldmann

Convidado(a) / ITA/DCTA / São José dos Campos - SP

( ) Participação por Video - Conferência

( ) Aprovado *(X) Reprovado*

Este trabalho foi aprovado por:

(X) maioria simples

( ) unanimidade

São José dos Campos, 21 de fevereiro de 2019

ABSTRACT

This work reviews and adapts the framework for in-flight attitude initialization of an Inertial Navigation System with Position-Velocity Integration formulas (PIF and VIF), which are based on measurements from a GPS receiver and inertial sensors. It is shown that some shortcomings of such methods are more critical for a simplified comparison algorithm (TRIAD), which in turn helps to create a derived method (FIL), based on some logical conditions checks which allow precluding large alignment errors. The algorithms are then analyzed by developing an original on-line error estimation method, based on an estimation of the covariance of the involved vectors yielding a direction error for the vectors, which allows a decision criterion for online convergence declaration employed for two additional methods (OPT and OPTc). Results are validated with simulated data and Monte-Carlo tests are employed to assess the performance and validate some assumptions associated with those methods.

Keywords: Inertial Navigation. In-Flight Alignment. GPS-Aided Navigation. Covariance Propagation.

SOBRE MÉTODOS PARA ALINHAMENTO EM VOO DE SISTEMAS DE NAVEGAÇÃO INERCIAL

RESUMO

Este trabalho revisa e adapta os métodos de inicialização de atitude em voo de um Sistema de Navegação Inercial via Fórmulas de Integração Posição-Velocidade (PIF e VIF), os quais são baseados em medições provenientes de um receptor GPS e sensores inerciais. São apresentadas algumas limitações desses métodos, que são mais críticas para um algoritmo de comparação simplificado (TRIAD), o qual permite desenvolver um método derivado (FIL), baseado em verificações de condições lógicas que mitigam erros grandes de alinhamento. Estes algoritmos são então analisados pelo desenvolvimento de um método original de estimação de erro embarcado, baseado na covariância estimada dos vetores envolvidos, produzindo então um erro de direção dos vetores, que permite um critério de declaração de convergência empregado em dois métodos adicionais (OPT e OPTc). Resultados são validados com dados simulados e testes de Monte-Carlo são empregados para aferir o desempenho e validar hipóteses associadas a esses métodos.

Palavras-chave: Navegação Inercial. Alinhamento em Voo. Navegação Auxiliada por GPS. Propagação de Covariância.

LIST OF FIGURES

Page

1.1 Performance for VIF and PIF Methods initialized at 15s. 3
1.2 Performance for VIF and PIF methods initialized at 0s. 4
1.3 Performance for methods VIF, PIF and TRIAD initialized at 0s. 5
1.4 Performance VIF, PIF, TRIAD and FIL methods initialized at 0s. 6
1.5 Performance for all six methods initialized at 0s with comparison methods. 7

2.1 High-Level Block Diagram VIF and PIF Methods. 18
2.2 Comparison of norms for v vectors. 21
2.3 Comparison of norms for p vectors. 22
2.4 Comparison of norms for v vectors over long term. 26
2.5 Angle Discrepancy. 27
2.6 Computed Angles between α and β vectors. 28
2.7 Discrepancy Between Norms. 30

3.1 General block diagram for implementing a recursive formula. 35
3.2 Example of Estimated Covariance Matrix of a Stochastic Process 43
3.3 Example of Graph for Probability Density. 45
3.4 Colormap plot for Probability Density. 46
3.5 Colormap plot for natural logarithm of Probability Density. 47
3.6 Sample Estimation versus predicted variance for case 1 sequence. Blue for predicted, black for sample estimated. 50
3.7 Color map plot of sample estimated probability density versus predicted standard deviation for case 1 sequence. 51
3.8 Bandwidths used for Kernel estimation of case 1 sequence. 52
3.9 Sample Estimation versus predicted variance for case 2 sequence. Blue for predicted, black for sample estimated. 53
3.10 Color map plot of sample estimated probability density versus predicted standard deviation for case 2 sequence. 54
3.11 Bandwidths used for Kernel estimation of case 2 sequence. 55
3.12 Sample Estimation versus predicted variance for case 3 sequence. Blue for predicted, black for sample estimated. 56
3.13 Color map plot of sample estimated probability density versus predicted standard deviation for case 3 sequence. 57
3.14 Bandwidths used for Kernel estimation of case 3 sequence. 58

3.15 Sample Estimation versus predicted variance for case 4 sequence. Blue for predicted, black for sample estimated. 59
3.16 Color map plot of sample estimated probability density versus predicted standard deviation for case 4 sequence. 60
3.17 Bandwidths used for Kernel estimation of case 4 sequence. 61
3.18 Worst case for angle error within given probability. 81

4.1 Simulated Reference Specific Force over the IMU relative to Inertial Frame Expressed in Body Frame. 84
4.2 Simulated Reference Angular Rate of IMU Relative to Inertial Frame expressed in Body Frame. 85
4.3 Simulated Reference Attitude with respect to both Inertial and Navigation Frames. 86
4.4 Simulated Roll, Pitch and Yaw Angles. 87
4.5 Simulated Reference Position in geodetic coordinates. 87
4.6 Simulated Reference Velocity in Navigation Frame. 88
4.7 Simulated real values of v vectors. 94
4.8 Simulated norm of real values of v vectors. 94
4.9 Simulated real values of p vectors. 95
4.10 Simulated norm of real values of p vectors. 95
4.11 Performance of Methods under no integration or noise errors. 96
4.12 Norm of v vectors under no noise errors. 97
4.13 Norm of p vectors under no noise errors. 97
4.14 GPS provided Velocity in NED frame. 98
4.15 GPS provided Position. 99
4.16 IMU provided specific force. 99
4.17 IMU provided angular rate. 100
4.18 Evolution of α_v and β_v starting at 0s. 101
4.19 Evolution of α_p and β_p in module starting at 0s. 102
4.20 Evolution of α_p and β_p in Module starting at 0s. 103
4.21 Angles between vector pairs. 104
4.22 Angle Discrepancy between vector pairs. 105
4.23 Norm Discrepancy between vectors. 106
4.24 Estimated angle errors. 107
4.25 Estimated angle error bound. 108
4.26 Angle errors. 109
4.27 Evolution of α_v and β_v for simulation starting at 15s. 110
4.28 Evolution of α_p and β_p in module for simulation starting at 15s. 111
4.29 Evolution of α_p and β_p in Module for simulation starting at 15s. 112
4.30 Angles between vector pairs for simulation starting at 15s. 113

4.31 Angle Discrepancy between vector pairs for simulation starting at 15s. 114
4.32 Norm Discrepancy between vectors for simulation starting at 15s. . . . 115
4.33 Estimated angle errors for simulation starting at 15s. 116
4.34 Estimated angle errors for simulation starting at 15s. 117
4.35 Angle errors for simulation initialized at 15s. 118
4.36 Average vector errors for simulation starting at 0s. 121
4.37 Average vector errors for simulation starting at 15s. 122
4.38 Relative average vector errors for simulation starting at 0s. 123
4.39 Relative average vector errors for simulation starting at 15s. 124
4.40 Ratio of covariance maximum difference to sample estimate for α_v, for simulations starting at 0s. 125
4.41 Ratio of covariance maximum difference to sample estimate for β_v, for simulations starting at 0s. 126
4.42 Ratio of covariance maximum difference to sample estimate for α_v, for simulations starting at 15s. 127
4.43 Ratio of covariance maximum difference to sample estimate for β_v, for simulations starting at 15s. 128
4.44 Sample Covariance against Average Estimated Covariances for α_v for simulation starting at 0s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances. 129
4.45 Sample Covariance against Average Estimated Covariances for α_v for simulation starting at 15s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances. 130
4.46 Sample Covariance against Average Estimated Covariances for β_v for simulation starting at 0s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances. 131
4.47 Sample Covariance against Average Estimated Covariances for β_v for simulation starting at 15s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances. 132
4.48 Kernel Estimated Probability Density for α_v for simulation starting at 0s.133
4.49 Kernel Smoother Bandwidth for α_v for simulation starting at 0s. 134
4.50 Kernel Estimated Probability Density for β_v for simulation starting at 0s.135
4.51 Kernel Smoother Bandwidth for β_v for simulation starting at 0s. 136
4.52 Kernel Estimated Probability Density for α_v for simulation starting at 15s. 137

4.53 Kernel Smoother Bandwidth for α_v for simulation starting at 15s. . . . 138
4.54 Kernel Estimated Probability Density for β_v for simulation starting at 15s. 139
4.55 Kernel Smoother Bandwidth for β_v for simulation starting at 15s. . . . 140
4.56 Error below bound frequency for simulations starting at 0s. 141
4.57 Error below bound frequency for simulations starting at 15s. 142
4.58 α_v Error bound comparison for simulations starting at 0s. 143
4.59 α_v Error bound comparison for simulations starting at 15s. 144
4.60 β_v Error bound comparison for simulations starting at 0s. 145
4.61 β_v Error bound comparison for simulations starting at 15s. 146
4.62 Probability Density for Squared Norm Ratio Between calculated and reference value of α_v for simulations starting at 0s. 147
4.63 Probability Density for Squared Norm Ratio Between calculated and reference value of β_v for simulations starting at 0s. 148
4.64 Probability Density for Squared Norm Ratio Between calculated and reference value of α_v for simulations starting at 15s. 149
4.65 Probability Density for Squared Norm Ratio Between calculated and reference value of β_v for simulations starting at 15s. 150
4.66 Mean estimated angle errors comparison for simulations starting at 0s. 151
4.67 Mean estimated angle errors comparison for simulations starting at 15s. 152
4.68 Variance of estimated angle errors comparison for simulations starting at 0s. 153
4.69 Variance of estimated angle errors comparison for simulations starting at 15s. 154
4.70 Comparison of average methods performance for simulations starting at 0s. 155
4.71 Comparison of average methods performance for simulations starting at 15s. 156
4.72 Comparison of average methods performance with their one standard deviation interval for simulations starting at 0s. 157
4.73 Comparison of average methods performance with their one standard deviation interval for simulations starting at 15s. 158
4.74 Comparison of average methods performance with reference percentiles for simulations starting at 0s. 159
4.75 Comparison of average methods performance with reference percentiles for simulations starting at 15s. 160

A.1 Body frame over an INS. Image Courtesy of SBG Systems (2017). . . . 173
A.2 ECI and ECEF Frames. Circumflexes in this figure indicate unit vectors. 175

A.3 NED Frame. 177

LIST OF TABLES

Page

4.1 Simulated Data Specifications 93
4.2 Comparison table for Simulation starting at 0s. 161
4.3 Comparison table for Simulation starting at 15s. 162

LIST OF ABBREVIATIONS

CDF	–	Cumulative Distribution Function
CIO	–	Conventional International Origin
ECEF	–	Earth Centered Earth Fixed
ECI	–	Earth Centered Inertial
EKF	–	Extended Kalman Filter
FIL	–	Filtered Method
FIP	–	Finite Induction Principle
GNSS	–	Global Navigation Satellite System
GPS	–	Global Positioning System
ICRF	–	International Celestial Reference Frame
IGRF	–	International Geomagnetic Reference Field
IID	–	Independent and Identically Distributed
IMU	–	Inertial Measurement Unit
INS	–	Inertial Navigation System
NED	–	North-East-Down Frame
OPT	–	Optimal Estimate Method
OPTc	–	Optimal Chebyschev Bound Method
PDF	–	Probability Density Function
PIF	–	Position Integration Formula
SVD	–	Singular Value Decomposition
TRIAD	–	TRI-axial Attitude Determination
VIF	–	Velocity Integration Formula
WGS	–	World Geodetic System
ICRF	–	International Celestial Reference Frame

LIST OF SYMBOLS

$\mathcal{A}$	–	An Event Space
B	–	Body Frame
C_A^B	–	Rotation matrix that transforms coordinates in reference frame A to B
I	–	Earth-Centered Inertial Frame, Identity matrix
E	–	Earth-Centered Earth-Fixed, the Expectancy operator
f_k	–	A known 3x3 matrix sequence
f_X	–	The probability density function of the random variable X
h_k	–	A known 3x3 matrix
K	–	A covariance matrix, a matrix storing entries for SVD method
$\mathcal{M}(\mathbb{R})_{n\times m}$	–	The set of n by m Matrices with real elements
N	–	Navigation Frame
$\mathcal{O}$	–	Null element (vector or matrix of zeros)
q	–	Quaternion
u	–	A vector indicating rotation direction
$\bar{X}$	–	The expected value of the random variable X
$\widehat{x}$	–	An estimation for the value of the random variable X
α	–	Alpha vector (derived from IMU measurements)
β	–	Beta vector (derived from GPS measurements)
γ	–	Theoretical normal gravity on the ellipsoid surface
δ	–	Indicator of a variation-type error
Δ	–	An angle discrepancy, a variation
ε	–	A noise type error with 3x3 covariance matrix σ_ε^2
ζ	–	An event
η	–	A bias type error with 3x3 covariance matrix σ_μ^2
θ	–	An angle
λ	–	Geodetic Longitude
Λ	–	Ensemble Average
μ	–	A constant bias with covariance σ_μ^2
μ_X	–	The expected value of the random variable X
ξ_m	–	A white noise with 3x3 covariance matrix σ_ξ^2
σ^2	–	A Variance matrix
τ	–	Time (used as integration variable)
ϕ	–	Geodetic Latitude
Φ	–	A rotation angle
ω	–	Angular velocity
ω_{BC}^A	–	Angular velocity of frame C with respect to frame B expressed in frame A
$\otimes$	–	Quaternion Multiplication
$\boxtimes$	–	Cross product Covariance Operator
$\angle$	–	Angle (between vectors)
$\vert$	–	Probability conditioning

CONTENTS

Page

1 INTRODUCTION **1**
1.1 Chapters Outline 7

2 CONCEPTS ON ALIGNMENT OF INERTIAL SYSTEMS **9**
2.1 Attitude Determination and Estimation 9
2.1.1 Attitude Determination 9
2.1.1.1 TRIAD Method 9
2.1.2 Optimal Attitude Estimation 11
2.2 Alignment of Inertial Systems 11
2.2.1 Stationary Alignment 11
2.2.2 Semi-Dynamic Alignment 12
2.2.3 In-flight Alignment 13
2.2.3.1 Transfer Alignment 13
2.2.3.2 Position-Velocity Integration Formulas (VIF and PIF) 14
2.2.4 Velocity-Position Formulas with TRIAD Attitude Determination . . . 19
2.3 Remarks on Position-Velocity Formulas 20
2.3.1 Norm threshold 20
2.3.2 Norm Trendline 21
2.3.3 Angle Between Vectors 26
2.3.4 Norm Discrepancy 28
2.3.5 Filtered Method (FIL) 30

3 COVARIANCE MODELING **33**
3.1 Recursive Formula 33
3.2 Covariance of a Cross Product 35
3.3 General Error Equations 38
3.3.1 Basic Errors Definitions 38
3.3.2 Handling Operations Involving Errors 39
3.4 Working with Covariance Matrices 42
3.4.1 Estimating Covariance Matrices 42
3.4.2 Illustrating Covariance Matrices 42
3.4.3 Creating Samples for Given Covariance 43
3.5 Working with Distributions 44

3.5.1 Illustrating Distributions 44
3.5.2 Kernel Density Estimation 47
3.6 Covariance Propagation 49
3.6.1 Notation 49
3.6.2 Case 1: Sum of Known Gains Times Noise 50
3.6.3 Case 2: Sum of Known Gains Times Bias Cross Product Matrix Times Case 1 52
3.6.4 Case 3: Sum of Known Gains Times Noise Cross Product Matrix Times Case 1 55
3.6.5 Case 4: Sum of Known Gains Times Case 1 58
3.7 Error Analysis of the Methods 61
3.7.1 Alpha Error 62
3.7.2 Beta Error 75
3.7.3 From Variance to Angle Error 77
3.7.3.1 Expected Angle Method (OPT) 77
3.7.3.2 Bounded at Probability Method (OPTc) 80
3.8 Employing The Estimated Error 82

4 SIMULATIONS 83
4.1 Simulation Setup 83
4.1.1 Simulated Trajectory 83
4.1.2 Initial Parameters 85
4.1.3 Algorithm Inputs 88
4.1.4 Simulation Parameters 89
4.1.5 Assessing Performance Results 90
4.1.6 Implementation Particularities 90
4.2 Simulation Cases and Specification 92
4.2.1 Unsampled Noiseless Simulation 93
4.2.2 Sampled Noiseless Simulation 96
4.2.3 Performance Simulation 98
4.2.4 Monte-Carlo Simulations 118
4.2.4.1 Managing Seeds for Random Number Generation 119
4.2.4.2 Average Vector Error 120
4.2.4.3 Estimating Covariance 124
4.2.4.4 Assessing Bound at Probability 140
4.2.4.5 On the Approximations for the Expected Angle Error Formulation 146
4.2.4.6 Assessing Estimated Angle Error 150
4.2.4.7 Assessing Methods Performances 154

5 CONCLUSION **163**

REFERENCES **165**

ANNEX A - CONCEPTS FROM NAVIGATION SYSTEMS **171**
A.1 Reference Frames 171
A.1.1 Body Frame B 172
A.1.2 Inertial Frame I 173
A.1.3 Earth Frame E 174
A.1.4 World Geodetic System 175
A.1.5 Navigation Frame N 176
A.2 Quaternions 177
A.3 Basic Sensor Models 179
A.3.1 Accelerometer 179
A.3.2 Gyrometer 180
A.3.3 GPS 180
A.4 Numerical Integration 182
A.5 The Extended Kalman Filter 183

ANNEX B - CONCEPTS FROM STATISTICS AND STOCHASTIC SYSTEMS **187**
B.1 Definitions and Theorems from Random Variables Theory 187
B.1.1 Sets, Samples and Events 187
B.1.2 Basics of Probability Theory 188
B.1.3 Basics of Random Variable Theory 189
B.1.4 Expectation 190
B.1.5 Variance and Related Terms 192
B.1.6 Other Theorems 201

ANNEX C - NOTES ON LINEAR ALGEBRA **203**
C.1 Review on useful matrix properties 203
C.2 Tools from Linear Algebra Theory 206

ANNEX D - CODE TO VERIFY PROPERTIES OF THE VARIANCE OF A CROSS PRODUCT **211**

ANNEX E - PROOF OF RECURSIVE FORMULAS FOR COVARIANCE . **215**
E.0.1 Case 1 Proof: Sum of Known Gains Times Noise 215

E.0.2 Case 2 Proof: Sum of Known Gains Times Bias Cross Product Matrix Times Case 1 216
E.0.3 Case 3 Proof: Sum of Known Gains Times Noise Cross Product Matrix Times Case 1 223
E.0.4 Case 4 Proof: Sum of Known Gains Times Case 1 226

1 INTRODUCTION

An Inertial Navigation System is a device containing inertial sensors able to provide navigation information (e.g. position, velocity and attitude). It may perform pure inertial navigation, or it may use external aids such as satellite-based positioning and thus perform hybrid or aided navigation. Its output is called the Navigation Solution and generally is computed based on some form of Kalman Filter for aided systems, which requires initial knowledge of the states to be estimated, a process called Initialization. The initialization of attitude is a major problem when compared with that of position and velocity (TITTERTON; WESTON, 2004). A significant research effort is dedicated to the development of techniques to initialize attitude, due both to the possible divergence when out of the linear region assumed by the Kalman filter (CRASSIDIS, 2006) and the improvement in convergence time that a good alignment provides.

The alignment of an Inertial Navigation System (INS) consists in obtaining an estimate for the initial attitude of the platform, such that the navigation solution can start being computed. It inherits this name from a platform inertial system, which had accelerometers mounted over gimbals that sustain their axis physically aligned with the navigation frame (GROVES, 2013). Strapdown inertial systems are rigidly fixed to a vehicle of interest, have their accelerometers in a fixed position, and gimbals replaced by angular rate sensors. Within such configuration, the alignment is a purely mathematical process rather than a mechanical.

Alignment methods can be broadly classified as Transfer Alignment if the INS attitude is determined by an external Master INS or similar device and directly informed to the platform. An aided alignment method requires external signals or sensors, such as GPS measurements to allow the INS to compute its attitude. Finally, a so-called Self-Alignment process is performed when the platform determines its attitude based on the information it receives from its own sensors. An example of transfer alignment is when an air-launched missile is attached to the fuselage of a carrier aircraft, and the navigation solution of the INS on-board the aircraft is informed or transferred to the INS on the missile. Such process often relies on Kalman filtering techniques (PSZCZEL; BUCCO, 1992).

Furthermore, methods for alignment can be divided into Stationary Alignment, when the platform can be assumed to be fixed in attitude and position relative to the Earth, or as In-flight alignment when the INS can neither be requested to remain stationary nor controlled to be leveled for the benefit of alignment,

it may also be called in-motion alignment (ALI; USHAQ, 2009). Such techniques can be useful when aiding systems had been unavailable for a time period after which the propagated attitude is no longer reliable. Several works describe Kalman-filter like techniques for in-flight alignment, such as (PEI et al., 2014) or (ALI; USHAQ, 2009).

The assessment criteria for the performance of an alignment method lies both on its timeliness, i.e. how long does it take to provide an attitude estimate, and on its accuracy, which can be measured by how small is the angle between the computed and the true attitude of the platform. Both these aspects are considered and explored in Chapter 4, but should be observed on the figures of this section.

An interesting approach for in-flight alignment is presented in (WU; PAN, 2013) and related works such as (WU et al., 2011) that is not based on Kalman-Filtering techniques but rather, on optimal attitude estimation methods. These techniques are explained in Chapter 2, along with some related background. (WU; PAN, 2013) presents two similar methods referred as VIF and PIF for Velocity and Position Integration Formula The main goal of the current work is to analyze and improve upon these techniques. Examples of the estimation accuracy of these two methods are presented in this section, which also comments briefly on their behavior and compares with the developments original to this work. A comprehensive description of how these results are obtained is given in Chapter 4.

Figure 1.1 - Performance for VIF and PIF Methods initialized at 15s.

This profile resembles a typical estimation convergence curve, in which an initial large error decreases significantly then oscillates around a steady-state value. However, for the presented methods, depending on the trajectory performed by the vehicle and the noise characteristics of the sensors, some erratic behavior may present itself as shown in Figure 1.2.

Figure 1.2 - Performance for VIF and PIF methods initialized at 0s.

It is however demonstrated that such methods are not guaranteed to converge, as shown in (KOJA; LEITE FILHO, 2017) and reviewed in Chapter 2. Also, the proposed optimization is shown to possibly provide little or no gain over a deterministic attitude determination method based on the TRIAD attitude computation scheme, which in turn makes the method's flaws more evident. This is illustrated in Figure 1.3.

Figure 1.3 - Performance for methods VIF, PIF and TRIAD initialized at 0s.

The filtered method (indicated as FIL) was introduced in (KOJA; LEITE FILHO, 2017) and was seen to add a degree of robustness against this erratic behavior but in some cases requiring longer convergence periods, as shown in Figure 1.4.

Figure 1.4 - Performance VIF, PIF, TRIAD and FIL methods initialized at 0s.

Research devoted to overcoming such issues and developing on-line criteria to assert convergence is presented. This is achieved with analytical deductions and analyzing simulated data presented in Chapter 4. Briefly, a solution associated with the best expected accuracy can be picked during the estimation process according to statistical methods developed in Chapter 3. An example result that showcases the improvement is provided by Figure 1.5, where faster convergence than the filtered method is achieved although with lower accuracy.

Figure 1.5 - Performance for all six methods initialized at 0s with comparison methods.

1.1 Chapters Outline

The chapters of this work are organized in the following manner:

- Chapter 2 Reviews some concepts related to the alignment of inertial systems, revisits the techniques in (WU; PAN, 2013) and introduces a counterpart presented in (KOJA; LEITE FILHO, 2017).
- Chapter 3: Introduces concepts developed for this work related to recursive on-line estimation of covariances and proposes its application to the alignment methods this chapter provides the main contributions of the present work.
- Chapter 4: Explains the simulations employed to implement and test the proposed methods for in-flight alignment.
- Chapter 5: Presents final remarks and considerations regarding the findings of this work.

The annexes contain basic concepts in the concerned fields treated in consistent

manner with the main chapters and are quoted as needed for usage of theorems and properties. They do not present original contributions.

Chapter 2 begins with a bibliographical review, while Chapter 3 use concepts explained in the literature that are reviewed in the annexes. The scientific methodology is explained in Chapter 4. Noticeably, Chapter 3 also presents developments of this work and the simulations to validate them, but those are of mathematical nature.

Results indicated as *"theorems"* are known from the literature and indicted as such with specific or general citations, contributions of this work are called *"propositions"* and were not found in the literature as presented or in similar form.

2 CONCEPTS ON ALIGNMENT OF INERTIAL SYSTEMS

This chapter briefly explains some classic and new techniques employed for the alignment of inertial systems. A brief overview of these concepts has been presented in Chapter 1. The classical techniques are briefly mentioned, and respective references are provided. The Position-Velocity Integration formulas are the focus of this chapter. They are introduced in Section 2.2.3.2 and some contributions of this research follow afterwards.

2.1 Attitude Determination and Estimation

Classical initialization schemes rely on solving an attitude estimation problem from the measurements available to the INS. The problem of attitude determination is well-known in spacecraft navigation, and some developments come from this area. In purely inertial navigation, the INS propagates attitude and velocity based on previous knowledge and rate measurements (acceleration and angular rate), rather than on absolute measurements such as position and attitude. This creates the need for initial values to be either provided to the INS or estimated by it, a process called initialization. Even with aided navigation, techniques such as Kalman filtering require initial states to be provided, and propagation of previous knowledge is still performed. Hence the need for an initial determination of the states, in particular the attitude, which is not measured by conventional GPS devices.

A distinction made in the literature concerns “Attitude Determination" which consists in obtaining attitude information from the minimum necessary amount of information and “Optimal Attitude Estimation" which consists in finding a solution to some optimization problem based on possibly redundant attitude information (SHUSTER; OH, 1981).

2.1.1 Attitude Determination

For the purpose of this work, only the TRIAD method will be explored.

2.1.1.1 TRIAD Method

The TRIAD algorithm was first described on (BLACK, 1964) and became popular both for being a very simple solution to the attitude (deterministic) determination problem. Its name stands for TRI-axial Attitude Determination (MARKLEY, 1999). It is used both on the current work and for simple stationary alignment

techniques presented on Section 2.2.1.

Given two unit vectors u^A and v^A expressed in A frame, and their counterparts u^B and v^B expressed in B frame, the rotation matrix between such frames C_A^B can be computed by:

$$M_A = \left[u^A, \frac{u^A \times v^A}{\|u^A \times v^A\|}, \frac{u^A \times (u^A \times v^A)}{\|u^A \times (u^A \times v^A)\|} \right] \tag{2.1a}$$

$$M_B = \left[u^B, \frac{u^B \times v^B}{\|u^B \times v^B\|}, \frac{u^B \times (u^B \times v^B)}{\|u^B \times (u^B \times v^B)\|} \right] \tag{2.1b}$$

$$C_A^B = M_B (M_A)^T \tag{2.1c}$$

If the vectors u^A and v^A and their counterparts in B frame are parallel, then $u^A \times v^A$ becomes zero, and therefore the method yields a matrix with undetermined entries. This singularity encapsulates the fact that two pairs of vectors are the minimum amount of information needed to compute a frame rotation. If the pairs are parallel, they represent a single vector, when two distinct vectors and their representations in each frame are necessary.

When noisy vectors are applied to this method, it is generally recommended using the most accurate one as the first (u^A) and the most uncertain as the second (v^A), because the resulting matrix C_A^B will yield $u^B = C_A^B u^A$ (exact equality) while $v^B \approx C_A^B v^A$ (approximate result) (MARKLEY; CRASSIDIS, 2014). Since both need to be unit vectors, it may also be necessary to normalize them, specially if the vectors used are not originally unitary as in the case shown in Section 2.2.4. Normalization for the sake of precluding numerical errors is more relevant when vectors are unitary by definition, such as the sun direction, but may be corrupted by numerical inaccuracies. There are implementations of TRIAD-like methods when the used vectors are not necessarily unitary nor normalized, and an inverse matrix rather than a transposed is used in Equation (2.1c). Some of these implementations, applied to stationary alignment, are explained and analyzed on (SILVA et al., 2016).

2.1.2 Optimal Attitude Estimation

Optimal attitude estimation traces back to the Wahba's problem (WHABA, 1965) and it is a least square solution to the problem of determining attitude when several measured vectors u_k are available and the following cost functional J is to be minimized:

$$J = \frac{1}{2} \sum_k a_k \left| u_k^B - C_A^B u_k^A \right|^2 \tag{2.2}$$

Where a_k is a positive weighting coefficient. Several algorithms to find the solution of this problem exist, one of the most popular being the QUEST algorithm (SHUSTER; OH, 1981). The Singular Value Decomposition (SVD) method will be used in this work for simplicity and robustness of implementation, and shown here for its analytical insights (MARKLEY; MORTARI, 2000), despite it being computationally more expensive.

Theorem 2.1.1. Singular Value Decomposition (SVD) Method:
Let a sequence of available vector measurements u_k^A and u_k^B be known along with a normalized weighting sequence a_k. The optimal rotation matrix estimated from those entries to minimize Equation (2.2) can be computed by:

$$K = \sum_k a_k \, u_k^B \, (u_k^A)^T \tag{2.3a}$$

$$K = U \, \Sigma \, V^T = U \, \mathrm{diag}\left([\Sigma_{11}, \, \Sigma_{22}, \, \Sigma_{33}] \right) \, V^T \tag{2.3b}$$

$$C_A^B = U \, \mathrm{diag}\left([1, \, 1, \, \det(U)\det(V)] \right) \, V^T \tag{2.3c}$$

Where U and V are unitary matrices and $\Sigma_{11} \geq \Sigma_{22} \geq \Sigma_{33} \geq 0$.

A proof of this method based on (MARKLEY, 1988) is presented after the statement of Theorem C.2.1 on Section C.2.

2.2 Alignment of Inertial Systems

2.2.1 Stationary Alignment

Procedures for stationary alignment, also called quasi-stationary alignment (GROVES, 2013), assume the position has been initialized and the inertial system

has no kinematic acceleration during the initialization period, such that the measured specific force in body frame matches the local gravity reaction $f^B = -g^B$, then the measured angular rate of the platform also matches the Navigation frame rate with respect to the inertial frame, which for a stationary object corresponds to the Earth's rotation: $\omega_{BI}^B = \omega_{NI}^B = \omega_{EI}^B$. Alternatively, the magnetic field may be used depending on the application (FARRELL, 2008). Both the local gravity and magnetic field can be computed from models such as the World Geodetic System WGS84 (DECKER, 1986) and the International Geomagnetic Reference Field (IGRF12) (THÉBAULT et al., 2015) respectively.

In navigation frame, the Earth's angular rate and local gravity can be expressed by:

$$\omega_{IE}^N = \begin{bmatrix} \omega_{IE}\cos(\phi) & 0 & -\omega_{IE}\sin(\phi) \end{bmatrix}^T \tag{2.4}$$

$$g^N = \begin{bmatrix} 0 & 0 & \gamma(\phi) \end{bmatrix}^T \tag{2.5}$$

With ϕ the geodetic latitude defined in Section A.1.4 and $\gamma(\phi)$ the local gravity defined on Equation (A.12).

This scheme is often called "Coarse Alignment" (FARRELL, 2008), in contrast with "Fine Alignment", which is a second step that requires the initial output of the former. There are several variations of such methods, as explained in (LEITE et al., 2017). As an example, one such approach uses a QUEST-like algorithm instead of TRIAD and a refinement based on neural-networks (CAMPOS et al., 2017).

2.2.2 Semi-Dynamic Alignment

It is possible to perform experiments with an INS installed on a turn-tables, which allow creating controlled motions for the INS depending on the specification of the equipment. This kind of experiment may be used to calibrate sensors, and during the experiment an alignment procedure may be carried as well. A technique based on an Unscented Kalman Filter method to compute both the attitude and sensor biases during one such experiment is described in (KUGA et al., 2007).

2.2.3 In-flight Alignment

In-flight alignment ditches the assumption of stationarity and is often needed in situations with stringent time convergence requirements. A few examples when such techniques are required can be listed:

- When a missile is to be launched from a carrier aircraft, its navigation system, that likely was turned off, needs to be initialized before detaching from the fuselage (TITTERTON; WESTON, 2004).

- When an airplane needs to initialize the INS either for first start or mid-flight, to reduce gate delays or to improve reliability (WEED et al., 2004).

- When a missile is launched from a ground, sea of air vehicle without initializing navigation due to schedule constraints and the navigation system must start providing navigation solutions at a given time of the flight envelope (HYSLOP et al., 1990).

There are also works which exploit the enhanced observability characteristics of the alignment scheme when it is possible to require a specific maneuvering, such as employing piece-wise constant specific force segments from the vehicle or rotating the INS within it as in (WALDMANN, 2007b) and (WALDMANN, 2007a). For the purpose of the current work, an arbitrary trajectory is desired, considering a strapdown inertial system stationary with respect to the vehicle, although a single trajectory will be used for Monte Carlo tests described on Section 4.2.4.

2.2.3.1 Transfer Alignment

As mentioned in Chapter 1, it is possible for a carrier vehicle, hosting a master INS to transfer its state knowledge to a slave INS embarked on a payload (BAZIW; LEONDES, 1972). This can be done as a one-shot transfer, which would provide coarse accuracy, but can proceed with techniques such as angular rate or velocity matching which will help the slave INS to converge despite misalignment between both body frames and alleviate the degradation of performance caused by structural flexibility of the arrangement. These techniques generally rely on some kind of Kalman filtering and are described in detail on (PSZCZEL; BUCCO, 1992).

2.2.3.2 Position-Velocity Integration Formulas (VIF and PIF)

When no external vehicle or source of attitude information can be assumed available, the INS needs to compute an attitude with its own means, a process that is called Self-Alignment (FARRELL, 2008). The Position-Velocity Formulas are a set of equations that provide a method for in-flight alignment with GPS aid. They are described in (WU; PAN, 2013), (WU et al., 2011) and other related works. Some derived methods are also presented in (WU et al., 2014), in which the biases of sensors and the GPS lever arm are estimated. It is noted that in most of those publications, the GPS is assumed to provide measurements at a fairly high rate, ranging from 2Hz (WU; PAN, 2013) with interpolation, up to 50Hz in (WU et al., 2014), while in this work a rate of 1Hz with no interpolation is used, with other relevant parameters described in Chapter 4.

Having established optimal attitude estimation methods in Section 2.1.2, they can be applied to INS alignment if a set of vectors is known in both Body frame (B) and Navigation Frame (N, or NED for North-East-Down). This allows computing the matrix C_B^N.

Throughout this work, frames with unspecified time are to be understood as with respect to current time (i.e., $N = N(t)$), while 0 stands for the starting time of the alignment process. In the following deduction, a color scheme is employed to indicate frames related to IMU measurements (red) and frames related to GPS measurements (blue).

Some equations in this section will be color coded to associate mathematical entities with the sensor they derive from. Red symbols indicate objects computed based on IMU data and blue indicates relation to the GPS output.

While most filtering techniques try to improve the estimation of C_B^N as t progresses, it is noted that:

$$C_B^N = C_{N(0)}^{N(t)} \, C_{B(0)}^{N(0)} \, C_{B(t)}^{B(0)} \tag{2.6}$$

$$C_B^N = C_{E(t)}^{N(t)} \, C_{E(0)}^{E(t)} \left(C_{E(0)}^{N(0)} \right)^T C_{B(0)}^{N(0)} \, C_{B(t)}^{B(0)} \tag{2.7}$$

Where E stands for the Earth-Fixed Earth-Centered (ECEF) frame. $C_{E(t)}^{N(t)}$ can be

computed directly from GPS position measurements, and its initial value $C_{E(0)}^{N(0)}$ can be stored. A constant Earth rotation rate model is used to compute $C_{E(0)}^{E(t)}$, as described on Section A.1.4.

Meanwhile, $C_{B(t)}^{B(0)}$ can be initialized with identity ($C_{B(0)}^{B(0)} = I$), and updated at high rate with measurements from the Inertial Measurement Unit (IMU), by using quaternions:

$$\hat{u} = \omega_{BI}^{B} / \left\| \omega_{BI}^{B} \right\| \tag{2.8a}$$

$$q_{B(t-dt)}^{B(t)} = \left[\cos(|\omega_{BI}^{B}| dt/2), \sin(|\omega_{BI}^{B}| dt/2) \hat{u}^{T} \right]^{T} \tag{2.8b}$$

$$q_{B(t)}^{B(0)} = q_{B(t-dt)}^{B(0)} \otimes q_{B(t)}^{B(t-dt)} \tag{2.8c}$$

Where $\otimes$ stands for quaternion multiplication, further details regarding the properties and conventions of quaternions is given in Section A.2. All frame transformations are performed with quaternions for the simulations described in Section 4, even the matrices computed with TRIAD and SVD methods are converted to quaternions with code adapted from the Propat Library (CARRARA, 2015). Normalization and sign checks are also performed. The analytical developments are done with matrices for the sake of conciseness.

The technique proposed in (WU; PAN, 2013) starts from the well-known kinematic equation (FARRELL, 2008):

$$\dot{v}^{N} = C_{B}^{N} f^{B} - (2\omega_{IE}^{N} + \omega_{EN}^{N}) \times v^{N} + g^{N} \tag{2.9}$$

Notably, assuming availability of GPS position and velocity measurements, the rotation rates and the local gravity acceleration can be computed with on-board models. Applying the expansion from equation 2.6:

$$\dot{v}^{N} = C_{N(0)}^{N(t)} C_{B(0)}^{N(0)} C_{B(t)}^{B(0)} f^{B} - (2\omega_{IE}^{N} + \omega_{EN}^{N}) \times v^{N} + g^{N} \tag{2.10}$$

Collecting alike terms:

$$C_{N(t)}^{N(0)} \left[\dot{v}^N + (2\omega_{IE}^N + \omega_{EN}^N) \times v^N - g^N \right] = C_{B(0)}^{N(0)} C_{B(t)}^{B(0)} f^B \tag{2.11}$$

The initial attitude matrix $C_{B(0)}^{N(0)}$ is the unknown constant to be estimated. The other right-hand side terms are computed from the IMU measurements. The left-hand side terms can be computed directly from the GPS measurements, except for $\dot{v}^N$, which can be eliminated by proceeding a time-integration:

$$\int_0^t C_{N(\tau)}^{N(0)} \dot{v}^N \, d\tau = C_{N(t)}^{N(0)} v^N - v(0) - \int_0^t C_{N(\tau)}^{N(0)} \left(\omega_{IN}^N \right)^\times v^N d\tau \tag{2.12}$$

Where $(.)^\times$ denotes the cross-product matrix. By integrating both sides of Equation (2.11), one obtains, as in (WU; PAN, 2013):

$$\alpha_v = \int_0^t C_{B(\tau)}^{B(0)} f^B d\tau \tag{2.13a}$$

$$\beta_v = C_{N(t)}^{N(0)} v^N - v^N(0) + \int_0^t C_{N(\tau)}^{N(0)} \left[\left(\omega_{IE}^N \right)^\times v^N - g^N \right] d\tau \tag{2.13b}$$

$$\beta_v = C_{B(0)}^{N(0)} \alpha_v \tag{2.13c}$$

The practical implementation of those equations is discussed on Section 4.1.6.

Hence, α_v and β_v can be used as entries for an optimal estimation method, with the K matrix from Equation (2.3c) being updated at each time step t_k. This is referred to as the "Velocity Integration Formula" (VIF) in (WU; PAN, 2013). Its solution at each time step t_k will be denoted as $\left(C_{B(0)}^{N(0)} \right)_k^{VIF}$. It comes from the SVD algorithm (Theorem 2.1.1) after building the K matrix in the following manner:

$$K_k^{VIF} = \sum_{i=1}^{k} \beta_v(t_i) \left(\alpha_v(t_i) \right)^T \tag{2.14}$$

It is noted that implicitly, the ponderation sequence a_k exists in this equation by recalling that the vectors should have been normalized. Therefore, it can be said that:

$$a_k^{VIF} = \|\beta_v(t_k)\| \, \|\alpha_v(t_k)\| \tag{2.15}$$

It will be shown in Section 2.3.2 that this generally tends to give higher importance to the latest data.

By integrating over time again, new vectors α_p and β_p are obtained:

$$\begin{aligned} \alpha_p &= \int_0^t \alpha_v d\tau \\ &= \int_0^t \int_0^{\tau_2} C_{B(\tau_1)}^{B(0)} f^B d\tau_1 d\tau_2 \end{aligned} \tag{2.16}$$

$$\begin{aligned} \beta_p &= \int_0^t \beta_v d\tau \\ &= \int_0^t C_{N(\tau)}^{N(0)} v^N d\tau - t v^N(0) + \int_0^t \int_0^{\tau_2} C_{N(\tau_1)}^{N(0)} \left[\left(\omega_{IE}^N \right)^\times v^N - g^N \right] d\tau_1 d\tau_2 \end{aligned} \tag{2.17}$$

Accumulating such vectors in the K matrix and estimating $C_{B(0)}^{N(0)}$ is what (WU; PAN, 2013) refers to as "Position Integration Formula" (PIF). Its solution at each time step t_k will be denoted as $\left(C_{B(0)}^{N(0)} \right)_k^{PIF}$. It comes from the SVD algorithm after building the K matrix in the following manner:

$$K_k^{PIF} = \sum_{i=1}^{k} \beta_p(t_i) \left(\alpha_p(t_i) \right)^T \tag{2.18}$$

Given that such quantities are corrupted by noise, they should always span the whole Euclidean space, thus avoiding singularities on the SVD method, yet not granting accuracy of the solution.

A block diagram is provided in Figure 2.1 to give an overview of the information flow of these methods for implementation. A summary of the equations involved and their sequencing is provided on (WU; PAN, 2013).

Figure 2.1 - High-Level Block Diagram VIF and PIF Methods.

Notably, this method assumes that the calculation of both α_v and β_v is synchronized. For implementation purposes, both calculations are only started when the first valid GPS measurement is available. This implies that the first IMU measurement must be associated with nearly the same time as the GPS first measurement. Due to different bus delays this may call for a memory buffer to store IMU measurements. Even then, the measurement times will not be an exact match, but

this is alleviated by a high enough sample rate from the IMU (100Hz for the current work). Causing a maximum mismatch in the order of 10 milliseconds, which should introduce introduces errors that are be small compared to the other noises and integration errors in the system.

2.2.4 Velocity-Position Formulas with TRIAD Attitude Determination

If all four vectors α_v, α_p, β_v and β_p are computed, then two pairs of vectors are available, each pair expressed in a respective reference frame. Therefore, the TRIAD algorithm explained in Section 2.1.1.1 can be used to determine the initial attitude:

$$M_{B(0)} = \left[\frac{\alpha_v}{\|\alpha_v\|}, \frac{\alpha_v \times \alpha_p}{\|\alpha_v \times \alpha_p\|}, \frac{\alpha_v \times (\alpha_v \times \alpha_p)}{\|\alpha_v \times (\alpha_v \times \alpha_p)\|} \right] \tag{2.19a}$$

$$M_{N(0)} = \left[\frac{\beta_v}{\|\beta_v\|}, \frac{\beta_v \times \beta_p}{\|\beta_v \times \beta_p\|}, \frac{\beta_v \times (\beta_v \times \beta_p)}{\|\beta_v \times (\beta_v \times \beta_p)\|} \right] \tag{2.19b}$$

$$C_{B(0)}^{N(0)} = (M_{N(0)})^T M_{B(0)} \tag{2.19c}$$

This method will be used as a comparison counterpart of the VIF and PIF methods and will generally be referred as the TRIAD method. Due to its simplicity, a performance improvement is not generally expected. However, it will be seen that the shortcomings of this method also have impacts on the other previous ones, but its pronounced effect contributes to the investigation of the influence of flight and noise conditions on performance.

The used vectors are all associated with the same time instant (t) for the purpose of this work, but since the matrix to be estimated by this method is constant ($C_{B(0)}^{N(0)}$), mismatched times could be used as long as each pair that represents the same vector in different frames keeps the same instant. For example, if the first pair is $\alpha_v(t_k)$ and $\beta_v(t_k)$, the second pair could be $\alpha_p(t_j)$ and $\beta_p(t_j)$. This kind of approach was however not presently pursued.

2.3 Remarks on Position-Velocity Formulas

This section is devoted to point some conditions under which the reference algorithms (VIF and PIF) and the proposed TRIAD fail to converge or are expected to deliver poor performance, which also helps to identify on-line if the output of this method can be declared as converged. This is relevant since most Kalman filter navigation systems are expected to be initialized with a valid initial state. The remarks may be illustrated with figures that are based on simulations, which are explained only in Chapter 4. Some basic convergence criteria are reproduced here in more details but were generally introduced in (KOJA; LEITE FILHO, 2017). Different data sets were used to produce this section, with some similar to the aforementioned reference, as well as the same data used in Chapter 4 and a quasi-stationary test case.

2.3.1 Norm threshold

Remark 2.3.1. Invalid if free-falling:
A first shortcoming of such method occurs when the INS is free-falling, since under this condition, the measured specific force should be zero $f^B = 0$, though it will accumulate noise and sensor errors (e.g. bias). This causes α_v and consequently α_p to be invalid. Additionally, a smaller magnitude for those vectors which are produced from noisy measurements, implies smaller signal-to-noise ratio.

It is then suggested to verify that the used vectors have their norm above a certain threshold, in order to assume they're usable.

$$\|\alpha_v\| > \|\alpha_v\|_{min} \tag{2.20a}$$

$$\|\alpha_p\| > \|\alpha_p\|_{min} \tag{2.20b}$$

$$\|\beta_v\| > \|\beta_v\|_{min} \tag{2.20c}$$

$$\|\beta_p\| > \|\beta_p\|_{min} \tag{2.20d}$$

2.3.2 Norm Trendline

Despite the previous claim, many INS use cases involve small accelerations compared with the gravity magnitude of roughly $9,8\mathrm{m/s}^2$. Hence the norm of these vectors tend to increase at a rate around $g_0\,\mathrm{m/s}$ for α_v depending as well on attitude changes. Likewise, a norm around $t^2 g_0\,/2\,\mathrm{m}$ is expected for α_p. This also causes the last computed sets of vectors to be more relevant than the first ones, since the lack of normalization on the aforementioned vectors causes the norm to take the role of the weights a_k in Equation (2.2). These trends, along with the computed norms for each vector in simulated quasi-stationary data are shown in Figure 2.2 and Figure 2.3.

In other words, the norms for α_v and β_v are not expected to be or remain small for any practical applications, in this sense, vehicles with large acceleration profiles should also cause large norms for those vectors, although considerations on the trajectory should be considered.

Figure 2.2 - Comparison of norms for v vectors.

Figure 2.3 - Comparison of norms for p vectors.

For longer simulations however, this trendline will not be sustained. A finer observation can be derived, which is shown in Figure 2.4, and stated as Proposition 2.3.1. It is first necessary to define the cardinal sine function:

Definition 2.3.1. Cardinal Sine Function:
The function denoted as sinc is given by (WOODWARD, 1953):

$$\text{sinc(x)} = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \tag{2.21}$$

Proposition 2.3.1. Trend Line for v Vectors:
A reference magnitude for α_v and β_v can be computed by:

$$\|\beta_v|_{v^N=\mathcal{O}}\| = g_0 t \sqrt{\sin^2(\phi) + \cos^2(\phi)\text{sinc}^2\left(\frac{\omega_{IE} t}{2\pi}\right)} \tag{2.22}$$

Where $\mathcal{O}$ denotes the null vector, hence expressing that the formula considers a

stationary case.

Proof: From the formula for β_v assuming $v^N = \mathcal{O}$:

$$\beta_v|_{v^N=\mathcal{O}} = C_{N(t)}^{N(0)}(\mathcal{O}) - (\mathcal{O}) + \int_0^t C_{N(\tau)}^{N(0)} \left[\left(\omega_{IE}^N\right)^\times (\mathcal{O}) - g^N \right] d\tau \tag{2.23}$$

$$\beta_v|_{v^N=\mathcal{O}} = -\int_0^t C_{N(\tau)}^{N(0)} \left[g^N \right] d\tau \tag{2.24}$$

Further assuming g^N given by Equation (2.5) and:

$$C_{N(t)}^{N(0)} = C_{E(0)}^{N(0)} C_{E(t)}^{E(0)} C_{N(t)}^{E(t)} \tag{2.25}$$

With $C_{N(t)}^{E(t)}$ given by Equation (A.13).

Another assumption, coherent with $v^N = \mathcal{O}$, is that $C_{N(t)}^{E(t)} = C_{N(0)}^{E(0)}$.

This allows evaluating Equation (2.24) to:

$$\beta_v|_{v^N=\mathcal{O}} = -\int_0^t C_{N(0)}^{E(0)} C_{E(t)}^{E(0)} C_{N(0)}^{E(0)} \begin{bmatrix} 0 \\ 0 \\ g_0 \end{bmatrix} d\tau \tag{2.26}$$

$$\beta_v|_{v^N=\mathcal{O}} = -C_{N(0)}^{E(0)} \left(\int_0^t C_{E(\tau)}^{E(0)} d\tau \right) \begin{bmatrix} \cos(\phi) \\ 0 \\ -\sin(\phi) \end{bmatrix} g_0 \tag{2.27}$$

Now, deriving both sides of this equation with respect to time yields:

$$\frac{d}{dt} \left[\beta_v|_{v^N=\mathcal{O}} \right] = -C_{N(0)}^{E(0)} C_{E(t)}^{E(0)} \begin{bmatrix} \cos(\phi) \\ 0 \\ -\sin(\phi) \end{bmatrix} g_0 \tag{2.28}$$

Noting as well that the squared norm of β_v can be derived with respect to time, yielding:

$$\frac{d}{dt}\left[\|\beta_v|_{v^N=\mathcal{O}}\|^2\right] = 2\beta_v|_{v^N=\mathcal{O}}{}^T \frac{d}{dt}\left[\beta_v|_{v^N=\mathcal{O}}\right] \tag{2.29}$$

Since $C_{E(t)}^{E(0)}$ is a rotation around the z axis by and angle $t\omega_{IE}$ (GROVES, 2013):

$$C_{E(t)}^{E(0)} = \begin{bmatrix} \cos(t\omega_{IE}) & -\sin(t\omega_{IE}) & 0 \\ \sin(t\omega_{IE}) & \cos(t\omega_{IE}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

Thus, the integral in Equation (2.27) becomes:

$$\begin{aligned} \int_0^t C_{E(\tau)}^{E(0)} d\tau &= \int_0^t \begin{bmatrix} \cos(\tau\omega_{IE}) & -\sin(\tau\omega_{IE}) & 0 \\ \sin(\tau\omega_{IE}) & \cos(\tau\omega_{IE}) & 0 \\ 0 & 0 & 1 \end{bmatrix} d\tau \\ &= \frac{1}{\omega_{IE}} \begin{bmatrix} \sin(t\omega_{IE}) & -1+\cos(t\omega_{IE}) & 0 \\ 1-\cos(t\omega_{IE}) & \sin(t\omega_{IE}) & 0 \\ 0 & 0 & t\omega_{IE} \end{bmatrix} \end{aligned} \tag{2.31}$$

Replacing this result in Equation (2.27):

$$\beta_v|_{v^N=\mathcal{O}} = -\frac{g_0}{\omega_{IE}} C_{N(0)}^{E(0)} \begin{bmatrix} \sin(t\omega_{IE}) & -1+\cos(t\omega_{IE}) & 0 \\ 1-\cos(t\omega_{IE}) & \sin(t\omega_{IE}) & 0 \\ 0 & 0 & t\omega_{IE} \end{bmatrix} \begin{bmatrix} \cos(\phi) \\ 0 \\ -\sin(\phi) \end{bmatrix} \tag{2.32}$$

Replacing this result in Equation (2.29), the rotation matrices times their transposed counterparts become identity matrices, thus:

$$\begin{aligned} &\frac{d}{dt}\left[\|\beta_v|_{v^N=\mathcal{O}}\|^2\right] = \\ &2\frac{go^2}{\omega_{IE}} \begin{bmatrix} \cos(\phi) \\ 0 \\ -\sin(\phi) \end{bmatrix}^T \begin{bmatrix} \sin(t\omega_{IE}) & -1+\cos(t\omega_{IE}) & 0 \\ 1-\cos(t\omega_{IE}) & \sin(t\omega_{IE}) & 0 \\ 0 & 0 & t\omega_{IE} \end{bmatrix} \begin{bmatrix} \cos(\phi) \\ 0 \\ -\sin(\phi) \end{bmatrix} \end{aligned} \tag{2.33}$$

Simplifying:

$$\frac{d}{dt}\left[\|\beta_v|_{v^N=\mathcal{O}}\|^2\right] = 2\frac{go^2}{\omega_{IE}}\left(2\cos(\phi)\sin(t\omega_{IE}) + \sin^2(\phi)t\omega_{IE}\right) \tag{2.34}$$

Then integrating this expression with respect to time:

$$\|\beta_v|_{v^N=\mathcal{O}}\|^2 = \frac{go^2}{\omega_{IE}^2}\left(t^2\omega_{IE}^2\sin^2(\phi) - 2\cos^2(\phi)(\cos(t\omega_{IE}) - 1)\right) \tag{2.35}$$

Then, taking the square root and rearranging, a better trendline is given by:

$$\|\beta_v|_{v^N=\mathcal{O}}\| = g_0 t\sqrt{\sin^2(\phi) + 4\cos^2(\phi)\left(\frac{\sin^2\left(t\omega_{IE}/2\right)}{\omega_{IE}^2 t^2}\right)} \tag{2.36}$$

Which can be simplified with the cardinal sine function:

$$\|\beta_v|_{v^N=\mathcal{O}}\| = g_0 t\sqrt{\sin^2(\phi) + \cos^2(\phi)\mathrm{sinc}\left(\frac{t\omega_{IE}}{2\pi}\right)} \tag{2.37}$$

Thus concluding the proof.

Figure 2.4 - Comparison of norms for v vectors over long term.

2.3.3 Angle Between Vectors

Remark 2.3.2. Angles Match in Both Frames:
Since the same transformation represented by $C_{B(0)}^{N(0)}$, transforms α_v and α_p into β_v and β_p respectively, the angle between these pairs of vectors should be the same.

$$\begin{aligned} -\Delta_{tol} &< \angle\left(\alpha_v, \alpha_p\right) - \angle\left(\beta_v, \beta_p\right) \\ &< \Delta_{tol} \end{aligned} \tag{2.38}$$

Difference between angles, shown in Figure 2.5 can be monitored, and a set of vectors can be declared invalid if the error (in absolute value) is above a certain threshold.

Figure 2.5 - Angle Discrepancy.

Due to noise and integration errors, these angles will have a difference that may be large, especially on the beginning of the simulation since these vectors are initialized at zero. When using the PIF or VIF methods separately, this property is not checked.

Remark 2.3.3. Minimum Angle:
The angle between such vectors may be zero, a situation that causes the TRIAD method to diverge. Even at small angles, TRIAD method's accuracy significantly decreases.

It can be verified by checking that the angles between each pair of vectors expressed in the same frame is larger than a given tolerance as follows:

$$\angle_{\alpha min} < \angle\left(\alpha_v,\, \alpha_p\right) \tag{2.39}$$

$$\angle_{\beta min} < \angle\left(\beta_v,\, \beta_p\right) \tag{2.40}$$

Such zero angle occurs in one simulation around 12s as can be seen in Figure 2.6. Unless the attitude matrix to be estimated is the identity, it is also necessary that such angles varies over time during the simulation, to provide enough information for the computation.

If the direction of vectors α_v and β_v remains constant for the whole experiment, the same will be true for the vectors α_p and β_p. This causes the TRIAD method to diverge but will also create a condition in which the SVD method has no observability of the full three-dimensional space, since the matrix K would have two degenerate singular values.

Figure 2.6 - Computed Angles between α and β vectors.

This check has not been employed for the current implementation of the presented algorithms, as the data mass caused angles to be consistently small as shown in Figure 4.21, which is expected for short trajectories.

2.3.4 Norm Discrepancy

At each simulation step, the norms of the α_v and β_v can be computed and ideally, they should be exactly the same. Nonetheless, as far as on-line estimation is concerned, it is not possible to verify that the computed norms are accurate. It is possible to compute the norms for the estimates and know that their discrepancy comes from errors in their calculations and noise from measurements. To allow

inferences to be drawn from the discrepancy, a normalization method is necessary, since smaller errors might be more meaningful if the norms are smaller. Normalization, in turn, requires a reference which ideally should be the true norm, which once again is not available.

The selected surrogate for the true norm is the average of the computed norms. In equation form:

$$\rho_v = 2\frac{\|\alpha_v\| - \|\beta_v\|}{\|\alpha_v\| + \|\beta_v\|} \tag{2.41a}$$

$$\rho_p = 2\frac{\|\alpha_p\| - \|\beta_p\|}{\|\alpha_p\| + \|\beta_p\|} \tag{2.41b}$$

And the criteria being:

$$-\rho_v^{max} < \rho_v < \rho_v^{max} \tag{2.42}$$

$$-\rho_p^{max} < \rho_p < \rho_p^{max} \tag{2.43}$$

Simulated results show that, indeed, a relatively large error appears between the vectors, shown in Figure 2.7. If the error was related only to the norm itself (and not to the direction), the TRIAD method (that normalizes vectors) would output the same result while the SVD method (that weights vectors by their norms when they're not unitary but assumes norms to be the same) would be slightly affected. Significant norm discrepancies, however, are an indication that errors are still relevant within the computed vectors.

Figure 2.7 - Discrepancy Between Norms.

2.3.5 Filtered Method (FIL)

While the TRIAD based method is mentioned as a base check, it has less memory of the computed vectors than other methods and is more susceptible to the aforementioned flaws. This is solved by other methods with the accumulation of vectors (either only v or p ones) on the K matrix as shown on Equation (2.14) and Equation (2.18). By performing the aforementioned checks, the sets of vectors used can be selected, or filtered, before being added to the matrix K, this approach constitutes a modified method that takes long before providing any output, which in turn can be considered close to convergence, unlike the previous methods (VIF, PIF and TRIAD), which provide no on-line indication to make inferences about its convergence state. This is shown in Chapter 4.

The usage of condition checking described on the previous sections will be called the "filtered" method and was originally proposed on (KOJA; LEITE FILHO, 2017). It depends heavily on user designed parameters but allows better on-line assessment of convergence. Its solution will be denoted with a FIL superscript and is also determined with the same vectors but does not use all instances of those vectors, thus not being updated on every cycle.

$$K_k^{FIL} = \begin{cases} K_{k-1}^{FIL}, & \text{if conditions not met} \\ \beta_p(t_k)\left(\alpha_p(t_k)\right)^T + \beta_v(t_k)\left(\alpha_v(t_k)\right)^T + K_{k-1}^{FIL}, & \text{if conditions met} \end{cases} \tag{2.44}$$

A discussion of the results obtained with this method will be done in Chapter 4.

Two more methods are explored in this work, but are based on developments to be presented on the next chapter, hence will be explained later on Section 3.8.

3 COVARIANCE MODELING

In this chapter, covariance models for the α_v and β_v vectors are developed. Since these models are intended to be part of an on-board algorithm, a concept of "Recursive Formula" is introduced with a distinct definition in this work from what would normally be found on standard mathematical textbooks. Some consideration is later given to how to present covariance matrices and which insight they can provide into computation errors of those vectors. Many tools are developed to compute covariance with recursive formulas and applied to the current problem. This chapter is mostly constituted by original contributions of this work.

3.1 Recursive Formula

Definition 3.1.1. Recursive Formula:
For the purpose of this work, a recursive formula is defined on the following basis. Given:

- An input sequence u_n
- An output sequence a_n
- A set of initial values $a_0, a_1, \cdots, a_k$

If the sequence a_n may be characterized with a formation law such as:

$$a_n = f(n, u_n, a_{n-1}, a_{n-2} \cdots, a_k \cdots, a_0) \tag{3.1}$$

Then a recursive formula is defined as a system of equations:

$$p_0 = I(a_k, a_{k-1}, \cdots, a_0) \tag{3.2a}$$

$$p_n = G(p_{n-1}, a_{n-1}, u_n) \tag{3.2b}$$

$$a_n = H(p_n, a_{n-1}) \tag{3.2c}$$

Where the functions G and H can be computed in polynomial time and do not depend on the index n. $p_n \in \mathbb{R}^m$ is called an internal state with fixed dimension.

This means that the recursive scheme may need knowledge of the initialization of the sequence a_n, but at each step no additional memory is needed, and no operation on the whole past sequence (neither a_n nor u_n) needs to be done to compute the next value a_{n+1}. The advantage of such formulation is that it allows implementation for real-time applications to run for an indefinitely long span of time, provided it can run the first step.

As an example, let u_n be generic and:

$$a_n = \sum_{k=1}^{n} (nu_n) \tag{3.3}$$

The operation to determine a_n would require knowledge of the whole history of u_n to be computed, thus it is not recursive. If the sum is rearranged as:

$$a_n = nu_n + a_{n-1} \tag{3.4}$$

The operation still requires knowledge of the index n to be computed. Again, it is not recursive. Rearranging again as:

$$p_0 = 0; \tag{3.5}$$

$$p_n = p_{n-1} + 1 \tag{3.6}$$

$$a_n = p_n u_n + a_{n-1} \tag{3.7}$$

This creates a recursive formula for the same sequence as before.

It is not always possible nor simple to find one such formula. For instance, let $u_n \in \mathbb{N}$:

$$a_n = \sum_{n-u_k}^{n} f(u_k) \tag{3.8}$$

No recursive formula is known for such situation, since the sum has variable size and the state p_n is required to have fixed size.

For summations of that admit the structure shown in Equation (3.9) it is straightforward to create a recursive formula:

$$a_n = \sum_{k=0}^{n} f(u_k) \tag{3.9}$$

This can be rewritten as:

$$p_0 = 0 \tag{3.10a}$$

$$p_n = a_{n-1} \tag{3.10b}$$

$$a_n = f(u_k) + p_n \tag{3.10c}$$

This will be used on Section 3.6

The general block diagram for a recursive formula is presented below.

Figure 3.1 - General block diagram for implementing a recursive formula.

3.2 Covariance of a Cross Product

It will be seen in Section 3.7 that the variance of the cross product between two vectors will be used several times. While it is not theoretically hard to compute, it is burdensome, and the developments made there will require a more compact notation along with knowledge of some properties, which are introduced in this

section.

Proposition 3.2.1. Covariance of Cross Product Operator:
Given two zero mean independent random vectors μ and ε with their respective variances σ_μ^2 and σ_ε^2. The variance of their cross-product can be computed by:

$$Var\left[\mu^\times\varepsilon\right]_{11} = \sigma_{\mu_{33}}^2\sigma_{\varepsilon_{22}}^2 - 2\sigma_{\mu_{23}}^2\sigma_{\varepsilon_{23}}^2 + \sigma_{\mu_{22}}^2\sigma_{\varepsilon_{33}}^2 \tag{3.11a}$$

$$Var\left[\mu^\times\varepsilon\right]_{12} = Var\left[\mu^\times\varepsilon\right]_{21} = -\sigma_{\mu_{33}}^2\sigma_{\varepsilon_{12}}^2 + \sigma_{\mu_{23}}^2\sigma_{\varepsilon_{13}}^2 + \sigma_{\mu_{13}}^2\sigma_{\varepsilon_{23}}^2 - \sigma_{\mu_{12}}^2\sigma_{\varepsilon_{33}}^2 \tag{3.11b}$$

$$Var\left[\mu^\times\varepsilon\right]_{13} = Var\left[\mu^\times\varepsilon\right]_{31} = \sigma_{\mu_{23}}^2\sigma_{\varepsilon_{12}}^2 - \sigma_{\mu_{22}}^2\sigma_{\varepsilon_{13}}^2 - \sigma_{\mu_{13}}^2\sigma_{\varepsilon_{22}}^2 + \sigma_{\mu_{12}}^2\sigma_{\varepsilon_{23}}^2 \tag{3.11c}$$

$$Var\left[\mu^\times\varepsilon\right]_{22} = \sigma_{\mu_{33}}^2\sigma_{\varepsilon_{11}}^2 - 2\sigma_{\mu_{13}}^2\sigma_{\varepsilon_{13}}^2 + \sigma_{\mu_{11}}^2\sigma_{\varepsilon_{33}}^2 \tag{3.11d}$$

$$Var\left[\mu^\times\varepsilon\right]_{23} = Var\left[\mu^\times\varepsilon\right]_{32} = -\sigma_{\mu_{23}}^2\sigma_{\varepsilon_{11}}^2 + \sigma_{\mu_{13}}^2\sigma_{\varepsilon_{12}}^2 + \sigma_{\mu_{12}}^2\sigma_{\varepsilon_{13}}^2 - \sigma_{\mu_{11}}^2\sigma_{\varepsilon_{23}}^2 \tag{3.11e}$$

$$Var\left[\mu^\times\varepsilon\right]_{33} = \sigma_{\mu_{22}}^2\sigma_{\varepsilon_{11}}^2 - 2\sigma_{\mu_{12}}^2\sigma_{\varepsilon_{12}}^2 + \sigma_{\mu_{11}}^2\sigma_{\varepsilon_{22}}^2 \tag{3.11f}$$

Noting that the value depends only on the respective covariance matrices, this result will generally be denoted by the $\boxtimes$ operator:

$$Var\left[\mu^\times\varepsilon\right] = \left(\sigma_\mu^2 \boxtimes \sigma_\varepsilon^2\right) \tag{3.12}$$

Sketch Proof:

$$Var\left[\mu^\times\varepsilon\right] = Var\left[\begin{bmatrix} 0 & -\mu_z & \mu_y \\ \mu_z & 0 & -\mu_x \\ -\mu_y & \mu_x & 0 \end{bmatrix}\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix}\right] \tag{3.13}$$

$$Var\left[\mu^\times\varepsilon\right] = Var\left[\begin{bmatrix} \mu_y\varepsilon_z - \mu_z\varepsilon_y \\ \mu_z\varepsilon_x - \mu_x\varepsilon_z \\ \mu_x\varepsilon_y - \mu_y\varepsilon_x \end{bmatrix}\right] \tag{3.14}$$

Using an identity relating variance with expectancy of a random variable (Theorem B.1.8):

$$
\begin{aligned}
Var\left[\mu^{\times}\varepsilon\right] &= \\
&= E\left[\begin{bmatrix} \mu_y\varepsilon_z - \mu_z\varepsilon_y \\ \mu_z\varepsilon_x - \mu_x\varepsilon_z \\ \mu_x\varepsilon_y - \mu_y\varepsilon_x \end{bmatrix}\begin{bmatrix} \mu_y\varepsilon_z - \mu_z\varepsilon_y \\ \mu_z\varepsilon_x - \mu_x\varepsilon_z \\ \mu_x\varepsilon_y - \mu_y\varepsilon_x \end{bmatrix}^T\right] \\
&\quad \underbrace{- E\left[\begin{bmatrix} \mu_y\varepsilon_z - \mu_z\varepsilon_y \\ \mu_z\varepsilon_x - \mu_x\varepsilon_z \\ \mu_x\varepsilon_y - \mu_y\varepsilon_x \end{bmatrix}\right]\left(E\left[\begin{bmatrix} \mu_y\varepsilon_z - \mu_z\varepsilon_y \\ \mu_z\varepsilon_x - \mu_x\varepsilon_z \\ \mu_x\varepsilon_y - \mu_y\varepsilon_x \end{bmatrix}\right]\right)^T}_{0}
\end{aligned}
\tag{3.15}
$$

The rest of the proof is omitted, but it consists purely on applying the linearity of the expectancy operator (Theorem B.1.1) and isolating terms involving each variable, then identifying their expectancy with the definition of the covariance. A few properties of this operator that are trivial to verify procedurally are presented next.

Proposition 3.2.2. $\boxtimes$ is Bilinear:
For deterministic scalars a and b:

$$
Var\left[a\mu^{\times}\ (b\varepsilon)\right] = a^2b^2(\sigma_\mu^2 \boxtimes \sigma_\varepsilon^2) \tag{3.16}
$$

Proposition 3.2.3. $\boxtimes$ is Commutative:

$$
\left(\sigma_\mu^2 \boxtimes \sigma_\varepsilon^2\right) = (\sigma_\varepsilon^2 \boxtimes \sigma_\mu^2) \tag{3.17}
$$

Proposition 3.2.4. Pre-multiplication of second vector:
For any deterministic matrix C:

$$
Var\left[\mu^{\times}\ (C\varepsilon)\right] = \left(\sigma_\mu^2 \boxtimes \left(C\sigma_\varepsilon^2 C^T\right)\right) \tag{3.18}
$$

Proposition 3.2.5. Pre-multiplication on $\boxtimes$:
For any matrix $D \in \mathcal{M}(\mathbb{R})_{3\times 3}$:

$$
Var\left[D\mu^{\times}\varepsilon\right] = D(\sigma_\mu^2 \boxtimes \sigma_\varepsilon^2)D^T \tag{3.19}
$$

These results are also implicitly used and checked in the upcoming simulations, such as the ones verifying Equation (3.52).

3.3 General Error Equations

This section explains some considerations that are used on Chapter 2, which are based on the previous concepts of this chapter.

3.3.1 Basic Errors Definitions

Starting with two definitions that are particular to this work:
Definition 3.3.1. Noise Type Error:
A signal ε_k is called a Noise Type Error if it is an Independent Identically distributed random sequence, with constant zero mean.

This implies that noise type errors are uncorrelated both over different experiments and at distinct times of the same experiment.
Definition 3.3.2. Bias Type Error:
A random variable η is called a Bias Type Error if it is a constant over any given experiment, but over different experiments, it is identically distributed with zero expectancy.

This implies that knowing its value at a given instant of an experiment is the same as knowing its value throughout the whole experiment. However, even if the distribution of a Bias Type Error is known, the knowledge of its value during one experiment offers no new information about its possible values for a different experiment.

Notably no assumptions are made about the distribution of those errors other than their mean and covariance. In all simulations presented in Section 3.6 and in Chapter 4, these errors are modeled with random number generators that produce normally distributed values. However, all developments in Section 3.6 and Section 3.7 make no such assumption.

In practical terms for simulations, Bias Type errors are picked at initialization and kept constant for the whole simulation, while Noise Type errors must be generated at every (relevant) cycle. These errors are the building blocks for the models used as inputs to the algorithms that are to be developed. It is noted that in general, sensor manufacturers should aim for creating products whose measurements have errors of such type, for two main reasons:

- Bias Type Errors can be removed by calibration.

- Noise Type Errors can be attenuated by filtering.

Both these tasks can be performed by Kalman Filtering techniques, as done for instance in (KOJA et al., 2013). It is well understood that some sensors may have slowly-varying errors, which are strongly correlated between time-steps, but lose or diminish their correlation over long time periods, such as the ionospheric delay in pseudorange measurements in GPS receivers (FARRELL, 2008). It is noted that both types of errors are part of the models for sensors described on Section A.3, where further considerations are given.

3.3.2 Handling Operations Involving Errors

Assume that a numerical computation of the integral of a generic deterministic function $f(t)$ is to be approximated by some weight factor times the integrand at discrete steps such that:

$$\int_0^t f(\tau)\, d\tau = \sum_k w_k f_k(t_k) \tag{3.20}$$

Noting that w_k in this case is the integration time step but could be a different weighting sequence depending on the numeric integration method. This is further discussed in Section A.4.

Assume now the integrand involves a Noise Type Error ε as per Definition 3.3.1 that is additive to some other generic deterministic function $g(t)$, this means:

$$\widehat{g}(t) = g(t) + \varepsilon(t) \tag{3.21}$$

$$\widehat{g}_k = g_k + \varepsilon_k \tag{3.22}$$

Now the function to be integrated is $f(t)g(t)$ while only the value of $f(t_k)\widehat{g}_k$ is available.

Then the product between f and g under an integral becomes:

$$\begin{aligned}\int_0^t f(\tau)\widehat{g}(\tau)\, d\tau &= \int_0^t f(\tau)\left(g(\tau) + \varepsilon(\tau)\right) d\tau \\ &= \int_0^t f(\tau)g(\tau)\, d\tau + \int_0^t f(\tau)\varepsilon(\tau) d\tau\end{aligned} \tag{3.23}$$

And discretizing:

$$\begin{aligned}\int_0^t f(\tau)\widehat{g}(\tau)\,d\tau &\approx \sum_k w_k f_k \widehat{g}_k \\ &= \sum_k w_k f_k g_k + \sum_k w_k f_k \varepsilon_k\end{aligned} \tag{3.24}$$

Noting that ε is a random variable, it is desired to assess the expected value of the computation being performed:

$$E\left[\sum_k w_k f_k \widehat{g}_k\right] = E\left[\sum_k w_k f_k g_k + \sum_k w_k f_k \varepsilon_k\right] \tag{3.25}$$

From the linearity of the expectation operator (Theorem B.1.1) :

$$E\left[\sum_k w_k f_k \widehat{g}_k\right] = E\left[\sum_k w_k f_k g_k\right] + E\left[\sum_k w_k f_k \varepsilon_k\right] \tag{3.26}$$

Noting that the term on the first expectation is not a random variable:

$$E\left[\sum_k w_k f_k g_k\right] = \sum_k w_k f_k g_k \tag{3.27}$$

Thus:

$$E\left[\sum_k w_k f_k \widehat{g}_k\right] = \sum_k w_k f_k g_k + E\left[\sum_k w_k f_k \varepsilon_k\right] \tag{3.28}$$

Again applying linearity property:

$$E\left[\sum_k w_k f_k \widehat{g}_k\right] = \sum_k w_k f_k g_k + \sum_k w_k f_k E\left[\varepsilon_k\right] \tag{3.29}$$

And from Definition 3.3.1, which requires $E\left[\varepsilon_k\right] = 0\,(\forall k)$, therefore:

$$E\left[\sum_k w_k f_k \widehat{g}_k\right] = \sum_k w_k f_k g_k \tag{3.30}$$

This implies that upon this kind of operation over a Noise Type Error, the expected error of the result is zero, which is a convenient result. This however might seem a bit suspicious as the summing of errors should cause the error magnitude to build up. However, this perception speaks about variance, rather than about expectancy of this particular expression.

Proceeding similarly for the variance:

$$\begin{aligned} Var\left[\sum_k w_k f_k \widehat{g}_k\right] = \\ E\Bigg[\Bigg(\sum_k w_k f_k \widehat{g}_k \\ - E\left[\sum_k w_k f_k \widehat{g}_k\right]\Bigg)\left(\sum_k w_k f_k \widehat{g}_k - E\left[\sum_k w_k f_k \widehat{g}_k\right]\right)^T\Bigg] \end{aligned} \tag{3.31}$$

From the above deduction:

$$\begin{aligned} Var\left[\sum_k w_k f_k \widehat{g}_k\right] = E\Bigg[\Bigg(\sum_k w_k f_k g_k + \sum_k w_k f_k \varepsilon_k \\ - \sum_k w_k f_k g_k\Bigg)\left(\sum_k w_k f_k g_k + \sum_k w_k f_k \varepsilon_k - \sum_k w_k f_k g_k\right)^T\Bigg] \end{aligned} \tag{3.32}$$

$$Var\left[\sum_k w_k f_k \widehat{g}_k\right] = E\left[\left(\sum_k w_k f_k \varepsilon_k\right)\left(\sum_k w_k f_k \varepsilon_k\right)^T\right] \tag{3.33}$$

It is notable that the actual values of g hold no influence over the covariance of the error, yet naturally the values of f_k do. Simplifying the previous equation:

$$Var\left[\sum_k w_k f_k \widehat{g}_k\right] = Var\left[\sum_k w_k f_k \varepsilon_k\right] \tag{3.34}$$

That is, for the variance, only the non-deterministic part matters.

The continuation of the deduction of the variance for this situation is provided on Section 3.6.2, where the weighting coefficient has been incorporated into the function f_k for conciseness of notation. But generally, it does indeed increase over time depending on the values of f_k. Again, it is noted that on this section, and not through the rest of this work, f ad g are generic time-dependent functions.

3.4 Working with Covariance Matrices

3.4.1 Estimating Covariance Matrices

Let, X_n be a k-dimensional stochastic process that is to be realized M times, and x_n^i be the i-th realization of such process. At each time-index n, an estimate $\widehat{K_n}$ for the variance K_n of such process can be computed with two steps (STARK; WOODS, 2002), first computing the sample mean with:

$$\widehat{\mu_n} = \sum_{i=1}^{M} \frac{x_n^i}{M} \tag{3.35}$$

Then the sample variance is computed with:

$$\widehat{K_n} = \sum_{i=1}^{M} \frac{\left(x_n^i - \widehat{\mu_n}\right)\left(x_n^i - \widehat{\mu_n}\right)^T}{M-1} \tag{3.36}$$

Notice that in Equation (3.36), an empirical estimation of a covariance is found. It required no prior knowledge of the structure or characteristics of the process, but realizations were needed for the covariance estimation.

3.4.2 Illustrating Covariance Matrices

Let X_n be a 3-dimensional random process with K_n its covariance matrix. Then K_n can be plotted in a graph with the index n on the x-axis and the values of one component of K_n on the y axis. For convenience, it is possible to display all 9 entries in a 3x3 matrix of graphs.

An example of such graph is given in Figure 3.2, where a process described by Equation (3.46) (presented later on Section 3.6.2) was simulated 50 times and its covariance computed with Equation (3.36).

It is noted that the diagonal terms are always positive as a consequence of Theorem B.1.6.

Figure 3.2 - Example of Estimated Covariance Matrix of a Stochastic Process

3.4.3 Creating Samples for Given Covariance

Most computing languages provide functions for generating either uniformly distributed independent random variables in the interval $[0, 1]$ or gaussian random variables with zero mean and unit standard deviation. Most other kinds of random variables can be obtained operating over the outputs of those functions. For instance, let σ_ε^2 be a variance matrix, it admits a Cholesky decomposition (LIMA, 1995) of the form:

$$\sigma_\varepsilon^2 = LL^T \tag{3.37}$$

With L a lower-triangular matrix. Then let G be a random vector to be generated by a function that outputs realizations of an independent gaussian random variable with zero mean and unit standard deviation. The variance of G will be:

$$\begin{aligned} Var[G] &= E[GG^T] \\ &= I \end{aligned} \tag{3.38}$$

However, the variance of LG is given by:

$$\begin{aligned} Var[LG] &= E[LG(LG)^T] \\ &= E[LGG^TL^T] \\ &= LE[GG^T]L^T \\ &= LIL^T \\ &= LIL^T \\ &= \sigma_\varepsilon^2 \end{aligned} \tag{3.39}$$

Therefore, LG has the desired variance of σ_ε^2.

3.5 Working with Distributions

This section comments on practical matters related to the estimation and presentation of distribution functions. Focus is given on probability density functions, but considerations would be analogous for cumulative distribution functions. These consideration help understanding the graphs presented both later in this chapter and on Chapter 4.

3.5.1 Illustrating Distributions

This section explains how to interpret some of the graphs shown later in this chapter and in Chapter 4. In a single dimension, the distribution of a random variable X can be plotted against the possible values x it may assume. For several dimensions, one graph for each dimension could be plotted. For a stochastic process, however, the additional dimension of time would need to be added, which would require a 3D graph for each dimension, with one axis for possible values, one axis for the time and another for the actual density function value. This is achieved in a simpler manner by a color map plot. Take as an example a one-dimensional sequence with the following distribution:

$$f_X(x, n) = \frac{e^{-\frac{x^2}{n}}}{\sqrt{n\pi}} \tag{3.40}$$

For this density function, the standard deviation $\sigma_X = \sqrt{n/2}$. It is possible to plot for individual values of the index n, the probability density as a function of x, as shown in Figure 3.3. A richer representation that better covers the variety of time indexes is the usage of a color map, as shown in Figure 3.4. In this plot, the

magnitude of the probability density is expressed in a color scale, shown left to the graph region. The x-axis is the time-index, which replaces the need for the multiple curves used in Figure 3.3. The possible values for the random variable are shown in the y-axis. Hence each point (x, y, z) in the $3D$ plot is a triple: Time-index, possible value and associated probability density.

Depending on the case it may add better contrast to the 3D figure if the color mapping is in logarithmic scale, which is applied in Figure 3.5. Lines showing the standard deviation multiplied by a few factors, namely $\{1, -1, 3, -3\}$, this shows how the standard deviation envelopes the regions where density is higher when the process has constant zero mean. These lines could be summed to the mean value for non-zero mean processes as will be done in Chapter 4.

Figure 3.3 - Example of Graph for Probability Density.

Figure 3.4 - Colormap plot for Probability Density.

Figure 3.5 - Colormap plot for natural logarithm of Probability Density.

3.5.2 Kernel Density Estimation

Estimating the distribution of a random process or variable from samples is a complex process with many issues of practical performance. Within the scope of this work, it will be done only for illustrative purposes, to create the probability density plots shown in Section 3.6, Section 4.2.4.3 and Section 4.2.4.5. Hence the details of the methods are not to be explored, nor their fine accuracy should be relevant.

A convenient although computationally expensive method for density estimation is the Kernel Density Estimation method (BOWMAN; AZZALINI, 1997). Given n realized samples x_k and a Kernel function $\kappa : \mathbb{R} \to \mathbb{R}^+$, and estimation $\widehat{f_X}$ of the probability density function f_X can be computed with:

$$\widehat{f_X}(x) = \frac{1}{nh} \sum_{k=1}^{n} \kappa \left(\frac{x - x_k}{h} \right) \tag{3.41}$$

The parameter h is called the bandwidth of the smoothing window, and there

are rules for picking optimal h values depending on the characteristics of the distribution to be estimated (BOWMAN; AZZALINI, 1997), a plot of these bandwidths can be jointly presented. The optimal bandwidth for estimating a normal density function is given by Equation (3.42):

$$h = \left(\frac{4}{3n}\right)^{\frac{1}{5}} \tilde{\sigma} \tag{3.42}$$

Where $\tilde{\sigma}$ is an estimator for the standard deviation, computed with Equation (3.43)

$$\tilde{\sigma} = \text{median}\left\{|x_k - \overline{x}|\right\} \frac{1}{0.6745} \tag{3.43}$$

In Equation (3.43), $\overline{x}$ is the sample mean.

For the purpose of this work, a normal kernel is used, i.e.:

$$\kappa(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{3.44}$$

This technique is used herein for estimating the distributions of random sequences X_n and presenting them in graphs. For plotting purposes, a discrete set of points x_j is chosen based on the sample to be included in the vertical scale, such that the matrix of points (n, x_j) can be color mapped to $\widehat{f_{X_n}}(x_j)$, which requires a square matrix of points and their associated color scale values as shown in Section 3.5.1. For better aesthetic results with the color plot, a normalization over this discrete set of points is performed as:

$$\widehat{f^*_{X_n}}(x_j) = \frac{\widehat{f_{X_n}}(x_j)}{\sum_p \widehat{f_{X_n}}(x_p)} \tag{3.45}$$

This normalization will specially distort sharp distributions if the amount of plot points x_j around the peaks is low, so the values obtained are not to be assumed exact, nor are precisely represented in a color plot. It is again recalled that these techniques serve mostly a visual purpose, and do not replace the simpler approach

based on matrix charts presented in Section 3.4.2. The advantage of proceeding with this normalization steams from the fact that most density plots begin with a peak near zero for the first index and show growing covariance towards the last, hence normalizing values at each index improves the color scale visibility throughout the whole plot.

3.6 Covariance Propagation

This section contains part of the present work's contributions. Computing the covariances of some particular cases of stochastic series allows constructing tools to infer the expected accuracy of estimations. For each particular case, a recursive formula is found (recalling Definition 3.1.1), such that it can be implemented in a block diagram with only a few memory or delay blocks, which effectively means low memory usage and no need to operate on the whole data history on each cycle. It is noted that the current approach differs meaningfully from the covariance propagation methods described on Section A.5, as will be seen when these formulas are employed on Section 3.7.

3.6.1 Notation

Establishing the notation to be used throughout this section:

$f_k \in \mathcal{M}(\mathbb{R})_{3\times 3}$ is a function with known value for each k.

$h_k \in \mathcal{M}(\mathbb{R})_{3\times 3}$ is a function with known value for each k.

μ is a Bias Type Error as per Definition 3.3.2 with covariance σ_μ^2

ε_k is a Noise Type Error as per Definition 3.3.1 with 3x3 covariance matrix σ_ε^2

ξ_m is a Noise Type Error as per Definition 3.3.1 with 3x3 covariance matrix σ_ξ^2

For each case, the propagation scheme is presented and its results against simulated data are shown. f_k and h_k were generated randomly with zero mean and kept the same values for each time index k between simulations. In the plots with the estimated density, the predicted values for the standard deviation, which are computed with the recursive formulas, are shown for comparison.

3.6.2 Case 1: Sum of Known Gains Times Noise

Proposition 3.6.1. Recursive Computation of the Covariance of a Sum of Known Gains Times Noise:
This case is described by the equation:

$$Var\left[\sum_{k=1}^{k} f_k \varepsilon_k\right] = \sum_{k=1}^{k} f_k \sigma_\varepsilon^2 f_k^T \tag{3.46}$$

Figure 3.6 - Sample Estimation versus predicted variance for case 1 sequence. Blue for predicted, black for sample estimated.

Figure 3.7 - Color map plot of sample estimated probability density versus predicted standard deviation for case 1 sequence.

Figure 3.8 - Bandwidths used for Kernel estimation of case 1 sequence.

3.6.3 Case 2: Sum of Known Gains Times Bias Cross Product Matrix Times Case 1

Proposition 3.6.2. Sum of Known Gains Times Bias Cross Product Matrix Times Case 1:
This case is described by the equation:

$$Var\left[\sum_{k=1}^{m} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = H_m\left(G_m\right) H_m^T - 2\left(J_m {H_m}^T\right)_{\oplus} + K_m \tag{3.47}$$

With:

$$G_m = \sum_{k=1}^{m} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) \tag{3.48}$$

$$J_m = \sum_{k=1}^{m} H_{k-1} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) \tag{3.49}$$

$$K_m = \sum_{k=1}^{m} \left(H_{k-1} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T \right) \right) {H_{k-1}}^T \right) \tag{3.50}$$

$$H_k = \sum_{i=1}^{k} h_i \tag{3.51}$$

Where the auxiliary sequences are initialized at zero, with $H_0 = 0$.

Figure 3.9 - Sample Estimation versus predicted variance for case 2 sequence. Blue for predicted, black for sample estimated.

Figure 3.10 - Color map plot of sample estimated probability density versus predicted standard deviation for case 2 sequence.

Figure 3.11 - Bandwidths used for Kernel estimation of case 2 sequence.

It is recalled that the operator $\boxtimes$ was described in Proposition 3.2.1 and Equation (3.12), and the symbol $\oplus$ denotes the matrix symmetric part (Definition C.1.4).

3.6.4 Case 3: Sum of Known Gains Times Noise Cross Product Matrix Times Case 1

Proposition 3.6.3. Sum of Known Gains Times Noise Cross Product Matrix Times Case 1:
This case is described by the equation:

$$Var\left[\sum_{m=1}^{n} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = h_n\left(L_n\right) h_n^T + Var\left[\sum_{m=1}^{n-1} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] \tag{3.52}$$

With:

$$L_n = \sum_{k=1}^{n}\left(\sigma_\xi^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) \tag{3.53}$$

Figure 3.12 - Sample Estimation versus predicted variance for case 3 sequence. Blue for predicted, black for sample estimated.

Figure 3.13 - Color map plot of sample estimated probability density versus predicted standard deviation for case 3 sequence.

Figure 3.14 - Bandwidths used for Kernel estimation of case 3 sequence.

3.6.5 Case 4: Sum of Known Gains Times Case 1

Proposition 3.6.4. Sum of Known Gains Times Case 1:
This case is described by the equation:

$$\begin{aligned} &Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\ &= h_{m+1} W_{m+1} h_{m+1}^T + 2\left(h_{m+1}\left(W_m H_m^T - X_m\right)\right)_{\oplus} \end{aligned} \tag{3.54}$$

With:

$$H_k = \sum_{i=1}^{k} h_i \tag{3.55a}$$

$$Q_k = f_i \sigma_\varepsilon^2 f_i^T \tag{3.55b}$$

$$W_k = Q_k + W_{k-1} \tag{3.55c}$$

$$= \left(\sum_{i=1}^{k} Q_i \right)$$

$$X_k = Q_k H_{k-1}^T + X_{k-1} \tag{3.55d}$$

$$= \sum_{k=1}^{m} Q_k H_{k-1}^T$$

All such sequences are initialized with zero, thus $H_0 = 0$, $W_0 = 0$ and $X_0 = 0$.

Figure 3.15 - Sample Estimation versus predicted variance for case 4 sequence. Blue for predicted, black for sample estimated.

Figure 3.16 - Color map plot of sample estimated probability density versus predicted standard deviation for case 4 sequence.

Figure 3.17 - Bandwidths used for Kernel estimation of case 4 sequence.

3.7 Error Analysis of the Methods

This section describes some approaches employed to assess the accuracy of the discussed methods. It develops error models for the α_v and β_v vectors, then attempts to relate such quantities with the expected accuracy of the estimated attitude. Test are conducted to validate these models, and presented in Section 4.2.4.

Usually, measurements are modeled as random variables, composed of a deterministic quantity being measured modified by another random variable associated with the uncertainty of the measurement. However, for the purpose of this section, measurements are considered as deterministic values (they are actually known), while the correct quantity being measured, and the associated noise are both considered to be random variables. From a purely mathematical point of view, the choice between these approaches is arbitrary. Within this concept, the variances of the errors $\delta\alpha_v = \widehat{\alpha_v} - \alpha_v$ and $\delta\beta_v = \widehat{\beta_v} - \beta_v$ can be computed based on the statistics of the sensor models and the measured values, by removing the true unknown values from the expressions for these vectors.

To illustrate this idea, usually if a measurement $\widehat{a}$ is performed over a real value a, with additive noise X:

$$\widehat{a} = a + X \tag{3.56}$$

Where the idea of X being random is transmitted to the measurement $\widehat{a}$. However, in this section, the consideration made is that $\widehat{a}$ is a deterministic value, and alike in the previous equation:

$$a = \widehat{a} - X \tag{3.57}$$

This expresses that in this perspective, the real value a is a seen as random variable.

3.7.1 Alpha Error

In this section, the methods from Chapter 3 are employed to estimate a variance for the error in vector α_v, namely $\delta\alpha_v$.

Recalling the formula for α_v:

$$\alpha_v = \int_0^t C_{B(\tau)}^{B(0)} f^B d\tau \tag{3.58}$$

Notably, $C_{B(t)}^{B(0)}$ is another integral gain:

$$C_{B(t)}^{B(0)} = \int_0^t C_{B(\tau)}^{B(0)} \left(\omega_{IB}^B\right)^\times d\tau \tag{3.59}$$

Using an approach similar to what (TITTERTON; WESTON, 2004) applies to model the structure of the error in attitude propagation, let's assume that an error of the following form is admissible:

$$\widehat{C_{B(t)}^{B(0)}} = C_{B(t)}^{B(0)} \left(I + \delta\Omega^\times\right) \tag{3.60}$$

Rearranging:

$$C_{B(t)}^{B(0)} = \widehat{C_{B(t)}^{B(0)}} \left(I - \delta\Omega^\times\right) \tag{3.61}$$

Note that this is a small angle approximation, which can be found with Equation (C.49) by approximating sine functions to their arguments and cosines to one, while neglecting second order terms.

Assuming that the errors derives from the integration of the gyro measurement error described in Section A.3.2, the following structure is proposed for the dynamics of $\delta\Omega$:

$$\frac{d}{dt}\left[\delta\Omega\right] = \mu + \varepsilon\left(\omega_{IB}^{B}\right) \tag{3.62}$$

Recalling that μ is a Bias Type Error (Definition 3.3.2) and $\varepsilon\left(\omega_{IB}^{B}\right)$ is the Noise Type Error (Definition 3.3.1) associated with the measurement of ω_{IB}^{B}. Notably, this assumption is an approximation that models the magnitude of errors, but the actual error in computing the rotation of the body is not directly expressed by this equation, similar to the assumption from Equation (3.60), used by (TITTERTON; WESTON, 2004).

Alternatively, integrating:

$$\delta\Omega = t\mu + \int_0^t \varepsilon\left(\omega_{IB}^{B}\right) d\tau \tag{3.63}$$

Which discretizes into:

$$\begin{aligned}\delta\Omega &= (ndt)\,\mu + \sum_{k=1}^{n} w_k \varepsilon_k\left(\omega_{IB}^{B}\right) \\ &= \sum_{k=1}^{n} w_k\left(\mu + \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\end{aligned} \tag{3.64}$$

Where w_k is a weighting coefficient as described in Section A.4, which equals the integration step dt for the purpose of this section.

Since $\varepsilon_k\left(\omega_{IB}^{B}\right)$ is assumed uncorrelated from the bias:

$$Var\left[\varepsilon_k\left(\omega_{IB}^{B}\right)\right] = \sigma_{\omega}^{2} \tag{3.65}$$

Therefore, noting that this is a particular case of Equation (3.46):

$$
\begin{aligned}
Var\left[\int_0^t \varepsilon\left(\omega_{IB}^B\right) d\tau\right] &= Var\left[\sum_{k=1}^n w_k \varepsilon_k\left(\omega_{IB}^B\right)\right] \\
&= \left(\sum_{k=1}^n w_k^2\right)\sigma_\omega^2
\end{aligned}
\tag{3.66}
$$

And a variance can be roughly estimated for $\delta\Omega$:

$$
Var\left[\delta\Omega\right] = t^2\sigma_\mu^2 + t(dt)\sigma_\omega^2 \tag{3.67}
$$

Note that this estimation is merely for qualitative assessments, since a more complex error model is to be derived later. It does however show that the uncertainty (in the sense of standard deviation) in angle alone tends to grow linearly with time, since its variance grows with the square of time, but starts at zero. It will be shown that the error in α_v inherits some of those characteristics.

The model for the accelerometer is presented in detail on Section A.3.1, but it is expressed by:

$$
f^B = \widehat{f^B} - \eta - \varepsilon\left(f^B\right) \tag{3.68}
$$

With η a Bias Type Error and $\varepsilon\left(f^B\right)$ a Noise Type Error.

Equation (3.68) discretizes into:

$$
f_k^B = \widehat{f_k^B} - \eta - \ \varepsilon_k\left(f^B\right) \tag{3.69}
$$

Recalling the formula for α_v

$$
\alpha_v = \int_0^t C_{b(\tau)}^{b(0)} f^B d\tau \tag{3.70}
$$

Define $\delta\alpha_v$ as the estimation error of α_v:

$$\delta\alpha_v = \widehat{\alpha_v} - \alpha_v \tag{3.71}$$

The goal is to derive the variance of $\delta\alpha_v$. Starting from its definition and the formula for α_v:

$$\widehat{\alpha_v} - \alpha_v = \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \widehat{f^B} d\tau - \int_0^t C_{b(\tau)}^{b(0)} f^B d\tau \tag{3.72}$$

Applying Equation (3.61):

$$\widehat{\alpha_v} - \alpha_v = \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \widehat{f^B} d\tau - \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \left(I - \delta\Omega^\times\right) f^B d\tau \tag{3.73}$$

Applying Equation (3.68):

$$\widehat{\alpha_v} - \alpha_v = \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \widehat{f^B} d\tau - \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \left(I - \delta\Omega^\times\right) \left[\widehat{f^B} - \eta - \varepsilon\left(f^B\right)\right] d\tau \tag{3.74}$$

Now all the remaining terms are either known values or random variables. Rearranging:

$$\begin{aligned}\widehat{\alpha_v} - \alpha_v = &\int_0^t \widehat{C_{b(\tau)}^{b(0)}} \widehat{f^B} d\tau - \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \left[\widehat{f^B} - \eta - \varepsilon\left(f^B\right)\right] d\tau \\ &+ \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \delta\Omega^\times \left[\widehat{f^B} - \eta - \varepsilon\left(f^B\right)\right] d\tau\end{aligned} \tag{3.75}$$

$$\begin{aligned}\widehat{\alpha_v} - \alpha_v = &\int_0^t \widehat{C_{b(\tau)}^{b(0)}} \left[\eta + \varepsilon\left(f^B\right)\right] d\tau + \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \delta\Omega^\times \widehat{f^B} d\tau \\ &- \int_0^t \widehat{C_{b(\tau)}^{b(0)}} \delta\Omega^\times \left[\eta + \varepsilon\left(f^B\right)\right] d\tau\end{aligned} \tag{3.76}$$

This completes the case without substitution of $\delta\Omega$.

Recalling Equation (3.64):

$$\delta\Omega = \left(\sum_{k=1}^{m} w_k \left(\mu + \varepsilon_k\left(\omega_{IB}^B\right)\right)\right) \tag{3.77}$$

Where the index n is switched to m because it refers to an internal integral.

For simplicity of notation, let:

$$C_k = \widehat{C^{b(0)}_{b(t_k)}} \tag{3.78}$$

And replacing the previous integrals with summations over the index m and coefficient a_m corresponding to the time step:

$$\begin{aligned}\widehat{\alpha_v} - \alpha_v = \sum_{m=1}^{n} a_m C_m \Bigg\{ \Bigg[\eta + \ \varepsilon_m \left(f^B \right) \\ + \left(\eta + \ \varepsilon_m \left(f^B \right) - \widehat{f^B_m} \right)^{\times} \left(\sum_{k=1}^{m} w_k \left(\ \mu + \varepsilon_k \left(\omega^B_{IB} \right) \right) \right) \Bigg] \Bigg\}\end{aligned} \tag{3.79}$$

$$\begin{aligned}\widehat{\alpha_v} - \alpha_v = \sum_{m=1}^{n} a_m C_m \left\{ \left[\eta \right] \right\} + \sum_{m=1}^{n} a_m C_m \left\{ \left[\ \varepsilon_m \left(f^B \right) \right] \right\} \\ + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(\eta \right)^{\times} \left(\sum_{k=1}^{m} w_k \left(\ \mu + \varepsilon_k \left(\omega^B_{IB} \right) \right) \right) \right] \right\} \\ + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(\varepsilon_m \left(f^B \right) \right)^{\times} \left(\sum_{k=1}^{m} w_k \left(\ \mu + \varepsilon_k \left(\omega^B_{IB} \right) \right) \right) \right] \right\} \\ + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f^B_m} \right)^{\times} \left(\sum_{k=1}^{m} w_k \left(\ \mu + \varepsilon_k \left(\omega^B_{IB} \right) \right) \right) \right] \right\}\end{aligned} \tag{3.80}$$

Further separating the summations:

$$
\begin{aligned}
\widehat{\alpha_v} - \alpha_v = & \sum_{m=1}^{n} a_m C_m \eta + \sum_{m=1}^{n} a_m C_m \, \varepsilon_m \left(f^B\right) + \sum_{m=1}^{n} a_m C_m \left\{ \left[(\eta)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \\
& + \sum_{m=1}^{n} a_m C_m \left\{ \left[(\eta)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right) \right) \right] \right\} \\
& + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \\
& + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right) \right) \right] \right\} \\
& + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \\
& + \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right) \right) \right] \right\}
\end{aligned}
\tag{3.81}
$$

Assigning letter names for the previous terms:

$$
A = \sum_{m=1}^{n} a_m C_m \eta \tag{3.82}
$$

$$
B = \sum_{m=1}^{n} a_m C_m \left\{ \varepsilon_m \left(f^B\right) \right\} \tag{3.83}
$$

$$
C = \sum_{m=1}^{n} a_m C_m \left\{ \left[(\eta)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \tag{3.84}
$$

$$
D = \sum_{m=1}^{n} a_m C_m \left\{ \left[(\eta)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right) \right) \right] \right\} \tag{3.85}
$$

$$
E = \sum_{m=1}^{n} a_m C_m \left\{ \left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right\} \tag{3.86}
$$

$$F = \sum_{m=1}^{n} a_m C_m \left\{ \left(\varepsilon_m \left(f^B \right) \right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B \right) \right) \right\} \tag{3.87}$$

$$G = \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f_m^B} \right)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \tag{3.88}$$

$$H = \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f_m^B} \right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B \right) \right) \right] \right\} \tag{3.89}$$

Then, recalling Theorem B.1.3:

$$\begin{aligned} Var\left[A+B+C+D+E+F+G+H\right] = {} & Var\left[A\right] + Var\left[B\right] + Var\left[C\right] + Var\left[D\right] + Var\left[E\right] \\ & + Var\left[F\right] + Var\left[G\right] + Var\left[H\right] + 2\left(Cov\left[A,B\right]\right. \\ & + Cov\left[A,C\right] + Cov\left[A,D\right] + Cov\left[A,E\right] \\ & + Cov\left[A,F\right] + Cov\left[A,G\right] + Cov\left[A,H\right] \\ & + Cov\left[B,C\right] + Cov\left[B,D\right] + Cov\left[B,E\right] \\ & + Cov\left[B,F\right] + Cov\left[B,G\right] + Cov\left[B,H\right] \\ & + Cov\left[C,D\right] + Cov\left[C,E\right] + Cov\left[C,F\right] \\ & + Cov\left[C,G\right] + Cov\left[C,H\right] + Cov\left[D,E\right] \\ & + Cov\left[D,F\right] + Cov\left[D,G\right] + Cov\left[D,H\right] \\ & + Cov\left[E,F\right] + Cov\left[E,G\right] + Cov\left[E,H\right] \\ & \left. + Cov\left[F,G\right] + Cov\left[F,H\right] + Cov\left[G,H\right]\right)_{\oplus} \end{aligned} \tag{3.90}$$

First, the cross-covariance terms may be handled. Generally it suffices to use Theorem B.1.10 (Expectancy of Independent Variables) or Theorem B.1.2 (Stacked Expectation), and recall that the random variables η, μ, $\varepsilon_m \left(f^B \right)$ and $\varepsilon_k \left(\omega_{IB}^B \right)$ have zero expectancy:

$$\begin{aligned} Cov\left[A,B\right] &= E\left[\left(\sum_{m=1}^{n} a_m C_m \left\{ \eta \right\} \right) \left(\sum_{m=1}^{n} a_m C_m \left\{ \left[\varepsilon_m \left(f^B \right) \right] \right\} \right)^T \right] \\ &= 0 \end{aligned} \tag{3.91}$$

$$Cov\left[A,C\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \tag{3.92}$$
$$= 0$$

$$Cov\left[A,D\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right] \tag{3.93}$$
$$= 0$$

$$Cov\left[A,E\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \tag{3.94}$$
$$= 0$$

$$Cov\left[A,F\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0 \tag{3.95}$$

$$Cov\left[A,G\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^{B}}\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \tag{3.96}$$
$$= 0$$

$$Cov\left[A,H\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \eta\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^{B}}\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0 \tag{3.97}$$

$$Cov\left[B,C\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m\ \varepsilon_m\left(f^B\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^T\right] \tag{3.98}$$
$$= 0$$

$$Cov\left[B,D\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m\ \varepsilon_m\left(f^B\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^B\right)\right)\right)^T\right]$$
$$= 0 \tag{3.99}$$

$$Cov\left[B,E\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m\ \varepsilon_m\left(f_m^B\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^B\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^T\right]$$
$$= 0 \tag{3.100}$$

$$Cov\left[B,F\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m \varepsilon_m\left(f_m^B\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(f_m^B\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^B\right)\right)\right)^T\right]$$
$$= 0 \tag{3.101}$$

$$Cov\left[B,G\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m\left\{\left[\ \varepsilon_m\left(f^B\right)\right]\right\}\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^T\right]$$
$$= 0 \tag{3.102}$$

$$Cov\left[B,H\right] = E\left[\left(\sum_{m=1}^{n} a_m C_m\ \varepsilon_m\left(f^B\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^B\right)\right)\right)^T\right]$$
$$= 0 \tag{3.103}$$

$$
\begin{aligned}
&Cov\left[C,D\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.104}
$$

$$
\begin{aligned}
&Cov\left[C,E\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.105}
$$

$$
\begin{aligned}
&Cov\left[C,F\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.106}
$$

$$
\begin{aligned}
Cov\left[C,G\right] &= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^{B}}\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.107}
$$

$$
\begin{aligned}
&Cov\left[C,H\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^{B}}\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.108}
$$

$$
\begin{aligned}
&Cov\left[D,E\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right] \\
&= 0
\end{aligned}
\tag{3.109}
$$

$$Cov\left[D,F\right] \tag{3.110}$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0$$

$$Cov\left[D,G\right]$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right]$$
$$= 0 \tag{3.111}$$

$$Cov\left[D,H\right] \tag{3.112}$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\eta\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0$$

$$Cov\left[E,F\right] \tag{3.113}$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0$$

$$Cov\left[E,G\right]$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)^{T}\right]$$
$$= 0 \tag{3.114}$$

$$Cov\left[E,H\right] \tag{3.115}$$
$$= E\left[\left(\sum_{m=1}^{n} a_m C_m\left(\varepsilon_m\left(f^{B}\right)\right)^{\times}\left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m\left(-\widehat{f_m^B}\right)^{\times}\left(\sum_{k=1}^{m} w_k \varepsilon_k\left(\omega_{IB}^{B}\right)\right)\right)^{T}\right]$$
$$= 0$$

$$
\begin{aligned}
&Cov\left[F,G\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m \left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right)\right)\left(\sum_{m=1}^{n} a_m C_m \left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu\right)\right)^T\right] \\
&= 0
\end{aligned}
\tag{3.116}
$$

$$
\begin{aligned}
Cov\left[F,H\right] = E\Bigg[&\left(\sum_{m=1}^{n} a_m C_m \left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right)\right)\cdots \\
&\left(\sum_{m=1}^{n} a_m C_m \left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right)\right)^T\Bigg] \\
&= 0
\end{aligned}
\tag{3.117}
$$

$$
\begin{aligned}
&Cov\left[G,H\right] \\
&= E\left[\left(\sum_{m=1}^{n} a_m C_m \left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu\right)\right)\left(\sum_{m=1}^{n} a_m C_m \left(-\widehat{f_m^B}\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right)\right)^T\right] \\
&= 0
\end{aligned}
\tag{3.118}
$$

Finally, it is concluded that all the cross-covariance terms are zero. Therefore, the variance is reduced to:

$$
\begin{aligned}
Var\left[A+B+C+D+E+F+G+H\right] = &\ Var\left[A\right] + Var\left[B\right] + Var\left[C\right] + Var\left[D\right] \\
&+ Var\left[E\right] + Var\left[F\right] + Var\left[G\right] + Var\left[H\right]
\end{aligned}
\tag{3.119}
$$

Analyzing the terms individually to compute their respective variances, used formulas are recalled when needed:

$$
\begin{aligned}
A &= \sum_{m=1}^{n} a_m C_m \eta \\
&\Rightarrow Var\left[A\right] \\
&= \left(\sum_{m=1}^{n} a_m C_m \eta\right) \sigma_\eta^2 \left(\sum_{m=1}^{n} a_m C_m \eta\right)^T
\end{aligned}
\tag{3.120}
$$

$$B = \sum_{m=1}^{n} a_m C_m \ \varepsilon_m \left(f^B\right) \tag{3.121}$$

B is reduced to Equation (3.46).

$$\begin{aligned} C &= \sum_{m=1}^{n} a_m C_m \left(\eta\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu\right) \\ &= \left(\sum_{m=1}^{n} a_m C_m (mdt)\right) \left(\eta\right)^{\times} \mu \end{aligned} \tag{3.122}$$

$$\begin{aligned} &\Rightarrow Var\left[C\right] \\ &= \left(\sum_{m=1}^{n} a_m C_m (mdt)\right) \left(\sigma_\eta^2 \boxtimes \sigma_\mu^2\right) \left(\sum_{m=1}^{n} a_m C_m (mdt)\right)^T \end{aligned} \tag{3.123}$$

$$D = \sum_{m=1}^{n} a_m C_m \left(\eta\right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right) \tag{3.124}$$

D is reduced to Equation (3.47).

$$\begin{aligned} E &= \sum_{m=1}^{n} a_m C_m \left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{m} w_k \mu\right) \\ &= \sum_{m=1}^{n} a_m C_m (t_m - t_0) \left\{ \left(\varepsilon_m \left(f^B\right)\right)^{\times} \mu \right\} \\ &= -\sum_{m=1}^{n} a_m C_m (t_m - t_0) (\mu)^{\times} \varepsilon_m \left(f^B\right) \end{aligned} \tag{3.125}$$

E is reduced to Equation (3.47).

$$F = \sum_{m=1}^{n} a_m C_m \left(\varepsilon_m \left(f^B\right)\right)^{\times} \left(\sum_{k=1}^{n} w_k \varepsilon_k \left(\omega_{IB}^B\right)\right) \tag{3.126}$$

F is reduced to Equation (3.52).

$$
\begin{aligned}
G &= \sum_{m=1}^{n} a_m C_m \left\{ \left[\left(-\widehat{f_m^B} \right)^{\times} \left(\sum_{k=1}^{m} w_k \mu \right) \right] \right\} \\
&= \left\{ \sum_{m=1}^{n} a_m C_m \left[\left(-\widehat{f_m^B} \right)^{\times} \left(\sum_{k=1}^{m} w_k \right) \right] \right\} \mu \\
&= \left\{ \sum_{m=1}^{n} a_m C_m \left[\left(-\widehat{f_m^B} \right)^{\times} (t_m - t_0) \right] \right\} \mu
\end{aligned}
\tag{3.127}
$$

G is reduced to a simpler case, represented by Theorem B.1.4.

$$
H = \sum_{m=1}^{n} a_m C_m \left(-\widehat{f_m^B} \right)^{\times} \left(\sum_{k=1}^{m} w_k \varepsilon_k \left(\omega_{IB}^B \right) \right) \tag{3.128}
$$

H is reduced to Equation (3.54).

3.7.2 Beta Error

Analogous to the previous section, the variance for $\delta\beta_v$ can be estimated, though with a much simpler development.

To approach the error for β_v, the velocity provided by the GPS is assumed synchronized with the IMU data, the GPS provides velocity outputs in NED (N) frame and position fixes in terrestrial (E) coordinates.

The models for the GPS measurements are discussed in Section A.3.3, which introduces Equation (A.29) and Equation (A.30). These equations are reproduced below, within them, a measurement is denoted with an over-hat, a Noise Type Error as defined by Definition 3.3.1 is denoted by $\varepsilon(.)$.

$$
\widehat{r^E} = r^E + \varepsilon \left(r^E \right) \tag{3.129}
$$

$$
\widehat{v^N} = v^N + \varepsilon \left(v^N \right) \tag{3.130}
$$

Because the system is GPS aided, $C_{N(0)}^{N(t)}$, ω_{EN}^{N} and ω_{IN}^{N} are assumed to have negligible errors, since they are all functions of position with small variations over displacements in the order of magnitude of the GPS position errors. Thus, recalling the expression for β_v and rewriting for the estimation case:

$$\beta_v = C_{N(t)}^{N(0)} v^N - v^N(0) + \int_0^t C_{N(\tau)}^{N(0)} \omega_{IE}^N \times v^N d\tau - \int_0^t C_{N(\tau)}^{N(0)} g^n d\tau \tag{3.131}$$

$$\widehat{\beta_v} = C_{N(t)}^{N(0)} \widehat{v^N} - \widehat{v^N(0)} + \int_0^t C_{N(\tau)}^{N(0)} \omega_{IE}^N \times \widehat{v^N} d\tau - \int_0^t C_{N(\tau)}^{N(0)} g^N d\tau \tag{3.132}$$

Therefore:

$$\begin{aligned} \widehat{\beta_v} - \beta_v &= \delta\beta_v \\ &= C_{N(t)}^{N(0)} \varepsilon\left(v^N\right) - \varepsilon\left(v^N(0)\right) + \int_0^t C_{N(\tau)}^{N(0)} \omega_{IE}^N \times \varepsilon\left(v^N\right) d\tau \end{aligned} \tag{3.133}$$

Recalling that integration is done with a weighting sequence with coefficients w_k:

$$\delta\beta_v = C_{N(t)}^{N(0)} \varepsilon\left(v^N\right) - \varepsilon\left(v^N(0)\right) + \sum_k w_k \left(C_{N(t_k)}^{N(0)} \left(\omega_{IE}^N\right)^\times\right) \varepsilon\left(v^N\right) \tag{3.134}$$

Then notably, as per the reasoning presented in Section 3.3.2:

$$E\left[\delta\beta_v\right] = \mathcal{O} \tag{3.135}$$

With $\mathcal{O}$ denoting the null vector. Then for the variance:

$$Var\left[\delta\beta_v\right] = C_{N(t)}^{N(0)} \sigma_v^2 \left(C_{N(t)}^{N(0)}\right)^T + \sigma_v^2 + \sum_k w_k^2 C_{N(t_k)}^{N(0)} \left(\omega_{IE}^N\right)^\times \sigma_v^2 \left(C_{N(t_k)}^{N(0)} \left(\omega_{IE}^N\right)^\times\right)^T \tag{3.136}$$

Apart from the terms related to the velocity at beginning and end, the variance falls back to a Case 1 propagation Equation (3.46).

3.7.3 From Variance to Angle Error

In the previous two sections, the variances for α_v and β_v were developed. Both vectors are ultimately used to compute an attitude and the interest in their error lies in estimating the error for such attitude. A method to fully achieve this goal is beyond the scope of the current work. Instead, a proxy that helps linking the estimated errors of these vectors to the performance of the alignment methods can be developed based on the tools introduced in this section. These developments are independent on which vector is being used, but β_v is used as a generic vector.

3.7.3.1 Expected Angle Method (OPT)

Since $Var\left[\delta\beta_v\right]_k$ is a covariance matrix, it is desired to determine the effect on the direction of β_v . Assuming θ is the angle between estimated and true vector. Stating that:

$$\begin{aligned} E\left[\left(\widehat{\beta_v}-\beta_v\right)\left(\widehat{\beta_v}-\beta_v\right)^T\right] &= Var\left[\delta\beta_v\right] \\ &= E\left[\left(\widehat{\beta_v}\widehat{\beta_v}^T\right)-\left(\beta_v\widehat{\beta_v}^T\right)-\left(\widehat{\beta_v}\beta_v^T\right)+\left(\beta_v\beta_v^T\right)\right] \end{aligned} \tag{3.137}$$

$$Var\left[\delta\beta_v\right] = E\left[\left(\widehat{\beta_v}\widehat{\beta_v}^T\right)\right]+E\left[\left(\beta_v\beta_v^T\right)\right]-E\left[\left(\beta_v\widehat{\beta_v}^T\right)+\left(\widehat{\beta_v}\beta_v^T\right)\right] \tag{3.138}$$

Noting that $\widehat{\beta_v}$ is a random variable, but β_v is not:

$$Var\left[\delta\beta_v\right] = E\left[\left(\widehat{\beta_v}\widehat{\beta_v}^T\right)\right]+\left(\beta_v\beta_v^T\right)-E\left[\left(\beta_v\widehat{\beta_v}^T\right)+\left(\widehat{\beta_v}\beta_v^T\right)\right] \tag{3.139}$$

Then taking the trace on the previous expression and using Equation (C.22):

$$\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right) = E\left[\left\|\widehat{\beta_v}\right\|^2\right]+\|\beta_v\|^2-E\left[\mathrm{Tr}\left(\beta_v\widehat{\beta_v}^T\right)+\mathrm{Tr}\left(\widehat{\beta_v}\beta_v^T\right)\right] \tag{3.140}$$

Using Theorem C.1.5:

$$\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right) = E\left[\left\|\widehat{\beta_v}\right\|^2\right]+\|\beta_v\|^2-2E\left[\widehat{\beta_v}^T\beta_v\right] \tag{3.141}$$

Rearranging:

$$E\left[\widehat{\beta_v}^T \beta_v\right] = \frac{E\left[\left\|\widehat{\beta_v}\right\|^2\right] + \|\beta_v\|^2 - \mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right)}{2} \tag{3.142}$$

Also assuming, as an approximation, that the norm of the estimated vector is much larger than the actual error, allows setting:

$$E\left[\left\|\widehat{\beta_v}\right\|^2\right] \approx \left\|\widehat{\beta_v}\right\|^2 \tag{3.143}$$

Within this assumption and recalling that the value of β_v is not available online, it is possible to replace the value of its module with the computed module of v:

$$\|\beta_v\|^2 \approx \left\|\widehat{\beta_v}\right\|^2 \tag{3.144}$$

Thus, Equation (3.142) is reduced to:

$$E\left[\widehat{\beta_v}^T \beta_v\right] = \left\|\widehat{\beta_v}\right\|^2 - \frac{\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right)}{2} \tag{3.145}$$

Recalling that the dot product between two vectors allows computing the cosine of the angle between them:

$$\cos\theta_\beta = \frac{\widehat{\beta_v}^T \beta_v}{\sqrt{\left(\widehat{\beta_v}^T \widehat{\beta_v}\right)\left(\beta_v^T \beta_v\right)}} \tag{3.146}$$

Thus, applying the expectancy operator:

$$E\left[\cos\theta_\beta\right] = E\left[\frac{\widehat{\beta_v}^T \beta_v}{\sqrt{\left(\widehat{\beta_v}^T \widehat{\beta_v}\right)\left(\beta_v^T \beta_v\right)}}\right] = E\left[\frac{\widehat{\beta_v}^T \beta_v}{\left\|\widehat{\beta_v}\right\| \|\beta_v\|}\right] \tag{3.147}$$

Reusing the previous approximations:

$$E\left[\frac{\widehat{\beta_v}^T \beta_v}{\left\|\widehat{\beta_v}\right\| \|\beta_v\|}\right] = \frac{E\left[\widehat{\beta_v}^T \beta_v\right]}{\left\|\widehat{\beta_v}\right\|^2} \tag{3.148}$$

Therefore:

$$E\left[\cos\theta_\beta\right] = \frac{E\left[\widehat{\beta_v}^T \beta_v\right]}{\left\|\widehat{\beta_v}\right\|^2} \tag{3.149}$$

Applying this result into Equation (3.145):

$$E\left[\cos\theta_\beta\right] \cong 1 - \frac{\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right)}{2\left\|\widehat{\beta_v}\right\|^2} \tag{3.150}$$

Thus an estimation for the direction error of $\widehat{\beta_v}$ can be given by:

$$\theta_\beta = \mathrm{acos}\left[1 - \frac{\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right)}{2\left\|\widehat{\beta_v}\right\|^2}\right] \tag{3.151}$$

For simulation purposes, an arc-cosine function cannot receive an argument outside the $[-1, 1]$ interval. Thus, the input value is constrained to lie within the intervals $[cos(\theta_\alpha)_{min}, 1]$ and $[cos(\theta_\beta)_{min}, 1]$, with this two variables being simulation parameters, listed on Section 4.1.

A similar calculation may be proceeded for α_v, to obtain θ_α from the variance of $\delta\alpha_v$.

Finally, a first rough estimation of the expected error proxy for the methods can be obtained by:

$$\theta_{err}^{est} = \theta_\alpha + \theta_\beta \tag{3.152}$$

An assessment of this estimate is the subject of Chapter 4. It is noted that the expectation of the error is given, but with no likelihood of the actual error being more or less than the computed amount.

3.7.3.2 Bounded at Probability Method (OPTc)

A minor extension from a result known in the literature as Chebyschev Inequality (Theorem B.1.13) is required in this section.

Proposition 3.7.1. Chebyschev Inequality (Multivariate):
Let Let X be an arbitrary n-dimensional random variable with mean μ and variance matrix K, then an upper bound on how likely X is to deviate from its mean, as measured by the vector norm, is given by:

$$P\left[\|X-\mu\| \geq k\sqrt{\mathrm{Tr}(K)}\right] \leq \frac{1}{k^2} \tag{3.153}$$

Proof: From Definition B.1.20, it follows:

$$K = \int_{\mathbb{R}^n} (x-\mu)(x-\mu)^T f_X(x)dx \tag{3.154}$$

Applying the trace operator:

$$\mathrm{Tr}(K) = \int_{\mathbb{R}^n} \mathrm{Tr}((x-\mu)(x-\mu)^T) f_X(x)dx \tag{3.155}$$

Using Theorem C.1.5:

$$\mathrm{Tr}(K) = \int_{\mathbb{R}^n} \|x-\mu\|^2 f_X(x)dx \tag{3.156}$$

Proceeding in analogous manner to the unidimensional case:

$$\int_{\mathbb{R}^n} \|x-\mu\|^2 f_X(x)dx \geq \int_{\|x-\mu\|\geq\delta} \|x-\mu\|^2 f_X(x)dx \tag{3.157}$$

$$\int_{\|x-\mu\|\ \geq\delta} \|x-\mu\|^2 f_X(x)dx \geq \delta^2 \int_{\|x-\mu\|\geq\delta} f_X(x)dx \tag{3.158}$$

$$\delta^2 \int_{\|x-\mu\|\ \geq\delta} f_X(x)dx = \delta^2 P\left[\|x-\mu\| \geq \delta\right] \tag{3.159}$$

Therefore:

$$\mathrm{Tr}(K) \geq \delta^2 P\left[\|x-\mu\| \geq \delta\right] \tag{3.160}$$

And setting $\delta = k\sqrt{\mathrm{Tr}(K)}$ completes the proof:

$$\frac{1}{k^2} \geq P\left[\|x-\mu\| \geq k\sqrt{\mathrm{Tr}(K)}\right] \tag{3.161}$$

Another approach to estimate an angle error relies on Proposition 3.7.1. Noting that β_v is estimated as $\widehat{\beta_v}$ and assuming again that this estimation is unbiased, Chebyschev's multivariate inequality allows asserting:

$$P\left[\left\|\widehat{\beta_v} - \beta_v\right\| \geq k\sqrt{\mathrm{Tr}\left(Var\left[\delta\beta_v\right]\right)}\right] \leq \frac{1}{k^2} \tag{3.162}$$

Noting that a probability p of the estimate being within the proposed bound can also be related to the parameter k by:

$$k = \frac{1}{\sqrt{1-p}} \tag{3.163}$$

Then, an upper bound in the error vector length is given. Therefore the estimation is known to be inside a sphere centered at the actual value with $1 - k^{-2}$ probability, as illustrated by Figure 3.18. Within this assumption, the largest possible angle between β_v and $\widehat{\beta_v}$ occurs when the latter is tangent to the sphere.

Figure 3.18 - Worst case for angle error within given probability.

In this case, the estimated worst angle error at $1 - k^{-2}$ probability is given by:

$$\theta_{\beta}^{p} = \arctan\left(\frac{k\sqrt{\mathrm{Tr}(Var\left[\delta\beta_v\right])}}{\left\|\widehat{\beta_v}\right\|}\right) \tag{3.164}$$

This approach is notably pessimistic, as the Chebyschev inequality is known to provide a generally loose bound (STARK; WOODS, 2002) and because the vector being tangent to the sphere is another very worst case. Hence, a likewise loose value can be chosen for p, as will be shown on Chapter 4.

And therefore, the expected error at probability p is given by:

$$\theta_{err}^{p} = \theta_{\alpha}^{p} + \theta_{\beta}^{p} \tag{3.165}$$

Once more, this derivation has taken no consideration for the characteristics of the TRIAD method. Its results are discussed in the next chapter.

3.8 Employing The Estimated Error

Once an expected error θ_{err} can be computed, it can be used on-line as a reference. First, it could provide a reference of the accuracy level of the performed alignment. On a second approach, knowing that the estimation quality varies over time, both improving or decreasing depending on the trajectory, the alignment computed at each time step can be either kept or discarded against a previously computed value associated with better estimated accuracy. While this approach may not yield the best possible result due to both the random nature of the process and the simplicity of disregarding the evolution of the error caused by propagation alone, it could provide a significant improvement in quality and timeliness over the methods explained in Chapter 2, these ideas are further discussed in Chapter 4. The results obtained with this strategy will be presented in Chapter 4 and labeled as the OPT and OPTc methods if the reference used is the estimated error or the error bound respectively. Since this computation was performed for the α_v and β_v vectors only, these are the vectors accumulated in the K matrix, causing the methods to behave similarly to the VIF method if the computed error proxies are always decreasing.

4 SIMULATIONS

This chapter explains the simulation setup and obtained results. Attention is given to remarks on general characteristics of the methods previously described that also required proper illustration from simulation results. The performances obtained are described for a particular case on Section 4.2.3 while the statistics estimated with Monte-Carlo simulations are shown on Section 4.2.4. Sections Section 4.2.1 and Section 4.2.2 provide important background on how to interpret the presented results. Notes on how the simulation was designed are given on Section 4.1.

4.1 Simulation Setup

Simulated data for a sounding rocket flight was used. The idea was to simulate the algorithm starting either on propelled stage or close to zero gravity. Hence for the result charts, the simulation may begin at different instants, before which no data is shown. The simulated flight is the same shown through the figures in Section 4.1.1 and Section 4.1.2, finishing at 60s.

It is recalled that many basic operations are performed using functions adapted from (CARRARA, 2015).

4.1.1 Simulated Trajectory

The raw data used for all simulations in this chapter are the reference (noise-free) IMU measurements at 100Hz frequency. These are the specific force and angular rate of the Body frame with respect to the Inertial frame expressed in Body Frame. They are shown in Figure 4.1 and Figure 4.2.

Figure 4.1 - Simulated Reference Specific Force over the IMU relative to Inertial Frame Expressed in Body Frame.

Figure 4.2 - Simulated Reference Angular Rate of IMU Relative to Inertial Frame expressed in Body Frame.

The beginning of the trajectory presents a high dynamics environment, which is beyond the operating range of most consumer and commercial application GPS receivers, but there are reports in the literature describing receivers for this very application being designed to maintain tracking ability up to 40g (RONCAGLIOLO et al., 2006), and earlier developments that were tested in circular trajectories to track pseudoranges while subjected to 50g (HURD et al., 1987).

4.1.2 Initial Parameters

Some parameters need to be set at time $t = 0$s so that they can be propagated until the end of the simulation at the IMU frequency with Runge-Kutta numerical integration. These are:

- The initial attitude of the Body frame with respect to the inertial frame $C_{B(0)}^{N(0)}$.
- The initial position in geodetic coordinates $[\phi_0, \lambda_0, h_0]$.
- The initial velocity in Navigation frame $v^N(0)$.

By setting those parameters, the profile over time of each of those magnitudes can be computed once by integration with the primary data and saved. The obtained values for each time instant t are retrieved to be used as initialization data when a simulation is performed starting at any later instant. This is done to compare the effect of initializing the algorithms at different moments on the same trajectory. The propagated values are shown in Figure 4.3, Figure 4.5 and Figure 4.6. The values for the rotation quaternion of the Body Frame with respect to both Inertial and Navigation frame are shown because, for the convenience of the simulation, the initial parameter used is indeed the one with respect to the Inertial Frame, however, the quantity that is to be estimated by the methods is the attitude with respect to the Navigation frame.

Figure 4.3 - Simulated Reference Attitude with respect to both Inertial and Navigation Frames.

For better interpretation, the attitude of the Body frame with respect to the navigation frame is also shown as Roll, Pitch and Yaw angles (FARRELL, 2008) in Figure 4.4.

Figure 4.4 - Simulated Roll, Pitch and Yaw Angles.

Figure 4.5 - Simulated Reference Position in geodetic coordinates.

Figure 4.6 - Simulated Reference Velocity in Navigation Frame.

4.1.3 Algorithm Inputs

The inputs that are provided by the simulation setup to each of the methods are listed below, these are all functions of time, and therefore associated with time stamps starting at zero. These are

v^N The GPS provided velocity.

ϕ The GPS provided latitude.

λ The GPS provided longitude.

h The GPS provided altitude.

a_{BI}^B The IMU measured specific force.

ω_{BI}^B The IMU measured angular rate.

Note that the real attitude matrix C_B^N is not informed to the algorithm but is generated by the simulation so that the estimation error can be computed. The signals associated with the position, such as the gravity acceleration are computed based on the GPS provided position, at the GPS rate, with no IMU coupling modeled.

4.1.4 Simulation Parameters

The parameters are values that remain unchanged between all simulations, these are:

σ_v^2 The GPS velocity variance matrix.

σ_ω^2 The IMU angular rate noise variance matrix.

σ_μ^2 The IMU angular rate bias variance matrix.

σ_f^2 The IMU acceleration noise variance matrix.

σ_η^2 The IMU acceleration bias variance matrix.

p The probability limit to be used by error estimation.

θ_{tol} The minimum angle between α_v and α_p and between β_v and β_p.

Δ_{tol} The maximum angle discrepancy between $\angle\left(\alpha_v, \alpha_p\right)$ and $\angle\left(\beta_v, \beta_p\right)$.

$\angle_{\alpha min}$ The minimum acceptable angle between α_v and α_p.

$\angle_{\beta min}$ The minimum acceptable angle between β_v and β_p.

$\|\alpha_v\|_{min}$ Minimum acceptable norm for α_v.

$\|\alpha_p\|_{min}$ Minimum acceptable norm for α_p.

$\|\beta_v\|_{min}$ Minimum acceptable norm for β_v.

$\|\beta_p\|_{min}$ Minimum acceptable norm for β_p.

ρ_v^{max} Maximum relative error between the norms of α_v and β_v.

ρ_p^{max} Maximum relative error between the norms of α_p and β_p.

$cos(\theta_\alpha)_{min}$ Minimum realistic value for estimating the angle error of α_v.

$cos(\theta_\beta)_{min}$ Minimum realistic value for estimating the angle error of β_v.

Note that variance matrices relate directly to characteristics of the sensors, hence their values should be changed for each specific equipment. The other parameters are tuning variables, which vary both with the INS itself and possibly with the application case i.e., a surface vehicle application such as a large ship would likely need meaningfully different parameters than those required by a sounding rocket.

4.1.5 Assessing Performance Results

For each estimated attitude $\widehat{C}_{B(t)}^{N(t)}$, the discrepancy between the real matrix $C_{B(t)}^{N(t)}$ from the simulated data and the estimate can be represented by the rotation angle between them. In a simplified manner:

$$\theta_{err} = \arccos\left[0.5\left(\text{Trace}\left(C_{B(t)}^{N(t)}\left(\widehat{C}_{B(t)}^{N(t)}\right)^T\right) - 1\right)\right] \tag{4.1}$$

This is the metric that the covariance models would ideally be able to estimate. Also, the smaller its value the better the estimation performance becomes.

Ideally, θ_{err} should be zero if a perfect estimation was possible. Since to the calculation of $C_{B(t)}^{B(0)}$ and the velocity with measurements diverge over time, there will be some discrepancy. In particular, biases in the angular rate sensor (and likewise in the accelerometer) are not considered nor computed on the alignment method, though present on simulation. This metric allows comparing results to see which method gave smaller errors and after how much time.

Also as important is to assess how θ_{err} relates to θ_{err}^{est} and θ_{err}^{p}, though this comparisons will be done separately. Again, it is recalled that θ_{err}^{est} and θ_{err}^{p} were computed exclusively from the expected errors for the α_v and β_v vectors, with little regard to the effect of these errors on the estimation methods. This is still relevant as a first step into further work to be performed.

4.1.6 Implementation Particularities

For implementation purposes, some computations differ in structure from their analytical counterparts presented earlier. The algorithm needs both vectors α_v and β_v to be initialized simultaneously, thus α_v computation is disabled until a first valid GPS measurement is available.

It is assumed that IMU outputs are provided regularly at a constant sample time dt. Thus for α_v:

$$\hat{u}_k = \left(\omega_{BI}^B\right)_k / \left\|\left(\omega_{BI}^B\right)_k\right\| \tag{4.2a}$$

$$q_{B(t_{k-1})}^{B(t_k)} = \left[\cos\left(\left\|\left(\omega_{BI}^B\right)_k\right\| dt/2\right), \sin\left(\left\|\left(\omega_{BI}^B\right)_k\right\| dt/2\right) \hat{u}_k{}^T\right]^T \tag{4.2b}$$

$$q_{B(t_k)}^{B(t_0)} = q_{B(t_{k-1})}^{B(t_0)} \otimes q_{B(t_k)}^{B(t_{k-1})} \tag{4.2c}$$

$$\alpha_v(t_k) = \alpha_v(t_{k-1}) + q_{B(0)}^{B(t_k)} \otimes f_k^B \otimes q_{B(0)}^{\bar{B(t_k)}} dt \tag{4.2d}$$

Afterwards, α_p is implemented with:

$$\alpha_p(t_k) = \alpha_p(t_{k-1}) + dt\alpha_v(t_k) \tag{4.3}$$

Let ϕ_0 and λ_0 be the first available GPS measurement of position at a time t_k, which may be varied between simulations, following Equation (A.13):

$$C_{E(0)}^{N(0)} = \begin{bmatrix} -\sin(\phi_0)\cos(\lambda_0) & -\sin(\phi_0)\sin(\lambda_0) & \cos(\phi_0) \\ -\sin(\lambda_0) & \cos(\lambda_0) & 0 \\ -\cos(\phi_0)\cos(\lambda_0) & -\cos(\phi_0)\sin(\lambda_0) & -\sin(\phi_0) \end{bmatrix} \tag{4.4}$$

This result is converted to a quaternion and stored for the whole simulation. For the consecutive measurements:

$$C_{E(t_k)}^{N(t_k)} = \begin{bmatrix} -\sin(\phi_k)\cos(\lambda_k) & -\sin(\phi_k)\sin(\lambda_k) & \cos(\phi_k) \\ -\sin(\lambda_k) & \cos(\lambda_k) & 0 \\ -\cos(\phi_k)\cos(\lambda_k) & -\cos(\phi_k)\sin(\lambda_k) & -\sin(\phi_k) \end{bmatrix} \tag{4.5}$$

Which is also converted to quaternion, and used along with $q_{E(0)}^{N(0)}$ as seen below.

The quaternion in scalar first convention representing the rotation of the Earth frame is computed with:

$$q_{E(t_0)}^{E(t_k)} = \begin{bmatrix} \cos(\omega_{IE}(t_k - t_0)/2) \\ 0 \\ 0 \\ \sin(-\omega_{IE}(t_k - t_0)/2) \end{bmatrix} \tag{4.6}$$

And finally:

$$q_{N(t_0)}^{N(t_k)} = q_{E(t_k)}^{N(t_k)} \otimes q_{E(t_0)}^{E(t_k)} \otimes q_{N(t_0)}^{E(t_0)} \tag{4.7}$$

Recalling that g_N is computed with Equation (A.12) as a function of ϕ_k. Then β_v is computed by:

$$\beta_v = q_{N(t_k)}^{N(t_0)} \otimes v^N(t_k) \otimes q_{N(t_0)}^{N(t_k)} - v^N(0) + \sum_{i=0}^{k}(t_k - t_{k-1}) q_{N(t_k)}^{N(t_0)} \left[\left(\omega_{IE}^N\right)^{\times} v^N(t_k) - g_k^N \right] \otimes q_{N(t_0)}^{N(t_k)} dt \tag{4.8}$$

Then β_p is implemented by:

$$\beta_p(t_k) = (t_k - t_{k-1})\beta_v(t_k) + \beta_p(t_{k-1}) \tag{4.9}$$

The equations for implementing the computation of the variance of $\delta\alpha_v$ and $\delta\beta_v$ are presented on Section 3.7 using the tools from Section 3.6.

The difference in the update rate of GPS and IMU computations can be seen for instance in Figure 2.2, where α_v evolves rather smoothly while β_v has a stair like growth.

Another particularity mentioned in Section 3.7.3.1, is that the arc-cosine functions used to estimate the angle error receive a saturated input. Whenever the input is below the lower-saturation bracket, the result should be considered invalid. This in informed by means of a validity flag. The saturation value is used as an upper scale value for the errors. The upper saturation bracket is set to 1, done merely as safeguard against numerical errors to forcibly prevent impossible inputs.

4.2 Simulation Cases and Specification

Over the next section, the four configuration cases of simulation are described and only the third and fourth cases, described on Section 4.2.3 and Section 4.2.4 simulate noises.

For each sensor, the covariances of the errors are chosen to be diagonal matrices with entries whose magnitudes are compatible with a GPS receiver and Inertial Measurement Unit suitable for the sounding rocket scenario. Their specific values are described on Table 4.1, all at 1σ confidence level.

The equations for modeling the errors of each sensor are explained in Section A.3.1, Section A.3.2, Section A.3.3 and Section 3.4.3.

The covariances used for the simulations are obtained by multiplying the square of the accuracies presented on Table 4.1 by an identity matrix. The associated

Table 4.1 - Simulated Data Specifications

Specification	Value
IMU Frequency	100Hz
Angular Rate Noise	$1,2'/\sqrt{h}$
Angular Rate Bias Repeatability	$2°/h$
Accelerometer Noise	7µG
Accelerometer Bias Repeatability	$<$ 550µG
GPS Frequency	1Hz
GPS Position Accuracy	5m
GPS Velocity Accuracy	$0,1m/s$

noises and biases are generated as described on Section 3.4.3.

4.2.1 Unsampled Noiseless Simulation

The GPS measurements are provided at 1Hz, hence numerical integration errors by themselves are expected to have some impact on accuracy, as well as the effect of the sensors' noise. As a general reference, these are the results obtained by integrating both vectors at the IMU rate of 100Hz when no noise is added to the process. These are referred as the real values of the abstract quantities α_v, α_p, β_v and β_p and shown in the following figures for the sake of completeness. The key takeaway note is that the norms of the vectors are exactly the same in this simulation.

In Figure 4.7 and Figure 4.9, the reference values for the vectors are shown. Naturally, since they're the same vector expressed in different frames, the graphs are different with some resemblances. As remarked on Section 2.3.4, the norms should be the same as shown in Figure 4.8 and Figure 4.10.

Figure 4.7 - Simulated real values of v vectors.

Figure 4.8 - Simulated norm of real values of v vectors.

Figure 4.9 - Simulated real values of p vectors.

Figure 4.10 - Simulated norm of real values of p vectors.

Consequently, all methods yield perfect estimation of the attitude once they have enough information to converge, which happens when the pair of vectors in each frame are not parallel for TRIAD method and when the matrix K becomes full rank for other methods. The graph is shown in Figure 4.11, and details on its computation were explained on Section 4.1.5.

Figure 4.11 - Performance of Methods under no integration or noise errors.

4.2.2 Sampled Noiseless Simulation

Because numerical integration with smaller sample rate for the GPS will introduce errors, which have been neglected on Section 3.7, a simulation is performed to verify both the impact of this error component and to create reference values for α_v, α_p, β_v and β_p, such that in the next section, the sensor's noise impact is assessed eliminating most of the sampling effect. The reason for this is that the trajectory itself cannot be varied over Monte-Carlo simulations, hence the error components found in this simulation will be present on all others and remain relatively unaffected by the sensor's noise.

In Figure 4.12, both vectors are plotted at the 100Hz frequency, however, β_v is updated at the GPS sample rate and kept constant between samples.

Figure 4.12 - Norm of v vectors under no noise errors.

Figure 4.13 - Norm of p vectors under no noise errors.

4.2.3 Performance Simulation

Two main test cases are used. One starting at 0s and another at 15s, recalling that the trajectory remains the same, only the starting time of the methods is shifted to a moment when the rocket is no longer propelled. These simulations provide insight on the behavior of the methods and allow making a preliminary calibration of some tuning parameters. However, care should be taken when drawing conclusions from few realizations of a random process. The performances obtained herein are no more than anecdotal evidence, since they have no statistical significance, which in turn can only be assessed with Monte-Carlo tests, such as those described on Section 4.2.4. Nonetheless, the better the sensors involved, the more representative an individual simulation becomes. This will be further discussed on Section 4.2.4. Furthermore, some characteristics of the behavior of those methods may be lost on averaged curves, and as such is important to provide at least a couple examples of the methods being executed.

For illustration, the sensor outputs with modeled errors are shown in Figures 4.14 to 4.17

Figure 4.14 - GPS provided Velocity in NED frame.

Figure 4.15 - GPS provided Position.

Figure 4.16 - IMU provided specific force.

Figure 4.17 - IMU provided angular rate.

For illustration, the evolution of α_v, β_v, α_p and β_p is shown in Figures 4.18 to 4.20 for a simulation starting at 0s.

Figure 4.18 - Evolution of α_v and β_v starting at 0s.

Likewise, their norms are compared in figures Figure 4.19 and Figure 4.20.

Figure 4.19 - Evolution of α_p and β_p in module starting at 0s.

Figure 4.20 - Evolution of α_p and β_p in Module starting at 0s.

As shown in Section 2.3.2, the norms of the vectors are very close together even under noise, although β_v and β_p are updated at a slower rate.

A few plots related to the functioning of the Filtered Method are presented next.

Figure 4.21 - Angles between vector pairs.

Figure 4.22 - Angle Discrepancy between vector pairs.

Figure 4.23 - Norm Discrepancy between vectors.

The estimated covariances for $\delta\alpha_v$ and $\delta\beta_v$ are not shown in this section since there would be nothing to compare them against at this point. The results that derived from their values are shown here nonetheless, to provide insight into the behavior of the covariance-based methods, again, the result is of little statistical representativeness.

Figure 4.24 - Estimated angle errors.

Figure 4.25 - Estimated angle error bound.

The evolution of all these errors depends on the trajectory. For this simulation, they are expected to decrease for a short while, then diverge. This happens because the covariance errors (as estimated online) grow quadratically with time, while the norm of the vectors grows linearly with time (with variations depending on the trajectory), thus the squared norm also grows quadratically with time. Depending on slopes and on the variations from the trendlines described in Section 2.3.2, it is to be expected that the error associated with α_v would grow over time due to the effect of errors in the calculation of the body rotation matrix as roughly indicated by Equation (3.67). Meanwhile, the variance of β_v increases slowly but starts at a relatively larger value, while the norm increases at the same pace as that of α_v. At the beginning however, and specially for this simulation that involves high accelerations, it is also expected that the errors should decrease for a short while.

Notably, the error estimated by both methods reaches a minimum around 6 seconds, with its specific location possibly varying between simulations. This will cause the OPT and OPTc methods to declare convergence at that instant.

Finally, the chart comparing the performances for this specific test case can be presented in Figure 4.26.

Figure 4.26 - Angle errors.

The black dotted line represents the error obtained by comparing the reference attitude with the actual value computed in the simulation with the simulated data, it remains on the chart mostly as a development artifact to verify if any initialization error could have taken place. It appears on the legend, but it is actually superimposed over the abscissa axis.

As seen on Chapter 1, the VIF and PIF method oscillate in performance for this

simulation, with their accuracy decreasing in several instances during the first ten seconds. The TRIAD method has no memory of the past values of the used vectors, hence it oscillates more than the others in accuracy. The filtered method takes 4 seconds before providing a valid attitude because not all its checks had converged before, however, this does not ensure sufficient performance of the method, which later oscillates significantly. The OPT and OPTc methods stop improving at 5s, but at this point they declare convergence, although the performance is not the best achieved, this is a very fast convergence time, with reasonable accuracy.

These graphs are now repeated for the simulation starting at 15s.

Figure 4.27 - Evolution of α_v and β_v for simulation starting at 15s.

Figure 4.28 - Evolution of α_p and β_p in module for simulation starting at 15s.

Figure 4.29 - Evolution of α_p and β_p in Module for simulation starting at 15s.

Figure 4.30 - Angles between vector pairs for simulation starting at 15s.

Figure 4.31 - Angle Discrepancy between vector pairs for simulation starting at 15s.

Figure 4.32 - Norm Discrepancy between vectors for simulation starting at 15s.

Figure 4.32 shows that for the example case starting at 15s, the relative norm discrepancies surpass or approach the tolerance lines more often than shown in Figure 4.23, this is likely due to the magnitude of errors being similar for both cases, but the algorithms starting at 15s yield smaller magnitudes for the actual norms, hence increasing the relative error.

Figure 4.33 - Estimated angle errors for simulation starting at 15s.

Figure 4.34 - Estimated angle errors for simulation starting at 15s.

Notably in this case, both the estimated error and the bound are monotonically decreasing, as shown in Figure 4.33 and Figure 4.34. Due only to initialization reasons, the initial angle error for the vectors is shown as zero at t = 15s in Figure 4.33 but, at that instant, both vectors have zero norm.

In both cases, the error of the β_v angle has a flatter curve while α_v may decrease or increase more steadily, this is because the main contributors to the error in β_v are the two velocity measurements, which tend to have nearly constant magnitude, while α_v has a more complex error model which is expected to diverge faster under the presence of sensors biases. The resulting performance is shown in Figure 4.35.

Figure 4.35 - Angle errors for simulation initialized at 15s.

Notably, the monotonically decreasing error estimations cause the OPTc methods to match exactly the results from the VIF method, as it is simply accumulating the α_v and β_v vectors on the matrix K. Meanwhile, OPT method gives slightly different performance because of an initialization difference that requires the angle error estimation to be valid twice before the vectors start being accumulated. The TRIAD method error still oscillates considerably, since this method has less memory than the others, for which the accumulation of values on the K matrix causes new inputs to be relatively less relevant at each updating step.

4.2.4 Monte-Carlo Simulations

This section describes the conduction of Monte-Carlos tests, presenting and analyzing their results. Notes are given on the handling of the seeds for random number generation and the considerations required to interpret the information exposed.

In this section, the analysis of α_v and β_v errors is performed by taking the difference between their values in each individual simulation and their values obtained from the reference simulation described in Section 4.2.2, this is meant to remove the effect of the integration error, exposing mostly the noise effects.

For the performance of methods, the comparison is done against the correct attitude, shown in Figure 4.3, which does not change between all simulations under the same trajectory, even if varying the stating moment. In simpler words, for the simulations starting at 15s, the position, velocity and attitude are initialized with the values shown at the 15s step in figures 4.3, 4.5 and 4.6.

Effectively these two comparison guidelines allow verifying the validity of the methods under the assumptions taken, while assessing the final performance as it should be obtained on real experiments.

The results are presented for both the case starting at 0s and 15s, in order to minimally represent the variation of the starting moment. The number of performed simulations for each case is shown on the graphs as needed.

4.2.4.1 Managing Seeds for Random Number Generation

Each random parameter (such as the biases) is generated by a random number generating function that requires an initial state, commonly referred as the "seed". Controlling this seed allows ensuring the reproducibility of results and the variation of relevant factors between simulations.

At the beginning of each simulation, that corresponds to a realization of the random process, a global seed is configured explicitly. It is 1 for the simulations shown in Section 4.2.3 to simplify reproducibility. When setting Monte Carlo tests, a file is either read or created to store the last global seed used for a Monte Carlo test. At the beginning of each run, the global seed is declared by incrementing the value of the previous seed, and the used value is stored in the same file, overwriting it. This ensures that the global seed is not repeated between simulations.

For sequences of random numbers, such as the Noise Type Errors, their generating functions have dedicated seeds, which are chosen during initialization as random integer values, whose choice depends only on the global seed.

Further discussions on the management aspects of seeds and the generation of

random numbers are presented in (GENTLE, 2006).

4.2.4.2 Average Vector Error

It has been assumed in Section 3.3.2 that the average error on the vectors α_v and β_v would be zero, if accounting for noise, but not for numerical integration errors. This hypothesis can be assessed with the sample average of the errors obtained for the simulations, which is shown in Figure 4.36 and Figure 4.37. In these figures, the obtained average is compared with the sample estimated covariance for each dimension of the vector divided by the square root of the number of simulations. As a consequence of the Central Limit Theorem (Theorem B.1.15), at each time index, this is expected to represent the variance of the estimation of the mean. Because the values are not far from these standard deviation lines, one could claim that the average error being zero cannot be falsified at two sigma confidence level, the actual length of the interval can be as small as required but at the cost of running an order of 100 times more simulations to reduce by a factor of 10.

Figure 4.36 - Average vector errors for simulation starting at 0s.

Figure 4.37 - Average vector errors for simulation starting at 15s.

Recalling from Figure 4.18 and Figure 4.27 that the norms of these vectors is in the order of 10^2m/s, hence the encountered values are likely negligible within this order of magnitude, and with respect to the actual errors in each simulation. To further illustrate how much this average error represents in terms of the actual value of the involved vectors, the relative error i.e., the error divided by the norm of the vector is plotted next.

Figure 4.38 - Relative average vector errors for simulation starting at 0s.

Figure 4.39 - Relative average vector errors for simulation starting at 15s.

This allows claiming that up to the confidence level allowable by the number of experiments, the claim that the average error would be zero is not falsified. A confidence interval could be constructed, but it is recalled that rather than saying that the average error should be small, it should be exactly zero, which is hard to prove statistically.

4.2.4.3 Estimating Covariance

In the previous section, the sample covariance was used for comparison with the average error, now its value will be compared against the predicted covariance values.

Prior to present the encountered values, an implicit hypothesis made while developing the methods which needs to be checked. It was assumed that with the trajectory being the same, and despite the different biases and noise realizations,

the estimated covariance would either accompany actual errors discrepancies or would vary very little between simulations. While the former assumption could be difficult to analyze with data, the latter can be proven correct by computing a dimensionless ratio that indicates the fact that the estimated covariances remain nearly unchanged between simulations.

Figures 4.40 to 4.43. show the difference between the largest and smallest computed values for the diagonal elements of the covariance matrices normalized by the sample estimated counterpart. This is done with each element of the diagonals and their quadratic sum. In these graphs, the ratios below 1.2×10^{-7} show that the encountered discrepancies between simulations are small relative to the actual values of the computed covariances. Allowing the conclusion that computed covariances remain nearly unchanged between different simulations of the Monte-Carlo campaign.

Figure 4.40 - Ratio of covariance maximum difference to sample estimate for α_v, for simulations starting at 0s.

Figure 4.41 - Ratio of covariance maximum difference to sample estimate for β_v, for simulations starting at 0s.

Figure 4.42 - Ratio of covariance maximum difference to sample estimate for α_v, for simulations starting at 15s.

Figure 4.43 - Ratio of covariance maximum difference to sample estimate for β_v, for simulations starting at 15s.

With the conclusion that between simulations the estimated covariance varies at least relatively little, it is possible to plot the sample covariance against average of the estimated covariances to verify the validity of the calculation, as was done in Section 3.4.2. It is noted that since the estimated angle error and the error bound depend mostly on the estimated covariance, the fact that this covariance is nearly unchanged between simulations allows using the average of those estimates as a reliable representation for the actually encountered values, hence why their discrepancies between simulations are not further analyzed.

Figure 4.44 - Sample Covariance against Average Estimated Covariances for α_v for simulation starting at 0s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances.

Figure 4.45 - Sample Covariance against Average Estimated Covariances for α_v for simulation starting at 15s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances.

Figure 4.46 - Sample Covariance against Average Estimated Covariances for β_v for simulation starting at 0s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances.

Figure 4.47 - Sample Covariance against Average Estimated Covariances for β_v for simulation starting at 15s. Black lines labeled 'est' are the average of the covariances estimated by the methods, blue lines labeled 'clc' are the sample estimated covariances.

The cross-terms in both graphs have significantly smaller magnitudes than the diagonal terms. This is partially a result of the covariance matrices given as inputs being all diagonal matrices. such that the estimated values for the off-diagonal terms is much smaller. As happened with the average error, it is therefore difficult to reliably estimate average values with are close to zero under relatively large variance. Noting that the methods OPT and OPTc operate on the diagonal elements of these matrices rather than on the off-diagonal ones.

Again, the kernel estimations are plotted together with their smoother's bandwidths providing further insight on the behavior of the errors.

Figure 4.48 - Kernel Estimated Probability Density for α_v for simulation starting at 0s.

Figure 4.49 - Kernel Smoother Bandwidth for α_v for simulation starting at 0s.

Figure 4.50 - Kernel Estimated Probability Density for β_v for simulation starting at 0s.

Figure 4.51 - Kernel Smoother Bandwidth for β_v for simulation starting at 0s.

Again, the same is done for the simulations starting at 15s.

Figure 4.52 - Kernel Estimated Probability Density for α_v for simulation starting at 15s.

Figure 4.53 - Kernel Smoother Bandwidth for α_v for simulation starting at 15s.

Figure 4.54 - Kernel Estimated Probability Density for β_v for simulation starting at 15s.

Figure 4.55 - Kernel Smoother Bandwidth for β_v for simulation starting at 15s.

4.2.4.4 Assessing Bound at Probability

Recalling that on Section 3.7.3.2, a bound was given for the error under some assumptions, it is possible to verify in each simulation if the angle error of each vector is below or above the proposed bound, and count in how many simulations each case was found, this allows raising a statistic of how often the bound was respected, recalling that ideally this should match the given p parameter, however, a larger value should be encountered since the sphere tangency assumption is rather pessimistic.

Figure 4.56 - Error below bound frequency for simulations starting at 0s.

Figure 4.57 - Error below bound frequency for simulations starting at 15s.

During the tuning of parameters, a value of $p = 50\%$ was selected, as it was seen to generate bounds for around 95% of the cases, as shown in Figure 4.56 and Figure 4.57. The initial values of the estimated bound are rather large, thus why both charts start at 100%.

The actual bound compared to the average value is presented in Figure 4.58, Figure 4.59, Figure 4.60 and Figure 4.61 where it is compared with the sample mean angle error curve, in turn two curves adding and removing one standard deviation from the mean are also plotted, thus showing that indeed, as expected from the previous charts, the bound indeed provides a realistic though conservative estimation of the error, since it has similar shape and sits above the aforementioned curves.

Figure 4.58 - α_v Error bound comparison for simulations starting at 0s.

Figure 4.59 - α_v Error bound comparison for simulations starting at 15s.

Figure 4.60 - β_v Error bound comparison for simulations starting at 0s.

Figure 4.61 - β_v Error bound comparison for simulations starting at 15s.

4.2.4.5 On the Approximations for the Expected Angle Error Formulation

Two approximations were adopted during the derivation of the equations for the expected angle error, shown in Section 3.7.3.1. These can be assessed by verifying the squared ratio between norms of the computed values for α_v and β_v and their reference magnitude. A unitary ratio would imply that the result is perfectly adequate, whereas values close to unity signal a reasonable approximation.

Figure 4.62 - Probability Density for Squared Norm Ratio Between calculated and reference value of α_v for simulations starting at 0s.

Figure 4.63 - Probability Density for Squared Norm Ratio Between calculated and reference value of β_v for simulations starting at 0s.

Figure 4.64 - Probability Density for Squared Norm Ratio Between calculated and reference value of α_v for simulations starting at 15s.

Figure 4.65 - Probability Density for Squared Norm Ratio Between calculated and reference value of β_v for simulations starting at 15s.

4.2.4.6 Assessing Estimated Angle Error

As was done with the error bound, the quality of the estimated angle error can be checked by comparing with the sample estimated average error, in this case the average estimation fell below the $+1\sigma$ line, which implies the estimation to be reasonable though slightly conservative, which in this case was not intended, but is a consequence of the approximations performed in Section 3.7.3.1. Again, the shape of the curves is quite similar.

Figure 4.66 - Mean estimated angle errors comparison for simulations starting at 0s.

Figure 4.67 - Mean estimated angle errors comparison for simulations starting at 15s.

For minimally checking this factor, it is recalled that the estimated error is expected to have little variance between simulations, this can be verified by plotting the sample covariance of the estimated angle errors, showing indeed that the variance of this computation is small.

Figure 4.68 - Variance of estimated angle errors comparison for simulations starting at 0s.

Figure 4.69 - Variance of estimated angle errors comparison for simulations starting at 15s.

4.2.4.7 Assessing Methods Performances

Finally, it is possible to have a relative assessment of the performances of the methods by plotting their average performances over the Monte-Carlo tests in the same chart. Notably, these results are specific to the given trajectory and to the tuning of the filtered method, but the noise and biases realizations are varied through the simulations.

Figure 4.70 - Comparison of average methods performance for simulations starting at 0s.

Figure 4.71 - Comparison of average methods performance for simulations starting at 15s.

The OPTc method error curve superposes that of the VIF method, for a few seconds in Figure 4.70 and almost until the end of the simulated time in Figure 4.71. Tables 4.2 and 4.3 provide the same data in selected instants.

As was seen in Section 4.2.3, in the simulations starting at 0s, the OPT and OPTc convergence do not improve after a moment around 6s, as the error proxies have reached a minimum around this iteration, which can be expected to be a consistent behavior since the minimum taking more or less to occur would cause the performance curve to oscillate more, this is not always the case as will be shown in Figure 4.74. In simulations starting at 15s, the proxies monotonically decreasing cause the OPT and OPTc methods to behave very similarly to the VIF method.

To provide further insight on the characteristics of the methods, their performances are also compared with their sample standard deviations and the ref-

erence error, which was obtained for each method as explained and presented on Section 4.2.2. By adding the standard deviation to the mean error an upper band is created and by subtracting the deviation a lower band is produced to present an interval of plus or minus one deviation around the mean alignment error of each method, shown in Figure 4.72 and Figure 4.73. This allows noting the effect of the early convergence of the covariance based methods, since in a few iterations the output their final estimation, the effect of noise varying between simulations cause the variance to keep high and nearly constant.

The PIF method has particularly poor performance in the beginning, which is due to the significant contribution of the numerical integration of the GPS, since it happens even for the noiseless case.

Figure 4.72 - Comparison of average methods performance with their one standard deviation interval for simulations starting at 0s.

Figure 4.73 - Comparison of average methods performance with their one standard deviation interval for simulations starting at 15s.

As an alternative to the standard deviation, it is possible to plot percentiles of performances, hence having an idea of how existing non-extreme cases behave.

Figure 4.74 - Comparison of average methods performance with reference percentiles for simulations starting at 0s.

Figure 4.75 - Comparison of average methods performance with reference percentiles for simulations starting at 15s.

With Figure 4.74 and Figure 4.75, it is possible to see that for at least 10% of the simulations, all methods may take long before providing acceptable estimates, seen by the 90% percentile line being stuck at 10°, the TRIAD method takes the longest before the referred line drops below the saturation limit, hence confirming that this method is indeed expected to take longer before converging or even start providing estimated. Then, the PIF and TRIAD methods also take long before dropping below saturation at the 90% percentile. VIF, OPT and OPTc methods have in this situation (which contains slower dynamics) taken similar lengths.

In all cases, both the average minus 1σ and the 10% percentile line are well above the reference line at least after a transition period, which allows concluding that indeed the integration errors play a minor role when compared to the sensor's noises and biases, as has been assumed on Section 3.3.2.

Tables 4.2 and 4.3 present a numerical comparison of the average performance for each method for the simulations starting at 0s and 15s respectively. In this table, the best accuracy at each time is highlighted in red. For the simulations starting at 0s the VIF method provided the best performance at all instants, whereas the OPT and OPTc methods have nearly constant performance because they have declared convergence, and thus stopped improving their results. This creates an obstacle for directly comparing those methods, since time earned might be traded off against accuracy. However, for the simulations starting at 15s, no convergence is declared since the error proxies are generally decreasing during the simulation, thus OPT and OPTc provide better performance in some occasions, as they are less prone to have increasing error and possibly due to initialization effects. The superposition of the VIF and OPTc performances is better seen in this table than in Figure 4.71.

Table 4.2 - Comparison table for Simulation starting at 0s.

Time\Method	VIF	PIF	TRIAD	FIL	OPT	OPTc
00s	10.000°	10.000°	10.000°	10.000°	10.000°	10.000°
05s	2.539°	3.038°	4.321°	3.347°	3.422°	2.539°
10s	1.591°	5.812°	2.875°	6.204°	3.422°	2.539°
15s	0.532°	1.119°	0.825°	0.870°	3.422°	2.539°
20s	0.353°	0.440°	0.557°	0.424°	3.422°	2.539°
25s	0.296°	0.334°	0.415°	0.330°	3.422°	2.539°
30s	0.271°	0.292°	0.371°	0.291°	3.422°	2.539°
35s	0.256°	0.271°	0.328°	0.270°	3.422°	2.539°
40s	0.249°	0.258°	0.304°	0.258°	3.422°	2.539°
45s	0.245°	0.251°	0.300°	0.251°	3.422°	2.539°
50s	0.242°	0.247°	0.283°	0.247°	3.422°	2.539°
55s	0.241°	0.244°	0.275°	0.244°	3.422°	2.539°
60s	0.240°	0.242°	0.273°	0.242°	3.422°	2.539°

As could be expected, the error proxies do not relate as directly to the methods' errors such that some clear relationship could stand out in the present charts, all the errors are in the order of few degrees while the proxies are in the order of tenths of degrees. This is partially due to the fact that the proxies encode the accuracy for the direction of the α_v and β_v vectors, while the accuracy in other directions also affects the alignment error. Additionally, the mean performance for the VIF method does not showcase the occurrence of oscillations in the performance, as

Table 4.3 - Comparison table for Simulation starting at 15s.

Time\Method	VIF	PIF	TRIAD	FIL	OPT	OPTc
15s	10.000°	10.000°	10.000°	10.000°	10.000°	10.000°
20s	7.973°	8.259°	8.035°	9.745°	7.988°	7.973°
25s	5.152°	5.948°	5.762°	7.576°	4.984°	5.152°
30s	2.534°	3.739°	2.926°	4.781°	2.427°	2.534°
35s	1.442°	2.187°	1.917°	2.931°	1.397°	1.442°
40s	1.184°	1.609°	1.730°	2.238°	1.155°	1.184°
45s	1.008°	1.334°	1.459°	1.825°	0.988°	1.008°
50s	0.861°	1.144°	1.121°	1.584°	0.846°	0.861°
55s	0.714°	0.983°	0.926°	1.389°	0.718°	0.725°
60s	0.611°	0.847°	0.779°	1.234°	0.619°	0.611°

was the case in Figure 4.26 and Figure 4.35 where the error of the VIF method increases after the proxies reach a minimum. Hence, the relation between the covariance of the vectors and the alignment performance remains as the subject of further research. Nonetheless, the proxies allow for decision making within the methods, preventing oscillations in performance and possibly allowing an early convergence declaration, which allows the nominal navigation filter to be initialized sooner.

5 CONCLUSION

A review of relevant topics on navigation systems and stochastic processes was performed to pave the way for new developments in in-flight alignment. An analysis of the methods proposed on (WU; PAN, 2013) was carried out, and statistical results were obtained comparing with derived methods with the current simulated experiments. The selectivity scheme originally presented on (KOJA; LEITE FILHO, 2017) provided indeed an added value in terms of reliability, albeit dependent on designer chosen threshold values, some additional considerations to this method were also introduced in this work. The variance propagation scheme based on error models was validated with Monte-Carlo tests, showing good adherence to sample estimated covariances and provided promising tools for developing methods less dependent on designer choices. The OPT and OPTc methods have been introduced in this work, and provide promising results, with properties both similar and distinguishable with respect to the VIF and PIF methods. An overall statistical comparison of all those methods was conducted for a few reference test cases in a given flight envelope, and particularly the low variance of the results obtained from the covariance estimation was demonstrated, thus corroborating that the approach is promising although requiring further developments.

Future work can be pursued by several venues, such as:

- Systematically simulating a wider variety of flight envelopes, this is paramount in establishing robustness of those methods.
- Developing a reset criterion to re-start the methods if necessary due to poor flight conditions, which might also be of interest for practical purposes.
- The covariance propagation employed for β_v could also be slightly modified by considering a variable GPS measurement covariance, which is expected to be informed by the receiver as the Dilution of Precision.
- A better connection between the actual alignment error and the predicted variance matrices could be investigated.
- Variance models for α_p and β_p could be developed, along with a model for the correlation with the respective v vectors. This would be a substantial extension of the current work.

- Since the TRIAD method having known methods for covariance computation given the variance of the input vectors, if the aforementioned models were available, a model for the accuracy of the TRIAD method could then be developed.

- The TRIAD algorithm could be applied to pairs of α_v and β_v vectors in different instants

- Due to the approximations used on the development of the expected angle error, a different approach could be considered employing the theory of stochastic processes over manifolds, which would incorporate form the beginning the assumption of the computed norm being the true norm of the vector.

- Methods to encode the covering of the three-dimensional space for the collected set of vectors should allow improving the error proxies relation to the alignment performance, since the angle errors are essentially a one-dimensional error proxy, while the alignment estimation involves a three dimensional space.

- In Chapter 4 it was shown that strong relations exist between the angles between the vectors in each frame and the performance on the estimation must exist, and as such should be investigated.

- A metric could be developed to allow single-criterion comparison between the performance of the methods, one that could encode aspects of average error, error distribution and speed of convergence.

REFERENCES

ALI, J.; USHAQ, M. A consistent and robust Kalman filter design for in-motion alignment of inertial navigation system. **Measurement**, v. 42, n. 4, p. 577–582, 2009. 2

BAZIW, J.; LEONDES, C. T. In-flight alignment and calibration of inertial measurement units-part i: general formulation. **IEEE Transactions on Aerospace and Electronic Systems**, n. 4, p. 439–449, 1972. 13

BLACK, H. D. A passive system for determining the attitude of a satellite. **AIAA Journal**, v. 2, n. 7, p. 1350–1351, 1964. 9

BOWMAN, A. W.; AZZALINI, A. **Applied smoothing techniques for data analysis: the kernel approach with S-plus illustrations**. [S.l.]: Oxford University Press, 1997. 47, 48

BRUMBERG, V.; GROTEN, E. IAU resolutions on reference systems and time scales in practice. **Astronomy & Astrophysics**, v. 367, n. 3, p. 1070–1077, 2001. 174

CALLIOLI, C. A.; DOMINGUES, H. H.; COSTA, R. C. F. **Álgebra linear e aplicações**. [S.l.]: Atual, 2007. 171

CAMPOS, R. E. D.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. Auto-alinhamento de sistemas de navegação inercial usando algoritmo REQUEST com redes neurais artificiais. In: SOCIEDADE BRASILEIRA DE AUTOMAÇÃO, XIII., 2004, Porto Alegre, Rio Grande do Sul. **Simpósio Brasileiro de Automação Inteligente**. [S.l.], 2017. 12

CARNAHAN, B.; LUTHER, H. A.; WILKES, J. O. **Applied numerical methods**. [S.l.]: John Wiley & Sons, 1969. 182

CARRARA, V. An open source satellite attitude and orbit simulator toolbox for Matlab. In: ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA E CIÊNCIAS MECÂNICAS, 17., 2015, Natal, RN. **Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics**. [S.l.], 2015. 15, 83

CRASSIDIS, J. L. Sigma-point Kalman filtering for integrated GPS and inertial navigation. **IEEE Transactions on Aerospace and Electronic Systems**, v. 42, n. 2, p. 750–756, 2006. 1, 183, 185

DECKER, B. L. **World geodetic system 1984**. [S.l.], 1986. 12, 174, 175, 176

FARRELL, J. **Aided navigation: GPS with high rate sensors**. [S.l.]: McGraw-Hill, 2008. 12, 14, 15, 39, 86, 173, 174, 177, 178, 181

FEY, A.; GORDON, D.; JACOBS, C.; MA, C.; GAUME, R.; ARIAS, E.; BIANCO, G.; BOBOLTZ, D.; BÖCKMANN, S.; BOLOTIN, S.; CHARLOT, P.; COLLIOUD, A.; ENGELHARDT, G.; GIPSON, J.; GONTIER, A.-M.; HEINKELMANN, R.; KURDUBOV, S.; LAMBERT, S.; LYTVYN, S.; MACMILLAN, D.; MALKIN, Z.; NOTHNAGEL, A.; OJHA, R.; SKURIKHINA, E.; SOKOLOVA, J.; SOUCHAY, J.; SOVERS, O.; TESMER, V.; TITOV, O.; WANG, G.; ZHAROV, V. The second realization of the international celestial reference frame by very long baseline interferometry. **The Astronomical Journal**, IOP Publishing, v. 150, n. 2, p. 58, 2015. 173

GENTLE, J. E. **Random number generation and Monte Carlo methods**. [S.l.]: Springer Science & Business Media, 2006. 120

GROVES, P. D. **Principles of GNSS, inertial, and multisensor integrated navigation systems**. [S.l.]: Artech house, 2013. 1, 11, 24, 181

HAMILTON, W. R. **Elements of quaternions**. [S.l.]: Longmans, Green, & Company, 1866. 178

HURD, W. J.; STATMAN, J. I.; VILNROTTER, V. A. High dynamic GPS receiver using maximum likelihood estimationand frequency tracking. **IEEE Transactions on Aerospace and Electronic Systems**, AES-23, n. 4, p. 425–437, July 1987. ISSN 0018-9251. 85

HYSLOP, G.; GERTH, D.; KRAEMER, J. GPS/INS integration on the standoff land attack missile (slam). In: **IEEE Symposium on Position Location and Navigation. A Decade of Excellence in the Navigation Sciences**. [S.l.: s.n.], 1990. p. 407–412. 13

JIA, Y.-B. Quaternions and rotations. **Handout of the Problem Solving Techniques for Applied Computer Science Lecture at Iowa State University**, p. 52, 2008. 178

KAPLAN, G. H. The IAU resolutions on astronomical reference systems, time scales, and earth rotation models. **arXiv preprint astro-ph/0602086**, U.S. Naval Observatory, 2005. 174

KOJA, R.; ARRUDA, J. U. d.; YONEYAMA, T.; GUINAMARD, A. **Algorithms for a MEMs based inertial navigation system.** São José dos Campos: ITA: [s.n.], 2013. (DCTA/ITA/TC-084/2013). Available from: <http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=cat00958a&AN=ita.65342&lang=pt-br&site=eds-live&authtype=ip,cookie,uid>. 39

KOJA, R.; LEITE FILHO, W. C. Revisited position-velocity integration formulas for in-flight self-alignment of GPS-aided inertial navigation systems. In: CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICACÕES, São José do Rio Preto. **Proceedings of DINCON 2017**. São José do Rio Preto, 2017. 4, 5, 7, 20, 30, 163

KUGA, H. K.; MILANI, P. G.; EINWOEGERER, W. Experimentos de alinhamento de unidade de medida inercial baseada em MEMs (micro electro-mechanical systems). In: **Simpósio Brasileiro de Engenharia Inercial**. [S.l.: s.n.], 2007. 12

LEITE, L. K. de C.; KUGA, H. K.; FILHO, W. de C. L. Comparing two methods of coarse self alignment using high accuracy SINS. In: **XXXVIII Ibero-Latin American Congress on Computational Methods in Engineering**. [S.l.: s.n.], 2017. 12

LIMA, E. **Álgebra linear**. Rio de Janeiro, RJ: IMPA, 1995. (Coleção matemática universitária). ISBN 9788524400896. Available from: <https://books.google.com.br/books?id=QKDqtgAACAAJ>. 43, 206

MARKLEY, F. L. Attitude determination using vector observations and the singular value decomposition. **Journal of the Astronautical Sciences**, v. 36, n. 3, p. 245–258, 1988. 11

____. Attitude determination using two vector measurements. In: ILWS WORKSHOP, 1999, Greenbelt, Maryland. **Proceedings of the 1999 Flight Mechanics Symposium**. [S.l.]: National Aeronautics and Space Administration, Goddard Space Flight Center, 1999. p. 39. 9

MARKLEY, F. L.; CRASSIDIS, J. L. **Fundamentals of spacecraft attitude determination and control**. [S.l.]: Springer, 2014. 10, 174

MARKLEY, F. L.; MORTARI, D. Quaternion attitude estimation using vector observations. **Journal of the Astronautical Sciences**, v. 48, n. 2, p. 359–380, 2000. 11

MOOD, A. M. **Introduction to the theory of statistics.** [S.l.]: McGraw-hill, 1950. 187, 201

OSBORNE, P. **The mercator projections**. [S.l.]: Edinburgh, 2008. 176

PEI, F.-J.; LIU, X.; ZHU, L. In-flight alignment using filter for strapdown INS on aircraft. **The Scientific World Journal**, v. 2014, 2014. 2

PSZCZEL, M.; BUCCO, D. **Review of techniques for in-flight transfer alignment**. [S.l.], 1992. 1, 13

RONCAGLIOLO, P. A.; BLASIS, C. E. D.; MURAVCHIK, C. H. GPS digital tracking loops design for high dynamic launching vehicles. In: **2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications**. [S.l.: s.n.], 2006. p. 41–45. ISSN 1943-7439. 85

SAVAGE, P. G. **Strapdown analytics**. Maple Plain, MN: Strapdown Associates, 2000. 178, 179

SHUSTER, M. D.; OH, S. D. Three-axis attitude determination from vector observations. **Journal of Guidance, Control, and Dynamics**, v. 4, n. 1, p. 70–77, 1981. 9, 11

SILVA, F. O.; HEMERLY, E. M.; FILHO, C. W. Error analysis of analytical coarse alignment formulations for stationary SINS. **Transactions on Aerospace and Electronic Systems**, IEEE, v. 52, n. 4, p. 1777–1796, 2016. 10

SOFFEL, M.; KLIONER, S. A.; PETIT, G.; WOLF, P.; KOPEIKIN, S.; BRETAGNON, P.; BRUMBERG, V.; CAPITAINE, N.; DAMOUR, T.; FUKUSHIMA, T. et al. The IAU 2000 resolutions for astrometry, celestial mechanics, and metrology in the relativistic framework: explanatory supplement. **The Astronomical Journal**, IOP Publishing, v. 126, n. 6, p. 2687, 2003. 174

STARK, H.; WOODS, J. **Probability and random processes with applications to signal processing**. Upper Saddle River, NJ: Prentice Hall, 2002. 42, 82, 184, 187, 192, 197

THÉBAULT, E.; FINLAY, C. C.; BEGGAN, C. D.; ALKEN, P.; AUBERT, J.; BARROIS, O.; BERTRAND, F.; BONDAR, T.; BONESS, A.; BROCCO, L. et al. International geomagnetic reference field: the 12th generation. **Earth, Planets and Space**, v. 67, n. 1, p. 79, 2015. 12

TITTERTON, D.; WESTON, J. L. **Strapdown inertial navigation technology**. [S.l.]: IET, 2004. 1, 13, 62, 63, 183

VERMEILLE, H. Direct transformation from geocentric coordinates to geodetic coordinates. **Journal of Geodesy**, v. 76, n. 8, p. 451–454, 2002. 181

WALDMANN, J. Feedforward INS aiding: an investigation of maneuvers for in-flight alignment. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 18, n. 4, p. 459–470, 2007. 13

____. In-flight alignment in INS-aiding with switched feedforward/feedback of error estimates. In: ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA E CIÊNCIAS MECÂNICAS, 19., 2007, Brasília. **Proceedings of The International Congress of Mechanical Engineering**. [S.l.], 2007. 13

WEED, D.; BRODERICK, J.; LOVE, J.; RYNO, T. GPS align in motion of civilian strapdown INS. In: **Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)**. [S.l.: s.n.], 2004. p. 184–192. 13

WHABA, G. A least squares estimate of spacecraft attitude. **SIAM Review**, v. 7, n. 3, p. 409, 1965. 11

WOLFRAM RESEARCH. **Mathematica, Version 11.3**. 2018. Champaign, IL, 2018. 211

WOODWARD, P. M. **Probability and information theory with applications to radar**. [S.l.]: Pergamon, 1953. 22

WU, M.; WU, Y.; HU, X.; HU, D. Optimization-based alignment for inertial navigation systems: theory and algorithm. **Aerospace Science and Technology**, v. 15, n. 1, p. 1–17, 2011. 2, 14

WU, Y.; PAN, X. Velocity/position integration formula part i: application to in-flight coarse alignment. **IEEE Transactions on Aerospace and Electronic Systems**, v. 49, n. 2, p. 1006–1023, 2013. 2, 7, 14, 15, 16, 17, 163

WU, Y.; WANG, J.; HU, D. A new technique for INS/GNSS attitude and parameter estimation using online optimization. **IEEE Transactions Signal Processing**, v. 62, n. 10, p. 2642–2655, 2014. 14

ANNEX A - CONCEPTS FROM NAVIGATION SYSTEMS

This annex reviews basic concepts from the field of Navigation Systems which are well known in the literature, but relevant review in a framework and notation coherent with the rest of this work.

A.1 Reference Frames

For the purpose of this work, a reference frame is defined by an origin $\mathcal{O}$ and a set of linearly independent vectors $\{v_1, v_2, ...v_k\}$ corresponding to a right-handed orthonormal ordered basis in some vector space (CALLIOLI et al., 2007). Thus a reference frame A within $\mathbb{R}^3$ can be determined by the set $\left\{\mathcal{O}_A^A, v_1^A, v_2^A, v_3^A\right\}$ where the upper script stands for the specific frame.

This implies that for any $r \in \mathbb{R}^3$ there exists a unique set of parameters $\left\{x^A, y^A, z^A\right\}$ such that:

$$r = \begin{bmatrix} v_1^A \, v_2^A \, v_3^A \end{bmatrix} \left(\begin{bmatrix} x^A \\ y^A \\ z^A \end{bmatrix} + \mathcal{O}_A^A \right) \tag{A.1}$$

Such parameters are the coordinates of r in the reference frame A. The origin has a lower script to inform that it is the origin of the frame A expressed in A coordinates. For a second reference frame B, the same applies:

$$r = \begin{bmatrix} v_1^B \, v_2^B \, v_3^B \end{bmatrix} \left(\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} + \mathcal{O}_B^B \right) \tag{A.2}$$

By equating both sides, one has:

$$\begin{bmatrix} v_1^A \, v_2^A \, v_3^A \end{bmatrix} \left(\begin{bmatrix} x^A \\ y^A \\ z^A \end{bmatrix} + \mathcal{O}_A^A \right) = \begin{bmatrix} v_1^B \, v_2^B \, v_3^B \end{bmatrix} \left(\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} + \mathcal{O}_B^B \right) \tag{A.3}$$

Therefore:

$$\begin{bmatrix} x^A \\ y^A \\ z^A \end{bmatrix} = \begin{bmatrix} v_1^A \, v_2^A \, v_3^A \end{bmatrix}^T \left\{ \begin{bmatrix} v_1^B \, v_2^B \, v_3^B \end{bmatrix} \left(\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} + \mathcal{O}_B^B \right) \right\} - \mathcal{O}_A^A \tag{A.4}$$

The term corresponding to a matrix combination of scalar products between basis vectors is the Rotation Matrix from frame B to frame A.

$$C_B^A = \begin{bmatrix} v_1^A \, v_2^A \, v_3^A \end{bmatrix}^T \begin{bmatrix} v_1^B \, v_2^B \, v_3^B \end{bmatrix} \tag{A.5}$$

Defining from now on:

r^A The coordinates in A frame:

$$r^A = \begin{bmatrix} x^A \\ y^A \\ z^A \end{bmatrix} \tag{A.6}$$

r^B The coordinates in B frame:

$$r^B = \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} \tag{A.7}$$

$\mathcal{O}_A^A$ The coordinates of the origin of frame A expressed in frame A

$\mathcal{O}_A^B$ The coordinates of the origin of frame A expressed in frame B

$$\mathcal{O}_A^B = C_B^A \mathcal{O}_A^A \tag{A.8}$$

r_{BA}^B The relative position between the origin of frames A and B (subscripts) expressed in coordinates on frame B (superscript).

A.1.1 Body Frame *B*

The Body Frame is defined with specific characteristic by the manufacturer of an Inertial Navigation System or Inertial Measurement Unit. It should be fixed within the instruments' presumably rigid structure. Generally, it is possible to

say that the origin of this frame is fixed and centered on the platform's body, which also contains indications of the axis directions, that in turn are orthogonal. Likewise, the measurements provided by accelerometers and gyros are expressed in this frame

Figure A.1 - Body frame over an INS. Image Courtesy of SBG Systems (2017).

A.1.2 Inertial Frame *I*

An ideal inertial frame of reference is defined in classical physics as one where Newton's laws are applicable (FARRELL, 2008), which requires the frame origin neither to accelerate nor to rotate with respect to the rest of the universe, which are very strong assumptions for practical purposes. An inertial reference system may be realized by listing coordinates for stars which are very far away. The International Celestial Reference Frame (ICRF) defined in (FEY et al., 2015). Its second edition, ICRF2 contains precise positions of 3414 compact radio astronomical

sources. It has a noise floor of 40μas (micro-arcseconds) and an axis stability of 10μas.

The nearest star from the Sun, Proxima Centauri (4.24 light years away), presents a parallax of up to 0.77as, which is the apparent angle deviation of its direction over the Earth's orbit. Some star catalogs only include stars at distances greater than 100 light years from the solar system, so that their parallax is less than 0.03as near the Earth (MARKLEY; CRASSIDIS, 2014).

Accelerometer and gyro measurements are expressed with coordinates in Body frame, yet they ideally measure rates with respect to an Inertial frame.

While the ICRF frame origin is located at the barycenter of the solar system, for most terrestrial applications it suffices to consider an inertial frame centered at the barycenter of the Earth (DECKER, 1986) thus, being usually called Earth-Centered Inertial (ECI).

Neither the ICRF nor the ECI frames are truly inertial frames, yet the ECI is sufficient for the navigation of both Earth-bound vehicles and Earth orbiting satellites (MARKLEY; CRASSIDIS, 2014) (FARRELL, 2008).

Another distinct approach adopted by the International Astronomical Union is to define distinct reference systems, the Barycentric Celestial Reference System (BCRS) and the Geocentric Celestial Reference System (GCRS), one centered at the barycenter of the solar system and another on the geocenter. Further details are provided on (BRUMBERG; GROTEN, 2001), (SOFFEL et al., 2003) and (KAPLAN, 2005).

A.1.3 Earth Frame *E*

The Earth frame, also known as the Terrestrial frame, is fixed on the Earth. It will be defined in this work alike to (DECKER, 1986) as having:

- Origin at the Earth's barycenter.
- Z-axis parallel to the direction of the Conventional International Origin for polar motion.
- X-axis at the intersection of WGS 84 reference meridian plane and the plane of the mean astronomic equator.

- Y-axis completing a right-handed orthonormal basis.

Notably, the reference meridian is parallel to the zero meridian defined by the *Bureau International de I'Heure*. This reference frame is that of a mean or standard earth rotation direction at a constant rate of $\omega_{EI} = 7.292115 \times 10^{-5} \text{rad s}^{-1}$ around a mean astronomic pole (the CIO) fixed in time.

Figure A.2 - ECI and ECEF Frames. Circumflexes in this figure indicate unit vectors.

A.1.4 World Geodetic System

The World Geodetic System, developed by the Defense Mapping Agency of the United States (DECKER, 1986), often referred as "WGS84" for the update published in 1984, provides a coordinate system on an ellipsoid approximation of the Earth. GPS coordinates are often given in the form of Geographic or Geodetic coordinates. Which are:

ϕ : (geodetic) Latitude

λ : Longitude

h : Altitude (above ellipsoid).

They may be used to obtain Terrestrial frame coordinates with:

$$x = (R_N + h)\cos(\phi)\cos(\lambda) \tag{A.9a}$$

$$y = (R_N + h)\cos(\phi)\sin(\lambda) \tag{A.9b}$$

$$z = \left(R_N(1 - e^2) + h\right)\sin(\phi) \tag{A.9c}$$

Definition A.1.1. Normal Radius:
Where R_N is the Normal Radius, prime vertical radius of curvature or great normal, defined as a function of latitude by:

$$R_N = \frac{a}{\sqrt{1 - e^2\sin(\phi)}} \tag{A.10}$$

Definition A.1.2. Meridian Radius:
R_M, the Meridian Radius, or ellipse radius, is also defined as a function of latitude by:

$$R_M = \frac{a\left(1 - e^2\right)}{\left(1 - e^2\sin(\phi)\right)^{\frac{3}{2}}} \tag{A.11}$$

This quantity is used for convenience is some transformations but is shown on (OSBORNE, 2008) to correspond to the inverse of the local curvature along the meridian line of the ellipsoid surface.

WGS84 also contains the Earth Gravitational model, based on spherical harmonic expansion and comprising 32755 geopotential coefficients. A simpler equation for the theoretical normal gravity γ as a function of latitude on the surface of the WGS84 Ellipsoid is also provided (DECKER, 1986):

$$\gamma(\phi) = \frac{9.7803267714\left(1 + 0.001931851386\sin\left(\phi\right)\right)}{\sqrt{1 - 0.00669437999013\sin^2\left(\phi\right)}}\mathrm{m/s^2} \tag{A.12}$$

A.1.5 Navigation Frame N

Also known as the local level navigation frame, this is the reference frame in which the outputs of the INS are generally provided.

- Origin coinciding with the origin of the Body frame

- Z-axis normal to the WGS84 ellipsoid surface pointing down.
- X-axis tangent to the ellipsoid, pointing towards the local North
- Y-axis completing a right-handed triad.

Figure A.3 - NED Frame.

The Transformation of frames between Terrestrial and Navigation frame can be performed by (FARRELL, 2008):

$$C_E^N = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\phi)\cos(\lambda) & -\cos(\phi)\sin(\lambda) & -\sin(\phi) \end{bmatrix} \tag{A.13}$$

A.2 Quaternions

Quaternions are used in the implemented code, as they provide a convenient method to parametrize attitude, since they have linear of the quaternion differential equations, no singularities, do not require usage of trigonometric functions

(unlike Euler angles), and are composed of only four parameters against 9 used in a rotation matrix (FARRELL, 2008). They were first introduced by (HAMILTON, 1866), and are expressed as an extension to the complex numbers:

$$q = q_0 + q_1\, i + q_2\, j + q_3\, k \tag{A.14}$$

Where q_0 is called the scalar part, and the other terms are referred as the imaginary or vector part, also denoted as:

$$q = q_0 + \overrightarrow{q} \tag{A.15}$$

A vector in $\mathbb{R}^3$ can be expressed as a quaternion with zero scalar part.

Within the code, it is simply stored as a four-dimensional array:

$$q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{A.16}$$

Let q and p denote quaternions, its multiplication rule, as defined by Hamilton, but with modern notation (JIA, 2008) is expressed as:

$$p \otimes q = p_0\, q_0 - \overrightarrow{p} \,.\, \overrightarrow{q} + p_0\, \overrightarrow{q} + q_0\, \overrightarrow{p} + \overrightarrow{p} \times \overrightarrow{q} \tag{A.17}$$

A vector $\overleftarrow{u}$ can be rotated around an unitary vector $\overrightarrow{v}$ by an angle of θ by the rotation operator defined as (JIA, 2008):

$$L_q(\overrightarrow{u}) = q \otimes u \otimes q^* \tag{A.18}$$

With the quaternion q composed by:

$$q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) . \overrightarrow{v} \tag{A.19}$$

Following the conventions from (SAVAGE, 2000), let u^A be a vector with coordinates expressed in frame A and u^B the same vector expressed in B frame. The

quaternion that relates frame A to frame B is denoted as q_B^A, and applying the operator from Equation (A.18), the rotation works as follows:

$$u^A = q_B^A \otimes u^B \otimes (q_B^A)^* \tag{A.20}$$

Likewise, the multiplication of rotation quaternions admits the following chain rule (SAVAGE, 2000):

$$q_C^A = q_B^A \otimes q_C^B \tag{A.21}$$

And it is also possible compute the associated rotation matrix with (SAVAGE, 2000):

$$C_B^A = C(q_B^A) = \left(q_0^2 - \overrightarrow{q}.\overrightarrow{v}\right) + 2\,\overrightarrow{v}\,\overrightarrow{v}^T + 2\,q_0\,\overrightarrow{v}^\times \tag{A.22}$$

A.3 Basic Sensor Models

Sensors are all prone to measurement errors, which ultimately interfere with the performance of the system they compose. This section explains the basic models that establish standard structures to those errors, such that they can be compensated by means of filtering or parameter estimation.

A.3.1 Accelerometer

An accelerometer measures specific acceleration over its sensitive axes. It is prone to several forms of errors, including misalignment and scale factor deviation, which generally are dealt in a unit-specific basis by means of calibration. The most significant contributors to the measurement errors in a high quality inertial sensor are the random zero mean noise represented by $\varepsilon\left(f^B\right)$ and the bias η.

The measurement denoted with an over-hat is modeled by:

$$\widehat{f^B} = f^B + \eta + \varepsilon\left(f^B\right) \tag{A.23}$$

The actual physical magnitude can be isolated by:

$$f^B = \widehat{f^B} - \eta - \varepsilon\left(f^B\right) \tag{A.24}$$

Since the measurements are periodic samples, they are to be treated as a sequence for most developments:

$$f_k^B = \widehat{f_k^B} - \eta - \ \varepsilon_k \left(f^B\right) \tag{A.25}$$

It may also be assumed that the IMU provides a velocity increment Δv^B. For long simulations, the bias η should be assumed variable, having a random walk of its own, but with small deviations over time for high quality instruments.

The characteristics assumed for η and ε_k are respectively described in Definition 3.3.2 and Definition 3.3.1, after the proper terminology and apparatus are presented.

A.3.2 Gyrometer

The model assumed for the angular rate sensor is analogous to the accelerometer's. With $\varepsilon\left(\omega_{IB}^B\right)$ denoting a Noise Type Error and μ a Bias Type Error (Definition 3.3.2 and Definition 3.3.1 respectively).

$$\widehat{\omega_{IB}^B} = \omega_{IB}^B + \mu + \varepsilon\left(\omega_{IB}^B\right) \tag{A.26}$$

$$\omega_{IB}^B = \widehat{\omega_{IB}^B} - \mu - \varepsilon\left(\omega_{IB}^B\right) \tag{A.27}$$

$$\left(\omega_{IB}^B\right)_k = \widehat{\left(\omega_{IB}^B\right)_k} - \mu - \varepsilon_k\left(\omega_{IB}^B\right) \tag{A.28}$$

It may also be assumed that the IMU provides a small angle increment $\Delta\Theta$. Once more, for long simulations, the bias μ should be assumed as slowly varying.

A.3.3 GPS

The Global Positioning System (GPS) is the most famous example of a Global Navigation Satellite System (GNSS), which employs a satellite network that broadcasts positioning signals. Upon contact with at least 4 of such satellites, it is possible to determine the position and velocity of the receiver antenna. The greatest advantage of such system is that it provides absolute rather than relative

measurements, in the sense that position is an absolute measurement, while velocity is a rate that can be integrated into position variation, while the position would remain unknown if no initial value was available and furthermore, the integration process using noisy measurements would introduce an error whose variance would grow with time, as shown in Chapter 3. An absolute measurement, however, has bounded errors at any time.

The error model employed for the purpose of this work assumes uncorrelated noise between measurements i.e., a Noise Type Bias denoted by $\varepsilon\left(r^E\right)$ and $\varepsilon\left(v^N\right)$ for the position and velocity measurements, as follows:

$$\widehat{r^E} = r^E + \varepsilon\left(r^E\right) \tag{A.29}$$

$$\widehat{v^N} = v^N + \varepsilon\left(v^N\right) \tag{A.30}$$

Much more complex models for errors, usually focused on their sources are discussed in the literature, such as in (FARRELL, 2008). This is relevant when developing GPS receivers, as for instance the standard GPS user range error is expected to be in the order of 8 to 11 meters, while differential operation yields 10 centimeters to 3 meter error for differential pseudorange by compensating some of the intervening effects. It is however understood to be reasonable to sustain the assumption of uncorrelated error for the user delivered measurement (FARRELL, 2008).

Furthermore, while there are complex models in the literature for characterizing the uncertainty of GPS based position and velocity fixes, and generally they are known as "Dilution of Precision" (GROVES, 2013), the developments presented in Chapter 3 and simulations discussed in Chapter 4, it is assumed a constant covariance matrix for the error $\varepsilon\left(v^e\right)$ at each iteration. The error in position itself is of lesser relevance because it will basically be used to compute the local gravity g^N.

Despite the formulation shown in Equation (A.29), GPS receivers often provide measurements in geodetic coordinates of latitude (ϕ), longitude (λ) and altitude (h). These can be translated if needed to Earth frame coordinates with either iterative methods such as the ones presented in (FARRELL, 2008) or exact closed form algebraic methods such as the one presented in (VERMEILLE, 2002).

A.4 Numerical Integration

Given a set of equations of the form:

$$\begin{cases} x(0) = x_0 \\ \dot{x} = f(t, x) \end{cases} \tag{A.31}$$

It is often impossible to find analytical solutions for $x(t)$, but is possible to select a set of points t_k and numerically evaluate an approximation for the value of $x(t_k)$. The most commonly used algorithm for such task is the Runge-Kutta of forth order (CARNAHAN et al., 1969), which uses a fixed interval between consecutive point t_k known as the step size h:

$$x_{k+1} = x_k + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{A.32a}$$

$$t_{k+1} = t_k + h \tag{A.32b}$$

With:

$$k_1 = f(t_k, x_k) \tag{A.33a}$$

$$k_2 = f\left(t_k + \frac{h}{2}, x_k + h\frac{k_1}{2}\right) \tag{A.33b}$$

$$k_3 = f\left(t_k + \frac{h}{2}, x_k + h\frac{k_2}{2}\right) \tag{A.33c}$$

$$k_4 = f\left(t_k + h, x_k + hk_3\right) \tag{A.33d}$$

Such method re-evaluates the function f at intermediate points over the interval $[t_k, t_{k+1}]$.

Accurate integration methods are paramount when:

- The interval of integration is long, but finite
- The initial value x_0 is known
- The system plant $f(t, x)$ is well known
- The system is not subject to perturbation

None of these conditions are true when dealing with estimations or with a navigation solution, because in this case:

- The interval of integration is indefinitely long, yet recomputed at every small cycle
- The initial value x_0 is unknown
- The system plant $f(t, x)$ is subject to unknown interference
- The system is subject to perturbations, including but not limited to noise.

Despite this, Runge-Kutta methods are used in inertial systems (TITTERTON; WESTON, 2004) with demanding applications.

However, a set of simpler methods have a remarkable property. Take for example, Euler's method:

$$x_{k+1} = x_k + hf(t_k, x_k) \tag{A.34}$$

As another example that is used in inertial systems is Simpson's rule (TITTERTON; WESTON, 2004), in which case the system plant is assumed as depending on time, but not on the state:

$$x_{k+1} = x_k + \frac{h}{3}f(t_{k-1}) + \frac{4h}{3}f(t_k) + \frac{h}{3}f(t_{k+1}, x_k) \tag{A.35}$$

In both cases, the value of x_{k+1} can be written as a summation, in Euler's case:

$$x_{k+1} = x_0 + \sum_{k=1}^{k} hf(t_k, x_k) \tag{A.36}$$

For Simpson's rule, and in a more general case:

$$x_{k+1} = x_0 + \sum_{k=1}^{k} w_k f_k \tag{A.37}$$

Where the argument of the function f is omitted and replaced by the index k. For different methods, the form above is the same with variation of the weighting coefficients w_k. While such form is of little practical utility, it will help the developments of Chapter 3.

A.5 The Extended Kalman Filter

The Extended Kalman Filter (EKF) is by far the most used tool for integrating IMU and GPS (CRASSIDIS, 2006). It can be regarded as an extension of the Linear

Kalman Filter, that is an optimal state observer for linear systems (STARK; WOODS, 2002). It is described here to lay basis for some comments and considerations made through the text.

Let two random process X and Z have their realizations governed by:

$$\dot{X} = f(X, u) + \varepsilon \tag{A.38a}$$

$$Z = h(X, u) + \xi \tag{A.38b}$$

With u a known control input. ε is called the process noise and ξ the measurement noise.

A discretized estimation $\widehat{x}_k$ for the state X, at time steps Δt is to be computed given the measurements z_k which are known realizations of Z at each step k, with future realizations being unknown. The EKF is initialized with an *a priori* estimation $\hat{x}_0$, and an *a priori* variance P_0^-.

The state and measurement functions are assumed known, computable but not linear. Their partial derivatives are given by:

$$F_k = \frac{\partial f}{\partial x}\Big|_{\widehat{x}_k} \tag{A.39a}$$

$$H_k = \frac{\partial h}{\partial x}\Big|_{\widehat{x}_k} \tag{A.39b}$$

The state at the next step is predicted based on the previous state, using some numerical method as described on Section A.4.

$$\widehat{x}_{k+1}^- = \int_{\Delta t} f(\widehat{x}_k^+, u)\, dt \tag{A.40}$$

The transition matrix Φ_k is computed with either Equation (A.41) or some approximation to it.

$$\Phi_k = e^{F_k \Delta t} \tag{A.41}$$

The *a priori* covariance is computed based on linearized equations:

$$P_{k+1}^- = \Phi_k^T P_k^+ \Phi_k + \sigma_\varepsilon^2 \tag{A.42}$$

Where σ_ε^2 often denoted as Q_k is called the process covariance.

The measurement prediction is computed:

$$\widehat{z}_{k+1} = h(\widehat{x}_{k+1}^{-}) \tag{A.43}$$

The innovation covariance is given by:

$$E_k = H_k \, P_k^{-} \, H_k^T + \sigma_{\tilde{\varsigma}}^2 \tag{A.44}$$

Where $\sigma_{\tilde{\varsigma}}^2$, often denoted as R_k is the measurement (or observation) covariance.

Then the Kalman gain vector will be:

$$K_{k+1} = P_{k+1}^{-} \, H_{k+1}^T \, E_{k+1}^{-1} \tag{A.45}$$

And the *a posteriori* state estimate is:

$$\widehat{x}_{k+1}^{+} = \widehat{x}_{k+1}^{-} + K_{k+1} \, (z_{k+1} - \widehat{z}_{k+1}) \tag{A.46}$$

Finally, the *a posteriori* covariance is given by:

$$P_{k+1}^{+} = (I - K_{k+1} \, H_{k+1}) \, P_{k+1}^{-} \tag{A.47}$$

Typically, the *a priori* variance P_0^{-} is chosen to be relatively large to allow the first measurements to play an important part in having $\widehat{x}_k$ approach to the real state value. While in linear systems without time varying parameters the EKF reaches a steady state where the matrix P_k^{+} becomes constant over time, this may not happen in non-linear estimation since the observability characteristics of the system may vary over time. Also, the *a posteriori* state estimate is basically a correction that is expected to improve accuracy in a linear vicinity of the true value. Being far away from such linear region may cause the EKF to diverge (CRASSIDIS, 2006).

ANNEX B - CONCEPTS FROM STATISTICS AND STOCHASTIC SYSTEMS

This annex reviews basic concepts on random variables and stochastic processes that are needed to build the online error estimation of the INS alignment. The notation and concepts are mostly taken from (MOOD, 1950) and (STARK; WOODS, 2002). It is recalled that on section Section 3.6 contributions of this work are presented, nominally the recursive formulas for a set of summations involving random variables.

B.1 Definitions and Theorems from Random Variables Theory

B.1.1 Sets, Samples and Events

Definition B.1.1. Set:
A **set** is a collection of abstract and or concrete objects. Each of those will be called a **point** or **element**. As an example of notation, ζ may be an element contained in the set Ω.
Definition B.1.2. Subset:
If for every element ζ contained in a set Ω ($\zeta \in \Omega$), ζ is also contained in Ψ ($\zeta \in \Psi$), then Ω is a **subset** of Ψ, which is denoted as:

$$\Omega \subseteq \Psi \tag{B.1}$$

If the possibility of $\Psi = \Omega$ can be discarded, the notation may be presented as follows:

$$\Omega \subset \Psi \tag{B.2}$$

Definition B.1.3. Disjoint Sets:
Two sets A and B are said to be **disjoint** if there exists no element x such that belongs to both. This can be expressed as:

$$A \cap B = \emptyset \tag{B.3}$$

Definition B.1.4. Sample Space:
The set containing all possible outcomes of a conceptual experiment is called **sample space** or alternatively a **description space**, and is generally denoted by Ω.
Definition B.1.5. Event:
For a given sample space Ω, every subset $A \in \Omega$ is called an event.
Definition B.1.6. Event Space:
Given a sample space Ω, let A be an event and $\mathcal{A}$ be a sample space satisfying the

following properties:

a) $\Omega \in \mathcal{A}$

b) If $A \in \mathcal{A}$ then $\bar{A} \in \mathcal{A}$

c) If $A_1 \in \mathcal{A}$ and $A_2 \in \mathcal{A}$ then $(A_1 \cup A_2) \in \mathcal{A}$

Then $\mathcal{A}$ is called an **Event Space** or alternatively an Experiment.

B.1.2 Basics of Probability Theory

Definition B.1.7. Probability Function:
A function $P : \mathcal{A} \to [0, 1]$ satisfying the following properties:

a) $P[A] \geq 0$ for every $A \in \mathcal{A}$.

b) $P[\Omega] = 1$.

c) If $A_1, A_2, \cdots$ is a sequence of mutually exclusive events in $\mathcal{A}$ such that $(\bigcup_{i=1}^{\infty} A_i) \in \mathcal{A}$ then $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P\left(A_i\right)$

Then P is called a **probability function** or alternatively a **probability measure**.

It is noted that item c requires sets A_i and A_j to be disjoint for all $i \neq j$.
Definition B.1.8. Independent Events:
Two events A and B are said to be independent if and only if:

$$P[A \cup B] = P[A]\, P[B] \tag{B.4}$$

Definition B.1.9. Conditional Probability:
Given two events $A, B \in \mathcal{A}$ with $B \neq \emptyset$, the conditional probability of A given B is denoted $P[A|B]$, and can be computed by:

$$P[A|B] = \frac{P[A \cup B]}{P[B]} \tag{B.5}$$

B.1.3 Basics of Random Variable Theory

Definition B.1.10. Random Variable:
Let $\mathcal{A}$ be an experiment with sample space Ω, then a function $X : \mathcal{A} \to \Omega$ satisfying:

a) For every Borel set of numbers B, the set $\{\zeta \in \Omega, X(\zeta) \in B\}$ is an event.

b) $P[X = \infty] = P[X = -\infty] = 0$.

Naturally, a point x such that $P[X = x] = 0$ is an impossible event.

With this definition, an event ζ can be represented by $\{\zeta : X(\zeta) \leq x\}$, which maps the event ζ to the numbers $]-\infty, x]$
Definition B.1.11. Cumulative Distribution Function:
Given a random variable X, its **Cumulative Distribution Function** (CDF), is defined as:

$$F_X(x) = P\left(\zeta : X(\zeta) \leq x\right) = P_X\left(\,]-\infty, x]\right) \tag{B.6}$$

Definition B.1.12. Probability Density Function:
Given a random Variable X, a function $f_X : \mathbb{R} \to \mathbb{R}$ satisfying:

$$\int_{-\infty}^{x} f_X(x)dx = F_X(x) \tag{B.7}$$

Is called the probability density function of the random variable X. It is noted that, axiomatically, any function satisfying:

a) $(\forall x) \to f(x) \geq 0$

b) $\int_{-\infty}^{\infty} f_X(x)dx = 1$

Is a **Probability Density Function** (PDF).
Definition B.1.13. Joint Cumulative distribution Function:
Let $X_1, X_2, \cdots, X_k$ be k random variables all defined on the same probability space $(\Omega, \mathcal{A}, P)$. Their **joint cumulative distribution function** $F_{X_1,X_2,\cdots,X_k}$ is defined as:

$$F_{X_1,X_2,\cdots,X_k}(x_1, x_2, \cdots, x_k) = P\left(X_1 \leq x_1, X_2 \leq x_2, \cdots, X_k \leq x_k\right) \tag{B.8}$$

For multivariate X, the notation is simplified to F_X, and the word "joint" is often omitted.

Definition B.1.14. Joint Density Function:
Let $X_1, X_2, \cdots, X_k$ be k random variables all defined on the same probability space $(\Omega, \mathcal{A}, P)$. A function satisfying:

$$F_{X_1,X_2,\cdots,X_k}(x_1, x_2, \cdots, x_k) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \cdots \int_{-\infty}^{x_k} f_{X_1,X_2,\cdots,X_k}(x_1, x_2, \cdots, x_k) dx_1 dx_2 \cdots dx_k \tag{B.9}$$

Is their **Joint Cumulative Density Function** $f_{X_1,X_2,\cdots,X_k}(x_1, x_2, \cdots, x_k)$.
Definition B.1.15. Conditional Density Function:
Let X and Y be jointly random variables with joint probability density function $f_{XY}(x, y)$. The conditional probability of X given $Y = y$ is:

$$f_{X|Y}(x|y) = \frac{f_{XY}(x, y)}{f_Y(y)} \tag{B.10}$$

An important consequence of this definition is:

$$f_{X|Y}(x|y) f_Y(y) = f_{XY}(x, y) \tag{B.11}$$

B.1.4 Expectation

Definition B.1.16. Expectation:
The value $\mu \in \mathbb{R}^k$ or $\bar{X}$ provided by:

$$\mu = \int_{\mathbb{R}^k} x f_X(x) dx \tag{B.12}$$

Is defined as the **Expectation** of X or the **Expected Value** of X, it is also generally denoted as $E[X]$ or μ_X when needed to make the underlying random variable explicit.

It is an unfortunate notation coincidence that the greek letter mu is used both for the expectation of a random variable and for a bias type error. It will be defined in each context and also the associated random variable will be made explicit with a subscript whenever necessary.
Definition B.1.17. General Expectation:
Given a random variable X and a generic function $g : \mathbb{R}^k \to^n$, the value given by:

$$E[g(X)] = \int_{\mathbb{R}^k} g(x) f_X(x) dx \tag{B.13}$$

is called the **Expectation** or **Expected Value** of $g(X)$.

Theorem B.1.1. Linearity of Expectation:

For two scalars a and b and two functions $g : \mathbb{R}^k \to^n$ and $h : \mathbb{R}^k \to^n$, the following identity holds:

$$E\left[ag(X) + bh(X)\right] = aE\left[g(X)\right] + bE\left[h(X)\right] \tag{B.14}$$

This is a direct consequence of the linearity of the integral operator. Noting for the involved terms that:

$$E\left[ag(X) + bh(X)\right] = \int_{\mathbb{R}} f_X(x)\left(ag(x) + bh(x)\right)dx \tag{B.15}$$

Definition B.1.18. Conditional Expectation:

Let X and Y be a k-dimensional and a p-dimensional random variable respectively and $g : \mathbb{R}^{k+p} \to \mathbb{R}^m$, then, the conditional expectation of $g(X, Y)$ given $Y = y$ is defined as:

$$E[g(X, Y)|Y = y] = \int_{\mathbb{R}^k} g(x, y) f_{X|Y}(x|y)dx \tag{B.16}$$

Theorem B.1.2. Stacked Expectation:

Let X and Y be a k-dimensional and a p-dimensional random variable respectively and $g : \mathbb{R}^{k+p} \to \mathbb{R}^m$ then, the expectation of $g(X, Y)$ can be computed by:

$$E[g(X, Y)] = E\left[E\left[g(X, Y)|Y\right]\right] \tag{B.17}$$

Proof: Applying the definition of Expectancy (Definition B.1.16) on the outer expectation:

$$E\left[E\left[g(X, Y)|Y\right]\right] = \int_{\mathbb{R}^p} E\left[g(X, Y|Y)\right] f_Y(y)dy \tag{B.18}$$

Applying the definition of conditional expectation (Definition B.1.18):

$$E\left[E\left[g(X, Y)|Y\right]\right] = \int_{\mathbb{R}^p} \left(\int_{\mathbb{R}^k} g(x, y) f_{X|Y}(x|y)dx\right) f_Y(y)dy \tag{B.19}$$

Reorganizing the integrals:

$$E\left[E\left[g(X, Y)|Y\right]\right] = \int_{\mathbb{R}^p} \int_{\mathbb{R}^k} g(x, y)\left(f_{X|Y}(x|y) f_Y(y)\right)dxdy \tag{B.20}$$

Recalling Equation (B.11), that derives from Definition B.1.15:

$$E\left[E\left[g(X,Y)|Y\right]\right] = \int_{\mathbb{R}^p}\int_{\mathbb{R}^k} g(x,y)f_{XY}(x,y)dxdy \tag{B.21}$$

And naturally:

$$\int_{\mathbb{R}^p}\int_{\mathbb{R}^k} g(x,y)f_{XY}(x,y)dxdy = E\left[g(X,Y)\right] \tag{B.22}$$

Therefore:

$$E[g(X,Y)] \ = E\left[E\left[g(X,Y)|Y\right]\right] \tag{B.23}$$

Thus concluding the proof. This result is also known as the "Total Expectation" theorem.

B.1.5 Variance and Related Terms

Definition B.1.19. Cross-Covariance:
Let X and Y be two k-dimensional random variables defined in the same probability space. The matrix $K_{X,Y} \in \mathcal{M}_{k\times k}(\mathbb{R})$ provided by:

$$K_{XY} = Cov[X,Y] = \sigma_{X,Y} = E\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)^T\right] \tag{B.24}$$

Is defined as the **Covariance** or **Cross-Covariance** of X and Y. It is also known in the literature are the **Correlation** of X and Y.
Definition B.1.20. Variance:
The matrix $K \in \mathcal{M}_{k\times k}(\mathbb{R})$ provided by:

$$K = E[(X-\mu)(X-\mu)^T] = K_{XX} = Cov[X,X] = \sigma_X^2 \tag{B.25}$$

$$K = \int_{\mathbb{R}^k}(x-\mu)(x-\mu)^T f_X(x)dx \tag{B.26}$$

Is defined as the **Autocovariance** of the random variable X. Throughout this work, it will be generally referred to as the **Variance**. It is sometimes known in the literature as the Covariance (STARK; WOODS, 2002) of X and Y.

It is possible to show that there always exists a matrix σ such that $\sigma\sigma = \sigma^2 = K$, therefore the superscript does not cause notation inconsistency.
Definition B.1.21. Standard Deviation:
Let X be a one dimensional random variable, with variance σ^2, the value $\sqrt{\sigma^2} = \sigma \in \mathbb{R}^+$ is called the standard deviation of X.

It is noted that no direct equivalent of the standard deviation for multi-dimensional variables is widespread in the literature

If the cross-covariance refers to random sequences, it is also useful to make the relevant time index explicit:

$$K_{XY}[m,n] = Cov[X_n, Y_m] = E\left[\left(X_n - \mu_X[n]\right)\left(Y_m - \mu_Y[m]\right)^T\right] \tag{B.27}$$

Theorem B.1.3. Variance of a Sum:
Let X and Y be two k-dimensional random variables, and $A, B \in \mathcal{M}(\mathbb{R})_{k\times k}$ Then:

$$Var\left[AX+BY\right] = AVar[X]A^T + ACov[X,Y]B^T + BCov[Y,X]A^T + BVar[Y]B^T \tag{B.28}$$

Proof:

$$Var\left[AX+BY\right] = E\left[\left(AX+BY\right)\left(AX+BY\right)^T\right] \tag{B.29}$$

$$Var\left[AX+BY\right] = E\left[AX\left(AX+BY\right)^T + BY\left(AX+BY\right)^T\right] \tag{B.30}$$

Using distribution of transposition (Theorem C.1.1 and Theorem C.1.2):

$$Var\left[AX+BY\right] = E\left[AX\left(X^TA^T + Y^TB^T\right) + BY\left(X^TA^T + Y^TB^T\right)\right] \tag{B.31}$$

$$Var\left[AX+BY\right] = E\left[AXX^TA^T + AXY^TB^T + BYX^TA^T + BYY^TB^T\right] \tag{B.32}$$

Using Linearity of Expectation (Theorem B.1.1):

$$Var\left[AX+BY\right] = E\left[AXX^TA^T\right] + E\left[AXY^TB^T\right] + E\left[BYX^TA^T\right] + E\left[BYY^TB^T\right] \tag{B.33}$$

Again, using linearity:

$$Var\left[AX+BY\right] = AE\left[XX^T\right]A^T + AE\left[XY^T\right]B^T + BE\left[YX^T\right]A^T + BE\left[YY^T\right]B^T \tag{B.34}$$

And from the definitions of Variance and Covariance (Definition B.1.20,Definition B.1.19):

$$Var\left[AX+BY\right] = AVar[X]A^T + ACov[X,Y]B^T + BCov[Y,X]A^T + BVar[Y]B^T \quad \text{(B.35)}$$

Which concludes the proof.

Theorem B.1.4. Corollary for Single Random Variable:
A direct consequence of the Theorem B.1.3, is:

$$Var\left[AX\right] = AVar[X]A^T \quad \text{(B.36)}$$

Theorem B.1.5. Covariance matrices are symmetric:
This is restricted to autocovariance and applicable at same time index, and follows directly from the definition:

$$K = E[(X-\mu)(X-\mu)^T] \quad \text{(B.37)}$$

$$K^T = \left(E\left[(X-\mu)(X-\mu)^T\right]\right)^T = E\left[\left((X-\mu)(X-\mu)^T\right)^T\right] \quad \text{(B.38)}$$

$$K^T = \left(E\left[(X-\mu)(X-\mu)^T\right]\right)^T = E\left[\left((X-\mu)^T\right)^T(X-\mu)^T\right] \quad \text{(B.39)}$$

$$K^T = E\left[(X-\mu)(X-\mu)^T\right] \quad \text{(B.40)}$$

Therefore:

$$K^T = K \quad \text{(B.41)}$$

Theorem B.1.6. Covariance matrices are positive semi-definite:
Any matrix A satisfying:

$$v^T Av \geq 0 \, \forall v \neq \mathcal{O} \quad \text{(B.42)}$$

Is called positive semi-definite. Noting that:

$$v^T Kv = v^T E[(X-\mu)(X-\mu)^T]v = E\left[v^T(X-\mu)(X-\mu)^T v\right] \quad \text{(B.43)}$$

Also noting:

$$(X-\mu)^T v = v^T(X-\mu) = \sum_i ((X-\mu)_i)v_i = a \quad \text{(B.44)}$$

Then:

$$v^T K v = E[a^2] \geq 0 \tag{B.45}$$

Theorem B.1.7. Identity with Expectancy and Variance:
Given a random unidimensional variable X with variance σ_X^2 and mean μ_X, the following identity holds:

$$Var[X] = E[X^2] - E[X]^2 \tag{B.46}$$

Or with different notation:

$$\sigma_X^2 = E[X^2] - \mu_X^2 \tag{B.47}$$

Theorem B.1.8. Identity with Expectancy and Variance (Multivariate Case):
Given an n-dimensional random variable X with variance σ_X^2 and mean μ_X, the following identity holds:

$$Var[X] = E[XX^T] - E[X]E[X]^T \tag{B.48}$$

Or with different notation:

$$\sigma_X^2 = E[XX^T] - \mu_X \mu_X^T \tag{B.49}$$

Theorem B.1.9. Distribution of Independent Random Variables:
Both density and cumulative probability joint distributions of two independent random variables X and Y can be factored into the individual distributions. Recalling Definition B.1.8:

$$P\left[(X \leq x) \wedge \left(Y \leq y\right)\right] \ = P\left[X \leq x\right] P\left[Y \leq y\right] \tag{B.50}$$

From Definition B.1.11:

$$F_{XY}(x, y) = F_X(x) F_Y(y) \tag{B.51}$$

Recalling as well:

$$\int_{-\infty}^{x} \int_{-\infty}^{y} f_{XY}(x, y) dx dy = \int_{-\infty}^{x} f_X(x) dx \int_{-\infty}^{y} f_Y(y) dy \tag{B.52}$$

Therefore:

$$f_{XY}(x, y) = f_X(x) f_Y(y) \tag{B.53}$$

Theorem B.1.10. Expectancy of Independent Variables (Product Case):
Let X be a k-dimensional and Y be a p-dimensional independent random variable. Then the expectation of a function $g : \mathbb{R}^{k+p} \to \mathbb{R}^l$ that assumes the form $g(X, Y) = r(X)s(Y)$ can be computed by:

$$E[g(X, Y)] = E[r(X)s(Y)] \tag{B.54}$$

$$\int_{\mathbb{R}^{k+p}} g(x, y) f_{XY}(x, y) dx dy = \int_{\mathbb{R}^{k+p}} r(x)s(y) f_{XY}(x, y) dx dy \tag{B.55}$$

From Theorem B.1.9:

$$\int_{\mathbb{R}^{k+p}} g(x, y) f_{XY}(x, y) dx dy = \int_{\mathbb{R}^{k+p}} r(x)s(y) f_X(x) f_Y(y) dx dy \tag{B.56}$$

$$\int_{\mathbb{R}^{k+p}} g(x, y) f_{XY}(x, y) dx dy = \int_{\mathbb{R}^p} \int_{\mathbb{R}^k} r(x) f_X(x) s(y) f_Y(y) dx dy \tag{B.57}$$

$$\int_{\mathbb{R}^{k+p}} g(x, y) f_{XY}(x, y) dx dy = \int_{\mathbb{R}^k} r(x) f_X(x) dx \int_{\mathbb{R}^p} s(y) f_Y(y) dy \tag{B.58}$$

Therefore:

$$E[r(X)s(Y)] \ = E[r(X)]E[s(Y)] \tag{B.59}$$

Thus, over products of functions of independent random variables, the expectancy operator admits distribution. This is particularly useful and convenient when either expectation is zero, as is often used on Section 3.7.1.

Theorem B.1.11. Expectancy of Independent Variables (Sum Case):
Let X be a k-dimensional random variable and Y be a p-dimensional independent random variable. Then the expectation of a function $g : \mathbb{R}^{k+p} \to \mathbb{R}^l$ that assumes the form $g(X, Y) = r(X) + s(Y)$ can be computed by:

$$E[g(X, Y)] = E[r(X) + s(Y)] \tag{B.60}$$

$$\int_{\mathbb{R}^{k+p}} g(x,y) f_{XY}(x,y) dxdy = \int_{\mathbb{R}^{k+p}} (r(x) + s(y)) f_{XY}(x,y) dxdy \tag{B.61}$$

From Theorem B.1.9:

$$\int_{\mathbb{R}^{k+p}} g(x,y) f_{XY}(x,y) dxdy = \int_{\mathbb{R}^{k+p}} (r(x) + s(y)) f_X(x) f_Y(y) dxdy \tag{B.62}$$

$$\int_{\mathbb{R}^{k+p}} g(x,y) f_{XY}(x,y) dxdy = \int_{\mathbb{R}^k} \int_{\mathbb{R}^p} \left(r(x) f_X(x) f_Y(y) + s(y) f_X f_Y(y) \right) dxdy \tag{B.63}$$

$$\begin{aligned} \int_{\mathbb{R}^{k+p}} g(x,y) f_{XY}(x,y) dxdy = & \\ & \int_{\mathbb{R}^k} \int_{\mathbb{R}^p} r(x) f_X(x) f_Y(y) dxdy + \cdots \\ & + \int_{\mathbb{R}^k} \int_{\mathbb{R}^p} s(y) f_X(x) f_Y(y) dxdy \end{aligned} \tag{B.64}$$

Definition B.1.22. Random Sequence:
Let $(\Omega, \mathcal{A}, P)$ be a probability space. Let $\zeta \in \Omega$ and $n \in \mathbb{N}$. Let $X_n(\zeta)$ be a mapping of the event ζ into the sample space Ω, then for each index n, $X_n(\zeta)$ is a random variable and in general $X_n(\zeta)$ is **Random Sequence**, also known as Stochastic Sequence. Some alternative definitions may extend the domain of X to the complex numbers and the indexes to the whole integer set (STARK; WOODS, 2002), but for the purposes of this work it shall suffice to treat the case defined. Additionally, the index generally relates to time and for clarity will often be referred as the "time index".

A lot of care should be taken to avoid confusion when referring to multiple realizations of random sequences. In this work, subscript indexes will refer to different time indexes (same process or realization) while superscripts will refer to different realizations (same random variable, different experiment result). This requires a few definitions that are very similar between themselves.

Definition B.1.23. Mean Function:
Given a random k-dimensional sequence X_n, the mean function at index n is defined by:

$$\mu_X[n] = E[X_n] = \int_{\mathbb{R}^k} x_n f_X(x_n, n) dx \tag{B.65}$$

It is important to note that the mean function is defined in terms on the expectation of a sum of random variables, that may or may not hold dependencies or correlations between themselves.

Definition B.1.24. Ensemble Average:
Given a random sequence X_n, for which m realizations were taken over the process at the same index (or time) n, each corresponding to a respective event ζ^i, $i \leq m$, their ensemble average is given by:

$$\Lambda_X[n] = \frac{1}{M} \sum_{m=1}^{M} x_n^m \tag{B.66}$$

The idea of an ensemble average is to take several experiments.
Definition B.1.25. Time Average:
Given a random sequence X_n, for which several realizations are made, and let the m-th realization taken over the process correspond to a respective event ζ_m, Let N be a time step on the process. Its Time Average μ_T is given by:

$$\mu_{TX} = \frac{1}{N} \sum_{n=1}^{N} x_n^m \tag{B.67}$$

The time average may also be regarded as a random variable, since the realization is both unique and arbitrary:

$$M_{T,X} = \frac{1}{N} \sum_{n=1}^{N} X_n \tag{B.68}$$

The notation T is related to a last observed instant, associated with the index N, and meant to differentiate from the notation used for ensemble average.
Definition B.1.26. Wide Sense Stationary Process:
A stochastic process X_n is called Wide-Sense Stationary if it satisfies the following properties:

a) The mean function is constant for time indexes n:

$$\mu_X[n] = \mu_X[0] \, \forall n \tag{B.69}$$

b) The auto-covariance is (time) shift-invariant:

$$K_{XX}[n, m] = K_{XX}[n + p, m + p] \, \forall p \tag{B.70}$$

Definition B.1.27. Mean-Square Convergence:
A stochastic sequence X_n converges in the mean-square sense to the random variable X if:

$$\lim_{n \to \infty} Var\left[X_n - X\right] = 0 \tag{B.71}$$

Definition B.1.28. Ergodic in the Mean:
Given a wide-sense stationary process X_n, if its time average converges in mean-square to the expected value, then X is called Ergodic in the Mean. This implies:

$$\lim_{N \to \infty} M_{T,X} = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} X_n \tag{B.72}$$

Such process is very useful because it can be observed over time in a single experiment rather than at a single instant over several experiments. It is generally the case of white noise.
Definition B.1.29. Independent Random Sequence:
An independent random sequence is one whose random variables X_n at any time-index are jointly independent for any combination of indexes.
Definition B.1.30. White Noise:
A wide-sense stationary, ergodic in the mean process X_n is called a White Noise if:

- It has zero mean:

$$E[X_n] = 0 \ (\forall n) \tag{B.73}$$

- It has no correlation at different times:

$$K_{XX}[n, p] = K_{XX}[n, n]\delta_{n,p} \ (\forall n, p) \tag{B.74}$$

  With $\delta_{n,p}$ the Kronecker delta:

$$\delta_{n,p} = \begin{cases} 1 & n = p \\ 0 & n \neq p \end{cases} \tag{B.75}$$

Theorem B.1.12. Expected Mean of White Noise:
The expected mean of a white noise X_n over a finite period is zero:

$$
\begin{aligned}
E[M] &= E\left[\frac{X_1 + X_2 + \cdots X_n}{N}\right] \\
&= \frac{E\left[X_1 + X_2 + \cdots X_n\right]}{N} \\
&= \frac{E\left[X_1\right] + E\left[X_2\right] + \cdots + E\left[X_n\right]}{N} \\
&= 0
\end{aligned}
\tag{B.76}
$$

Definition B.1.31. Independent And Identically Distributed Random Variable: A set of random variables X_n (which may or may not be a sequence) is called Independent and Identically Distributed (IDD) if for any $i \neq j$ X_i and X_j are independent but have the same distribution.

Theorem B.1.13. Chebyschev Inequality (Univariate):
Let X be an arbitrary random variable with mean μ and standard deviation σ, an upper bound can be given to how likely a realization of a random variable is to deviate a certain amount from the mean. This bound is given by:

$$
P[\|X - \mu\| \geq k\sigma] \leq \frac{1}{k^2} \tag{B.77}
$$

Proof: From Definition B.1.20, it follows:

$$
\sigma^2 = \int_{\mathbb{R}} (x - \mu)^2 f_X(x) dx \tag{B.78}
$$

From Definition B.1.12 $f_X(x) \geq 0$, thus:

$$
\int_{\mathbb{R}} (x - \mu)^2 f_X(x) dx \geq \int_{\|x-\mu\| \geq \delta} (x - \mu)^2 f_X(x) dx \tag{B.79}
$$

$$
\int_{\|x-\mu\| \geq \delta} (x - \mu)^2 f_X(x) dx \geq \delta^2 \int_{\|x-\mu\| \geq \delta} f_X(x) dx \tag{B.80}
$$

$$
\int_{\mathbb{R}} (x - \mu)^2 f_X(x) dx \geq \delta^2 \int_{\|x-\mu\| \geq \delta} f_X(x) dx \tag{B.81}
$$

Finally from Definition B.1.7

$$\delta^2 \int_{\|x-\mu\| \geq \delta} f_X(x)dx = \delta^2 P[\|x-\mu\| \geq \delta] \tag{B.82}$$

Therefore:

$$\sigma^2 \geq \delta^2 P[\|x-\mu\| \geq \delta] \tag{B.83}$$

Setting $\delta = k\sigma$ completes the proof:

$$\frac{1}{k^2} \geq P[\|x-\mu\| \geq k\sigma] \tag{B.84}$$

B.1.6 Other Theorems

Theorem B.1.14. Central Limit Theorem:
If for each positive integer n, $X_1, X_2, \cdots X_n$ are independent and identically distributed random variables with constant mean μ_X and variance σ_X^2, and defining the random variable the random variable Z with:

$$Z[n] = \frac{\overline{X}[n] - \mu_X}{\sigma_X/\sqrt{n}} \tag{B.85}$$

Where $\overline{X}[n]$ is stands for:

$$\overline{X}[n] = \sum_{k=1}^{n} \frac{X[k]}{n} \tag{B.86}$$

Which is a mean estimator.

Then the distribution of $Z[n]$ converges to a normal distribution with zero mean and unitary variance as n tends to infinity (MOOD, 1950).
Theorem B.1.15. Corollary to Limit Theorem:
Under the assumption and notation of Theorem B.1.14, the mean estimator, defined by Equation (B.86), has normal distribution with mean μ_X and variance σ_X^2/n.

ANNEX C - NOTES ON LINEAR ALGEBRA

This section explains some results from Linear Algebra theory that are referred throughout this work.

C.1 Review on useful matrix properties

Definition C.1.1. Transposition Operator:
The transpose of a matrix $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ with elements a_{ij} is the matrix $A^T \in \mathcal{M}(\mathbb{R})_{m\times n}$ with elements a_{ji}

.

Theorem C.1.1. Distribution of Transposition (Sum):
For any matrices $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ and $B \in \mathcal{M}(\mathbb{R})_{m\times n}$:

$$(A+B)^T = A^T + B^T \tag{C.1}$$

Theorem C.1.2. Distribution of Transposition (Product):
For any matrices $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ and $B \in \mathcal{M}(\mathbb{R})_{m\times n}$:

$$(AB)^T = B^T A^T \tag{C.2}$$

Definition C.1.2. Symmetric Matrix:
A matrix $A \in \mathcal{M}(\mathbb{R})_{n\times n}$ is said to be symmetric if and only if $A^T = A$.
Definition C.1.3. Anti-Symmetric Matrix:
A matrix $A \in \mathcal{M}(\mathbb{R})_{n\times n}$ is said to be anti-symmetric if and only if $A^T = -A$.
Definition C.1.4. Matrix Symmetric Part:
$A \in \mathcal{M}(\mathbb{R})_{n\times n}$, the symmetric part of A, denoted by $A_\oplus$, given by

$$A_\oplus = \frac{A + A^T}{2} \tag{C.3}$$

Definition C.1.5. Matrix Anti-Symmetric Part:
$A \in \mathcal{M}(\mathbb{R})_{n\times n}$, the anti-symmetric part of A, denoted by $A_\ominus$, given by

$$A_\ominus = \frac{A - A^T}{2} \tag{C.4}$$

Theorem C.1.3. Any matrix is the sum of a symmetric and an anti-symmetric matrix:

Let $A \in \mathcal{M}(\mathbb{R})_{n\times n}$, then one has:

$$A = \frac{A + A^T}{2} + \frac{A - A^T}{2} \tag{C.5}$$

$$A = A_{\oplus} + A_{\ominus} \tag{C.6}$$

Theorem C.1.4. Distribution of Symmetric Part operator:
For any matrices $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ and $B \in \mathcal{M}(\mathbb{R})_{m\times n}$:

$$(A + B)_{\oplus} = A_{\oplus} + B_{\oplus} \tag{C.7}$$

Proof:

$$(A + B)_{\oplus} = \frac{(A + B) + (A + B)^T}{2} \tag{C.8}$$

Using Theorem C.1.1

$$(A + B)_{\oplus} = \frac{A + B + A^T + B^T}{2} \tag{C.9}$$

Rearranging:

$$(A + B)_{\oplus} = \frac{A + A^T}{2} + \frac{B + B^T}{2} \tag{C.10}$$

Therefore:

$$(A + B)_{\oplus} = A_{\oplus} + B_{\oplus} \tag{C.11}$$

Which concludes the proof.

Definition C.1.6. Trace Operator:
The trace of a square matrix $A \in \mathcal{M}(\mathbb{R})_{n\times n}$ with elements a_{ij} is defined by:

$$Tr(A) = \sum_{i=1}^{n} a_{ii} \tag{C.12}$$

Theorem C.1.5. Trace is Invariant over Commutation:
For any matrices $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ and $B \in \mathcal{M}(\mathbb{R})_{m\times n}$

$$Tr(AB) = Tr(BA) \tag{C.13}$$

Proof: Let

$$C = AB \tag{C.14}$$

$$D = BA \tag{C.15}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \tag{C.16}$$

$$\therefore Tr(C) = \sum_{i=1}^{m} \sum_{k=1}^{n} a_{ik} b_{ki} \tag{C.17}$$

$$d_{ij} = \sum_{k=1}^{m} b_{ik} a_{kj} \tag{C.18}$$

$$\therefore Tr(D) = \sum_{i=1}^{m} \sum_{k=1}^{n} b_{ik} a_{ki} \tag{C.19}$$

Since summation is order independent:

$$\sum_{i=1}^{m} \sum_{k=1}^{n} a_{ik} b_{ki} = \sum_{i=1}^{m} \sum_{k=1}^{n} b_{ik} a_{ki} \tag{C.20}$$

Therefore:

$$Tr(C) = Tr(D) \tag{C.21}$$

Particularly this has one useful corollary. For any vector $u \in \mathcal{M}(\mathbb{R})_{n\times 1}$

$$\begin{aligned} Tr(uu^T) &= Tr(u^T u) \\ &= u^T u \\ &= \langle u, u \rangle \end{aligned} \tag{C.22}$$

Theorem C.1.6. Trace is Invariant over Transposition:
For any matrix $A \in \mathcal{M}(\mathbb{R})_{n\times n}$

$$Tr(A) = Tr(A^T) \tag{C.23}$$

Theorem C.1.7. Linearity Trace:
For any matrices $A \in \mathcal{M}(\mathbb{R})_{n\times m}$ and $B \in \mathcal{M}(\mathbb{R})_{m\times n}$:

$$Tr(A + B) = Tr(A) + Tr(B) \tag{C.24}$$

C.2 Tools from Linear Algebra Theory

Theorem C.2.1. Singular Value Decomposition:
For any matrix $A \in \mathcal{M}(\mathbb{R})_{n\times m}$, there exist unitary matrices $U \in \mathcal{M}(\mathbb{R})_{n\times n}$ and $V \in \mathcal{M}(\mathbb{R})_{m\times m}$ and a diagonal matrix $\Sigma \in \mathcal{M}(\mathbb{R})_{n\times m}$ such that:

$$A = U^T \Sigma V \tag{C.25}$$

And:

$$\Sigma ii \geq \Sigma jj \geq 0 \iff i > j \tag{C.26}$$

More details over the properties and existence of such decomposition are presented by (LIMA, 1995).

With this result, it is possible to present the proof for Theorem 2.1.1.

Assuming first that the weighting coefficient can be normalized:

$$1 = \sum_k a_k \tag{C.27}$$

Then, express J as in terms of the trace operator using Equation (C.22) and linearity of the trace operator (Theorem C.1.7):

$$J = \frac{1}{2}\sum_k a_k \mathrm{Tr}\left(\left(u_k^B - C_A^B u_k^A\right)\left(u_k^B - C_A^B u_k^A\right)^T\right) \tag{C.28}$$

$$J = \frac{1}{2}\sum_k a_k Tr\left(\left(u_k^B\right)\left(u_k^B\right)^T - C_A^B u_k^A\left(u_k^B\right)^T - u_k^B\left(u_k^A\right)^T\left(C_A^B\right)^T + C_A^B u_k^A\left(u_k^A\right)^T\left(C_A^B\right)^T\right) \tag{C.29}$$

Noting that the vectors are assumed normalized:

$$\mathrm{Tr}\left(\left(u_k^B\right)\left(u_k^B\right)^T\right) = 1 \tag{C.30}$$

$$\mathrm{Tr}\left(C_A^B u_k^A\left(u_k^A\right)^T\left(C_A^B\right)^T\right) = 1 \tag{C.31}$$

Also, by invariance of trace over transposition and commutation (Theorem C.1.5):

$$\mathrm{Tr}\left(-C_A^B u_k^A\left(u_k^B\right)^T - u_k^B\left(u_k^A\right)^T\left(C_A^B\right)^T\right) = 2\mathrm{Tr}\left(-C_A^B u_k^A\left(u_k^B\right)^T\right) \tag{C.32}$$

Reducing Equation (C.29) to:

$$J = 1 - \sum_k a_k \mathrm{Tr}\left(C_A^B u_k^A \left(u_k^B\right)^T\right) \tag{C.33}$$

By linearity of trace:

$$J = 1 - \mathrm{Tr}\left(C_A^B \sum_k \left(a_k u_k^A \left(u_k^B\right)^T\right)\right) \tag{C.34}$$

Defining as before:

$$K = \sum_k \left(a_k u_k^A \left(u_k^B\right)^T\right)^T \tag{C.35}$$

$$J = 1 - \mathrm{Tr}\left(C_A^B K^T\right) \tag{C.36}$$

Now it is necessary to find C_A^B that minimizes the expression above. Factoring K with Singular Value Decomposition (Theorem C.2.1):

$$K = U\,\Sigma\,V^T = U\,diag[\Sigma_{11},\,\Sigma_{22},\,\Sigma_{33}]\,V^T \tag{C.37}$$

Noting that U and V are unitary matrices, it is useful to define proper rotation matrices:

$$U_+ = U\,[1,\,1,\,(detU)] \tag{C.38}$$

$$V_+ = V\,[1,\,1,\,(detV)] \tag{C.39}$$

Thus it is also possible to rewrite K as:

$$K = U_+\,\Sigma\,V^T = U\,diag[\Sigma_{11},\,\Sigma_{22},\,\Sigma_{33}det(U)det(V)]\,V_+^T \tag{C.40}$$

Express the diagonal matrix as:

$$S = diag[\Sigma_{11},\,\Sigma_{22},\,\Sigma_{33}det(U)det(V)] \tag{C.41}$$

$$K = U_+ \Sigma S V_+^T \tag{C.42}$$

Since a product of rotation matrices is a rotation matrix itself, let:

$$W = U_+^T C_A^B V_+ \tag{C.43}$$

Then, conversely:

$$C_A^B = U_+ W V_+^T \tag{C.44}$$

This effectively becomes a change of variables in Equation (C.36), that becomes:

$$J = 1 - Tr\left(U_+ W V_+^T K^T\right) \tag{C.45}$$

Using the decomposition of K given by Equation (C.40):

$$J = 1 - Tr\left(U_+ W V_+^T \left(U S V_+^T\right)^T\right) \tag{C.46}$$

Simplifying:

$$J = 1 - Tr\left(U_+ W S U_+^T\right) \tag{C.47}$$

Applying the invariance of trace over commutation (Theorem C.1.5)

$$\begin{aligned} \mathrm{Tr}\left(U_+ W S U_+^T\right) &= \mathrm{Tr}\left(U_+^T U_+ W S\right) \\ &= \mathrm{Tr}\left(W S\right) \end{aligned} \tag{C.48}$$

As rotation matrix, W admits the following decomposition:

$$W = \cos(\Phi) I + (1 - \cos(\Phi))\, w w^T - \sin(\Phi) w^\times \tag{C.49}$$

With Φ a rotation angle and w a unit vector corresponding to the rotation axis. Therefore:

$$W S = \cos(\Phi) S + (1 - \cos(\Phi))\, w w^T S - \sin(\Phi) w^\times S \tag{C.50}$$

Applying the trace:

$$\mathrm{Tr}\,(WS) = \mathrm{Tr}\left(\cos(\Phi)S + (1-\cos(\Phi))\,ww^T S - \sin(\Phi)w^{\times}S\right) \tag{C.51}$$

Performing the operations:

$$\begin{aligned}\mathrm{Tr}\,(WS) = \mathrm{Tr}(S) - (1-\cos(\Phi))\Big(&\Sigma_{22} + \Sigma_{33}det(U)det(V) + (\Sigma_{11}-\Sigma_{22})\,w_2^2 \\ &+ (\Sigma_{11} - \Sigma_{33}det(U)det(V))\,w_3^2\Big)\end{aligned} \tag{C.52}$$

Since $\Sigma_{11} \geq \Sigma_{22} \geq \Sigma_{33} \geq 0$, the term in parentheses is always positive. Thus Equation (C.52) is maximized for Φ, which in turn minimizes Equation (C.46) as:

$$J = 1 - Tr\,(S) \tag{C.53}$$

Since S depends only on K, it cannot be further minimized since there are no (relevant) variables left to vary. But recalling Equation (C.49), and $\Phi = 0$, this form is now reduced to $W = I$, and from Equation (C.44):

$$\begin{aligned}C_A^B &= U_+ W V_+^T \\ &= U\,diag[1,\,1,\,(detU)(detV)]\,V^T\end{aligned} \tag{C.54}$$

Completing the proof.

Definition C.2.1. Projection Matrix:

The projection of a vector v over a direction given by a vector u is given by:

$$\begin{aligned}\mathrm{Proj}_u(v) &= uu^t\,\|u\|^{-2}\,v \\ &= P_u v\end{aligned} \tag{C.55}$$

With:

$$P_u = uu^t\,\|u\|^{-2} \tag{C.56}$$

And P_u is referred as the Projection Matrix over the direction of u.

Definition C.2.2. Projection Removal Matrix:
The projection of a vector v over a direction given by a vector u is given by:

$$\begin{aligned} v - \mathrm{Proj}_u(v) &= v - (uu^t \left\|u\right\|^{-2})v \\ &= (I - P_u)v \\ &= R_u v \end{aligned} \tag{C.57}$$

With:

$$R_u = I - uu^t \left\|u\right\|^{-2} \tag{C.58}$$

R_u is called the Projection Removal Matrix over the direction of u.

ANNEX D - CODE TO VERIFY PROPERTIES OF THE VARIANCE OF A CROSS PRODUCT

This annex contains the computer code that deduces the equations for the ⊠ operator. It is developed with Wolfram Mathematica (WOLFRAM RESEARCH, 2018).

Although the random variables involved are modeled as being normally distributed, it was already noted in Section 3.2 that only properties of the Variance matrix and the expectancy operator are needed to perform those demonstrations, hence the actual distribution used does not affect the produced results, provided the zero expectancy hypothesis is sustained.

The formatting in these pages inherits the template used by Mathematica notebooks, with the font set to Lucida Sans Unicode, in order to correctly print the ⊠ symbol.

Demonstrating properties of the ⊠ operator

Clearing the workspace and making basic definitions:

In[190]:=
```
ClearAll["Global`*"]
aaTProd[v__, u__] := Transpose[{u}].{v}
```

Defining the concerned vectors and covariances:

In[192]:=
```
μv = {Subscript[μ, x], Subscript[μ, y], Subscript[μ, z]};
εv = {Subscript[ε, x], Subscript[ε, y], Subscript[ε, z]};
Kμ = {{σμxx, σμxy , σμxz }, {σμxy , σμyy, σμyz }, {σμxz, σμyz , σμzz}};
Kε = {{σεxx, σεxy , σεxz }, {σεxy , σεyy, σεyz }, {σεxz, σεyz , σεzz}};
μv ≈ NormalDistribution[{0, 0, 0}, Kμ];
εv ≈ NormalDistribution[{0, 0, 0}, Kε];
Σ = ArrayFlatten[ Kμ                    Table[0, {i, 3}, {j, 3}]
                  Table[0, {i, 3}, {j, 3}]   Kε                  ];
```

Noting the cross product:

In[199]:=
```
μvXεv = Cross[μv, εv]
```

Out[199]= $\{\varepsilon_z \mu_y - \varepsilon_y \mu_z, -\varepsilon_z \mu_x + \varepsilon_x \mu_z, \varepsilon_y \mu_x - \varepsilon_x \mu_y\}$

Showing that the expectation of the cross product is zero:

In[200]:=
```
Expectation[μvXεv, Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]]
```

Out[200]= $\{0, 0, 0\}$

Evaluating the cross product times its transpose:

In[201]:=
```
aTa = aaTProd[μvXεv, μvXεv]
```

Out[201]= $\{\{(\varepsilon_z \mu_y - \varepsilon_y \mu_z)^2, (-\varepsilon_z \mu_x + \varepsilon_x \mu_z)(\varepsilon_z \mu_y - \varepsilon_y \mu_z), (\varepsilon_y \mu_x - \varepsilon_x \mu_y)(\varepsilon_z \mu_y - \varepsilon_y \mu_z)\},$
$\{(-\varepsilon_z \mu_x + \varepsilon_x \mu_z)(\varepsilon_z \mu_y - \varepsilon_y \mu_z), (-\varepsilon_z \mu_x + \varepsilon_x \mu_z)^2, (\varepsilon_y \mu_x - \varepsilon_x \mu_y)(-\varepsilon_z \mu_x + \varepsilon_x \mu_z)\},$
$\{(\varepsilon_y \mu_x - \varepsilon_x \mu_y)(\varepsilon_z \mu_y - \varepsilon_y \mu_z), (\varepsilon_y \mu_x - \varepsilon_x \mu_y)(-\varepsilon_z \mu_x + \varepsilon_x \mu_z), (\varepsilon_y \mu_x - \varepsilon_x \mu_y)^2\}\}$

Computing the expectation yields the covariance:

In[202]:=
```
Expectation[aTa, Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]]
```

Out[202]= $\{\{\sigma_{\varepsilon zz} \sigma_{\mu yy} - 2 \sigma_{\varepsilon yz} \sigma_{\mu yz} + \sigma_{\varepsilon yy} \sigma_{\mu zz},$
$-\sigma_{\varepsilon zz} \sigma_{\mu xy} + \sigma_{\varepsilon yz} \sigma_{\mu xz} + \sigma_{\varepsilon xz} \sigma_{\mu yz} - \sigma_{\varepsilon xy} \sigma_{\mu zz}, \sigma_{\varepsilon yz} \sigma_{\mu xy} - \sigma_{\varepsilon yy} \sigma_{\mu xz} - \sigma_{\varepsilon xz} \sigma_{\mu yy} + \sigma_{\varepsilon xy} \sigma_{\mu yz}\},$
$\{-\sigma_{\varepsilon zz} \sigma_{\mu xy} + \sigma_{\varepsilon yz} \sigma_{\mu xz} + \sigma_{\varepsilon xz} \sigma_{\mu yz} - \sigma_{\varepsilon xy} \sigma_{\mu zz}, \sigma_{\varepsilon zz} \sigma_{\mu xx} - 2 \sigma_{\varepsilon xz} \sigma_{\mu xz} + \sigma_{\varepsilon xx} \sigma_{\mu zz},$
$-\sigma_{\varepsilon yz} \sigma_{\mu xx} + \sigma_{\varepsilon xz} \sigma_{\mu xy} + \sigma_{\varepsilon xy} \sigma_{\mu xz} - \sigma_{\varepsilon xx} \sigma_{\mu yz}\}, \{\sigma_{\varepsilon yz} \sigma_{\mu xy} - \sigma_{\varepsilon yy} \sigma_{\mu xz} - \sigma_{\varepsilon xz} \sigma_{\mu yy} + \sigma_{\varepsilon xy} \sigma_{\mu yz},$
$-\sigma_{\varepsilon yz} \sigma_{\mu xx} + \sigma_{\varepsilon xz} \sigma_{\mu xy} + \sigma_{\varepsilon xy} \sigma_{\mu xz} - \sigma_{\varepsilon xx} \sigma_{\mu yz}, \sigma_{\varepsilon yy} \sigma_{\mu xx} - 2 \sigma_{\varepsilon xy} \sigma_{\mu xy} + \sigma_{\varepsilon xx} \sigma_{\mu yy}\}\}$

Defining the ⊠ operator as a function of Kμ and Kε:

In[203]:=
```
⊠[Kμ__, Kε__] := Module[{μv, εv, Σ, aTa, μvXεv, a, b, c, d, e, f, g, h, i, j, k, l, A, B, C, D, E, F},
    a = Kμ[[1]][[1]]; b = Kμ[[2]][[1]]; c = Kμ[[3]][[1]];
    d = Kμ[[2]][[2]]; e = Kμ[[2]][[3]]; f = Kμ[[3]][[3]];
    g = Kε[[1]][[1]]; h = Kε[[2]][[1]]; i = Kε[[3]][[1]];
    j = Kε[[2]][[2]]; k = Kε[[2]][[3]]; l = Kε[[3]][[3]];
    Σ = ArrayFlatten[{{Kμ, Table[0, {ii, 3}, {jj, 3}]}, {Table[0, {ii, 3}, {jj, 3}], Kε}}];
    μvXεv = Cross[{A, B, C}, {D, E, F}];
    aTa = aaTProd[μvXεv, μvXεv];
    Expectation[aTa, Join[{A, B, C, D, E, F}] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]]]
```

Checking the result:

In[204]:=
```
⊠[Kμ, Kε] == Expectation[aTa, Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]]
```

Out[204]= True

Verifying that ⊠ is bilinear:

In[205]:=
```
a^2 b^2 ⊠[Kμ, Kε] == Expectation[aaTProd[Cross[a μv, b εv], Cross[a μv, b εv]],
      Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]] // Simplify
```

Out[205]= True

Verifying that ⊠ is commutative:

In[206]:=
```
⊠[Kμ, Kε] == ⊠[Kε, Kμ] // Simplify
```

Out[206]= True

Verifying pre-multiplication of second vector:

In[211]:=
```
M = {{a, b, c}, {d, e, f}, {g, h, i}};
⊠[Kμ, M.Kε.(M^T)] == Expectation[aaTProd[Cross[ μv, M.εv], Cross[μv, M.εv]],
      Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]] // FullSimplify
```

Out[212]= True

Verifying pre-multiplication of cross product matrix:

In[217]:=
```
M.(⊠[Kμ, Kε]).(M^T) == Expectation[aaTProd[M.Cross[ μv, εv], M.Cross[μv, εv]],
      Join[μv, εv] ≈ MultinormalDistribution[{0, 0, 0, 0, 0, 0}, Σ]] // Simplify
```

Out[217]= True

ANNEX E - PROOF OF RECURSIVE FORMULAS FOR COVARIANCE

This annex contains the demonstrations for the recursive formulas the allow computing the covariances for each of the four cases of random processed shown in Section 3.6. The notation and definitions are same introduced in Section 3.6.1, including the definitions of the auxiliary sequences.

While the developments on this annex are original from this work, Proposition 3.6.1 can be regarded as a particular case of applying Theorem B.1.7 to a conveniently constructed random variable. Nonetheless, this development is still shown here as it exemplifies the framework applied for the other cases.

It is also recalled that the variance formulas or their auxiliary sequences are generally shown in summation forms that follow the pattern in Equation (3.9), which are easily convertible to recursive formulas defined and explained in Section 3.1.

E.0.1 Case 1 Proof: Sum of Known Gains Times Noise

Recalling Equation (3.46):

$$Var\left[\sum_{k=1}^{k} f_k \varepsilon_k\right] = \sum_{k=1}^{k} f_k \sigma_\varepsilon^2 f_k^T \tag{E.1}$$

From the definition of variance (Definition B.1.20):

$$Var\left[\sum_{k} f_k \varepsilon_k\right] = E\left[\left(\sum_{k} f_k \varepsilon_k\right)\left(\sum_{k} f_k \varepsilon_k\right)^T\right] \tag{E.2}$$

Changing the index in the second summation:

$$Var\left[\sum_{k} f_k \varepsilon_k\right] = E\left[\left(\sum_{k} f_k \varepsilon_k\right)\left(\sum_{j} f_j \varepsilon_j\right)^T\right] \tag{E.3}$$

Using Theorem C.1.1 and Theorem C.1.2 to handle the transposition operation::

$$Var\left[\sum_{k} f_k \varepsilon_k\right] = E\left[\left(\sum_{k} f_k \varepsilon_k\right)\left(\sum_{j} {\varepsilon_j}^T f_j^T\right)\right] \tag{E.4}$$

Noting that:

$$\left(\sum_i a_i\right)\left(\sum_j b_j\right) = \sum_i \sum_j a_i b_j \tag{E.5}$$

Applying to the previous equation:

$$Var\left[\sum_k f_k \varepsilon_k\right] = E\left[\sum_k \sum_j f_k \varepsilon_k {\varepsilon_j}^T f_j^T\right] \tag{E.6}$$

From the linearity of the expectation operator (Theorem B.1.1) :

$$Var\left[\sum_k f_k \varepsilon_k\right] = \sum_k \sum_j f_k E\left[\varepsilon_k {\varepsilon_j}^T\right] f_j^T \tag{E.7}$$

From the independence of ε_k and ε_j for $k \neq j$, and using the kronecker delta (Equation (B.75)):

$$E\left[\varepsilon_k \varepsilon_j^T\right] = \sigma_\varepsilon^2 \delta_{kj} \tag{E.8}$$

Thus the double summation in Equation (E.7) is reduced to:

$$Var\left[\sum_k f_k \varepsilon_k\right] = \sum_k f_k \sigma_\varepsilon^2 f_k^T \tag{E.9}$$

Concluding the proof.

E.0.2 Case 2 Proof: Sum of Known Gains Times Bias Cross Product Matrix Times Case 1

Recalling Equation (3.47):

$$Var\left[\sum_{k=1}^{m} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = H_m\left(G_m\right) H_m^T - 2\left(J_m {H_m}^T\right)_{\oplus} + K_m \tag{E.10}$$

The demonstration of this result will be performed by finite induction. The cases for $m \in \{1, 2, 3\}$ will be done both as the first step and to find a first guess for the

general result. The obtained equation however does not allow recursive implementation, thus it will have to be manipulated before the proof is concluded and Equation (3.47) demonstrated.

For $m = 1$:

$$Var\left[\sum_{k=1}^{1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_1 \mu^{\times} f_1 \varepsilon_1\right] \tag{E.11}$$

Using Theorem B.1.3 and Proposition 3.2.4:

$$Var\left[\sum_{k=1}^{1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_1 \left(\sigma_{\mu}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 {f_1}^T\right)\right) {h_1}^T \tag{E.12}$$

For $m = 2$:

$$Var\left[\sum_{k=1}^{2} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_2 \mu^{\times} \left(f_2 \varepsilon_2 + f_1 \varepsilon_1\right) + h_1 \mu^{\times} f_1 \varepsilon_1\right] \tag{E.13}$$

$$Var\left[\sum_{k=1}^{2} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_2 \mu^{\times} f_2 \varepsilon_2 + (h_2 + h_1) \mu^{\times} f_1 \varepsilon_1\right] \tag{E.14}$$

Note that:

$$Cov\left[h_2 \mu^{\times} f_2 \varepsilon_2 \ , \ (h_2 + h_1) \mu^{\times} f_1 \varepsilon_1\right] = E\left[\left(h_2 \mu^{\times} f_2 \varepsilon_2\right) \left((h_2 + h_1) \mu^{\times} f_1 \varepsilon_1\right)^T\right] \tag{E.15}$$

Using Theorem C.1.2 to handle the transposition operation:

$$Cov\left[h_2 \mu^{\times} f_2 \varepsilon_2 \ , \ (h_2 + h_1) \mu^{\times} f_1 \varepsilon_1\right] = E\left[h_2 \mu^{\times} f_2 \varepsilon_2 \varepsilon_1^T f_1^T (-\mu^{\times}) \left(h_2 + h_1\right)^T\right] \tag{E.16}$$

From the linearity of the expectation operator (Theorem B.1.1) :

$$Cov\left[h_2 \mu^{\times} f_2 \varepsilon_2 \ , \ (h_2 + h_1) \mu^{\times} f_1 \varepsilon_1\right] = h_2 E\left[\mu^{\times} f_2 \varepsilon_2 \varepsilon_1^T f_1^T (-\mu^{\times})\right] (h_2 + h_1)^T \tag{E.17}$$

The uncorrelated terms in the middle, cause the cross-covariance to become zero, thus, applying the identity for the variance of a sum (Theorem B.1.3) on Equation (E.14):

$$Var\left[\sum_{k=1}^{2} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_2 \mu^{\times} f_2 \varepsilon_2 + (h_2 + h_1)\mu^{\times} f_1 \varepsilon_1\right] \tag{E.18}$$
$$= Var\left[(h_2 + h_1)\mu^{\times} f_1 \varepsilon_1\right] + Var\left[h_2 \mu^{\times} f_2 \varepsilon_2\right]$$

Using Proposition 3.2.1, with properties described on Proposition 3.2.2 and Proposition 3.2.4:

$$Var\left[\sum_{k=1}^{2} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = (h_2 + h_1)\left(\sigma_{\mu}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right)(h_2 + h_1)^T \tag{E.19}$$
$$+ h_2\left(\sigma_{\mu}^2 \boxtimes \left(f_2 \sigma_{\varepsilon}^2 f_2^T\right)\right) h_2^T$$

For $m = 3$:

$$Var\left[\sum_{k=1}^{3} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_3 \mu^{\times}\left(f_3 \varepsilon_3 + f_2 \varepsilon_2 + f_1 \varepsilon_1\right) + h_2 \mu^{\times}\left(f_2 \varepsilon_2 + f_1 \varepsilon_1\right)\right. \tag{E.20}$$
$$\left. + h_1 \mu^{\times} f_1 \varepsilon_1\right]$$

$$Var\left[\sum_{k=1}^{3} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_3 \mu^{\times} f_3 \varepsilon_3 \ + (h_3 + h_2)\,\mu^{\times} f_2 \varepsilon_2 + (h_3 + h_2 + h_1)\,\mu^{\times} f_1 \varepsilon_1\right] \tag{E.21}$$

$$Var\left[\sum_{k=1}^{3} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_3 \mu^{\times} f_3 \varepsilon_3\right] + Var\left[(h_3 + h_2)\,\mu^{\times} f_2 \varepsilon_2\right] \tag{E.22}$$
$$+ Var\left[(h_3 + h_2 + h_1)\,\mu^{\times} f_1 \varepsilon_1\right]$$

Recalling Equation (3.51):

$$H_k \ = H_{k-1} + h_k = \sum_{i=1}^{k} h_i \tag{E.23}$$

Then replacing terms in Equation (E.22):

$$Var\left[\sum_{k=1}^{3} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[(H_3 - H_2)\,\mu^{\times} f_3 \varepsilon_3\right] + Var\left[(H_3 - H_1)\,\mu^{\times} f_2 \varepsilon_2\right] + Var\left[(H_3 - H_0)\,\mu^{\times} f_1 \varepsilon_1\right] \tag{E.24}$$

$$\begin{aligned} Var\left[\sum_{k=1}^{3} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (H_3 - H_0)\left(\sigma_\mu^2 \boxtimes \left(f_1 \sigma_\varepsilon^2 f_1^T\right)\right)(H_3 - H_0)^T \\ &+ (H_3 - H_1)\left(\sigma_\mu^2 \boxtimes \left(f_2 \sigma_\varepsilon^2 f_2^T\right)\right)(H_3 - H_1)^T \\ &+ (H_3 - H_2)\left(\sigma_\mu^2 \boxtimes \left(f_3 \sigma_\varepsilon^2 f_3^T\right)\right)(H_3 - H_2)^T \end{aligned} \tag{E.25}$$

This allows proposing the inductive hypothesis:

$$Var\left[\sum_{k=1}^{m} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{m} (H_m - H_{k-1})\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)(H_m - H_{k-1})^T \tag{E.26}$$

Assume the hypothesis valid for $m = M$:

$$Var\left[\sum_{k=1}^{M} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{M} (H_M - H_{k-1})\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)(H_M - H_{k-1})^T \tag{E.27}$$

For $m = M + 1$:

$$Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_{M+1} \mu^{\times} \sum_{i=1}^{M+1} f_i \varepsilon_i + \sum_{k=1}^{M} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] \tag{E.28}$$

$$Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_{M+1} \mu^{\times} f_{M+1} \varepsilon_{M+1} + h_{M+1} \mu^{\times} \sum_{i=1}^{M} f_i \varepsilon_i + \sum_{k=1}^{M} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] \tag{E.29}$$

Noting that the hypothesis is assumed valid for any arbitrary sequence h_k, define an auxiliary sequence:

$$h'_k = \begin{cases} h_k & k < M \\ h_M + h_{M+1} & k = M \end{cases} \tag{E.30}$$

Then the summation multiplied by $h_{M+1}\mu^{\times}$ inside the variance in Equation (E.29) is incorporated into the later sum:

$$Var\left[\sum_{k=1}^{M+1} h_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] = Var\left[h_{M+1}\mu^{\times}f_{M+1}\varepsilon_{M+1} + \sum_{k=1}^{M} h'_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] \tag{E.31}$$

From the uncorrelation between ε_{M+1} and ε_i for $i \leq M$, and using Theorem B.1.3

$$Var\left[\sum_{k=1}^{M+1} h_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] = Var\left[h_{M+1}\mu^{\times}f_{M+1}\varepsilon_{M+1}\right] + Var\left[\sum_{k=1}^{M} h'_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] \tag{E.32}$$

Recalling that the hypothesis is assumed valid for h'_k as well:

$$\begin{aligned} Var\left[\sum_{k=1}^{M+1} h_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] &= Var\left[h_{M+1}\mu^{\times}f_{M+1}\varepsilon_{M+1}\right] \\ &+ \sum_{k=1}^{M}\left(H'_M - H'_{k-1}\right)\left(\sigma_\mu^2 \boxtimes \left(f_k\sigma_\varepsilon^2 f_k^T\right)\right)\left(H'_M - H_{k-1}\right)^T \end{aligned} \tag{E.33}$$

Where H'_k is analogously defined:

$$H'_k = \begin{cases} H_k & k < M \\ H_M + h_{M+1} & k = M \end{cases} \tag{E.34}$$

Noticeably: $H'_M = H_{M+1}$, thus:

$$\begin{aligned} Var\left[\sum_{k=1}^{M+1} h_k\mu^{\times}\sum_{i=1}^{k} f_i\varepsilon_i\right] &= Var\left[h_{M+1}\mu^{\times}f_{M+1}\varepsilon_{M+1}\right] \\ &+ \sum_{k=1}^{M}\left(H_{M+1} - H_{k-1}\right)\left(\sigma_\mu^2 \boxtimes \left(f_k\sigma_\varepsilon^2 f_k^T\right)\right)\left(H_{M+1} - H_{k-1}\right)^T \end{aligned} \tag{E.35}$$

Then using the properties described in Proposition 3.2.4 and Proposition 3.2.5 of the $\boxtimes$ operator:

$$Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_{M+1}\left(\sigma_\mu^2 \boxtimes \left(f_{M+1}\sigma_\varepsilon^2 f_{M+1}^T\right)\right) h_{M+1}^T + \sum_{k=1}^{M} (H_{M+1} - H_{k-1})\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)(H_{M+1} - H_{k-1})^T \tag{E.36}$$

Since $h_{M+1} = H_{M+1} - H_M$:

$$Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = (H_{M+1} - H_M)\left(\sigma_\mu^2 \boxtimes \left(f_{M+1}\sigma_\varepsilon^2 f_{M+1}^T\right)\right)(H_{M+1} - H_M)^T + \sum_{k=1}^{M} (H_{M+1} - H_{k-1})\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)(H_{M+1} - H_{k-1})^T \tag{E.37}$$

And the last term can be incorporated into the summation:

$$Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{M+1} (H_{M+1} - H_{k-1})\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)(H_{M+1} - H_{k-1})^T \tag{E.38}$$

Which concludes the inductive step.

There remains the need to find recursive formulas for the computation of Equation (E.38). Noting that:

$$\begin{aligned} Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{M+1} \Big(& H_{M+1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T \\ & - H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T \\ & - H_{M+1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{k-1}}^T \\ & + H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{k-1}}^T \Big) \end{aligned} \tag{E.39}$$

Then the two middle terms can be grouped as twice a symmetric part matrix:

$$
\begin{aligned}
Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{M+1} \Big(& H_{M+1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T \\
& - 2\Big(H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T\Big)_{\oplus} \\
& + H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{k-1}}^T\Big)
\end{aligned}
\tag{E.40}
$$

Distributing the summation operator:

$$
\begin{aligned}
Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = & \sum_{k=1}^{M+1} \left(H_{M+1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T\right) \\
& - 2\sum_{k=1}^{M+1} \left(H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{M+1}}^T\right)_{\oplus} \\
& + \sum_{k=1}^{M+1} \left(H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{k-1}}^T\right)
\end{aligned}
\tag{E.41}
$$

Removing from summations terms that do not depend on the summation index, and using Theorem C.1.4

$$
\begin{aligned}
Var\left[\sum_{k=1}^{M+1} h_k \mu^{\times} \sum_{i=1}^{k} f_i \varepsilon_i\right] = & H_{M+1}\left(\sum_{k=1}^{M+1} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)\right) H_{M+1}^T \\
& - 2\left(\left(\sum_{k=1}^{M+1} H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right)\right) {H_{M+1}}^T\right)_{\oplus} \\
& + \sum_{k=1}^{M+1} \left(H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) {H_{k-1}}^T\right)
\end{aligned}
\tag{E.42}
$$

Naming a few summations:

$$
G_m = \sum_{k=1}^{m} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) \tag{E.43}
$$

$$
J_m = \sum_{k=1}^{m} H_{k-1}\left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T\right)\right) \tag{E.44}
$$

$$K_m = \sum_{k=1}^{m} \left(H_{k-1} \left(\sigma_\mu^2 \boxtimes \left(f_k \sigma_\varepsilon^2 f_k^T \right) \right) {H_{k-1}}^T \right) \tag{E.45}$$

Replacing those new summations into Equation (E.42):

$$Var \left[\sum_{k=1}^{m} h_k \mu^\times \sum_{i=1}^{k} f_i \varepsilon_i \right] = H_m \left(G_m \right) H_m^T - 2 \left(J_m {H_m}^T \right)_\oplus + K_m \tag{E.46}$$

Thus concluding the proof.

E.0.3 Case 3 Proof: Sum of Known Gains Times Noise Cross Product Matrix Times Case 1

Recalling Equation (3.52):

$$Var \left[\sum_{m=1}^{n} h_m \xi_m^\times \sum_{k=1}^{m} f_k \varepsilon_k \right] = h_n \left(L_n \right) h_n^T + Var \left[\sum_{m=1}^{n-1} h_m \xi_m^\times \sum_{k=1}^{m} f_k \varepsilon_k \right] \tag{E.47}$$

In this case, finite induction is unnecessary, since it is possible to directly establish a recurrence formula. Although some examples are helpful:

For $n = 1$:

$$Var \left[\sum_{m=1}^{1} h_m \xi_m^\times \sum_{k=1}^{m} f_k \varepsilon_k \right] = Var \left[h_1 \xi_1^\times f_1 \varepsilon_1 \right] \tag{E.48}$$

Then, using the properties described in Proposition 3.2.4 and Proposition 3.2.5 of the $\boxtimes$ operator:

$$Var \left[\sum_{m=1}^{1} h_m \xi_m^\times \sum_{k=1}^{m} f_k \varepsilon_k \right] = h_1 \left(\sigma_\xi^2 \boxtimes \left(f_1 \sigma_\varepsilon^2 f_1^T \right) \right) h_1^T \tag{E.49}$$

For $n = 2$:

$$Var \left[\sum_{m=1}^{2} h_m \xi_m^\times \sum_{k=1}^{m} f_k \varepsilon_k \right] = Var \left[h_2 \xi_2^\times f_2 \varepsilon_2 + h_2 \xi_2^\times f_1 \varepsilon_1 + h_1 \xi_1^\times f_1 \varepsilon_1 \right] \tag{E.50}$$

Noting that:

$$Cov \left[h_2 \xi_2^\times f_2 \varepsilon_2 \ , h_2 \xi_2^\times f_1 \varepsilon_1 \ \right] = E \left[h_2 \xi_2^\times f_2 \varepsilon_2 \ \varepsilon_1^T f_1^T {\xi_2^\times}^T h_2^T \right] \tag{E.51}$$

Then, using the distributivity of expectancy for independent variables (Theorem B.1.10):

$$Cov\left[h_2\xi_2^{\times}f_2\varepsilon_2\ ,h_2\xi_2^{\times}f_1\varepsilon_1\ \right] = E\left[h_2\xi_2^{\times}f_2\varepsilon_2\ \varepsilon_1^T f_1^T {\xi_2^{\times}}^T h_2^T\right] \tag{E.52}$$

Now applying Stacked Expectation (Theorem B.1.2):

$$Cov\left[h_2\xi_2^{\times}f_2\varepsilon_2\ ,h_2\xi_2^{\times}f_1\varepsilon_1\ \right] = E\left[h_2\xi_2^{\times}f_2E\left[\varepsilon_2\ \varepsilon_1^T|\xi_1,\xi_2\right]f_1^T{\xi_2^{\times}}^T h_2^T\right] \tag{E.53}$$

Where it is recalled that $|$ denotes statistical conditioning, as per Definition B.1.18 and Definition B.1.9.

Since every ξ_j is independent from any ε_i:

$$E\left[\varepsilon_2\ \varepsilon_1^T|\xi_1,\xi_2\right] = E\left[\varepsilon_2\ \varepsilon_1^T\right] \tag{E.54}$$

Thus:

$$Cov\left[h_2\xi_2^{\times}f_2\varepsilon_2\ ,h_2\xi_2^{\times}f_1\varepsilon_1\ \right] = E\left[h_2\xi_2^{\times}f_2E\left[\varepsilon_2\ \varepsilon_1^T\right]f_1^T{\xi_2^{\times}}^T h_2^T\right] \tag{E.55}$$

And the inner expectation is zero, therefore:

$$Cov\left[h_2\xi_2^{\times}f_2\varepsilon_2\ ,h_2\xi_2^{\times}f_1\varepsilon_1\ \right] = 0 \tag{E.56}$$

This means that cross-covariances that do not repeat indexes for both random variables are zero. Therefore:

$$Var\left[\sum_{m=1}^{2}h_m\xi_m^{\times}\sum_{k=1}^{m}f_k\varepsilon_k\right] = Var\left[h_2\xi_2^{\times}f_2\varepsilon_2\right] + Var\left[h_2\xi_2^{\times}f_1\varepsilon_1\right] + Var\left[h_1\xi_1^{\times}f_1\varepsilon_1\right] \tag{E.57}$$

Using the $\boxtimes$ operator:

$$\begin{aligned} Var\left[\sum_{m=1}^{2}h_m\xi_m^{\times}\sum_{k=1}^{m}f_k\varepsilon_k\right] &= h_2\left(\sigma_{\xi}^2\boxtimes\left(f_2\sigma_{\varepsilon}^2f_2^T\right)\right)h_2^T + h_2\left(\sigma_{\xi}^2\boxtimes\left(f_1\sigma_{\varepsilon}^2f_1^T\right)\right)h_2^T \\ &+ h_1\left(\sigma_{\xi}^2\boxtimes\left(f_1\sigma_{\varepsilon}^2f_1^T\right)\right)h_1^T \end{aligned} \tag{E.58}$$

Factoring out h_2 and h_2^T:

$$Var\left[\sum_{m=1}^{2} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = h_2 \left(\left(\sigma_{\xi}^2 \boxtimes \left(f_2 \sigma_{\varepsilon}^2 f_2^T\right)\right) + \left(\sigma_{\xi}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right)\right) h_2^T + h_1 \left(\sigma_{\xi}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right) h_1^T \tag{E.59}$$

For $n = 3$:

$$Var\left[\sum_{m=1}^{3} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = Var\left[h_3 \xi_3^{\times} f_3 \varepsilon_3 + h_3 \xi_3^{\times} f_2 \varepsilon_2 + h_3 \xi_3^{\times} f_1 \varepsilon_1 + h_2 \xi_2^{\times} f_2 \varepsilon_2 + h_2 \xi_2^{\times} f_1 \varepsilon_1 + h_1 \xi_1^{\times} f_1 \varepsilon_1\right] \tag{E.60}$$

Under the same argument:

$$Var\left[\sum_{m=1}^{3} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = Var[h_3 \xi_3^{\times} f_3 \varepsilon_3] + Var[h_3 \xi_3^{\times} f_2 \varepsilon_2] + Var[h_3 \xi_3^{\times} f_1 \varepsilon_1] + Var[h_2 \xi_2^{\times} f_2 \varepsilon_2] + Var[h_2 \xi_2^{\times} f_1 \varepsilon_1] + Var[h_1 \xi_1^{\times} f_1 \varepsilon_1] \tag{E.61}$$

$$\begin{aligned} Var\left[\sum_{m=1}^{3} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = {} & h_3 \left\{\left(\sigma_{\xi}^2 \boxtimes \left(f_3 \sigma_{\varepsilon}^2 f_3^T\right)\right) + \left(\sigma_{\xi}^2 \boxtimes \left(f_2 \sigma_{\varepsilon}^2 f_2^T\right)\right) \right. \\ & \left. + \left(\sigma_{\xi}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right)\right\} h_3^T \\ & + h_2 \left(\left(\sigma_{\xi}^2 \boxtimes \left(f_2 \sigma_{\varepsilon}^2 f_2^T\right)\right) + \left(\sigma_{\xi}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right)\right) h_2^T \\ & + h_1 \left(\sigma_{\xi}^2 \boxtimes \left(f_1 \sigma_{\varepsilon}^2 f_1^T\right)\right) h_1^T \end{aligned} \tag{E.62}$$

Which allows a straightforward formula:

$$Var\left[\sum_{m=1}^{n} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = h_n \left(\sum_{k=1}^{n} \left(\sigma_{\xi}^2 \boxtimes \left(f_k \sigma_{\varepsilon}^2 f_k^T\right)\right)\right) h_n^T + Var\left[\sum_{m=1}^{n-1} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] \tag{E.63}$$

Naming a sequence:

$$L_n = \sum_{k=1}^{n} \left(\sigma_{\xi}^2 \boxtimes \left(f_k \sigma_{\varepsilon}^2 f_k^T \right) \right) \tag{E.64}$$

Therefore:

$$Var\left[\sum_{m=1}^{n} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] = h_n \left(L_n\right) h_n^T + Var\left[\sum_{m=1}^{n-1} h_m \xi_m^{\times} \sum_{k=1}^{m} f_k \varepsilon_k\right] \tag{E.65}$$

Which concludes the proof.

E.0.4 Case 4 Proof: Sum of Known Gains Times Case 1

Recalling Equation (3.54):

$$\begin{aligned} &Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\ &= h_{m+1} W_{m+1} h_{m+1}^T + 2\left(h_{m+1}\left(W_m H_m^T - X_m\right)\right)_{\oplus} \end{aligned} \tag{E.66}$$

Proof is provided by finite induction. It is again necessary to showcase a few examples in order to infer the form adopted by the general equation. The result, however, will again require manipulation in order to admit recursive implementation.

For $m = 2$:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = E\left[\left(\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right)\left(\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right)^T\right] \tag{E.67}$$

Expanding the summations:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = E\left[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T\right] \tag{E.68}$$

Performing the distributions and using Theorem C.1.2 and Theorem C.1.1:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = E\left[h_1 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_1^T + h_1 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_2^T + h_1 f_1 \varepsilon_1 \varepsilon_2^T f_2^T h_2^T \right. \\ + h_2 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_1^T + h_2 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_2^T + h_2 f_1 \varepsilon_1 \varepsilon_2^T f_2^T h_2^T \\ \left. + h_2 f_2 \varepsilon_2 {\varepsilon_1}^T f_1^T h_1^T + h_2 f_2 \varepsilon_2 {\varepsilon_1}^T f_1^T h_2^T + h_2 f_2 \varepsilon_2 \varepsilon_2^T f_2^T h_2^T\right] \tag{E.69}$$

Applying the linearity property of the expectancy operator (Theorem B.1.1) :

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = E\left[h_1 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_1^T\right] + E\left[h_1 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_2^T\right] + E[h_1 f_1 \varepsilon_1 \varepsilon_2^T f_2^T h_2^T] \\ + E\left[h_2 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_1^T\right] + E[h_2 f_1 \varepsilon_1 {\varepsilon_1}^T f_1^T h_2^T] + E\left[h_2 f_1 \varepsilon_1 \varepsilon_2^T f_2^T h_2^T\right] \\ + E\left[h_2 f_2 \varepsilon_2 {\varepsilon_1}^T f_1^T h_1^T\right] + E\left[h_2 f_2 \varepsilon_2 {\varepsilon_1}^T f_1^T h_2^T\right] + E[h_2 f_2 \varepsilon_2 \varepsilon_2^T f_2^T h_2^T] \tag{E.70}$$

Reusing the argument from Equation (E.53) and Equation (E.55) allows eliminating terms containing the noise at different time indexes. Thus:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_1 f_1 \sigma_\varepsilon^2 f_1^T h_1^T + h_1 f_1 \sigma_\varepsilon^2 f_1^T h_2^T + h_2 f_1 \sigma_\varepsilon^2 f_1^T h_1^T \\ + h_2 f_1 \sigma_\varepsilon^2 f_1^T h_2^T + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \tag{E.71}$$

Factoring out some terms:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_1 f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \tag{E.72}$$

Again, performing factorization:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = (h_1 + h_2) f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \tag{E.73}$$

Now, for $m = 3$:

$$Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = \cdots$$

$$= E\left[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2 + h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\right.$$
$$\left.\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2 + h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\right] \tag{E.74}$$

Reusing the result from Equation (E.71) to some extent:

$$Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] =$$
$$= E\left[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right) \cdots\right.$$
$$\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2 + h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T + \cdots$$
$$+ \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T + \cdots$$
$$\left. + \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\right] \tag{E.75}$$

Factoring some alike terms:

$$Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] =$$
$$= E\left[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T \cdots\right. \tag{E.76}$$
$$+ \left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T + \cdots$$
$$+ \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T + \cdots$$
$$\left. + \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\right]$$

Distributing the expectancy operator:

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= \\
&= (h_1 + h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T + \cdots \\
&\quad + E\Big[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T \\
&\qquad + \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T \\
&\qquad + \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\Big]
\end{aligned}
\tag{E.77}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= \\
&= (h_1 + h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T + \ldots \\
&\quad + E\Big[\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\Big] \\
&\quad + E\Big[\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T\Big] \\
&\quad + E\Big[\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)\left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\Big]
\end{aligned}
\tag{E.78}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1 + h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \\
&\quad + E\Big[h_1 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T \\
&\qquad + h_2 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T \\
&\qquad + h_2 f_2 \varepsilon_2 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\Big] \\
&\quad + E\Big[\Big(h_3 f_1 \varepsilon_1 \left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T \\
&\qquad + h_3 f_2 \varepsilon_2 \left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T \\
&\qquad + h_3 f_3 \varepsilon_3 \left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1 + h_2 f_2 \varepsilon_2\right)^T\Big)\Big] \\
&\quad + E\Big[\Big(h_3 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T \\
&\qquad + h_3 f_2 \varepsilon_2 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T \\
&\qquad + h_3 f_3 \varepsilon_3 \left(h_3 f_1 \varepsilon_1 + h_3 f_2 \varepsilon_2 + h_3 f_3 \varepsilon_3\right)^T\Big)\Big]
\end{aligned}
\tag{E.79}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \\
&+ E\left[h_1 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1\right)^T + h_2 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1\right)^T + h_2 f_2 \varepsilon_2 \left(h_3 f_2 \varepsilon_2\right)^T\right] \\
&+ E\left[\left(h_3 f_1 \varepsilon_1 \left(h_1 f_1 \varepsilon_1 + h_2 f_1 \varepsilon_1\right)^T + h_3 f_2 \varepsilon_2 \left(h_2 f_2 \varepsilon_2\right)^T\right)\right] \\
&+ E\left[\left(h_3 f_1 \varepsilon_1 \left(h_3 f_1 \varepsilon_1\right)^T + h_3 f_2 \varepsilon_2\right) \left(h_3 f_2 \varepsilon_2\right)^T \right. \\
&\qquad \left. + h_3 f_3 \varepsilon_3 \left(h_3 f_3 \varepsilon_3\right)^T\right]
\end{aligned}
\tag{E.80}
$$

Evaluating the expectancy operations:

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \\
&+ (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T (h_3)^T + h_2 f_2 \sigma_\varepsilon^2 f_2^T (h_3)^T + h_3 f_1 \sigma_\varepsilon^2 f_1^T (h_1+h_2)^T \\
&+ h_3 f_2 \sigma_\varepsilon^2 f_2^T (h_2)^T + h_3 \left(f_1 \sigma_\varepsilon^2 f_1^T + f_2 \sigma_\varepsilon^2 f_2^T + f_3 \sigma_\varepsilon^2 f_3^T\right) h_3^T
\end{aligned}
\tag{E.81}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2 \sigma_\varepsilon^2 f_2^T h_2^T \\
&+ (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T (h_3)^T + h_2 f_2 \sigma_\varepsilon^2 f_2^T (h_3)^T + h_3 f_1 \sigma_\varepsilon^2 f_1^T (h_1+h_2)^T \\
&+ h_3 f_2 \sigma_\varepsilon^2 f_2^T (h_2)^T + h_3 \left(f_1 \sigma_\varepsilon^2 f_1^T + f_2 \sigma_\varepsilon^2 f_2^T + f_3 \sigma_\varepsilon^2 f_3^T\right) h_3^T
\end{aligned}
\tag{E.82}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1+h_2+h_3)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) \\
&+ (h_1+h_2)\, f_1 \sigma_\varepsilon^2 f_1^T (h_3)^T + h_2 f_2 \sigma_\varepsilon^2 f_2^T (h_2+h_3)^T \\
&+ h_3 f_2 \sigma_\varepsilon^2 f_2^T (h_2+h_3)^T + h_3 \left(f_1 \sigma_\varepsilon^2 f_1^T + f_3 \sigma_\varepsilon^2 f_3^T\right) h_3^T
\end{aligned}
\tag{E.83}
$$

$$
\begin{aligned}
Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (h_1+h_2+h_3)\, f_1 \sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + (h_1+h_2+h_3)\, f_1 \sigma_\varepsilon^2 f_1^T (h_3)^T \\
&+ (h_2+h_3) f_2 \sigma_\varepsilon^2 f_2^T (h_2+h_3)^T + h_3 \left(f_3 \sigma_\varepsilon^2 f_3^T\right) h_3^T
\end{aligned}
\tag{E.84}
$$

$$Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = (h_1+h_2+h_3)\, f_1\sigma_\varepsilon^2 f_1^T (h_1+h_2+h_3)^T + (h_2+h_3) f_2\sigma_\varepsilon^2 f_2^T (h_2+h_3)^T + h_3\left(f_3\sigma_\varepsilon^2 f_3^T\right) h_3^T \tag{E.85}$$

Comparing to the result for case $m = 2$:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = (h_1+h_2) f_1\sigma_\varepsilon^2 f_1^T \left(h_1^T + h_2^T\right) + h_2 f_2\sigma_\varepsilon^2 f_2^T h_2^T \tag{E.86}$$

Recalling a previous definition:

$$H_k = \sum_{i=1}^{k} h_i \tag{E.87}$$

Then:

$$Var\left[\sum_{k=1}^{2} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = H_2 Q_1 H_2^T + (H_2 - H_1)\, Q_2 (H_2 - H_1)^T \tag{E.88}$$

$$Var\left[\sum_{k=1}^{3} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = H_3 Q_1 (H_3)^T + (H_3 - H_1) Q_2 (H_3 - H_1)^T + (H_3 - H_2)\, Q_3 (H_3 - H_2)^T \tag{E.89}$$

For the inductive step hypothesis $m = M$:

$$Var\left[\sum_{k=1}^{M} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = \sum_{k=1}^{M} (H_M - H_{k-1}) Q_k (H_M - H_{k-1})^T \tag{E.90}$$

Then for $m = M + 1$:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] = Var\left[h_{M+1} \sum_{i=1}^{M+1} f_i\varepsilon_i + \sum_{k=1}^{M} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] \tag{E.91}$$

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] =$$
$$= Var\left[h_{M+1} f_{M+1} \varepsilon_{M+1} + h_{M+1} \sum_{i=1}^{M} f_i \varepsilon_i + \sum_{k=1}^{M} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \tag{E.92}$$

By setting:

$$h'_k = \left\{ \begin{array}{c} h_k,\ k < M \\ h_M + h_{M+1},\ k = M \end{array} \right\} \tag{E.93}$$

The two summations in the last expression can be grouped together:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_{M+1} f_{M+1} \varepsilon_{M+1} + \sum_{k=1}^{M} h'_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \tag{E.94}$$

Using the property from Theorem B.1.3 and noting that the term ε_{M+1} does not appear in the second summation, and it is assumed independent from any previous value ε_i, so the cross-covariance term does not appear:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = Var\left[h_{M+1} f_{M+1} \varepsilon_{M+1}\right] + Var\left[\sum_{k=1}^{M} h'_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \tag{E.95}$$

Applying the hypothesis to the term with summations:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_{M+1} Q_{M+1} h_{M+1}^T + \sum_{k=1}^{M} (H'_M - H'_{k-1}) Q_k \left(H'_M - H'_{k-1}\right)^T \tag{E.96}$$

In the last term, it is possible to retrieve H_k from H'_k noting:

$$H'_k = \left\{ \begin{array}{c} H_k,\ k < M \\ H_M + h_{M+1},\ k = M \end{array} \right. \tag{E.97}$$

Which implies for $k \leq M$:

$$\begin{aligned} H'_M - H'_{k-1} &= h_{M+1} + H_M - H_{k-1} \\ &= H_{M+1} - H_{k-1} \end{aligned} \tag{E.98}$$

Replacing this in the summation:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_{M+1} Q_{M+1} h_{M+1}^T + \sum_{k=1}^{M} (H_{M+1} - H_{k-1}) Q_k (H_{M+1} - H_{k-1})^T \tag{E.99}$$

Rewriting the first term with:

$$h_{M+1} = H_{M+1} - H_M \tag{E.100}$$

$$\begin{aligned} Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] &= (H_{M+1} - H_M)\, Q_{M+1} (H_{M+1} - H_M)^T \\ &+ \sum_{k=1}^{M} (H_{M+1} - H_{k-1}) Q_k (H_{M+1} - H_{k-1})^T \end{aligned} \tag{E.101}$$

Then, the outside term can be added to the summation:

$$Var\left[\sum_{k=1}^{M+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{M+1} (H_{M+1} - H_{k-1}) Q_k (H_{M+1} - H_{k-1})^T \tag{E.102}$$

Thus, concluding the inductive step. So for any m:

$$Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = \sum_{k=1}^{m} (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \tag{E.103}$$

This result, albeit simple, does not allow recursive implementation. To derive a recursive formula, one approach is to investigate the increment between steps:

$$\begin{aligned} &Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = \\ &= \sum_{k=1}^{m+1} (H_{m+1} - H_{k-1}) Q_k (H_{m+1} - H_{k-1})^T - \sum_{k=1}^{m} (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \end{aligned} \tag{E.104}$$

Making the summations match the summation span:

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\
&= (H_{m+1} - H_m)\, Q_{m+1} (H_{m+1} - H_m)^T \\
&\quad + \sum_{k=1}^{m} \Big((H_{m+1} - H_{k-1})\, Q_k (H_{m+1} - H_{k-1})^T - (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \Big)
\end{aligned}
\tag{E.105}
$$

Making h_{m+1} explicit:

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] = h_{m+1} Q_{m+1} h_{m+1}^T \\
&+ \sum_{k=1}^{m} \Big((h_{m+1} + H_m - H_{k-1})\, Q_k (h_{m+1} + H_m - H_{k-1})^T - (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \Big)
\end{aligned}
\tag{E.106}
$$

Expanding the term in the summation:

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\
&= h_{m+1} Q_{m+1} h_{m+1}^T + \sum_{k=1}^{m} \Big((h_{m+1})\, Q_k (h_{m+1} + H_m - H_{k-1})^T \\
&\qquad + (H_m - H_{k-1})\, Q_k (h_{m+1} + H_m - H_{k-1})^T - (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \Big)
\end{aligned}
\tag{E.107}
$$

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\
&= h_{m+1} Q_{m+1} h_{m+1}^T \\
&\quad + \sum_{k=1}^{m} \Big((h_{m+1})\, Q_k (h_{m+1})^T + (h_{m+1})\, Q_k (H_m - H_{k-1})^T + (H_m - H_{k-1})\, Q_k (h_{m+1})^T \\
&\qquad + (H_m - H_{k-1})\, Q_k (H_m - H_{k-1})^T - (H_m - H_{k-1}) Q_k (H_m - H_{k-1})^T \Big)
\end{aligned}
\tag{E.108}
$$

Simplifying:

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\
&= h_{m+1} Q_{m+1} h_{m+1}^T \\
&\quad + \sum_{k=1}^{m} \left((h_{m+1})\, Q_k (h_{m+1})^T + (h_{m+1})\, Q_k (H_m - H_{k-1})^T + (H_m - H_{k-1})\, Q_k (h_{m+1})^T \right)
\end{aligned}
\tag{E.109}
$$

Separating into two summations:

$$
\begin{aligned}
&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\
&= h_{m+1} \left(\sum_{k=1}^{m+1} Q_k \right) h_{m+1}^T + \sum_{k=1}^{m} \left((h_{m+1})\, Q_k (H_m - H_{k-1})^T + (H_m - H_{k-1})\, Q_k (h_{m+1})^T \right)
\end{aligned}
\tag{E.110}
$$

Defining the symmetric part and anti-symmetric part operators:

$$
(M)_{\oplus} = \frac{M + M^T}{2} \tag{E.111}
$$

$$
(M)_{\ominus} = \frac{M - M^T}{2} \tag{E.112}
$$

Notice the property:

$$
\begin{aligned}
(A + B)_{\oplus} &= \frac{A + B + (A + B)^T}{2} \\
&= \frac{A + B + A^T + B^T}{2} \\
&= \frac{A + A^T + B + B^T}{2} \\
&= A_{\oplus} + B_{\oplus}
\end{aligned}
\tag{E.113}
$$

Therefore:

$$
\left(\sum A_k\right)_{\oplus} = \sum (A_k)_{\oplus} \tag{E.114}
$$

Thus:

$$\begin{aligned}\sum (BA_k)_{\oplus} &= \left(\sum BA_k\right)_{\oplus} \\ &= \left(B\sum A_k\right)_{\oplus}\end{aligned} \tag{E.115}$$

Notice that:

$$h_{m+1}Q_k(H_m - H_{k-1})^T + (H_m - H_{k-1})\,Q_k(h_{m+1})^T = 2\Big((h_{m+1})\,Q_k(H_m - H_{k-1})^T\Big)_{\oplus} \tag{E.116}$$

Then:

$$\begin{aligned}&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] \\ &= h_{m+1}\left(\sum_{k=1}^{m+1} Q_k\right) h_{m+1}^T + \sum_{k=1}^{m}\Big((h_{m+1})\,Q_k(H_m - H_{k-1})^T + (H_m - H_{k-1})\,Q_k(h_{m+1})^T\Big)\end{aligned} \tag{E.117}$$

$$\begin{aligned}&Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i\varepsilon_i\right] \\ &= h_{m+1}\left(\sum_{k=1}^{m+1} Q_k\right) h_{m+1}^T + 2\left((h_{m+1})\sum_{k=1}^{m} Q_k(H_m - H_{k-1})^T\right)_{\oplus}\end{aligned} \tag{E.118}$$

Looking at this particular term:

$$\sum_{k=1}^{m} Q_k(H_m - H_{k-1})^T = \sum_{k=1}^{m} Q_k H_m^T - \sum_{k=1}^{m} Q_k H_{k-1}^T \tag{E.119}$$

$$\sum_{k=1}^{m} Q_k(H_m - H_{k-1})^T = \left(\sum_{k=1}^{m} Q_k\right) H_m^T - \sum_{k=1}^{m} Q_k H_{k-1}^T \tag{E.120}$$

Now the last summation:

$$\sum_{k=1}^{m} Q_k H_{k-1}^T = Q_1 H_0^T + Q_2 H_1^T + Q_3 H_2^T + \ldots Q_m H_{m-1}^T \tag{E.121}$$

Thus, defining two additional recurrent sequences:

$$H_k = \sum_{i=1}^{k} h_i \tag{E.122}$$

$$\begin{aligned} W_k &= Q_k + W_{k-1} \\ &= \left(\sum_{i=1}^{k} Q_i \right) \end{aligned} \tag{E.123}$$

$$\begin{aligned} X_k &= Q_k H_{k-1}^T + X_{k-1} \\ &= \sum_{k=1}^{m} Q_k H_{k-1}^T \end{aligned} \tag{E.124}$$

Then the variance increment can be given by:

$$\begin{aligned} &Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\ &= h_{m+1} W_{m+1} h_{m+1}^T + \left(h_{m+1} \left(\sum_{k=1}^{m} Q_k \right) H_m^T - h_{m+1} \sum_{k=1}^{m} Q_k H_{k-1}^T \right)_{\oplus} \end{aligned} \tag{E.125}$$

$$\begin{aligned} &Var\left[\sum_{k=1}^{m+1} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] - Var\left[\sum_{k=1}^{m} h_k \sum_{i=1}^{k} f_i \varepsilon_i\right] \\ &= h_{m+1} W_{m+1} h_{m+1}^T + 2\left(h_{m+1} \left(W_m H_m^T - X_m \right) \right)_{\oplus} \end{aligned} \tag{E.126}$$

Which in turn is also a recurrence formula for the variance.

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.