

Attitude Estimation Algorithms and Comprehensive Error Analysis in the Generic
Multi-sensor Integration Strategy

Benjamin Brunson

A Dissertation submitted to the Faculty of Graduate Studies in partial fulfilment of
the requirements for the degree of Doctor of Philosophy.

Graduate Program in Earth and Space Science

York University

Toronto, Ontario

July 2025

© Benjamin Brunson, 2025

Abstract

Modern multi-sensor integrated kinematic position/attitude systems serve a diverse range of industries, including precision agriculture, construction, mobile mapping, and autonomous driving. These industries have diverse needs that must be met by their position/attitude systems, which emphasizes the need for kinematic position/attitude systems that are highly customizable and robust. Lower-cost sensors have become increasingly prevalent in modern position/attitude systems, and accurately modeling the systematic errors in these sensors is of paramount importance for high-accuracy applications.

This dissertation focuses on extending and refining the tools for comprehensive error analysis in the Generic Multi-sensor Integration Strategy (GMIS), with a particular focus on statistical analysis in pre-and post-processing environments. This research leverages the strengths of the GMIS to rigorously characterize the sensor performance and systematic errors of tactical grade MEMS IMU sensors, even in situations where a reference solution is unavailable.

The principal research contributions of this dissertation are as follows:

1. A generalized framework for imposing system state equality constraints in a Kalman Filter, which allows for the direct estimation of the constrained observation, predicted system state, and process noise residuals, along with their associated covariance matrices. This framework opens opportunities for comprehensive error analysis in system state constrained Kalman Filters, including: residual analysis, reliability analysis, and Variance Component Estimation (VCE).
2. The formulation of three unique attitude models under the GMIS framework: the Roll-Pitch-Heading (RPH) model, the Direction Cosine Matrix (DCM) model, and the Quaternion model. This research defines each attitude model's corresponding system model, system state constraints, and IMU observation models within the GMIS. This work uniquely uses the process noise residual vector to conduct VCE to directly compare the performance of each different attitude model. Each of the three attitude models used in this research performed similarly over a road test dataset, and the DCM model in particular exhibited resistance to a sudden trajectory variation that was captured by the IMU.

3. The formulation of Multi-IMU (MIMU) array modeling in the GMIS, allowing for the separate estimation of IMU systematic errors and stochastic properties for every sensor in the MIMU array. This work allows for a detailed, time-varying analysis of constituent IMU sensors' stochastic properties via VCE alongside rigorous Fault Detection and Exclusion (FDE) for each sensor in a MIMU array. None of this is possible under other data fusion methods for MIMU arrays. The *a posteriori* position/attitude standard deviations were compared between a MIMU and Single-IMU (SIMU) solution for the same road test dataset, and the MIMU solution provided an average accuracy improvement of ca. 14-16% in the estimated position, 30% in the estimated roll and pitch, and 40% in the estimated heading.
4. A generalized framework for the pre-analysis of online sensor calibration procedures under the GMIS. The proposed framework uses a candidate trajectory with a defined sensor configuration to estimate the following quantities: minimum significant values for individual systematic error parameters; minimum detectable values for observation biases from each sensor; and observability analysis for each sensor systematic error being modeled in the Kalman Filter. This framework allows for the establishment of best field procedures and sensor configurations without requiring the construction and testing of physical systems. This framework was used to evaluate three different case studies to determine the observability of IMU systematic error states, alongside their minimum estimable values and minimum detectable errors under different calibration maneuvers. Across all three case studies, the accelerometer biases generally had minimum estimable values within 1 cm/s and the gyroscope biases generally had minimum estimable values within 3.5 °/s. Accelerometer/gyroscope scale factor errors and the reference-IMU lever arm vector had their observability much more dependent upon the specific calibration maneuver being used.

This research takes great strides towards establishing the GMIS as a fully matured strategy for sensor integration that has much more modeling flexibility than traditional approaches. The only remaining challenge is addressing the increased computational load of the GMIS relative to the TMIS, and this is a high priority for future work.

This research's contributions lay a strong foundation for future research in adaptive filtering techniques, rigorous fault detection and exclusion, and comprehensive stochastic modeling of the sensors that are being used in a position/attitude system.

Acknowledgements

I would like to extend my sincere gratitude to everyone who has provided me with support and encouragement throughout my time at York University. I will always cherish my experience.

I would first like to thank my supervisor, Jianguo Wang, for the tremendous amount of guidance and support he has offered me throughout my studies. His enthusiasm in the field and passion for learning have been inspiring, and I hope to carry a similar passion for the field into the future.

I would also like to thank the members of my supervisory committee, Prof. Baoxin Hu and Prof. Sunil Bisnath, for their encouragement and advice. Our discussions have been very enlightening, and the breadth of experience and perspective you have brought to the table has greatly benefited my work and me as a person.

I would like to thank Mike Ma for all his work in setting up and testing the system we used to collect the multi-IMU road test dataset.

More broadly, I would like to extend my thanks to all of the other Geomatics Engineering professors at York University: Prof. Pagiatakis, Prof. Jadidi, Prof. Armenakis, and Prof. Sohn. Their mentorship has meant so much to me throughout my studies, and the opportunities I was afforded (even as an undergraduate student) have inspired me to continue exploring this fascinating field of study.

I would like to also thank York University and NSERC for their generous funding of my research.

To my parents, thank you for your unconditional love and support through the years. Your companionship has made my experiences here truly meaningful, and I will always be grateful.

To my siblings, thank you for your friendship and camaraderie. It has been very comforting to know that I always have someone to talk to when I am feeling discouraged. Your support means a lot to me.

This has been a long journey, and I am glad to be sharing this moment with the people who have made it possible.

Table of Contents

Abstract.....	ii
Acknowledgements.....	iv
Table of Contents.....	v
List of Tables	xi
List of Figures	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Goal and Objectives.....	3
1.3 Structure of the Dissertation	5
2 Preliminaries and Related Literature	7
2.1 Coordinate Frames in Positioning and Navigation	7
2.2 Attitude Models in Positioning, Orientation, and Navigation	10
2.2.1 Direction Cosine Matrix (DCM)	11
2.2.2 Roll-Pitch-Heading (RPH).....	11
2.2.3 Quaternion Representation	13
2.3 Kinematics and Rigid Body Motion.....	14
2.3.1 Kinematics of a Non-Rigid Body	15
2.3.2 Kinematics of a Rigid Body	16
2.3.3 Motion in a Rotating Coordinate Frame	16
2.3.4 The Poisson Equation	18
2.3.5 Relating Inertial Measurements to the Navigation Frame	18
2.4 Sensors Used in Integrated Navigation	18
2.4.1 Global Navigation Satellite Systems (GNSS) Receivers	19
2.4.2 Inertial Measurement Units (IMUs)	25

2.4.3	Other Sensors.....	31
2.5	Kalman Filters	33
2.5.1	Extended Kalman Filter (EKF).....	33
2.5.2	Other Filtering Techniques.....	41
2.6	Multi-Sensor Integration Strategies	44
2.6.1	Classification of Multi-sensor Integration Architectures	44
2.6.2	The Traditional Multi-Sensor Integration Strategy (TMIS).....	53
2.6.3	Generic Multi-Sensor Integration Strategy (GMIS)	58
2.6.4	Comparison Between the TMIS and the GMIS.....	61
3	Formulation of Discrete Kalman Filtering with Constraints – A Generic Framework for Comprehensive Error Analysis	63
3.1	Preamble.....	63
3.2	Existing Methods for Imposing State Equality Constraints	64
3.2.1	Prominent Methods for Imposing State Equality Constraints	64
3.2.2	General Remarks and Alternative Approaches	67
3.3	Alternative Derivation of Discrete Kalman Filtering with Constraints	69
3.3.1	Formulation of the Problem.....	69
3.3.2	Row-Reduction of the System of Normal Equations.....	72
3.3.3	Relating the Constrained Solution to the Unconstrained Solution.....	75
3.3.4	The Constrained System State and Process Noise Covariance Matrices	76
3.3.5	The Process Noise, Predicted System State, and Observation Residual Vectors and their Associated Covariance Matrices	78
3.3.6	Extended Kalman Filter with State Constraints	82
3.4	Summary of the System State-Constrained Kalman Filter	83
3.4.1	Kalman Filter Equations Without System State Constraints	85

3.4.2	Enforcing System State Constraints	86
3.5	Comprehensive Error Analysis in the System State Constrained Kalman Filter	87
3.5.1	Residual Analysis	89
3.5.2	Reliability Analysis	90
3.5.3	Variance Component Estimation	93
3.6	Numerical Example: Planar Motion of a Ground Vehicle	96
3.6.1	Formulation of the Problem	97
3.6.2	Estimated Trajectory for the Simulated Dataset	99
3.6.3	Estimated Residuals for the Simulated Dataset	101
3.6.4	Variance Component Estimation for the Simulated Dataset	106
3.7	Summary	110
4	The Position and Attitude (PosAtt) Engine	112
4.1	Preamble	112
4.2	Generic Multi-Sensor Integration Strategy	114
4.3	Software Structure	117
4.4	Data Input Module	118
4.4.1	INI File Input	119
4.4.2	IMU Data File Input	122
4.4.3	GPS Data File Input	122
4.5	Initialization Module	122
4.5.1	Selecting the GPS “Base” Satellites	123
4.5.2	Coarse Alignment	123
4.6	Positioning Module	124
4.7	Data Output Module	126
4.8	Summary	127

5	Three Attitude Models and their Characterization in the Generic Multi-Sensor Integration Strategy	128
5.1	Preamble.....	128
5.2	Partitioning of the EKF Matrices.....	129
5.3	Position System Model in the GMIS.....	132
5.4	IMU Observation Equations	133
5.5	Attitude Models in the GMIS.....	133
5.5.1	Roll-Pitch-Heading (RPH) Model	133
5.5.2	Direction Cosine Matrix (DCM) Model.....	135
5.5.3	Quaternion Model.....	141
5.6	Comprehensive Error Analysis.....	144
5.7	Processing Workflow	147
5.8	Comparative Analysis	149
5.8.1	Definition of the Test Dataset	149
5.8.2	Directly Comparing System State Estimates	151
5.8.3	Analyzing Observation Residuals	154
5.8.4	Analyzing Process Noise Residuals.....	160
5.8.5	Analyzing Variance Component Estimates.....	165
5.8.6	Analyzing the System's Response to a Jolt.....	170
5.9	Summary.....	178
6	Modeling of IMU Arrays Under the Generic Multi-Sensor Integration Strategy	181
6.1	Preamble.....	181
6.2	Position and Attitude System Models for IMU Arrays	185
6.3	Partitioning of the EKF Matrices.....	186
6.4	MIMU Array Observation Equations in the GMIS	189

6.4.1	Modeling Accelerometer Observations in the MIMU Array	189
6.4.2	Modeling Gyroscope Observations in the MIMU Array.....	191
6.4.3	Modeling Atypical IMU Sensors	192
6.5	Comprehensive Error Analysis.....	196
6.6	Processing Workflow	197
6.7	Land Vehicle Test Dataset and Analysis of Results.....	199
6.7.1	Definition of the Road Test Dataset.....	199
6.7.2	Estimating Sensor Systematic Errors and Measurement Residuals.....	203
6.7.3	Characterizing Sensor Performance Using Variance Component Estimation.....	209
6.7.4	Comparing Multi-IMU Results to Single-IMU Results	213
6.8	Summary.....	218
7	Analytical Framework for Online Calibration of Sensor Systematic Errors Under the Generic Multisensor Integration Strategy.....	221
7.1	Preamble.....	221
7.2	Analytical Framework for Online Sensor Calibration	223
7.2.1	Observability and Degree of Observability of States	224
7.2.2	Minimum Estimable Values of Systematic Error States	228
7.2.3	Directly Comparing Covariance Matrices.....	231
7.2.4	Minimum Detectable Values of Systematic Errors as Outliers after the Internal Reliability	232
7.2.5	Workflow of the Analytical Pre-Analysis Framework.....	234
7.3	Case Studies.....	235
7.3.1	Case #1: No Calibration Maneuvers.....	236
7.3.2	Case #2: Simulated Trajectories with a Circular Motion Maneuver	247
7.3.3	Case #3: Simulated Trajectories with Varying System Roll and Pitch Maneuver.....	255

7.4	Summary.....	264
8	Conclusions and Future Work.....	266
8.1	Conclusions.....	266
8.2	Future Work.....	270
	References	274

List of Tables

Table 2.4.1: The effects that the single- and double-differencing process has on the different sources of error that affect GPS observations [Van Sickle, 2023]. Note that multipath error is environment-dependent, but can potentially be exacerbated by the single- and double-differencing process. .	24
Table 2.4.2: Classification of IMU performance grades [Advanced Navigation, 2025; VectorNav, 2025] .	26
Table 2.6.1: Comparison between the Direct and Indirect Kalman Filters in positioning and navigation applications.....	48
Table 2.6.2: Comparison of some of the unique sensor attributes of different positioning sensors.....	49
Table 2.6.3: Comparison between the traditional sensor integration strategy and the GMIS.	62
Table 5.8.1: Average estimated horizontal and vertical standard deviations for each attitude model over the kinematic portion of the dataset.....	151
Table 5.8.2: The RMSE between the three-dimensional estimated positions for each attitude model, expressed in cm. These values were calculated over the kinematic portion of the dataset.	152
Table 5.8.3: Average estimated standard deviations in the estimated system orientation in the roll/pitch and heading for each attitude model over the kinematic portion of the dataset.	152
Table 5.8.4: RMSE between the attitude estimates for each attitude model, expressed in arcminutes. These values were calculated over the kinematic portion of the dataset.	153
Table 5.8.5: Overall estimated standard error factors for each type of measurement, and for all observations together. Results are presented separately for each attitude model, for comparison. Each standard error factor was evaluated solely over the kinematic portion of the dataset.....	167
Table 5.8.6: Overall estimated a posteriori standard deviations for each type of measurement. A posteriori standard deviations were estimated by scaling the a priori standard deviation estimate of the observation accuracy by its associated standard error factor from Table 5.8.5.	168
Table 5.8.7: Overall estimated standard error factors for each process noise component related to the system's motion. Separate standard errors were estimated for the components of the process noise vector that were pertaining to the system's linear motion and those pertaining to the system's angular motion. Note that the components of the process noise vector related to the	

wander of IMU systematic errors and float GPS ambiguity estimates are not included in this table, but did contribute to the overall standard error estimate.....	168
Table 5.8.8: Overall estimated a posteriori estimates of the process noise jerk and angular acceleration components, calculated by scaling their assumed a priori values by the associated standard error factors in Table 5.8.7.	169
Table 5.9.1: Overview of the different attitude models in the GMIS.	179
Table 6.7.1: The global a posteriori standard deviations for each observation type in the MIMU array. These values were calculated by scaling the a priori standard deviation estimate for the accelerometer and gyroscope observations by the corresponding global estimate of the standard error factor. Values were estimated separately for each IMU in the MIMU array.....	213
Table 6.7.2: Characterization of the MIMU solution performance improvement compared to the SIMU solution performance, expressed as a percentage improvement in the estimated a posteriori standard deviations of corresponding quantities.....	217
Table 7.3.1: The technical specifications of the sensor measurements used in the case studies.....	235
Table 7.3.2: Results of evaluating the eigenvalues and eigenvectors of the normalized state covariance matrix after (7.5).....	238
Table 7.3.3: Summary of the minimum estimable values of the IMU systematic error states (Case #1). Augmented states that were not estimated due to their unobservability are marked with “-”. ..	242
Table 7.3.4: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.....	243
Table 7.3.5: Final values of the a posteriori minimum estimable IMU systematic error estimates for the road test dataset. Augmented states that were not estimated due to their unobservability are marked with “-”.	246
Table 7.3.6: Results of evaluating the eigenvalues and eigenvectors of the normalized state covariance matrix.....	249
Table 7.3.7: Minimum estimable values of the augmented systematic error states under circular motion with a radius of 30 m and a tangent velocity of 40 km/h (3 repetitions of the circular motion and	

15 repetitions). Augmented states that were not estimated due to their unobservability are marked with “-”.	253
Table 7.3.8: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.....	255
Table 7.3.9: The observability results after the eigenvalues and eigenvectors of the normalized state covariance matrix.	258
Table 7.3.10: The minimum estimable values of the augmented IMU systematic error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (after 3 repetitions and after 15 repetitions of the tilting maneuver). Augmented states that were not estimated due to their unobservability are marked with “-”.	262
Table 7.3.11: The minimum detectable values of the IMU accelerometer and gyroscope systematic errors as observation outliers, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.	264

List of Figures

Figure 2.1.1: An overview of the coordinate frames used in this research, arranged in order of increasing mobility.....	8
Figure 2.1.2: The relationship between the ECI and ECEF coordinate frames. Image courtesy of [Grewal et al, 2007].....	9
Figure 2.3.1: Diagram illustrating the lack of deformation in rigid body motion. Note that although the displacement vectors PP' , QQ' , and RR' are different from one another, the relative positions between P , Q , and R are the same as the relative positions between P' , Q' , and R'	15
Figure 2.4.1: The signal received from each GNSS satellite may be used to infer the range between it and the receiver and used for multilateration. Diagram courtesy of [Lotfy et al, 2020].	19
Figure 2.4.2: Diagram illustrating the ambiguity problem for carrier phase observables. Note that in each of the 3 situations, the fractional wavelength is the same, even though the total length of each wave is different.	21
Figure 2.4.3: Diagram illustrating the generation of single- and double-differenced GNSS observations, adapted from [Souza et al, 2009].	23
Figure 2.4.4: Illustration of the operating principles of spring-mass accelerometers. (a) shows a linear spring-mass accelerometer, and (b) shows a torsional spring-mass accelerometer. Images courtesy of [Kempe, 2011]. Note that in (b), the springs are not arranged symmetrically about the center of gravity of the proof mass. If this were not the case, only angular motion would cause the proof mass to deflect.....	28
Figure 2.4.5: Illustration of the operating principles of rigid-body MEMS gyroscopes. (a) shows a linear rigid-body gyroscope, where any rotation about axis e_3 produces a Coriolis force along the sensing axis e_1 for a proof mass vibrating along the drive axis e_2 . (b) shows a rotatory rigid-body gyroscope, where rotation about axis e_2 produces oscillations about axis e_1 for a proof mass oscillating about axis e_3 . Images courtesy of [Kempe, 2011].	28
Figure 2.6.1: The basic workflow for the open-loop, feed-forward INS using an Indirect KF. Adapted from [Tan et al, 2007; Sirtkaya et al, 2013].	47

Figure 2.6.2: The basic workflow for the closed-loop, feed-back INS using an Indirect KF. Adapted from [Tan et al, 2007].	47
Figure 2.6.3: Workflow of the snapshot fusion integration architecture [Groves, 2013].	51
Figure 2.6.4: Workflow of the cascade fusion integration architecture [Groves, 2013].	51
Figure 2.6.5: Workflow of the centralized fusion integration architecture [Lashley and Bevly, 2013; Breuer et al, 2016; Konrad et al, 2017].	52
Figure 2.6.6: Workflow detailing a specific realization of the federated fusion integration architecture.	52
Figure 2.6.7: Workflow describing the loosely coupled GNSS/IMU integration under the traditional integration strategy, as per [Scherzinger, 2000].	54
Figure 2.6.8: Workflow describing the tightly coupled GNSS/IMU integration under the traditional integration strategy, adapted from [Scherzinger, 2000; Quinchia et al, 2013; Deepika and Arun, 2018; Hamidi et al, 2020; Yi et al, 2022].	55
Figure 2.6.9: The general structure of an ANN for positioning and navigation. Note that while there is only one hidden layer in this ANN, multiple hidden layers may be included depending on the implementation [Jwo et al, 2023].	58
Figure 2.6.10: The workflow of the Generic Multi-Sensor Integration Strategy (GMIS), as described in [Qian, 2017].	61
Figure 3.4.1: A workflow diagram describing the overall processing flow for the system state-constrained EKF. Note that the system state constraints are enforced after the standard Kalman Filter equations have been applied.	84
Figure 3.6.1: The trajectory used to simulate positioning data for the processing example in this section. Note that the motion of the vehicle is constrained by the planar road surface, and in this processing example, the entire local road network fits well to a planar surface.	97
Figure 3.6.2: Simulated position observations for the ground vehicle trajectory. Position observations were generated with a standard deviation of 1.0 metres, with no correlation between x-, y-, and z-component measurements.	98
Figure 3.6.3: Estimated trajectory for the simulated positioning observations.	99
Figure 3.6.4: Position profiles for the estimated trajectory for the simulated data.	100

Figure 3.6.5: Velocity profiles for the simulated data.	100
Figure 3.6.6: Acceleration profiles for the simulated data.	101
Figure 3.6.7: Standardized observation residuals for the simulated dataset after the application of the system state-constrained Kalman Filter.	102
Figure 3.6.8: Standardized process noise residuals for the simulated dataset after the application of the system state-constrained Kalman Filter.	102
Figure 3.6.9: Standardized residuals of the predicted system state vector for the simulated dataset after the application of the system state-constrained Kalman Filter.	103
Figure 3.6.10: Histogram of the standardized observation residuals. Standard normal distribution shown in red for reference.....	103
Figure 3.6.11: Histogram of the standardized process noise residuals. Standard normal distribution shown in red for reference. Note the sharp peak in the histograms near the 0 mark.	104
Figure 3.6.12: Histogram of the standardized residuals of the predicted system state vector. Standard normal distribution shown in red for reference. Note the sharp peak in the histograms near the 0 mark.	104
Figure 3.6.13: Estimated redundancy contributions for the predicted system state, observation, and process noise vectors in the system state-constrained Kalman Filter over the simulated dataset.	106
Figure 3.6.14: Epoch-wise estimated standard error components for the overall observation vector over the simulated dataset.....	108
Figure 3.6.15: Estimated standard error components for the observation vector, evaluated using different time window sizes.	109
Figure 3.6.16: Estimated standard error components for the process noise vector, evaluated using different time window sizes.	109
Figure 3.6.17: Estimated standard error components for the system state vector, evaluated using different time window sizes.	110
Figure 4.3.1: General workflow describing the interactions between the four primary modules of the PosAtt Engine.....	118

Figure 4.6.1: Workflow describing the processing module of the PosAtt Engine.	125
Figure 5.7.1: The modified processing workflow to accommodate different attitude models in the PosAtt Engine.	148
Figure 5.8.1: Horizontal trajectory of the road test dataset used to validate the different attitude models used in this research. Coordinates are local geodetic coordinates relative to the PosAtt system's starting location.	149
Figure 5.8.2: System velocity profile over the road test dataset. The velocity values are expressed in the navigation frame local geodetic coordinates.	150
Figure 5.8.3: System acceleration profile over the road test dataset. The acceleration values are expressed in the navigation frame local geodetic coordinates.	150
Figure 5.8.4: L1 C/A residuals for each attitude model.	155
Figure 5.8.5: Histogram of L1 C/A standardized residuals. Standard normal distribution superimposed for reference.	155
Figure 5.8.6: L1 Carrier Phase residuals for each attitude model.	156
Figure 5.8.7: Histogram of L1 Carrier Phase standardized residuals. Standard normal distribution superimposed for reference.	156
Figure 5.8.8: L2 Carrier Phase residuals for each attitude model.	157
Figure 5.8.9: Histogram of L2 Carrier Phase standardized residuals. Standard normal distribution superimposed for reference.	157
Figure 5.8.10: Accelerometer specific force residuals for each attitude model.	158
Figure 5.8.11: Histogram of accelerometer standardized residuals. Standard normal distribution superimposed for reference.	158
Figure 5.8.12: Gyroscope angular rate residuals for each attitude model.	159
Figure 5.8.13: Histograms of gyroscope standardized residuals. Standard normal distribution superimposed for reference.	159
Figure 5.8.14: System jerk residuals (process noise of the system's linear position).	160
Figure 5.8.15: Histograms of the standardized system jerk process noise residuals.	161

Figure 5.8.16: Estimated process noise residuals for the attitude parameters in each of the attitude models. Each colour represents a different second-order time derivative in the associated attitude model, and none of the values are directly comparable across attitude models.	162
Figure 5.8.17: Histograms of the standardized attitude process noise residuals for the RPH attitude model.	162
Figure 5.8.18: Histograms of the standardized attitude process noise residuals for the DCM attitude model.	163
Figure 5.8.19: Histograms of the standardized attitude process noise residuals for the quaternion attitude model.	163
Figure 5.8.20: The standard error factor plots for the observation residual vector in the RPH, DCM, and Quaternion attitude models (moving window: 20 seconds).	166
Figure 5.8.21: The standard error factor plots for the process noise residual vector in the RPH, DCM, and Quaternion attitude models (moving window: 20 seconds).	166
Figure 5.8.22: IMU raw measurement plots over the duration of the road test dataset. Note the sharp jumps in both the accelerometer and gyroscope measurements at approximately $t = 63987$ s. .	171
Figure 5.8.23: Estimated observation standard error factors for each attitude model (moving window: 20 s). Note the sharp jump in all attitude models around the time $t = 63900$ s.	172
Figure 5.8.24: Estimated gyroscope observation standard error factors for each attitude model (moving window: 20 s).	172
Figure 5.8.25: Estimated accelerometer observation standard error factors for each attitude model (moving window: 20 s).	173
Figure 5.8.26: Estimated L1 C/A observation standard error factors for each attitude model (moving window: 20 s).	173
Figure 5.8.27: Estimated L1 carrier phase observation standard error factors for each attitude model (moving window: 20 s).	174
Figure 5.8.28: Estimated L2 carrier phase observation standard error factors for each attitude model (moving window: 20 s).	174

Figure 5.8.29: Estimated standard error factors for the overall observation vector (moving window: 1 s).	175
Figure 5.8.30: Estimated system jerk standard errors for each attitude model (moving window: 20 s). 177	
Figure 5.8.31: Estimated second time-derivative of attitude parameters' standard errors for each attitude model (moving window: 20 s).	177
Figure 6.6.1: The modified processing workflow to accommodate an arbitrary number of IMU sensors in a MIMU array.....	198
Figure 6.7.1: A top-down view of the trajectory of the kinematic dataset. Coordinates are expressed in the local geodetic coordinate system with an origin coincident to the vehicle's starting location.	200
Figure 6.7.2: The velocity profile for the kinematic dataset. The velocity components are expressed in the navigation frame's local geodetic coordinates.	200
Figure 6.7.3: The acceleration profile for the kinematic dataset. The acceleration components are expressed in the navigation frame's local geodetic coordinates.	201
Figure 6.7.4: The roll, pitch, and heading profiles for the kinematic dataset. The attitude is presented relative to the navigation frame's local geodetic coordinates.	201
Figure 6.7.5: The estimated first-order time derivatives for the roll, pitch, and heading values for the kinematic dataset. These values are expressed relative to the navigation frame's local geodetic coordinates.	202
Figure 6.7.6: The estimated accelerometer specific force residuals for each IMU in the MIMU array over the kinematic test dataset.	203
Figure 6.7.7: The estimated gyroscope angular rate residuals for each IMU in the MIMU array over the kinematic test dataset.	204
Figure 6.7.8: Histograms of the standardized residual values for the accelerometer specific force observations. The standard normal distribution is superimposed for reference.	205
Figure 6.7.9: Histograms of the standardized residual values for the gyroscope angular rate observations. The standard normal distribution is superimposed for reference.	205

Figure 6.7.10: Histograms of the standardized L1 C/A, L1 Carrier Phase, and L2 Carrier Phase residuals over the entire study period. The standard normal distribution is superimposed for reference..	206
Figure 6.7.11: The estimated accelerometer biases for the individual accelerometers in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.....	207
Figure 6.7.12: The estimated accelerometer scale factor errors for the individual accelerometers in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.	207
Figure 6.7.13: The estimated gyroscope biases for the individual gyroscopes in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.....	208
Figure 6.7.14: The estimated gyroscope scale factor errors for the individual gyroscopes in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.....	208
Figure 6.7.15: The estimated overall standard error factor for the MIMU-integrated system (moving window: 20s).	211
Figure 6.7.16: The estimated standard error factors for the specific force measurements of each IMU's accelerometer triad in the MIMU-integrated system (moving window: 20s).	211
Figure 6.7.17: The estimated standard error factors for the specific force measurements of each IMU's gyroscope triad in the MIMU-integrated system (moving window: 20s).	212
Figure 6.7.18: The ratio of the a posteriori MIMU position standard deviations to the a posteriori SIMU position standard deviations. The horizontal dashed lines show the average ratios for the component of the associated colour.....	214
Figure 6.7.19: The ratio of the a posteriori MIMU attitude standard deviations to the a posteriori SIMU attitude standard deviations. The horizontal dashed lines show the average ratios for the quantity of the associated colour.	215
Figure 6.7.20: The ratio of the a posteriori MIMU attitude time derivative standard deviations to the a posteriori SIMU attitude time derivative standard deviations. The horizontal dashed lines show the average ratios for the quantity of the associated colour.....	216
Figure 6.7.21: The estimated overall standard error factors for the SIMU-integrated system (moving window: 20s).	218

Figure 7.2.1: Workflow for estimating the specific quantities used in the proposed framework using the measurement models, system models, and output of the Kalman Filter pre-analysis routine.....	234
Figure 7.3.1: The top-down view of the trajectory used in Case #1 (left) and the velocity profile of the trajectory used in Case #1 (right). The components of the velocity are provided in the forward-right-down vehicle frame.	236
Figure 7.3.2: The acceleration profile of the trajectory used in Case #1 (left) and the attitude profile of the trajectory used in Case #1 (right). The components of the acceleration are provided in the forward-right-down vehicle frame.	237
Figure 7.3.3: The observability of the systematic errors in the accelerometers (left: biases; right: scale factor errors).....	239
Figure 7.3.4: The observability of the systematic errors in the gyroscopes (left: biases; right: scale factor errors).	240
Figure 7.3.5: The observability of the lever arm components.....	240
Figure 7.3.6: The minimum estimable systematic errors in the accelerometers (left: biases; right: scale factor errors).....	241
Figure 7.3.7: The minimum estimable systematic errors in the gyroscopes (left: biases; right: scale factor errors).	241
Figure 7.3.8: The minimum estimable lever arm components for the trajectory.	241
Figure 7.3.9: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, using a variety of window sizes, ranging from 0.4 seconds to the entire dataset.	243
Figure 7.3.10: Ratio between the a priori and a posteriori standard deviations of the accelerometer bias estimates (left) and scale factor estimates (right).....	245
Figure 7.3.11: Ratio between the a priori and a posteriori standard deviations of the gyroscope bias estimates (left) and scale factor error estimates (right).	245
Figure 7.3.12: Ratio between the a priori and a posteriori standard deviations of the lever arm component estimates.	246

Figure 7.3.13: The minimum estimable values of the augmented accelerometer bias states (left) and scale factor error states (right) under circular motion (speeds ranging from 20 - 80 km/h and radii ranging from 10 - 200 m).....	247
Figure 7.3.14: The minimum estimable values of the augmented gyroscope bias states (left) and scale factor error states (right) under circular motion (speeds ranging from 20 – 80 km/h and radii ranging from 10 – 200 m).	248
Figure 7.3.15: The observability of the systematic errors in the accelerometers (left: biases; right: scale factor errors).....	250
Figure 7.3.16: The observability of the systematic errors in the gyroscopes (left: biases; right: scale factor errors).	250
Figure 7.3.17: The observability of the lever arm components.....	251
Figure 7.3.18: Minimum estimable values of the augmented accelerometer bias states (left) and scale factor states (right) for circular motion with a radius of 30 m and a tangent velocity of 40 km/h. Vertical red grid lines are aligned to the starting position of each circle.....	251
Figure 7.3.19: Minimum estimable values of the gyroscope bias states (left) and scale factor states (right) for circular motion with a radius of 30 m and a tangent velocity of 40 km/h. Vertical red grid lines are aligned to the starting position of each circle.	252
Figure 7.3.20: The estimated normalized covariance comparison measures for circular motion with a radius of 30 metres at different vehicle speeds. The covariance comparison measure is <i>cab</i> , with <i>Da</i> determined using Speed a (X-axis) and <i>Db</i> determined using Speed b (Y-axis).....	254
Figure 7.3.21: The minimum estimable values of the augmented accelerometer bias states (left) and scale factor error states (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).	256
Figure 7.3.22: The minimum estimable values of the augmented gyroscope bias states (left) and scale factor error states (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).....	256
Figure 7.3.23: The minimum estimable values of the augmented lever arm states (left) and the absolute lever arm length (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).....	257

Figure 7.3.24: The observability of the augmented systematic error states for the accelerometers (left: biases; right: scale factor errors).	259
Figure 7.3.25: The observability of the augmented systematic error states for the gyroscopes (left: biases; right: scale factor errors).	260
Figure 7.3.26: The observability of the augmented lever arm states.	260
Figure 7.3.27: The minimum estimable values of the augmented accelerometer systematic error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).	261
Figure 7.3.28: The minimum estimable values of the augmented gyroscope error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).	261
Figure 7.3.29: The minimum estimable values of the augmented lever arm states (left) and length of the absolute lever arm vector (right) under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).	262
Figure 7.3.30: The estimated normalized covariance comparison measures for the tilting maneuver with a maximum tilt angle of 50° at different rotation rates. The covariance comparison measure is cab , with $\mathbf{Da}$ determined using Rotation Rate a (X-axis) and $\mathbf{Db}$ determined using Rotation Rate b (Y-Axis).	263

1 Introduction

Positioning and navigation technologies form a crucial component of many diverse modern scientific and industrial research and development applications, including geospatial infrastructure, precision agriculture, mobile mapping systems, autonomous driving systems, warehouse asset management systems, and wearable medical technology. Positioning needs vary significantly between different areas of application, and it is essential to have a robust, purpose-built positioning/attitude (PosAtt) system underpinning each of these technologies.

Depending on the application, the coverage requirements of a PosAtt system can vary from the entire Earth to a single room, with accuracy requirements ranging from tens of metres to the millimetre level. The system dynamics a PosAtt system is required to model can range from a slow-moving terrestrial robot to supersonic planes and munitions. Given such diverse needs, a one-size-fits-all approach to the development of positioning/navigation algorithms is not feasible. Particularly in the realm of high-precision positioning applications, it is valuable to be able to tweak any part of the core positioning algorithms to maximize the performance of a multi-sensor integrated PosAtt system.

1.1 Motivation

To date, positioning and navigation applications are dominated by the use of what will be referred to in this dissertation as the Traditional Multi-Sensor Integration strategy (TMIS). This strategy typically has a single IMU at its core and uses its “core” sensor to predict the PosAtt system’s kinematic trajectory by conducting the free inertial navigation calculation (also called dead reckoning). The free inertial solution is then used to simulate measurements corresponding to any “aiding” sensors (e.g. GNSS receivers, odometers, cameras, etc.) for generating “error measurements”. The system states being estimated are modeled as “error states” that correct the IMU-only solution, and the error measurements corresponding to the aiding sensors are used to update these error states in the Kalman Filter. The TMIS has been very successful in many areas of application, but does have some limitations arising from its development under more restrictive hardware requirements and more limited availability of mobile processing power, especially when using low-cost sensors.

In recent years, Artificial Intelligence (AI) based estimation techniques for positioning and attitude estimation have been of particular interest to researchers [Li et al, 2020; Chen and Pan, 2025]. These approaches have shown promise in improving the accuracy of sensor integration in complex

environments, but generally provide limited insight into a multi-sensor integrated system's constituent sensors [Chen and Pan, 2025] and rely on having training data that accurately represents the full range of realistic scenarios that a PosAtt system can experience [Li et al, 2020].

The TMIS generally relies on making several simplifying assumptions regarding the system's kinematic models and sensor error models, particularly when using non-standard kinematic models as is necessary when using different methods of attitude estimation, and when using non-standard sensor configurations as is the case when using arrays of IMU sensors. As such, the TMIS has limitations when it comes to fully characterizing individual positioning/orientation sensor errors and when it comes to characterizing different system kinematic models. Moreover, the TMIS is particular to specific sensor configurations, and integrating other positioning/orientation sensors requires potentially extensive modifications to the overall workflow.

Thoroughly characterizing the performance of a multi-sensor integrated positioning/navigation system can be very important in two environments that are focused on in this research:

- i. A system design environment, where it is valuable to determine optimal sensor configurations to maximize the reliability of any estimated PosAtt solutions. In such environments, accommodating and assessing diverse sensor configurations is essential. The TMIS offers limited insight in a system design environment and as a consequence many industry leaders do not focus on pre-analysis tasks. Providing a clear framework for evaluating system configuration before field testing would open new avenues for analysis that have not traditionally been included in system design procedures.
- ii. A post-processing environment, where comprehensively characterizing system performance is useful in diagnosing problems in the sensor platform when navigating particularly challenging environments. The TMIS offers limited insight into sensor-specific performance for multi-sensor systems, and enabling a more granular approach to analyzing system performance would put more powerful analytical tools in the hands of both industry professionals and end users.

Statistical analysis conducted under the TMIS is typically focused on interpreting and evaluating the system innovation vector in the Kalman Filter. This is very efficient for performing statistical analysis and fault detection in real-time but does also come with limitations. Of particular concern is the fact that, although there are three statistically independent sources of random error in Kalman Filtering (the previously estimated system state vector, the process noise vector, and the observation noise vector),

these sources of errors get mixed together in the system innovation vector, in addition to the correlation introduced via the generation of error measurements. Practically speaking, this limits the scope and quality of the conclusions that may be drawn from the statistical analysis. A robust and adaptable comprehensive error analysis must therefore extend beyond using only the system innovation vector and the error measurements.

This research is focused on exploring and developing the adaptable and robust comprehensive error analysis tools that enable more flexibility and rigour in both system design and post-processing workflows. This is approached through the use, extension, and enhancement of the Generic Multi-sensor Integration Strategy (GMIS) and the framework for comprehensive error analysis beyond system innovations in Kalman filtering, which were formalized in [Wang, 1997, 2008, 2009; Wang et al, 2015, 2021, 2022; Qian et al, 2015, 2016; Qian, 2017]. The GMIS has additionally been applied in [Yu et al, 2018a, 2018b]. The GMIS is unique in its integration approach, not relying on a core positioning sensor but instead modeling all positioning/orientation observations directly in the positioning Kalman Filter. Directly modeling all positioning/orientation observations in the positioning Kalman Filter allows for much more thorough analysis of their systematic errors and stochastic properties through time, which is essential to the core objective of this research. This is particularly important when modeling observations from IMU sensors, and the GMIS was found to significantly improve solution drift over the TMIS during a GNSS outage in [Kun, 2017].

1.2 Goal and Objectives

The main goal of this research aims at extending and refining the comprehensive error analysis tools in the GMIS that allow for more flexible and rigorous system design and post-processing workflows. This increased flexibility and rigour is approached from two main perspectives:

- i. Extending and exploring the framework for comprehensive error analysis in discrete Kalman filtering and its implementation under the GMIS. This includes re-evaluating the roles that comprehensive error analysis can play for a PosAtt system, and exploring the breadth of possible use cases for the quantities associated with comprehensive error analysis. This research specifically surpasses the use of system innovations for characterizing the performance of a PosAtt system, instead focusing on analysis that allows for the separation of quantities representing the noise in the observation, process noise, and predicted system state vectors.

- ii. Further extending observation and system models in the GMIS in ways that allow for them to be more fully characterized in Kalman filtering. Ensuring flexibility in how sensors are accounted for in Kalman filtering provides more depth in the conclusions that may be drawn from a dataset and allows for more insight into system performance, even prior to any data collection procedures.

These two broad opportunities to improve solution quality are explored through the following specific research objectives:

- i. Extending the algorithmic framework for comprehensive error analysis in Discrete Kalman Filtering with state equality constraints. There are many circumstances under which (typically) geometric constraints may be naturally imposed on the state vector in Kalman filtering. Applying constraints is an issue that has been well-explored in the literature, even establishing a broad consistency across different methods of imposing system state equality constraints. What is missing from the literature is the ability to estimate the quantities required for comprehensive error analysis, as well as a standardization of the process of applying state equality constraints. An alternate derivation of the state equality constrained Kalman Filter has been successfully achieved [Wang and Brunson, 2022], the outcome of which allows for the estimation of the quantities demanded in much of the analysis conducted in this dissertation.
- ii. Systematically studying three fundamental attitude representations under the GMIS and directly comparing their solutions using the powerful error analysis tools that are enabled by the GMIS [Brunson and Wang, 2023]. The attitude models under consideration in this dissertation include the Roll-Pitch-Heading (RPH) representation, the Direction Cosine Matrix (DCM) representation, and the Quaternion representation. Each of these attitude models were formulated in the GMIS along with any associated geometric constraints to avoid over-parameterization in the state vector. The comprehensive error analysis tools enabled by the GMIS were used to directly compare the solution quality for each attitude model over the same dataset using Variance Component Estimation (VCE). This work is the first time these models have been defined in the GMIS, and the first time there has been a direct comparison of their respective PosAtt solutions without needing to convert between different attitude representations. This work demonstrates the breadth of the GMIS' kinematic modeling capabilities, with an emphasis on the detailed insight that the GMIS' comprehensive error analysis tools provide.

- iii. Systematically studying the integration of data from Multi-IMU (MIMU) arrays to provide insight into the systematic errors and stochastic properties of each individual sensor in a MIMU array [Brunson and Wang, 2024]. The key challenge in integrating data from a MIMU array lies in how to account for the individual biases and scale factor errors present in the observations from each constituent IMU sensor. The IMU observation equations were generalized for accommodating multiple IMU sensors in the GMIS, and the GMIS' comprehensive error analysis tools were used to characterize and compare the performance of each sensor in a MIMU array. This work is the first time that the individual IMUs in a MIMU array have been able to be directly modeled and have their time-varying performance assessed. This work demonstrates the breadth of the GMIS' observation modeling capabilities and shows the positioning sensor-specific stochastic information that may be determined through the GMIS' comprehensive error analysis.
- iv. Further enhancing the GMIS' potential for pre-analysis by exploring its use for establishing best practices for online sensor calibration. This research established a framework for the pre-analysis of online sensor calibration procedures, including assessments of the state observability, minimum estimable systematic error parameters, minimum detectable biases in sensor observations, and means of directly comparing multiple covariance matrices from multiple candidate solutions. This framework was then applied to three test cases to illustrate its general utility in evaluating the potential performance of online sensor calibration procedures and establishing best practices in the field. This work demonstrates the utility of the GMIS' comprehensive error analysis tools for anticipating system performance, and presents a new paradigm for system design analysis in Kalman Filtering.

1.3 Structure of the Dissertation

This dissertation is structured as follows:

Following this introduction, Chapter 2 provides both the preliminary fundamentals of kinematic positioning and navigation used in this research. A literature review of the state-of-art in the topics relevant to each subsequent chapter.

Chapter 3 presents the generic framework for comprehensive error analysis in discrete Kalman filtering with constraints. The generic framework provides a standard algorithm for both applying system state constraints as an epoch-wise correction to the estimated state vector and calculating the

quantities that are necessary to conduct a comprehensive error analysis. Chapter 3 additionally provides a numerical example to illustrate the utility of the framework. The outcomes of Chapter 3 are fundamental to many of the subsequent chapters. A summary of the necessary algorithmic equations for conducting comprehensive error analysis for both unconstrained and constrained systems is provided in Section 3.4.

Chapter 4 describes York University's Position/Attitude (PosAtt) Engine, which includes information about the Engine's specific realization of the GMIS, details on the processing workflow and sub-tasks, and structure of the input/output files. This positioning engine forms the foundation that was built upon to accomplish most tasks across my PhD research.

Chapter 5 presents a systematic study of multiple attitude models in the GMIS, specifically the Roll-Pitch-Heading (RPH) model, the Direction Cosine Matrix (DCM) model, and the Quaternion model. It documents the development of their respective system models, state constraint equations for the DCM and Quaternion models, and IMU observation equations associated with each specific attitude model. This chapter presents the results of processing a road test dataset using each attitude model, alongside a comparative analysis of the performance between each attitude model.

Chapter 6 presents a novel modeling method for integrating data from a Multi-IMU (MIMU) array on the basis of the GMIS, allowing for the estimation of individual systematic errors and the analysis of the stochastic properties of each IMU sensor. A road test using three MEMS IMUs was conducted and processed to illustrate the unique technique developed through this research. As a result of this work, it is feasible to model each IMU in a MIMU array independently for the first time.

Chapter 7 proposes an analytical framework for conducting a rigorous performance pre-analysis for online sensor calibration for different candidate trajectories. This framework was applied to three different test cases, which are presented to illustrate the utility of the proposed framework in establishing best practices in the field.

2 Preliminaries and Related Literature

This chapter provides an overview of and background on the principles of inertial navigation. A review of the state-of-the-art practices related to each subsequent chapter's content can be found at the start of each chapter.

Topics covered in this chapter include:

- The coordinate frames used in positioning and navigation
- Methods for modeling orientation in three-dimensional space
- The laws governing the motion of a body in three-dimensional space
- Sensors that are commonly used for integrated navigation
- Kalman filtering and other methods for estimating time-varying system states
- Different strategies for sensor integration

Given the modeling work conducted throughout this research, this section has a particular focus on sensor measurement and stochastic models, and on data integration strategies.

2.1 Coordinate Frames in Positioning and Navigation

Generally, a supplied position or attitude is meaningless unless it is tied to a specifically defined coordinate frame. In positioning and navigation applications, and particularly when using inertial navigation technologies, there are many relevant coordinate frames that must be defined. These coordinate frames can be thought of as forming a hierarchy of sorts, ranging from the most “stationary” (the inertial frame) to the most “mobile” (the sensor body frames). An overview of this hierarchy is shown in Figure 2.1.1.

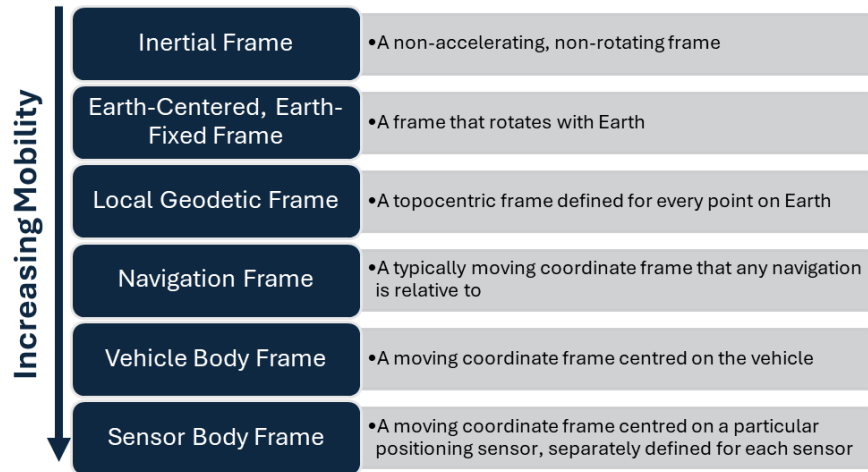


Figure 2.1.1: An overview of the coordinate frames used in this research, arranged in order of increasing mobility.

Broadly speaking, the coordinate frames that are commonly used in positioning and navigation can be divided into the following categories [Rogers, 2003]:

- The Inertial Reference Frame (*i*):
 - An Inertial Reference Frame refers to a Cartesian coordinate frame that is not accelerating through space (i.e. either stationary or moving at a constant linear velocity). There are many potential realizations that satisfy this criterion, but the specific Inertial Reference Frame typically used in positioning and navigation applications is the Earth-Centered Inertial (ECI) frame which has its origin defined to be coincident with Earth's center, but does not rotate with Earth [Herman, 1995; Chang et al, 2022; Fang et al, 2024]. The X- and Y-axes of the ECI frame lie in Earth's equatorial plane with its X-axis pointing towards the Vernal Point, its Z-axis towards Earth's North pole, and its Y-axis completing a right-handed system [Grewal et al, 2007]. Generally, strapdown inertial sensors may be considered to observe quantities that are impacted by the sensors' motion with respect to the Inertial Reference Frame.
- The Earth-Centered, Earth-Fixed (ECEF) Frame (*e*):
 - An ECEF Frame has its origin defined to be coincident with Earth's center but unlike the ECI frame, the ECEF frame rotates with Earth. The X- and Y-axes of the ECEF frame define Earth's equatorial plane, with the X-axis pointing towards the Greenwich Meridian. The Z-axis of the ECEF frame points towards Earth's North pole, and the Y-axis completes a right-handed system [Vanicek and Krakiwsky, 1986; Wu et al, 2003]. The relationship between the ECEF and ECI frame is shown in Figure 2.1.2.

The ECEF Frame is widely used to represent global coordinates for positioning applications such as navigation [Rahman et al, 2020] and land surveying [Teskey, 1979; Mercurio and Reyes, 2024], but also forms the basis for determining projected coordinates in mapping systems [Maling, 1992].

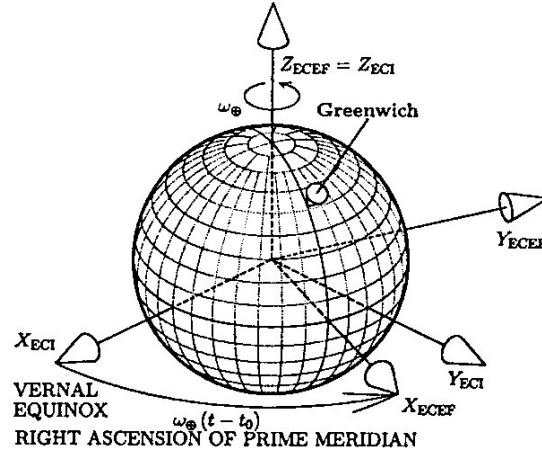


Figure 2.1.2: The relationship between the ECI and ECEF coordinate frames. Image courtesy of [Grewal et al, 2007].

- The Local Geodetic Frame (g):
 - The Local Geodetic Frame is a topocentric coordinate system. Its Z-axis is defined to be collinear with the local gravity vector. Two main realizations of the Local Geodetic Frame are used in practice: the North-East-Down (NED) frame and the East-North-Up (ENU) frame. In this research, the ENU frame is used. The X-axis of the ENU frame points in the local East direction, the Z-axis points in the opposite direction to the local gravity vector, and the Y-axis completes a right-handed system (i.e. the Y-axis points in the local North direction). One consideration that must be taken into account when using the ENU frame is that the vehicle's heading is measured clockwise about the Up-axis, when in the NED frame it would be measured counter clockwise about the Down-axis [Noureldin et al, 2012].
- The Navigation Frame (n):
 - The Navigation Frame is the coordinate frame in which positioning and navigation is realized for a PosAtt system. It is typically used to define the mechanization of an Inertial Navigation System (INS) [Schmidt et al, 1973; Heller, 1975]. There are different ways to practically define a n -frame. In this research, the n -frame is considered to translate with the Vehicle Body Frame (which is described below), and its orientation is periodically updated to be coincident with the local ENU frame.

- The Vehicle Body Frame (v):
 - The Vehicle Body Frame is a coordinate frame that has its origin rigidly fixed to a point in the vehicle's body, and rotates with the vehicle's body. There are two specific realizations of the v -frame that are commonly used in practice: the Forward-Right-Down (FRD) frame [Fan, 2024], and the Right-Forward-Up (RFU) frame [Liu et al, 2019]. In this research, the RFU frame is used due to its definition being analogous to the definition of the ENU frame.
- The Sensor Body Frame (b):
 - In positioning and navigation applications, the motion of the v -frame is typically the desired output. However, a PosAtt sensor generally refers to the movement or position of the sensor itself, and not of the v -frame. Each PosAtt sensor generally has a linear misalignment with the v -frame origin (i.e. the sensor origin requires translation to be coincident with the v -frame origin), which is referred to as its lever-arm vector. Additionally, each sensor also generally has angular misalignments with respect to the v -frame axes (i.e. the sensor axes require rotation to be collinear with the v -frame axes), which is referred to as the sensor's mounting or boresight angles. The models used in this dissertation take the 3D sensor lever-arm vectors and three boresight angles into account, but they are considered to be known and calibrated prior as part of the construction of the multi-sensor integrated system.

2.2 Attitude Models in Positioning, Orientation, and Navigation

Traditionally, the main use of attitude modeling in positioning was to facilitate the inertial dead reckoning calculation. However, in modern multi-sensor integrated navigation systems, it is often desirable to directly georeference the data from imaging sensors such as LiDAR units and cameras. For such applications, maintaining an accurate representation of the PosAtt system's attitude is essential in order to reference any imaging sensors relative to the local horizon for any given observation epoch. Considering the importance of attitude modeling in these application areas, and the focus on attitude modeling in this dissertation, this subsection provides an overview of the most prominent ways to represent orientation in PosAtt systems.

Let $\mathbf{p}^m$ be a vector in coordinate frame m and $\mathbf{p}^n$ be its equivalent representation in coordinate frame n . If frame m and frame n have coincident origins and the same defined scale, the transformation from m to n may be expressed using a rotation matrix $\mathbf{C}_m^n$ such that

$$\mathbf{p}_n = \mathbf{C}_m^n \mathbf{p}_m \quad (2.1)$$

wherein $\mathbf{C}_m^n$ is a 3 x 3 rotation matrix and satisfies all the properties of a rotation matrix, such as it being orthonormal. Given this relationship between two coordinate frames m and n , this rotation matrix may be represented in several different ways, with the most notable representations being the Direction Cosine Matrix (DCM), Roll-Pitch-Heading (RPH), and Quaternion. The mathematics of these representations are overviewed from Section 2.2.1 to Section 2.2.3, respectively.

Generally, the elements of a rotation matrix may be defined using the dot products between the basis vectors in frame m and frame n as follows:

$$\mathbf{C}_m^n = \begin{bmatrix} \hat{\mathbf{i}}_m \cdot \hat{\mathbf{i}}_n & \hat{\mathbf{i}}_m \cdot \hat{\mathbf{j}}_n & \hat{\mathbf{i}}_m \cdot \hat{\mathbf{k}}_n \\ \hat{\mathbf{j}}_m \cdot \hat{\mathbf{i}}_n & \hat{\mathbf{j}}_m \cdot \hat{\mathbf{j}}_n & \hat{\mathbf{j}}_m \cdot \hat{\mathbf{k}}_n \\ \hat{\mathbf{k}}_m \cdot \hat{\mathbf{i}}_n & \hat{\mathbf{k}}_m \cdot \hat{\mathbf{j}}_n & \hat{\mathbf{k}}_m \cdot \hat{\mathbf{k}}_n \end{bmatrix} \quad (2.2)$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are the X-, Y-, and Z-axial basis vectors, respectively, and their subscripts denote their respective coordinate frames [Bernstein, 2013].

2.2.1 Direction Cosine Matrix (DCM)

The Direction Cosine Matrix (DCM) consists of the 9 elements of the rotation matrix as in (2.2) to represent the transformation from frame m to frame n . Since the attitude parameters are defined by the elements of the rotation matrix, they may be determined by dividing $\mathbf{C}_m^n$ into its constituent 3 x 1 column vectors

$$\mathbf{C}_m^n = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = [\mathbf{C}_1 \quad \mathbf{C}_2 \quad \mathbf{C}_3] \quad (2.3)$$

Under this attitude model, the attitude parameters are defined to be

$$\boldsymbol{\theta}_{DCM} = [\mathbf{C}_1^T \quad \mathbf{C}_2^T \quad \mathbf{C}_3^T]^T = [C_{11} \quad C_{21} \quad \dots \quad C_{33}] \quad (2.4)$$

2.2.2 Roll-Pitch-Heading (RPH)

The elements of the rotation matrix are sufficiently determined by using three Euler angles, which represent a series of three rotations about the principal axes of frame m [Milligan, 1999]. Each of these rotations may be represented using its own rotation matrix in a selected order

$$\begin{aligned} \mathbf{R}_1(\theta_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 0 & -\sin \theta_1 & \cos \theta_1 \end{bmatrix}, \mathbf{R}_2(\theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & -\sin \theta_2 \\ 0 & 1 & 0 \\ \sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix}, \\ \mathbf{R}_3(\theta_3) &= \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.5)$$

where $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ denote rotations about the X-, Y-, and Z-axes of frame m , respectively, and θ_1 , θ_2 , and θ_3 denote the respective rotation angles.

The generalized 3-dimensional rotation matrix may be constructed using a sequence of 3 rotations from (2.5) [Milligan, 1999]. The order of these rotations has an impact on the identified rotation angles, but different series of rotations may be used to represent the same 3-dimensional rotation matrix. One of the most popular methods of defining the sequence of the 3 rotations is using the Tait-Bryan angles [Bernardes and Viollet, 2022]. For positioning applications, this is typically accomplished using the rotation sequence of Z-X-Y, which allows the 3 rotation parameters to be conventionally defined as the system heading/yaw (γ) (rotation about the Z-axis for an ENU system), system pitch (β) (rotation about the X-axis for an ENU system), and system roll (α) (rotation about the Y-axis for an ENU system). The generalized rotation matrix may be defined using these parameters with the following equation:

$$\mathbf{C}_m^n = \mathbf{R}_2(\alpha)\mathbf{R}_1(\beta)\mathbf{R}_3(\gamma) \quad (2.6)$$

from which the attitude angles are defined to be

$$\boldsymbol{\theta}_{RPH} = [\alpha \quad \beta \quad \gamma]^T \quad (2.7)$$

Substituting the 3 matrices in (2.5) into (2.6) yields the following DCM matrix for the general 3-dimensional rotation using the RPH angles:

$$\mathbf{C}_m^n = \begin{bmatrix} \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma & \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \beta \sin \gamma & \cos \beta \cos \gamma & \sin \beta \\ \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma & \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma & \cos \alpha \cos \beta \end{bmatrix} \quad (2.8)$$

The RPH angles may be calculated from (2.8) as follows

$$\boldsymbol{\theta}_{RPH} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{C_{13}}{C_{33}} \\ \sin^{-1} C_{23} \\ \tan^{-1} \frac{C_{21}}{C_{22}} \end{bmatrix} \quad (2.9)$$

Practically, the roll and heading are calculated using the “atan2” function to determine their four-quadrant values.

The RPH attitude representation is the most commonly-applied attitude model in positioning and navigation applications, but does typically come with a major limitation in the Gimbal Lock phenomenon due to singularities in (2.9). The Gimbal Lock phenomenon occurs when the orientation of coordinate frame m is approaching a pitch of $\pm 90^\circ$. When this happens, the definition of the system roll and heading in (2.9) becomes ambiguous, which often leads to numerical instability in positioning and navigation solutions [Hemingway and O'Reilly, 2018; De Paula et al, 2024].

2.2.3 Quaternion Representation

A general 3-dimensional rotation may also be represented using a unit quaternion [Kuipers, 1999; Kong, 2000; Mukundan, 2012; Voight, 2021]. The unit quaternion consists of 4 elements: a scalar component q_0 , and a 3-dimensional vector $\mathbf{q}$, which may both be represented by using the 3-dimensional rotation's rotation axis and angle as follows:

$$\mathbf{q} = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = \cos\frac{\theta}{2} + \mathbf{u}\sin\frac{\theta}{2} \quad (2.10)$$

wherein θ denotes the rotation angle and $\mathbf{u}$ denotes the 3-dimensional unit rotation axis vector. The quaternion is defined in components as $\{q_0, q_1, q_2, q_3\}$.

Several basic arithmetic operations on quaternions are given below for our further use in this work [Mukundan, 2012]. Firstly, the product of two quaternions p and q is given by the equation

$$pq = p_0q_0 - \mathbf{p} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} \quad (2.11)$$

The conjugate of a quaternion $\mathbf{p}$ is given by

$$\mathbf{p}^* = p_0 - \mathbf{p} \quad (2.12)$$

The norm of a quaternion p is given by

$$|\mathbf{p}| = \sqrt{\mathbf{p}^* \mathbf{p}} \quad (2.13)$$

The inverse of a quaternion p is given by

$$\mathbf{p}^{-1} = \frac{\mathbf{p}^*}{|\mathbf{p}|} \quad (2.14)$$

Let $\mathbf{v}$ denote a 3-dimensional vector in frame m . Its equivalent quaternion representation may be defined as $\mathbf{v} = 0 + \mathbf{v}$, where the vector portion of the quaternion is equal to $\mathbf{v}$ and the scalar portion of the quaternion is 0. The rotation transformation of a vector $\mathbf{v}$ using the rotation quaternion $\mathbf{q}$ is accomplished via the equation [Mukundan, 2012]:

$$L_q(\mathbf{v}) = \mathbf{q}\mathbf{v}\mathbf{q}^* = (q_0^2 + |\mathbf{q}|^2)\mathbf{v} + 2(\mathbf{q} \times \mathbf{v})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{v}) \quad (2.15)$$

The rotation matrix defining this transformation may be determined as follows:

$$\mathbf{C}_m^n = (q_0^2 - |\mathbf{q}|^2)\mathbf{I}_3 - 2q_0[\mathbf{q}] + 2\mathbf{q}\mathbf{q}^T \quad (2.16)$$

where [...] denotes the cross-product matrix [Mukundan, 2012], defined as

$$[\mathbf{q}] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (2.17)$$

The quaternion rotation representation overcomes several major limitations of the RPH attitude representation, namely [Voight, 2021]:

- Calculating the rotation matrix from a rotation quaternion is more computationally efficient than using the RPH, since the rotation matrix may be represented as simple multiplication of the rotation quaternion.
- The quaternion representation may be more readily interpolated between successive epochs than the RPH representation. This has led the quaternion representation to be particularly popular in computer graphics applications.
- The quaternion representation does not suffer from the Gimbal Lock phenomenon.

2.3 Kinematics and Rigid Body Motion

Generally, the motion of a particle or rigid body is considered to be governed by Newton's Law of Motion, expressed mathematically as [Ovenden, 1972]

$$\mathbf{F} = m\mathbf{a} \quad (2.18)$$

where $\mathbf{F}$ is the net force acting on the particle or rigid body's center of mass, m is the mass of the particle or rigid body, and $\mathbf{a}$ is the linear acceleration that the particle or rigid body is experiencing.

2.3.1 Kinematics of a Non-Rigid Body

In positioning and navigation, the goal is generally to define the trajectory of a vehicle through the navigation frame. It is convenient to think of the vehicle as a rigid body that does not experience any relative displacement between constituent points (i.e. non-deforming, see Figure 2.3.1 for an illustration of this principle) [Spencer, 2003; Chandrasekharaiah and Debnath, 2014]. However, this is rarely the case. For example, an airplane's wing is naturally subjected to flexing during flight [Schmidt and Raney, 2012], and the suspension on ground vehicles also violates this assumption [Cao et al, 2008]. As a result, it is important to be able to represent dynamics for a non-rigid body, and this section describes the general equations of motion for a non-rigid body.

In practice, many PosAtt systems are produced in a housing or mounted on a frame that is considered to be a rigid body [Ghanizadegan and Hashim, 2024; Chepuri et al, 2013; Wagner, 2004]. Such systems' motion is generally modeled as rigid body motion, where the geometrical relationship between constituent sensors and the PosAtt system (i.e. lever arm vectors and mounting angles) remains constant. This research uses this approach to characterizing the motion of a PosAtt system.

to better contextualize how those constraints affect the characterization of the vehicle's motion.

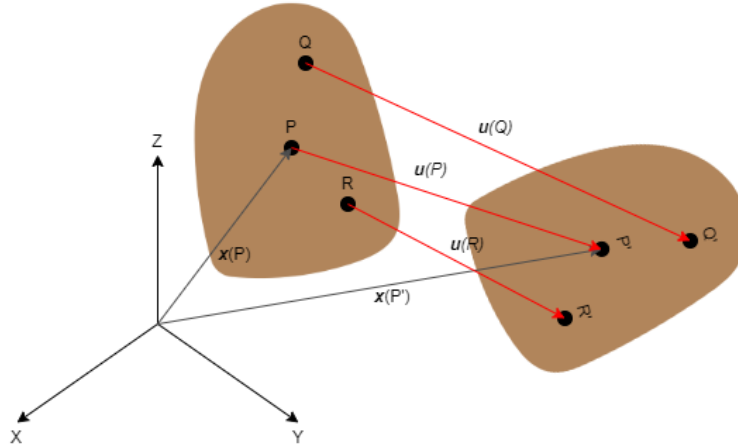


Figure 2.3.1: Diagram illustrating the lack of deformation in rigid body motion. Note that although the displacement vectors $\overrightarrow{PP'}$, $\overrightarrow{QQ'}$, and $\overrightarrow{RR'}$ are different from one another, the relative positions between P , Q , and R are the same as the relative positions between P' , Q' , and R' .

In General, the position of a point P in a non-rigid body may be represented at time t in coordinate frame n as

$$\mathbf{x}_n(P, t) = \begin{bmatrix} x_n(P, t) \\ y_n(P, t) \\ z_n(P, t) \end{bmatrix} \quad (2.19)$$

The transition of the body from one position to another may be characterized using a displacement vector field $\mathbf{u}_n$ as [Abeyaratne, 1998]

$$\mathbf{x}_n(P, t) = \mathbf{x}_n(P, t_0) + \mathbf{u}_n(P, t) \quad (2.20)$$

This displacement vector field defines a one-to-one mapping between time t_0 and time t , and it encodes both the motion of the body as well as any deformation or transformation it experiences [Abeyaratne, 1998].

The linear velocity of the non-rigid body in frame n may be represented by the time-derivative of (2.20) as follows

$$\mathbf{v}_n(P, t) = \frac{d\mathbf{u}_n}{dt}(P, t) \quad (2.21)$$

The linear acceleration may similarly be expressed using the second time-derivative of (2.20) as

$$\mathbf{a}(P, t) = \frac{d^2\mathbf{u}_n}{dt^2}(P, t) \quad (2.22)$$

2.3.2 Kinematics of a Rigid Body

Any special properties regarding the deformation of the moving body are realized as constraints on $\mathbf{u}_n$ in (2.20). A rigid body maintains the distance between all constituent particles, and $\mathbf{u}_n$ therefore takes the form of

$$\mathbf{u}_n(t) = \mathbf{C}(t)\mathbf{x}_n + \mathbf{b}(t) \quad (2.23)$$

where $\mathbf{C}(t)$ is an orthogonal matrix (i.e. a rotation matrix) and $\mathbf{b}(t)$ is a vector describing the system's translation. Note that for a rigid body, this transformation is defined consistently across the entire rigid body and its constituent parts are therefore defined only as a function of time [Abeyaratne, 1998]. From this point forward, the working platform is assumed to be a rigid body and conform to rigid body motion.

2.3.3 Motion in a Rotating Coordinate Frame

Many of the observables in a multi-sensor integrated PosAtt system relate the moving platform's motion to the i -frame; however, the n -frame is not inertial, and it translates/rotates with respect to the

i -frame. The effects of the n -frame's motion on the inertial observations must therefore be accounted for when attempting to perform positioning in the n -frame, and the n -frame's rotation with respect to the i -frame has a particularly significant impact.

It is instructive to first characterize the effect that rotation has on a stationary point to better understand the impact of angular velocity. When coordinate frame n rotates, the resulting transformation of $\mathbf{x}_n$ is

$$\mathbf{x}_{n'}(t) = \mathbf{C}_n^{n'}(t)\mathbf{x}_0 \quad (2.24)$$

where the use of $\mathbf{x}_0$ merely reinforces that the body is not experiencing motion outside of the rotation of frame n . It is important to note that since $\mathbf{C}_n^{n'}(t)$ represents a rotation matrix it is orthogonal, or mathematically for any rotation matrix $\mathbf{C}$:

$$\mathbf{C}\mathbf{C}^{-1} = \mathbf{C}\mathbf{C}^T = \mathbf{I} \quad (2.25)$$

The resultant time derivative of the body's position may then be calculated as

$$\dot{\mathbf{x}}_{n'}(t) = \dot{\mathbf{C}}_n^{n'}(t)\mathbf{x}_0 = \dot{\mathbf{C}}_n^{n'}(t)(\mathbf{C}_n^{n'})^T(t)\mathbf{x}_n(t) = \mathbf{\Omega}\mathbf{x}_n(t) \quad (2.26)$$

with $\mathbf{\Omega} = \dot{\mathbf{C}}_n(t)(\mathbf{C}_n^{n'})^T(t)$ [Crampin, 1986].

Differentiating (2.24) with respect to time gives the relation

$$\dot{\mathbf{C}}_n^{n'}(\mathbf{C}_n^{n'})^T + \mathbf{C}_n^{n'}(\dot{\mathbf{C}}_n^{n'})^T = \mathbf{\Omega} + \mathbf{\Omega}^T = \mathbf{0} \quad (2.27)$$

wherein $\mathbf{\Omega}$ is a skew-symmetric matrix and may therefore be defined by using three elements, which comprise the angular velocity vector $\boldsymbol{\omega}_n$ [Crampin, 1986]. The relation in (2.26) may therefore be rewritten as

$$\dot{\mathbf{x}}_n(t) = [\boldsymbol{\omega}_n]\mathbf{x}_0 = \boldsymbol{\omega}_n \times \mathbf{x}_0 \quad (2.28)$$

Of course, (2.24) to (2.28) describe the effects of a rotating coordinate frame on a static body. For a body that itself is experiencing motion, (2.24) may be modified to

$$\mathbf{x}_{n'}(t) = \mathbf{C}_n^{n'}(t)\mathbf{x}_n(t) \quad (2.29)$$

Taking the time derivative of (2.29) yields

$$\dot{\mathbf{x}}_{n'}(t) = \dot{\mathbf{C}}_n^{n'}(t)\mathbf{x}_n(t) + \boldsymbol{\omega}_n \times \mathbf{x}_n(t) \quad (2.30)$$

which fully characterizes the velocity of a moving body in a rotating coordinate frame n [Crampin, 1986].

2.3.4 The Poisson Equation

The Poisson Equation relating the angular velocity to the rotation matrix and its time derivative is particularly important, and may be easily derived via the definition of $\mathbf{\Omega}$ in (2.26) for an arbitrary rotation matrix $\mathbf{C}$ [Salychev, 1998; Salytcheva, 2004; Alexiev, 2019]

$$\dot{\mathbf{C}} = \mathbf{C}[\boldsymbol{\omega}] \quad (2.31)$$

2.3.5 Relating Inertial Measurements to the Navigation Frame

This section lays out the fundamental navigation equations used throughout this research when modeling or characterizing IMU observables [Qian, 2017]. Let $\mathbf{v}_{ev}^i$ be the linear velocity of a moving platform with respect to Earth in the inertial frame. It may be related to Earth's rotation rate through applying (2.30):

$$\mathbf{v}_{ev}^i = \mathbf{v}_{iv}^i - \boldsymbol{\omega}_{ie}^i \times \mathbf{x}_v^i \quad (2.32)$$

wherein $\boldsymbol{\omega}_{ie}^i$ denotes the angular velocity of the Earth in the inertial frame with respect to the inertial frame and $\mathbf{x}_v^i$ denotes the position of the platform in the inertial frame.

The first-order time derivative of (2.32) in the inertial frame is

$$\left. \frac{d\mathbf{v}_{ev}^i}{dt} \right|_i = \mathbf{a}_{iv}^i - \boldsymbol{\omega}_{ie}^i \times \left. \frac{d\mathbf{x}_v^i}{dt} \right|_i = \mathbf{a}_{iv}^i - \boldsymbol{\omega}_{ie}^i \times (\mathbf{v}_{ev}^i + \boldsymbol{\omega}_{ie}^i \times \mathbf{x}_v^i) \quad (2.33)$$

The left side of (2.33) may be alternatively expressed as

$$\left. \frac{d\mathbf{v}_{ev}^i}{dt} \right|_i = \left. \frac{d\mathbf{v}_{ev}^n}{dt} \right|_i + \boldsymbol{\omega}_{in}^i \times \mathbf{v}_{ev}^i \quad (2.34)$$

which may be combined with (2.33) to produce

$$\mathbf{a}_{iv}^n = \mathbf{a}_{ev}^n + (\boldsymbol{\omega}_{in}^n + \boldsymbol{\omega}_{ie}^n) \times \mathbf{v}_{ev}^n + \mathbf{g}^n \quad (2.35)$$

wherein all vectors have been transformed to the n -frame.

2.4 Sensors Used in Integrated Navigation

This section describes sensors that are commonly used in integrated navigation, with a particular focus on Global Navigation Satellite Systems (GNSS) receivers and Inertial Measurement Units (IMUs).

2.4.1 Global Navigation Satellite Systems (GNSS) Receivers

Global Navigation Satellite Systems (GNSS) use passive ranging from radio signals that are transmitted from satellites to provide global coverage for autonomous positioning, navigation and timing. Prominent examples include the United States' Global Positioning System (GPS), China's BeiDou constellation, the European Union's Galileo constellation, and Russia's GLONASS constellation. These systems may be augmented by Regional Navigation Satellite Systems (RNSS) such as India's IRNSS and Japan's QZSS. While this research specifically uses GPS for positioning, the principles applied in this research are readily applied to other GNSS constellations. As such, this section primarily focuses on the shared principles among the GNSS.

2.4.1.1 The Principle of Multilateration for GNSS

The principle of multilateration may generally be used to determine the position of a GNSS receiver. In practice, the measured pseudo-ranges from the receiver to at least 4 satellites are required in order to resolve both the receiver's 3D coordinates and its clock error (Figure 2.4.1).

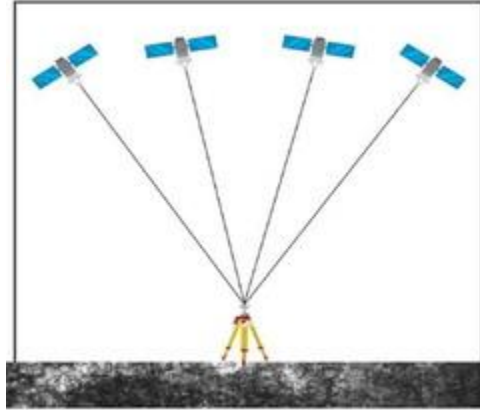


Figure 2.4.1: The signal received from each GNSS satellite may be used to infer the range between it and the receiver and used for multilateration. Diagram courtesy of [Lotfy et al, 2020].

Considering only the geometric distance r^i between a GNSS receiver and a specific GNSS satellite i , the receiver's position is related to the satellite's position via the equation

$$r^i = \sqrt{(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2} \quad (2.36)$$

where (x, y, z) denotes the coordinates of the GNSS receiver to be determined and (X_i, Y_i, Z_i) denotes the known coordinates of satellite i in the ECEF. All GNSS observation equations make use of this simple equation.

GNSS satellites emit modulated signals over multiple carrier frequencies. These signals each contain a Pseudo-Random Noise (PRN) ranging code, navigation data specifying the satellite ephemeris, and the sinusoidal carrier wave itself [Langley et al, 2017]. Most GNSS use Code Division Multiple Access (CDMA) to differentiate between each satellite in the constellation, where each satellite generates a different PRN code. Legacy GLONASS signals use Frequency Division Multiple Access (FDMA), where different satellites transmit signals over different frequencies and all satellites generate the same PRN code [Chen et al, 2023]. More recently, the GLONASS-K satellites use CDMA for better interoperability with other GNSS.

The PRN code for each satellite is also generated by a GNSS receiver and used to determine the time difference between the emission of the PRN code and its receipt. This time difference is used to determine the pseudo-range between the satellite and the receiver, which is one of the primary GNSS observables.

The signal's carrier wave phase may be tracked by a GNSS receiver for each satellite and once tracked, has two components: the instantaneously observed fractional wavelength of the carrier wave, and an accumulated number of integer cycles determined via signal tracking. These comprise the carrier phase observable, which is another primary GNSS observable. Carrier phase observables do not describe the total number of wavelengths between the GNSS receiver and the satellite, and estimating this value is referred to as the ambiguity problem. When using double-differenced GNSS observables as described in Section 2.4.1.3, the ambiguous number of wavelengths has an integer value. For single-point positioning, the receiver and satellite phase offsets cause the ambiguous number of wavelengths to not have an integer value [Blewitt, 1997]. An illustration of the GNSS ambiguity problem is shown in Figure 2.4.2.

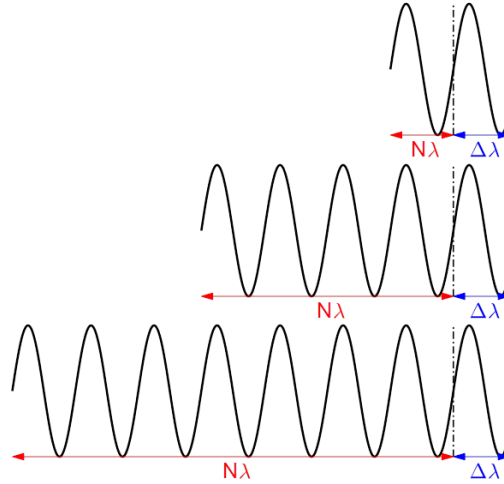


Figure 2.4.2: Diagram illustrating the ambiguity problem for carrier phase observables. Note that in each of the 3 situations, the fractional wavelength is the same, even though the total length of each wave is different.

The relative velocity between a GNSS receiver and a GNSS satellite causes a Doppler shift in the signal that changes the received carrier wave frequency. This Doppler shift comprises the Doppler observable and may be used to provide insight into the velocity of a GNSS receiver [Petovello, 2015].

There are two positioning modes in GNSS: Single-Point Positioning (SPP), in which the GNSS observables are directly used to estimate the position of a single receiver, and Relative Positioning, in which the GNSS observables are processed over multiple receivers. There are many GNSS processing architectures that fall under these general categories. For example, Precise Point Positioning (PPP) is a SPP technique that uses precise ephemeris data, atmospheric models, and processing strategies to improve the positioning solution. Real-Time Kinematic (RTK) positioning is a relative positioning technique that uses a network of GNSS base stations to provide real-time error corrections to receivers that connect to the network. The general observation equations for SPP are discussed in Section 2.4.1.2, and the general observation equations for Relative Positioning are discussed in Section 2.4.1.3

2.4.1.2 GNSS Single-Point Positioning

A pseudo-range observable is coarsely analogous to the geometric range in (2.36), with an added scale defect due to the combined effects of the satellite clock error, dt , and the receiver clock error, dT . Additional systematic errors must also be modeled and accounted for such as tropospheric delay, d_{trop} , ionospheric delay, d_{ion} , and ephemeris errors, d_o . As a measurement, the pseudo-range measurement to satellite i may be expressed as follows

$$\rho^i = r^i + c(dt - dT) + d_o + d_{trop} + d_{ion} + d_M + \varepsilon_\rho \quad (2.37)$$

wherein d_M represents the multipath error and ε_ρ represents the random error in the measurement [Gong et al, 2009; Li et al, 2021].

When the carrier phase ambiguity is modeled, the carrier phase observables may also be related to the geometric range in a similar form as in (2.37)

$$\phi^i = \frac{1}{\lambda} [r^i + c(dt - dT) + d_o + d_{trop} + d_{ion} + \varepsilon_M] + N^i + \varepsilon_\phi \quad (2.38)$$

where λ is the wavelength of the carrier wave, N^i is the carrier phase ambiguity parameter, and ε_ϕ is the random error in the carrier phase measurements [Gong et al, 2009; Ding et al, 2021].

The standard SPP solution can generally achieve meter-level accuracy [Beran et al, 2007]. Using Precise Point Positioning (PPP) techniques, this accuracy can reach the decimeter and even the centimeter level for a single receiver, even when using low-cost sensors [Aggrey et al, 2020; An et al, 2020; Yi et al, 2021; Carlin et al, 2021].

2.4.1.3 GNSS Relative Positioning

Relative positioning between two GNSS receivers tends to drastically improve the accuracy of the positioning solution [Janssen, 2001; Chen et al, 2024]. This research adapts the GNSS relative positioning mode to target the accuracy level of a few centimeters using L1 C/A, L1 and L2 carrier phases from the GPS constellation. In particular, this research uses double-differenced GPS observations between a reference receiver and a rover receiver (Figure 2.4.3). This section presents the general principles for using single- and double-differenced GNSS observables, as well as the associated observation equations.

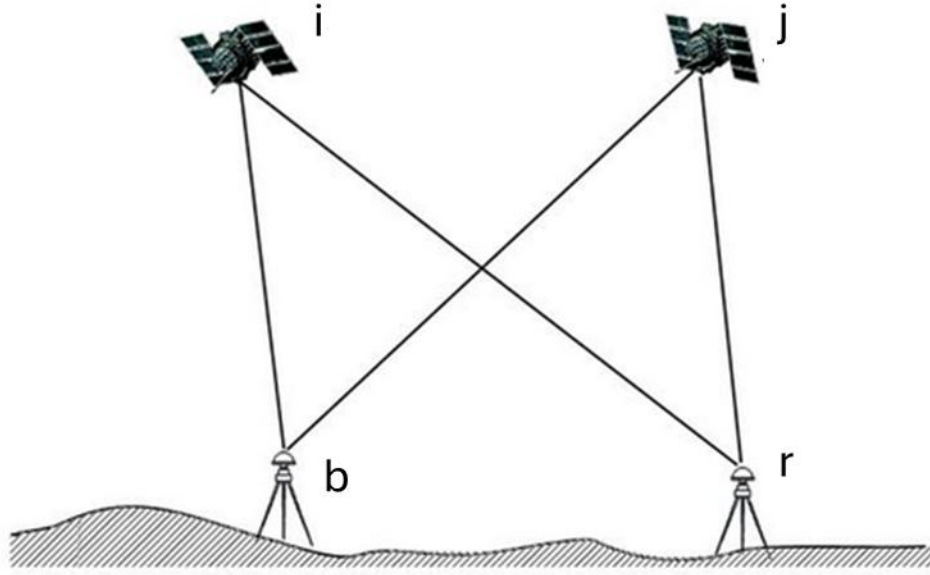


Figure 2.4.3: Diagram illustrating the generation of single- and double-differenced GNSS observations, adapted from [Souza et al, 2009].

When two GNSS receivers are being used, one may be assigned as a base receiver b and one may be assigned as a rover receiver r . When both receivers simultaneously record observations from a group of GNSS satellites, the single-differenced GNSS observables between two receivers may be constructed with respect to a satellite i as follows:

$$\Delta\rho_{rb}^i = \rho_b^i - \rho_r^i = \Delta r_{rb}^i + c\Delta T_{rb} + \Delta d_{trop}^i + \Delta d_{ion}^i + \Delta\varepsilon_M + \Delta\varepsilon_{\Delta\rho} \quad (2.39)$$

$$\Delta\phi_{rb}^i = \phi_b^i - \phi_r^i = \frac{1}{\lambda} [\Delta r_{rb}^i + c\Delta T_{rb} + \Delta d_{trop}^i + \Delta d_{ion}^i + \Delta\varepsilon_M] + \Delta N_{rb}^i + \Delta\varepsilon_{\Delta\phi} \quad (2.40)$$

Note that the single-differencing process between receivers eliminates the satellite-related errors, since these quantities are identical at both receivers for the same satellite. Additionally, the single-differencing process reduces the magnitude of the tropospheric delay and ionospheric delay terms, as long as both the rover and base receivers are close to one another.

Double-differenced GNSS observations can be generated by taking the difference between two single-differenced observations referring to two GNSS satellites at the same time. When neglecting multipath errors, the double-differenced GNSS observables between GNSS satellites i and j may be given as

$$\nabla\Delta\rho_{rb}^{ij} = \Delta\rho_{rb}^j - \Delta\rho_{rb}^i = \nabla\Delta r_{rb}^{ij} + \varepsilon_{\nabla\Delta\rho} \quad (2.41)$$

$$\nabla\Delta\phi_{rb}^{ij} = \Delta\phi_{rb}^j - \Delta\phi_{rb}^i = \frac{1}{\lambda}\nabla\Delta r_{rb}^{ij} + \nabla\Delta N_{rb}^{ij} + \varepsilon_{\nabla\Delta\phi} \quad (2.42)$$

Note that the double-differencing process in (2.41) and (2.42) eliminates the receiver clock errors. The tropospheric and ionospheric delays are greatly reduced via the double-differencing process and are even sufficiently small to be ignorable in (2.41) and (2.42).

Resolving the integer ambiguity problem for the double-differenced GNSS carrier phase observations is an essential part of GNSS processing for Relative Positioning. The integer ambiguity is typically resolved by first estimating it as a floating-point unknown, and then fixing it to an integer value once the floating-point estimate's accuracy allows for a sufficiently high confidence in identifying its integer value [Teunissen et al, 1997; Teunissen et al, 1999; Collins et al, 2010; Xu et al, 2012]. There are other strategies to handling integer ambiguities, such as constraining phase biases rather than fixing them [Bertiger et al, 2010], but these are not considered in this research.

One negative impact on the single- and double-differencing process is that the random errors in the individual GNSS observations propagates into their respective single- and double-differenced observations. This results in doubling and quadrupling the associated measurement variances for the GNSS single- and double-differenced observations, albeit with a significant reduction of or even elimination of systematic errors.

Table 2.4.1 summarizes the status of the major error sources associated with the single- and double-differenced observables. In this research, double-differenced GNSS observations have been implemented in GNSS/IMU data fusion.

Table 2.4.1: The effects that the single- and double-differencing process has on the different sources of error that affect GPS observations [Van Sickle, 2023]. Note that multipath error is environment-dependent, but can potentially be exacerbated by the single- and double-differencing process.

Error source	Effects of Single-Differencing Observations with Two Receivers	Effects of Double-Differencing Observations
Orbit Error	Eliminated	Eliminated
Satellite Clock Error	Eliminated	Eliminated
Receiver Clock Error	Requires Estimation	Eliminated

Ionosphere Error	Reduced	Greatly Reduced
Troposphere Error	Reduced	Greatly Reduced
Measurement Noise	Variance doubled	Variance quadrupled

2.4.2 Inertial Measurement Units (IMUs)

An Inertial Measurement Unit (IMU) is a collection of sensors that typically provide information about the sensor body frame's motion relative to the inertial frame. The IMUs modeled as a part of this research are 6 DOF IMUs, consisting of:

- A set of 3 mutually orthogonal accelerometers that measures the specific force that the *b*-frame experiences relative to the *i*-frame; and
- A set of 3 mutually orthogonal gyroscopes that measures the angular rate that the *b*-frame experiences relative to the *i*-frame.

[Ahmad et al, 2013].

IMUs may contain additional sensors, such as:

- A set of 3 mutually orthogonal magnetometers in a 9 DOF IMU that can provide a non-wandering estimate of the vehicle's heading [Ahmad et al, 2013; Höflinger et al, 2013]. In a MEMS IMU, the purpose of magnetometer data is typically to provide a quick initialization of the system heading to an accuracy level of a few degrees, since even tactical grade MEMS IMUs can not successfully realize an independent gyro-compassing process.
- Temperature sensors that allow for the calibration of IMU errors as the ambient/operating temperature fluctuates [Martin et al, 2013]; and
- Barometric pressure sensors that can provide a non-wandering estimate of the vehicle's altitude [Tanigawa et al, 2008; Choo et al, 2023].

For the purposes of this research, the accelerometers and gyroscopes are considered to form the core functionality of an IMU sensor.

IMUs are generally graded by their performance, and the IMU grades are summarized in Table 2.4.2. Micro-Electro-Mechanical Systems (MEMS) IMUs are very dominant in lower-cost IMU sensors,

while Fibre-Optic Gyroscopes (FOG) and Ring-Laser Gyroscopes (RLG) are more prominent in navigation-strategic grade IMUs. This research focuses on the use of tactical-grade MEMS IMUs.

Table 2.4.2: Classification of IMU performance grades, adapted from [Advanced Navigation, 2025; VectorNav, 2025] without the inclusion of application-domain examples.

Performance Grade	Bias Instability	Cost	Dead-Reckoning Capability	Gyroscope Type
Consumer	>20°/h	<\$10	None	MEMS
Industrial	5-20°/h	\$100-\$1,000	3-5 minutes	MEMS
Tactical	0.1-5°/h	\$5000-\$50,000	10 minutes	MEMS/FOG/RLG
Navigation	0.01-0.1°/h	>\$100,000	Several hours	FOG/RLG
Strategic	0.0001-0.01°/h	>\$100,000	Several hours	FOG/RLG

Since the advent of MEMS technology, the cost barrier to purchasing an IMU has decreased significantly [Geiger et al, 2008; Prikhodko et al, 2018], and the attainable accuracy of MEMS IMUs has increased significantly [Johnson et al, 2021]. Generally, MEMS IMUs do tend to suffer from larger and more unstable systematic errors than other IMUs [Du et al, 2016; Wang et al, 2021; Li et al, 2021]. In particular, MEMS IMUs' systematic errors have generally proven to be more temperature-dependent than for, as an example, ring-laser gyroscopes [Capriglione et al, 2021; Li et al, 2022; Miyazaki et al, 2024]. Successful gyro-compassing still remains an unsolved challenge when using tactical grade MEMS IMUs. This research uses tactical grade MEMS IMUs for positioning and navigation.

2.4.2.1 Micro-Electro-Mechanical Systems (MEMS) Inertial Sensors

The development of Micro-Electro-Mechanical Systems (MEMS) IMUs forms the basis of many low-cost positioning and navigation systems. This technology involves the combination of electronic and mechanical parts in a microscopic device, and the development of MEMS technology has largely been driven by the economics of large-scale electronics manufacturing [Kempe, 2011].

There are two main factors of MEMS technology that have been significant drivers of innovation in the field of inertial positioning and navigation [Juillard, 2020]:

- The miniaturization of IMUs using MEMS technology:
 - Smaller, lighter IMU sensors provide researchers and end users with access to IMU data on smaller sensing platforms, such as Micro Aerial Vehicles [Loianno et al, 2017] or articles of clothing [Höflinger et al, 2013]. Additionally, the smaller sensors may be used to provide insight into differential motions on the smaller sensing platforms, for example, characterizing the motion of individual fingers on the human hand [Yang et al, 2020].
- The inexpensive manufacturing of IMUs using MEMS technology:
 - Inexpensive but accurate IMUs allows for the larger-scale production of low-cost positioning and navigation systems [Geiger et al, 2008; Prikhodko et al, 2018], with MEMS IMUs even approaching tactical-grade or better performance [Johnson et al, 2021].

In general, any sensor consists of two main components:

- A transducer, which converts one form of energy (e.g. kinetic energy) into another form of energy (typically an electrical signal).
- A conditioning circuit, which transforms the transducer's output into an electronic voltage. This typically involves amplification of the transducer output.

2.4.2.2 Operating Principles of MEMS Accelerometers

Each accelerometer in an IMU provides a measure of the specific force experienced along the corresponding sensing axis. For a MEMS accelerometer, this is often accomplished by suspending a proof mass with springs such that the proof mass has 1 degree-of-freedom in its motion along the desired sensing axis. As the IMU experiences acceleration, the proof mass deflects along the sensing axis and this deflection is then converted to an electrical signal that comprises the accelerometer output [Kempe, 2011]. There are two primary types of spring-mass accelerometers, both of which are illustrated in Figure 2.4.4: linear spring-mass accelerometers, and torsional spring-mass accelerometers.

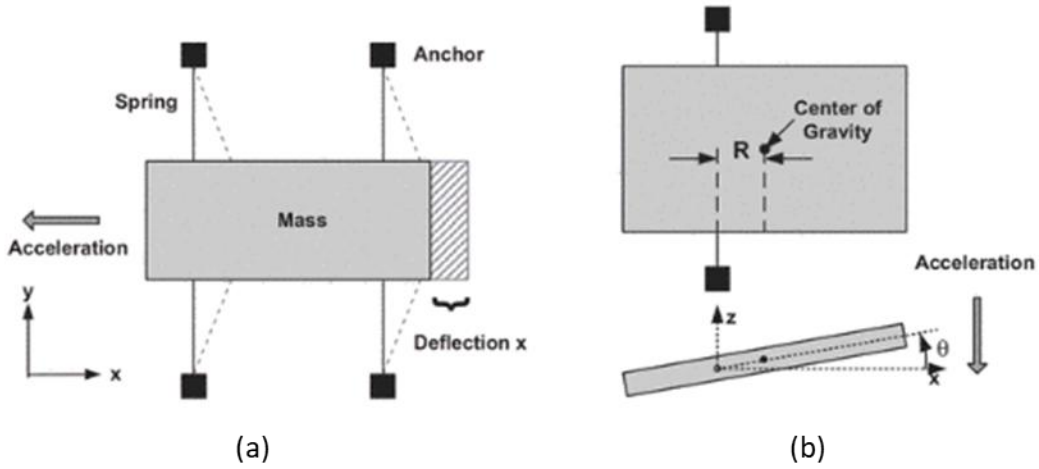


Figure 2.4.4: Illustration of the operating principles of spring-mass accelerometers. (a) shows a linear spring-mass accelerometer, and (b) shows a torsional spring-mass accelerometer. Images courtesy of [Kempe, 2011]. Note that in (b), the springs are not arranged symmetrically about the center of gravity of the proof mass. If this were not the case, only angular motion would cause the proof mass to deflect.

2.4.2.3 Operating Principles of MEMS Gyroscopes

The IMU gyroscopes each provide a measure of the angular rate experienced by the IMU along the corresponding sensing axis. This is typically accomplished by sensing the effects of the Coriolis force generated by a system's angular velocity with respect to the inertial frame, and in a MEMS gyroscope this is usually done by vibrating a proof mass along a drive axis and sensing its off-drive deflections [Kempe, 2011]. There are two primary types of rigid-body MEMS gyroscopes, both of which are illustrated in Figure 2.4.5: linear and rotatory [Kempe, 2011].

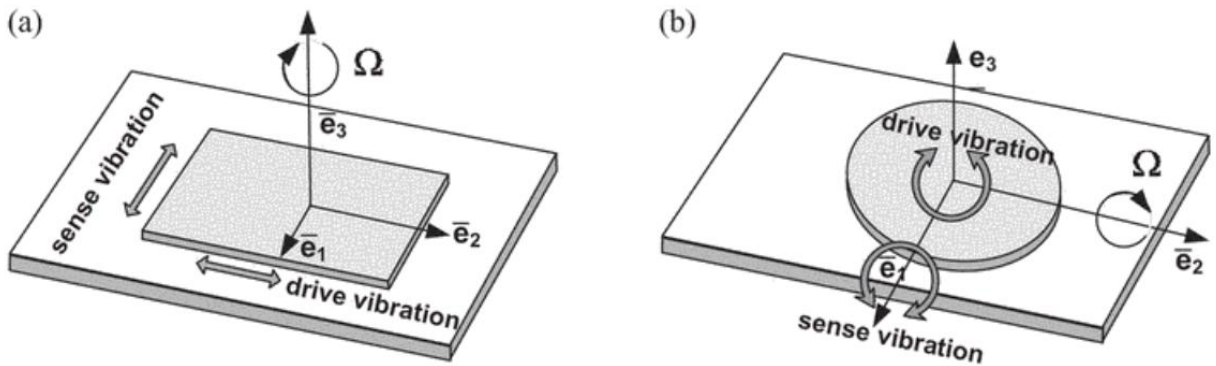


Figure 2.4.5: Illustration of the operating principles of rigid-body MEMS gyroscopes. (a) shows a linear rigid-body gyroscope, where any rotation about axis e_3 produces a Coriolis force along the sensing axis e_1 for a proof mass vibrating along the drive axis e_2 . (b) shows a rotatory rigid-body gyroscope, where rotation about axis e_2 produces oscillations about axis e_1 for a proof mass oscillating about axis e_3 . Images courtesy of [Kempe, 2011].

The mechanical systems underlying a MEMS gyroscope are much more complex than those underlying a MEMS accelerometer, since a 1 DOF MEMS gyroscope requires a proof mass that has 2 degrees of freedom in its motion [Kempe, 2011]. This can make MEMS gyroscopes particularly susceptible to errors in the manufacturing process [Kempe, 2011].

2.4.2.4 IMU Sensor Measurement Models

The individual accelerometers and gyroscopes are both considered to conform to the same basic generalized sensor model that accounts for sensor-specific scale errors, biases, and random noise. Generally, the measurements from a single IMU accelerometer or gyroscope sensor may be modeled using the equation

$$y = (1 + s)x + b_0 + b_1(t) + \varepsilon \quad (2.43)$$

where the measured specific force/angular rate component is denoted as y , its “true” value is denoted as x , b_0 is a random constant bias, b_1 is the time-varying bias drift, and ε is white noise [Lin et al, 2004; Balaban et al, 2009; Kirkko-Jaakkola, 2012; Grammenos et al, 2018; Yang et al, 2020].

There are several common ways of characterizing the time-varying bias drift in (2.43): it may be represented using a constant value, representing a non-drifting bias estimate in the sensor under consideration, or it may be modeled using a first order Gauss-Markov process as

$$\dot{b}_1(t) = -\frac{1}{\tau}b_1(t) + \varepsilon_{b_1} \quad (2.44)$$

where $\tau > 0$ is the time constant and ε_{b_1} is white noise [Kastella et al, 2000]. In practice, (2.44) is realized through a shaping filter that augments the standard system model in Kalman filtering.

Of course, the accelerometer triad and gyroscope triad may have cross-coupling errors. These may be accounted for by modifying (2.43) to describe all three sensors in the triad together, as

$$\mathbf{y} = (1 + \mathbf{S})\mathbf{x} + \mathbf{b}_0 + \mathbf{b}_1(t) + \varepsilon \quad (2.45)$$

wherein all lowercase bolded quantities describe the corresponding quantity in (2.43) using 3x1 vectors to encode the information from all three sensors in the triad. Here, $\mathbf{S}$ is a 3x3 matrix whose non-diagonal elements indicate the cross-coupling errors of the sensor triad. Generally, $\mathbf{S}$ should be a symmetric matrix, and is often treated as a diagonal matrix when no cross-coupling errors are being estimated [Becker et al, 2010; Kuzmanovic and Pajic, 2021].

2.4.2.4.1 Accelerometer Measurement Model

The measured accelerometer specific force vector from an accelerometer triad may be represented using (2.35) and (2.45), yielding the model equation:

$$\mathbf{f}_{ib}^b = \mathbf{C}_n^b \mathbf{a} + \mathbf{C}_n^b \mathbf{g} - \mathbf{b}_a - \mathbf{S}_a \mathbf{f}_{ib}^b + \mathbf{C}_n^b \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} + \mathbf{C}_n^b \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r} + \Delta_{f_{ib}}^b \quad (2.46)$$

where:

$\mathbf{f}_{ib}^b$ is the specific force that the IMU experiences with respect to the inertial frame i , expressed in the body frame b ;

$\mathbf{a}$ is the linear acceleration of the vehicle frame with respect to the navigation frame;

$\mathbf{v}$ is the linear velocity of the vehicle frame with respect to the navigation frame;

$\boldsymbol{\omega}_{ie}^n$ is the angular velocity of Earth with respect to the inertial frame, expressed in the navigation frame n ;

$\boldsymbol{\omega}_{en}^n$ is the angular velocity of the navigation frame with respect to Earth, expressed in the navigation frame n ;

$\boldsymbol{\omega}_{ib}^b$ is the angular velocity of the body frame with respect to the inertial frame, expressed in the body frame b ;

$\mathbf{C}_n^b$ is the rotation matrix from navigation frame n to body frame b ;

$\mathbf{r}$ is the lever arm vector from the IMU sensor to the origin of the PosAtt system;

$\mathbf{b}_a$ is the accelerometer bias vector, which includes both the constant bias and bias drift components;

$\mathbf{S}_a$ is the scale factor error matrix, including any cross-coupling errors; and

$\Delta_{f_{ib}}^b$ is the random white noise in the accelerometer specific force vector.

2.4.2.4.2 Gyroscope Measurement Model

The measured gyroscope angular rate vector from a gyroscope triad may be represented by using (2.45) and removing the effects of Earth's rotation and the navigation frame's rotation with respect to Earth, as

$$\boldsymbol{\omega}_{ib}^b = \boldsymbol{\omega}_{nb}^b - \mathbf{b}_g - \mathbf{S}_g \boldsymbol{\omega}_{ib}^b + \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) + \Delta_{\boldsymbol{\omega}_{ib}^b} \quad (2.47)$$

where:

$\boldsymbol{\omega}_{ib}^b$ is the angular rate vector for the IMU with respect to the inertial frame i , expressed in the body frame b ;

$\boldsymbol{\omega}_{nb}^b$ is the angular velocity the IMU experiences in the navigation frame n , expressed in the body frame b ;

$\boldsymbol{\omega}_{ie}^n$ is the angular velocity of Earth with respect to the inertial frame i , expressed in the navigation frame n ;

$\boldsymbol{\omega}_{en}^n$ is the angular velocity of the navigation frame with respect to Earth, expressed in the navigation frame n ;

$\mathbf{C}_n^b$ is the rotation matrix from navigation frame n to body frame b ;

$\mathbf{b}_g$ is the gyroscope bias vector, including both the constant and time-varying bias components;

$\mathbf{S}_g$ is the scale factor error matrix, including any cross-coupling errors; and

$\Delta_{\boldsymbol{\omega}_{ib}^b}$ is the random white noise in the gyroscope angular rate vector.

2.4.3 Other Sensors

GNSS receivers and IMU sensors complement each other in terms of the position/attitude information they provide and the environments that they operate well in. As such, PosAtt systems typically consist of a GNSS receiver and IMU sensor at their core. Other position/orientation sensors are often used to supplement the information provided by GNSS receivers and IMU sensors, particularly for applications in GNSS-degraded environments (e.g. indoor positioning).

Wheel odometry is often used for land vehicles, and relies on tracking the rotation of the land vehicles' wheels, which is directly related to the traveled distance [Brossard and Bonnabel, 2019]. Additionally, quadrature-encoding on wheel odometers provides information about the direction of motion [Sibai et al, 2012]. For cars, it is common to have wheel odometers mounted on the rear wheels, although there has been research that has used four wheel odometers on a car to assist in sensitive maneuvers such as parking [Brunker et al, 2018]. The use of wheel odometers is complicated by the fact that they are mounted to the wheel, and are therefore often separated from the positioning system by the car's suspension.

Imaging sensors and Light Detection and Ranging (LiDAR) sensors are often at the core of mobile mapping systems, and they may also be used for odometry.

Visual Odometry (VO) uses information from imaging sensors to estimate relative changes in a PosAtt system's position and attitude. VO approaches can generally be divided into one of two categories [Aqel et al, 2016]:

- i. Feature-based approaches: these VO approaches rely on identifying and extracting distinctive image features, and tracking their change in position/orientation across multiple image frames. ORB-SLAM is a prominent feature-based approach to VO [Mur-Artal et al, 2015].
- ii. Appearance-based approaches: these VO approaches rely on monitoring the changes in the appearance of image frames through time, instead of identifying features. This is often accomplished by using the changes in pixel intensity from one frame to the next. LSD-SLAM is a prominent appearance-based approach to VO [Engel et al, 2014].

LiDAR Odometry uses information extracted from a LiDAR point cloud as it is being collected to estimate relative changes in a PosAtt system's position and attitude. There are two main approaches to LiDAR Odometry, which are roughly analogous to the VO approaches [Lee et al, 2024]:

- i. Feature-based matching approaches: these approaches identify and extract distinctive features in successive LiDAR scans, and then perform a co-registration between the extracted features to determine the sensor's relative change in position and orientation. LOAM is a prominent example of this that uses points on sharp edges and planar surfaces to estimate the system's motion [Zhang and Singh, 2014].
- ii. Direct matching approaches: these approaches seek to maximize the registration between a new LiDAR scan and the already-mapped environment. This is generally accomplished through

the application of the Iterative Closest Point (ICP) algorithm, with LiTAMIN being a prominent example [Yokozuka et al, 2020]. Additionally, since each LiDAR “scan” is actually a series of individual point scans, a “continuous-time” odometry estimate may be determined by applying the ICP algorithm on each point individually as in CT-ICP [Dellenbach et al, 2022].

2.5 Kalman Filters

The Kalman Filter (KF) is a recursive estimation technique used to solve systems that have dynamic states by incorporating information governing how that system should change with time (i.e. dynamic/kinematic models) with the discrete observations connecting to the current state of the system (i.e. measurement models). KFs form the backbone of most PosAtt systems, both under the Traditional Multi-Sensor Integration Strategy (TMIS) detailed in Section 2.6.2 and under the Generic Multi-Sensor Integration Strategy (GMIS) detailed in Section 2.6.3. The recursive estimation process in Kalman filtering consists of two primary steps that are carried out epoch-wise:

i. The Time Update:

The Time Update (also frequently referred to as the Prediction Step) is where the system model is used to predict the current system state based upon the most recently estimated system state.

ii. The Measurement Update:

The Measurement Update (also frequently referred to as the Update Step) is where the measurement models for any observations are used to optimally refine the predicted system state estimate from the Time Update.

The Standard Kalman Filter (SKF) uses linear system model equations and linear measurement model equations to define a dynamic system. For PosAtt systems, these models are rarely linear and the SKF is generally insufficient for modeling such systems accurately. As a result, the Extended Kalman Filter (EKF) is often used instead, since it can directly use the non-linear system and observation models. In fact, the SKF may be considered to be a special case of the EKF where the system and observation models are linear. As such, the EKF is presented in Section 2.5.1.

2.5.1 Extended Kalman Filter (EKF)

Generally, a Kalman Filter seeks to estimate the time-varying system state based upon an initial state $\mathbf{x}(0)$ and its covariance $\mathbf{P}(0)$. As opposed to the standard KF, the Extended Kalman Filter (EKF)

makes use of non-linear system model equations and non-linear measurement model equations in the KF estimation process.

The EKF ties the system state at epoch $k + 1$ to the system state at epoch k through the system model, defined as

$$\mathbf{x}(k + 1) = \mathbf{A}(\mathbf{x}(k)) + \mathbf{B}(\mathbf{w}(k)) \quad (2.48)$$

where:

$\mathbf{x}(k)$ is the estimate of the system state at epoch k ;

$\mathbf{A}(\mathbf{x}(k))$ is the non-linear transition function describing the system model of how the system state vector changes from epoch k to epoch $k + 1$;

$\mathbf{B}(\mathbf{w}(k))$ is the non-linear process noise transition function describing the overall contribution of the process noise to errors in the system state estimate; and

$\mathbf{w}(k)$ is the system's process noise at epoch k , and is assumed to be white noise.

The system state is also connected to a set of observations at epoch $k + 1$ via the measurement model

$$\mathbf{z}(k + 1) = \mathbf{C}(\mathbf{x}(k + 1)) - \mathbf{\Delta}(k + 1) \quad (2.49)$$

where:

$\mathbf{z}(k + 1)$ is a vector containing all observations that were recorded at epoch $k + 1$;

$\mathbf{C}(\mathbf{x}(k + 1))$ is the non-linear measurement function describing the relationship between the observations at epoch $k + 1$ and the system state vector; and

$\mathbf{\Delta}(k + 1)$ is the observation noise vector at epoch $k + 1$, and is assumed to be white noise.

The KF process begins by *predicting* the system state using the kinematic model in (2.48), yielding the time update equation

$$\hat{\mathbf{x}}(k + 1, k) = \mathbf{A}(\hat{\mathbf{x}}(k)) \quad (2.50)$$

with an associated covariance matrix of

$$\mathbf{P}(k+1, k) = \mathbf{A}(k+1, k)\mathbf{P}(k)\mathbf{A}^T(k+1, k) + \mathbf{B}(k+1, k)\mathbf{Q}(k)\mathbf{B}^T(k+1, k) \quad (2.51)$$

where:

$\hat{\mathbf{x}}(k+1, k)$ is the predicted state vector at epoch $k+1$, given the previously-estimated state vector at epoch k .

$\mathbf{P}(k+1, k)$ is the covariance matrix of the estimate of the predicted system state vector;

$\mathbf{P}(k)$ is the covariance matrix for the estimated system state vector at epoch k ;

$\mathbf{Q}(k)$ is the covariance matrix of the process noise vector at epoch k ;

$\mathbf{A}(k+1, k)$ is the transition matrix resulting from the linearization of $\mathbf{A}(\mathbf{x}(k))$; and

$\mathbf{B}(k+1, k)$ is the process noise transition matrix resulting from the linearization of $\mathbf{B}(\mathbf{w}(k))$.

The predicted system state in (2.50) may be corrected to the best estimate of the system state at epoch $k+1$ by applying the measurement update

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \hat{\mathbf{x}}(k+1, k) + \mathbf{K}(k+1)[\mathbf{z}(k+1) - \mathbf{C}(\hat{\mathbf{x}}(k+1, k))] \\ &= \hat{\mathbf{x}}(k+1, k) + \mathbf{K}(k+1)\mathbf{d}(k+1, k) \end{aligned} \quad (2.52)$$

Where $\mathbf{d}(k+1, k)$ is the system innovation, and $\mathbf{K}(k+1)$ is the Kalman gain matrix, which is defined to be

$$\mathbf{K}(k+1) = \mathbf{P}(k+1, k)\mathbf{C}^T(k+1)[\mathbf{C}(k+1)\mathbf{P}(k+1, k)\mathbf{C}^T(k+1) + \mathbf{R}(k+1)]^{-1} \quad (2.53)$$

Where $\mathbf{C}(k+1)$ is the design matrix resulting from the linearization of $\mathbf{C}(\mathbf{x}(k+1))$ in (2.49) and $\mathbf{R}(k+1)$ denotes the covariance matrix of the observation vector at epoch $k+1$.

The covariance matrix for $\hat{\mathbf{x}}(k+1)$ is given by the equation

$$\mathbf{P}(k+1) = (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}(k+1))\mathbf{P}(k+1, k) \quad (2.54)$$

2.5.1.1 Error Analysis in the EKF

When analyzing errors in a Least-Squares (LS) problem, there is only one source of random error, namely the measurement noise, that impacts the estimation of the unknowns. In the KF, there are

three statistically independent sources of random error that affect the estimation of the system state vector at any given time instant:

- i. The errors in the direct observation vector – this error source is the same as the one in a LS problem, where the random errors in the observation vector need to be characterized and accounted for;
- ii. The errors in the process noise vector – this error source is due to the limitations of any system model in characterizing the system dynamics and is a modeling error. These errors are commonly assumed to conform to a zero-mean multivariate normal distribution; and
- iii. The uncertainty associated with the predicted state vector propagated from the previously-estimated system state vector – this error source represents the accumulated effect of all the random errors propagated from all previous epochs in Kalman filtering.

It is relatively straightforward to estimate the observation residuals for a KF application, since the observation residuals are the difference between the adjusted “ideal” observations (i.e. their estimated values) and their observed values. The residuals of the system state vector may be calculated similarly. However, estimating the residuals of the process noise vector in Kalman filtering is more involved, and requires reformulating the solution analytics to independently weigh these three independent types of the available stochastic information [Wang, 1997; Caspary and Wang, 1998]. Since this approach is very similar to the one used in this research to determine residual vectors for a system state constrained KF, the set-up of the derivation for the unconstrained KF is provided in this section.

For the purposes of deriving the three sets of residual vectors that describe the sources of independent errors in a KF, the epoch-wise KF problem is instead characterized as a LS problem with the following three sets of pseudo-observations:

$$\mathbf{l}_x(k) + \mathbf{v}_x(k) = \mathbf{A}(k+1, k)\hat{\mathbf{x}}(k) \quad (2.55)$$

$$\mathbf{l}_w(k) + \mathbf{v}_w(k) = \mathbf{w}(k) = \mathbf{0} \quad (2.56)$$

and

$$\mathbf{l}_z(k) + \mathbf{v}_z(k) = \mathbf{z}(k) \quad (2.57)$$

where

$\mathbf{l}_x(k)$ is the pseudo-observation vector corresponding to the predicted system state at epoch k ,

with its associated residuals denoted as $\mathbf{v}_x(k)$;

$\mathbf{l}_w(k)$ is the pseudo-observation vector corresponding to the process noise estimate at epoch k (this vector is commonly taken to be a zero vector, but here it could take on a different numerical value), with its associated residuals denoted as $\mathbf{v}_w(k)$;

$\mathbf{l}_z(k)$ is the pseudo-observation vector corresponding to the actual observations that would traditionally be included in the KF measurement models at epoch k , with its associated residuals denoted as $\mathbf{v}_z(k)$.

Note that, from (2.55) forward, specification of the epoch is left out of the notation where it is not ambiguous (e.g. $\mathbf{A}(k + 1, k)$ is instead written as $\mathbf{A}$).

The covariance matrices associated with these pseudo-observation vectors are

$$\mathbf{D}_{l_x} = \mathbf{A}\mathbf{P}(k)\mathbf{A}^T \quad (2.58)$$

$$\mathbf{D}_{l_w} = \mathbf{Q} \quad (2.59)$$

and

$$\mathbf{D}_{l_z} = \mathbf{R} \quad (2.60)$$

[Wang, 1997; Caspary and Wang, 1998].

Combining all three sets of pseudo-observations to make the system of equations defining the KF yields

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_x \\ \mathbf{l}_w \\ \mathbf{l}_z \end{bmatrix} + \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_w \\ \mathbf{v}_z \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k + 1) \\ \mathbf{w} \end{bmatrix} \quad (2.61)$$

The system of equations in (2.61) defines an overdetermined system which can be solved via LS to achieve exactly the same KF measurement update equation as defined in Section 2.5.1. While the resultant equation for the estimated system state vector is identical to the pre-existing equations, modeling the three independent sources of error in the KF as part of the pseudo-observations allows for the additional direct calculation of the observation residuals, residuals of the predicted system state vector, and process noise residuals. These may be calculated via the equations

$$\hat{\mathbf{v}}_x = \mathbf{P}(k + 1)\mathbf{D}_{x(k+1,k)}^{-1}\mathbf{K}\mathbf{d} \quad (2.62)$$

$$\hat{v}_w = QB^T D_{x(k+1,k)}^{-1} Kd \quad (2.63)$$

and

$$\hat{v}_z = (CK - I)d \quad (2.64)$$

[Wang, 1997; Caspary and Wang, 1998].

By applying Covariance Law, these residual vectors may be determined to have propagated covariance matrices of [Wang, 1997; Caspary and Wang, 1998]

$$D_{v_x} = P(k+1)C^T D_d^{-1} CP(k+1) \quad (2.65)$$

$$D_{v_w} = QB^T C^T D_d^{-1} CBQ \quad (2.66)$$

and

$$D_{v_z} = (I - CK)R \quad (2.67)$$

These three sets of residual vectors and their associated covariance matrices form the basis of any comprehensive error analysis scheme for the EKF.

2.5.1.1.1 Residual Analysis

In an EKF, there are three quantities that may have their residual values analyzed, in accordance with the residual vectors calculated in Section 2.5.1.1:

- i. **Observations:** Observation residuals are generally indicative of either the quality of the specific measurements in the observation vector, or of how well these measurements fit the prescribed measurement model equations.
- ii. **Process Noise:** Process noise residuals generally indicate either significant shifts in system dynamics, or how well the system model equations reflect the actual dynamics of the system.
- iii. **System State:** Residuals of the predicted system state vector provide insight into the accuracy of the previously estimated system state vector. Analyzing the residuals of the predicted system state vector tends to be complicated due to the fact that individual elements of the system state vector tend to be correlated with one another.

In practice, the observation residuals are the most commonly used quantity in EKF residual analysis. For some real-time processing applications, the system innovation vector is instead used to

represent the observation quality, despite the fact that the system innovation includes observation errors, process noise errors, and predicted system state errors.

The most common form of residual analysis is for outlier detection and management, and this can be accomplished through a variety of methods:

- Outliers may be detected via statistical tests on the individual elements of the residual/system innovation vector under their assumed statistical distribution to determine whether or not these elements constitute outliers [Groves, 2013]. Here, if the individual elements of $\hat{\mathbf{v}}_z$ or $\mathbf{d}$ are greater than some threshold ξ_z , the corresponding observation is considered an outlier and removed from processing. Using the system innovation vector has an additional benefit in this process in that outlier removal may be performed before the KF measurement update is applied, thereby reducing computational load for real-time applications.
- Outliers may be managed by modifying the estimated variance of suspected outliers in an Adaptive KF. Here, suspected outliers are assigned a larger variance, thereby lowering its corresponding weight in the measurement update [Thomas et al, 2000; Yang et al, 2001; Le et al, 2003]. This weight modification has also been accomplished by updating observation variance estimates in real-time using a sigmoid function [Lv et al, 2024], or by updating observation variance estimates using Iterative ReWeighted Least-Squares (IRWLS) [Tan et al, 2020].
- Outliers may also be managed by modifying the system innovation values for potential outliers prior to applying the KF measurement update. This has been accomplished by introducing a “saturation function” that imposes a hard limit on the system innovation’s absolute value [Fang et al, 2021; Jiang et al, 2021].
- When the system is highly overdetermined, robust methods such as RANSAC may be applied to cope with high proportions of outlying observations to find the solution that provides the largest set of coherent associated inlying observations [Civera et al, 2009; Civera et al, 2010; Kitt et al, 2010; Doer and Trommer, 2020].

2.5.1.1.2 Variance Component Estimation and Reliability Analysis

Variance Component Estimation (VCE) provides a measure of how well the *a priori* assumed accuracies of system state quality, process noise, and observations lines up with their *a posteriori*

residual estimates. In Variance Component Analysis, the true covariance matrix of an observation vector Σ is taken to be the sum of multiple scaled cofactor matrices D_i , expressed as

$$\Sigma = \sum_{i=1}^n \Sigma_i = \sum_{i=1}^n \sigma_i D_i \quad (2.68)$$

where σ_i represents the i th variance component [Förstner, 1979; Searle, 1995; Teunissen and Amiri-Simkooei, 2007].

A LS solution's measurement residual vector may be directly calculated via the annihilator matrix

$$F = I - C(C^T R^{-1} C)^{-1} C^T R^{-1} \quad (2.69)$$

Using the annihilator matrix, the LS cost function in may be written as

$$\Phi = l^T W \Sigma W l \quad (2.70)$$

where $W := F \Sigma^{-1} = \Sigma^{-1} F^T$ is a symmetric matrix. Substituting (2.68) into (2.70) yields

$$\Phi = \sum_{i=1}^n \omega_i = l^T W \left(\sum_{i=1}^n \Sigma_i \right) W l \quad (2.71)$$

Since Φ is a scalar value, it is equal to the value of its trace. Using this, and the fact that $\text{tr}(AB) = \text{tr}(BA)$, (2.71) may be rewritten as

$$E\{\Phi\} = \text{tr} \left(W \sum_i \Sigma_i W E\{l l^T\} \right) \quad (2.72)$$

where $E\{x\}$ denotes the expected value of x . This expression may be further simplified to

$$E\{\Phi\} = \sum_i \sum_j \sigma_j \text{tr}(W \Sigma_i W D_j) \quad (2.73)$$

Equating (2.71) with (2.73) provides the following equation for each ω_i :

$$\omega_i = \sum_j \sigma_j \text{tr}(W \Sigma_i W D_j) \quad (2.74)$$

The expression in (2.74) may be used to generate a system of equations to directly estimate values for σ_j . The process of estimating these variance component factors can be simplified when the

variance components are statistically independent from one another and satisfy $\mathbf{W}\boldsymbol{\Sigma}_i\mathbf{W}\mathbf{D}_j = \mathbf{0} \forall i \neq j$. This results in the simplified equation

$$\omega_i = \sigma_i \text{tr}(\mathbf{W}\boldsymbol{\Sigma}_i\mathbf{W}\mathbf{D}_i) \quad (2.75)$$

Rearranging to isolate σ_i and expanding ω_i gives the following estimate of the i -th variance component factor:

$$\hat{\sigma}_i = \frac{\hat{\mathbf{v}}^T \boldsymbol{\Sigma}_i^{-1} \hat{\mathbf{v}}}{\text{tr}(\mathbf{W}\boldsymbol{\Sigma}_i\mathbf{W}\mathbf{D}_i)} \approx \frac{\hat{\mathbf{v}}^T \mathbf{D}_i^{-1} \hat{\mathbf{v}}}{\text{tr}(\mathbf{W}\mathbf{D}_i\mathbf{W}\mathbf{D}_i)} \quad (2.76)$$

where it is assumed that $\boldsymbol{\Sigma}_i \approx \mathbf{D}_i$. When the estimated cofactor matrix components are sufficiently different from their associated covariance matrix components, (2.76) may be iteratively applied until each value of $\hat{\sigma}_i$ converges [Förstner, 1979; Teunissen and Amiri-Simkooei, 2007].

This derivation is valid when applying the LS principle to solve a problem. The reformulation of the KF as a LS problem in Section 2.5.1.1 allows for the direct application of the equations for VCE in a KF. Given the definition of the pseudo-observations in Section 2.5.1.1, scaling factors may be estimated for observation variance components and process noise variance components. System state variance components are not practical to estimate, given the general correlation between system state elements.

There are many ways to define the variance components for a KF system. [Wang et al, 2009] conducted VCE for both static and kinematic single point positioning GPS data, and followed up with an exploration of VCE for relative GPS positioning [Gopaul et al, 2010]. [Yang and Xu, 2003] used separate variance components to characterize the predicted system state vector's and observation vector's overall quality. [Wang et al, 2010] used VCE to determine separate scaling factors for each GPS satellite for double-differenced pseudo-range and carrier phase measurements. [Gao et al, 2016; Bahadur and Nohutcu, 2021] determined separate scaling factors to characterize the performance of pseudo-range and carrier phase measurements for different GNSS constellations. [Wang et al, 2021] conducted VCE, separating variance components for the process noise components and GPS/IMU observables. [Wang et al, 2010, 2021; Yu et al, 2018; Qian et al, 2016] used VCE in an adaptive KF to separately estimate individual elements of the diagonal observation and process noise covariance matrices.

2.5.2 Other Filtering Techniques

The EKF provides a means to modeling non-linear models, but relies on neglecting higher-order terms in the Taylor Series expansion of these models. For highly non-linear system or observation

models, this results in a significant reduction in the reliability of the EKF, where these higher-order terms are significant. Both the Unscented Kalman Filter (UKF) and Particle Filters were developed to account for highly non-linear system and observation models.

2.5.2.1 *Unscented Kalman Filter (UKF)*

The Unscented Kalman Filter (UKF) was developed under the assumption that Gaussian distributions are easier to approximate than arbitrary non-linear functions [Julier and Uhlmann, 1997]. Where the EKF approximates the non-linear models to determine closed-form estimates of the stochastic properties of the estimated state vector, the UKF uses exact representations of the non-linear models to determine approximations of the stochastic properties of the estimated state vector. In this way, the UKF takes the “reverse” of the approach taken by the EKF.

At its base, the UKF uses the Unscented Transform to model the propagation of uncertainty through the non-linear system and observation models. The Unscented Transform applies the non-linear model equations to a set of pre-determined points. Propagating these pre-determined points through the system and observation models yields a transformed set of points, and these may be used to approximate their statistical distribution [Julier and Uhlmann, 1997].

The UKF still follows the same basic structure of predicting the state vector using the system model through the KF time update, and then correcting that prediction using the KF measurement update.

Before the time update, the Unscented Transform is applied to the system state estimate for epoch k . The transformed state estimate is propagated through the non-linear system model, producing a set of predicted values that may be used to determine an estimate of the predicted state vector and its covariance. The predicted state vector is then propagated through the non-linear measurement model, and the estimates of the predicted observation vector that are produced may be used to estimate the system innovation vector and its covariance, allowing for the application of the measurement update.

The UKF has seen significant use in the field of indoor positioning, using a variety of sensor platforms. It is commonly used for processing ultrawideband (UWB) and pseudolite observation models, which typically have their quality be heavily impacted by the observing environment (e.g. significant multipath errors) [Khan et al, 2017; Li et al, 2019; Feng et al, 2020]. The UKF’s efficiency, even for highly unreliable or non-linear systems makes it particularly well-suited for these applications.

Additionally, the UKF has seen recent application in the field of autonomous vehicles, specifically in interpreting and processing information from Intelligent Connected Vehicle networks [Sun et al, 2023; Wang et al, 2023]. Cooperative navigation of shared spaces such as intersections presents a challenging problem for autonomous navigation, and the potential to use inter-connectivity between intelligent vehicles has significant potential to make improvements in this area, particularly as autonomous driving technologies see wider adoption. The networks created by these inter-connected systems are highly dynamic and their structure can change significantly in a short span of time. The efficiency and robustness of the UKF therefore makes it an ideal approach to processing these challenging datasets.

2.5.2.2 Particle Filter (PF)

Particle Filters (PFs), also referred to as sequential Monte Carlo methods, approximate the distributions of quantities in the filter using “particles,” or discrete random samples. One of the defining parameters of a PF is the number of particles being used, and particles are used to approximate the continuous distribution of potential system states.

Each particle may be propagated through the system model to generate predictions of the state vector at epoch $k + 1$ as a Time Update in the PF. These predictions may be propagated through the observation model to estimate the conditional probability of the observations at epoch $k + 1$ given the particle’s postulated trajectory. This estimation is analogous to the Measurement Update. The particles are then classified by their feasibility [Gustafsson, 2010].

PFs have a large advantage in terms of their general applicability to non-linear systems and to non-Gaussian noise distribution. The ability of a PF to approximate the distribution of the system state is completely determined by the number of particles being generated and the resampling method being used. Where PFs struggle to perform is in high-dimensional problems, where the number of necessary particles to provide a reasonable approximation to the state distribution increases dramatically.

PFs have seen broad adoption across many different application areas, even seeing use in real-time processing [Elfring et al, 2021].

PFs are well-suited to applications in which large ambiguities in the filter solution may persist through large stretches of time. Since PFs evaluate the feasibility of an entire trajectory, ambiguous alternatives may be propagated forward and resolved once additional contextual information is available. This includes situations where there is potential for significant outlying observations to remain

undetected at their associated observation epoch, where PFs allow for the evaluation of solution consistency at one instant in time with subsequent observation epochs. This situation arises naturally in many indoor positioning applications using ultrawideband (UWB) ranging, which commonly use PFs to account for high uncertainties in the ranging observations [Carrera Villacrés et al, 2019; Shen et al, 2020; Shi et al, 2023]. This has also been applied to indoor visual odometry [Mráz et al, 2024].

Map-aware positioning has also been an active area of research that uses PFs. Identifying local similarity in the three-dimensional maps used in map-aware positioning often results in ambiguous identification of location, and PFs are well-suited to resolving this ambiguity. PFs have been applied to outdoor map-aided positioning as in [Suzuki, 2016; Zhong and Groves, 2022a; Zhong and Groves, 2022b] and indoor map-aware localization as in [Junoh et al, 2021].

2.6 Multi-Sensor Integration Strategies

The goal of every Multi-sensor Integration Strategy (MIS) in positioning is to use data from multiple position/orientation sensors to determine an optimal PosAtt solution that maximizes solution availability, reliability, and accuracy. This is a very challenging task that typically requires many compromises and concessions; there is no one-size-fits-all solution.

There are two existing integration strategies: the Traditional Multi-sensor Integration Strategy (TMIS) and the Generic Multi-sensor Integration Strategy (GMIS). These strategies may be further classified by the specific processing techniques adopted, and these additional classifications are presented in Section 2.6.1. An overview of the TMIS in Section 2.6.2 and an overview of the GMIS is presented in Section 2.6.3, followed by a general comparison between them in Section 2.6.4.

This section provides an overview of existing multi-sensor integration strategies, including both the TMIS and the GMIS.

2.6.1 Classification of Multi-sensor Integration Architectures

Multi-sensor integration architectures may generally be classified by the definition and structure of the system state vector, and by the processes of fusing data from aiding sensors. Sections 2.6.1.1 and 2.6.1.2 explore these broad classifications.

2.6.1.1 Classification By Definition of the System State Vector

One of the defining features of a KF-based integration strategy is in the definition of the system state vector. There are two prevailing types of system state vectors used in positioning KFs, which result in the following broad categorizations:

- i. Indirect KF: This refers to a KF where the system state vector is defined to be the system's error states, meaning that the quantities of interest for positioning are indirectly, rather than directly, estimated.
- ii. Direct KF: This refers to a KF where the system state vector is defined such that it provides a direct estimate of the quantities of interest for positioning.

[Wendel et al, 2001; Qi and Moore, 2002; Giroux et al, 2005; Trawny and Roumeliotis, 2005].

Under the indirect KF, the whole state vector of interest is expressed as the sum of an initial value $\mathbf{x}_0$ and an error state $\Delta\mathbf{x}$ as

$$\hat{\mathbf{x}} = \mathbf{x}^{(0)} + \Delta\mathbf{x} \quad (2.77)$$

where $\hat{\mathbf{x}}$ is the optimal estimate of the whole state vector. Considering the definition of this optimal estimate in (2.77), the system state equations in (2.48) may be approximated and linearized using their first-order Taylor Series expansion

$$\dot{\mathbf{x}}_0 + \Delta\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}_0) + \mathbf{B}(\mathbf{w}_0) + \mathbf{A}(k+1, k)\Delta\mathbf{x} + \mathbf{B}(k+1, k)\Delta\mathbf{w} + \delta\mathbf{w} \quad (2.78)$$

where $\mathbf{w} = \mathbf{w}_0 + \Delta\mathbf{w}$ and $\delta\mathbf{w}$ is the error from the higher-order terms of the Taylor Series expansion.

Taking the definition of $\dot{\mathbf{x}}_0 = \mathbf{A}(\mathbf{x}_0) + \mathbf{B}(\mathbf{w}_0)$, (2.78) may be simplified and rewritten as

$$\Delta\dot{\mathbf{x}} = \mathbf{A}(k+1, k)\Delta\mathbf{x} + \mathbf{B}(k+1, k)\Delta\mathbf{w} + \delta\mathbf{w} \quad (2.79)$$

which constitutes the error state system model, and is analogous to the linearized system model for the EKF.

Using a similar logic, the measurement model in (2.49) may be linearized about the predicted system state vector as

$$\mathbf{z} = \mathbf{C}(\mathbf{x}(k+1)) - \mathbf{C}\Delta\mathbf{x} + \mathbf{v}_z + \delta\mathbf{z} \quad (2.80)$$

where $\delta \mathbf{z}$ is the error from the higher-order terms of the Taylor Series expansion of the measurement equations. Rearranging (2.80) and defining $\Delta \mathbf{z} := \mathbf{C}(\mathbf{x}(k+1)) - \mathbf{z}$ gives the following equation for the measurement error vector:

$$\Delta \mathbf{z} = \mathbf{C} \Delta \mathbf{x} + \mathbf{v}_z + \delta \mathbf{z} \quad (2.81)$$

Central to the Indirect KF approach is the generation of the measurement error vector that is tied to the system error states. In positioning and navigation, this is traditionally accomplished using an IMU/GNSS integrated system, with the IMU acting as the core positioning sensor. The raw IMU measurements are fed through the IMU mechanization equations to determine the IMU-only positioning solution, and this estimated position may be used to generate error measurements for observations from the GNSS receiver. It is important to note that, while GNSS/IMU pairings are the most common sensor pairing, other sensor pairings are also used in practice, including camera/IMU integrated systems [Sirtkaya et al, 2013] and LiDAR/IMU integrated systems. For all positioning and navigation Indirect KFs, the KF measurement update can only be performed when there are available positioning observations from both positioning sensors.

For positioning and navigation applications, the system error states need to account for the system position and attitude, but also for the IMU sensor errors. This is particularly important for lower-cost IMU sensors, which exhibit less stable behaviour over time. When using an Indirect KF, there are two prevailing methods of accounting for the IMU sensor errors:

- i. Open-Loop: Under this scheme, the system error states are fed forward as corrections to the positioning solution from the INS/optimal estimator, with the basic workflow described by Figure 2.6.1.

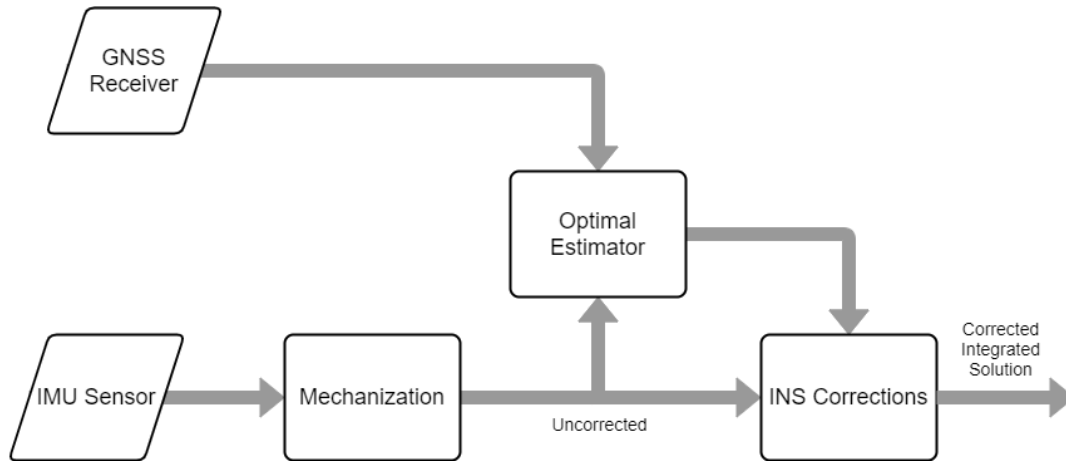


Figure 2.6.1: The basic workflow for the open-loop, feed-forward INS using an Indirect KF. Adapted from [Tan et al, 2007; Sirtkaya et al, 2013].

- ii. Closed-Loop Feed-Backward: Under this scheme, the system error states are fed backward as corrections to the IMU mechanization equations themselves, with the basic workflow described by Figure 2.6.2.

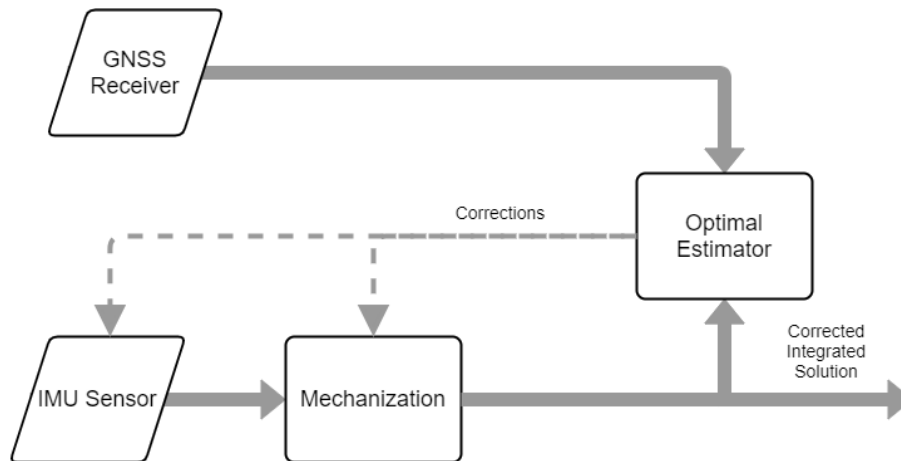


Figure 2.6.2: The basic workflow for the closed-loop, feed-back INS using an Indirect KF. Adapted from [Tan et al, 2007].

In contrast to the Indirect KF, the Direct KF involves the direct estimation of the system state vector, rather than the system error states. In a Direct KF, the KF equations detailed in Section 2.5.1 may be directly applied, rather than requiring the generation of error measurements. Practically, this means that the specific sensor configuration when using a Direct KF is much more flexible, since sensors do not need to be paired to generate error measurements; however, this also means that positioning and orientation observations must be modeled directly in the KF, which can present additional challenges in terms of added computational load.

Unlike the Indirect KF, the Direct KF allows for the direct modeling of all raw IMU measurements, rather than necessarily relying on the INS output, which is a property of Direct KFs that is exploited in the Generic Multi-Sensor Integration Strategy (GMIS). While this provides distinct advantages when it comes to analyzing the errors present in raw IMU observations, it does come at the cost of increased computational load, since modeling IMU observations directly requires that the KF measurement update be performed at the same rate as the IMU sensor's data rate. This makes real-time processing more challenging when directly modeling raw IMU observations.

Under the TMIS, the IMU raw measurements are not modeled directly in the Direct KF, but rather the INS output is modeled as measurements in the KF. This circumvents the added computational load from directly modeling raw IMU observations, since the KF measurement update may be performed at a lower frequency than the IMU sampling frequency.

A comparison between the Indirect and Direct KF is provided in Table 2.6.1 [Maybeck, 1982].

Table 2.6.1: Comparison between the Direct and Indirect Kalman Filters in positioning and navigation applications.

	Advantages	Disadvantages
Indirect KF	- Provides an accurate positioning solution when INS mechanization is accurate. - Has a lower computational load due to less frequent KF measurement updates.	- Has difficulty detecting measurement blunders, particularly in IMU measurements. - Its system model accuracy is limited by the accuracy of the INS mechanization.
Direct KF	- Provides more in-depth information for statistical analysis and blunder detection. - Mitigates random walk errors more effectively in the event of GNSS outages or gaps.	- Has a higher computation load when conducting measurement updates at the same rate as higher-rate positioning sensors.

2.6.1.2 Classification By Definition of Aiding Sensors

When using an Indirect KF, the IMU typically acts as the core sensor, with another positioning sensor acting as an aiding sensor to generate error measurements. The choice of aiding sensor has a significant impact on the positioning solution quality and positioning performance in different operating environments. Combining an aiding sensor with a core IMU should ideally result in a system where core and aiding sensors provide unique and complementary information, producing more accurate results than either sensor would be able to produce on their own [Titterton and Weston, 2004; Ramanandan et al, 2011]. It is therefore valuable to understand the unique benefits and drawbacks of each positioning sensor, to identify what they can uniquely contribute to an integrated PosAtt system. Table 2.6.2 provides an overview of some of the unique attributes to different positioning sensors. There are other aiding sensors that may be used that are not mentioned in Table 2.6.2. These include RADAR, external tracking information, speed sensors, altimeters, etc.

Table 2.6.2: Comparison of some of the unique sensor attributes of different positioning sensors.

Sensor	Unique Attributes
IMU	- IMU-only solution suffers from unbounded random walk error - Provides both position and attitude information - Good solution coverage and availability
GNSS	- GNSS-only solution provides a non-wandering estimate with bounded error - Provides only position information - Cannot operate indoors or when there is significant overhead cover (e.g. urban “canyons,” forested areas, etc.)
Camera	- Visual Odometry solution suffers from random walk error, but can be bounded by revisiting locations over the operating period - Provides positioning and attitude information - Provides information about the camera’s environment, allowing for both positioning and mapping - Can operate indoors and outdoors

	- Generally higher computational load to interpret images for the extraction of positioning information
Laser Ranging	- Laser ranging positioning solution suffers from random walk error, but can be bounded by revisiting locations over the operating period - Provides positioning and attitude information - Provides information about the laser ranging sensor's environment, allowing for both positioning and mapping - Can operate indoors and outdoors - Generally higher computational load due to the volume of data being collected.

The ways in which aiding sensor data is integrated with core sensor data, or the fusion/integration architecture, also has a significant impact on positioning performance. There are four prevailing classes of integration architecture for multi-sensor systems:

- **Uncoupled, or snapshot fusion integrated systems:** Under this integration architecture, the independent positioning solutions from all sensors are fused through an estimator to produce the integrated positioning solution. The estimator can take multiple forms, ranging from a LS optimized estimator to identifying the most promising candidate solution from those available. A workflow diagram detailing the snapshot fusion integration architecture is shown in Figure 2.6.3 [Groves, 2013].

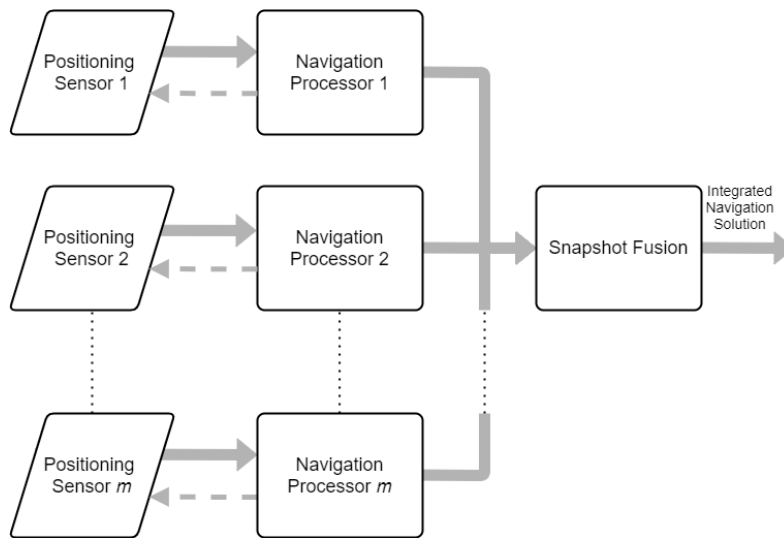


Figure 2.6.3: Workflow of the snapshot fusion integration architecture [Groves, 2013].

- Loosely coupled, or cascade integrated systems: This integration architecture is very similar to the uncoupled architecture, but rather than the sensors' positioning solutions being fully independent, they are instead integrated through a KF, whose integrated solution is then fed back into the individual navigation processors. A workflow diagram detailing the loosely coupled integration architecture is shown in Figure 2.6.4 [Groves, 2013].

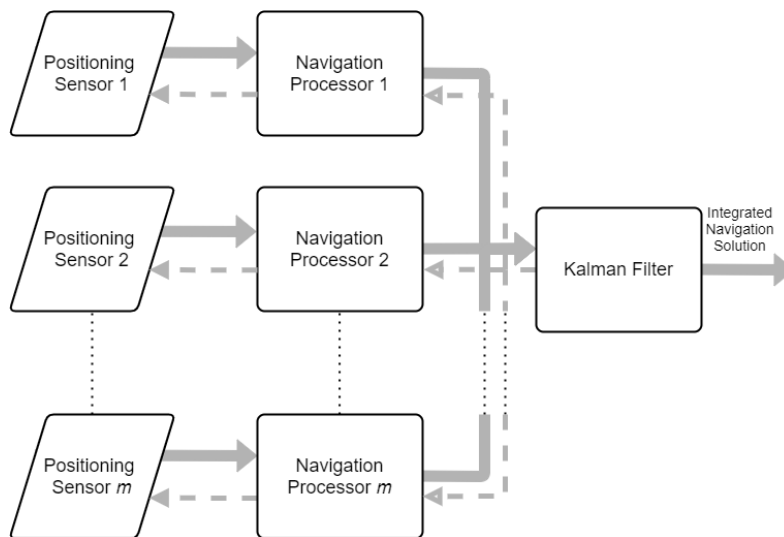


Figure 2.6.4: Workflow of the cascade fusion integration architecture [Groves, 2013].

- Tightly coupled, or centralized integrated systems: This integration architecture is similar to the loosely coupled architecture, with the main difference being that positioning sensors' observations are modeled directly in the integration KF rather than the integration KF

processing the processed positioning solutions from each sensor. A workflow diagram detailing the tightly coupled architecture is shown in Figure 2.6.5 [Lashley and Bevly, 2013].

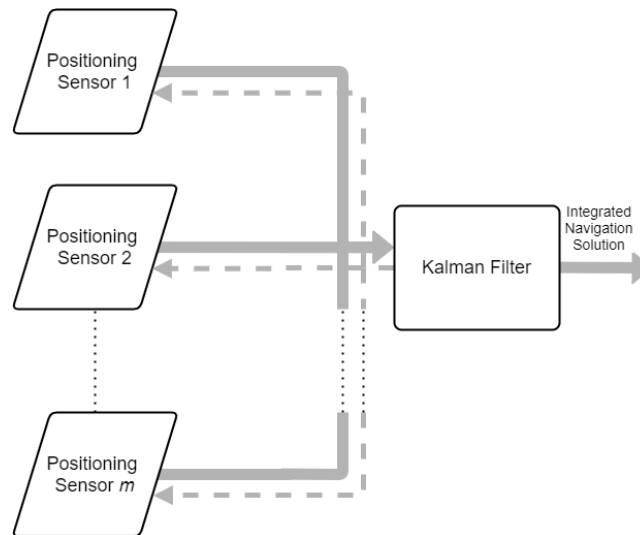


Figure 2.6.5: Workflow of the centralized fusion integration architecture [Lashley and Bevly, 2013; Breuer et al, 2016; Konrad et al, 2017].

- Deeply coupled, or federated integrated systems: This integration architecture makes use of local KFs to determine an optimal positioning solution for each navigation sub-system, and these positioning solutions are then integrated through a master KF. The structure of this approach is very flexible and robust, but also comes with additional computational load and complexity to facilitate multiple KFs running concurrently. An example workflow diagram is shown in Figure 2.6.6 [Lashley and Bevly, 2013].

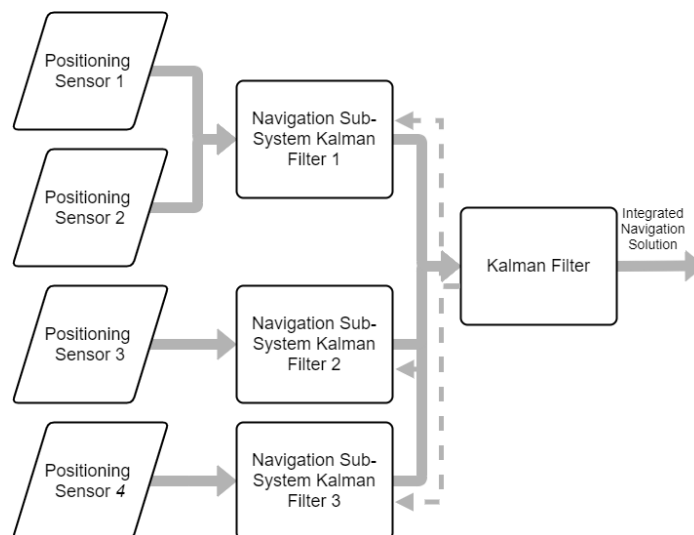


Figure 2.6.6: Workflow detailing a specific realization of the federated fusion integration architecture.

Additionally, outside the paradigm set by these four integration architectures, a hybrid integration architecture may be adopted that combines two or more of the architectures to leverage their specific strengths and mitigate their weaknesses. Such a hybrid system is more flexible in its overall structure, but is often highly specific to the vehicle's sensor configuration, since the structure of the overall workflow has to be adapted to the positioning sensors available [Groves, 2013].

2.6.2 The Traditional Multi-Sensor Integration Strategy (TMIS)

The Traditional Multi-Sensor Integration Strategy (TMIS) sees near-universal adoption across a wide variety of applications. This section provides an overview of the realization of the TMIS using indirect and direct Kalman Filters, as well as several unconventional realizations of the TMIS.

2.6.2.1 *The TMIS Using Indirect Kalman Filters*

Typically, sensor integration is performed using an Indirect KF that accounts for both navigational error states (i.e. position, velocity, and attitude error states) as well as error states corresponding to the systematic errors present in any positioning sensors. Under the specific scenario of integrating double-differenced GNSS and IMU measurements, the sensor systematic error states typically correspond to the IMU biases and scale factor errors.

The system error model and measurement error model are generally the defining features of the Indirect KF used for traditional sensor integration. In particular, the measurement error model chosen impacts whether sensor integration is approached using a loosely coupled or tightly coupled architecture.

There are a variety of system models that can be used to predict the system error states. Ultimately, the validity of any particular system model depends upon the actual expected system dynamics, and is therefore application-dependent. Specific system models have been developed for aircraft navigation [Koifman and Bar-Itzhack, 1999], navigation for precision agriculture applications [Backman et al, 2012], sea vessel navigation [Pietrzykowski et al, 2022], land vehicle navigation [Crocoll et al, 2014], navigation for UAV systems [Bryson and Sukkarieh, 2004], and even pedestrian navigation [Godha and Lachapelle, 2008]. Other system models have been developed to reduce the errors resulting from the removal of higher-order terms in the linearization of the system error models [Nassar, 2003].

For a GNSS/IMU integrated system, there are two predominant strategies to generating and modeling error measurements:

- Calculate the IMU-only positioning solution and the GNSS-only positioning solution. The error measurements in this case are taken to be the difference between these two positioning solutions, constituting a loosely coupled integration architecture [Scherzinger, 2000]. A workflow for the loosely coupled GNSS/IMU integration under the traditional integration strategy is shown in Figure 2.6.7.

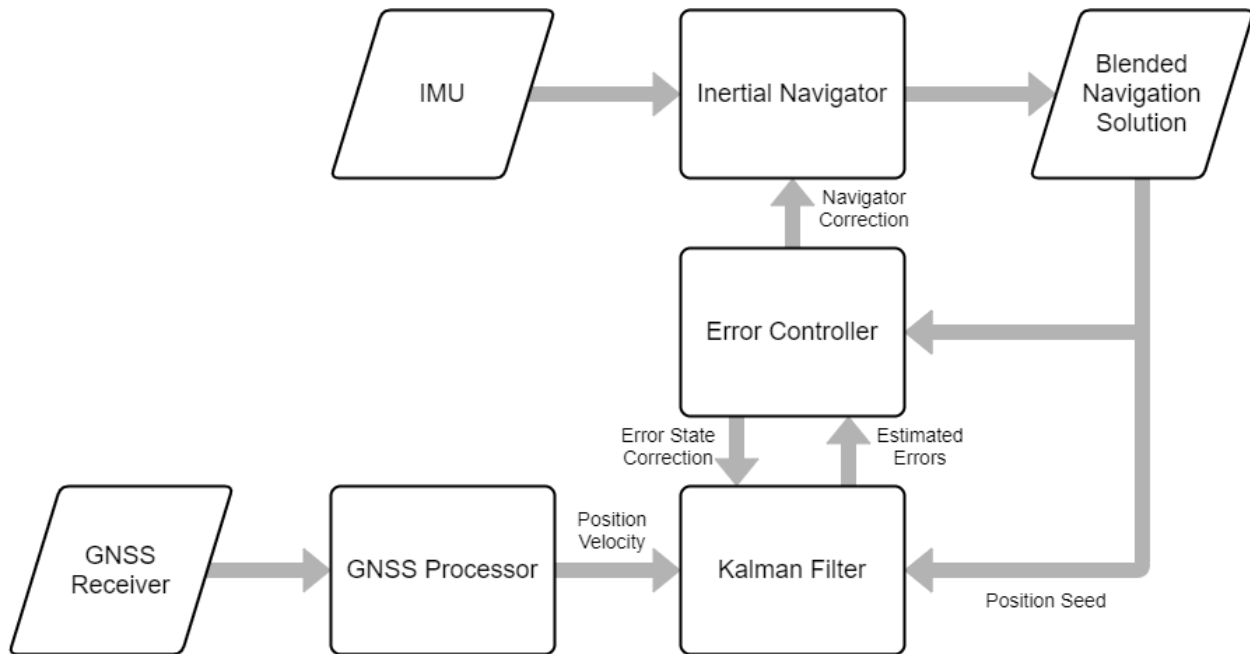


Figure 2.6.7: Workflow describing the loosely coupled GNSS/IMU integration under the traditional integration strategy, as per [Scherzinger, 2000].

- Predict the GNSS observables using the IMU-only positioning solution. The error measurements in this case are taken to be the difference between the predicted GNSS measurements and the actual GNSS measurements, constituting a tightly coupled integration architecture [Scherzinger, 2000]. A common workflow for the tightly coupled GNSS/IMU integration under the traditional integration strategy is shown in Figure 2.6.8.

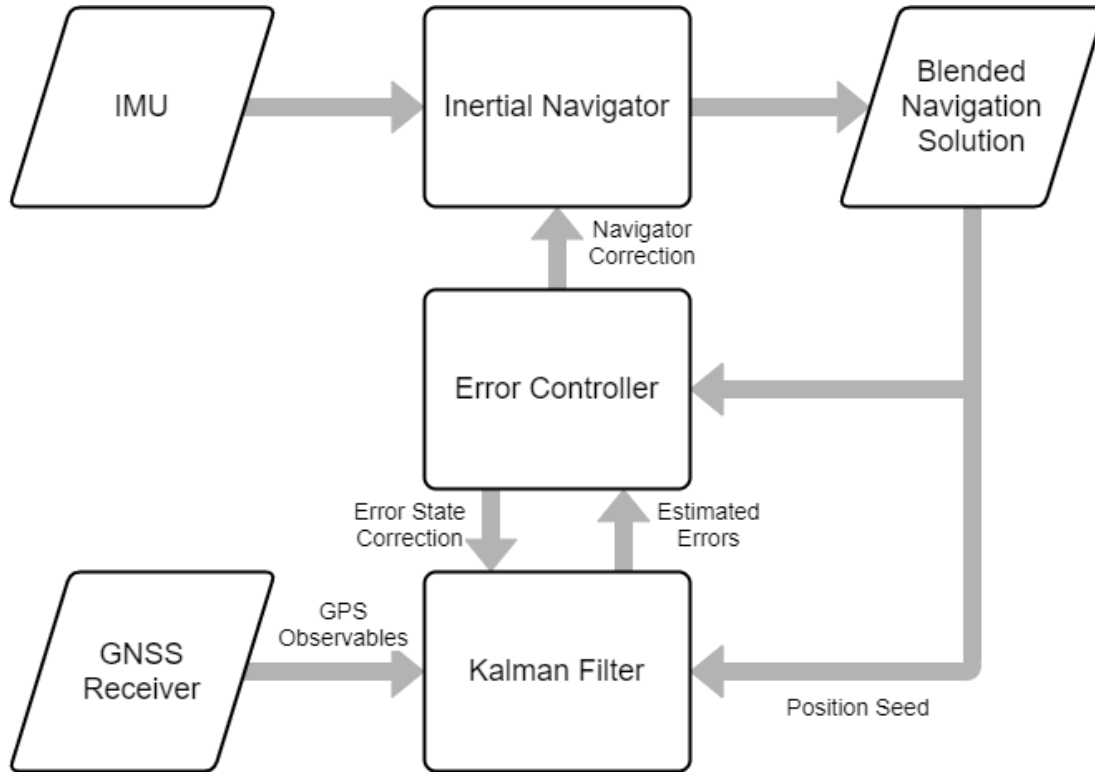


Figure 2.6.8: Workflow describing the tightly coupled GNSS/IMU integration under the traditional integration strategy, adapted from [Scherzinger, 2000; Quinchia et al, 2013; Deepika and Arun, 2018; Hamidi et al, 2020; Yi et al, 2022].

2.6.2.2 The TMIS Using Direct Kalman Filters

Generally, the Direct KF has not been commonly applied under the traditional sensor integration strategy due to the fact that applying a Direct KF for positioning is typically less efficient than estimating error states in an Indirect KF. When using a Direct KF under the traditional sensor integration strategy, there are two approaches that may be taken when selecting states for inclusion in the system state vector: full-state filters where all states of interest are modeled including system position, velocity, attitude, and systematic error parameters [Maybeck, 1982]; and compact-state filters which rely on fewer system state elements, allowing for more efficient estimation of the compact system state vector [Wagner and Wieneke, 2003].

The full-state Direct KF, as described in [Wendel, 2001], includes 21 elements in the system state vector, including the n -frame position, velocity, and attitude vectors, the true angular rate and specific force vectors in the IMU b -frame, and the systematic IMU errors. The full-state Direct KF is not common in practice, due to the inclusion of the true angular rate and specific force vectors in the system state

vector, which requires the KF to be run at the same rate as the core IMU sensor and increases the computational load.

The compact-state Direct KF addresses the additional computational load of the full-state Direct KF by removing the true angular rate and specific force vectors from the system state vector, and instead including them as a known input to the positioning KF. Under this realization, the positioning KF's system model is defined by the IMU mechanization between successive GNSS epochs, allowing it to account for highly non-linear motion between GNSS epochs [Wendel, 2001].

2.6.2.3 Other Techniques for Sensor Integration

There are unconventional realizations of the TMIS and this section focuses on three categories of alternative processing techniques: Dynamic Networks, Pose Graph Optimization, and Machine Learning-Based Strategies.

2.6.2.3.1 Dynamic Networks

The Dynamic Networks strategy for sensor integration has its roots in the analysis of geodetic control networks. Rather than performing an epoch-by-epoch estimate of the system position, the positions along the entire trajectory are solved estimated simultaneously through a LS estimation process [Friess, 1990; Seeber, 2003]. Since the Dynamic Networks strategy uses LS estimation, it does not use a system model, and instead models system kinematics as a relationship between different points in the trajectory [Rouzaud and Skaloud, 2011].

If the set of positions in the trajectory are given by $\{x_i\}, i \in 1, \dots, n$, then a constant velocity system model may be modeled as

$$\dot{x}_i = \frac{x_{i+1} - x_{i-1}}{t_{i+1} - t_{i-1}} \quad (2.82)$$

which may be encoded as conditions in a LS formulation of the problem alongside the measurement models for the positioning sensors being used [Rouzaud and Skaloud, 2011].

This method of trajectory estimation is particularly useful for ensuring consistency between the estimated trajectory and mapped environment in direct georeferencing applications [Rouzaud and Skaloud, 2011]. This is especially important for high-rate sensors such as LiDAR, where estimating the system's position/orientation more frequently than a constituent IMU sensor records observations is desirable [Rouzaud and Skaloud, 2011].

2.6.2.3.2 Pose Graph Optimization

Pose Graph Optimization (PGO) frames the goal of the positioning and navigation problem as estimating the poses that a mobile robot has at various instants in time, realized as nodes in a graph structure [Thrun and Montemerlo, 2006]. Defining poses in a graph structure allows for a natural model of the dynamics connecting subsequent poses in the graph, and also natural constraints for closed loops and revisited environments [Carlone et al, 2014]. As a processing approach, PGO developed from batch processing/bundle adjustment in Simultaneous Location and Mapping (SLAM) problems [Carlone et al, 2014].

The PGO problem is a Maximum Likelihood Estimation problem but presents a non-convex optimization problem where multiple local maxima exist [Carlone et al, 2014]. Additionally, the graph structure and optimization complexity increases dramatically with the number of nodes in the pose graph [Olson et al, 2006; Grisetti et al, 2009, 2010]. Practically, this makes constraining or simplifying the PGO problem necessary. The complexity of the PGO problem has been addressed by: decoupling the estimation of the pose orientations and positions [Sharp et al, 2004]; assuming small-angle changes in orientations between successive poses [Crandall et al, 2011]; performing estimation of pose orientations before refining the pose positions [Martinec and Pajdla, 2007; Crandall et al, 2011]; using windowed pose graphs [Li et al, 2019]; and exploiting the sparse nature of many PGO problems [Bai et al, 2021]. Additionally, the process of solving the PGO problem using matrix manipulation has been made much more efficient by [Moreira et al, 2021]. There has also been significant research in applying neural networks to solving the PGO problem [Li et al, 2019; Li et al, 2021; Azzam et al, 2021; Turkoglu et al, 2021].

The PGO problem additionally takes on different characteristics for different areas of application. In particular, many robotics applications involve planar motion, which greatly reduces the complexity of PGO [Carlone et al, 2014, 2016].

2.6.2.3.3 Machine Learning-Based Strategies

Significant research has also been conducted to apply machine learning algorithms directly to solving the epoch-wise positioning problem. Generally, this approach may be taken to use elements of the previously-estimated system state, and current positioning/orientation observations as inputs to an Artificial Neural Network (ANN) to determine an estimate of the change in system state elements to the current observation epoch [Marques Filho et al, 2010]. In particular, Recurrent Neural Networks (RNN)

are naturally suited to this particular ANN application, since they explicitly include the neural network output as part of the next iteration's input [El-Sheimy et al, 2006; Jwo et al, 2023]. A diagram showing the general structure of an ANN for positioning and navigation is shown in Figure 2.6.9.

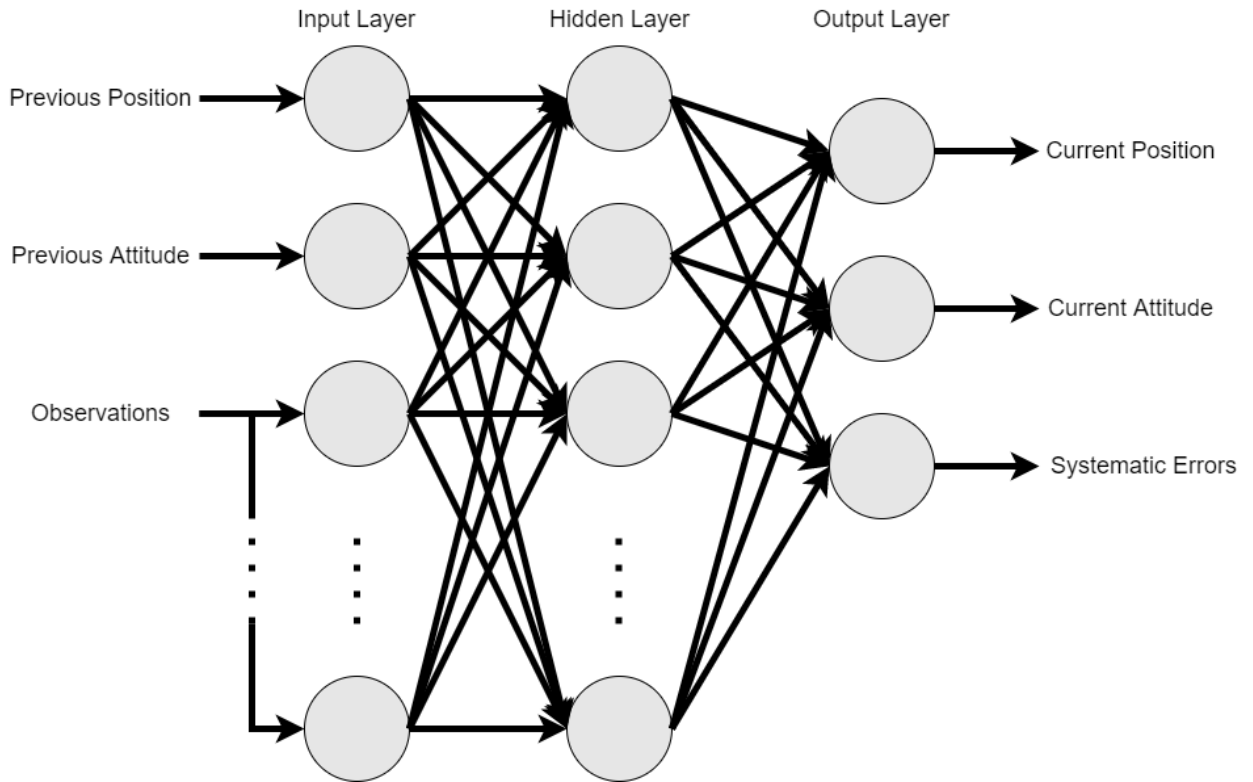


Figure 2.6.9: The general structure of an ANN for positioning and navigation. Note that while there is only one hidden layer in this ANN, multiple hidden layers may be included depending on the implementation [Jwo et al, 2023].

Machine learning-based strategies benefit from not having explicit measurement/system models that can potentially be ill-suited to specific areas of application; however, machine learning approaches to positioning and navigation are only as versatile as their training datasets, so it is essential to have high-quality training data that reflects the expected system dynamics. Additionally, since the models are not explicitly defined outside of the structure of the neural network, statistical analysis of the neural network's output is more challenging.

2.6.3 Generic Multi-Sensor Integration Strategy (GMIS)

The Generic Multi-Sensor Integration Strategy (GMIS) is the integration strategy that is used and further developed in this research. The core of the GMIS was developed gradually through [Wang, 1997, 2008, 2009; Caspary and Wang, 2009] and was formalized as a framework for sensor integration in [Qian et al, 2015; Wang et al, 2015; Qian, 2017].

The GMIS was developed with several main objectives:

- The integration strategy does not treat any sensor fundamentally differently from any other sensor. In other words, the GMIS does not define a core sensor, and does not rely on data being available from any specific sensor to calculate a positioning solution. This also means that IMU data is integrated in the positioning KF as raw measurements, allowing for more thorough characterization and analysis of their stochastic properties.
- The integration strategy defines the system model using kinematic equations, rather than using data from a core positioning sensor to define the motion of the PosAtt system.

Under the GMIS, the system model is defined using kinematic equations describing the motion of the IMU body frame through the navigation frame. The parameters of primary interest in the GMIS are the position vector $\mathbf{r}_{nb}^n$ describing the location of the body frame in the navigation frame, expressed in the navigation frame, and the body frame's orientation expressed in the navigation frame, communicated through the three Euler angles that define the rotation matrix between body frame and navigation frame coordinates. The system state vector in [Qian, 2017] includes: the linear position, velocity, and acceleration vectors of the body frame in the navigation frame; the Euler angles describing the roll, pitch, and heading of the body frame with respect to the navigation frame; the body frame's angular velocity vector, or angular rate, in the navigation frame; and systematic error parameters for each positioning sensor.

The differential equations defining the system model were given in [Qian, 2017] as

$$\dot{\mathbf{r}}_{nb}^n = \mathbf{v}_{nb}^n \quad (2.83)$$

$$\dot{\mathbf{v}}_{nb}^n = \mathbf{a}_{nb}^n \quad (2.84)$$

and

$$\dot{\mathbf{a}}_{nb}^n = \mathbf{0} \quad (2.85)$$

for the linear motion of the system. The differential equations describing the system orientation's change through time were given in [Qian, 2017] as

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \frac{1}{\cos \alpha} \begin{bmatrix} \cos \beta \cos \alpha & 0 & \sin \beta \cos \alpha \\ \sin \beta \sin \alpha & \cos \alpha & -\cos \beta \sin \alpha \\ \sin \beta & 0 & -\cos \beta \end{bmatrix} \boldsymbol{\omega}_{nb}^n \quad (2.86)$$

and

$$\dot{\omega}_{nb}^b = \mathbf{0} \quad (2.87)$$

Practically, the system models in the GMIS are determined by discretizing the differential equations from (2.83) to (2.87).

The measurement models under the GMIS in [Qian, 2017] are defined as follows:

- The accelerometer measurement model is defined by (2.46) in Section 2.4.2.4.1.
- The gyroscope measurement model is defined by (2.47) in Section 2.4.2.4.2.
- The GNSS observables' measurement models are defined by (2.41) and (2.42) in Section 2.4.1.3.

The general workflow of the GMIS is provided in Figure 2.6.10.

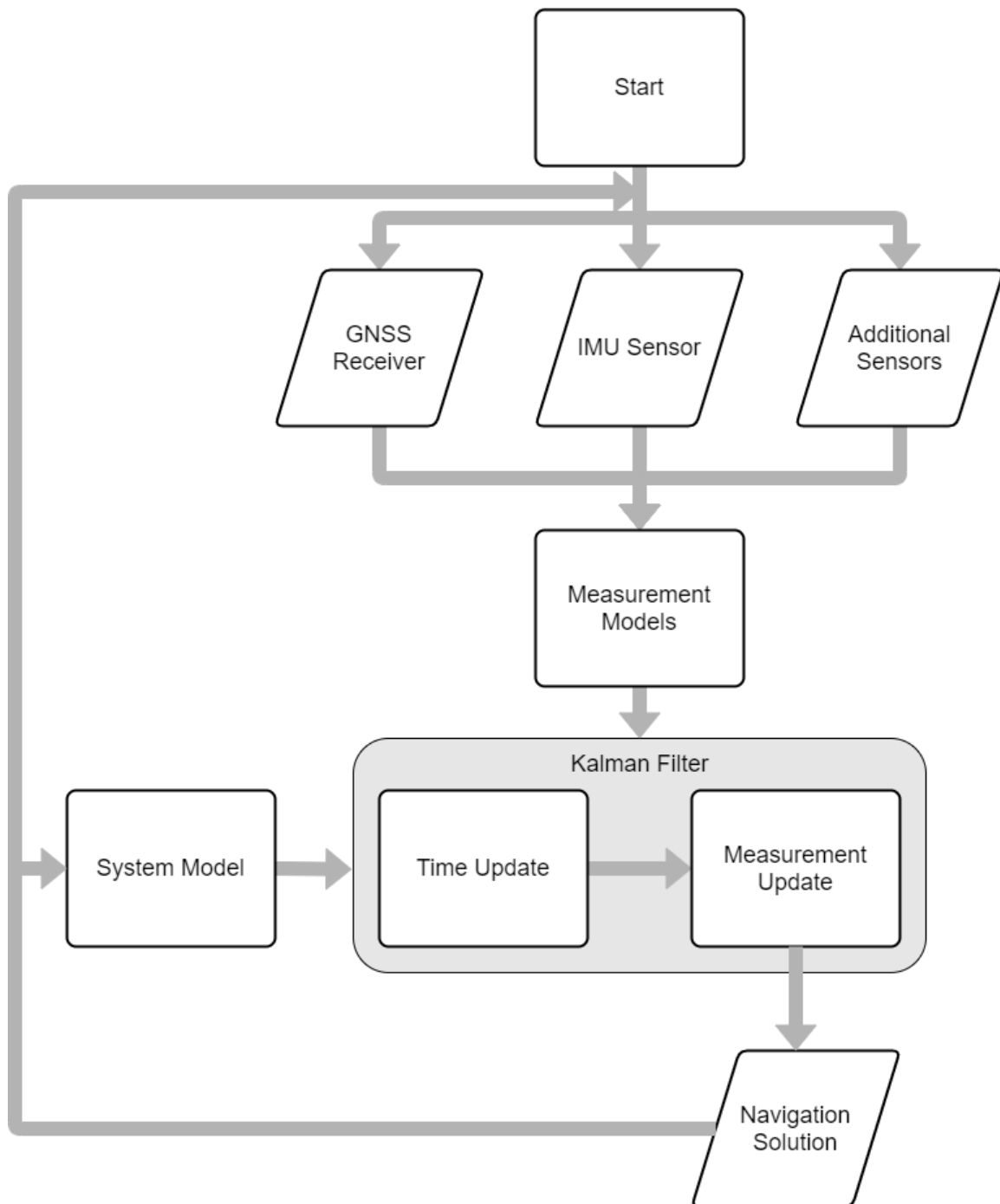


Figure 2.6.10: The workflow of the Generic Multi-Sensor Integration Strategy (GMIS), as described in [Qian, 2017].

2.6.4 Comparison Between the TMIS and the GMIS

Table 2.6.3 provides an overview of the comparative advantages and disadvantages of the traditional sensor integration strategy and the GMIS

Table 2.6.3: Comparison between the traditional sensor integration strategy and the GMIS.

	Advantages	Disadvantages
Traditional Multi-Sensor Integration Strategy	- Computationally efficient - Good system model performance when using a high-quality IMU - Many existing software packages support traditional sensor integration	- System model performance decreases for lower-cost IMU sensors - Relies on specific sensor configurations
Generic Multi-Sensor Integration Strategy	- Mitigates IMU-only solution drift by using kinematic models - Easily adaptable to different sensor configurations - Sensor-level error analysis is enabled due to directly modeling all positioning observations	- High computational load, especially when estimating quantities needed for comprehensive error analysis

3 Formulation of Discrete Kalman Filtering with Constraints – A Generic Framework for Comprehensive Error Analysis

This chapter is based upon the published work:

Wang, J.; Brunson*, B. and Hu, B. (2022): Innovative Formulation in Discrete Kalman Filtering with Constraints – A Generic Framework for Comprehensive Error Analysis, Journal of Global Positioning Systems, vol. 18, issue 1, 56–69. (*corresponding author)

3.1 Preamble

There are many circumstances in which *a priori* information about a dynamic system can be enforced through equality constraints on the system state vector. System state constraints occur frequently in Kalman filtering applications, with several particularly frequently occurring types of constraints:

- The Zero Velocity/Zero Angular Rate Update: When a positioning/navigation system is known to be stationary, the system's linear and angular velocity, as well as its linear acceleration, may be constrained to be equal to zero. Mathematically, when the system's velocity $\mathbf{v}$, acceleration $\mathbf{a}$, and angular velocity $\boldsymbol{\omega}$ are elements of the system state vector, these constraints take the form

$$\mathbf{v} = \mathbf{0}, \mathbf{a} = \mathbf{0}, \boldsymbol{\omega} = \mathbf{0} \quad (3.1)$$

- Path-constrained motion along roadways: When a positioning/navigation system is mounted to a land vehicle moving along a roadway, its position may be constrained to be on a planar surface. Mathematically, for the system's position $\mathbf{r}$, these constraints take the form

$$\mathbf{F}\mathbf{r} = \mathbf{b} \quad (3.2)$$

where $\mathbf{F}$ and $\mathbf{b}$ define the constraining planar surface.

- Constant-velocity motion of tracked targets: When tracking targets that are known to move at a constant velocity, their acceleration may be constrained to be equal to zero. Mathematically, these constraints take the form

$$\mathbf{a} = \mathbf{0} \quad (3.3)$$

- Objects moving in formation: When tracking or positioning vehicles that are moving in a defined formation, the relative positions between them may be geometrically constrained.

Additionally, the attitude modeling in Section 5.5 relies on applying system state constraints to ensure that the attitude parameters for different attitude models and their time derivatives define proper rotations and do not over-parameterize the system's orientation.

While the methods of applying constraints to a Kalman filter described in Section 3.2 are sufficient to solve these problems, they are incapable of estimating all the quantities needed for a comprehensive error analysis. Given this dissertation's focus on error analysis in positioning/navigation, it is necessary to further develop and standardize the system state-constrained discrete Kalman filter to allow for comprehensive error analysis.

The alternative formulation of the standard Kalman filter originally determined by [Wang, 1997] is summarized in Section 2.5.1.1, and the alternative formulation of the system state-constrained Kalman filter developed here builds from this characterization of the Kalman filtering problem as a least-squares problem.

3.2 Existing Methods for Imposing State Equality Constraints

In general, equality constraints on the system state vector may be given as follows

$$\mathbf{H}(\mathbf{x}(k+1)) = \mathbf{h}(k+1) \quad (3.4)$$

where $\mathbf{H}(\mathbf{x})$ is a vector function of the system state equality constraints and $\mathbf{h}(k+1)$ is a vector of constants.

There are a variety of existing methods for imposing constraints on the system state vector in a KF. Most prominent methods for imposing constraints yield similar constrained solutions, but differ in the underlying philosophy of their derivation. Constraints may typically be imposed by either introducing them as “perfect observations” with zero variance, or reprojecting the Kalman gain matrix using the constraint equations (see Section 3.2.1 for documentation of these approaches).

3.2.1 Prominent Methods for Imposing State Equality Constraints

One method of enforcing system state equality constraints is by eliminating them via model reduction. This relies on reparametrizing the problem to remove what are effectively redundant parameters in the system state vector [Simon, 2010].

There are several shortcomings to this approach of handling system state equality constraints:

- Reparametrizing the system state vector does not present a one-size-fits-all solution and must be purpose-built for the application at hand. While this does not affect the quality of the solutions determined via this approach, it does negatively affect its ease-of-use and tractability for some applications.
- If there is significance to the original parametrization of the problem, reparametrizing the problem to remove the state constraints can affect the physical meaning of the individual elements of the system state vector. While solutions for the reparametrized problem may still be estimated, their apparent meaning becomes less obvious. This issue is particularly acute for complex constraint equations, or constraint equations that vary with time.

Given these shortcomings, the Model Reduction method of applying system state constraints is very uncommon in practice [Simon, 2010].

It is worth noting that, when proper care is taken in the method of model reduction, the remaining elements of the system state vector can maintain their physical meaning. This is especially true of geometry-based problems, and compelling arguments are made in favour of this perspective in [Pizzinga, 2010].

Another method of enforcing system state constraints is to model the constraints as “perfect observations”. This approach models the constraints as “perfect,” or zero error, observations [Alouani and Blair, 1991]. This is accomplished by augmenting the observation equations to

$$\mathbf{L} = \begin{bmatrix} \mathbf{z}(k+1) \\ \mathbf{h}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{C}(k+1) \\ \mathbf{H}(k+1) \end{bmatrix} \mathbf{x}(k+1) \quad (3.5)$$

for the constraint equations in (3.4).

The augmented observation vector in (2.86) has an associated covariance matrix of

$$\mathbf{P}_L = \begin{bmatrix} \mathbf{R}(k+1) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^\mu(k+1) \end{bmatrix} \quad (3.6)$$

In (3.6), $\mathbf{R}^\mu(k+1)$ represents the “uncertainty” in the condition equations, and is typically considered to be equal to zero. For cases where $\mathbf{R}^\mu(k+1) \neq \mathbf{0}$, the conditions on the system state are relaxed and may be violated to a certain degree [Alouani and Blair, 1991].

Substituting (3.5) and (3.6) into the standard KF measurement update equation provides an estimate of the constrained system state. Directly using the augmented observation vector can result in numerical instability when $\mathbf{R}^\mu(k+1) = \mathbf{0}$. To counteract this issue, the results of the substitution may be simplified, yielding the following constrained estimate of the system state:

$$\hat{\mathbf{x}}_h(k+1) = (\mathbf{I} - \mathbf{K}_h(k+1))(\mathbf{H}(k+1)\hat{\mathbf{x}}(k+1) - \mathbf{h}(k+1)) \quad (3.7)$$

where $\mathbf{K}_h(k+1)$ is given by

$$\mathbf{K}_h(k+1) = \mathbf{P}(k+1)\mathbf{H}^T(k+1) \left(\mathbf{H}(k+1)\mathbf{P}(k+1)\mathbf{H}^T(k+1) + \mathbf{R}^\mu(k+1) \right)^{-1} \quad (3.8)$$

[Alouani and Blair, 1991].

System state constraints may also be enforced via reprojection of the system state vector. This method takes the unconstrained solution to the KF and reprojects the system state estimate onto the constraint surface [Simon and Chia, 2002]. This results in the following optimization problem to determine the best constrained estimate of the system state vector $\hat{\mathbf{x}}_h$:

$$\min_{\hat{\mathbf{x}}_h} (\hat{\mathbf{x}}_h(k+1) - \hat{\mathbf{x}}(k+1))^T \mathbf{W} (\hat{\mathbf{x}}_h(k+1) - \hat{\mathbf{x}}(k+1)) \quad (3.9)$$

such that

$$\mathbf{H}(k+1)\hat{\mathbf{x}}_h(k+1) = \mathbf{h}(k+1) \quad (3.10)$$

where $\mathbf{W}$ is a weighting matrix for the reprojection [Simon and Chia, 2002].

The solution to this optimization problem is simply

$$\begin{aligned} \hat{\mathbf{x}}_h(k+1) &= \hat{\mathbf{x}}(k+1) \\ &- \mathbf{W}^{-1}\mathbf{H}^T(k+1) \left(\mathbf{H}(k+1)\mathbf{W}^{-1}\mathbf{H}^T(k+1) \right)^{-1} (\mathbf{H}(k+1)\hat{\mathbf{x}}(k+1) - \mathbf{h}(k+1)) \end{aligned} \quad (3.11)$$

[Simon and Chia, 2002].

With this method, (3.11) may take on different values of $\mathbf{W}$ to represent different weighting schemes for the reprojection of the system state vector, but there are two values of $\mathbf{W}$ that represent special cases:

1. When $\mathbf{W} = \mathbf{P}^{-1}(k+1)$, (3.11) provides an estimate for the constrained system state estimate that is identical to its Maximum Likelihood Estimate (MLE) [Simon and Chia, 2002].

2. When $\mathbf{W} = \mathbf{I}$, (3.11) provides an estimate for the constrained system state estimate that is identical to its Mean Square minimization estimate [Simon and Chia, 2002].

In a similar vein, system state constraints may be reinforced by reprojecting the Kalman Gain matrix. The Kalman Gain Matrix may be derived for a standard KF by solving the optimization problem

$$\min_{\mathbf{K}} \text{tr}[(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}(k+1))\mathbf{P}(k+1)(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}(k+1))^T + \mathbf{K}(k+1)\mathbf{R}(k+1)\mathbf{K}(k+1)] \quad (3.12)$$

[Simon, 2010].

If (3.12) is solved subject to the constraint equations in (3.4), the reprojected Kalman Gain matrix may be calculated and substituted into the KF measurement update equation in (2.52) to give the following estimate of the constrained system state vector:

$$\hat{\mathbf{x}}_h(k+1) = \hat{\mathbf{x}}(k+1) - \mathbf{H}^T(k+1)(\mathbf{H}(k+1)\mathbf{H}^T(k+1))^{-1}(\mathbf{H}(k+1)\hat{\mathbf{x}}(k+1) - \mathbf{h}(k+1)) \quad (3.13)$$

[Simon, 2010].

3.2.2 General Remarks and Alternative Approaches

Despite the variety in the methods used to derive the system state constrained KF equations detailed in the preceding sections, the resulting equations are very similar to one another.

When the “uncertainty” weighting $\mathbf{R}^\mu(k+1)$ for the constraint equations under the “perfect observations” approach is defined as a zero matrix (i.e. the constraint equations are considered strict, with no errors), the resulting estimate for the constrained system state vector is identical to that of the reprojected system state vector under the weight definition $\mathbf{W} = \mathbf{P}^{-1}(k+1)$.

The estimate of the constrained system state vector under the Kalman Gain Reprojection approach is identical to that of the reprojected system state vector under the weight definition $\mathbf{W} = \mathbf{I}$.

The results of the Model Reduction approach are less obvious in their relationship to the results of the other approaches, but can be intuitively understood by how it affects the KF solution. Under the Model Reduction approach, some elements of the system state vector are replaced by linear combinations of other system state vector elements, thereby occupying a vector subspace corresponding to the coefficient matrix of the constraint equations (i.e. the “constraint surface”)

[Pizzinga, 2010]. This model reduction can therefore be intuitively understood as reprojecting the system state vector onto the constraint surface, which is exactly what is being done under the System State Vector Reprojection approach. Therefore, depending upon the definition of the weighting matrix in the System State Vector Reprojection approach, and the covariance matrix of the system state vector for the reduced model, these methods may intuitively be considered equivalent to one another.

The approaches to applying state constraints discussed thus far are very consistent in terms of their results and represent the status quo for applying state constraints in KF problems. There are other alternative methods to applying state constraints that do not provide this level of consistency in their results, but which are also noteworthy in their strategies for handling state constraints.

One such method is referred to as System Projection, where it is assumed that the noise-free KF system should also obey the constraints on the system state vector (i.e. the predicted system state vector should conform to the system state constraints) [Ko and Bitmead, 2007]. This effectively enforces the system state constraints on the process noise, rather than on the measurement noise as is done by the other approaches [Simon, 2010].

Another method proposed in [Ghanbarpourasl and Zobar, 2022] is based on separating the system state into groups of variables using singular value decomposition. This separation results in a set of deterministic, or fully constrained, variables and a stochastic component.

There are several methods to handling system state constraints when the constraint equations are non-linear:

- i. A Taylor Series expansion of the constraint equations may be used to linearize the constraint equations, in a similar way to how the KF system and measurement models are treated in the EKF [Simon, 2006]. Higher-order terms in the Taylor series expansion may be used to improve the approximation to the non-linear constraint equations [Yang and Blasch, 2009].
- ii. Under the Smoothly Constrained Kalman Filter (SCKF), non-linear constraints are managed by linearizing them, and then iteratively enforcing them under the Perfect Observation Method with non-zero R^{μ} values that gradually get stricter as the solution goes through more iterations [De Geeter et al, 1997].

3.3 Alternative Derivation of Discrete Kalman Filtering with Constraints

This section details this research's alternative derivation of Discrete Kalman Filtering with state-equality constraints after the least-squares principle.

3.3.1 Formulation of the Problem

Similarly to Section 2.5.1.1, the alternative formulation of the discrete Kalman filter with system-state equality constraints uses three sets of pseudo-observations, each corresponding to one of the mutually independent sources of random error in a Kalman filter. These three pseudo-observation vectors are defined below. As in Section 2.5.1.1, specification of the epoch is omitted whenever it is unambiguous (e.g. $A(k + 1, k)$ is instead written as A).

- l_x , which corresponds to the predicted system state vector and is modeled as

$$l_x + Bw + v_x = Ax(k) \quad (3.14)$$

with the associated covariance matrix of

$$D_{l_x} = AP(k)A^T \quad (3.15)$$

- l_w , which corresponds to the process noise vector and is modeled as

$$l_w + v_w = w(k) \quad (3.16)$$

with the associated covariance matrix of

$$D_{l_w} = Q \quad (3.17)$$

- l_z , which corresponds to the observation vector and is modeled as

$$l_z + v_z = z + v_z = Cx(k + 1) \quad (3.18)$$

with the associated covariance matrix of

$$D_{l_z} = R \quad (3.19)$$

Here, the vectors l_x , l_w , and l_z are assumed to be statistically independent from one another.

Combining these three pseudo-observation vectors into a single, unified observation vector yields the overdetermined system of equations

$$\mathbf{l} + \mathbf{v} = \begin{bmatrix} \mathbf{l}_x \\ \mathbf{l}_w \\ \mathbf{l}_z \end{bmatrix} + \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_w \\ \mathbf{v}_z \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{w}(k+1) \end{bmatrix} \quad (3.20)$$

The constraints on the Kalman filter's system state vector may be expressed as

$$\mathbf{H}\mathbf{x}_h(k+1) - \mathbf{h} = \mathbf{0} \quad (3.21)$$

where $\mathbf{H}$ is a coefficient matrix describing the linear or linearized system state constraint equations. The dimensions of $\mathbf{H}$ is $n \times h$, where n is the number of elements in the system state vector and h is the number of constraint equations. It is worth noting that the rank of $\mathbf{H}$ should be equal to the number of constraint equations, ensuring that there are no redundant constraint equations. $\mathbf{h}$ is a constant vector specific to the constraint equations under consideration.

Under this formulation, the problem of discrete Kalman filtering with system state equality constraints becomes the solution of the overdetermined system of equations in (3.20) such that the equality constraints in (3.21) are satisfied. This optimization problem may be solved using the least-squares principle by minimizing the cost function

$$\Phi = \mathbf{v}^T \mathbf{W} \mathbf{v} + 2\mathbf{k}_h^T (\mathbf{H}\mathbf{x}_h - \mathbf{h}) \rightarrow \min \quad (3.22)$$

where $\mathbf{W}$ is a weighting matrix for the combined observation vector, defined in this case to be the inverse of its covariance matrix, $\mathbf{k}$ is a vector of Lagrange multipliers, and the $(k+1)$ is omitted after $\mathbf{x}$ for clarity of presentation.

Expanding the cost function in (3.22) using the definition of $\mathbf{v}$ and $\mathbf{W}$ yields the cost function

$$\Phi = [\mathbf{v}_x^T \quad \mathbf{v}_w^T \quad \mathbf{v}_z^T] \begin{bmatrix} \mathbf{D}_{l_x}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_w \\ \mathbf{v}_z \end{bmatrix} + 2\mathbf{k}_h^T (\mathbf{H}\mathbf{x} - \mathbf{h}) \rightarrow \min \quad (3.23)$$

which simplifies to

$$\Phi = \mathbf{v}_x^T \mathbf{D}_{l_x}^{-1} \mathbf{v}_x + \mathbf{v}_w^T \mathbf{Q}^{-1} \mathbf{v}_w + \mathbf{v}_z^T \mathbf{R}^{-1} \mathbf{v}_z + 2\mathbf{k}_h^T (\mathbf{H}\mathbf{x}_h - \mathbf{h}) \rightarrow \min \quad (3.24)$$

Minimizing the cost function in (3.24) requires taking its partial derivatives with respect to the quantities that can vary: $\mathbf{x}$, $\mathbf{w}$, and $\mathbf{k}$. At a local minima in the cost function, these partial derivatives should all be equal to zero. This leads to the following three matrix equations:

$$\frac{\partial \Phi}{\partial \mathbf{x}} = \frac{\partial \Phi}{\partial \mathbf{v}_x} \frac{\partial \mathbf{v}_x}{\partial \mathbf{x}} + \frac{\partial \Phi}{\partial \mathbf{v}_z} \frac{\partial \mathbf{v}_z}{\partial \mathbf{x}} + 2\mathbf{k}_h^T \mathbf{H}$$

$$\frac{\partial \Phi}{\partial \mathbf{x}} = 2\mathbf{v}_x^T \mathbf{D}_{l_x}^{-1} + 2\mathbf{v}_z^T \mathbf{R}^{-1} \mathbf{C} + 2\mathbf{k}_h^T \mathbf{H} = \mathbf{0}$$

$$\mathbf{D}_{l_x}^{-1} \mathbf{v}_x + \mathbf{C}^T \mathbf{R}^{-1} \mathbf{v}_z + \mathbf{H}^T \mathbf{k}_h = \mathbf{0}$$

$$(\mathbf{D}_{l_x}^{-1} + \mathbf{C}^T \mathbf{R}^{-1} \mathbf{C}) \mathbf{x}_h - \mathbf{D}_{l_x}^{-1} \mathbf{B} \mathbf{w}_h + \mathbf{H}^T \mathbf{k}_h - \mathbf{D}_{l_x}^{-1} \mathbf{l}_x - \mathbf{C}^T \mathbf{R}^{-1} \mathbf{l}_z = \mathbf{0} \quad (3.25)$$

$$\frac{\partial \Phi}{\partial \mathbf{w}} = \frac{\partial \Phi}{\partial \mathbf{v}_x} \frac{\partial \mathbf{v}_x}{\partial \mathbf{w}} + \frac{\partial \Phi}{\partial \mathbf{v}_w} \frac{\partial \mathbf{v}_w}{\partial \mathbf{w}}$$

$$\frac{\partial \Phi}{\partial \mathbf{w}} = -2\mathbf{v}_x^T \mathbf{D}_{l_x}^{-1} \mathbf{B} + 2\mathbf{v}_w^T \mathbf{Q}^{-1} = \mathbf{0}$$

$$-\mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{v}_x + \mathbf{Q}^{-1} \mathbf{v}_w = \mathbf{0}$$

$$-\mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{x}_h + (\mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{B} + \mathbf{Q}^{-1}) \mathbf{w}_h + \mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{l}_x - \mathbf{Q}^{-1} \mathbf{l}_w = \mathbf{0} \quad (3.26)$$

$$\frac{\partial \Phi}{\partial \mathbf{k}} = \mathbf{H} \mathbf{x}_h - \mathbf{h} = \mathbf{0} \quad (3.27)$$

Equations (3.25) to (3.27) define a system of equations that, when solved, provide a solution to the system state-constrained Kalman filter. This system of equations may be organized as follows

$$\begin{bmatrix} \mathbf{D}_{l_x}^{-1} + \mathbf{C}^T \mathbf{R}^{-1} \mathbf{C} & -\mathbf{D}_{l_x}^{-1} \mathbf{B} & \mathbf{H}^T \\ -\mathbf{B}^T \mathbf{D}_{l_x}^{-1} & \mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{B} + \mathbf{Q}^{-1} & \mathbf{0} \\ \mathbf{H} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_h \\ \hat{\mathbf{w}}_h \\ \hat{\mathbf{k}}_h \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{l_x}^{-1} \mathbf{l}_x + \mathbf{C}^T \mathbf{R}^{-1} \mathbf{l}_z \\ -\mathbf{B}^T \mathbf{D}_{l_x}^{-1} \mathbf{l}_x + \mathbf{Q}^{-1} \mathbf{l}_w \\ \mathbf{h} \end{bmatrix} \quad (3.28)$$

The system of equations in (3.28) is called the system of normal equations and is not overdetermined. The system of normal equations may be solved explicitly for $\hat{\mathbf{x}}$, $\hat{\mathbf{w}}$, and $\hat{\mathbf{k}}$ as long as the problem is not ill-formed. Solving the system of normal equations directly, however, is not desirable due to the size of the coefficient matrix, and the fact that it contains many null elements and has a very well-defined symmetric structure. It is therefore useful to simplify (3.28) to isolate the solutions for $\hat{\mathbf{x}}$, $\hat{\mathbf{w}}$, and $\hat{\mathbf{k}}$, which is accomplished via row-reduction of the 3 x 3 partitioned block matrix in (3.28).

Section 3.3.2 conducts the row reduction of the system of normal equations to determine the constrained least-squares estimates of the system state and process noise vectors. Section 3.3.3 relates the constrained estimates of the system state and process noise vectors to their unconstrained estimates. Section 3.3.4 derives the covariance matrices for the system state and process noise vectors. Section 3.3.5 derives the constrained estimates of the predicted system state, process noise, and observation residuals, alongside their associated covariance matrices. Section 3.3.6 describes the process of adjusting the solutions derived in this chapter for an EKF. Section 3.4 provides an overview of

both the unconstrained and constrained Kalman Filter problems, as well as a description of the overall process for applying system state constraints to a Kalman Filter. Sections 3.3.2 to 3.3.6 provide many of the intermediate steps of the derivation, and so Section 3.4 is a good entry point if the reader wishes to simply see the final equations and process.

3.3.2 Row-Reduction of the System of Normal Equations

This section provides the intermediate steps that are part of the alternative derivation of the system state-constrained discrete Kalman filter. To skip the intermediate steps of the derivation, see Section 3.4 for the final formulae and description of the overall process for applying system state constraints to a Kalman Filter.

The system of normal equations in (3.28) may be reduced to a 2 x 2 block matrix equation via the row reduction method. With this row reduction, the system of equations becomes

$$\begin{aligned} & \left(\begin{bmatrix} B^T D_{l_x}^{-1} B + Q^{-1} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} -B^T D_{l_x}^{-1} \\ H \end{bmatrix} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} \begin{bmatrix} -D_{l_x}^{-1} B & H^T \end{bmatrix} \right) \begin{bmatrix} \hat{w}_h \\ \hat{k}_h \end{bmatrix} \\ &= \begin{bmatrix} -B^T D_{l_x}^{-1} l_x + Q^{-1} l_w \\ h \end{bmatrix} - \begin{bmatrix} -B^T D_{l_x}^{-1} \\ H \end{bmatrix} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} (D_{l_x}^{-1} l_x + C^T R^{-1} l_z) \end{aligned}$$

and following through on the block matrix arithmetic results in

$$\begin{aligned} & \begin{bmatrix} B^T D_{l_x}^{-1} B + Q^{-1} - B^T D_{l_x}^{-1} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} D_{l_x}^{-1} B & B^T D_{l_x}^{-1} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} H^T \\ H (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} D_{l_x}^{-1} B & -H (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} H^T \end{bmatrix} \begin{bmatrix} \hat{w}_h \\ \hat{k}_h \end{bmatrix} \\ &= \begin{bmatrix} (-B^T + B^T D_{l_x}^{-1} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1}) D_{l_x}^{-1} l_x + Q^{-1} l_w + B^T D_{l_x}^{-1} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} C^T R^{-1} l_z \\ h - H (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} (D_{l_x}^{-1} l_x + C^T R^{-1} l_z) \end{bmatrix} \quad (3.29) \end{aligned}$$

The system of equations in (3.29) is still very unwieldy, in part due to the term $(D_{l_x}^{-1} + C^T R^{-1} C)^{-1}$. Fortunately, this term may be simplified to

$$\begin{aligned} (D_{l_x}^{-1} + C^T R^{-1} C)^{-1} &= D_{l_x} - D_{l_x} C^T (R + C D_{l_x} C^T)^{-1} C D_{l_x} \\ &= D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x} \end{aligned} \quad (3.30)$$

where D_d is the covariance matrix of the system innovation vector.

The relation in (3.30) may be substituted into (3.29), yielding

$$\begin{aligned}
& \begin{bmatrix} B^T D_{l_x}^{-1} B + Q^{-1} - B^T D_{l_x}^{-1} (D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) D_{l_x}^{-1} B & B^T D_{l_x}^{-1} (D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \\ H(D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) D_{l_x}^{-1} B & -H(D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \end{bmatrix} \begin{bmatrix} \hat{w}_h \\ \hat{k}_h \end{bmatrix} \\
&= \begin{bmatrix} (-B^T + B^T D_{l_x}^{-1} (D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x})) D_{l_x}^{-1} l_w + Q^{-1} l_w + B^T D_{l_x}^{-1} (D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z \\ h - H(D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) (D_{l_x}^{-1} l_w + C^T R^{-1} l_z) \end{bmatrix}
\end{aligned}$$

which simplifies to

$$\begin{aligned}
& \begin{bmatrix} B^T D_{l_x}^{-1} B - B^T D_{l_x}^{-1} B + Q^{-1} + B^T C^T D_d^{-1} C B & B^T H^T - B^T C^T D_d^{-1} C D_{l_x} H^T \\ HB - HD_{l_x} C^T D_d^{-1} C B & -H(D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \end{bmatrix} \begin{bmatrix} \hat{w}_h \\ \hat{k}_h \end{bmatrix} \\
&= \begin{bmatrix} (-B^T D_{l_x}^{-1} + B^T D_{l_x}^{-1} - B^T C^T D_d^{-1} C) l_w + Q^{-1} l_w + B^T (I - C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z \\ h - H l_x + HD_{l_x} C^T D_d^{-1} C D_{l_x} C^T R^{-1} l_z \end{bmatrix}
\end{aligned}$$

which further simplifies to

$$\begin{aligned}
& \begin{bmatrix} B^T C^T D_d^{-1} C B + Q^{-1} & B^T H^T - B^T C^T D_d^{-1} C D_{l_x} H^T \\ HB - HD_{l_x} C^T D_d^{-1} C B & -H(D_{l_x} - D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \end{bmatrix} \begin{bmatrix} \hat{w}_h \\ \hat{k}_h \end{bmatrix} \\
&= \begin{bmatrix} -B^T C^T D_d^{-1} C l_x + Q^{-1} l_w + B^T (I - C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z \\ h - H l_x + HD_{l_x} C^T D_d^{-1} C D_{l_x} C^T R^{-1} l_z \end{bmatrix} \quad (3.31)
\end{aligned}$$

The system of equations in (3.31) may be reduced to a simple matrix equation by again applying the row reduction method. Doing this results in the following system of equations:

$$\begin{aligned}
& (-H(D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \\
& - (HB - HD_{l_x} C^T D_d^{-1} C B) (B^T C^T D_d^{-1} C B + Q^{-1})^{-1} (B^T H^T - B^T C^T D_d^{-1} C D_{l_x} H^T) \hat{k}_h \\
& = h - H l_x + HD_{l_x} C^T D_d^{-1} C D_{l_x} C^T R^{-1} l_z \\
& - (HB - HD_{l_x} C^T D_d^{-1} C B) (B^T C^T D_d^{-1} C B + Q^{-1})^{-1} (-B^T C^T D_d^{-1} C l_x + Q^{-1} l_w \\
& + B^T (I - C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z)
\end{aligned}$$

which may be simplified by recognizing that $(Q^{-1} + B^T C^T D_d^{-1} C B)^{-1} = Q - QB^T C^T (D_d + CBQB^T C^T)^{-1} CBQ$, and becomes

$$\begin{aligned}
& \left(-H(D_{l_x} C^T D_d^{-1} C D_{l_x}) H^T \right. \\
& \quad - (HB - HD_{l_x} C^T D_d^{-1} CB) \left(Q - QB^T C^T (D_d + QBQB^T C^T)^{-1} CBQ \right) (B^T H^T \\
& \quad \left. - B^T C^T D_d^{-1} C D_{l_x} H^T) \right) \hat{k}_h \\
& = h - Hl_x + HD_{l_x} C^T D_d^{-1} C D_{l_x} C^T R^{-1} l_z \\
& - (HB - HD_{l_x} C^T D_d^{-1} CB) \left(Q - QB^T C^T (D_d + QBQB^T C^T)^{-1} CBQ \right) (-B^T C^T D_d^{-1} Cl_x + Q^{-1} l_w \\
& \quad + B^T (I - C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z) \\
& \hat{k}_h = (HP(k+1)H^T)^{-1} \{ HP(k+1) [D_{l_x}^{-1} l_x + C^T R^{-1} l_z] - h \} \tag{3.32}
\end{aligned}$$

The solution for the Lagrange multipliers in (3.32) may then be substituted into (3.31) to determine constrained estimates of the process noise vector, as follows:

$$\begin{aligned}
& (B^T C^T D_d^{-1} CB + Q^{-1}) \hat{w}_h \\
& + (B^T H^T - B^T C^T D_d^{-1} C D_{l_x} H^T) (HP(k+1)H^T)^{-1} \{ HP(k+1) [D_{l_x} l_x + C^T R^{-1} l_z] - h \} \\
& = -B^T C^T D_d^{-1} Cl_x + Q^{-1} l_w + B^T (I - C^T D_d^{-1} C D_{l_x}) C^T R^{-1} l_z
\end{aligned}$$

which simplifies to

$$\hat{w}_h = l_w + QB^T P^{-1}(k+1, k) Kd - QB^T P^{-1}(k+1, k) P(k+1) H^T (HP(k+1)H^T)^{-1} (H\hat{x} - h) \tag{3.33}$$

Finally, the constrained estimate of the system state vector may be determined by substituting (3.32) and (3.33) into the original system of normal equations in (3.28) as follows:

$$\begin{aligned}
& (D_{l_x}^{-1} + C^T R^{-1} C) \hat{x}_h - D_{l_x}^{-1} B \hat{w}_h + H^T \hat{k}_h = D_{l_x}^{-1} l_x + C^T R^{-1} l_z \\
& (D_{l_x}^{-1} + C^T R^{-1} C) \hat{x}_h \\
& - D_{l_x}^{-1} B \left(l_w + QB^T P^{-1}(k+1, k) Kd - QB^T P^{-1}(k+1, k) P(k+1) H^T (HP(k+1)H^T)^{-1} (H\hat{x} - h) \right) \\
& + H^T (HP(k+1)H^T)^{-1} \{ HP(k+1) [D_{l_x}^{-1} l_x + C^T R^{-1} l_z] - h \} \\
& = D_{l_x}^{-1} l_x + C^T R^{-1} l_z
\end{aligned}$$

which simplifies to

$$\hat{x}_h = \hat{x}(k+1, k) + Kd - P(k+1) H^T (HP(k+1)H^T)^{-1} (H\hat{x} - h) \tag{3.34}$$

3.3.3 Relating the Constrained Solution to the Unconstrained Solution

The constrained estimates of the system state vector and the process noise vector may be expressed relative to their unconstrained estimates given in Section 2.5.1. This enables a two-step solution expression and is more practical in calculation. Recognizing that $\hat{\mathbf{x}} = \hat{\mathbf{x}}(k+1, k) + \mathbf{K}\mathbf{d}$ in (3.34) allows for the equation to be equivalently expressed as

$$\hat{\mathbf{x}}_h = \hat{\mathbf{x}} - \mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}} - \mathbf{h}) \quad (3.35)$$

Similarly, recognizing that $\hat{\mathbf{w}} = \mathbf{l}_w + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{K}\mathbf{d}$ in (3.2.20) allows for the equation to be equivalently expressed as

$$\hat{\mathbf{w}}_h = \hat{\mathbf{w}} - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}} - \mathbf{h}) \quad (3.36)$$

Interestingly, (3.35) and (3.36) make it clear that this research's approach to applying system state equality constraints to a discrete-time Kalman Filter is functionally reprojecting the unconstrained solution onto a constraint surface using the inverse of the unconstrained solution's covariance matrix as the weight matrix for the reprojection.

Given the expression of the constrained system state estimate in (3.35), a direct comparison between this research's approach to applying system state equality constraints and other popular approaches may be made:

- The “Perfect Observations” Approach:
 - Comparing the estimates of $\hat{\mathbf{x}}_h$ in (3.35) and (3.7), it is clear that the equations are nearly identical. It is important to note that $\mathbf{R}^\mu$ in (3.7) represents a relaxation of the constraint equations, allowing them to not exactly be satisfied by $\hat{\mathbf{x}}_h$. For the special case where $\mathbf{R}^\mu := \mathbf{0}$, there is no relaxation of the constraint equations and they must be perfectly satisfied by $\hat{\mathbf{x}}_h$, and under this special case, the “perfect observations” approach and this research's approach gives identical results.
- The System State Reprojection Approach:
 - Comparing the estimates of $\hat{\mathbf{x}}_h$ in (3.35) and (3.11), it is clear that the estimates are identical for the special case where the system state reprojecting weighting matrix is defined to be $\mathbf{W} := \mathbf{P}^{-1}(k+1)$. This equivalency also allows for several inferences to be made about the properties of this research's approach to applying system state

constraints, chief among which is that the constrained system state estimate derived through this research constitutes a Maximum Likelihood Estimate (MLE).

- The Kalman Gain Reprojection Approach:
 - Comparing the estimates of $\hat{\mathbf{x}}_h$ in (3.35) and (3.13), it is clear that the estimates of the constrained system state are similar but not identical. Under the special case where $\mathbf{P}(k+1) = \mathbf{I}$, these estimates would be identical to one another, but this situation is unlikely to arise in practice.

It is encouraging that the approach taken to apply system state constraints in this research provides equivalent system state estimates to the most widely-used approaches to applying system state constraints. It is worth noting that this research's approach additionally allows for the calculation of quantities that are otherwise not estimable when using other methods to apply system state constraint equations. For example, the estimate of the constrained process noise vector is unique to this research. Additionally, given the formulation of the constrained Kalman Filter problem using the three sets of pseudo-observations, this research enables the calculation of the process noise residuals, residuals of the predicted system state vector, and observation residuals, along with their associated covariance matrices. The derivation of these residual vectors, along with their covariance matrices, is described in Section 3.3.5.

3.3.4 The Constrained System State and Process Noise Covariance Matrices

The covariance matrix for the constrained system state estimate may be readily derived by applying Covariance Law to (3.34), as follows:

$$\mathbf{P}_h = \begin{bmatrix} \mathbf{I} + \mathbf{K}\mathbf{C} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}(\mathbf{H} + \mathbf{H}\mathbf{K}\mathbf{C}) & \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K} - \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{P}(k+1, k) & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}.$$

$$\cdot \begin{bmatrix} \mathbf{I} + \mathbf{C}^T\mathbf{K}^T - (\mathbf{H}^T + \mathbf{C}^T\mathbf{K}^T\mathbf{H}^T)(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P} \\ \mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P} - \mathbf{K}^T \end{bmatrix}$$

$$\mathbf{P}_h = \mathbf{P}(k+1, k) + \mathbf{P}(k+1, k)\mathbf{C}^T\mathbf{K}^T - \mathbf{P}(k+1, k)(\mathbf{H}^T + \mathbf{C}^T\mathbf{K}^T\mathbf{H}^T)(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}$$

$$+ \mathbf{K}\mathbf{C}\mathbf{P}(k+1, k) + \mathbf{K}\mathbf{C}\mathbf{P}(k+1, k)\mathbf{C}^T\mathbf{K}^T - \mathbf{K}\mathbf{C}\mathbf{P}(k+1, k)(\mathbf{H}^T + \mathbf{C}^T\mathbf{K}^T\mathbf{H}^T)(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}$$

$$- \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}(\mathbf{H} + \mathbf{H}\mathbf{K}\mathbf{C})\mathbf{P}(k+1, k) - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}(\mathbf{H} + \mathbf{H}\mathbf{K}\mathbf{C})\mathbf{P}(k+1, k)\mathbf{C}^T\mathbf{K}^T$$

$$+ \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}(\mathbf{H} + \mathbf{H}\mathbf{K}\mathbf{C})\mathbf{P}(k+1, k)(\mathbf{H}^T + \mathbf{C}^T\mathbf{K}^T\mathbf{H}^T)(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}$$

$$\begin{aligned}
& +PH^T(HPH^T)^{-1}HKRK^TH^T(HPH^T)^{-1}HP - PH^T(HPH^T)^{-1}HKRK^T \\
& -KRK^TH^T(HPH^T)^{-1}HP + KRK^T
\end{aligned}$$

This equation eventually simplifies to

$$P_h = P(k+1, k) - P(k+1, k)C^TD_d^{-1}CP(k+1, k) - PH(HPH^T)^{-1}H^TP \quad (3.37)$$

or equivalently,

$$P_h = P - PH(HPH^T)^{-1}H^TP \quad (3.38)$$

The covariance matrix for the constrained process noise estimate may be similarly derived as follows:

$$\begin{aligned}
D_{w_h} &= \begin{bmatrix} I & -QB^TP^{-1}(k+1, k)PH^T(HPH^T)^{-1}H & QB^TP^{-1}(k+1, k)(K - PH^T(HPH^T)^{-1}HK) \end{bmatrix} \cdot \\
& \cdot \begin{bmatrix} Q & QB^T & QB^TC^T \\ BQ & P(k+1, k) & P(k+1, k)C^T \\ CBQ & CP(k+1, k) & D_d \end{bmatrix} \begin{bmatrix} I \\ -H^T(HPH^T)^{-1}HPP^{-1}(k+1, k)BQ \\ (K^T - K^TH^T(HPH^T)^{-1}HP)P^{-1}(k+1, k)BQ \end{bmatrix} \\
D_{w_h} &= Q + QB^TP^{-1}(k+1, k)PH^T(HPH^T)^{-1}HBQ \\
& + QB^TP^{-1}(k+1, k)(K - PH^T(HPH^T)^{-1}HK)CBQ - QB^TH^T(HPH^T)^{-1}HPP^{-1}(k+1, k)BQ \\
& + QB^TP^{-1}(k+1, k)PH^T(HPH^T)^{-1}HP(k+1, k)H^T(HPH^T)^{-1}HPP^{-1}(k+1, k)BQ \\
& - QB^TP^{-1}(k+1, k)(K - PH^T(HPH^T)^{-1}HK)CP(k+1, k)H^T(HPH^T)^{-1}HPP^{-1}(k+1, k)BQ \\
& + QB^TC^T(K^T - K^TH^T(HPH^T)^{-1}HP)P^{-1}(k+1, k)BQ \\
& - QB^TP^{-1}(k+1, k)PH^T(HPH^T)^{-1}HP(k+1, k)C^T(K^T - K^TH^T(HPH^T)^{-1}HP)P^{-1}(k+1, k)BQ \\
& + QB^TP^{-1}(k+1, k)(K - PH^T(HPH^T)^{-1}HK)D_d(K^T - K^TH^T(HPH^T)^{-1}HP)P^{-1}(k+1, k)BQ
\end{aligned}$$

This equation eventually simplifies to

$$D_{w_h} = Q - QB^TC^TD_d^{-1}CBQ - QB^T(I - C^TK^T)H^T(HPH^T)^{-1}H(I - KC)BQ \quad (3.39)$$

or equivalently,

$$\mathbf{D}_{w_h} = \mathbf{D}_w - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \quad (3.40)$$

For the sake of completeness, the covariance matrix between the constrained estimates of the system state vector and process noise vector may be determined similarly:

$$\begin{aligned} \mathbf{D}_{x_h w_h} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H} & (\mathbf{K} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K}) \end{bmatrix} \cdot \\ &\cdot \begin{bmatrix} \mathbf{Q} & \mathbf{Q}\mathbf{B}^T & \mathbf{Q}\mathbf{B}^T\mathbf{C}^T \\ \mathbf{B}\mathbf{Q} & \mathbf{P}(k+1, k) & \mathbf{P}(k+1, k)\mathbf{C}^T \\ \mathbf{C}\mathbf{B}\mathbf{Q} & \mathbf{C}\mathbf{P}(k+1, k) & \mathbf{D}_d \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ (\mathbf{K}^T - \mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P})\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \end{bmatrix} \\ \mathbf{D}_{x_h w_h} &= (\mathbf{I} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H})\mathbf{B}\mathbf{Q} + (\mathbf{K} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K})\mathbf{C}\mathbf{B}\mathbf{Q} \\ &- (\mathbf{I} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H})\mathbf{P}(k+1, k)\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &- (\mathbf{K} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K})\mathbf{C}\mathbf{P}(k+1, k)\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &+ (\mathbf{I} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H})\mathbf{P}(k+1, k)\mathbf{C}^T(\mathbf{K}^T - \mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P})\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &+ (\mathbf{K} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K})\mathbf{D}_d(\mathbf{K}^T - \mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P})\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \end{aligned}$$

This equation eventually simplifies to

$$\mathbf{D}_{x_h w_h} = (\mathbf{I} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H})(\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{B}\mathbf{Q} \quad (3.41)$$

or equivalently,

$$\mathbf{D}_{x_h w_h} = \mathbf{D}_{xw} - \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}(\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{B}\mathbf{Q} \quad (3.42)$$

3.3.5 The Process Noise, Predicted System State, and Observation Residual Vectors and their Associated Covariance Matrices

Once the constrained estimates of the system state and process noise vectors have been evaluated, the process of estimating the process noise, predicted system state, and observation residual vectors is accomplished by simply substituting the least-squares estimated quantities into the residual equations in (3.20). This section follows this process for each residual vector, and performs error propagation to determine the corresponding covariance matrix.

3.3.5.1 Process Noise Residual Vector and Its Covariance Matrix

The process noise residual vector may be estimated from (3.16) as

$$\hat{\mathbf{v}}_{w_h} = \hat{\mathbf{w}}_h - \mathbf{l}_w \quad (3.43)$$

which may be evaluated using the expression for the constrained process noise vector in (3.36) as

$$\hat{\mathbf{v}}_{w_h} = \hat{\mathbf{w}} - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}} - \mathbf{h}) - \mathbf{l}_w$$

In practice, the process noise pseudo-observation vector is typically taken to be equal to zero, which allows this equation to be simplified to

$$\hat{\mathbf{v}}_{w_h} = \hat{\mathbf{w}} - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}} - \mathbf{h}) \quad (3.44)$$

Considering the expression of $\hat{\mathbf{v}}_w$ in (2.62), (3.44) may also be rewritten as

$$\hat{\mathbf{v}}_{w_h} = \hat{\mathbf{v}}_w - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}} - \mathbf{h}) \quad (3.45)$$

The covariance matrix of $\hat{\mathbf{v}}_{w_h}$ may be determined through application of covariance law, as follows:

$$\begin{aligned} \mathbf{D}_{\hat{\mathbf{v}}_{w_h}} &= \begin{bmatrix} \mathbf{I} & -\mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H} & -\mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K} \end{bmatrix} \cdot \\ &\quad \cdot \begin{bmatrix} \mathbf{Q} & \mathbf{Q}\mathbf{B}^T & \mathbf{Q}\mathbf{B}^T\mathbf{C}^T \\ \mathbf{B}\mathbf{Q} & \mathbf{P}(k+1, k) & \mathbf{P}(k+1, k)\mathbf{C}^T \\ \mathbf{C}\mathbf{B}\mathbf{Q} & \mathbf{C}\mathbf{P}(k+1, k) & \mathbf{D}_d \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ -\mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \end{bmatrix} \\ \mathbf{D}_{\hat{\mathbf{v}}_{w_h}} &= \mathbf{Q} - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{B}\mathbf{Q} - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K}\mathbf{C}\mathbf{B}\mathbf{Q} \\ &\quad - \mathbf{Q}\mathbf{B}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &\quad + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1, k)\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &\quad + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K}\mathbf{C}\mathbf{P}(k+1, k)\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &\quad - \mathbf{Q}\mathbf{B}^T\mathbf{C}^T\mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &\quad + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1, k)\mathbf{C}^T\mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \\ &\quad + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{K}\mathbf{D}_d\mathbf{K}^T\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(k+1, k)\mathbf{B}\mathbf{Q} \end{aligned}$$

This equation eventually simplifies to

$$D_{\hat{v}_{w_h}} = QB^T C^T D_d^{-1} CBQ + QB^T (I - C^T K^T) H^T (HPH^T)^{-1} H (I - KC) BQ \quad (3.46)$$

which may be equivalently written as

$$D_{\hat{v}_{w_h}} = D_{\hat{v}_w} + QB^T P^{-1}(k+1, k) P(k+1) H^T (HPH^T)^{-1} HP(k+1) P^{-1}(k+1, k) BQ \quad (3.47)$$

3.3.5.2 Residual Vector of the Predicted System State Vector and Its Covariance Matrix

The predicted system state residual vector may be estimated from (3.14) as

$$\hat{v}_{x_h} = \hat{x} - l_x - B\hat{w} - PH^T (HPH^T)^{-1} (H\hat{x} - h) + BQB^T P^{-1}(k+1, k) PH^T (HPH^T)^{-1} (H\hat{x} - h) \quad (3.48)$$

which may be rewritten as

$$\hat{v}_{x_h} = \hat{v}_x - (I + BQB^T P^{-1}(k+1, k)) PH^T (HPH^T)^{-1} (H\hat{x} - h) \quad (3.49)$$

The covariance matrix of $\hat{v}_{x_h}$ may be determined by covariance law as

$$D_{\hat{v}_{x_h}} = \begin{bmatrix} -B & I + (BQB^T P^{-1}(k+1, k) - I) PH^T (HPH^T)^{-1} H \\ K + (BQB^T P^{-1}(k+1, k) - I) PH^T (HPH^T)^{-1} HK & -I \end{bmatrix} \cdot \begin{bmatrix} Q & QB^T & QB^T C^T & 0 \\ BQ & P(k+1, k) & P(k+1, k) C^T & D_{l_x} \\ CBQ & CP(k+1, k) & D_d & CD_{l_x} \\ 0 & D_{l_x} & D_{l_x} C^T & D_{l_x} \end{bmatrix} \begin{bmatrix} -B^T \\ I + H^T (HPH^T)^{-1} HP (P^{-1}(k+1, k) BQB^T - I) \\ K^T + K^T H^T (HPH^T)^{-1} HP (P^{-1}(k+1, k) BQB^T - I) \\ -I \end{bmatrix}$$

Defining $S = (BQB^T P^{-1}(k+1, k) - I) PH^T (HPH^T)^{-1}$, this equation becomes

$$D_{\hat{v}_{x_h}} = [-B \quad I + SH \quad K + SHK \quad -I] \begin{bmatrix} Q & QB^T & QB^T C^T & 0 \\ BQ & P(k+1, k) & P(k+1, k) C^T & D_{l_x} \\ CBQ & CP(k+1, k) & D_d & CD_{l_x} \\ 0 & D_{l_x} & D_{l_x} C^T & D_{l_x} \end{bmatrix} \begin{bmatrix} -B^T \\ I + H^T S^T \\ K^T + K^T H^T S^T \\ -I \end{bmatrix}$$

$$\begin{aligned} D_{\hat{v}_{x_h}} &= BQB^T - (I + SH)BQB^T - (K + SHK)CBQB^T - BQB^T(I + H^T S^T) \\ &+ (I + SH)P(k+1, k)(I + H^T S^T) + (K + SHK)CP(k+1, k)(I + H^T S^T) - D_{l_x}(I + H^T S^T) \\ &- BQB^T C^T(K^T + K^T H^T S^T) + (I + SH)P(k+1, k)C^T(K^T + K^T H^T S^T) \\ &+ (K + SHK)D_d(K^T + K^T H^T S^T) - D_{l_x}C^T(K^T + K^T H^T S^T) - (I + SH)D_{l_x} - (K + SHK)CD_{l_x} + D_{l_x} \end{aligned}$$

which eventually simplifies to

$$\mathbf{D}_{\hat{\mathbf{v}}_{x_h}} = \mathbf{D}_{l_x} \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C} \mathbf{D}_{l_x} + \mathbf{D}_{l_x} (\mathbf{I} - \mathbf{C}^T \mathbf{K}^T) \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} (\mathbf{I} - \mathbf{K} \mathbf{C}) \mathbf{D}_{l_x} \quad (3.50)$$

The expression in (3.50) may be equivalently expressed as

$$\mathbf{D}_{\hat{\mathbf{v}}_{x_h}} = \mathbf{D}_{\hat{\mathbf{v}}_x} + \mathbf{D}_{l_x} \mathbf{P}^{-1}(k+1, k) \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{P}^{-1}(k+1, k) \mathbf{D}_{l_x} \quad (3.51)$$

3.3.5.3 Observation Residual Vector and Its Covariance Matrix

The observation residual vector may be calculated from (3.18) as

$$\hat{\mathbf{v}}_{z_h} = \mathbf{C} \left(\hat{\mathbf{x}} - \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} (\mathbf{H} \hat{\mathbf{x}} - \mathbf{h}) \right) - \mathbf{l}_z \quad (3.52)$$

which may be rewritten as

$$\hat{\mathbf{v}}_{z_h} = (\mathbf{C} \mathbf{K} - \mathbf{I}) \mathbf{d} - \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} (\mathbf{H} \hat{\mathbf{x}} - \mathbf{h}) \quad (3.53)$$

or equivalently,

$$\hat{\mathbf{v}}_{z_h} = \hat{\mathbf{v}}_z - \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} (\mathbf{H} \hat{\mathbf{x}} - \mathbf{h}) \quad (3.54)$$

The covariance matrix of $\hat{\mathbf{v}}_{z_h}$ may be determined via covariance law as

$$\begin{aligned} \mathbf{D}_{\hat{\mathbf{v}}_{z_h}} &= \begin{bmatrix} -\mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} & \mathbf{C} \mathbf{K} - \mathbf{I} - \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{P}(k+1, k) & \mathbf{P}(k+1, k) \mathbf{C}^T \\ \mathbf{C} \mathbf{P}(k+1, k) & \mathbf{D}_d \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} -\mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \\ \mathbf{K}^T \mathbf{C}^T - \mathbf{I} - \mathbf{K}^T \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \end{bmatrix} \\ \mathbf{D}_{\hat{\mathbf{v}}_{z_h}} &= \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1, k) \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T - \mathbf{C} \mathbf{K} \mathbf{C} \mathbf{P}(k+1, k) \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \\ &\quad + \mathbf{C} \mathbf{P}(k+1, k) \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T + \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K} \mathbf{C} \mathbf{P}(k+1, k) \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \\ &\quad - \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1, k) \mathbf{C}^T \mathbf{K}^T \mathbf{C}^T + \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1, k) \mathbf{C}^T \\ &\quad + \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1, k) \mathbf{C}^T \mathbf{K}^T \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T + \mathbf{C} \mathbf{K} \mathbf{D}_d \mathbf{K}^T \mathbf{C}^T - \mathbf{C} \mathbf{K} \mathbf{D}_d \\ &\quad - \mathbf{C} \mathbf{K} \mathbf{D}_d \mathbf{K}^T \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T - \mathbf{D}_d \mathbf{K}^T \mathbf{C}^T + \mathbf{D}_d + \mathbf{D}_d \mathbf{K}^T \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \\ &\quad - \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K} \mathbf{D}_d \mathbf{K}^T \mathbf{C}^T + \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K} \mathbf{D}_d \\ &\quad + \mathbf{C} \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K} \mathbf{D}_d \mathbf{K}^T \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P} \mathbf{C}^T \end{aligned}$$

which eventually simplifies to

$$\mathbf{D}_{\hat{\mathbf{v}}_{z_h}} = (\mathbf{I} - \mathbf{C}\mathbf{K})\mathbf{R} + \mathbf{C}\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{C}^T \quad (3.55)$$

or equivalently,

$$\mathbf{D}_{\hat{\mathbf{v}}_{z_h}} = \mathbf{D}_{\hat{\mathbf{v}}_z} + \mathbf{C}\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}\mathbf{C}^T \quad (3.56)$$

3.3.6 Extended Kalman Filter with State Constraints

In the Extended Kalman Filter (EKF), the system model, measurement model, and system state constraints may be extended to accommodate non-linear equations in general:

$$\mathbf{x}(k+1) = \mathbf{A}(\mathbf{x}(k)) + \mathbf{B}\mathbf{w}(k) \quad (3.57)$$

$$\mathbf{z}(k+1) + \mathbf{v}_z = \mathbf{C}(\mathbf{x}(k+1)) \quad (3.58)$$

and

$$\mathbf{H}(\mathbf{x}(k+1)) - \mathbf{h} = \mathbf{0} \quad (3.59)$$

The three sets of pseudo-observations are modified to become:

$$\mathbf{l}_x + \mathbf{B}(\mathbf{w}(k)) + \mathbf{v}_x = \mathbf{A}(\mathbf{x}(k)) \quad (3.60)$$

$$\mathbf{l}_w + \mathbf{v}_w = \mathbf{w}(k) \quad (3.61)$$

and

$$\mathbf{l}_z + \mathbf{v}_z = \mathbf{C}(\mathbf{x}(k+1)) \quad (3.62)$$

To facilitate variance propagation and model linearization, each non-linear model has an associated Jacobian matrix, defined as:

$$\mathbf{A}(k+1, k) = \frac{\partial \mathbf{A}(\mathbf{x}(k))}{\partial \mathbf{x}(k)} \quad (3.63)$$

$$\mathbf{C}(k+1) = \frac{\partial \mathbf{C}(\mathbf{x}(k+1))}{\partial \mathbf{x}(k+1)} \quad (3.64)$$

and

$$\mathbf{H}(k+1) = \frac{\partial \mathbf{H}(\mathbf{x}(k+1))}{\partial \mathbf{x}(k+1)} \quad (3.65)$$

In the EKF, each non-linear model is linearized using the Jacobian matrices given above to facilitate the matrix computations necessary to estimate the optimal system state and process noise vectors. Given the models provided from (3.57) to (3.62) and the Jacobian matrices provided from (3.63) to (3.65), the entire procedure for the linear system state-constrained Kalman Filter may be followed for the EKF as well, with some small adjustments:

- Any instance of $\mathbf{A}(k + 1, k)$, $\mathbf{B}(k + 1, k)$, $\mathbf{C}(k + 1)$, or $\mathbf{H}(k + 1)$ in the linear system state-constrained Kalman Filter equations is replaced by its associated Jacobian matrix instead.
- The predicted state vector is calculated as $\hat{\mathbf{x}}(k + 1, k) = \mathbf{A}(\mathbf{x}(k))$.
- The system innovation is defined to be $\mathbf{d} = \mathbf{C}(\hat{\mathbf{x}}(k + 1, k)) - \mathbf{z}$ instead of $\mathbf{d} = \mathbf{C}\hat{\mathbf{x}}(k + 1, k) - \mathbf{z}$.
- Any instance of $\mathbf{H}\hat{\mathbf{x}}(k + 1) - \mathbf{h}$ in the linear system state-constrained Kalman Filter equations is replaced by an evaluation of the non-linear constraint equations in (3.59).

Aside from these adjustments, the overall process of the EKF is identical to that of the linear system state-constrained Kalman Filter.

3.4 Summary of the System State-Constrained Kalman Filter

A workflow diagram describing the overall processing flow for the system state-constrained EKF is provided in Figure 3.4.1.

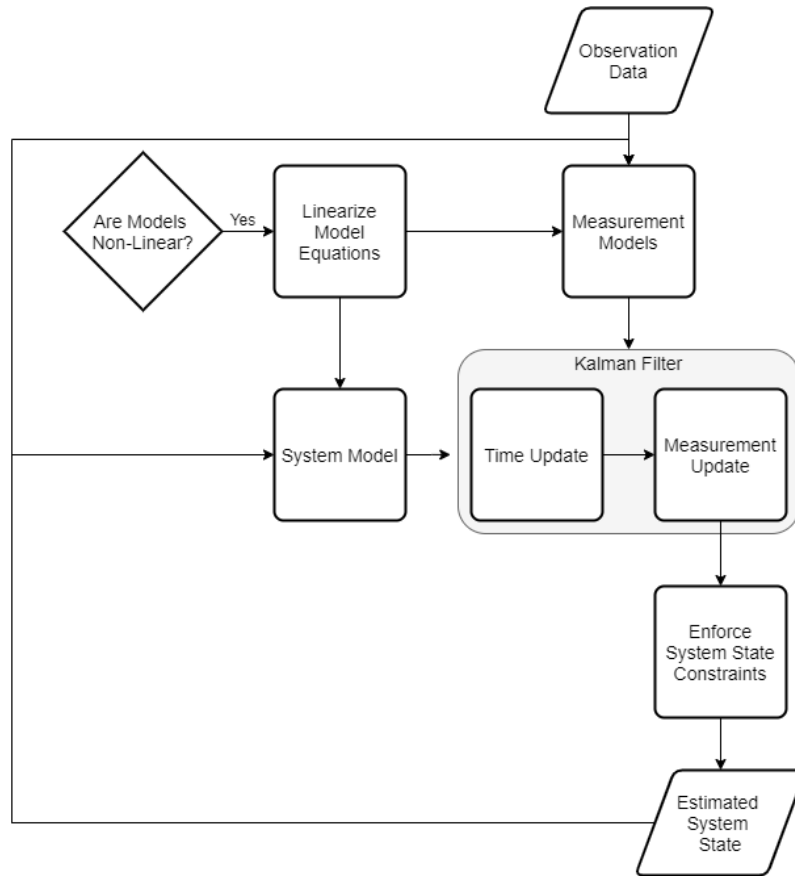


Figure 3.4.1: A workflow diagram describing the overall processing flow for the system state-constrained EKF. Note that the system state constraints are enforced after the standard Kalman Filter equations have been applied.

From Figure 3.4.1 and the derivations in the preceding sections, the process of enforcing system state constraints may be considered to be separate from the core Kalman Filtering process, since all constrained quantities may be expressed relative to their unconstrained counterparts. The overall process of applying system state constraints to an EKF may therefore be defined as:

- i. Following the standard EKF process workflow.
- ii. Enforcing the system state constraints as adjustments to the unconstrained quantities determined through the standard EKF.

With this process in mind, Section 3.4.1 describes the Kalman Filter equations without system state constraints, as derived in [Wang, 1997; Caspary and Wang, 1998]. Section 3.4.2 then goes on to describe how to modify the unconstrained quantities to enforce the system state constraint equations.

3.4.1 Kalman Filter Equations Without System State Constraints

As determined in [Wang, 1997; Caspary and Wang, 1998], the optimal estimate of the system state vector is given by

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1, k) + \mathbf{K}\mathbf{d} \quad (3.66)$$

with an associated covariance matrix of

$$\mathbf{P}(k+1) = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}(k+1, k)(\mathbf{I} - \mathbf{C}^T\mathbf{K}^T) + \mathbf{K}\mathbf{R}\mathbf{K}^T \quad (3.67)$$

The optimal estimate of the process noise vector is given by

$$\hat{\mathbf{w}}(k+1) = \mathbf{w}_0(k+1) + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{K}\mathbf{d} \quad (3.68)$$

with an associated covariance matrix of

$$\mathbf{D}_{\hat{\mathbf{w}}} = \mathbf{Q} - \mathbf{Q}\mathbf{B}^T\mathbf{C}^T\mathbf{D}_d^{-1}\mathbf{C}\mathbf{B}\mathbf{Q} \quad (3.69)$$

Once these quantities have been estimated, the residual vectors for the predicted system state, process noise, and observations may be calculated alongside their associated covariance matrices as follows:

- The residuals of the predicted system state vector:

$$\hat{\mathbf{v}}_x = \mathbf{D}_{l_x}\mathbf{P}^{-1}(k+1, k)\mathbf{K}\mathbf{d} \quad (3.70)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_x} = \mathbf{D}_{l_x}\mathbf{C}^T\mathbf{D}_d^{-1}\mathbf{C}\mathbf{D}_{l_x} \quad (3.71)$$

- The process noise residuals:

$$\hat{\mathbf{v}}_w = \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1, k)\mathbf{K}\mathbf{d} \quad (3.72)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_w} = \mathbf{Q}\mathbf{B}^T\mathbf{C}^T\mathbf{D}_d^{-1}\mathbf{C}\mathbf{B}\mathbf{Q} \quad (3.73)$$

- The observation residuals:

$$\hat{\mathbf{v}}_z = (\mathbf{C}\mathbf{K} - \mathbf{I})\mathbf{d} \quad (3.74)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_z} = (\mathbf{I} - \mathbf{C}\mathbf{K})\mathbf{R} \quad (3.75)$$

3.4.2 Enforcing System State Constraints

Once the unconstrained quantities in Section 3.4.1 have been estimated, the system state constraints may be enforced by making the necessary adjustments to them. The constrained estimate of the system state vector is given by

$$\hat{\mathbf{x}}_h(k+1) = \hat{\mathbf{x}}(k+1) - \mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}}(k+1) - \mathbf{h}) \quad (3.76)$$

with an associated covariance matrix of

$$\mathbf{P}_h(k+1) = \mathbf{P}(k+1) - \mathbf{P}(k+1)\mathbf{H}(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}\mathbf{H}^T\mathbf{P}(k+1) \quad (3.77)$$

The constrained estimate of the process noise vector is given by

$$\hat{\mathbf{w}}_h(k+1) = \hat{\mathbf{w}}(k+1) - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1,k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}}(k+1) - \mathbf{h}) \quad (3.78)$$

with an associated covariance matrix of

$$\mathbf{D}_{w_h} = \mathbf{D}_w - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1,k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1)\mathbf{P}^{-1}(k+1,k)\mathbf{B}\mathbf{Q} \quad (3.79)$$

As before, the constrained residual vectors may be calculated alongside their associated covariance matrices as follows:

- The residuals of the predicted system state vector:

$$\hat{\mathbf{v}}_{x_h} = \hat{\mathbf{v}}_x - \left(\mathbf{I} + \mathbf{B}\mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1,k)\right)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}}(k+1) - \mathbf{h}) \quad (3.80)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_{x_h}} = \mathbf{D}_{\hat{\mathbf{v}}_x} + \mathbf{D}_{l_x}\mathbf{P}^{-1}(k+1,k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1)\mathbf{P}^{-1}(k+1,k)\mathbf{D}_{l_x} \quad (3.81)$$

- The process noise residuals:

$$\hat{\mathbf{v}}_{w_h} = \hat{\mathbf{v}}_w - \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1,k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}}(k+1) - \mathbf{h}) \quad (3.82)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_{w_h}} = \mathbf{D}_{\hat{\mathbf{v}}_w} + \mathbf{Q}\mathbf{B}^T\mathbf{P}^{-1}(k+1,k)\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1)\mathbf{P}^{-1}(k+1,k)\mathbf{B}\mathbf{Q} \quad (3.83)$$

- The observation residuals:

$$\hat{\mathbf{v}}_{z_h} = \hat{\mathbf{v}}_z - \mathbf{C}\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}(\mathbf{H}\hat{\mathbf{x}}(k+1) - \mathbf{h}) \quad (3.84)$$

$$\mathbf{D}_{\hat{\mathbf{v}}_{z_h}} = \mathbf{D}_{\hat{\mathbf{v}}_z} + \mathbf{C}\mathbf{P}(k+1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k+1)\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}(k+1)\mathbf{C}^T \quad (3.85)$$

3.5 Comprehensive Error Analysis in the System State Constrained Kalman Filter

The preceding sections of this chapter laid a foundation for a more thorough and rigorous error analysis in Kalman Filters with system state constraints. This section explores the statistical analysis that is made possible by this research's approach to enforcing system state constraints in a Kalman Filter.

In Kalman Filtering, it is very common to use the system innovation vector for most statistical tests. This is largely a matter of practicality for embedded real-time systems, since the system innovation vector is evaluated prior to performing the measurement update, which therefore allows statistical testing to be done without needing to apply the measurement update multiple times. This allows for very efficient statistical testing to be accomplished, but does have some disadvantages in terms of the detailed conclusions that may be drawn from these tests in a post-processing environment. The system innovation vector includes errors from all three statistically-independent error sources in the Kalman Filter, since it is influenced by observation errors, process noise errors, and errors in the predicted system state. It is therefore challenging to know what precisely is impacting the system innovation vector, since it mixes together errors from the system model, measurement model, and previously-estimated system state.

This research moves beyond the system innovation vector for the statistical analysis of errors and towards using the residual quantities as described in Section 3.3.5. There are three main categories of the types of error analysis enabled for state-constrained Kalman Filters by this work:

- i. Residual Analysis

This analysis focuses on evaluating the estimated residual vectors themselves. This includes the observation residuals, process noise residuals, and residuals of the predicted system state vector.

The observation residuals provide insight into how well the observations fit the measurement model. This can be used for the standard outlier detection procedure, or to fuel improvements to observation models that are underperforming.

The process noise residuals provide insight into how well the system models are performing. This can be used to evaluate the performance of the system model to determine how well it reflects the actual motion of the system. Additionally, for a system that sometimes uses a

simplified system model (e.g. constant linear velocity instead of constant linear acceleration), the process noise residuals can provide insight into when the simplified system model is sufficiently or insufficiently describing the system dynamics.

The residuals of the predicted system state vector can provide insight into whether there are issues in the overall system state estimation process from the previous epoch to the current one. These residuals generally have high correlation with one another, and so it can be impractical to use the individual elements in the residual vector of the predicted system states to draw conclusions.

ii. Reliability Analysis

Reliability analysis can be thought of as focusing on the system's resilience to errors. Generally, reliability analysis may be conducted from two essential perspectives: the redundancy contributions made by different parts of the Kalman Filter and the minimum detectable outliers in the Kalman Filter [Knight et al, 2010].

For any residual vector that has a known cofactor or covariance matrix, the contribution of that residual vector to the overall redundancy of an overdetermined system may be determined. Additionally, the residual vectors may be divided into partitioned subvectors with a separately determined redundancy contribution for each partitioned element as long as the subvectors are not correlated to one another. Taking this to its extreme, the redundancy contribution for each element in a residual vector associated with a group of uncorrelated measurements may be determined, and these values are called redundancy indices. Redundancy indices take on values between 0 and 1, with a value of 0 corresponding to having no redundancy (i.e., a high leverage observation element or strong influential observation element) and a value of 1 corresponding to being completely redundant. This can be used to evaluate the necessity of different observations in the positioning Kalman Filter, with lower redundancy indices corresponding to observations that might benefit from being supplemented and higher redundancy indices corresponding to observations that are not providing unique information, without requiring data to be collected first.

iii. Variance Component Estimation

Variance Component Estimation (VCE) provides information about the overall level of agreement between the *a posteriori* residual values and the assumed *a priori* accuracies of

elements in the Kalman Filter. When applied ongoingly throughout a dataset, VCE allows for an adaptive dynamic rescaling of the covariance matrices for the predicted system state, process noise, and observations.

VCE can be seen as a culmination of both residual analysis and reliability analysis, since it combines both of them together to provide an overall measure of system performance. As with both residual analysis and reliability analysis, the vectors being analysed can be partitioned to determine separate variance component factors for each partitioned element, which allows for more granular information to be gleaned from the VCE analysis.

These three key categories work together to characterize a PosAtt system's errors, reliability, and solution quality. Using comprehensive error analysis tools provides a complete picture of the performance of a PosAtt system on the whole, as well as the performance of its constituent parts. In the event of equipment failure or unreliable data, these tools serve to provide important information that may be used to diagnose and rectify specific problems.

The following sections provide more detail on these three categories of error analysis in the system state-constrained Kalman Filter.

3.5.1 Residual Analysis

The residual analysis made possible by this research's approach to enforcing system state constraints is largely identical to the typical residual analysis discussed in Section 2.5.1.1.1, with the most common use case being outlier detection. The main point of departure from the literature here is in the interpretation of the process noise residuals, since these were previously unable to be estimated for a state-constrained Kalman Filter and were therefore inaccessible for residual analysis.

This research generally uses the residual elements to test for outliers. If a residual value is *not* an outlier, it should be statistically equal to zero, corresponding to the null hypothesis

$$H_0: \tilde{v}_i = 0, H_a: \tilde{v}_i \neq 0 \quad (3.86)$$

where $\tilde{v}_i$ denotes the true value of residual element i .

According to Central Limit Theorem, the sampling distribution of the mean of a dataset converges to a normal distribution. As the number of observations increases, the *estimated* residual value v_i should therefore converge towards a normal distribution with zero mean and an *a priori*

standard deviation of σ_{v_i} . Under this assumed distribution, the null hypothesis may be tested using the test statistic

$$s_i = \frac{v_i}{\sigma_{v_i}} \sim N(0,1) \quad (3.87)$$

where s_i is the standardized residual corresponding to residual element v_i .

The *a posteriori* standard deviation of the residual element, $\hat{\sigma}_{v_i}$, may also be used to test the null hypothesis using the test statistic

$$s_i = \frac{v_i}{\hat{\sigma}_{v_i}} \sim t(dof) \quad (3.88)$$

where *dof* denotes the degrees of freedom in estimating $\hat{\sigma}_{v_i}$.

The test statistics in (3.87) and (3.88) are applicable to both observation and process noise residual vector elements. Any testing of the residuals of the predicted state vector is complicated by the fact that state vector elements are typically strongly correlated with one another.

3.5.2 Reliability Analysis

Reliability analysis provides unique opportunities in system design and testing, since the estimation of the redundancy contribution matrices in the state-constrained Kalman Filter do not depend upon having actual observations. This allows for reliability analysis to be performed over simulated trajectories, which provides the opportunity to test and compare the performance of different system configurations without needing to assemble a physical system to test.

3.5.2.1 Redundancy Contribution Estimation

The estimation of the redundancy contribution of each residual vector follows a similar process to that described in Section 2.5.1.1.2, with the general formula used to estimate the redundancy contribution given by

$$r_v = \text{tr}(\mathbf{D}_v \mathbf{D}_l^{-1}) \quad (3.89)$$

where $\mathbf{D}_v$ is the estimated covariance matrix for the residual vector, and $\mathbf{D}_l$ is the estimated covariance matrix (i.e. the cofactor matrix) for the corresponding observations.

For the three residual vectors in the **unconstrained** Kalman Filter, (3.89) becomes

$$r_x = \text{tr}(\mathbf{D}_{l_x} \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C}) \quad (3.90)$$

$$r_w = \text{tr}(\mathbf{Q} \mathbf{B}^T \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C} \mathbf{B}) \quad (3.91)$$

and

$$r_z = \text{tr}(\mathbf{I} - \mathbf{C} \mathbf{K}) \quad (3.92)$$

The total redundancy across all three pseudo-observation vectors is given by

$$r = r_x + r_w + r_z = p \quad (3.93)$$

where p is the number of elements in the observation vector [Wang, 1997; Wang, 2009].

For the **constrained** predicted system state residual vector, (3.89) becomes

$$\begin{aligned} r_{x_h} &= \text{tr}(\mathbf{D}_{l_x} \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C}) \\ &+ \text{tr} \left(\mathbf{D}_{l_x} \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \right) \\ r_{x_h} &= r_x + \text{tr} \left(\mathbf{D}_{l_x} \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \right) \end{aligned} \quad (3.94)$$

For the **constrained** process noise residual vector, (3.89) becomes

$$\begin{aligned} r_{w_h} &= \text{tr}(\mathbf{Q} \mathbf{B}^T \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C} \mathbf{B}) \\ &+ \text{tr} \left(\mathbf{Q} \mathbf{B}^T \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \mathbf{B} \right) \\ r_{w_h} &= r_w + \text{tr} \left(\mathbf{Q} \mathbf{B}^T \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \mathbf{B} \right) \end{aligned} \quad (3.95)$$

For the **constrained** observation residual vector, (3.89) becomes

$$\begin{aligned} r_{z_h} &= \text{tr}(\mathbf{D}_{v_z} \mathbf{R}^{-1}) + \text{tr} \left(\mathbf{C} \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{C}^T \mathbf{R}^{-1} \right) \\ r_{z_h} &= r_z + \text{tr} \left(\mathbf{C} \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{C}^T \mathbf{R}^{-1} \right) \end{aligned} \quad (3.96)$$

The total redundancy across all three pseudo-observation vectors in the constrained Kalman Filter is given by

$$r = r_{x_h} + r_{w_h} + r_{z_h}$$

$$\begin{aligned}
r = p &+ \text{tr} \left(\mathbf{D}_{l_x} \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \right) \\
&+ \text{tr} \left(\mathbf{Q} \mathbf{B}^T \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \mathbf{B} \right) \\
&+ \text{tr} \left(\mathbf{C} \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{C}^T \mathbf{R}^{-1} \right)
\end{aligned}$$

Using the fact that $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}) = \text{tr}(\mathbf{A}^T \mathbf{B}) = \text{tr}(\mathbf{B}^T \mathbf{A})$, this equation simplifies to

$$r = p + \text{tr} \left((\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T \right) = p + \text{tr}(\mathbf{I}_{h \times h})$$

leading to an overall redundancy of

$$r = p + h \quad (3.97)$$

where h is the number of system state constraints being enforced in the state-constrained Kalman Filter.

There is one very important property illustrated by (3.94), (3.95) and (3.96). As long as all covariance matrices used in the Kalman Filter are positive definite, the quantities inside the $\text{tr}(\cdot)$ in (3.94), (3.95), and (3.96) are all also positive definite and therefore their trace will be positive. This means that in the presence of system state constraints, the overall system redundancy is higher, which is very intuitive. This added system redundancy is distributed amongst the system state, process noise, and observation system redundancies.

Additionally, the overall redundancy that is added to the system by the system state constraint equations is equal to the number of system state constraints that are being enforced, which is also very intuitive.

The redundancy indices for individual elements in a residual vector may be estimated to evaluate each element's contribution to the overall system redundancy as long as those elements are uncorrelated to each other. In the standard form of the Kalman filter, this is usually a valid argument for the process noise vector when $\mathbf{Q}$ is diagonal and for the measurement vector when $\mathbf{R}$ is diagonal; however, this assumption is generally invalid for the predicted state vector, since its elements are typically correlated to each other. The individual elements' contributions to the overall system redundancy is determined with the help of the matrix products found within the $\text{tr}(\cdot)$ in each of equations (3.95), and (3.96). These matrices are defined to be

$$\begin{aligned}
\mathbf{R}_{w_h} &= \mathbf{Q} \mathbf{B}^T \mathbf{C}^T \mathbf{D}_d^{-1} \mathbf{C} \mathbf{B} \\
&+ \mathbf{Q} \mathbf{B}^T \mathbf{P}^{-1}(k+1, k) \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{P}^{-1}(k+1, k) \mathbf{B}
\end{aligned} \quad (3.98)$$

and

$$\mathbf{R}_{z_h} = \mathbf{D}_{v_z} \mathbf{R}^{-1} + \mathbf{C} \mathbf{P}(k+1) \mathbf{H}^T (\mathbf{H} \mathbf{P}(k+1) \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{P}(k+1) \mathbf{C}^T \mathbf{R}^{-1} \quad (3.99)$$

When $\mathbf{Q}$ is diagonal, the redundant index of an individual process noise element may be determined from (3.98) and is given by

$$(r_{w_h})_i = (\mathbf{R}_{w_h})_{ii} \quad (3.100)$$

When $\mathbf{R}$ is diagonal, the redundant index of an individual observation element may be determined from (3.99) and given by

$$(r_{z_h})_i = (\mathbf{R}_{z_h})_{ii} \quad (3.101)$$

As previously mentioned, the redundancy index for a residual corresponding to a statistically independent observation has a value between 0 and 1, and describes how redundant the corresponding quantity is. A value of 1 indicates that the corresponding quantity is completely redundant, and could be removed from the filter with no effect. A value of 0 indicates that the corresponding quantity provides wholly unique information, and that if it were removed from the filter a solution would not be possible to achieve. When a quantity is wholly non-redundant, any associated residual would be equal to 0, so these quantities are not able to have their quality evaluated readily and are undesirable. These extreme values are very uncommon in practice, and in rough terms, it is generally desirable for observations to have redundant indices that are not too close to 0 or to 1. In other words, it is desirable for observations to provide unique, but overlapping information to facilitate both a stable Kalman Filter solution and reliable statistical testing for quality assurance.

When correlation exists between observations, the redundancy indices can have values less than 0 or greater than 1, and are not readily interpreted on their own. To conduct a proper reliability analysis under these circumstances, it is essential to account for the off-diagonal elements of $\mathbf{R}_{w_h}$ and $\mathbf{R}_{z_h}$, as in [Wang, 1999]. However, the subtotal redundancy index of a group correlated observations remains an essential parameter for estimating their group variance factor.

3.5.3 Variance Component Estimation

Variance Component Estimation (VCE) forms an essential part of evaluating the overall performance of a Kalman Filter. As defined in Section 2.5.1.1.2, variance factors describe the overall level of agreement in performance between the *a posteriori* estimates and their *a priori* assumed

accuracies, on the assumption that they conform to specified normal distributions. VCE can be performed at different levels, ranging from an all-inclusive variance of unit weight that describes the overall system's performance, to individual variances of unit weight for each type of uncorrelated measurement. Additionally, VCE may be estimated globally over the entire observation period, regionally over a time window, or locally at a specific time instant.

A variance factor may be estimated to describe the scale factor between the *a priori* cofactor matrices corresponding to each set of pseudo-observations comprising the Kalman Filter and the *a posteriori* estimate of the corresponding covariance matrix. Variance factors are unitless quantities that describe the ratio of the *a posteriori* estimated variance to the *a priori* variance. Generally, if an estimated variance factor has a value near 1, the system can usually be considered well-tuned, but a value significantly different from 1 may indicate the following:

i. **When the variance factor is greater than 1:**

A large variance factor broadly indicates that the system is performing worse than expected, with the following potential sources for performance issues:

- a. The observations (or pseudo-observations) have higher variances than had been assumed.
- b. There is an issue causing the models of the observations to not accurately describe the actual observations. This could be indication of an under-parameterized state vector.

ii. **When the variance factor is less than 1:**

A small variance factor broadly indicates that the system is performing better than would be expected. While it may be tempting to see this as a positive outcome, it still indicates poor tuning of the Kalman Filter, and can be a sign of serious problems if the state vector is over-parameterized.

For the predicted system state, process noise, and observation pseudo-observation vectors at an arbitrary observation epoch, the individual variance factors are estimated as follows

$$\hat{\sigma}_{0,x_h}^2 = \frac{\mathbf{v}_{x_h}^T \mathbf{D}_{l_x}^{-1} \mathbf{v}_{x_h}}{r_{x_h}} \quad (3.102)$$

$$\hat{\sigma}_{0,w_h}^2 = \frac{\mathbf{v}_{w_h}^T \mathbf{Q}^{-1} \mathbf{v}_{w_h}}{r_{w_h}} \quad (3.103)$$

and

$$\hat{\sigma}_{0,z_h}^2 = \frac{v_{z_h}^T R^{-1} v_{z_h}}{r_{z_h}} \quad (3.104)$$

Recognizing that the sum of $r_{x_h} + r_{w_h} + r_{z_h} = p + h$ as determined in Section 3.5.2.1, an overall variance factor or variance of unit weight for the whole filtering system at an arbitrary observation epoch can be estimated as follows

$$\hat{\sigma}_0^2 = \frac{v_{x_h}^T D_{l_x}^{-1} v_{x_h} + v_{w_h}^T Q^{-1} v_{w_h} + v_{z_h}^T R^{-1} v_{z_h}}{p + h} \quad (3.105)$$

Since each variance factor describes a scale factor between an *a priori* cofactor matrix and its *a posteriori* estimate of the corresponding covariance matrix, the estimated variance factors are unitless.

Additionally, variance factors are not limited to being estimated for the entirety of each pseudo-observation vector, and each pseudo-observation vector can be partitioned into functionally-related elements (or independent measurement types) that could be considered to share a variance component. For example, when partitioning an observation vector in a positioning Kalman Filter for VCE, a separate variance component factor can be estimated for the L1 C/A measurements, L1 carrier phase measurements, L2 carrier phase measurements, IMU gyroscope measurements, and IMU accelerometer measurements. This allows for a separate analysis of the overall quality of each of these types of positioning measurements.

To accomplish this more granular estimation of variance component factors, the following definition of a partitioned residual vector is used:

$$v = [v_1^T \quad v_2^T \quad \dots \quad v_n^T]^T \quad (3.106)$$

with the associated cofactor matrices defined as

$$D_1 = \begin{bmatrix} I & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} D_v, D_2 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} D_v, \dots, D_n = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \end{bmatrix} D_v \quad (3.107)$$

For the purposes of VCE, it is also important to have access to full-length residual vectors containing only the elements of the partitioned residual vectors, defined as

$$V_1 = \begin{bmatrix} I & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} v, V_2 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} v, \dots, V_n = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \end{bmatrix} v \quad (3.108)$$

Using this definition of the partitioned residual vector and the definition of the redundancy contribution in (3.89), the redundancy contribution of partitioned element i may be calculated using the equation

$$r_{v_i} = \text{tr}(\mathbf{D}_i \mathbf{D}_i^{-1}) \quad (3.109)$$

and the value in (3.109) may be used to calculate the associated variance component factor as

$$\hat{\sigma}_{0,i}^2 = \frac{\mathbf{V}_i^T \mathbf{D}_i^{-1} \mathbf{V}_i}{r_{v_i}} \quad (3.110)$$

This strategy can be used to conduct VCE for any partitioning of each of the independent pseudo-observation vectors, provided that there is no expected correlation between different partitioned vectors.

Notably, the double-differenced GNSS observations become correlated due to the use of the same “base” satellite observations in the calculation of the double-differenced code and carrier phase observations. The vectors of the double-differenced code and carrier phase observations can be used to estimate their group variance factors using (3.109) and (3.110) for the overall groupwise GNSS observables, i.e., L1 C/A, L1 and L2 carrier phases.

3.6 Numerical Example: Planar Motion of a Ground Vehicle

To illustrate the behaviour of the state-constrained Kalman Filter formulation proposed in this chapter and the additional processing tools it provides, a simple example of the simulated planar motion of a ground vehicle is explored in this section. The data for the simulated trajectory, which is shown in Figure 3.6.1, was generated using the estimated trajectory for an actual ground vehicle’s motion (see Section 5.8 for processing examples using real GPS and IMU observations for the trajectory’s estimation).

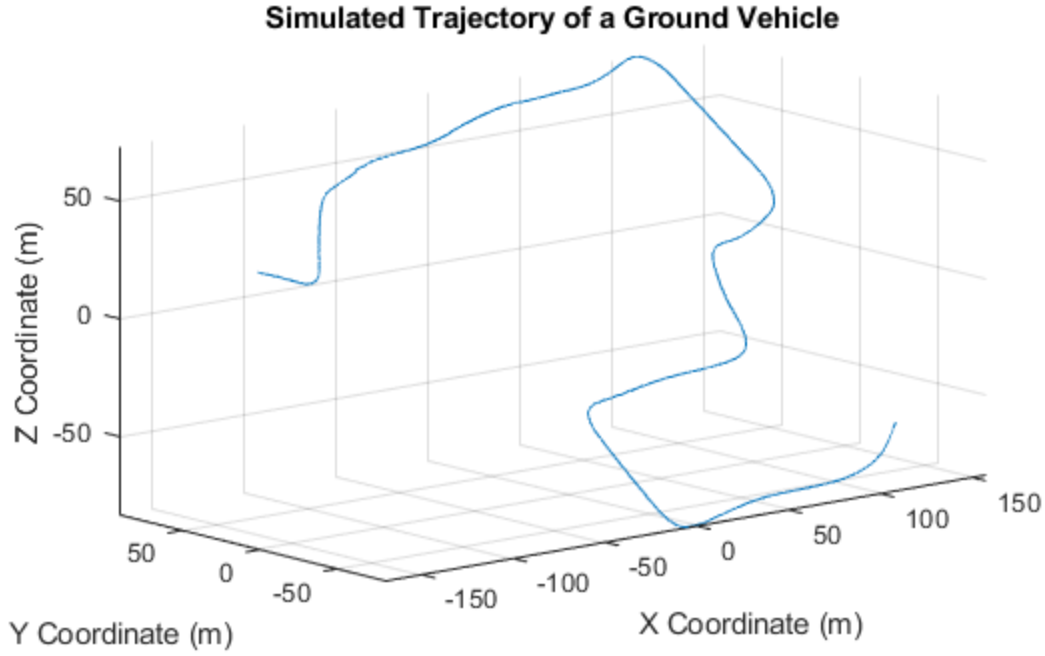


Figure 3.6.1: The trajectory used to simulate positioning data for the processing example in this section. Note that the motion of the vehicle is constrained by the planar road surface, and in this processing example, the entire local road network fits well to a planar surface.

3.6.1 Formulation of the Problem

For the purposes of this example, the system state vector is defined to include elements corresponding to the system's three-dimensional position, velocity, and acceleration:

$$\mathbf{x} = [\mathbf{r}^T \quad \mathbf{v}^T \quad \mathbf{a}^T]^T \quad (3.111)$$

where $\mathbf{r}$ is the system's position vector, $\mathbf{v}$ is the system's velocity vector, and $\mathbf{a}$ is the system's acceleration vector, all expressed in the (x, y, z) coordinate frame shown in Figure 3.6.1.

Using the approach of a standard Position Velocity and Acceleration (PVA) filter and assuming that the system experiences constant acceleration between successive observation epochs, the discretized system model may be defined as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{w}(k) = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 & \frac{\Delta t^2}{2} \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \Delta t \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \frac{\Delta t^3}{6} \mathbf{I}_3 \\ \frac{\Delta t^2}{2} \mathbf{I}_3 \\ \Delta t \mathbf{I}_3 \end{bmatrix} \mathbf{j}(k) \quad (3.112)$$

where $\mathbf{j}$ represents the system jerk, or third-order time derivative of the system position, and constitutes the process noise in the system model.

In this example, the position of the system is directly observed as (x, y, z) coordinates at every observation epoch. Mathematically, the measurement model is therefore given by

$$\mathbf{z}(k+1) + \mathbf{v}_z = \mathbf{C}\mathbf{x}(k+1) = [\mathbf{I}_3 \quad \mathbf{0}_3 \quad \mathbf{0}_3] \mathbf{x}(k+1) \quad (3.113)$$

For the observed system positions, a standard deviation of 1.0 metres was used in generating the simulated observations. A plot of all simulated observations is shown in Figure 3.6.2.

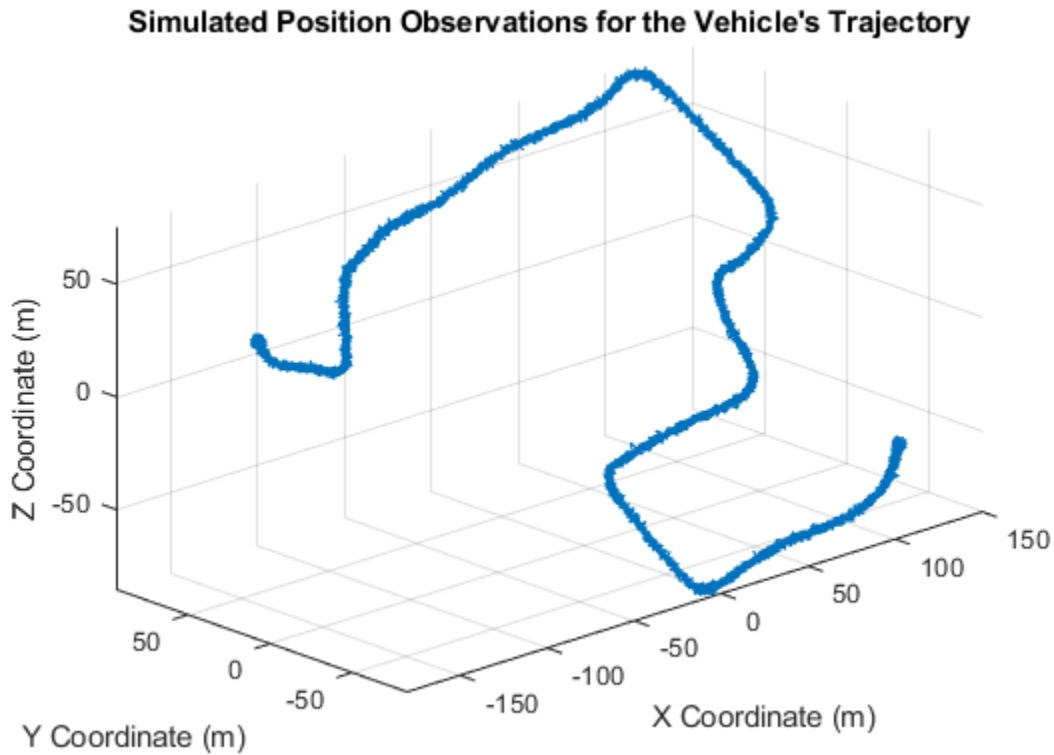


Figure 3.6.2: Simulated position observations for the ground vehicle trajectory. Position observations were generated with a standard deviation of 1.0 metres, with no correlation between x -, y -, and z -component measurements.

In this particular example, there are two sets of constraint equations that may be applied. The first corresponds to the periods of time where the vehicle is stationary, at the beginning and at the end

of the simulated dataset. At these times, the zero velocity/acceleration update is conducted, which is realized through the linear constraint equations

$$\mathbf{H}\mathbf{x}(k+1) - \mathbf{h} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{x}(k+1) - \mathbf{0} = \mathbf{0} \quad (3.114)$$

The second set of constraints is applied at all times in the dataset, and corresponds to the ground vehicle being constrained to move along the planar road surface. Constraining the system's motion to the planar surface is realized through the linear constraint equations

$$\mathbf{H}\mathbf{x}(k+1) - \mathbf{h} = [-0.1915 \quad 1.0389 \quad -1.0000 \quad \mathbf{0}] \mathbf{x}(k+1) - 0 = 0 \quad (3.115)$$

where $(-0.1915, 1.0389, -1.0000)$ are the coefficients defining the specific planar surface used in this processing example, and $\mathbf{h} = 0$ defines the z-axis intercept of that planar surface.

3.6.2 Estimated Trajectory for the Simulated Dataset

Following the processing workflow laid out in Section 3.4, the estimated trajectory of the ground vehicle with all applied system-state constraints is shown in Figure 3.6.3, with the associated position profiles shown in Figure 3.6.4, the associated velocity profiles shown in Figure 3.6.5, and the associated acceleration profiles shown in Figure 3.6.6.

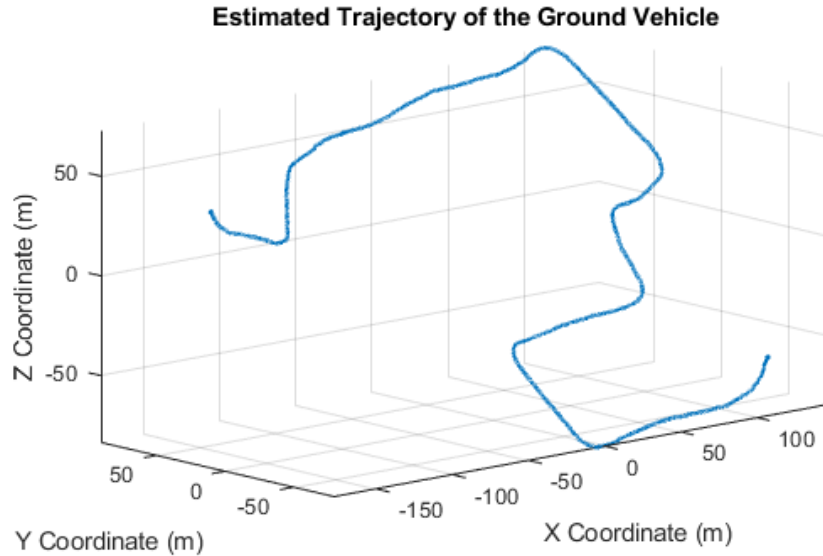


Figure 3.6.3: Estimated trajectory for the simulated positioning observations.

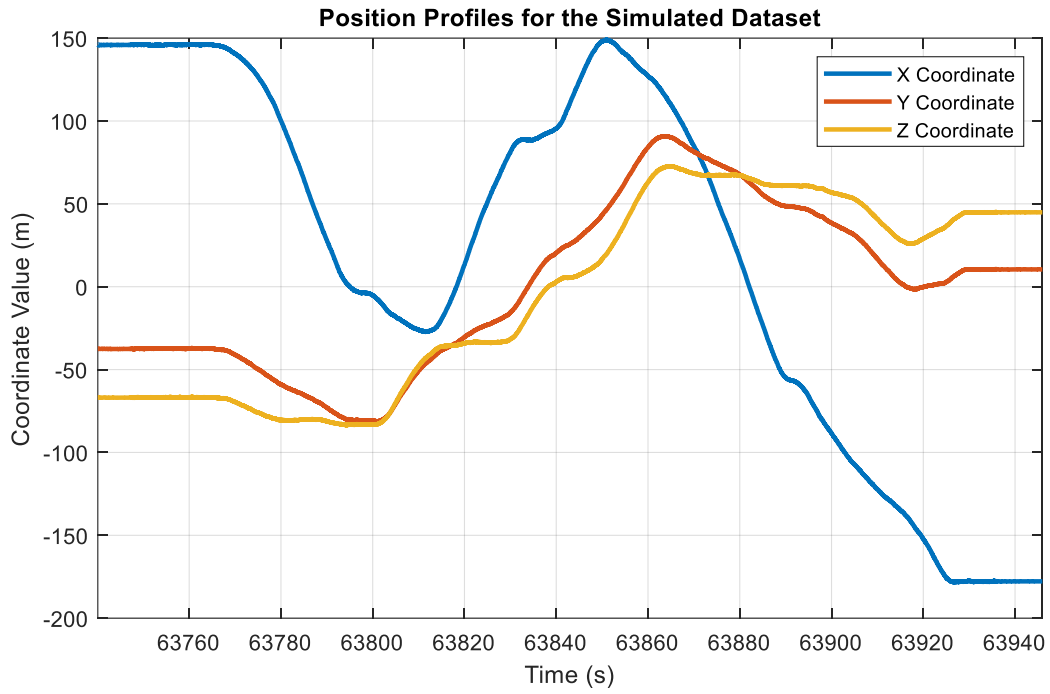


Figure 3.6.4: Position profiles for the estimated trajectory for the simulated data.

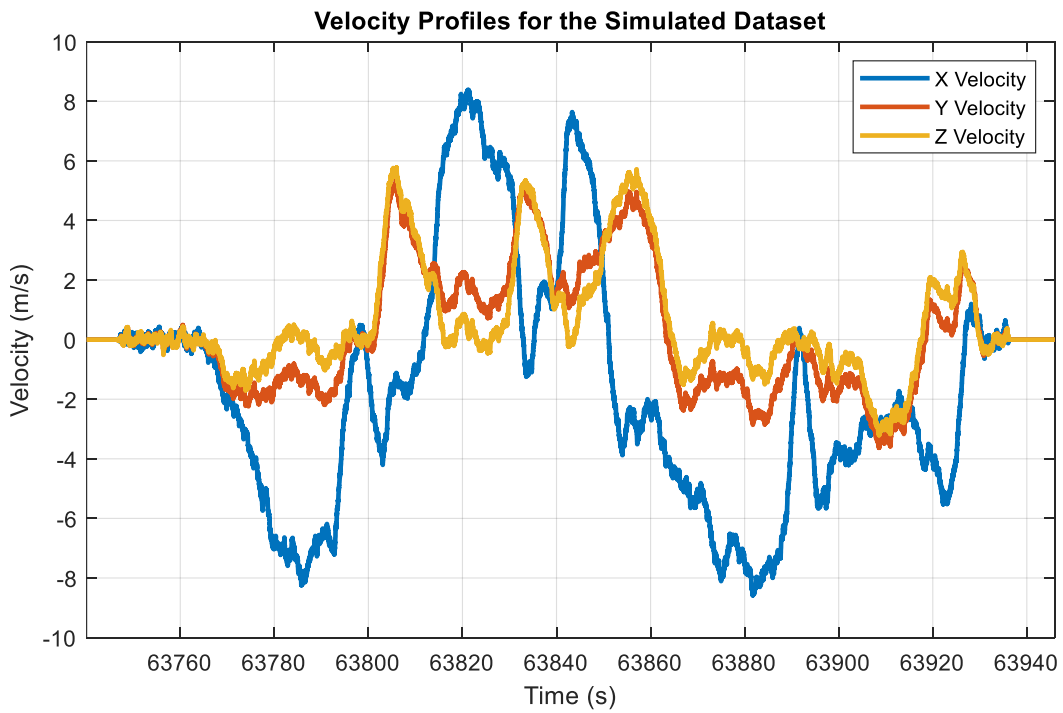


Figure 3.6.5: Velocity profiles for the simulated data.

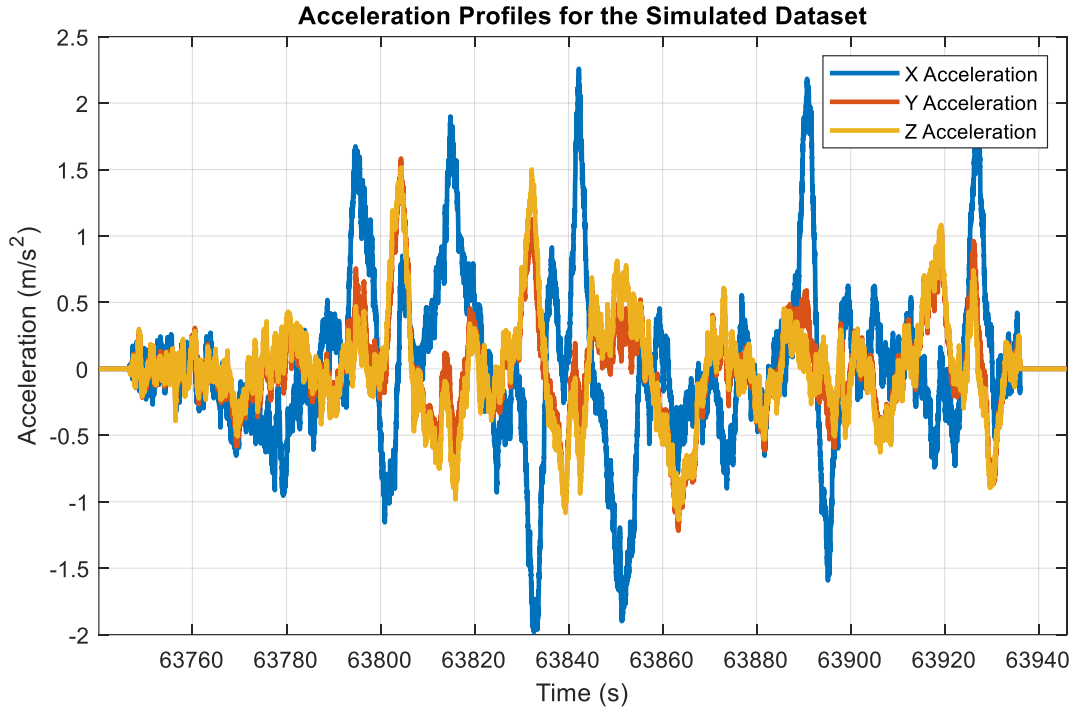


Figure 3.6.6: Acceleration profiles for the simulated data.

3.6.3 Estimated Residuals for the Simulated Dataset

While the constrained system state vector may be readily calculated via other approaches to applying system state constraints, the approach adopted by this research allows for a much more thorough analysis of the sources of random error in the Kalman Filter. This is largely accomplished through the calculation of the residual vectors corresponding to the predicted system state noise, process noise, and observation noise, along with their associated covariance matrices. Previously, these quantities were not able to be estimated.

To illustrate the estimation of both the three residual vectors and their associated covariance matrices, the standardized residual values are shown in Figure 3.6.7, Figure 3.6.8, and Figure 3.6.9 for the observation, process noise, and residuals of the predicted system state vector, respectively. The associated histograms of the standardized residuals are shown in Figure 3.6.10, Figure 3.6.11, and Figure 3.6.12.

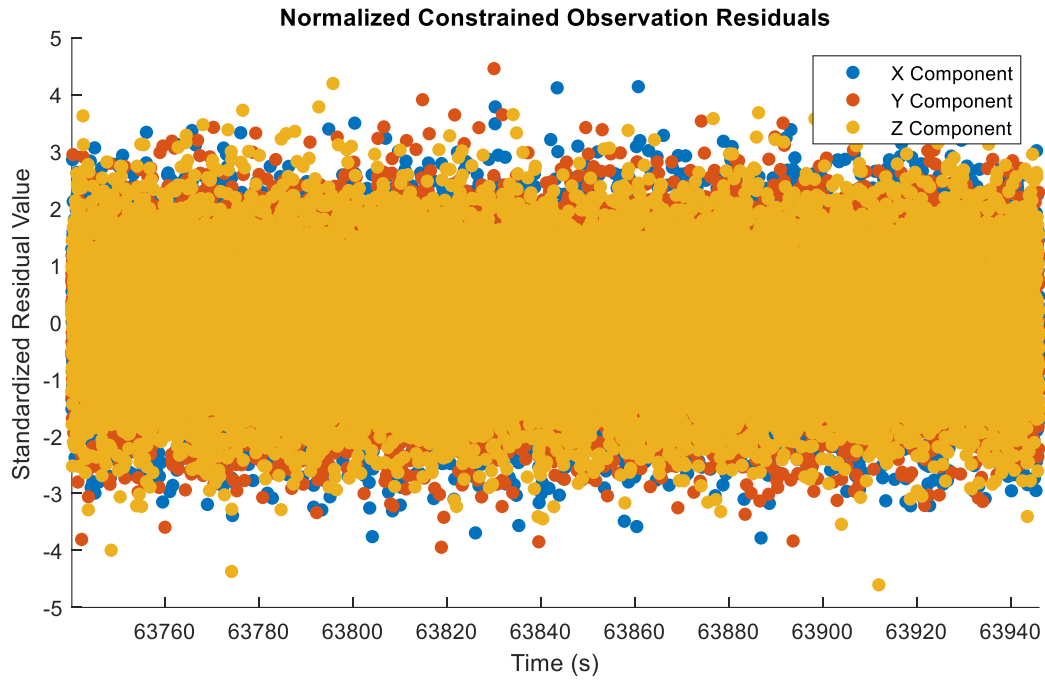


Figure 3.6.7: Standardized observation residuals for the simulated dataset after the application of the system state-constrained Kalman Filter.

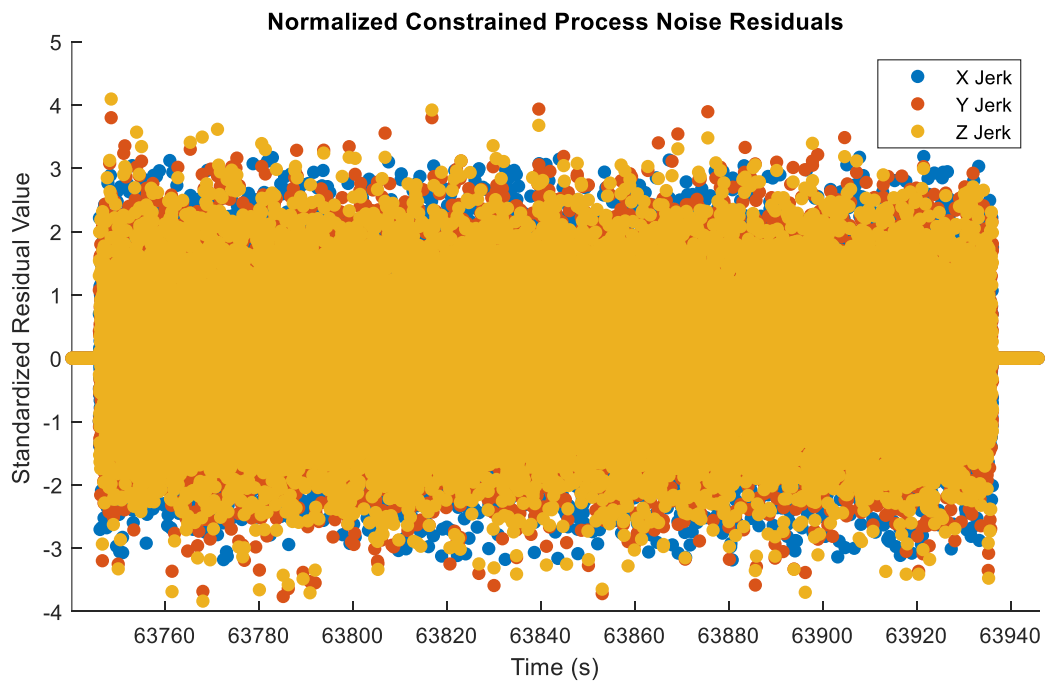


Figure 3.6.8: Standardized process noise residuals for the simulated dataset after the application of the system state-constrained Kalman Filter.

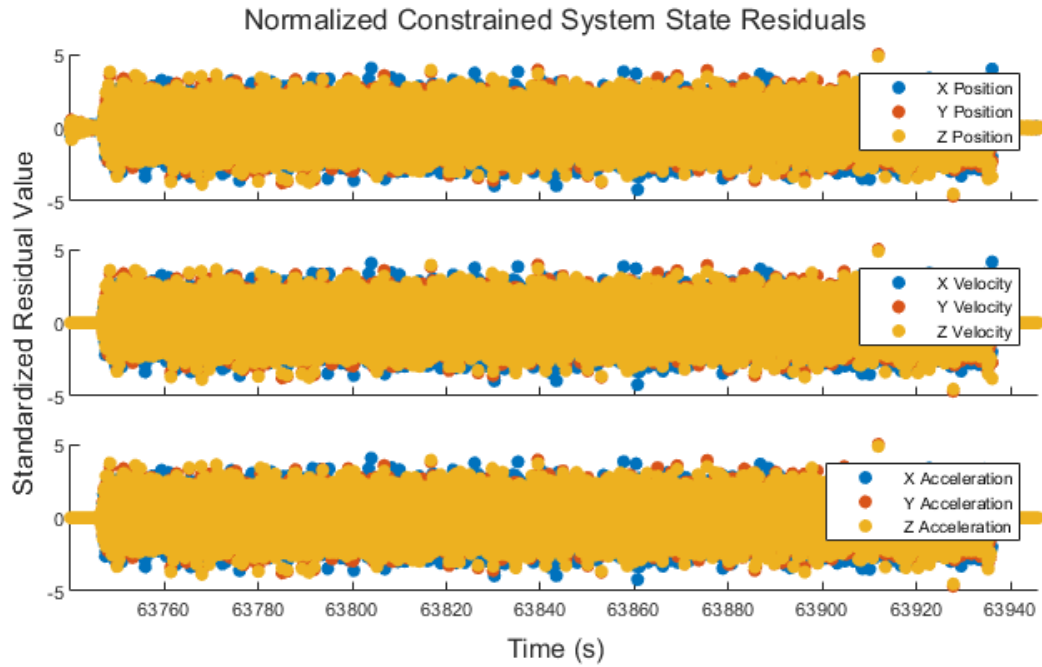


Figure 3.6.9: Standardized residuals of the predicted system state vector for the simulated dataset after the application of the system state-constrained Kalman Filter.

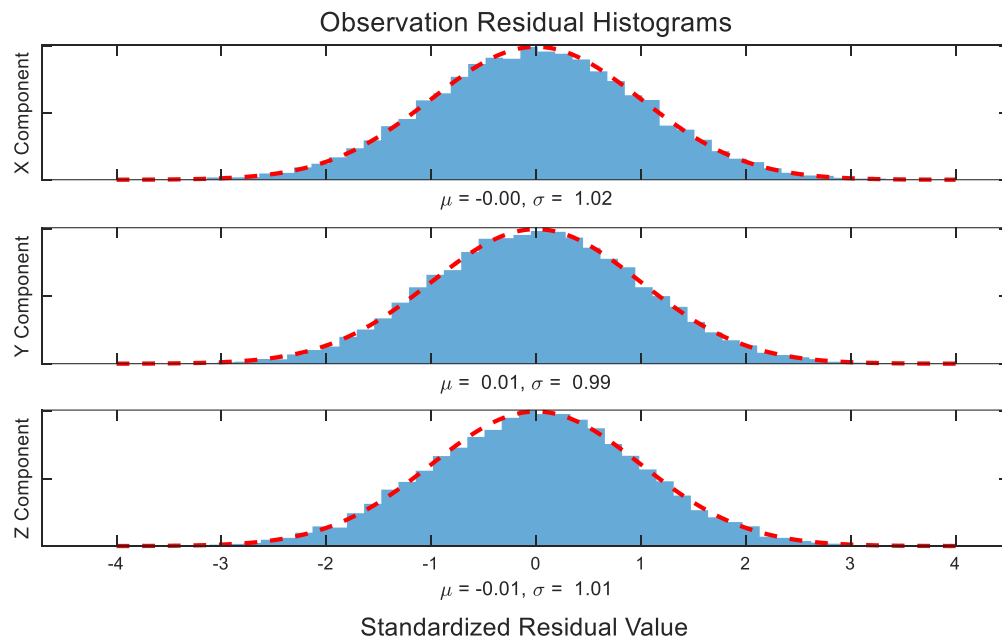


Figure 3.6.10: Histogram of the standardized observation residuals. Standard normal distribution shown in red for reference.

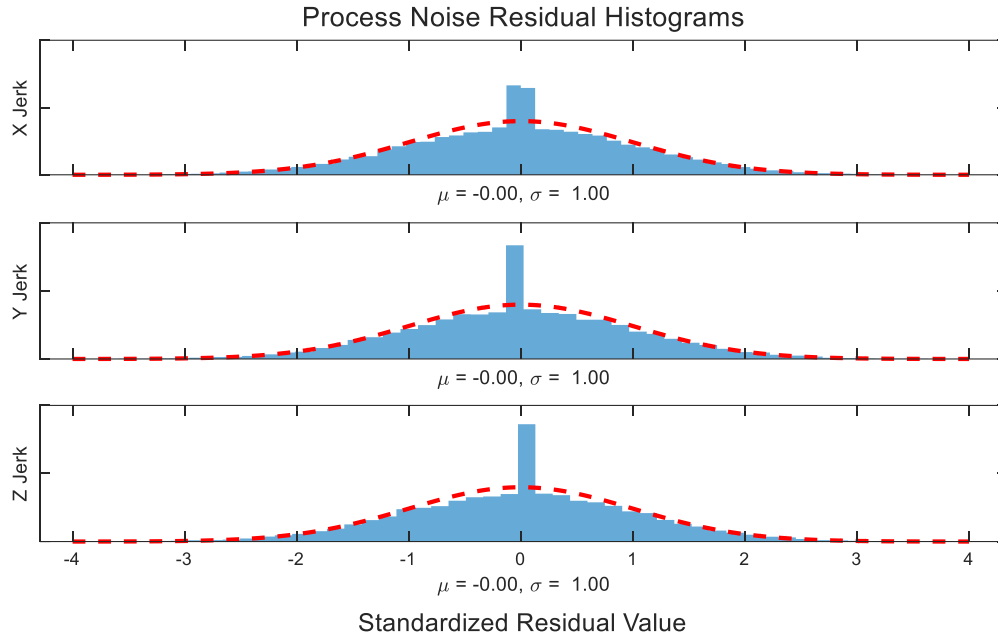


Figure 3.6.11: Histogram of the standardized process noise residuals. Standard normal distribution shown in red for reference. Note the sharp peak in the histograms near the 0 mark.

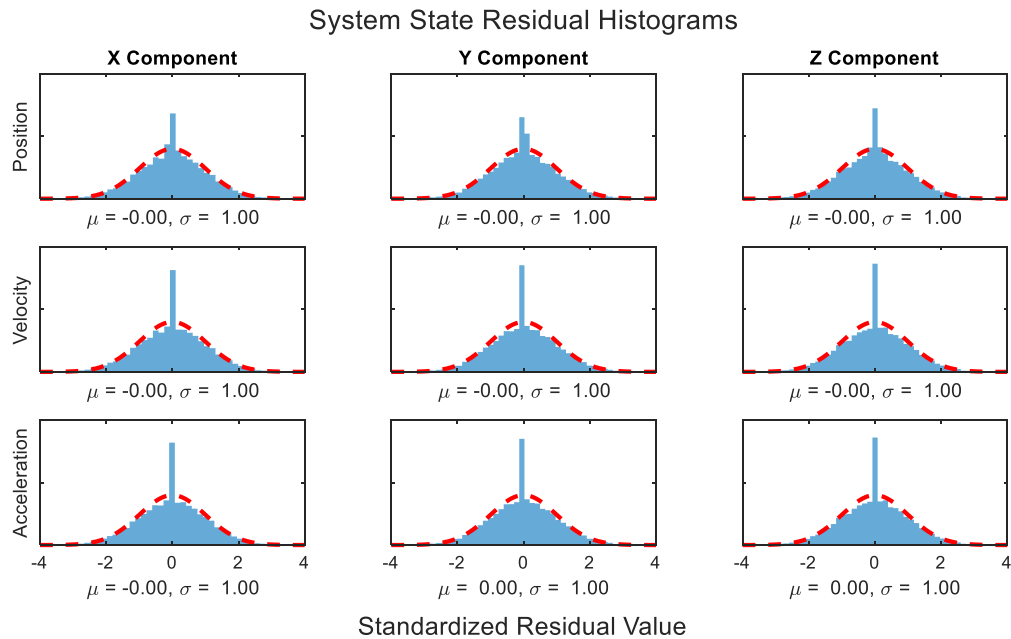


Figure 3.6.12: Histogram of the standardized residuals of the predicted system state vector. Standard normal distribution shown in red for reference. Note the sharp peak in the histograms near the 0 mark.

There are several behaviours apparent from the residual plots and histograms:

- Residuals:
 - The observation residuals are very well-behaved over the entire dataset, and their performance does not appear to be influenced by varying system dynamics. It is important to recognize that, since these observations were simulated with uniform quality through time, this behaviour is to be expected, and is not a basis for any broader conclusions on positioning observation quality.
 - The process noise and predicted system state vector residuals exhibit two types of behaviour:
 - Very small residual values when the system is stationary; and
 - Larger residual values when the system is in motion.

These two behaviours are not surprising. Since the process noise in this example represents the jerk, or the third-order time-derivative of the system's position, it stands to reason that the estimated system jerk would be almost exactly zero when the system is stationary, especially when the zero velocity/zero acceleration update is being applied. Since the system state is expected to be constant for the duration of the stationary period, it also makes sense that the residuals of the predicted system state vector would be smaller for this period. Additionally, the system state elements corresponding to the system's velocity and acceleration have residual values of zero during the stationary period, since the system state constraints force both the velocity and acceleration estimates to be zero.

- Histograms:
 - The observation residuals behave quite well over the entire dataset, and the histograms of their standardized values approximate the standard normal distribution quite well.
 - As before, the process noise and predicted system state residual histograms exhibit two types of behaviour:
 - The histograms of their standardized residuals very closely approximate the standard normal distribution for most of its domain; but
 - The histograms of their standardized residuals exhibit a sharp peak near the standardized residual value of zero.

This phenomenon exists because of the behaviours of these residual quantities during the stationary periods of the dataset. Since the residual values are very close to zero for these periods, all the observation epochs in the stationary period produce many standardized residual values near zero, which in turn produces the sharp jumps visible in Figure 3.6.11 and Figure 3.6.12.

3.6.4 Variance Component Estimation for the Simulated Dataset

Another strength of this research's formulation of the system state-constrained Kalman Filter is that it facilitates a more thorough VCE and analysis of estimated variance components. The first step in this process is evaluating the redundancy contributions of the three pseudo-observation vectors used in the Kalman Filter formulation. The estimated redundancy contributions of the predicted system state, observation, and process noise vectors are shown in Figure 3.6.13.

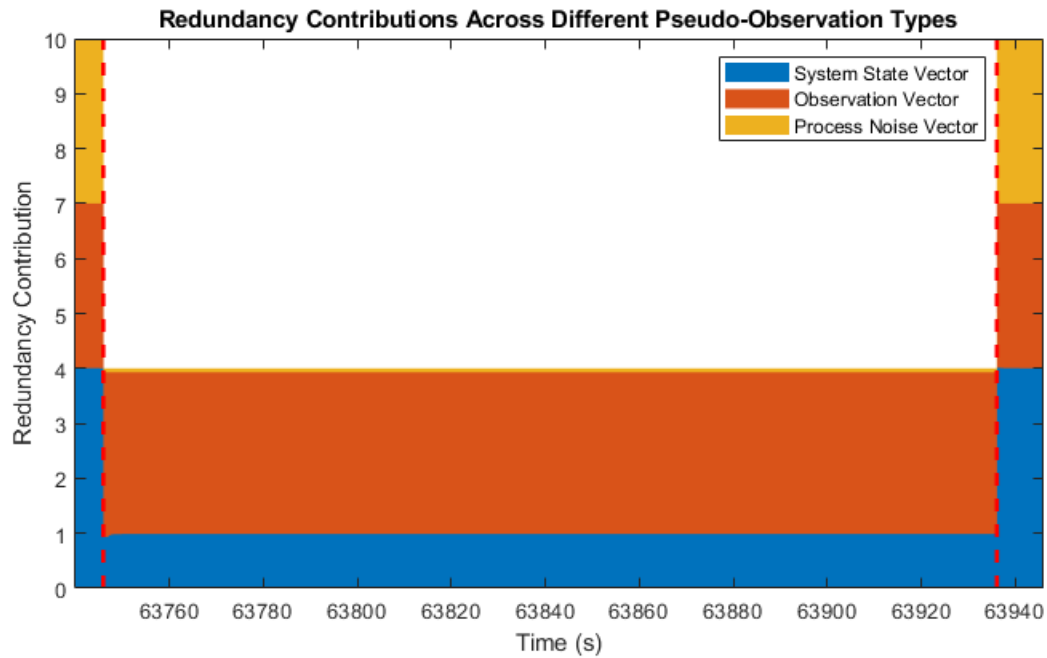


Figure 3.6.13: Estimated redundancy contributions for the predicted system state, observation, and process noise vectors in the system state-constrained Kalman Filter over the simulated dataset.

From Figure 3.6.13, it is clear that the total system redundancy is equal to 10 during the system's stationary periods and is equal to 4 when the system is mobile. This confirms the equation for the overall system redundancy in (3.97), given that:

- During stationary periods, there are 6 constraints on the system states enforcing that the system be stationary, and 1 constraint on the system states enforcing that the solution lies on the planar surface, for a total of 7 constraints. Additionally, each epoch has 3 observations, one for each position component, leading to a total of 10 pieces of redundant information whenever the system is stationary.
- During mobile periods, there is 1 constraint on the system state enforcing that the position be on the planar surface. Again, each epoch has 3 observations, leading to a total of 4 pieces of redundant information whenever the system is in motion.

In this research, we use the standard error components to describe the overall quality of a quantity in the Kalman Filter. The standard error components are defined to be the square root of the variance component factors calculated in (3.102), (3.103), (3.104), and (3.110).

The VCE algorithm used here is the mostly widely used practical approximation of the rigorous Helmert algorithm [Wang et al, 2010], in which the redundancy contributions and residual values are essential to conduct VCE. Significant difficulty can arise when estimating variance component factors for a single observation epoch. Referring to Figure 3.6.13, the epoch-wise redundancy contribution of each pseudo-observation vector is relatively low to be drawing statistically significant conclusions. To illustrate the issues caused by this, the epoch-wise estimated standard error component for the overall observation vector is shown in Figure 3.6.14.

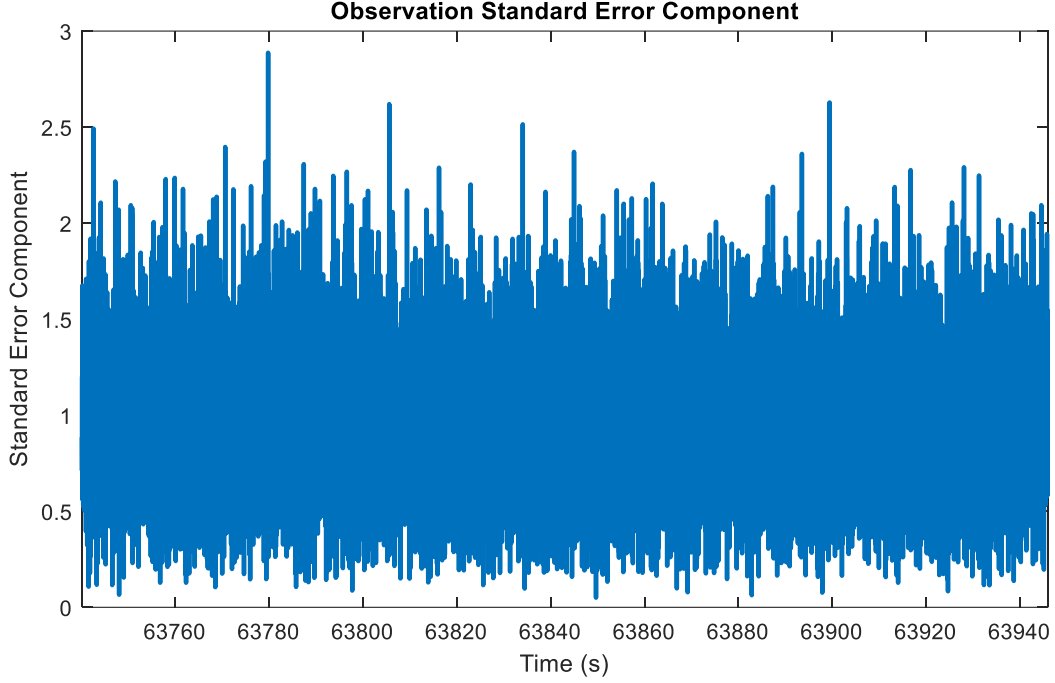


Figure 3.6.14: Epoch-wise estimated standard error components for the overall observation vector over the simulated dataset.

From Figure 3.6.14, it is clear that the estimated standard error component is very noisy over time. This negatively impacts the ability to draw significant conclusions from the standard error component's value at any given time. This is especially true for the process noise vector's redundancy contribution, which is almost zero whenever the system is mobile. To mitigate this issue, we may calculate the standard error components over a defined time window, instead of calculating them epoch-wise. Mathematically, this is accomplished via the equation

$$\hat{\sigma}_{v_i}^2(t_k, w) = \frac{\sum_{j=k-\lfloor \frac{w}{2} \rfloor}^{k+\lfloor \frac{w}{2} \rfloor} \mathbf{v}_i^T(t_j) \mathbf{D}_I^{-1}(t_j) \mathbf{v}_i(t_j)}{\sum_{j=k-\lfloor \frac{w}{2} \rfloor}^{k+\lfloor \frac{w}{2} \rfloor} r_{v_i}} \quad (3.116)$$

where w is the window size in number of epochs, and is defined as an odd integer value to ensure that the window is centred on the epoch being considered.

Figure 3.6.15, Figure 3.6.16, and Figure 3.6.17 show the estimated standard error components for the observations, process noise, and system states, respectively. Each plot shows the estimated standard error component evaluated over multiple different time windows for illustrative purposes.

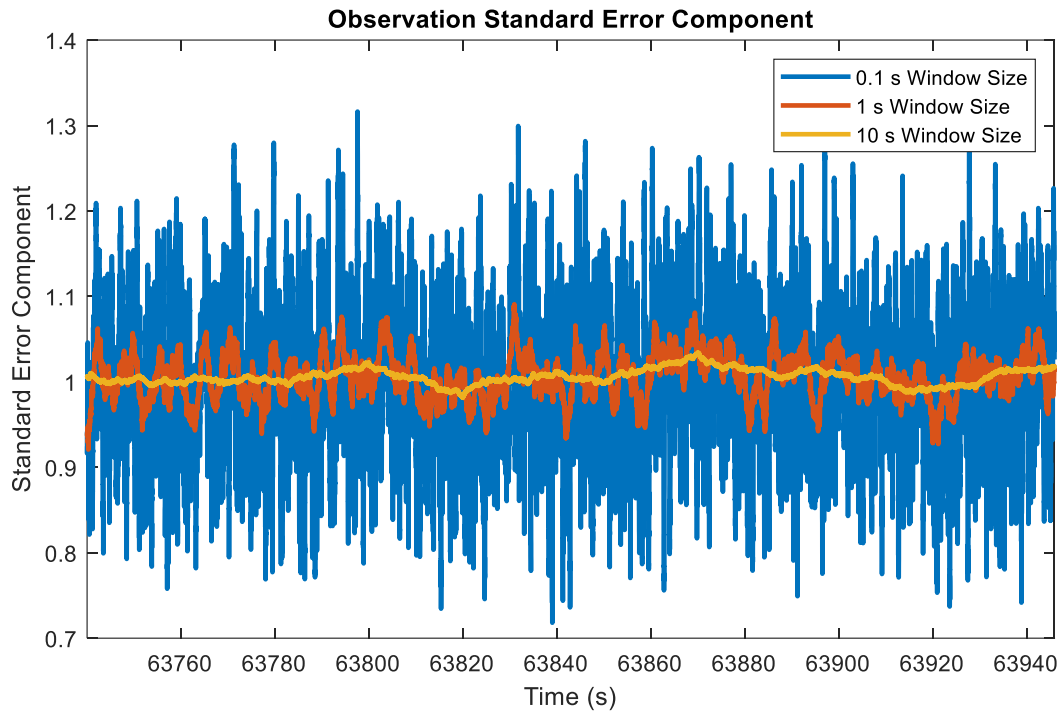


Figure 3.6.15: Estimated standard error components for the observation vector, evaluated using different time window sizes.

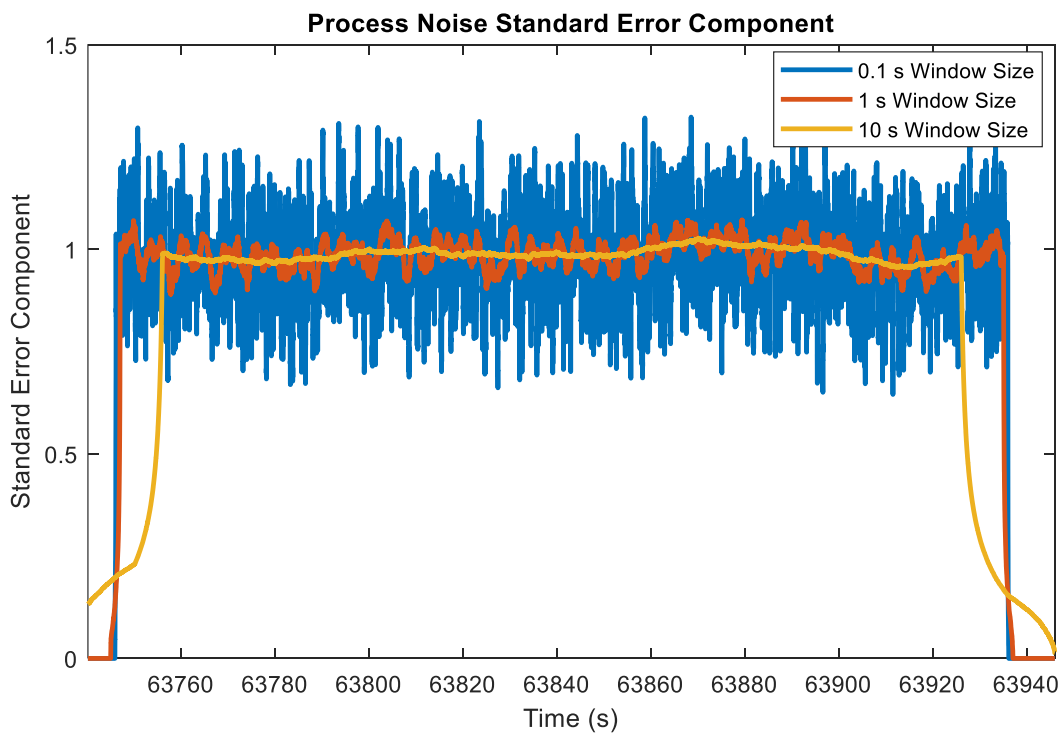


Figure 3.6.16: Estimated standard error components for the process noise vector, evaluated using different time window sizes.

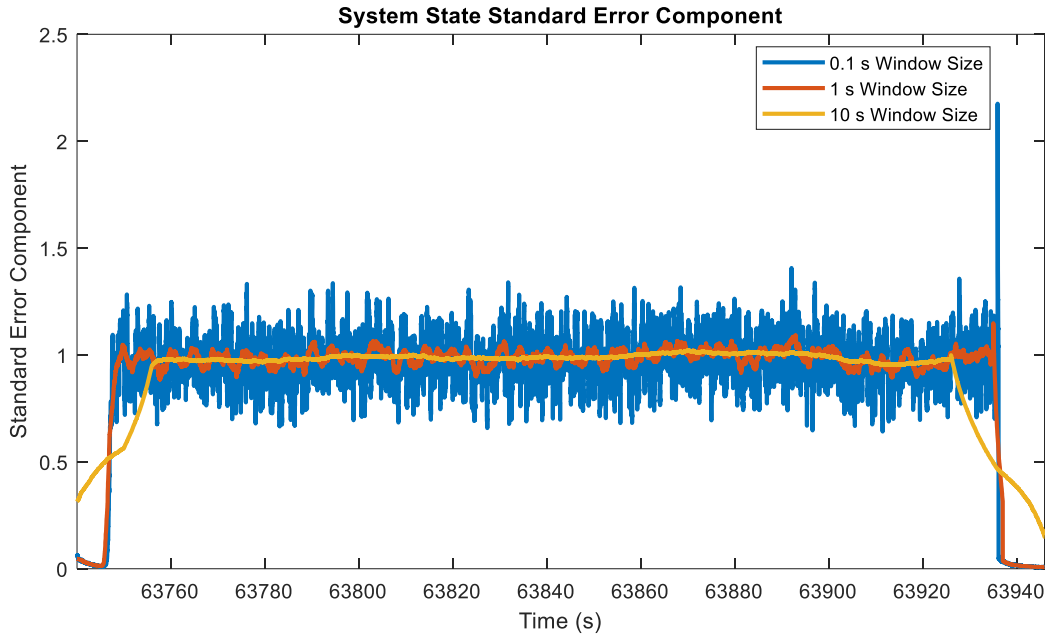


Figure 3.6.17: Estimated standard error components for the system state vector, evaluated using different time window sizes.

There are several patterns apparent in these standard error component plots:

- The process noise and system state standard error component estimates are significantly smaller during the stationary period than during the mobile period, and experience sudden changes when the system transitions from stationary to mobile or from mobile to stationary.
- Larger time window sizes improve the quality of the standard error component estimates, making them more reliable as more observation epochs are included in their estimation.
- Smaller time windows improve the time resolution of detecting where significant shifts in the standard error component estimates are happening. In Figure 3.6.16 and Figure 3.6.17, this may be seen at the transitions between the system being stationary and it being mobile. The smaller time windows yield a much more sudden change in the standard error component, which means that the point at which it changes is more readily able to be identified.

3.7 Summary

This chapter presents an alternative derivation of the system state-constrained discrete-time Kalman Filter that allows for the calculation of observation, predicted system state, and process noise residuals, which are essential for comprehensive error analysis such as residual analysis, reliability analysis, and Variance Component Estimation (VCE). The alternative derivation extends the logic of

[Wang, 1997], formulating the Kalman Filter problem as a least-squares problem with three sets of pseudo-observations that correspond to the three sources of random error in a Kalman Filter: the observation noise, the process noise, and the errors in the previous system state estimate. Its cost function is modified to include Lagrange multipliers of any system state equality constraints, and then is solved to determine the least-squares estimate of the system state vector and process noise vector, as well as residuals of the predicted system state vector, process noise vector, and observation vector, together with their associated covariance matrices. The estimated value of the system state is shown to be identical to the constrained estimates that can be determined from other popular methods for applying system state constraints.

This chapter also explores some of the practical considerations when conducting a comprehensive error analysis in a system state-constrained Kalman Filter, particularly when performing VCE. Of particular importance is that the redundancy contributions of each pseudo-observation vector tend to be relatively low for a single observation epoch. To improve the quality of the estimated standard error components, it is important to evaluate them over a time window. While larger time windows improve the quality of the estimated standard error components, this comes at the cost of the time resolution of the standard error component estimates.

4 The Position and Attitude (PosAtt) Engine

The work presented in this section has not been published on its own, but forms the basis for the software used to analyze kinematic datasets in the following publications:

Brunson, B., Wang, J., and Ma, W. (2024): Innovative Modeling of IMU Arrays Under the Generic Multi-Sensor Integration Strategy. *Sensors*, Vol. 24, Issue 23, 7754

Brunson, B., and Wang, J. (2023): Three Attitude Models and their Characterization in the Generic Multisensor Integration Strategy for Kinematic Positioning and Navigation. *The Journal of Global Positioning Systems*, 19(1), 79-98.

4.1 Preamble

Most of the research presented in this dissertation is focused on revisiting, modifying, and restructuring significant parts of the typical workflow for performing multi-sensor integrated navigation. It is very challenging to make this sort of large-scale change in an existing software package, and the needs of this research therefore required the development of a custom software package. This has been accomplished in York's Position/Attitude (PosAtt) Engine which was developed as part of this research in C/C++. The PosAtt Engine was developed under the Microsoft Visual Studio IDE (visualstudio.microsoft.com/). Matrix computations were handled by linking to the Eigen Matrix library (eigen.tuxfamily.org/), but otherwise all software functionality was built from the ground up in C/C++. The PosAtt engine consists of approximately 12,000 lines of C++ code and took two years to develop.

The PosAtt Engine is a post-processing kinematic positioning software that meets the following custom specifications:

- Directly modeling every individual observation for GPS/IMU integrated systems in a tightly-coupled Kalman Filter, including:
 - Double-differenced GPS L1 C/A code measurements;
 - Double-differenced GPS L1/L2 carrier phase measurements with ambiguity resolution;
 - Three-axis strapdown IMU gyroscope angular rate measurements; and
 - Three-axis strapdown IMU accelerometer specific force measurements.
- Implementing the system state-constrained positioning EKF to accommodate zero velocity/acceleration/angular rate updates as well as any system state equality constraints

necessary to avoid overparameterization when using different realizations of the system state vector.

- Performing epoch-wise comprehensive error analysis, including:
 - The covariance matrix of the system state estimates;
 - The predicted system state, process noise, and observation residual vectors, along with their associated covariance matrices;
 - The redundancy contributions of the system state, process noise, and observation vectors, as well as the redundancy contributions of different partitioning schemes for these vectors;
 - The quantities needed to estimate variance component factors for the system state, process noise, and observation vectors, as well as for the partitioned elements that have had their redundancy contributions estimated.
- Accommodating different attitude models to conduct a comparative analysis on their performance under the GMIS. This includes accepting different parameterizations of the system state attitude elements, and modifying system and measurement models accordingly.
- Accommodating an arbitrary number of IMU sensors in the PosAtt system, accepting custom values of the IMU lever arm vectors and sensor orientations. This includes individual modeling of all IMU sensor systematic errors for each constituent IMU sensor.

By following these specifications, the PosAtt Engine uses the Generic Multi-Sensor Integration Strategy (GMIS) to realize the goals of the research presented in this dissertation. The specific realization of the GMIS in this research is based upon its formalization in [Qian et al, 2015; Wang et al, 2015; Qian, 2017], which is summarized in Section 2.6.3, but the approach taken here differs in several key ways that are summarized in Section 4.2.

The PosAtt Engine currently uses GPS observables, with plans to more broadly include observables from other GNSS constellations in the future. This current limitation facilitated the development of a mature integration architecture without significantly sacrificing the quality of the PosAtt solution.

Following the description of the GMIS in Section 4.2, the overall structure of the PosAtt Engine is introduced in Section 4.3. Sections 4.4, 4.5, 4.6, and 4.7 provide documentation for each of the four main modules in the PosAtt Engine.

4.2 Generic Multi-Sensor Integration Strategy

This section provides the overall realization of the GMIS in the PosAtt Engine. More detailed information on the partitioning of the system state vector for attitude modeling and accommodating IMU arrays is provided in Chapters 5 and 6, respectively.

The specific realization of the GMIS in the PosAtt Engine is largely based on the GMIS's formalization in [Qian et al, 2015; Wang et al, 2015; Qian, 2017] as described in Section 2.6.3, with several key differences. The PosAtt Engine implemented its state vector as the following basic subvectors:

- The system position, $\mathbf{r}_{nb}^b$;
- The system velocity, $\mathbf{v}_{nb}^b$;
- The system acceleration, $\mathbf{a}_{nb}^b$;
- The system attitude, represented in the Euler model as α , β , and γ ;
- The time derivatives of the system attitude states, represented in the Euler model as $\dot{\alpha}$, $\dot{\beta}$, and $\dot{\gamma}$;
- The float ambiguities for the double-differenced L1 and L2 carrier phase measurements that have not yet had their ambiguity fixed; and
- IMU systematic error states, including the states to model bias and scale factor errors for each accelerometer and gyroscope.

As described in Section 2.6.3, the linear motion of the system is described using a constant linear acceleration model, as defined in (2.83), (2.84) and (2.85). The discrete form of these equations is

$$\mathbf{r}_{nb}^b(k+1) = \mathbf{r}_{nb}^b(k) + \mathbf{v}_{nb}^b(k)\Delta t + \frac{1}{2}\mathbf{a}_{nb}^b(k)\Delta t^2 + \frac{1}{6}\mathbf{j}(k)\Delta t^3 \quad (4.1)$$

$$\mathbf{v}_{nb}^b(k+1) = \mathbf{v}_{nb}^b(k) + \mathbf{a}_{nb}^b(k)\Delta t + \frac{1}{2}\mathbf{j}(k)\Delta t^2 \quad (4.2)$$

and

$$\mathbf{a}_{nb}^b(k+1) = \mathbf{a}_{nb}^b(k) + \mathbf{j}(k)\Delta t \quad (4.3)$$

where $\mathbf{j}$ is the system jerk vector, which represents the process noise in the linear motion of the system.

The PosAtt Engine differs in its formalization of the GMIS from that of [Qian, 2017] in its modeling of attitude dynamics. Instead of including the system's angular rate in the system state vector,

the PosAtt Engine uses the attitude parameters' time derivatives to model the angular motion of the system. Assuming a constant time-derivative of the attitude parameters between successive observation epochs, the discrete forms of the differential equations defining the attitude system model are

$$\alpha(k+1) = \alpha(k) + \dot{\alpha}(k)\Delta t + \frac{1}{2}\ddot{\alpha}(k)\Delta t^2 \quad (4.4)$$

$$\dot{\alpha}(k+1) = \dot{\alpha}(k) + \ddot{\alpha}(k)\Delta t \quad (4.5)$$

$$\beta(k+1) = \beta(k) + \dot{\beta}(k)\Delta t + \frac{1}{2}\ddot{\beta}(k)\Delta t^2 \quad (4.6)$$

$$\dot{\beta}(k+1) = \dot{\beta}(k) + \ddot{\beta}(k)\Delta t \quad (4.7)$$

$$\gamma(k+1) = \gamma(k) + \dot{\gamma}(k)\Delta t + \frac{1}{2}\ddot{\gamma}(k)\Delta t^2 \quad (4.8)$$

and

$$\dot{\gamma}(k+1) = \dot{\gamma}(k) + \ddot{\gamma}(k)\Delta t \quad (4.9)$$

Where $\dot{\theta}$ and $\ddot{\theta}$ are the first and second-order time derivatives of a random variable θ . In (4.4) – (4.9), the second-order time derivatives represent the process noise.

For each of the GPS float ambiguity and the IMU systematic error states, the PosAtt Engine uses a random-constant variable to construct its shaping filter over time.

Together, the equations from (4.1) to (4.9) define the core of the system model in the PosAtt Engine, and are used to populate both the system state transition matrix and process noise transition matrix.

The process noise covariance matrix $\mathbf{Q}$ is defined as follows

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_p & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{GPS} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{IMU} \end{bmatrix} \quad (4.10)$$

where $\mathbf{Q}_p$ describes the covariance matrix of the process noise elements relating to the linear motion of the system (i.e. system jerk) and is a 3x3 diagonal matrix; $\mathbf{Q}_a$ is the covariance matrix of the process noise elements relating to the angular motion of the system (i.e. angular second-derivatives) and is a 3x3 diagonal matrix for the Roll-Pitch-Heading attitude model; $\mathbf{Q}_{GPS}$ is the process noise covariance

matrix for the GPS float ambiguity estimates' random-constant shaping filter and is a diagonal matrix; and $\mathbf{Q}_{IMU}$ is the process noise covariance matrix for the IMU systematic errors' random-constant shaping filter and is a diagonal matrix.

With the GPS and IMU measurements, the PosAtt Engine uses the double-differenced GPS L1 C/A code and L1 and L2 carrier phase observations, plus the 3-DOF IMU accelerometer and gyroscope observations. These measurement models are described in Sections 2.4.1.3, 2.4.2.4.1, and 2.4.2.4.2, respectively.

Equations (2.41) and (2.42), (2.46), and (2.47) constitute the GPS and IMU measurement models for the data fusion in Kalman Filtering, and are used to populate the design matrix of the observation models.

The observation covariance matrix for the GPS double-differenced C/A code observations is defined to be

$$\mathbf{R}_{GPS,C} = \begin{bmatrix} 4\sigma_{GPS,C}^2 & 2\sigma_{GPS,C}^2 & \cdots & 2\sigma_{GPS,C}^2 \\ 2\sigma_{GPS,C}^2 & 4\sigma_{GPS,C}^2 & \cdots & 2\sigma_{GPS,C}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 2\sigma_{GPS,C}^2 & 2\sigma_{GPS,C}^2 & \cdots & 4\sigma_{GPS,C}^2 \end{bmatrix} \quad (4.11)$$

where $\sigma_{GPS,C}^2$ is the variance of the undifferenced GPS C/A code observations.

Similarly, the observation matrix for the GPS double-differenced L1 and L2 carrier phase observations is defined to be

$$\mathbf{R}_{GPS,L1} = \begin{bmatrix} 4\sigma_{GPS,L1}^2 & 2\sigma_{GPS,L1}^2 & \cdots & 2\sigma_{GPS,L1}^2 \\ 2\sigma_{GPS,L1}^2 & 4\sigma_{GPS,L1}^2 & \cdots & 2\sigma_{GPS,L1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 2\sigma_{GPS,L1}^2 & 2\sigma_{GPS,L1}^2 & \cdots & 4\sigma_{GPS,L1}^2 \end{bmatrix} \quad (4.12)$$

$$\mathbf{R}_{GPS,L2} = \begin{bmatrix} 4\sigma_{GPS,L2}^2 & 2\sigma_{GPS,L2}^2 & \cdots & 2\sigma_{GPS,L2}^2 \\ 2\sigma_{GPS,L2}^2 & 4\sigma_{GPS,L2}^2 & \cdots & 2\sigma_{GPS,L2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 2\sigma_{GPS,L2}^2 & 2\sigma_{GPS,L2}^2 & \cdots & 4\sigma_{GPS,L2}^2 \end{bmatrix} \quad (4.13)$$

where $\sigma_{GPS,L1}^2$ and $\sigma_{GPS,L2}^2$ are the variances of the undifferenced GPS L1 and L2 carrier phase observations, respectively. In this research, $\sigma_{GPS,C} = 0.5 \text{ m}$, $\sigma_{GPS,L1} = 5 \text{ mm}$, and $\sigma_{GPS,L2} = 5 \text{ mm}$.

The observation covariance matrix for the IMU gyroscope and accelerometer observations is defined to be

$$R_{IMU} = \begin{bmatrix} R_g & \mathbf{0} \\ \mathbf{0} & R_s \end{bmatrix} \quad (4.14)$$

where R_g is the 3x3 diagonal covariance matrix for the gyroscope angular rate observations and R_s is the 3x3 diagonal covariance matrix for the accelerometer specific force observations. It is worth noting that these matrices are assumed to be diagonal, but that cross-coupling errors may introduce correlation between the IMU observations. In this research, $\sigma_g = 40'/s$ and $\sigma_s = 0.1m/s^2$.

Assuming the presence of GPS C/A, L1 carrier phase, L2 carrier phase, gyroscope, and accelerometer observations at an observation epoch, the overall observation covariance matrix is determined from (4.11) to (4.14) to be

$$R = \begin{bmatrix} R_{IMU} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_{GPS,C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R_{GPS,L1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & R_{GPS,L2} \end{bmatrix} \quad (4.15)$$

4.3 Software Structure

The PosAtt Engine contains four main modules:

- The Data Input Module,
- The Initialization Module,
- The Positioning Module, and
- The Data Output Module.

Each of these modules undertakes multiple processing subtasks. The general flow of interaction between these modules is shown in Figure 4.3.1.

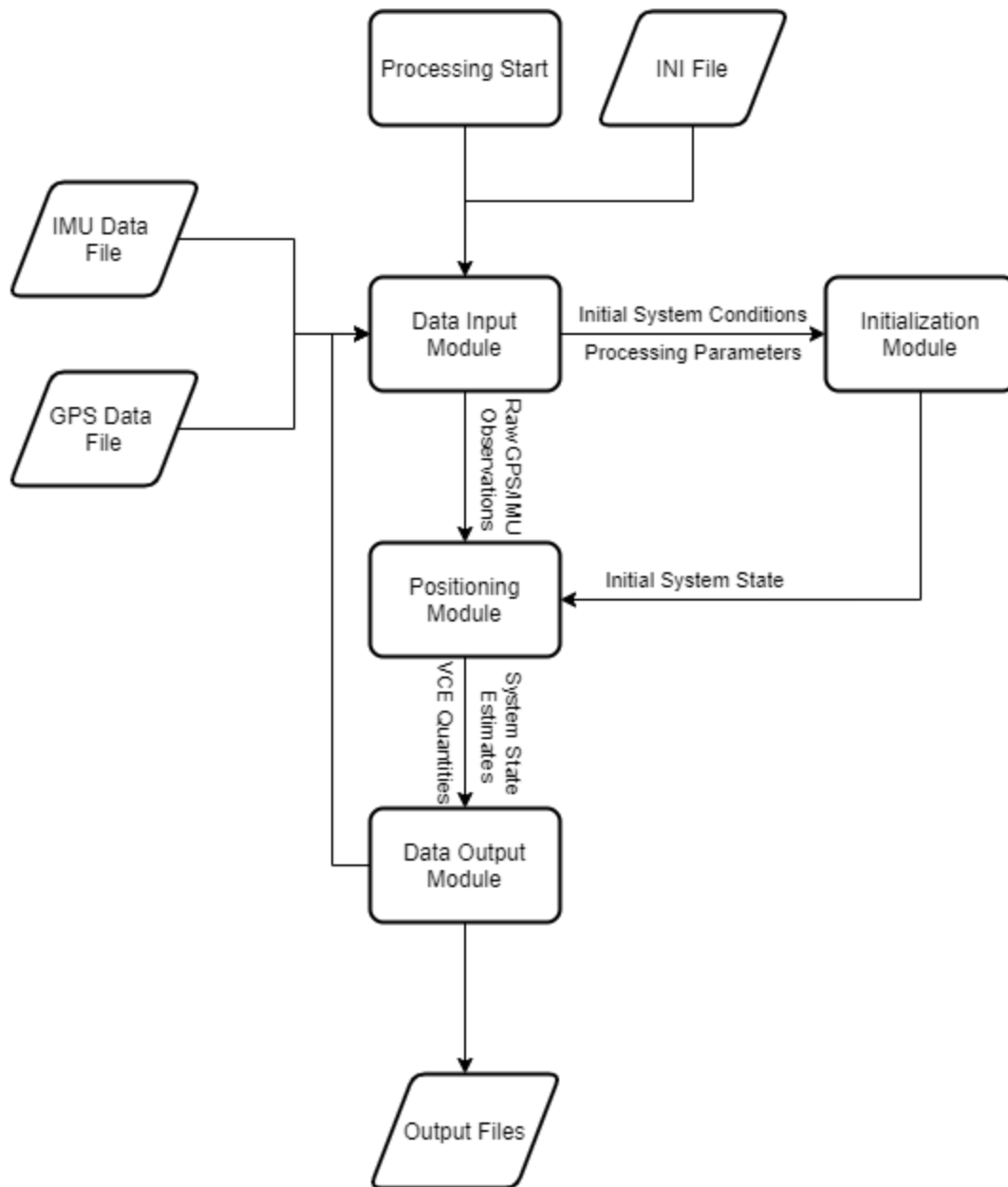


Figure 4.3.1: General workflow describing the interactions between the four primary modules of the PosAtt Engine.

4.4 Data Input Module

The Data Input Module is composed of three components:

- INI File Input, which reads the set of processing parameters and initial system conditions that are defined in the INI file;
- IMU Data File Input, which reads IMU data from a data file epoch-wise and arranges it into appropriate data structures; and

- GPS Data File Input, which reads GPS data from a data file epoch-wise and arranges it into appropriate data structures.

These three components are documented in more detail in Sections 4.4.1, 4.4.2, and 4.4.3.

4.4.1 INI File Input

The PosAtt Engine initialization file (i.e. INI file) contains specifications of the processing parameters and the parameters governing the initial system conditions. The parameters in the INI file are divided into blocks depending upon their function, which are summarized below:

- Software Mode Block: specifies the mode that the software will run in. There are currently two modes: NAV for kinematic positioning, and RINEX2TXT to perform file conversion tasks for a RINEX file.
- File Input Block: specifies the input files for processing and some parameters specific to those files. The parameters Include:
 - GPS File Parameters
 - GPS Rover File: the path to the file containing the GPS observations for the GPS receiver that is in motion.
 - GPS Base File: the path to the file containing the GPS observations for the GPS receiver that is not in motion.
 - IMU File Parameters: to accommodate multiple IMU files, all of these parameters may be defined multiple times, which adds their values to arrays defining the parameters for every IMU in an IMU array
 - IMU File: the path to the file containing the IMU observations
 - Lever Arm: the lever arm vector defining the translation in vehicle frame coordinates between the origin of the IMU body frame and the origin of the vehicle frame.
 - IMU Orientation: defines the orientation of the IMU axes relative to the vehicle frame. For example, a value of “FRU” indicates that the IMU is arranged in a “forward-right-up” orientation for its x, y, and z axes, respectively.
 - Gyro Direction: defines whether the IMU gyroscopes measure angular rates positively in the clockwise or the counterclockwise directions.
- File Output Block: specifies the paths of the output files:
 - Output Path: defines the folder that all output files will be generated in.

- File Output Options: specifies what output files will be generated. The files that are not generated do not have their corresponding quantities estimated in the Positioning Module, to save processing time. A user can choose to
 - Calculate State Residuals: this controls whether residuals of the predicted system state vector are estimated and output.
 - Calculate Process Noise Residuals: this controls whether process noise residuals are estimated and output.
 - Calculate Observation Residuals: this controls whether observation residuals are estimated and output.
 - Calculate Residual Covariance: this controls whether the covariance matrices are estimated for each residual vector, and whether this information is output alongside the residual values.
 - Calculate Variance Components: this controls whether the variance components are estimated for each residual vector, and whether this information is output.
- Base Station Position Block
 - Base Station Position: provides the X, Y, and Z coordinates of the base station in the ECEF coordinate frame.
 - Starting Time: defines the desired starting time for processing rover data.
 - Ending Time: defines the desired ending time for processing rover data.
- Static Period Block: defines static periods in the dataset for the event application of the zero velocity/acceleration/angular rate updates. Since there can be multiple static periods in the same dataset, each parameter in this block can be defined multiple times in the INI file.
 - Static Start: defines the start of a static period in the dataset.
 - Static End: defines the end of a static period in the dataset.
- Initial Position/Attitude Block
 - Rover Station Position: defines the initial X, Y, and Z coordinates of the rover station in the ECEF coordinate frame.
 - Initial Heading: defines the initial heading of the vehicle.
- Process Noise Variance Definition Block
 - Linear Jerk: defines the expected variance of the system jerk. Separate values for the E, N, and U components in the navigation frame.

- Angular Acceleration: defines the expected variance of the second time-derivative of the attitude parameters.
- Ambiguity Drift: defines the expected variance of the first time-derivative of the float ambiguity estimates.
- Gyro Bias Drift: defines the expected variance of the first time-derivative of the gyroscope bias parameters.
- Gyro Scale Drift: defines the expected variance of the first time-derivative of the gyroscope scale factor parameters.
- Accelerometer Bias Drift: defines the expected variance of the first time-derivative of the accelerometer bias parameters.
- Accelerometer Scale Drift: defines the expected variance of the first time-derivative of the accelerometer scale factor parameters.
- Observation Variance Definition Block
 - L1 Factor: defines the variance of the L1 double-differenced carrier phase observations.
 - L2 Factor: defines the variance of the L2 double-differenced carrier phase observations.
 - C/A Factor: defines the variance of the double-differenced L1 C/A code observations.
 - Accelerometer Factor: defines the variance of the accelerometer specific force observations.
 - Gyroscope Factor: defines the variance of the gyroscope angular rate observations.
 - X Factor: defines the variance factor being applied to the predicted system state estimate. Defaults to a value of 1.
- Attitude Model Selection Block
 - Attitude Model: specifies the attitude model that is being used for the current processing task. There are three options: EULER for the RPH model; MATRIX for the DCM model; and QUATERNION for the quaternion model.
- Processing Restrictions Block: defines certain rules regarding how the usage of data are restricted:
 - Cutoff Angle: defines the cutoff angle for GPS raw data, which means that any satellites with an elevation angle below the cutoff angle will have their associated observations excluded.
 - Process L1: defines whether or not L1 carrier phase observations are included.
 - Process L2: defines whether or not L2 carrier phase observations are included.

- Process Gyro: defines whether or not gyroscope angular rate observations are included.
- Process Accelerometer: defines whether or not accelerometer specific force observations are included.

4.4.2 IMU Data File Input

The PosAtt Engine's IMU Data File Input module defines the IMUSensor object, which reads the raw data in from an IMU data file one epoch at a time and stores it in variables that describe the current observation time, the current accelerometer measurements, and the current gyroscope measurements. The IMUSensor object also stores information about the IMU systematic error parameters/effects, such as the sensor rotation relative to the vehicle frame, the sensor lever arm vector, and accelerometer/gyroscope bias and scale factor errors.

4.4.3 GPS Data File Input

The PosAtt Engine's GPS Data File Input module defines the GPSSensor object, which reads the raw data in from a GPS data file one epoch at a time and stores it in variables that describe the current observation time and double differenced GPS observations, along with the satellites referred to by them. The GPSSensor object also contains the GPS processing parameters, such as the cutoff angle and whether or not to process L1/L2 carrier phase observations.

4.5 Initialization Module

There are several pre-processing submodules developed in the PosAtt Engine to conduct necessary pre-processing tasks. These include the following tasks:

- Selecting the GPS "base" satellites for calculating double-differenced GPS observations so as to maximize the time coverage/average elevation angle of any base satellite.
- Performing coarse alignment using IMU data from the initial stationary period of the dataset to initialize the system's roll and pitch angles reliably.

The selection of the GPS "base" satellites is documented in Section 4.5.1 and the coarse calibration process is documented in Section 4.5.2.

4.5.1 Selecting the GPS “Base” Satellites

When selecting GPS “base” satellites for constructing the double-differenced GPS observations, there are two possible scenarios:

- i. There are one or more satellites that are available for the entire observation period of the dataset. In this scenario, all satellites that have data available over all observation epochs are considered candidates to be the “base” satellite, and the candidate satellite with the highest average elevation angle is selected as the “base” satellite. The elevation angle is used as the selection criterion since satellites at low elevation angles can be expected to have larger tropospheric and ionospheric delay affecting their observations.
- ii. There are no satellites that are available for the entire observation period of the dataset. In this scenario, the dataset is scanned from its starting time, and the satellite with the longest availability is selected as the first “base” satellite. This process is then repeated from the time the “base” satellite is dropped, selecting new “base” satellites until there is a “base” satellite defined at every GPS epoch over the dataset.

4.5.2 Coarse Alignment

Coarse alignment is also referred to as coarse leveling, which is performed to initialize the system pitch and roll commonly during the stationary period at the start of the dataset. Since the vehicle is motionless for this period, the main specific force vector it experiences is resulting from the local gravity vector, and the main angular rate vector it experiences comes from Earth’s self rotation. So, the local vertical may be reliably determined for this period, allowing for the consistent estimation of system roll and pitch. This research adopts the Direct Method coarse alignment procedure from [Zhao et al, 2011], and the basis of this method is described below.

The gravity vector in the navigation frame is related to the gravity vector observed by the accelerometers in the IMU body frame by

$$\mathbf{g}^n = \mathbf{C}_b^n \mathbf{g}^b \quad (4.16)$$

The angular rate vector of Earth in the navigation frame is related to the angular rate vector of Earth observed by the gyroscopes in the IMU body frame by the equation

$$\boldsymbol{\omega}_{ie}^n = \mathbf{C}_b^n \boldsymbol{\omega}_{ie}^b \quad (4.17)$$

wherein C_b^n is the direction cosine matrix from the body frame to the navigation frame.

Additionally, the quantity $g^n \times \omega_{ie}^n$ may be calculated alongside $g^b \times \omega_{ie}^b$, and these vectors are related as follows

$$g^n \times \omega_{ie}^n = C_b^n g^b \times \omega_{ie}^b \quad (4.18)$$

Combining the relationships in (4.16), (4.17) and (4.18) yields a 3x3 matrix

$$[g^n \quad \omega_{ie}^n \quad g^n \times \omega_{ie}^n] = C_b^n [g^b \quad \omega_{ie}^b \quad g^b \times \omega_{ie}^b] \quad (4.19)$$

which may be rearranged to estimate C_b^n as

$$C_b^n = [g^n \quad \omega_{ie}^n \quad g^n \times \omega_{ie}^n] [g^b \quad \omega_{ie}^b \quad g^b \times \omega_{ie}^b]^{-1} \quad (4.20)$$

This estimate of the rotation matrix between the body and navigation frames may be used to accurately determine the system's roll and pitch angles for the duration of the stationary period.

Unfortunately, it is not realistic to directly use the measurements from a MEMS IMU (even a tactical MEMS IMU) to realize fine alignment (i.e. gyrocompassing) over a static period. Because of this, the initial system heading comes from an initial approximation defined in the INI file, while the system roll and pitch are determined via this coarse alignment procedure.

4.6 Positioning Module

This module follows the general flow of the GMIS as defined in Section 2.6.3, Figure 2.6.10, with several modifications to fit the specific sensor arrangement used in this research and to accommodate the enforcement of system state equality constraints. The specific workflow used in the positioning module is shown in Figure 4.6.1.

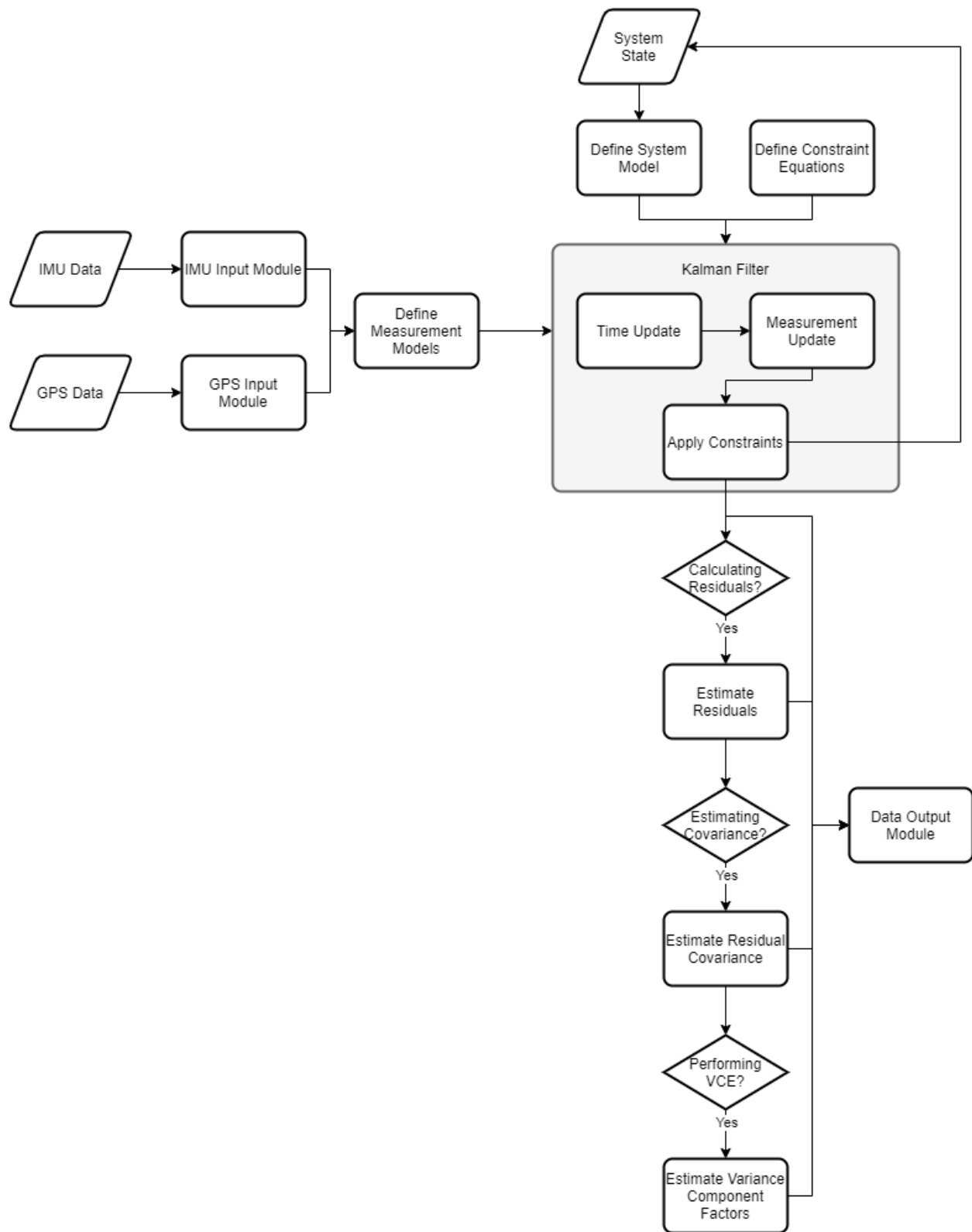


Figure 4.6.1: Workflow describing the processing module of the PosAtt Engine.

4.7 Data Output Module

This module extracts the salient information from the KalmanFilter object and outputs it to data files located in the specified output folder, which includes:

- **Attitude.txt:** This includes estimates of each attitude parameter, its first-order time-derivative, and their associated estimated standard deviations.
- **GPS_Ambiguity_Parameters_L1.txt:** This includes the estimated float/integer ambiguity parameters for the L1 carrier phase measurements.
- **GPS_Ambiguity_Parameters_L2.txt:** This includes the estimated float/integer ambiguity parameters for the L2 carrier phase measurements.
- **GPS_observations.txt:** This includes the raw GPS data, modified into a more readable format.
- **Observation_Variance_Factors.txt:** This includes the information required to calculate the standard errors of unit weight for the observation vector and several of its defined sub-vectors. This information includes the quadratic form of the residual vector in the numerator of (3.4.27), as well as the contribution to the overall system redundancy in the denominator of (3.4.27).
- **IMU_Parameters.txt:** This includes estimates of the modeled IMU systematic errors, along with their associated standard deviations.
- **Innovations_L1_Code.txt:** This includes the estimated system innovation vector for the L1 C/A measurements.
- **Innovations_L1_Phase.txt:** This includes the estimated system innovation vector for the L1 carrier phase measurements.
- **Innovations_L2_Phase.txt:** This includes the estimated system innovation vector for the L2 carrier phase measurements.
- **Predicted_Attitude.txt:** This includes the predicted system attitude, as determined in the Kalman Filter Time Update.
- **Predicted_Position.txt:** This includes the predicted system position, velocity, and acceleration, as determined in the Kalman Filter Time Update.
- **Process_Noise_Variance_Factors.txt:** This includes the information required to calculate the standard errors of unit weight for the process noise vector and several of its defined sub-vectors. This information includes the quadratic form of the residual vector in the numerator of (3.4.27) and the contribution to the overall system redundancy in the denominator of (3.4.27).

- **Residuals_IMU.txt:** This includes the IMU accelerometer/gyroscope measurement residuals, alongside their associated standard deviations, if they are being calculated.
- **Residuals_L1_Code.txt:** This includes the L1 C/A code measurement residuals, alongside their associated standard deviations, if they are being calculated.
- **Residuals_L1_Phase.txt:** This includes the L1 carrier phase measurement residuals, alongside their associated standard deviations, if they are being calculated.
- **Residuals_L2_Phase.txt:** This includes the L2 carrier phase measurement residuals, alongside their associated standard deviations, if they are being calculated.
- **Residuals_Process_Noise.txt:** This includes the estimated elements of the process noise residual vector, alongside their associated standard deviations, if they are being calculated.
- **Residuals_System_State.txt:** This includes the estimated elements of the predicted system state residual vector, alongside their associated standard deviations, if they are being calculated.
- **Solution.txt:** This includes the estimated trajectory, including its position, velocity, and acceleration, along with their associated standard deviations.

4.8 Summary

This chapter details the software package developed for this research, including:

- The specific details of our realization of the GMIS;
- The processing and pre-processing flow for the PosAtt Engine;
- The PosAtt Engine's defining input parameters; and
- The output files generated by the PosAtt Engine.

The development of the PosAtt Engine was necessary to accommodate the structural changes to the sensor integration strategy that was required for the work summarized in Chapters 5 and 6. The PosAtt Engine was designed and developed as an object-oriented software utility, which has allowed robust modifications to be made to accomplish the research goals of this dissertation.

5 Three Attitude Models and their Characterization in the Generic Multi-Sensor Integration Strategy

This chapter is based upon the following published work:

Brunson, B. and Wang, J. (2023): Three Attitude Models and their Characterization in the Generic Multisensor Integration Strategy for Kinematic Positioning and Navigation. *The Journal of Global Positioning Systems*, 19(1), 79-98.

5.1 Preamble

Direct Geo-referencing Technology (DGT) is defined by its ability to directly assign Exterior Orientation (EO) to collected image frames or scan lines directly, thereby eliminating the need to use traditional aerial triangulation techniques [Hutton and Mostafa, 2005; Dreier et al, 2021]. DGT greatly simplifies the process of handling the large amounts of data collected by imaging sensors, and as a result has become popular across many different applications, ranging from surveying/mapping to kinematic positioning/navigation for airborne, marine, and land vehicles. With the increasing availability of low-cost positioning and orientation sensors on the market, such as MEMS IMUs, GNSS receivers, cameras, and LiDAR, there has been a significant push towards leveraging these sensors to provide high-accuracy Direct Geo-referencing at a lower cost.

A core aspect of any DGT system is its ability to accurately represent the system attitude and how it changes through time. The standard Roll-Pitch-Heading (RPH) representation discussed in Section 2.2.2 is the most common and direct way to represent the three-dimensional system attitude. Unfortunately, this attitude model famously suffers from the Gimbal Lock phenomenon when the system roll reaches $\pm 90^\circ$, which causes ambiguity and numerical instability in the attitude representation. To avoid this phenomenon, alternative attitude models such as the Direction Cosine Matrix (DCM) (discussed in Section 2.2.1) [Choukroun et al, 2010; Wang and Rajamani, 2018] and the Quaternion (discussed in Section 2.2.3) [Zhu et al, 2021; Yang, 2012; Sabatini, 2006] representations may be used.

A significant amount of research has been conducted that compares the *efficiency* of these alternative attitude models when used in the framework of the traditional multi-sensor integration strategy [e.g. Fresk and Nikolakopoulos, 2013; Song et al, 2020], but there is comparatively little

literature exploring the impact of changing the attitude model on the *accuracy* of the positioning solution. Moreover, the literature that does explore this accuracy impact tends to rely on either simulated data or having access to a reliable ground truth trajectory [Prasetyo and Musa, 2021; Sheng and Zhang, 2015; Golabek et al, 2022]. Given that the different attitude models make use of different parameterizations of the system attitude, it is very difficult to make direct comparisons between the estimated attitudes from different models without requiring conversion between them for comparison's sake.

This chapter focuses on developing multiple attitude models in the GMIS, along with their associated system models and any required constraint equations to avoid overparametrization. This chapter also leverages the unique comprehensive error analysis tools enabled by the GMIS to provide a direct means of comparing the accuracy of each alternative attitude model without either requiring access to a ground truth trajectory or converting estimated attitudes between different attitude representations. Specifically, this chapter uses the following quantities to thoroughly compare the positioning solutions between different attitude models:

- Residual estimates for the observation and process noise vectors;
- Redundancy contributions for the observation and process noise vectors; and
- Variance factors for each element in the observation and process noise vectors.

All of these quantities provide insight into the performance of the system and observation models without requiring a ground truth solution. Of particular note is the fact that the standard errors of unit weight estimated through Variance Component Estimation (VCE) are unitless, which allows for the direct comparison of performance across different attitude models without requiring conversion between different attitude representations.

5.2 Partitioning of the EKF Matrices

As described in Section 2.5.1, the EKF uses the system model to define system kinematics and measurement models to define how the observations at a given epoch relate to the current system state vector using (2.48) and (2.49).

Additionally, constraint equations may be used to impose *a priori* knowledge on the system state vector using equality constraint equations as defined in (3.4).

As its core components, the navigation Kalman Filter used in this research describes the linear position, velocity, and acceleration of the system, as well as the attitude parameters and their first-order time derivatives. The system model here is realized after kinematic equations that govern the motion of a moving platform, specifically using a constant linear acceleration and constant first-order time derivative of the attitude parameters between successive observation epochs to model the system's motion through time.

Given this research's goal of accommodating multiple attitude models within the GMIS, careful consideration has to be given to the general structure of the system state vector and the definition of any related vectors/matrices. To this end, it is useful to partition both the system state vector and the observation vector into related components that simplify the generation of the related design/transition matrices and allow them to be constructed in a modular fashion.

In this chapter, the system state vector is partitioned into the following groups of related quantities:

- Position, velocity, and acceleration;
- Attitude, and its rate of change;
- GNSS float ambiguity parameter estimates; and
- IMU systematic error states (gyroscope/accelerometer bias and scale factor errors).

The partitioned system state vector is expressed as

$$\mathbf{x} = [\mathbf{x}_p^T \quad \mathbf{x}_a^T \quad \mathbf{x}_\lambda^T \quad \mathbf{x}_{IMU}^T]^T \quad (5.1)$$

wherein the subscripts p , a , λ , and IMU stand for the linear kinematic state vector (i.e. concatenated position, velocity, and acceleration vectors), the angular kinematic state vector (i.e. concatenated attitude parameter and attitude parameters' time derivative vectors), the GNSS ambiguity vector (only present when float estimates are being made), and the IMU systematic error vector, respectively.

Similarly, the observation vector may also be partitioned into the following groups of related quantities:

- All gyroscope angular rate measurements;
- All accelerometer specific force measurements; and
- All GNSS observables (i.e. double-differenced L1 C/A, L1 Carrier Phase, and L2 Carrier Phase observations).

This division results in the following definition of the partitioned system innovation vector:

$$\mathbf{d} = [\mathbf{d}_g^T \quad \mathbf{d}_s^T \quad \mathbf{d}_G^T]^T \quad (5.2)$$

where $\mathbf{d}_g$, $\mathbf{d}_s$, and $\mathbf{d}_G$ refer to the gyroscope, accelerometer, and GNSS observables at epoch $k + 1$, respectively.

The definition of the two partitioned vectors in (5.1) and (5.2) additionally allow for the corresponding definitions of the partitioned system state transition matrix $\mathbf{A}$, process noise vector $\mathbf{w}$, process noise transition matrix $\mathbf{B}$, and design matrix $\mathbf{C}$.

Assuming the linear motion of the system to be modeled separately from the system's attitude dynamics, the system state transition matrix may be partitioned as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_\lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{IMU} \end{bmatrix} \quad (5.3)$$

The process noise vector may similarly be separated into components that independently influence the system's position, attitude, float ambiguity estimates, and IMU systematic errors as follows:

$$\mathbf{w} = [\mathbf{w}_p^T \quad \mathbf{w}_a^T \quad \mathbf{w}_\lambda^T \quad \mathbf{w}_{IMU}^T]^T \quad (5.4)$$

Again, assuming that the attitude process noise elements do not affect the system's position estimates, the definition of the process noise vector in (5.4) may have an associated process noise transition matrix of

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_p & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_\lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{IMU} \end{bmatrix} \quad (5.5)$$

Since both the system state vector and the system innovation vector are partitioned, the design matrix must accommodate both partitioned vectors' definitions. This is accomplished as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{g,p} & \mathbf{C}_{g,a} & \mathbf{C}_{g,\lambda} & \mathbf{C}_{g,IMU} \\ \mathbf{C}_{s,p} & \mathbf{C}_{s,a} & \mathbf{C}_{s,\lambda} & \mathbf{C}_{s,IMU} \\ \mathbf{C}_{G,p} & \mathbf{C}_{G,a} & \mathbf{C}_{G,\lambda} & \mathbf{C}_{G,IMU} \end{bmatrix} \quad (5.6)$$

Any system state equality constraints may also be represented using the partitioned constraint Jacobian matrix

$$\mathbf{H} = [\mathbf{H}_p \quad \mathbf{H}_a \quad \mathbf{H}_\lambda \quad \mathbf{H}_{IMU}] \quad (5.7)$$

The partitioning of the matrices from (5.3) to (5.6) is particularly useful when making changes to the models being used in the GMIS. When only some of the elements of the system state vector are changed, these equations allow for a more fine-tuned identification of which block elements of the transition/design matrices also need to change, and which may be considered to be model-independent. This eliminates the need to reformulate these entire matrices when changing modeling procedures, and makes the definition of these matrices much more flexible and robust.

As an example, in this research the attitude models are changed, but the models of the system's linear position, float ambiguity parameters, and IMU systematic errors remain the same. As such, in (5.3), only $\mathbf{A}_a$ should change as the attitude model is changed, only the definition of $\mathbf{w}_a$ in (5.4) should change, and only the definition of $\mathbf{B}_a$ should change. Additionally, only the elements $\mathbf{C}_{g,a}$, $\mathbf{C}_{s,a}$, and $\mathbf{C}_{G,a}$ can potentially change as the attitude model is changed, and these elements may be focused on in the modeling process. In fact, the matrix $\mathbf{C}_{G,a}$ may be considered to not change with the attitude model if the on-board GNSS sensor is considered to be at the origin of the vehicle frame.

5.3 Position System Model in the GMIS

Since only the alternative attitude models are being changed in this chapter, the definition of the system model for the linear position states are consistent across all attitude models. Specifically, the system is assumed to have a constant linear acceleration from one observation epoch to the next. This may be expressed using the equation

$$\mathbf{x}_p(k+1) = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 & \frac{1}{2} \Delta t^2 \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \Delta t \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{x}_p(k) + \begin{bmatrix} \frac{1}{6} \Delta t^3 \mathbf{I}_3 \\ \frac{1}{2} \Delta t^2 \mathbf{I}_3 \\ \Delta t \mathbf{I}_3 \end{bmatrix} \mathbf{j}(k) = \mathbf{A}_p \mathbf{x}_p(k) + \mathbf{B}_p \mathbf{w}_p \quad (5.8)$$

where $\mathbf{j}(k)$ is the third-order motion process noise vector, also called the system jerk, which is expressed in the local ENU navigation frame. The system model in (5.8) is identical to the system model defined by (4.1), (4.2), and (4.3) but uses the partitioned state vector for simpler notation. In general, all

linear motion of the system is modeled in the local ENU navigation frame, since this ensures that (5.8) is not impacted by different choices of how attitude is being represented.

5.4 IMU Observation Equations

Before moving on to describing the attitude model-specific IMU observation equations, it is important to provide the general IMU observation equations that will be adapted for each attitude model used in this research. As described in Section 2.4.2.4.1, the general accelerometer observation equation is given by

$$\begin{aligned} f_{ib}^b &= C_n^b a + C_n^b g - b_a - S_a f_{ib}^b + C_n^b \{2\omega_{ie}^n + \omega_{en}^n\} \times v + C_n^b \omega_{ib}^b \times \omega_{ib}^b \times r + \Delta_{f_{ib}}^b \\ &= C_n^b s - b_a - S_a f_{ib, meas}^b + \Delta_{f_{ib}}^b \end{aligned} \quad (2.46)$$

where s is a helping parameter vector used in collecting like terms for C_n^b and is defined by the equation $s = a + g + \{2\omega_{ie}^n + \omega_{en}^n\} \times v + \omega_{ib}^b \times \omega_{ib}^b \times r$.

As described in Section 2.4.2.4.2, the general gyroscope observation equation is given by

$$\omega_{ib}^b = \omega_{nb}^b - b_g - S_g \omega_{ib}^b + C_n^b (\omega_{ie}^n + \omega_{en}^n) + \Delta_{\omega_{ib}}^b \quad (2.47)$$

In both (2.46) and (2.47), the definitions of ω_{nb}^b and C_n^b do not specifically refer to any particular attitude model, but rather to the general rotation rate of the body frame with respect to the navigation frame, and the rotation matrix from the navigation frame to the body frame, respectively. Both quantities may be calculated using the parameters from different attitude representations, allowing these equations to be the basis for modeling IMU observations in multiple attitude models.

5.5 Attitude Models in the GMIS

This research considers three attitude models: the Roll-Pitch-Heading (RPH) model, the Direction Cosine Matrix (DCM) model, and the Quaternion model. This section develops the system models and measurement models for each attitude model within the GMIS.

5.5.1 Roll-Pitch-Heading (RPH) Model

The roll-pitch-heading (RPH) attitude model uses the roll, pitch, and heading angles to uniquely represent attitude in three-dimensional space. This attitude model is a specific realization of an Euler rotation, and the rotation matrix from the navigation frame to the body frame used in this research is given as follows:

$$\mathbf{C}_n^b = \mathbf{R}_2(\alpha)\mathbf{R}_1(\beta)\mathbf{R}_3(\gamma) \quad (5.9)$$

where α , β , and γ are the system roll, pitch, and heading angles, respectively. The system's angular motion is defined by the first-order time derivatives $\dot{\alpha}$, $\dot{\beta}$, and $\dot{\gamma}$ and are included in the system state vector so that the definition of $\mathbf{x}_a$ is given by

$$\mathbf{x}_a = [\alpha \quad \beta \quad \gamma \quad \dot{\alpha} \quad \dot{\beta} \quad \dot{\gamma}]^T \quad (5.10)$$

Considering the attitude angles to vary linearly with time, their second-order time derivatives $\ddot{\alpha}$, $\ddot{\beta}$, and $\ddot{\gamma}$ define their associated process noise, yielding the following definition of $\mathbf{w}_a$:

$$\mathbf{w}_a = [\ddot{\alpha} \quad \ddot{\beta} \quad \ddot{\gamma}]^T \quad (5.11)$$

Under the assumed linear first-order time derivative models, the overall system model for the roll-pitch-heading parameters are identical to those defined in (4.4) to (4.9).

The definition of the system models for the RPH attitude model allows for the subsequent definition of the corresponding blocks of the system state and process noise transition matrices, as follows:

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad (5.12)$$

and

$$\mathbf{B}_a = \begin{bmatrix} \frac{1}{2} \Delta t^2 \mathbf{I}_3 \\ \Delta t \mathbf{I}_3 \end{bmatrix} \quad (5.13)$$

After [Zhao, 2016], the differential equation relating $\boldsymbol{\omega}_{nb}^b$ and $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$ is defined by

$$\boldsymbol{\omega}_{nb}^b = \begin{bmatrix} 0 & \cos \alpha & \sin \alpha \cos \beta \\ 1 & -\sin \beta & 0 \\ 0 & \sin \alpha & -\cos \alpha \cos \beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{D} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (5.14)$$

The substitution of (5.14) into (2.47) results in the following definition of the gyroscope submatrix in the design matrix:

$$\mathbf{C}_{g,a} = \begin{bmatrix} 0 & \cos \alpha & \sin \alpha \cos \beta \\ \mathbf{0}_3 & 1 & -\sin \beta & 0 \\ 0 & \sin \alpha & -\cos \alpha \cos \beta \end{bmatrix} \quad (5.15)$$

The accelerometer observation equations remain unchanged when using the RPH attitude model, but the partial derivatives of the accelerometer observation equations with respect to the RPH

parameters need to be expressed. Here, the partial derivatives of the rotation matrix with respect to the three parameters are useful to derive, and they are defined as follows:

$$\frac{\partial \mathbf{C}_n^b}{\partial \alpha} = \frac{\partial \mathbf{R}_2(\alpha)}{\partial \alpha} \mathbf{R}_1(\beta) \mathbf{R}_3(\gamma) \quad (5.16)$$

$$\frac{\partial \mathbf{C}_n^b}{\partial \beta} = \mathbf{R}_2(\alpha) \frac{\partial \mathbf{R}_1(\beta)}{\partial \beta} \mathbf{R}_3(\gamma) \quad (5.17)$$

and

$$\frac{\partial \mathbf{C}_n^b}{\partial \gamma} = \mathbf{R}_2(\alpha) \mathbf{R}_1(\beta) \frac{\partial \mathbf{R}_3(\gamma)}{\partial \gamma} \quad (5.18)$$

Using (5.16) to (5.18) to evaluate the partial derivatives of the rotation matrix with respect to each of the attitude parameters allows for the subsequent calculation of the partial derivatives of the accelerometer observation equations with respect to the attitude parameters. This allows for the following definition of the accelerometer submatrix in the design matrix:

$$\mathbf{C}_{s,a} = \begin{bmatrix} \frac{\partial \mathbf{C}_n^b}{\partial \alpha} \mathbf{s} & \frac{\partial \mathbf{C}_n^b}{\partial \beta} \mathbf{s} & \frac{\partial \mathbf{C}_n^b}{\partial \gamma} \mathbf{s} & \mathbf{0}_3 \end{bmatrix} \quad (5.19)$$

These redefinitions in the positioning Kalman Filter are sufficient to represent the RPH attitude model. Since the RPH attitude model uses the minimum necessary parameters to represent orientation in three-dimensional space (three parameters) and the minimum necessary parameters to represent the orientation rate of change in three-dimensional space (three parameters), there are no redundant parameters included in the system state vector, and there is therefore no need to enforce any system state constraints except for when additional *a priori* information on the system attitude is available.

5.5.2 Direction Cosine Matrix (DCM) Model

In the Direction Cosine Matrix (DCM) representation, the individual elements of the attitude matrix are each considered elements of the system state vector and their first-order time derivatives are used to model how the system attitude changes through time. This results in a total of eighteen system state elements for the DCM attitude model. Using these parameters, the rotation matrix from the navigation frame to the body frame is given by the equation

$$\mathbf{C}_n^b = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (5.20)$$

and the first-order time derivative of the rotation matrix is given by the equation

$$\dot{\mathbf{C}}_n^b = \begin{bmatrix} \dot{c}_{11} & \dot{c}_{12} & \dot{c}_{13} \\ \dot{c}_{21} & \dot{c}_{22} & \dot{c}_{23} \\ \dot{c}_{31} & \dot{c}_{32} & \dot{c}_{33} \end{bmatrix} \quad (5.21)$$

Including all of these attitude parameters in the system state vector results in the following definition of $\mathbf{x}_a$:

$$\mathbf{x}_a = [\mathbf{C}_1 \quad \mathbf{C}_2 \quad \mathbf{C}_3 \quad \dot{\mathbf{C}}_1 \quad \dot{\mathbf{C}}_2 \quad \dot{\mathbf{C}}_3] \quad (5.22)$$

where $\mathbf{C}_i = [c_{i1} \quad c_{i2} \quad c_{i3}]$ and $\dot{\mathbf{C}}_i = [\dot{c}_{i1} \quad \dot{c}_{i2} \quad \dot{c}_{i3}]$ for $i = 1, 2, 3$.

Considering the attitude parameters to change linearly with time, their second-order derivatives $\ddot{\mathbf{C}}_i = [\ddot{c}_{i1} \quad \ddot{c}_{i2} \quad \ddot{c}_{i3}] \forall i = 1, 2, 3$ define their associated process noise, resulting in the attitude process noise being defined as

$$\mathbf{w}_a = [\ddot{\mathbf{C}}_1 \quad \ddot{\mathbf{C}}_2 \quad \ddot{\mathbf{C}}_3] \quad (5.23)$$

Using these definitions, the eighteen system equations are given by

$$c_{ij}(k+1) = c_{ij}(k) + \dot{c}_{ij}(k)\Delta t + \frac{1}{2}\ddot{c}_{ij}(k)\Delta t^2, i \in (1, 2, 3), j \in (1, 2, 3) \quad (5.24)$$

and

$$\dot{c}_{ij}(k+1) = \dot{c}_{ij}(k) + \ddot{c}_{ij}(k)\Delta t, i \in (1, 2, 3), j \in (1, 2, 3) \quad (5.25)$$

The transition matrix associated with this system model is given by

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{I}_9 & \Delta t \mathbf{I}_9 \\ \mathbf{0}_9 & \mathbf{I}_9 \end{bmatrix} \quad (5.26)$$

and the associated process noise transition matrix is given by

$$\mathbf{B}_a = \begin{bmatrix} \frac{1}{2}\Delta t^2 \mathbf{I}_9 \\ \Delta t \mathbf{I}_9 \end{bmatrix} \quad (5.27)$$

The relationship between the DCM attitude parameters and $\boldsymbol{\omega}_{nb}^b$ may be determined via the Poisson Equation in Section 2.3.4 as

$$\dot{\mathbf{C}}_n^b = \mathbf{C}_n^b [\boldsymbol{\omega}_{nb}^b] \quad (5.28)$$

The skew-symmetric cross product matrix of the $\boldsymbol{\omega}_{nb}^b$ vector may be isolated by rearranging (5.28) as

$$[\omega_{nb}^b] = (C_n^b)^T \dot{C}_n^b \quad (5.29)$$

Considering the elements of the skew-symmetric cross product matrix, the elements of ω_{nb}^b may be individually evaluated as

$$\omega_{nb}^b = \begin{bmatrix} C_{13}\dot{C}_{12} + C_{23}\dot{C}_{22} + C_{33}\dot{C}_{32} \\ C_{11}\dot{C}_{13} + C_{21}\dot{C}_{23} + C_{31}\dot{C}_{33} \\ C_{12}\dot{C}_{11} + C_{22}\dot{C}_{21} + C_{32}\dot{C}_{31} \end{bmatrix} \quad (5.30)$$

and this evaluation of ω_{nb}^b may be substituted directly into (2.47).

The portion of the design matrix referring to the gyroscope measurements may therefore be defined as

$$C_{g,a} = \begin{bmatrix} 0 & 0 & \dot{c}_{12} & 0 & 0 & \dot{c}_{22} & 0 & 0 & \dot{c}_{32} & 0 & c_{13} & 0 & 0 & c_{23} & 0 & 0 & c_{33} & 0 \\ \dot{c}_{13} & 0 & 0 & \dot{c}_{23} & 0 & 0 & \dot{c}_{33} & 0 & 0 & 0 & 0 & c_{11} & 0 & 0 & c_{21} & 0 & 0 & c_{31} \\ 0 & \dot{c}_{11} & 0 & 0 & \dot{c}_{21} & 0 & 0 & \dot{c}_{31} & 0 & c_{12} & 0 & 0 & c_{22} & 0 & 0 & c_{32} & 0 & 0 \end{bmatrix} \quad (5.31)$$

The accelerometer observation equations remain unchanged when using the DCM attitude model, but the partial derivatives of the rotation matrix with respect to the DCM attitude parameters needs to be expressed. The partial derivative of the rotation matrix with respect to element c_{ij} of the DCM attitude model may be expressed as

$$\frac{\partial C_n^b}{\partial c_{ij}}(k, l) = \delta_{ik}\delta_{jl}, k \in (1, 2, 3), l \in (1, 2, 3) \quad (5.32)$$

where δ_{ik} is the Kronecker delta function evaluated at i and k .

This definition of the partial derivatives of the rotation matrix with respect to the DCM attitude parameters allows for the following definition of the portion of the design matrix referring to the accelerometer measurements:

$$C_{s,a} = \begin{bmatrix} s^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,9} \\ \mathbf{0}_{1,3} & s^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,9} \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & s^T & \mathbf{0}_{1,9} \end{bmatrix} \quad (5.33)$$

5.5.2.1 DCM State Constraints

Given that the attitude of a system may be fully described using three parameters, and the system's orientation rate of change may also be fully described using three parameters, using the eighteen parameters in the DCM attitude model results in twelve of the eighteen attitude parameters

being functionally redundant. This requires the inclusion of twelve constraint equations on the DCM attitude states to avoid over-parameterizing the system attitude.

The first six constraint equations may be readily derived from the well-known orthogonal property of a rotation matrix:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} [c_1^T \quad c_2^T \quad c_3^T] = I_3 \quad (5.34)$$

The upper or lower triangular elements of the identity matrix in (5.34) leads directly to the following six constraint equations:

$$c_1 c_1^T = 1 \quad (5.35)$$

$$c_2 c_2^T = 1 \quad (5.36)$$

$$c_3 c_3^T = 1 \quad (5.37)$$

$$c_1 c_2^T = 0 \quad (5.38)$$

$$c_1 c_3^T = 0 \quad (5.39)$$

$$c_2 c_3^T = 0 \quad (5.40)$$

These six constraint equations effectively eliminate the six redundant system states when it comes to representing the system's orientation, but they do not address the fact that there are also redundant system states representing the rate of change of the system's orientation. There are two strategies that may be applied to enforce system state constraints on the first-order time derivatives of the DCM attitude parameters:

1. Using the matrix product in the rearranged Poisson Equation in (5.29) to enforce that the resultant product is a skew-symmetric matrix (i.e. diagonal elements should be equal to 0 and the sum of an off-diagonal element with the corresponding element of its transpose should be 0); or
2. Recognizing that the constraint equations from (5.35) to (5.40) remain constant through time to enforce that the first-order time derivatives of the six constraint equations should all be equal to 0.

Here, the derivation of both sets of constraint equations on the attitude parameters' time derivatives is presented to demonstrate that both approaches are equivalent to one another.

Using the Poisson Equation:

Combining the definition of the system state vector elements with the rearranged Poisson Equation in (5.29) yields the following expression of the Poisson Equation:

$$[\omega_{nb}^b] = [C_1^T \quad C_2^T \quad C_3^T] \begin{bmatrix} \dot{C}_1 \\ \dot{C}_2 \\ \dot{C}_3 \end{bmatrix} \quad (5.41)$$

Recognizing that $[\omega_{nb}^b]$ is a skew-symmetric matrix leads to the following three constraint equations for the diagonal elements of the cross-product matrix:

$$C_1 \dot{C}_1^T = 0 \quad (5.42)$$

$$C_2 \dot{C}_2^T = 0 \quad (5.43)$$

$$C_3 \dot{C}_3^T = 0 \quad (5.44)$$

and taking the difference between each off-diagonal element in (5.41) and its transpose yields the following three additional constraint equations:

$$C_1 \dot{C}_2^T + \dot{C}_1 C_2^T = 0 \quad (5.45)$$

$$C_1 \dot{C}_3^T + \dot{C}_1 C_3^T = 0 \quad (5.46)$$

$$C_2 \dot{C}_3^T + \dot{C}_2 C_3^T = 0 \quad (5.47)$$

The six equations from (5.42) to (5.47) constitute the six additional constraint equations needed to effectively remove the additional six redundant parameters in the system state vector.

Using the Time Derivatives of the Other Constraint Equations:

The Chain Rule may be used to calculate the first-order time derivatives of (5.35) to (5.37) as follows:

$$\frac{\partial(C_1 C_1^T)}{\partial t} = 2C_1 \dot{C}_1^T \quad (5.48)$$

$$\frac{\partial(C_2 C_2^T)}{\partial t} = 2C_2 \dot{C}_2^T \quad (5.49)$$

$$\frac{\partial(C_3 C_3^T)}{\partial t} = 2C_3 \dot{C}_3^T \quad (5.50)$$

The Product Rule may be used to calculate the first-order time derivatives of (5.38) to (5.40) as follows:

$$\frac{\partial(\mathbf{c}_1 \mathbf{c}_2^T)}{\partial t} = \mathbf{c}_1 \dot{\mathbf{c}}_2^T + \dot{\mathbf{c}}_1 \mathbf{c}_2^T \quad (5.51)$$

$$\frac{\partial(\mathbf{c}_1 \mathbf{c}_3^T)}{\partial t} = \mathbf{c}_1 \dot{\mathbf{c}}_3^T + \dot{\mathbf{c}}_1 \mathbf{c}_3^T \quad (5.52)$$

$$\frac{\partial(\mathbf{c}_2 \mathbf{c}_3^T)}{\partial t} = \mathbf{c}_2 \dot{\mathbf{c}}_3^T + \dot{\mathbf{c}}_2 \mathbf{c}_3^T \quad (5.53)$$

Intuitively, the constraint equations from (5.35) to (5.40) remain constant-valued through time, so their first-order time derivatives should be equal to 0. Setting the time derivatives from (5.48) to (5.53) to be equal to 0 yields the following six constraint equations:

$$\mathbf{c}_1 \dot{\mathbf{c}}_1^T = 0 \quad (5.54)$$

$$\mathbf{c}_2 \dot{\mathbf{c}}_2^T = 0 \quad (5.55)$$

$$\mathbf{c}_3 \dot{\mathbf{c}}_3^T = 0 \quad (5.56)$$

$$\mathbf{c}_1 \dot{\mathbf{c}}_2^T + \dot{\mathbf{c}}_1 \mathbf{c}_2^T = 0 \quad (5.57)$$

$$\mathbf{c}_1 \dot{\mathbf{c}}_3^T + \dot{\mathbf{c}}_1 \mathbf{c}_3^T = 0 \quad (5.58)$$

$$\mathbf{c}_2 \dot{\mathbf{c}}_3^T + \dot{\mathbf{c}}_2 \mathbf{c}_3^T = 0 \quad (5.59)$$

Notably, the six constraint equations from (5.54) to (5.59) are identical to those from (5.42) to (5.47). This supports the use of the first-order time derivatives to constrain the system state elements as a valid approach to eliminating redundant system state elements. Establishing this fact is important when deriving the system state constraints for the quaternion attitude model, where the Poisson Equation cannot be as readily applied to deriving system state constraints.

Together, the twelve system state constraints coming from (5.35) to (5.40) and from (5.54) to (5.59) are enough to eliminate the functional singularity coming from the use of twelve redundant attitude states when using the DCM model. All of these constraint equations are non-linear, and their Jacobian matrix is given by

$$H_{DCM,a} = \begin{bmatrix} 2C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2C_3 & 0 & 0 & 0 \\ C_2 & C_1 & 0 & 0 & 0 & 0 \\ C_3 & 0 & C_1 & 0 & 0 & 0 \\ 0 & C_3 & C_2 & 0 & 0 & 0 \\ \dot{C}_1 & 0 & 0 & C_1 & 0 & 0 \\ 0 & \dot{C}_2 & 0 & 0 & C_2 & 0 \\ 0 & 0 & \dot{C}_3 & 0 & 0 & C_3 \\ \dot{C}_2 & \dot{C}_1 & 0 & C_2 & C_1 & 0 \\ \dot{C}_3 & 0 & \dot{C}_1 & C_3 & 0 & C_1 \\ 0 & \dot{C}_3 & \dot{C}_2 & 0 & C_3 & C_2 \end{bmatrix} \quad (5.60)$$

where practically, (5.60) may be evaluated using the most recently-estimated DCM attitude states.

5.5.3 Quaternion Model

Using rotation quaternions to represent system attitude is the most generally preferred formulation of the attitude for inertial navigation to this point. The rotation quaternion is defined to be

$$q = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (5.61)$$

with the four defining parameters (q_0, q_1, q_2, q_3) . This attitude representation is very popular due to its resistance to degenerate states such as the Gimbal Lock phenomenon and its representation of rotation using algebraic representations, rather than relying on trigonometric functions that can be more costly to evaluate. Under this attitude model, the rotation matrix from the navigation frame to the body frame is expressed as per [Zhang and Sun, 1992] as

$$C_n^b = (q_0^2 - |\mathbf{q}|^2)\mathbf{I}_3 - 2q_0[\mathbf{q}] + 2\mathbf{q}\mathbf{q}^T \quad (5.62)$$

The four defining parameters in (5.61) along with their first-order time derivatives are used in the system state vector to model the system attitude and its dynamics through time. Including all of these parameters in the system state vector results in the following definition of $\mathbf{x}_a$:

$$\mathbf{x}_a = [q_0 \quad \mathbf{q}^T \quad \dot{q}_0 \quad \dot{\mathbf{q}}^T]^T \quad (5.63)$$

Similarly to the RPH and DCM attitude models, the quaternion attitude states may be considered to change linearly between successive observation epochs, with the process noise defined to be the second-order time derivatives of the quaternion attitude states as

$$\mathbf{w}_a = [\ddot{q}_0 \quad \ddot{\mathbf{q}}^T]^T \quad (5.64)$$

Using these definitions, the eight system model equations defining the system kinematics are given by the equations

$$q_i(k+1) = q_i(k) + \dot{q}_i(k)\Delta t + \frac{1}{2}\ddot{q}_i(k)\Delta t^2, i \in (0, 1, 2, 3) \quad (5.65)$$

and

$$\dot{q}_i(k+1) = \dot{q}_i(k) + \ddot{q}_i(k)\Delta t, i \in (0, 1, 2, 3) \quad (5.66)$$

The system state transition matrix associated with these system model equations is given by

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{I}_4 & \Delta t \mathbf{I}_4 \\ \mathbf{0}_4 & \mathbf{I}_4 \end{bmatrix} \quad (5.67)$$

and the associated process noise transition matrix is given by

$$\mathbf{B}_a = \begin{bmatrix} \frac{1}{2}\Delta t^2 \mathbf{I}_4 \\ \Delta t \mathbf{I}_4 \end{bmatrix} \quad (5.68)$$

The Poisson Equation may be used to relate ω_{nb}^b to the quaternion attitude representation as

$$\dot{q} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_{nb}^b \end{bmatrix} \quad (5.69)$$

Rearranging (5.69) to isolate ω_{nb}^b allows it to be explicitly expressed as follows:

$$\omega_{nb}^b = 2 \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \dot{q} = 2\mathbf{W}(q)\dot{q} \quad (5.70)$$

Substituting (5.70) into (2.47) yields the following definition of the partitioned attitude elements of the design matrix:

$$\mathbf{C}_{g,a} = [\mathbf{0}_{3,4} \quad 2\mathbf{W}(q)] \quad (5.71)$$

As for the other attitude models, the accelerometer observation equations remain unchanged, but the partial derivatives of the attitude matrix with respect to the attitude parameters of the quaternion model must be derived, as follows:

$$\frac{\partial \mathbf{C}_n^b}{\partial q_0} = 2q_0 \mathbf{I}_3 - 2[\mathbf{q}] \quad (5.72)$$

$$\frac{\partial \mathbf{C}_n^b}{\partial q_1} = -2q_1 \mathbf{I}_3 - 2q_0 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + 2 \begin{bmatrix} 2q_1 & q_2 & q_3 \\ q_2 & 0 & 0 \\ q_3 & 0 & 0 \end{bmatrix} \quad (5.73)$$

$$\frac{\partial \mathbf{C}_n^b}{\partial q_2} = -2q_2 \mathbf{I}_3 - 2q_0 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} + 2 \begin{bmatrix} 0 & q_1 & 0 \\ q_1 & 2q_2 & q_3 \\ 0 & q_3 & 0 \end{bmatrix} \quad (5.74)$$

$$\frac{\partial \mathbf{C}_n^b}{\partial q_3} = -2q_3 \mathbf{I}_3 - 2q_0 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 2 \begin{bmatrix} 0 & 0 & q_1 \\ 0 & 0 & q_2 \\ q_1 & q_2 & 2q_3 \end{bmatrix} \quad (5.75)$$

Using the partial derivatives defined from (5.72) to (5.75), the partitioned element describing the partial derivatives of the accelerometer measurements with respect to the quaternion model's attitude parameters may be defined to be

$$\mathbf{C}_{s,a} = \begin{bmatrix} \frac{\partial \mathbf{C}_n^b}{\partial q_0} \mathbf{s} & \frac{\partial \mathbf{C}_n^b}{\partial q_1} \mathbf{s} & \frac{\partial \mathbf{C}_n^b}{\partial q_2} \mathbf{s} & \frac{\partial \mathbf{C}_n^b}{\partial q_3} \mathbf{s} & \mathbf{0}_4 \end{bmatrix} \quad (5.76)$$

5.5.3.1 Quaternion State Constraints

The quaternion attitude model makes use of eight total system state elements when only six are necessary to represent the system's attitude and how it changes through time.

The first quaternion state constraint comes from the definition of a rotation quaternion:

$$qq^* = q_0^2 + |\mathbf{q}|^2 = 1 \quad (5.77)$$

Since the constraint equation in (5.77) should remain constant through time, its first-order time derivative should intuitively be equal to 0, which provides the second quaternion state constraint:

$$q_0 \dot{q}_0 + q_1 \dot{q}_1 + q_2 \dot{q}_2 + q_3 \dot{q}_3 = 0 \quad (5.78)$$

Upon calculating the Jacobian matrix of the non-linear constraint equations in (5.77) and (5.78), the partitioned block of the constraint coefficient matrix for the attitude parameters is given by the equation

$$\mathbf{H}_{q,a} = \begin{bmatrix} 2q_0 & 2q_1 & 2q_2 & 2q_3 & 0 & 0 & 0 & 0 \\ \dot{q}_0 & \dot{q}_1 & \dot{q}_2 & \dot{q}_3 & q_0 & q_1 & q_2 & q_3 \end{bmatrix} \quad (5.79)$$

where practically, (5.79) is evaluated using the most recently-estimated quaternion states and their estimated time derivatives.

5.6 Comprehensive Error Analysis

Generally, to estimate the process noise and observation residual vectors, the framework of the system state constrained Extended Kalman Filter described in Chapter 3 may be applied. This framework allows for the direct calculation of the constrained residuals of the predicted system state vector $\hat{\mathbf{v}}_{x_h}$, the process noise residuals $\hat{\mathbf{v}}_{w_h}$, and the observation residuals $\hat{\mathbf{v}}_{z_h}$ by evaluating (3.80), (3.82), and (3.84), respectively. Their associated covariance matrices may be calculated by evaluating (3.81), (3.83), and (3.85), respectively.

Since this chapter focuses on using these quantities to characterize the performance of different attitude models in the GMIS, it is important to clarify several things with respect to the meaning and significance of these three residual vectors. As described in Chapter 3, each of these residual vectors represents one of the three independent sources of error in a Kalman Filter for any given observation epoch: the random error in the previously-estimated system state used for state prediction in the Time Update (residuals of the predicted system state vector); the random error in the process noise estimates (process noise residuals); and the random error in the measurements (observation residuals). Generally, the process noise residuals and observation residuals provide more useful information than the residuals of the predicted system state vector, because the system state elements are expected to be more strongly correlated with one another.

Changing the attitude model primarily impacts two things:

- 1) The measurement models: While the ways that measurements are represented in each attitude model are different, there are no approximations or assumptions being made for any individual attitude model. In other words, the measurement models are expressed differently but are expected to be equivalent to one another. The general behaviour of the observation residual vector is therefore expected to be consistent across the different attitude models. Most differences in the measurement residuals may be attributed to how the system state is reprojected by the constraint equations, and any errors that may accumulate differently across different attitude models.
- 2) The system model: Since each attitude model uses the first-order time derivatives of the attitude parameters to characterize the angular motion of the system, there are actual differences between the kinematic models being used to define each of the attitude models.

Most of the differences in the performance of each attitude model is therefore expected to be reflected in their respective process noise residuals.

It is very common to analyze the double-differenced GNSS observation residuals, since they may be readily calculated from the estimated system position under the traditional integration strategy; however, the traditional integration strategy cannot accommodate the calculation of the raw IMU observation residuals since it generally integrates the INS solution rather than using raw IMU measurements directly.

Unique to this research, the raw IMU observation residuals may be estimated for the IMU accelerometers and gyroscopes, in addition to the process noise vector characterizing the performance of the kinematic models. These are only possible to calculate due to the combined contributions of the modeling work in this chapter along with the comprehensive error analysis tools developed in Chapter 3.

Accelerometer and gyroscope measurements tend to be particularly closely tied to and impacted by the system's estimated attitude, and so the accelerometer and gyroscope residuals are therefore expected to be more indicative of the attitude models' performance than the GNSS observation residuals. Similarly, the process noise residual vector has elements that describe the system jerk vector (i.e. third-order time derivative of the system's position) and elements that describe the second-order time derivatives of the system's attitude parameters. Clearly, the process noise residuals tied to the system's attitude parameters may be considered more informative on the performance of the system models associated with each attitude model.

One particularly important tool in this analysis is Variance Component Estimation (VCE). The process noise elements referring to the system attitude are defined differently across the different attitude models, and are therefore not directly comparable between them. VCE may be used to estimate the standard errors of unit weight for the attitude process noise, which provides a single unitless value that describes how the *a posteriori* behaviour of the system compares to its assumed *a priori* accuracy. Since VCE produces a unitless quantity as its output, the VCE output may be directly compared across different attitude models without requiring conversions between the different attitude models. While the gyroscope and accelerometer measurement residuals do not suffer from this issue, and they are directly comparable across different attitude models, it is much more intuitive to compare their associated standard errors of unit weight to characterize the overall performance of the system for the alternative attitude models.

Additionally, VCE directly uses the estimated observation and process noise residuals. Since these residuals are estimated *a posteriori* without requiring a fixed reference solution, VCE may be used to characterize the performance of a system without requiring a ground-truth dataset.

When conducting VCE, appropriately partitioning the observation and process noise residuals is important in order to obtain sufficiently detailed information to draw significant conclusions. There are competing priorities: maximizing the amount of meaningful information that may be gleaned from the partitioned residual vectors, while also ensuring that there is enough redundancy in the estimation of their variance component factors to allow these quantities to remain significant. In this chapter, the observation and process noise residuals are partitioned to allow for variance component factors to be estimated for the following quantities:

- i. For the process noise residuals:
 - a. An overall variance component accounting for the entire process noise vector, which may be used to provide an overview of the overall performance of the positioning Kalman Filter's system model.
 - b. A variance component accounting for the position-related elements of the process noise vector. This may be used to provide more granular information about how well the system model represents the linear motion of the system.
 - c. A variance component accounting for the orientation-related elements of the process noise vector. This may be used to provide more granular information about how well the system model represents the angular motion of the system.
- ii. For the observation residuals:
 - a. An overall variance component accounting for the entire observation vector, which may be used to characterize how well the measurement models perform overall for each attitude model.
 - b. A variance component dedicated to characterizing the performance of the raw gyroscope angular rate measurements, and another variance component dedicated to characterizing the performance of the raw accelerometer specific force measurements. The characterization of these variance components is unique to this research, and is a result of the collective contributions of the GMIS' modelling capabilities alongside the additional modeling work developed in Chapter 3 and this chapter.

- c. A variance component dedicated to characterizing the performance of the double-differenced L1 C/A observations, and two other variance components dedicated to characterizing the performance of the double-differenced L1 Carrier Phase and L2 Carrier Phase observations.

As has been previously alluded to, it is important to ensure that there is enough redundancy in the estimates of any variance components being estimated to have them provide reliable estimates of the system performance. The process noise vector in particular tends to have very small redundant indices across its elements, and so it is generally not possible to calculate reliable variance components for process noise elements at a single observation epoch. Practically, this means that variance components must instead be estimated over a window of time, accounting for the *cumulative* redundancy contribution over that time period. Increasing the size of the moving window generally decreases the time-domain resolution of the VCE, whereas decreasing it generally decreases the reliability of any quantities calculated as part of the VCE. In other words, epoch-wise VCE produces more identifiable “spikes” that indicate exactly when problems in the dataset occur, whereas using larger time windows in the VCE produces more reliable estimates of the system’s actual performance.

5.7 Processing Workflow

The processing module of the PosAtt Engine described in Section 4.6 required modification to accommodate the modeling for multiple attitude models detailed from Sections 5.2 to 5.6. Specifically, the attitude model being used needed to be considered in the definition of the system state constraint equations, system model equations, and measurement model equations. The modified workflow for the processing module is shown in Figure 5.7.1.

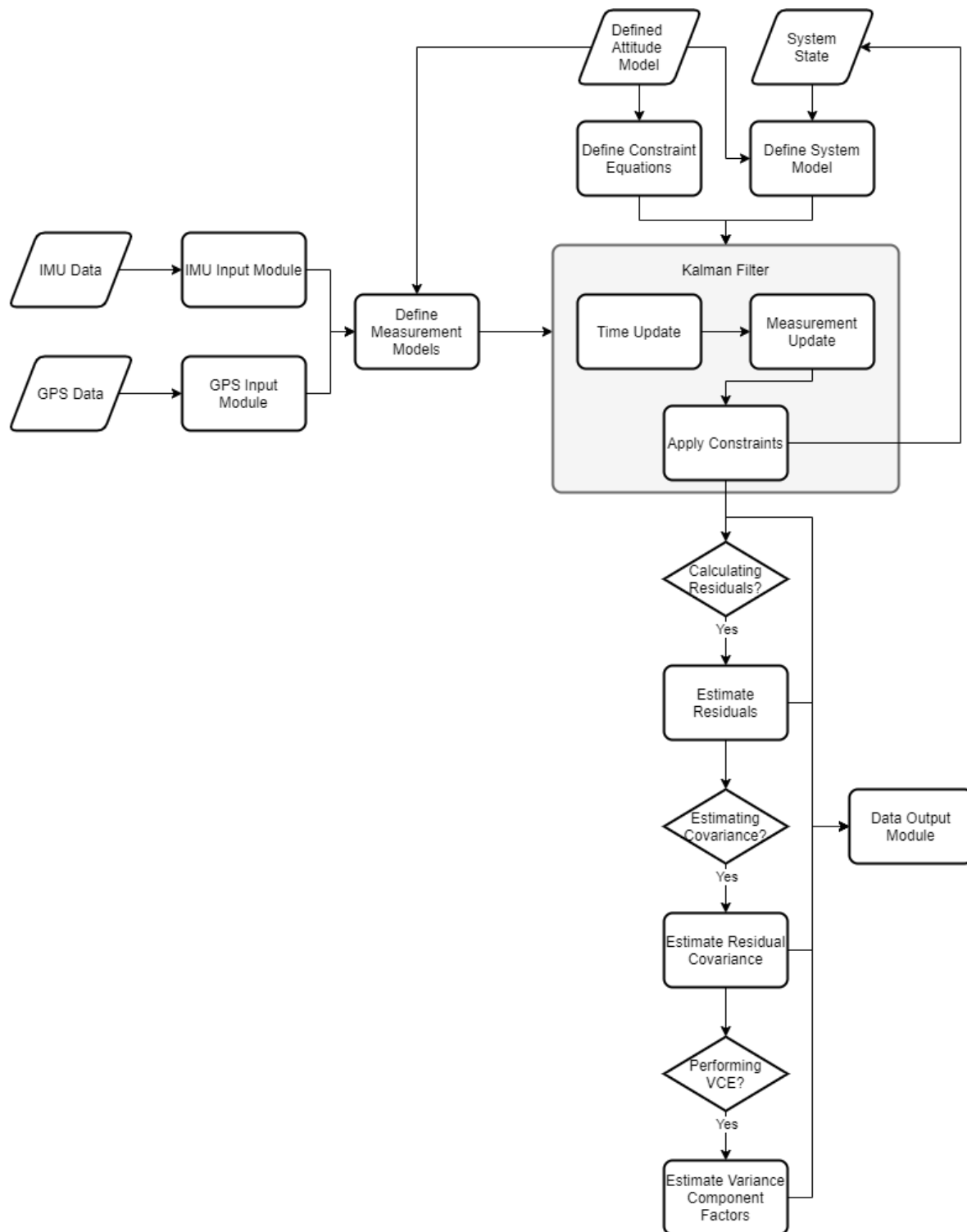


Figure 5.7.1: The modified processing workflow to accommodate different attitude models in the PosAtt Engine.

5.8 Comparative Analysis

This section presents the road test dataset used to validate the different attitude models used in this research, and provides a comparative analysis of the performance of each of the RPH, DCM, and Quaternion attitude models.

5.8.1 Definition of the Test Dataset

The road test dataset was collected from a land vehicle equipped with our in-house integrated navigation system consisting of an IMU 440 sensor operating at 100 Hz and a NovAtel OEM4 receiver operating at 1 Hz. A second NovAtel OEM4 receiver operating at 1 Hz was operated at a fixed location to act as a base station, enabling the generation of double-differenced measurements. The *a priori* standard deviation of the accelerometers was taken to be 0.1 m/s^2 , and the *a priori* standard deviation of the gyroscopes was taken to be $0.8^\circ/\text{s}$.

For the first 7.5 minutes of data collection, the vehicle was kept stationary, followed by 7.5 minutes of being driven in kinematic, and then another 5 minutes of being kept stationary. The horizontal trajectory that the vehicle followed is shown in Figure 5.8.1, and the velocity and acceleration profiles of the vehicle are shown in Figure 5.8.2 and Figure 5.8.3, respectively.

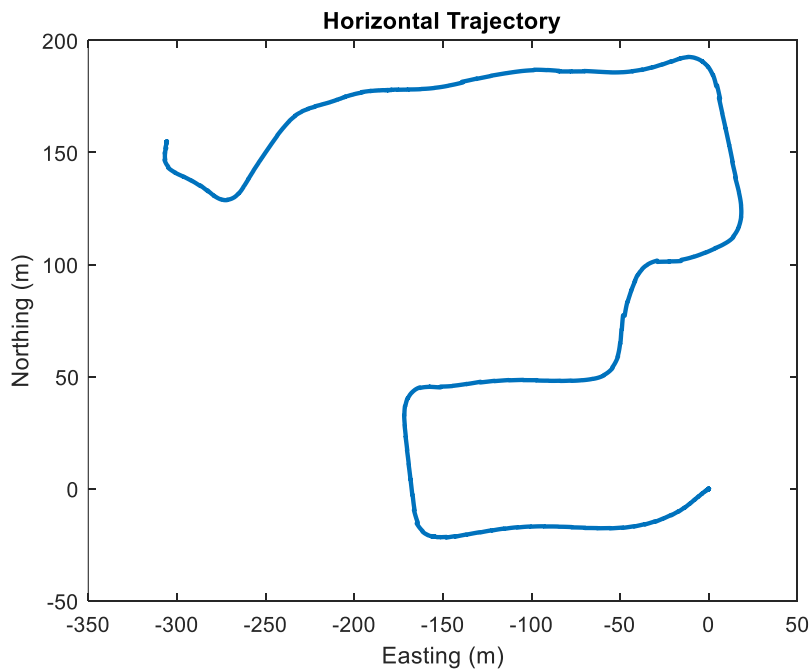


Figure 5.8.1: Horizontal trajectory of the road test dataset used to validate the different attitude models used in this research. Coordinates are local geodetic coordinates relative to the PosAtt system's starting location.

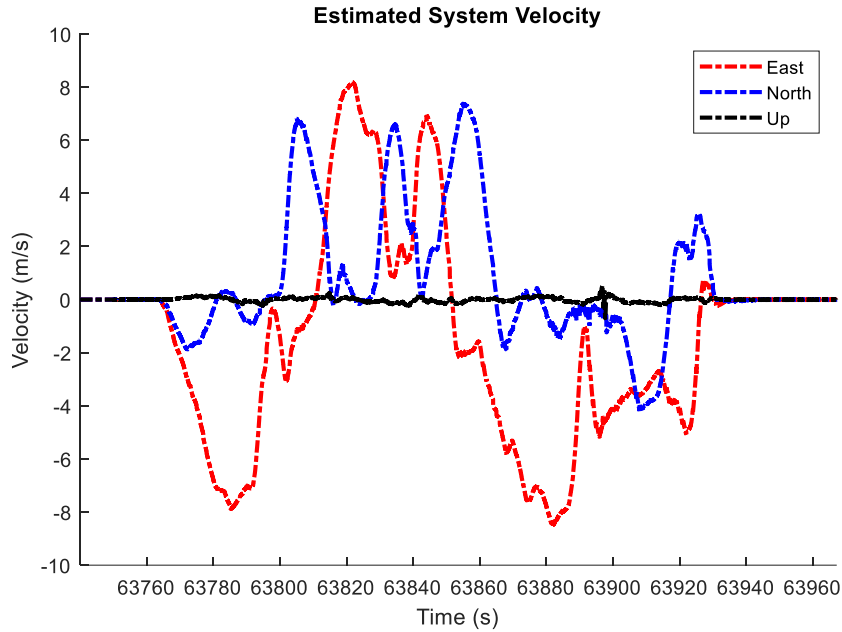


Figure 5.8.2: System velocity profile over the road test dataset. The velocity values are expressed in the navigation frame local geodetic coordinates.

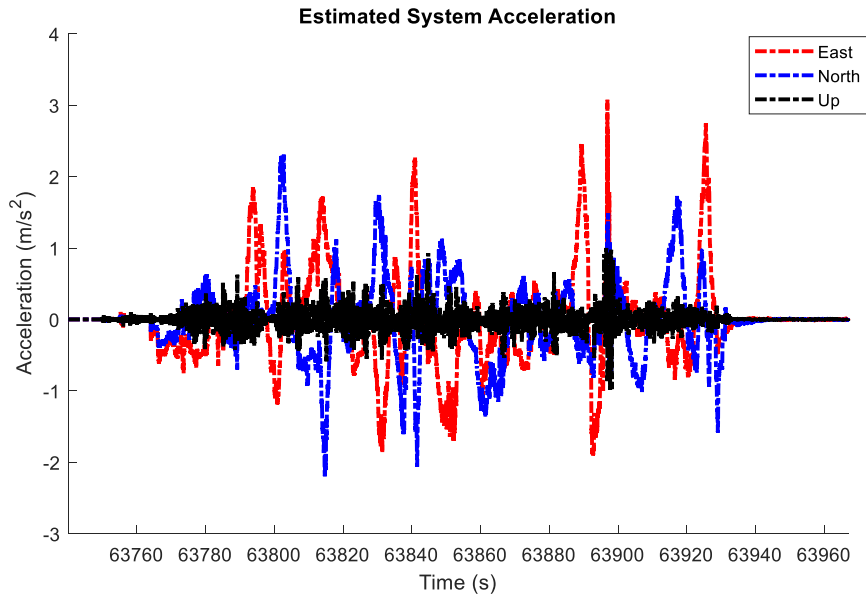


Figure 5.8.3: System acceleration profile over the road test dataset. The acceleration values are expressed in the navigation frame local geodetic coordinates.

Each of the attitude models was implemented under the GMIS to process the road test dataset. This is followed up with a comprehensive error analysis that directly compares the performance of the solutions associated with each different attitude model. This comparison is specifically focused on:

- i. Directly comparing the values of the state estimates. This is realized through estimating the RMSE between different position/attitude estimates, and provides insight into the overall consistency between the three different attitude models. This particular mode of comparison is possible under the traditional sensor integration strategy, and requires conversion of different attitude representations into a common system for comparison purposes.
- ii. Direct comparison of the *a posteriori* observation residuals when using each attitude model. Since the same dataset is being processed using different attitude models, the definition of the observation vectors is exactly the same across each attitude model. This allows their elements to be directly compared between attitude models and gives insight into the behaviour of sensor errors under each attitude model. This mode of comparison would be possible for GPS observations under the traditional sensor integration strategy, but is only made possible for the IMU observations under the GMIS.
- iii. Direct comparison of the *a posteriori* variance component factors for quantities in each attitude model, including the observation vectors and process noise vectors. Using the standard error factors provides a unitless quantity that allows direct comparisons to be made between attitude models without requiring any conversion between different attitude representations. This analysis is only made possible under the GMIS.

5.8.2 Directly Comparing System State Estimates

A summary of the average standard deviations for the system's position in the local horizontal and vertical is provided in Table 5.8.1. All attitude models achieved similar estimated accuracies in their linear positions, both in the horizontal and vertical. This is to be expected, since the position accuracy under any of the attitude models is primarily determined by the availability/geometry of the GPS observations.

Table 5.8.1: Average estimated horizontal and vertical standard deviations for each attitude model over the kinematic portion of the dataset.

Attitude Model	Average Estimated Horizontal Standard Deviation (cm)	Average Estimated Vertical Standard Deviation (cm)
RPH	4.48	6.69
DCM	4.65	6.91

Quaternion	4.76	6.86
-------------------	------	------

A summary of the RMSE between the kinematic portion of the position estimates using each attitude model is provided in Table 5.8.2. This table demonstrates that the position estimates are very consistent across the different attitude models, and these RMSE values are particularly reasonable considering the standard deviations presented in Table 5.8.1.

Table 5.8.2: The RMSE between the three-dimensional estimated positions for each attitude model, expressed in cm. These values were calculated over the kinematic portion of the dataset.

	RPH	DCM	Quaternion
RPH	0.00	4.57	3.84
DCM		0.00	3.95
Quaternion			0.00

A summary of the average estimated standard deviations in the system's orientation for each attitude model is provided in Table 5.8.3. To produce quantities that are directly comparable to one another, the DCM and quaternion attitude estimates were converted to their equivalent RPH representation. The estimated covariance matrix for the attitude parameters in the DCM and quaternion attitude models was then propagated through to the conversion to the equivalent RPH representation, and these propagated covariance estimates were used to generate the standard deviations in the system roll/pitch and the system heading. The average estimated standard deviations are very consistent across the different attitude models over the road test dataset.

Table 5.8.3: Average estimated standard deviations in the estimated system orientation in the roll/pitch and heading for each attitude model over the kinematic portion of the dataset.

Attitude Model	Average Standard Deviation in the Roll/Pitch (°)	Average Standard Deviation in the Heading (°)
RPH	6.6	20.4
DCM	7.2	21.6

Quaternion	6.6	19.2
-------------------	-----	------

A summary of the RMSE between the attitude estimates for each attitude model is provided in Table 5.8.4. The angle difference between the estimated orientation under each attitude model was determined by first converting the DCM and Quaternion orientation estimates to the RPH attitude model. These values were then be used to determine the corresponding axis of rotation as a unit vector. The dot product between the axes of rotation estimated under each attitude model was used to calculate the angular difference between them via the equation

$$\theta_{12} = \cos^{-1} \mathbf{u}_1 \cdot \mathbf{u}_2 \quad (5.80)$$

where θ_{12} describes the angular separation between the unit rotation axis vector $\mathbf{u}_1$ and the unit rotation axis vector $\mathbf{u}_2$.

Table 5.8.4: RMSE between the attitude estimates for each attitude model, expressed in arcminutes. These values were calculated over the kinematic portion of the dataset.

	RPH	DCM	Quaternion
RPH	00.0	21.8	21.2
DCM		00.0	28.3
Quaternion			00.0

The estimated average angular separation between the different attitude models' orientation estimates is consistent with the standard deviations presented in Table 5.8.3. Disregarding needs for software efficiency and degenerate cases such as Gimbal Lock, each of the three attitude models developed for this research may be used interchangeably over the road test dataset. Since this research involved developing the models for each of the attitude models under the GMIS, establishing this consistency is important for validating the models used in this research.

It is important to note that these results only establish the consistency of the attitude models over the road test dataset, where only the system heading changed appreciably over the course of data collection. To conclusively determine the relative quality of each attitude model a system with more pitch/roll dynamics, such as an aerial system, would be necessary. This falls outside the scope of this research but is an important future research task.

5.8.3 Analyzing Observation Residuals

The observation residuals were estimated and separated by observation type to compare how they behaved across all different attitude models used in this research. These observation residuals were separated into:

- i. L1 C/A double-differenced observation residuals, with the measurement residuals' scatter plot shown in Figure 5.8.4 and the histogram of their standardized residual values shown in Figure 5.8.5.
- ii. L1 Carrier Phase double-differenced observation residuals, with the measurement residuals' scatter plot shown in Figure 5.8.6 and the histogram of their standardized residual values shown in Figure 5.8.7.
- iii. L2 Carrier Phase double-differenced observation residuals, with the measurement residuals' scatter plot shown in Figure 5.8.8 and the histogram of their standardized residual values shown in Figure 5.8.9.
- iv. Accelerometer specific force observation residuals, with the measurement residuals' scatter plot shown in Figure 5.8.10 and the histogram of their standardized residual values shown in Figure 5.8.11.
- v. Gyroscope angular rate observation residuals, with the measurement residuals' scatter plot shown in Figure 5.8.12 and the histogram of their standardized residual values shown in Figure 5.8.13.

While the positioning Kalman Filter does not require data to be normally-distributed to still provide optimal estimates of the system state, the most commonly-applied statistical tests do assume normally-distributed data, for example for fault detection and exclusion and for overall quality assurance on the estimated variance components. Plotting the histograms of the standardized residual values therefore provides an important means to validating the applicability of many of the statistical tests at the core of a comprehensive error analysis.

It is worth noting that under the traditional integration strategy, the IMU accelerometer and gyroscope residuals would not be estimable. These residual values are only estimable when the measurements are being directly modeled through the GMIS. The IMU measurement residuals are necessary for estimating any observation variance components related to the IMU observations, which is central to the comparative analysis being conducted in this chapter.

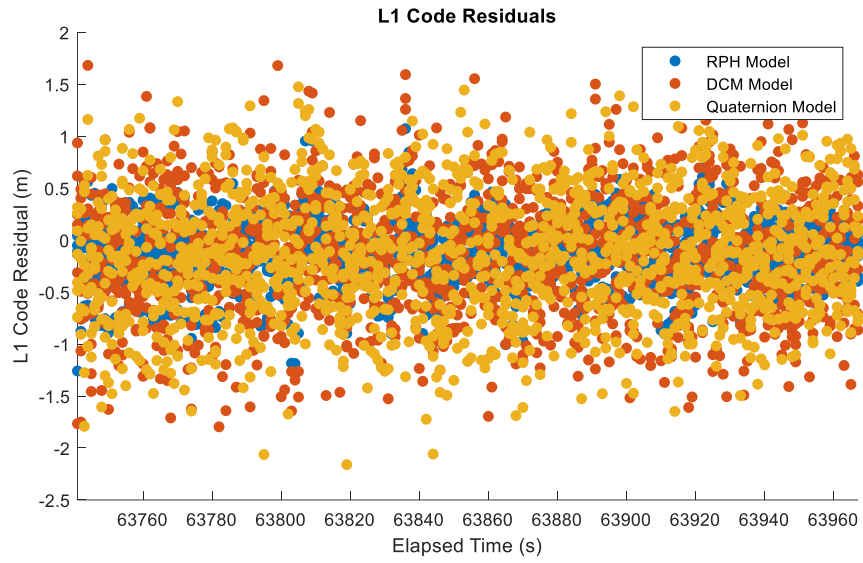


Figure 5.8.4: L1 C/A residuals for each attitude model.

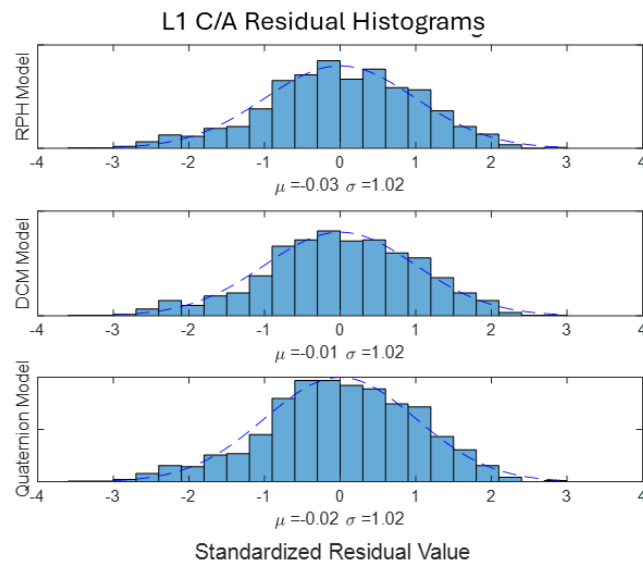


Figure 5.8.5: Histogram of L1 C/A standardized residuals. Standard normal distribution superimposed for reference.

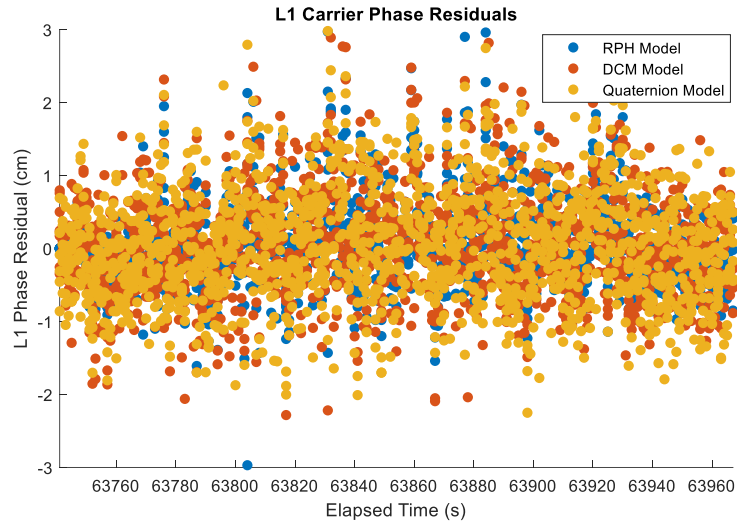


Figure 5.8.6: L1 Carrier Phase residuals for each attitude model.

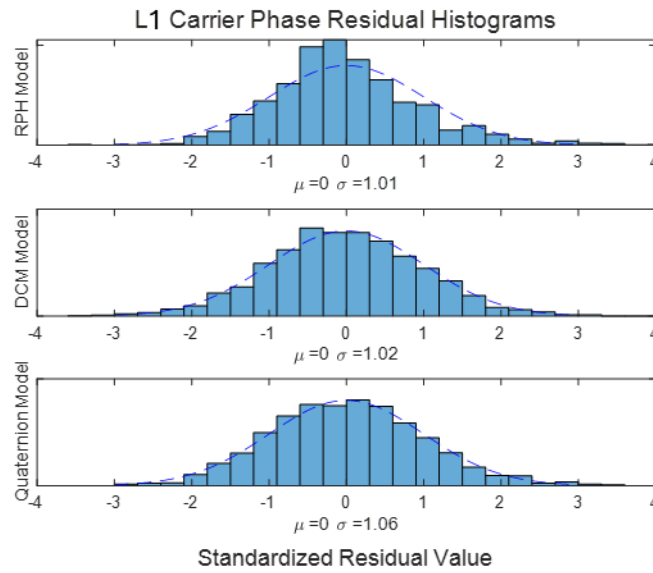


Figure 5.8.7: Histogram of L1 Carrier Phase standardized residuals. Standard normal distribution superimposed for reference.

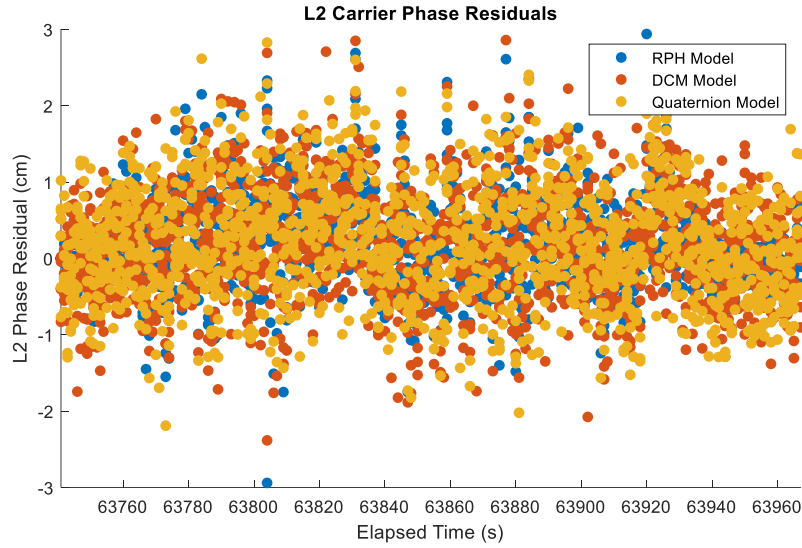


Figure 5.8.8: L2 Carrier Phase residuals for each attitude model.

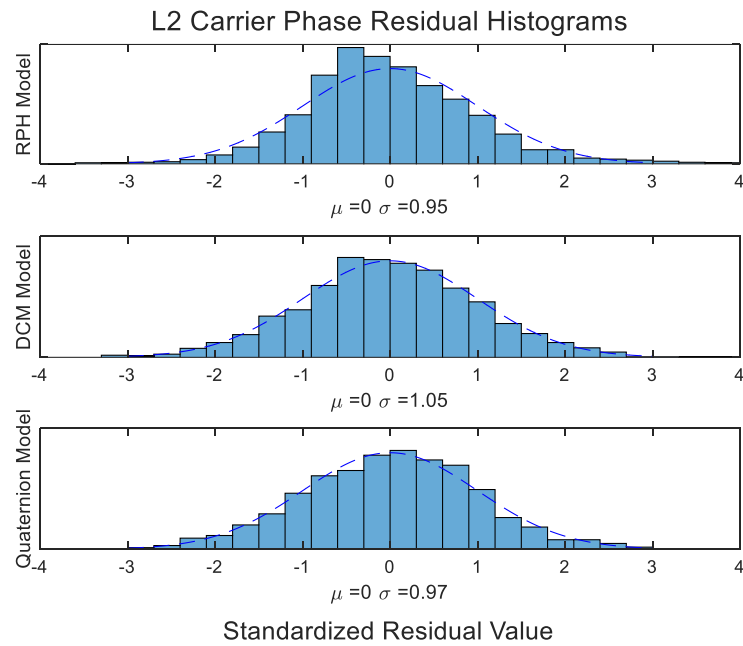


Figure 5.8.9: Histogram of L2 Carrier Phase standardized residuals. Standard normal distribution superimposed for reference.

The histograms of the GPS measurement residuals in Figure 5.8.5, Figure 5.8.7, and Figure 5.8.9 clearly show that there are no persistent biases in the GPS observations over the road test dataset. Each of the histograms fit well to the standard normal distribution, which suggests that their stochastic properties were properly modeled in the positioning Kalman Filter.

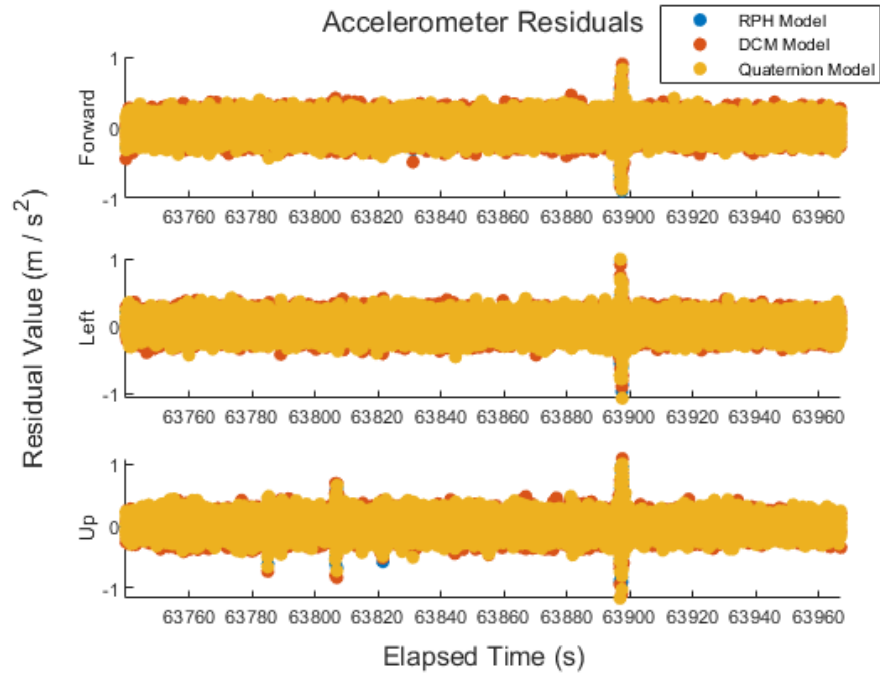


Figure 5.8.10: Accelerometer specific force residuals for each attitude model.

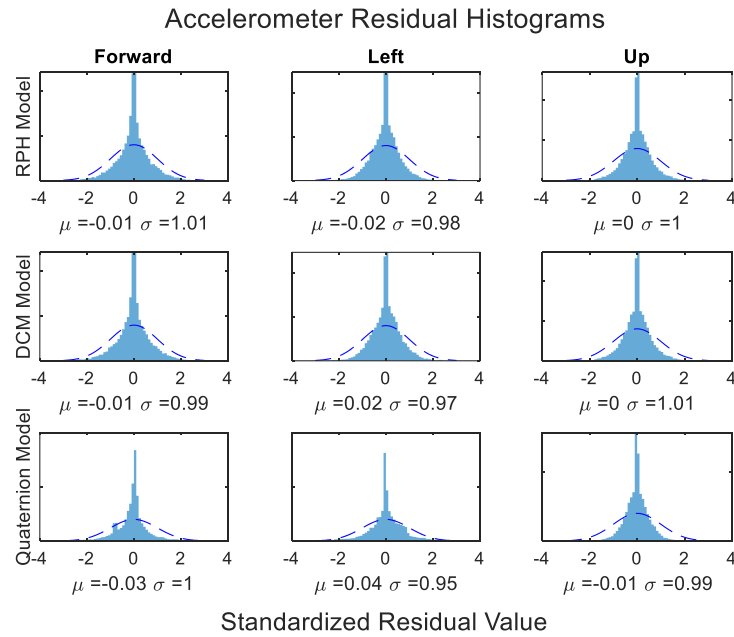


Figure 5.8.11: Histogram of accelerometer standardized residuals. Standard normal distribution superimposed for reference.

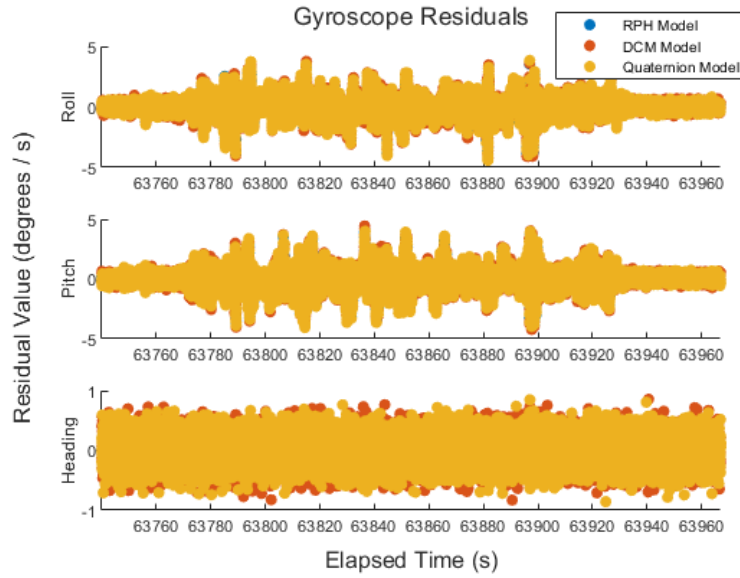


Figure 5.8.12: Gyroscope angular rate residuals for each attitude model.

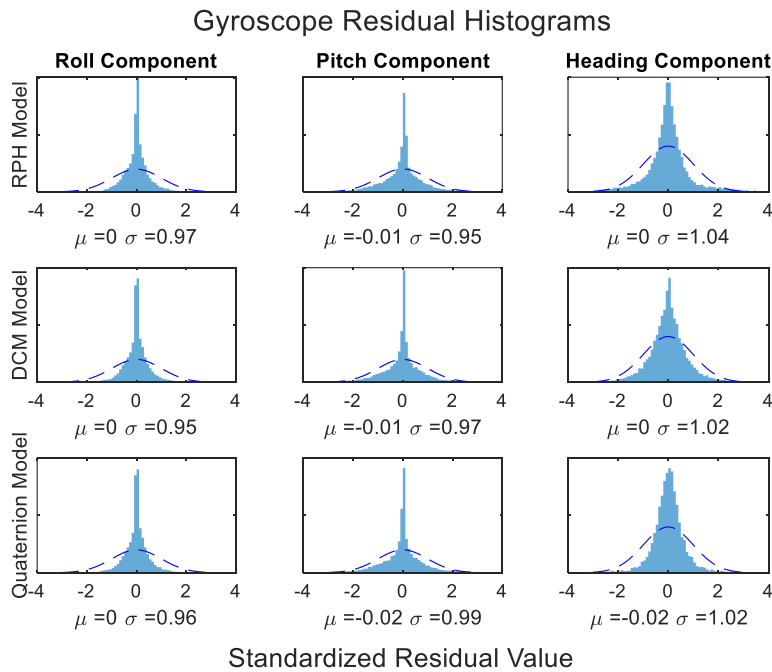


Figure 5.8.13: Histograms of gyroscope standardized residuals. Standard normal distribution superimposed for reference.

The accelerometer standardized measurement residual histograms in Figure 5.8.11 and the gyroscope standardized measurement residual histograms in Figure 5.8.13 also show that there are no persistent unmodeled biases in the IMU observations over the road test dataset. These IMU residual histograms all exhibit noticeable spikes near zero. The near-zero values largely come from periods in the

dataset where the system was experiencing lower-dynamics (i.e. at rest, or moving at a consistent velocity), and these periods generally correspond with smaller IMU measurement errors.

5.8.4 Analyzing Process Noise Residuals

The components of the estimated process noise residual vector are of particular importance in characterizing the performance of the different kinematic models used for each alternative attitude model. Here, it is important to bear in mind that each attitude model has a different definition of the process noise vector. Since all attitude models make use of the same underlying kinematic model of the linear motion of the system, the system jerk (i.e. third-order position time derivatives) estimates are directly comparable between the different attitude models. The system jerk residuals are shown in Figure 5.8.14 and their associated standardized residual histograms are shown in Figure 5.8.15. Since all attitude models use the second-order time derivatives of the attitude parameters to define the process noise of their angular motion, each attitude model has a different number of elements in their angular process noise vectors. This means that the estimated process noise residuals shown in Figure 5.8.16 are not directly comparable across the different attitude models. Figure 5.8.16 therefore provides limited useful information, but is still valuable in generating the attitude process noise standardized residual histograms in Figure 5.8.17, Figure 5.8.18, and Figure 5.8.19.

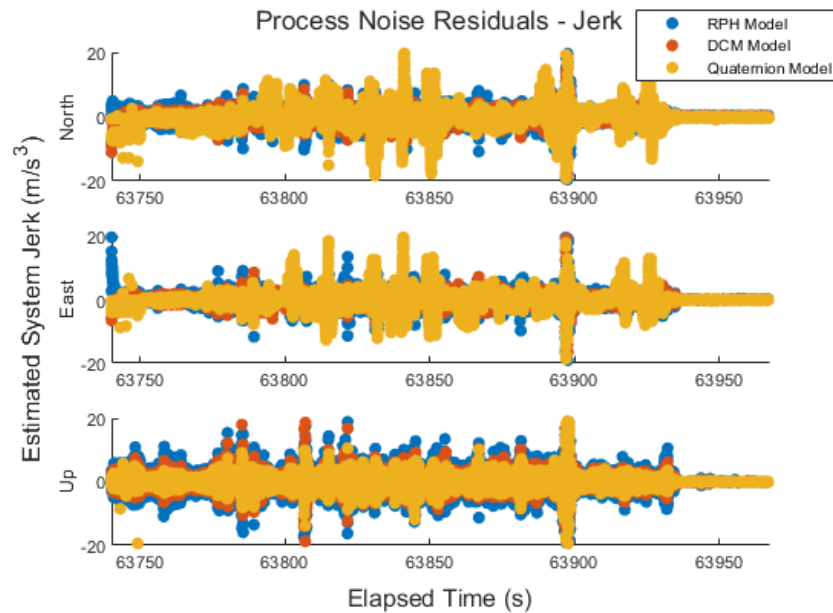


Figure 5.8.14: System jerk residuals (process noise of the system's linear position).

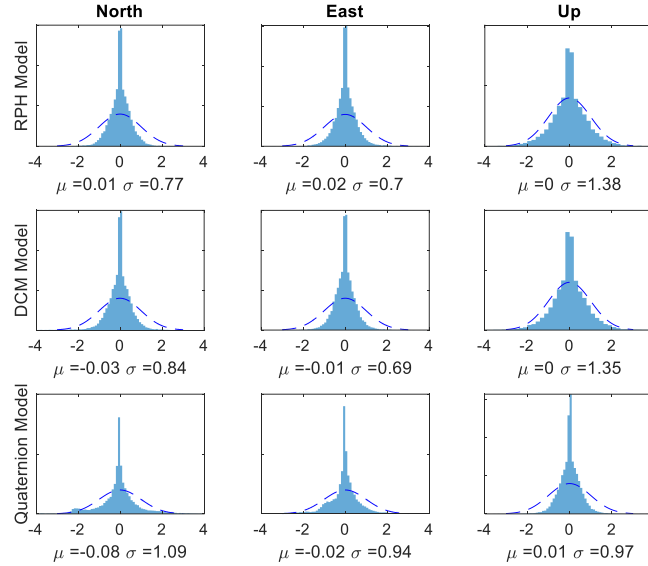


Figure 5.8.15: Histograms of the standardized system jerk process noise residuals.

The system jerk process noise residuals exhibit similar behaviour across each of the different attitude models. This similarity is clearly shown in Figure 5.8.15, with the RPH and DCM attitude models behaving particularly consistently with one another. The quaternion attitude model has notably larger standard deviations in the horizontal system jerk estimate values than for the RPH and DCM models, but also has notably lower standard deviations in the vertical system jerk estimate values.

As was the case with the IMU measurement residual histograms, the system jerk histograms in Figure 5.8.15 exhibit strong spikes near zero. Again, these values frequently coincide with periods of time in which the system was either stationary or moving at a near-constant velocity. It is generally reasonable to expect process noise to have different stochastic properties under different dynamic conditions, and it makes sense that the estimated third-order linear motion of the system should be smaller when the system is moving at near-constant velocity.

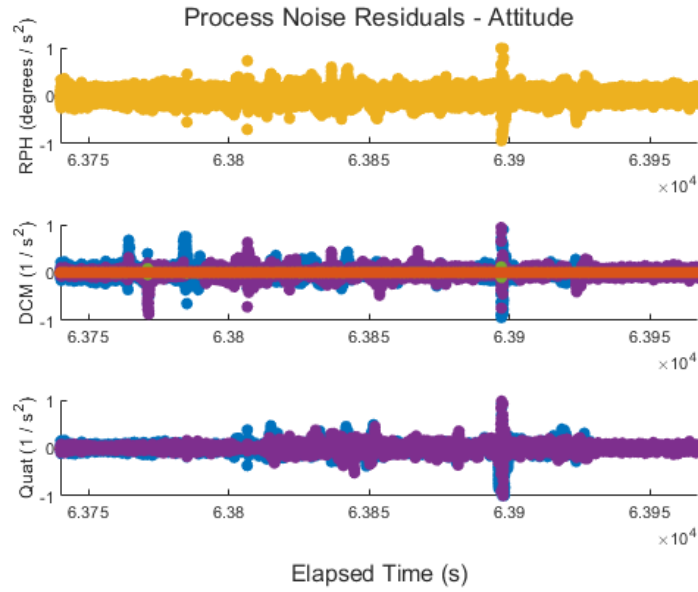


Figure 5.8.16: Estimated process noise residuals for the attitude parameters in each of the attitude models. Each colour represents a different second-order time derivative in the associated attitude model, and none of the values are directly comparable across attitude models.

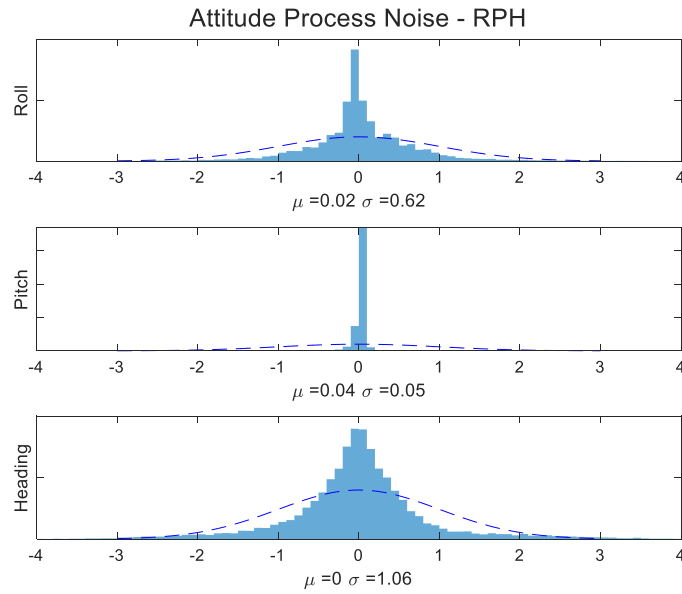


Figure 5.8.17: Histograms of the standardized attitude process noise residuals for the RPH attitude model.

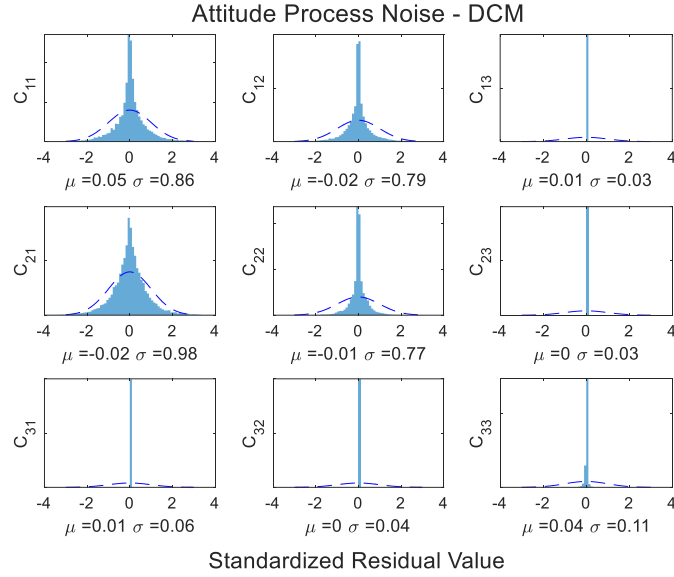


Figure 5.8.18: Histograms of the standardized attitude process noise residuals for the DCM attitude model.

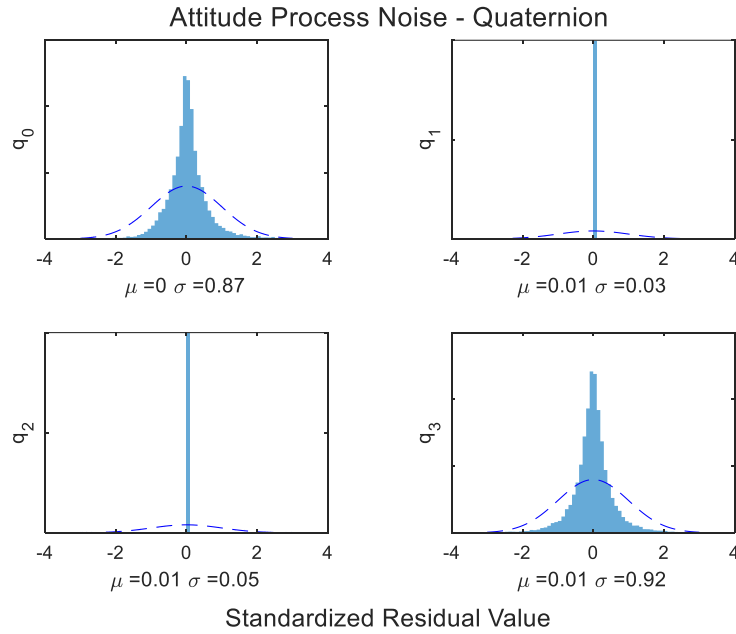


Figure 5.8.19: Histograms of the standardized attitude process noise residuals for the quaternion attitude model.

The attitude process noise residuals have some interesting patterns. In all of Figure 5.8.17, Figure 5.8.18, and Figure 5.8.19, there are multiple attitude process noise elements that have standardized values almost exclusively at or near zero. To understand why this should be the case, it is important to bear in mind that the road test dataset was collected from a ground vehicle, where the heading changed significantly over the course of the dataset, but the roll/pitch remained almost

constant. Given this physical constraint on the system's motion, the second-order time derivatives of the attitude parameters should generally only be significant for the attitude parameters that relate to the system's heading.

For the RPH attitude model's histograms in Figure 5.8.17, this clearly results in a larger spread of values for the heading second-order time derivatives than for the roll/pitch second-order time derivatives.

For the DCM attitude model's histograms in Figure 5.8.18, this relationship is less immediately obvious. Considering that the PosAtt system was approximately level over the course of the road test dataset (i.e. with roll and heading angles of 0°), the rotation from the navigation frame to the body frame may be approximated as

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (5.81)$$

From this approximation of the rotation matrix, it is clear that elements C_{11} , C_{12} , C_{21} , and C_{22} in the DCM attitude model vary significantly with changing heading angle and that all other attitude parameters in the DCM attitude model do not. This is why all of the histograms for elements C_{13} , C_{23} , C_{31} , C_{32} , and C_{33} in Figure 5.8.18 have exclusively near-zero values over the course of the dataset.

For the quaternion attitude model's histograms in Figure 5.8.19, the attitude parameters that relate to the system's heading are also less immediately apparent. Again, taking the PosAtt system to be approximately level over the course of the road test dataset, the axis of rotation between the navigation frame and the body frame may be considered to be aligned with the local vertical (i.e. the unit rotation axis vector is identical to the unit basis vector in the Z-axis). Combining this information with the relationship between the quaternion model and the rotation axis, the quaternion model may be approximated as

$$\mathbf{q} = q_0 + \mathbf{q} = \cos \frac{\gamma}{2} + \mathbf{u} \sin \frac{\gamma}{2} = \cos \frac{\gamma}{2} + \mathbf{k} \sin \frac{\gamma}{2} \quad (5.82)$$

where element $q_0 = \cos \frac{\gamma}{2}$ and element $q_3 = \sin \frac{\gamma}{2}$. From this approximation, it is clear that elements q_1 and q_2 do not vary significantly as the heading angle changes, and this is why the histograms for elements q_1 and q_2 in Figure 5.8.19 have exclusively near-zero values over the course of the dataset.

Beyond the characteristically different behaviours of the process noise residuals related to the non-heading-related attitude parameters, the standardized process noise residual histograms all exhibit similar peaks near zero as for the system jerk residual histograms. The reason for these peaks is similar as well, since the times at which the system was actively turning were accompanied by higher process noise levels for the system's angular motion, and process noise generally varies with the system's dynamics.

5.8.5 Analyzing Variance Component Estimates

The variance factors (i.e. the individual variances of unit weight) were estimated regionally over a moving time period of 20 seconds to provide time-varying measures of the performance of the different constituent parts of each positioning solution. This time period could be increased to allow more degrees of freedom to better estimate the overall performance of each variance component, or could be decreased to improve the time-resolution and obtain a more precise estimate of when issues arise in the positioning solution. The effects of the choice of window size is discussed in more detail in Section 3.6.4 and is also briefly discussed in Section 5.8.6 for this dataset. In this research, all plots show the standard error factor, which is the square root of the variance factor.

The overall *a posteriori* observation standard error factor plots for each attitude model are shown in Figure 5.8.20 and the overall *a posteriori* process noise standard error factor plots for each attitude model are shown in Figure 5.8.21.

Both the observation and process noise standard error factor estimates are very consistent across the different attitude models. At the beginning and end of the dataset, the standard error factors generally have much lower values. This corresponds to periods of time when the system was stationary. This behaviour is to be expected for the process noise, since it represents the higher-order motion of the system, which should be zero for a stationary system. This behaviour also makes sense for the observation standard error factors, since observation quality can be negatively impacted by the motion of the PosAtt system, for example by the sensors' vibrations as the system moves.

It is worth noting that even the relatively minor fluctuations in the standard error factor estimates are very consistent between the different attitude models, and that the shapes of each plot in Figure 5.8.20 and Figure 5.8.21 are very consistent throughout the whole dataset.

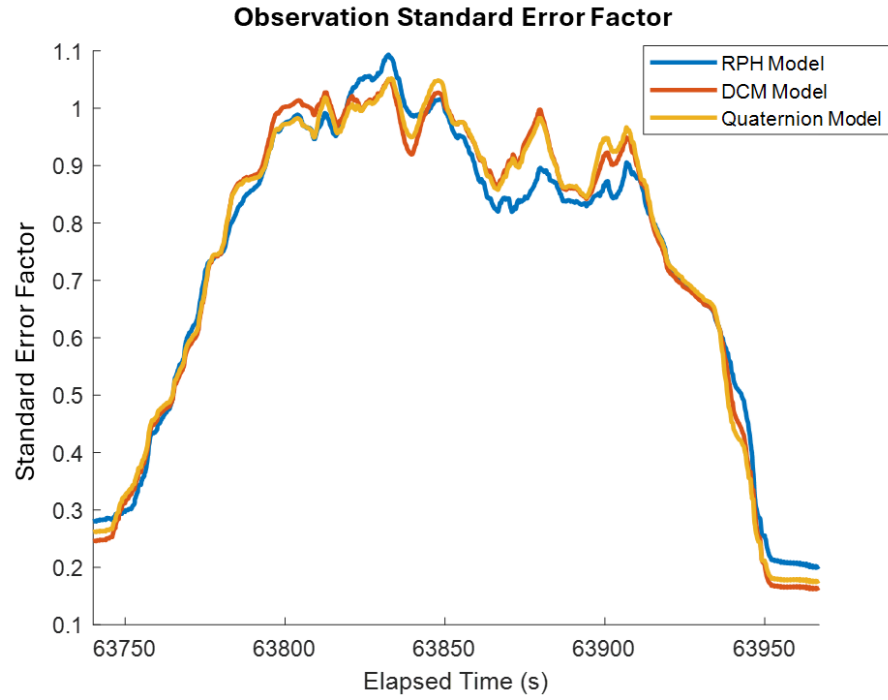


Figure 5.8.20: The standard error factor plots for the observation residual vector in the RPH, DCM, and Quaternion attitude models (moving window: 20 seconds).

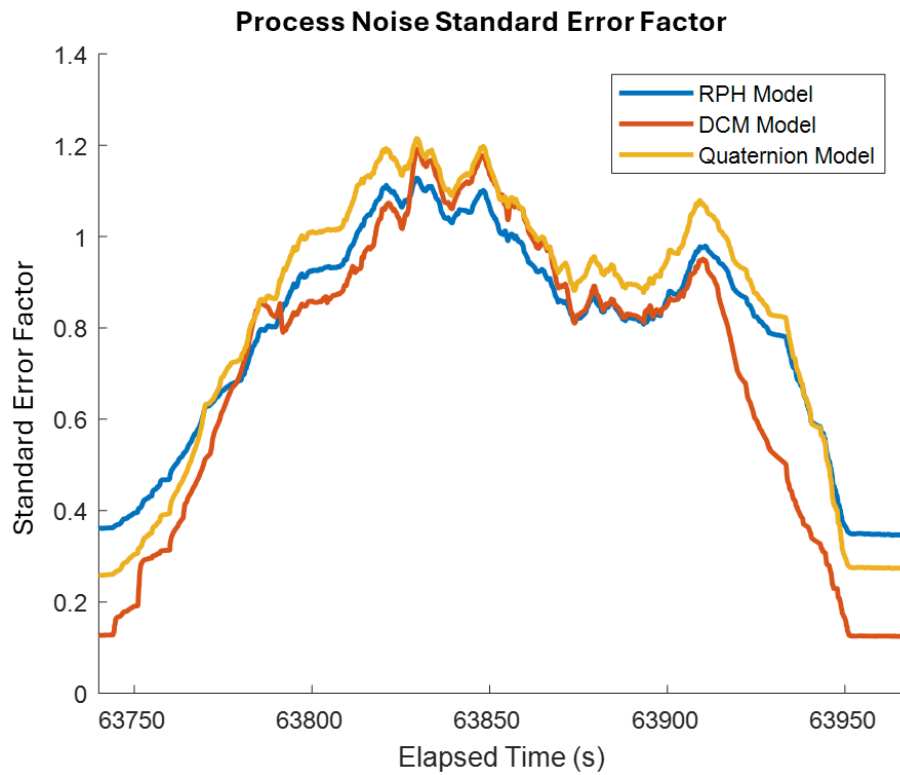


Figure 5.8.21: The standard error factor plots for the process noise residual vector in the RPH, DCM, and Quaternion attitude models (moving window: 20 seconds).

It is possible to get a single representative value of the standard error factor for the entire dataset using the cumulative redundancy contributions and weighted sum of the squared residual vectors. Table 5.8.5 provides the standard error factors, estimated over the mobile portion of the dataset. Since the standard error factor represents the ratio of the *a posteriori* estimated accuracy to the *a priori* assumed accuracy, having values near one indicate that there are no significant scale defects in the observation covariance matrix. Selecting appropriate *a priori* accuracies for a diverse set of observations is particularly important, since each observation type may have a different scale defect, and the *a posteriori* estimates of the observation accuracy are less reliable when there are significant scale defects in the observation covariance matrix. The standard error factors in Table 5.8.5 show that the *a priori* accuracies were selected appropriately for the dataset. Since these values were generally aligned with the stated instrument accuracies, this means that the PosAtt system largely performed very consistently and reliably.

The estimated standard error factors are very consistent across the different attitude models. This is to be expected if each model is properly formulated, since the standard error factors are generated using exactly the same set of observations across the different attitude models.

Table 5.8.5: Overall estimated standard error factors for each type of measurement, and for all observations together. Results are presented separately for each attitude model, for comparison. Each standard error factor was evaluated solely over the kinematic portion of the dataset.

	Standard Error Factors					
Attitude Model	Accelerometer	Gyroscope	L1 C/A	Carrier Phase		Overall
				L1	L2	
RPH	1.0154	1.0132	1.0255	1.0236	0.9955	1.0026
DCM	0.9613	1.0097	1.0256	1.0512	1.0032	0.9966
Quaternion	0.9646	0.9921	1.0254	1.0601	0.9648	0.9789

The *a posteriori* standard deviations of each observation type may be determined by scaling their assumed *a priori* standard deviations by the associated standard error factor, and these estimated *a posteriori* standard deviations are provided in Table 5.8.6. These values are obviously also near-

identical across the different attitude models, since they are derived via scaling the *a priori* observation accuracies by the values in Table 5.8.5, which are also near-identical.

Table 5.8.6: Overall estimated a posteriori standard deviations for each type of measurement. A posteriori standard deviations were estimated by scaling the a priori standard deviation estimate of the observation accuracy by its associated standard error factor from Table 5.8.5.

Attitude Model	A Posteriori Standard Deviations				
	Accelerometer	Gyroscope	L1 C/A	Carrier Phase [cm]	
	[m/s ²]	[°/s]	[m]	L1	L2
RPH	0.102	0.811	0.513	0.512	0.498
DCM	0.096	0.808	0.513	0.526	0.502
Quaternion	0.096	0.794	0.513	0.530	0.482

Just as with the observations, it is possible to generate a single value that describes the process noise standard errors of unit weight via the cumulative redundancy contributions and weighted sum of the squared process noise residual vector. The global estimates of the process noise standard error factors across the different attitude models are summarized in Table 5.8.7.

The values in Table 5.8.7 are once again very consistent across the different attitude models, with the global standard error factors all converging to approximately one.

Table 5.8.7: Overall estimated standard error factors for each process noise component related to the system's motion. Separate standard errors were estimated for the components of the process noise vector that were pertaining to the system's linear motion and those pertaining to the system's angular motion. Note that the components of the process noise vector related to the wander of IMU systematic errors and float GPS ambiguity estimates are not included in this table, but did contribute to the overall standard error estimate.

Attitude Model	Standard Errors of Unit Weight ($\hat{\sigma}_0$)		
	Jerk Vector	Angular Acceleration Vector	Overall Process Noise Vector
RPH	0.9231	0.9580	0.9718

DCM	0.8975	0.9216	0.9329
Quaternion	0.9012	0.9289	0.9826

As with the observation standard error factors, the values in Table 5.8.7 may be used to scale the *a priori* process noise standard deviations to determine their *a posteriori* estimates, and these values are presented in Table 5.8.8. Again, these values show a consistency across the different attitude models, which indicates that they all perform very similarly for the duration of the kinematic dataset.

Table 5.8.8: Overall estimated a posteriori estimates of the process noise jerk and angular acceleration components, calculated by scaling their assumed a priori values by the associated standard error factors in Table 5.8.7.

Attitude Model	A Posteriori Standard Deviations	
	Jerk [m/s³]	Angular Acceleration [°/s²]
RPH	9.231	0.958
DCM	8.975	0.922
Quaternion	9.012	0.929

It is important to note that, while the values in Table 5.8.8 are considered representative of the average performance of each attitude model over the entire kinematic portion of the dataset, the process noise's stochastic properties can vary significantly with changing system dynamics. This means that, while the global *a posteriori* standard deviations provide valuable information about the performance of the PosAtt system, it is important to also have their time-varying estimates as shown in Figure 5.8.21.

The results here are very encouraging and suggest that each of the attitude models was formulated correctly. Since a large part of this research was focused on developing these models in the GMIS, establishing their consistency for relatively low-dynamic data devoid of degenerate motions is important.

In order to make evaluative statements on the relative quality of each attitude model, additional tests are necessary. Ideally, to make conclusive statements, these tests should include a wide range of

different dynamic conditions (including more dynamic roll/pitch values), different observation geometries and sensor configurations, and different observing conditions. This is considered beyond the scope of this dissertation due to physical limitations and budgetary constraints. Section 5.8.6 does explore how each attitude model responds to the sudden introduction of outlying observations via a jolt that the PosAtt system experienced in this test dataset.

5.8.6 Analyzing the System's Response to a Jolt

In the preceding Sections, the consistency between the different attitude models has been established, and it appears that all three attitude models perform well under the normal operating conditions of a ground vehicle. In the context of drawing comparisons between the performance of each attitude models, it is important to evaluate how each model responds to “stress,” and how they cope with outlying measurements.

In the road test dataset, the system was jolted at approximately the 63900 second mark, as illustrated by plotting the raw high-rate IMU measurements in Figure 5.8.22. This jolt impacted IMU measurements for a duration of approximately 0.5 seconds. The impacted observations were successfully identified as outliers and removed for all attitude models in the preceding sections, and so the generated results were not affected by the jolt that the system experienced. All results in this section purposefully use those observations, so that the strain they place on each attitude model can be investigated more thoroughly.

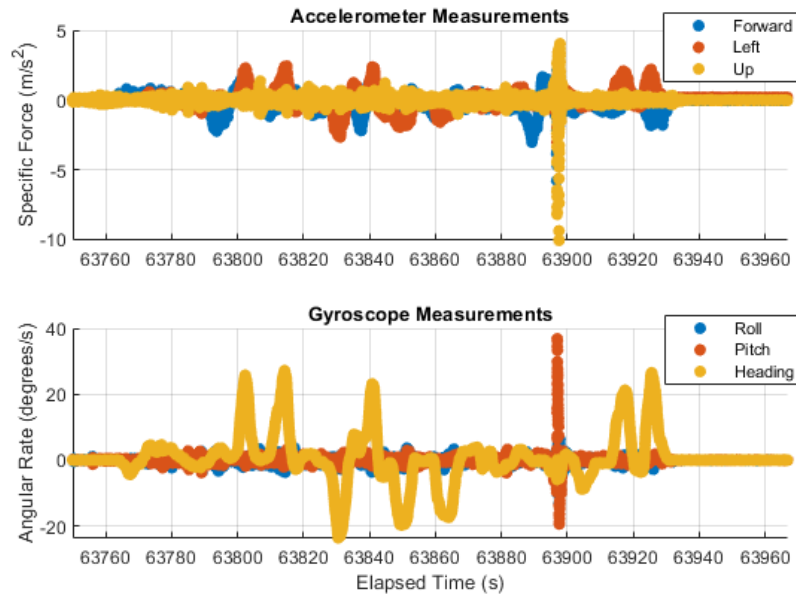


Figure 5.8.22: IMU raw measurement plots over the duration of the road test dataset. Note the sharp jumps in both the accelerometer and gyroscope measurements at approximately $t = 63987$ s.

The system jolt at $t = 63987$ s is easily noticeable in Figure 5.8.22 as a sharp jump in all components of the IMU accelerometer and gyroscope measurements. The visual effect of this jump is very unnatural compared to the behaviour of the rest of the dataset, and the accelerometer values in particular go beyond the range of reasonably-expected values.

In Section 5.8.5, the standard error factor plots only show estimates for the overall observation and process noise standard error factors. In this section, the standard error factor plots are shown for all of the individual observation types and the linear/angular process noise components. This more granular information allows for a more thorough comparison to be made on the impact of the system jolt on the different constituent parts of the positioning Kalman Filter.

The estimated overall observation standard error factor plots are shown in Figure 5.8.23, with specific estimates for the gyroscope angular rate observations shown in Figure 5.8.24 and accelerometer specific force observations shown in Figure 5.8.25. The GPS observation standard error factor plots are shown in Figure 5.8.26, Figure 5.8.27, and Figure 5.8.28 for the L1 C/A, L1 Carrier Phase, and L2 Carrier Phase observations, respectively.

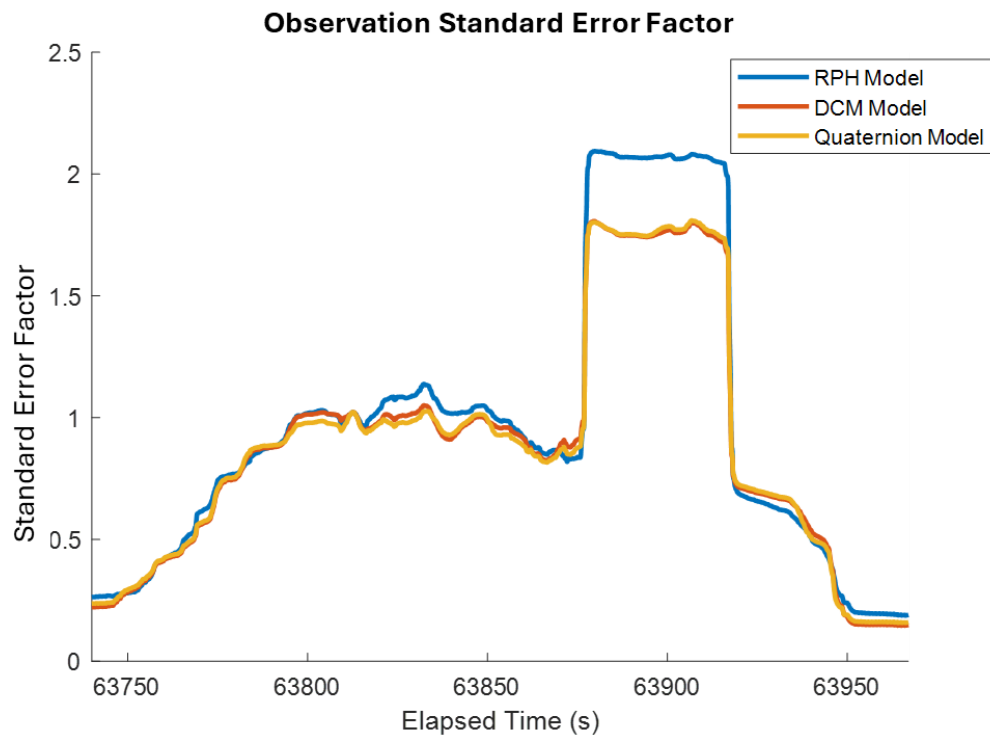


Figure 5.8.23: Estimated observation standard error factors for each attitude model (moving window: 20 s). Note the sharp jump in all attitude models around the time $t = 63900$ s.

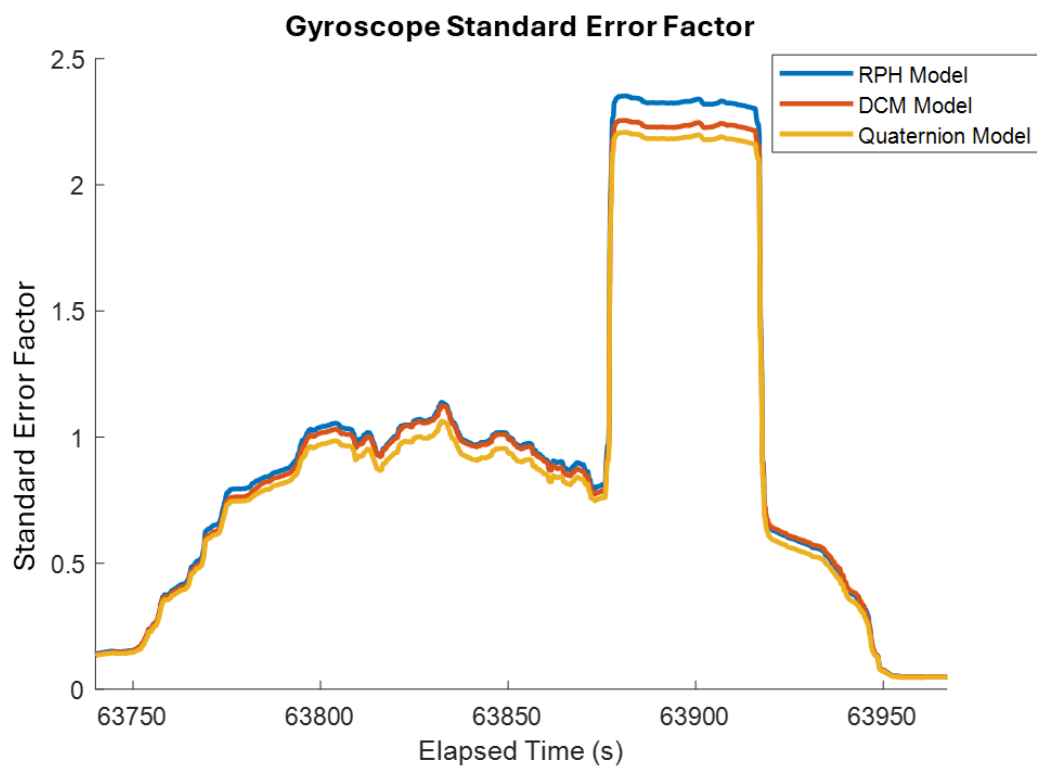


Figure 5.8.24: Estimated gyroscope observation standard error factors for each attitude model (moving window: 20 s).

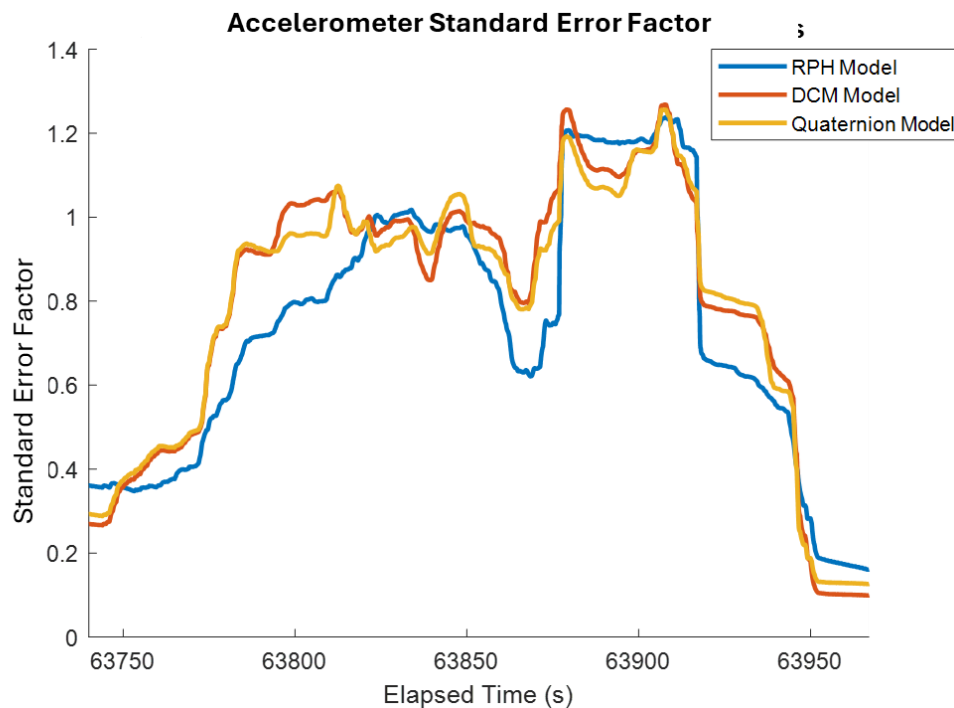


Figure 5.8.25: Estimated accelerometer observation standard error factors for each attitude model (moving window: 20 s).

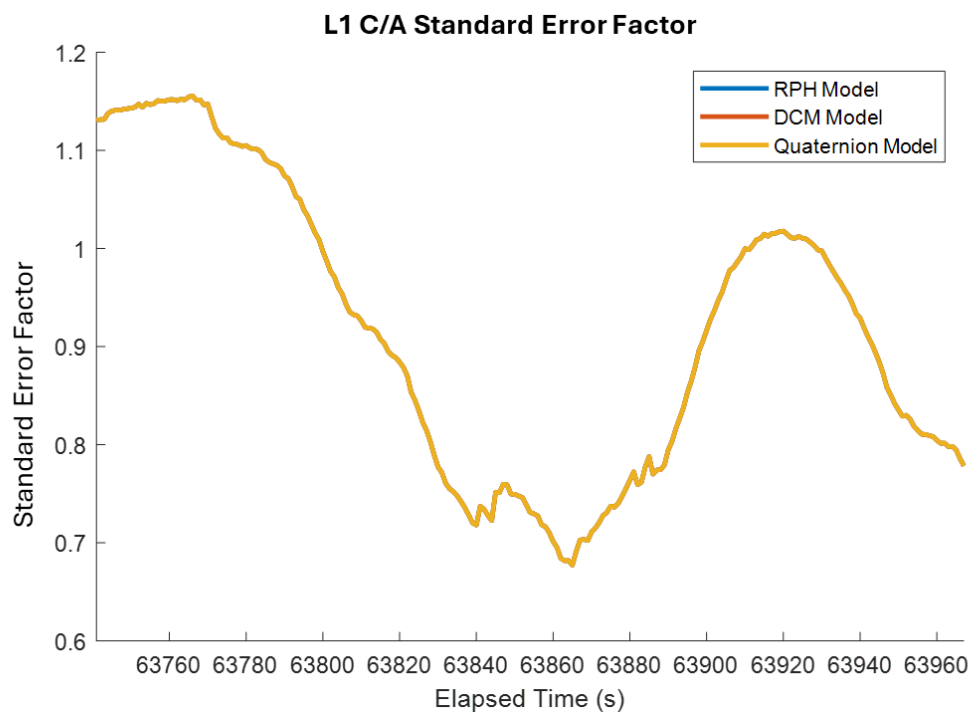


Figure 5.8.26: Estimated L1 C/A observation standard error factors for each attitude model (moving window: 20 s).

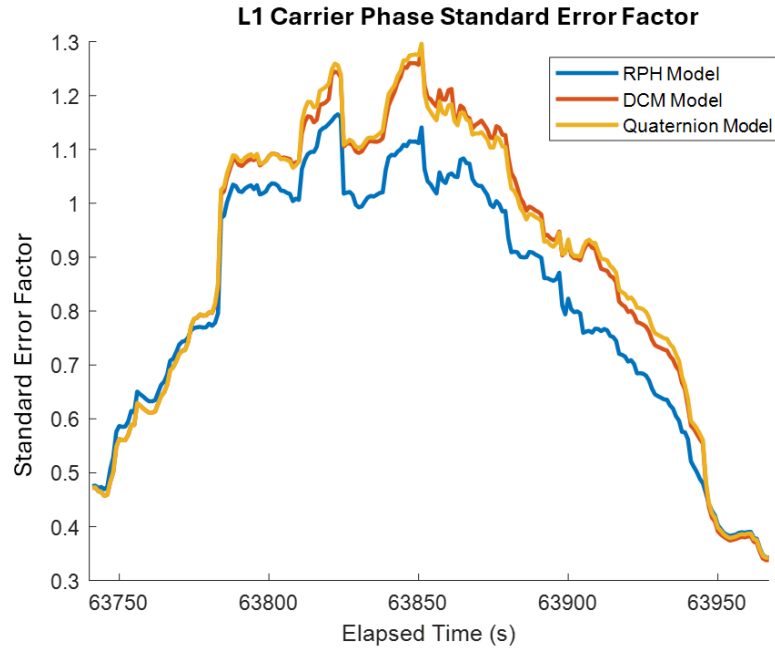


Figure 5.8.27: Estimated L1 carrier phase observation standard error factors for each attitude model (moving window: 20 s).

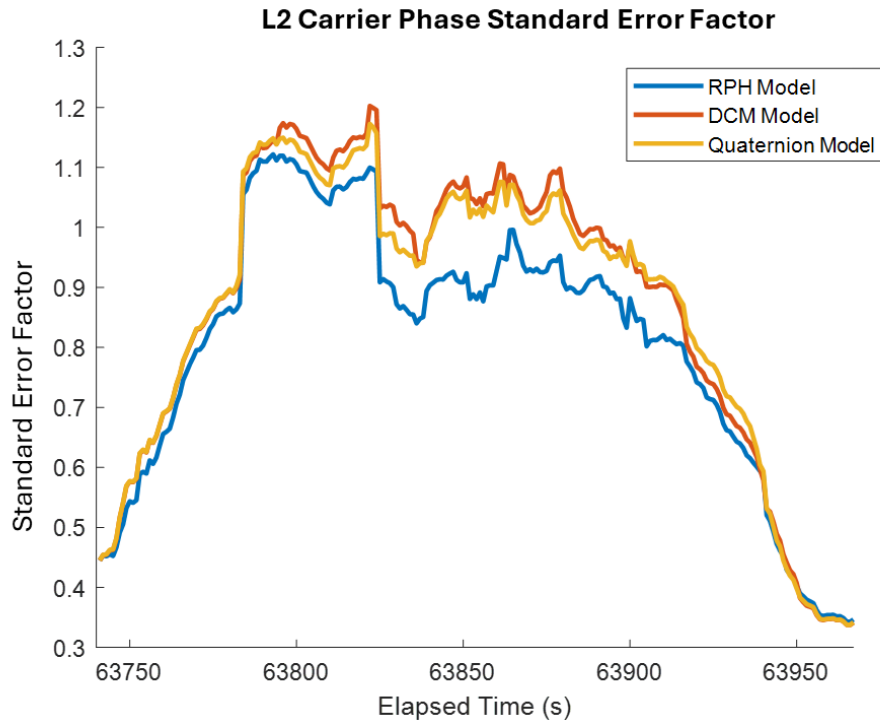


Figure 5.8.28: Estimated L2 carrier phase observation standard error factors for each attitude model (moving window: 20 s).

In Figure 5.8.23, the impact of the system jolt on the observation standard error factor estimates is very apparent. A sharp jump in all of the standard error plots starts at approximately $t =$

63880 s and ends at approximately $t = 63920$ s. This jump occurs over such a broad time period because of the fact that windowed standard error factor estimates are being used. This means that, with the 20 second moving window, the system jolt has an impact on the standard errors estimated for a period of time of 20 seconds before and 20 seconds after the jolt happens. For datasets with potentially complex issues, this can make it more difficult to determine exactly when a problem is occurring in the dataset. To illustrate the improvement that can be made to the time resolution of the standard error factor estimates, the standard error factors for the overall observation vector when using a moving window of 1 second are shown in Figure 5.8.29. In Figure 5.8.29, the sharp jump in the estimated standard error plots is much more pronounced than in Figure 5.8.23 and the time the jump occurs is much more readily identifiable. This increased ease of identifying when a problem is occurring comes at the cost of noisier standard error factor estimates over the course of the road test dataset. It is also worth noting that both plots only show the standard error factor estimates for the overall observation vector. These effects get more pronounced when using subsets of the observation vector, and much more pronounced when estimating standard error factors for the process noise vector since the process noise vector typically has a very low contribution to the overall system redundancy.

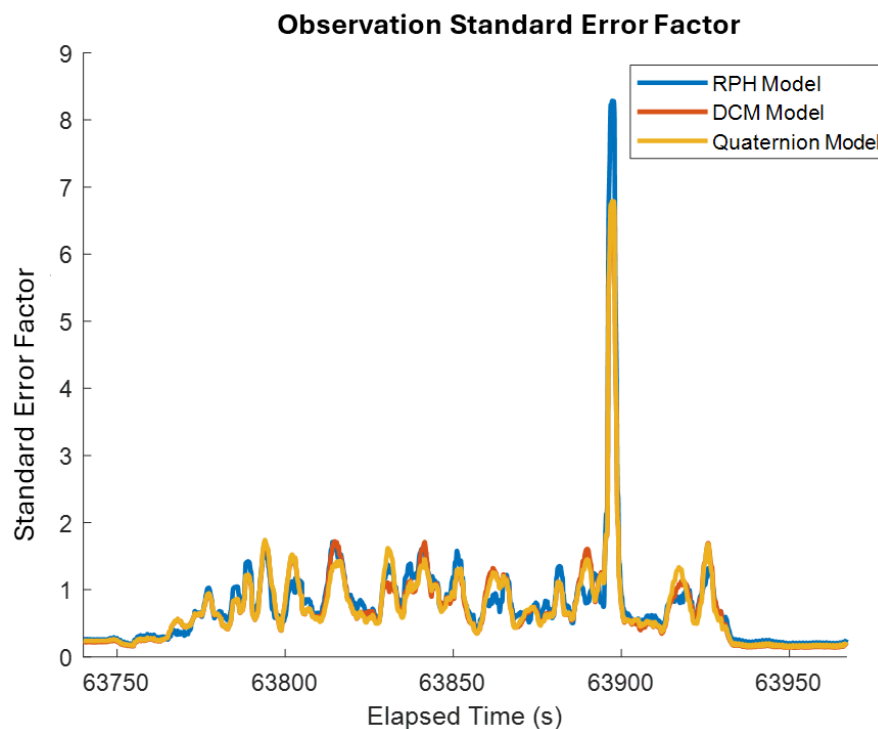


Figure 5.8.29: Estimated standard error factors for the overall observation vector (moving window: 1 s).

Notably, in Figure 5.8.23, all attitude models have similar standard error factor estimates, and are impacted similarly by the system jolt. This suggests that each attitude model has a similar response to erroneous observations. The fact that each standard error factor estimate returns to normal behaviour immediately following the system jolt's period of impact additionally indicates that each attitude model recovers quickly from the inclusion of the measurements at the time of the system jolt.

Looking at the gyroscope and accelerometer standard error factors in Figure 5.8.24 and Figure 5.8.25, respectively, there are strong artifacts in the observation standard errors around the time of the system jolt. The jump in the gyroscope standard errors is much more pronounced than in the accelerometer standard errors, suggesting that the jolt was more impactful on the system's angular measurements. Again, the impact across different attitude models is very similar.

Comparing the L1 C/A, L1 Carrier Phase, and L2 Carrier Phase standard error factors in Figure 5.8.26, Figure 5.8.27, and Figure 5.8.28, respectively, reveals that the GPS measurements were not significantly impacted by the system jolt. The L1 C/A measurements were not appreciably impacted by even the system motion, since the shifts in their standard error estimates are not correlated with static/kinematic portions of the dataset. The L1/L2 Carrier Phase measurements did perform better when the PosAtt system was at rest, but were still not impacted by the system jolt. It stands to reason that the GPS measurements would remain unimpacted by the system jolt, since:

- i. The system jolt would not have appreciably changed the location of the PosAtt system itself. Since the GPS observables are directly related to the system position, nothing changes in the observations' geometry when the system is jolted.
- ii. The system jolt did not impact any of the electrical components of the GPS receiver.

The process noise standard error factors are a core part of any comparison of the performance of each attitude model's associated system model. The system jerk standard error factor estimates for each attitude model are shown in Figure 5.8.30, and the standard error factor estimates for the process noise elements relating to the second-order time derivatives of the attitude parameters are shown in Figure 5.8.31.

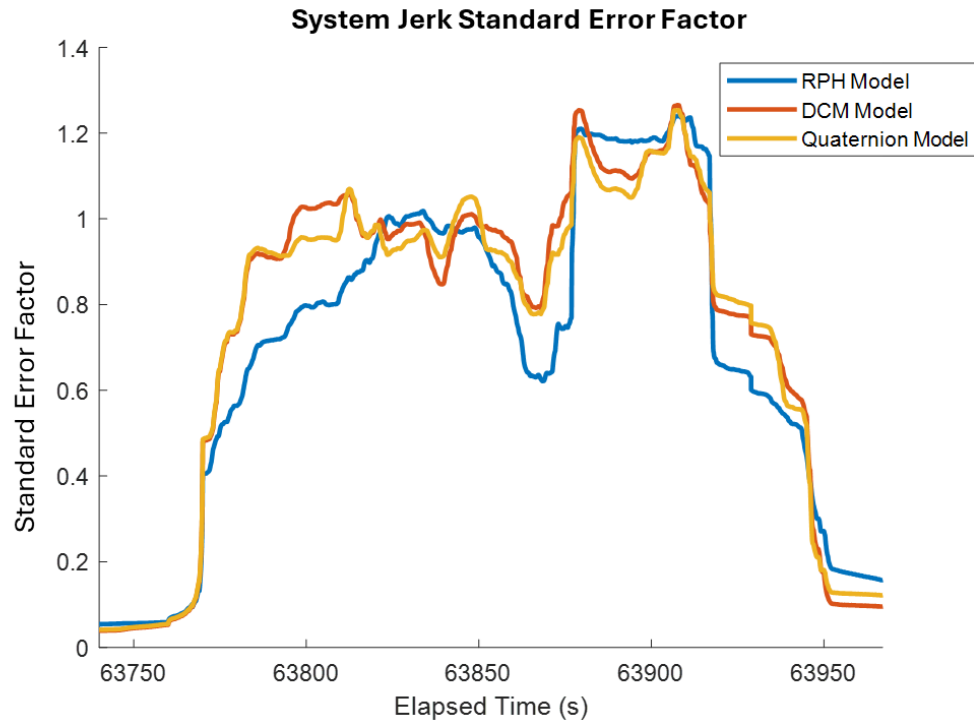


Figure 5.8.30: Estimated system jerk standard errors for each attitude model (moving window: 20 s).

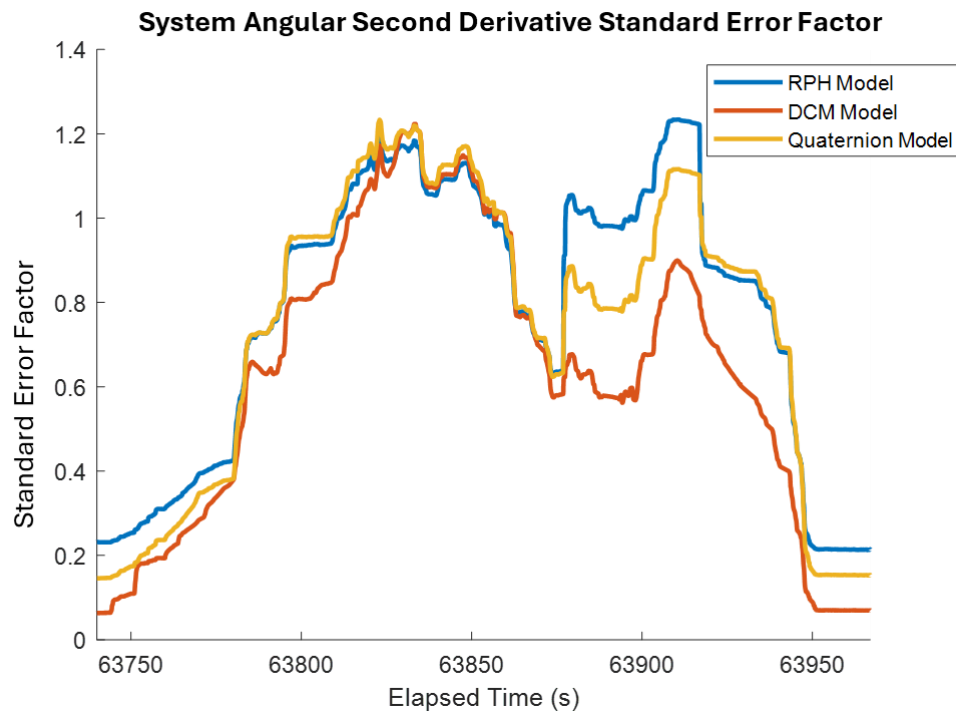


Figure 5.8.31: Estimated second time-derivative of attitude parameters' standard errors for each attitude model (moving window: 20 s).

The standard error factor estimates in Figure 5.8.30 and Figure 5.8.31 also have artifacts from the system jolt. The behaviour of each attitude model's process noise elements are very similar to one another. The attitude parameters' associated process noise standard error factors in Figure 5.8.31 exhibit slightly different responses to the system jolt depending upon the attitude model, with the DCM model being impacted the least and the RPH model being impacted the most. Given that all the attitude models represent the system's angular motion using the first-order time derivatives of the attitude parameters, each attitude model is using a different kinematic model for the system's angular motion. It is therefore reasonable that each attitude model's process noise elements would respond differently to the same strain placed on the system.

The results in Figure 5.8.31 indicate that the DCM model is the most resistant to observation errors creeping into the KF system state prediction. This may be due to the fact that the DCM model imposes a larger number of constraints than the other attitude models do. This larger number of constraints might provide stricter limits to sudden changes in the attitude parameter estimates in the DCM model. Of course, this comes with the additional computational load of the DCM model's 18 attitude parameters compared the Quaternion model's 8 attitude parameters and the RPH model's 6 attitude parameters. It is worth noting that this does not imply that the DCM model outperforms the other attitude parameters, but rather that each attitude model responds differently to erroneous observations and internalizes them differently.

5.9 Summary

This chapter formulates and presents three unique attitude models in the framework of the GMIS: the Roll-Pitch-Heading (RPH) model, the Quaternion Model, and the Direction Cosine Matrix (DCM) model. This work specifically extends the attitude modeling capabilities of the GMIS and further explores some of the unique tools that the GMIS provides for comprehensive error analysis.

Under traditional sensor fusion, different attitude models are realized via the IMU-derived free inertial navigation calculations, which makes the quality of the KF time update tied specifically to the quality of the IMU observations. This work instead predicts the attitude states using their time derivatives in their system model and then incorporates the IMU measurements through the measurement update in the positioning EKF, allowing for the time update's quality to be independent from the quality of the IMU measurements. This method of modeling is particularly valuable for lower-

cost IMUs, to reign in some of the negative impacts these sensors have on traditional multi-sensor integrated systems.

The models derived and used in this chapter are also unique in the formulation of the system state constraint equations for not only the attitude parameters in the DCM and Quaternion attitude models, but also their first-order time derivatives. In fact, the method used to derive the system state constraints on the attitude parameters' first-order time derivatives could also be iteratively applied to derive system state constraints on the attitude parameters' higher-order time derivatives as well. This opens the door to further exploring the impact of adopting more complex attitude kinematic models that incorporate these higher-order terms in the system states.

The models developed in this research were fully realized in the PosAtt Engine and successfully used to process real road data, displaying strong capability in identifying and isolating specific elements that may degrade solution quality. Combining the GMIS, the alternative derivation of the state-constrained KF in Section 3, and comprehensive error analysis allows for a thorough analysis of system performance. Moreover, this analysis is highly customizable in terms of the desired information, which makes it well-suited both in a post-processing environment, where the additional computational load of calculating residual vectors and standard error factors can be more readily accommodated, and in a real-time processing environment, where the scope of comprehensive error analysis can be reduced to accommodate real-time processing needs.

Table 5.9.1 provides an overview of the required parameters and constraints to use each of the three attitude models explored in this research in the GMIS. The computational complexity of the positioning KF increases with the size of the state vector (i.e. the number of states in an attitude model) and with the number of constraints being imposed on the state vector. The DCM and Quaternion models come with an additional associated computational cost compared to the RPH model. This added computational burden is less worrisome considering the processing capabilities of modern computers compared to those of computers from a few decades ago. However, it is still a relevant issue when considering that using the DCM or Quaternion models gives less processing overhead for other tasks.

Table 5.9.1: Overview of the different attitude models in the GMIS.

Attitude Model	No. of States	Number of Constraints	Process Noise
RPH	6	0	2 nd Order

DCM	18	12	2 nd Order
Quaternion	8	2	2 nd Order

Each of the attitude models used in this research were directly linked to the IMU raw measurement models, allowing for KF measurement updates even when only IMU data is available. This additionally allows for the direct estimation of residuals for the raw specific force and angular rate measurements, enabling a much deeper analysis of the system's performance than would otherwise be possible, particularly in a post-processing environment.

VCE shows particular promise as a tool to conduct Adaptive Kalman Filtering to dynamically tune the observation and process noise covariance matrices [Wang, et al, 2010]. The Kalman Filters used for position and attitude estimation are very sensitive to tuning errors in these covariance matrices. Having a window into the time-varying performance of the positioning sensor observations and system's process noise is very valuable in this context, and provides many avenues for further analysis and development.

Altogether, the contributions of this chapter enable a direct comparison of the performance of different attitude models to be made, without requiring either simulated data or a ground truth solution. Over the dataset explored in this chapter, comparing the performance of each attitude model used in this research revealed a general compatibility between them for ground vehicle data, and also similar responses to erroneous measurements.

6 Modeling of IMU Arrays Under the Generic Multi-Sensor Integration Strategy

This chapter is based upon the published work:

Brunson, B., Wang, J., and Ma, W. (2024): Innovative Modeling of IMU Arrays Under the Generic Multi-Sensor Integration Strategy, *Sensors*, 24(23), 7754.

6.1 Preamble

The accessibility of Direct Geo-referencing Technology (DGT) has improved dramatically with the development of MEMS IMUs and other low-cost positioning and orientation sensors. MEMS technology in particular has had many advancements that make satisfying high position/attitude accuracy requirements while reducing system cost more attainable than ever before.

MEMS IMUs are generally less expensive to manufacture and acquire than non-MEMS IMUs, but this reduction in cost typically comes with lower observation accuracy and less stable performance through time. Mitigating the performance loss of the less-expensive MEMS IMUs has been a significant driver of ongoing research in both academia and industry.

One particular challenge when it comes to sensor integration is handling some of the limitations that exist when using lower-cost MEMS IMU sensors. Since MEMS IMU sensors have typically less stable behaviour over time than non-MEMS IMU sensors, their use needs to come with proper modeling of their sensor errors. Even when properly accounting for MEMS IMU sensor errors, though, they typically produce lower-quality data than their more expensive counterparts. To mitigate this issue, many researchers have opted to arrange multiple lower-cost IMUs in an array so that their data may be fused to simulate a higher-quality IMU than any of the constituent sensors. There are two primary approaches to the physical construction of a Multi-IMU (MIMU) array:

- i. Building an IMU array chip, such as those used in [Greenheck, 2015; Shahri and Rasoulzadeh, 2017]. Mounting multiple IMUs on an array chip allows for the relative geometry between different IMUs in the array to be fixed precisely, which in turn allows for their sensitive axes to be aligned very accurately. Changing the relative orientation of the sensitive axes of constituent IMU sensors has a significant impact on the MIMU array output [Martin, 2016]. Different relative orientations of the constituent IMUs' sensitive axes were explored in [Waegli et al,

2008], and it was determined that skew-aligned axes yield more accurate results at the cost of additional computational complexity.

- ii. Building an IMU array from multiple strapdown IMUs that are not mounted on a common chip, such as the system described in [Carlsson et al, 2022]. A MIMU array constructed in this manner can accommodate arbitrary sensor placement, allowing for the construction of large sensor arrays that have large lever arm vectors that may be leveraged (sic) to provide more precise attitude information, for example by placing IMUs on either side of an airplane's fuselage [Martin, 2016]. Since the constituent IMU sensors are not necessarily mounted as precisely in this approach, it is important to calibrate both the lever arm vector and boresight angles for each sensor in the array.

Practically, there are applications of MIMU arrays that extend beyond just simulating a higher-quality sensor. For example, in highly dynamic systems, constructing a MIMU array comprised of sensors with different Full-Scale Range values allows for the effective Full-Scale Range of the MIMU array to be larger than any constituent sensor [Jao et al, 2023].

Beyond the MIMU array's physical construction, there is the added complexity of data fusion with a MIMU array, particularly when the traditional integration strategy only allows for one core IMU sensor. There are 3 primary methods of MIMU array data fusion:

- i. "Virtual" IMU fusion, as seen in [Libero and Klein, 2024; Zhang et al, 2020; Gagnon et al, 2018; Bancroft and Lachapelle, 2011]: Under this fusion approach, the MIMU output is converted to that of a single, virtual IMU (VIMU). The VIMU generally has a pre-defined location in the array, and each sensor's output is transformed to refer to the VIMU's position. The single VIMU's simulated data may then be integrated as an equivalent single IMU, allowing for the direct application of traditional sensor integration as described in Section 2.6.2. This approach is very popular for MIMU array data fusion, but has a limited capacity for fault detection and exclusion [Lachapelle, 2011].
- ii. Federated Filter Fusion, as seen in [Gagnon et al, 2018; Bancroft and Lachapelle, 2011]: Under this fusion approach, the MIMU output is determined by breaking the overall problem down into multiple single IMU (SIMU) positioning problems, which are then combined via LS estimation. This approach also does not accommodate fault detection and exclusion [Lachapelle, 2011].

- iii. Centralized Filter Fusion, as seen in [Libero and Klein, 2022; Gagnon et al, 2018; Bancroft and Lachapelle, 2011]: This approach augments the error-state estimation process to account for the array's output, allowing for the addition of geometric constraints on the relative positions between constituent IMU sensors in the MIMU array. This approach also cannot accommodate fault detection and exclusion [Lachapelle, 2011].

It is worth noting that the Federated Filter Fusion and Centralized Filter Fusion approaches function very similarly to the centralized and federated integration architectures described in Section 2.6.1.2.

One particular weakness of the VIMU approach is that it cannot accommodate individual estimates of IMU biases and scale factor errors for individual constituent sensors, unless this information is known *a priori* and remains stable through time [Lachapelle, 2011]. MEMS IMUs, even from the same model, often have vastly different systematic errors. In fact, even for a single MEMS IMU, turn-on biases may vary significantly for different operations of the PosAtt system. The Federated Filter Fusion and Centralized Filter Fusion approaches can model individual sensor biases and scale factor errors, but these sensor errors are often assumed to be the same across the individual IMUs in the MIMU array [Gagnon et al, 2018; Lachapelle, 2011].

It is generally well-known that when multiple IMUs are integrated together, even using the same IMU model, they often exhibit vastly different systematic error behaviour through time. In fact, systematic errors can vary significantly from turn-on to turn-on for the same IMU, which makes thoroughly modeling and accounting for individual sensor errors in an array of lower-cost sensors important. Existing approaches to MIMU array data fusion have limitations when it comes to characterizing individual sensor errors and conducting fault detection and exclusion for specific sensors in the array. The VIMU approach cannot accommodate individual estimates of IMU sensor biases/scale factor errors, unless this information is imposed as *a priori* knowledge and behaves stably through time [Lachapelle, 2011]. The Federated Filter Fusion and Centralized Filter Fusion approaches can estimate sensor biases/scale factor errors for individual IMU sensors, but in practice sensor errors are typically assumed to be the same across all individual IMUs comprising the MIMU array [Gagnon et al, 2018; Lachapelle, 2011].

All three of the dominant approaches for MIMU array data fusion make use of the traditional sensor integration strategy described in Section 2.6.2 at their core. The associated workflow presents a

consistent limitation on the potential methods to fuse MIMU array data, since it only readily accommodates either a single core IMU sensor or a single INS navigation solution.

With the traditional integration strategy being the dominant framework for MIMU array data fusion, one significant area of further research lies in modifying the overall integration strategy used to process data from a MIMU array, in order to improve the versatility/performance of the associated position/attitude solution. The traditional sensor integration strategy was developed with computational efficiency as a core objective due to limited mobile computing power, but modern developments in mobile processing power, sensors, and data science make computational efficiency less of a severe limitation, even in real-time processing environments.

The Generic Multi-sensor Integration Strategy (GMIS) presents a valuable alternative to the traditional sensor integration strategy in its modeling capabilities and error analysis prospects for MIMU arrays. Since the GMIS does not rely on a single core IMU sensor, but rather models all position/orientation observations in the positioning EKF, modeling a MIMU array in the GMIS removes the need to “fuse” the MIMU array output into that of a single simulated IMU, allowing instead for each sensor in the MIMU array to be modeled individually. This opens up several possibilities for further analysis:

- Modeling arbitrary MIMU array configurations: When the observations from each sensor in the MIMU array is modeled individually, the process to account for different sensor geometries becomes more straightforward. This allows for more ready accommodation of non-standard sensor configurations and idiosyncratic/atypical sensors, such as the tetrahedral configuration of four gyroscopes as in [Kuga et al, 2012; Hanson and Pachter, 2005]. Additionally, modeling each sensor individually and not having any sensor being a “core” sensor means that IMU sensors that collect data at different rates may be readily integrated under this framework.
- Modeling individual sensor systematic errors for constituent sensors in the MIMU array: Separately accounting for the systematic errors present in each constituent IMU sensor and directly including them in the system states removes the need to assume that sensor errors are homogeneous and allows for them to be estimated individually over the entire dataset. Additionally, including the individual IMU systematic errors in the state vector allows for their estimates to vary with time, which accounts for MEMS IMUs’ at times unstable performance.
- Providing separate estimates of measurement residuals for the raw IMU specific force and angular rate measurements: Since the GMIS does not rely on error state estimation and instead

models IMU measurements directly in the positioning EKF, the measurement residuals for the raw IMU measurements are able to be estimated, which allows for fault detection and exclusion to be performed for every sensor in the MIMU array.

- Using Variance Component Estimation (VCE) to characterize individual IMU sensor errors for constituent sensors in the MIMU array: Directly estimating the individual sensors' observation residuals additionally allows for an estimation of their standard error factors, which describe how well *a priori* assumptions of sensor accuracy align with their actual *a posteriori* performance. These estimates can also be time-varying, which is particularly valuable for adaptive systems and for MEMS IMUs, whose performance can vary significantly over time.

While a thorough exploration of some of these factors lies beyond the scope of this dissertation, this chapter puts forward a strong foundation for a thorough quantitative assessment of IMU sensor performance in a MIMU array. This is particularly valuable for the design of a PosAtt system, but also for detailed post-processing of MIMU array data, completely removing many of the limitations of existing MIMU array data fusion techniques.

6.2 Position and Attitude System Models for IMU Arrays

The position kinematic model used in this chapter uses a constant linear acceleration to describe the change in the system's linear position through time. The position system model and its corresponding partitioned elements in the state vector and process noise vector are defined identically to Section 5.3 in this chapter.

In this chapter, the Roll-Pitch-Heading (RPH) attitude model is used to describe the system's orientation and its changes through time. This model is chosen here for its ease of interpretation compared to the Quaternion and Direction Cosine Matrix (DCM) attitude models, but it is important to note that the contributions of Chapter 5 allow for any of the three attitude models to be used to process MIMU array data.

Using the RPH attitude model, the system's angular rate of change is expressed through the first-order time derivatives of the roll, pitch, and heading parameters. The corresponding definitions of the attitude system model and partitioned elements in the system state/process noise vectors are identical to the definitions provided in Section 5.5.1.

6.3 Partitioning of the EKF Matrices

The defined system models in Section 6.2 fit neatly into the system of partitioning the EKF matrices presented in Section 5.2. When modeling individual IMUs in a MIMU array, it is valuable to further partition the system state elements corresponding to individual IMU systematic error parameters. This further partitioning enables a more intuitive process of modeling these errors and accounting for them. The partitioning of the EKF matrices here are designed to be compatible with the partitioning developed in Section 5.2 so that the research contributions of both chapters may be readily used together.

Considering that a MIMU array may contain any arbitrary number of IMU sensors, there are separate parameters describing the accelerometer/gyroscope biases and scale factor errors for each sensor in the array. Taking this into account, the state vector may be partitioned into the following groups:

- Linear position, velocity, and acceleration;
- Attitude and its rate of change;
- GNSS ambiguity parameters that have yet to be fixed; and
- IMU systematic errors for each individual accelerometer and gyroscope in the MIMU array (one bias and scale factor error for each accelerometer/gyroscope).

The corresponding partitioned state vector is defined as

$$\mathbf{x} = [\mathbf{x}_p^T \quad \mathbf{x}_a^T \quad \mathbf{x}_\lambda^T \quad \mathbf{x}_{imu1}^T \quad \dots \quad \mathbf{x}_{imun}^T]^T \quad (6.1)$$

where the subscripts p , a , λ , and $imui$ denote the linear kinematic state (i.e. position, velocity, and acceleration) vector, the angular state (i.e. attitude and its first-order time derivatives) vector, the GNSS ambiguity vector, and the IMU systematic error vector for the i th IMU in the MIMU array, respectively. Taking this partitioning of the system state vector, the corresponding partitioned definitions of the transition matrix, process noise vector, and process noise transition matrix are given by the following three equations:

$$A = \begin{bmatrix} A_p & 0 & 0 & 0 & 0 & 0 \\ 0 & A_a & 0 & 0 & 0 & 0 \\ 0 & 0 & A_\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{imu1} & \dots & 0 \\ 0 & 0 & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & A_{imun} \end{bmatrix} \quad (6.2)$$

$$w = [w_p^T \quad w_a^T \quad w_\lambda^T \quad w_{imu1}^T \quad \dots \quad w_{imun}^T]^T \quad (6.3)$$

$$B = \begin{bmatrix} B_p & 0 & 0 & 0 & 0 & 0 \\ 0 & B_a & 0 & 0 & 0 & 0 \\ 0 & 0 & B_\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{imu1} & \dots & 0 \\ 0 & 0 & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & B_{imun} \end{bmatrix} \quad (6.4)$$

Focusing on a GNSS/MIMU array integrated system as considered in this work, there are three types of raw measurements that may be available at any given observation epoch:

- i. Angular rate observations from the gyroscopes in each strapdown IMU sensor in the MIMU array;
- ii. Specific force observations from the accelerometers in each strapdown IMU sensor in the MIMU array; and
- iii. Double-differenced L1 C/A and L1/L2 carrier phase observations between a static reference GNSS receiver and a rover GNSS receiver affixed to the PosAtt system.

The main difference between the groupings of observations presented here and those presented in Section 5.2 is that there is a full set of specific forces/angular rate observations supplied by every sensor in the MIMU array. As such, the partitioning of the system innovation vector follows a very similar process as was followed in Section 5.2.

The three sets of raw measurements can naturally be expected to have different stochastic properties and behaviours throughout any kinematic dataset, and so it is intuitive to separate them when partitioning the system innovation vector. Additionally, the individual specific forces/angular rates from different sensors can also be separated when partitioning the system innovation vector to make the analysis of the individual sensors' stochastic properties more intuitive. Taking all of this into account, the system innovation vector may therefore be partitioned as

$$d = [d_{g1}^T \quad d_{s1}^T \quad \dots \quad d_{gn}^T \quad d_{sn}^T \quad d_G^T]^T \quad (6.5)$$

where the subscripts gi , si , and G refer to the gyroscope observations from the i th IMU in the MIMU array, the specific force observations from the i th IMU in the MIMU array, and the GNSS observations, respectively, all observed at t_{k+1} .

Taking the partitioned definitions of both the system state vector and the system innovation vector, the associated design matrix may be partitioned as

$$C = \begin{bmatrix} C_{g1,p} & C_{g1,a} & C_{g1,\lambda} & C_{g1,imu1} & \cdots & C_{g1,imun} \\ C_{s1,p} & C_{s1,a} & C_{s1,\lambda} & C_{s1,imu1} & \cdots & C_{s1,imun} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{gn,p} & C_{gn,a} & C_{gn,\lambda} & C_{gn,imu1} & \cdots & C_{gn,imun} \\ C_{sn,p} & C_{sn,a} & C_{sn,\lambda} & C_{sn,imu1} & \cdots & C_{sn,imun} \\ C_{G,p} & C_{G,a} & C_{G,\lambda} & C_{G,imu1} & \cdots & C_{G,imun} \end{bmatrix} \quad (6.6)$$

The partitioning of the EKF matrices greatly simplifies the process of modeling individual IMU sensors in the MIMU array, and allows for clear documentation of the influence that each sensor has on the partitioned elements of the state vector.

Many elements of the partitioned design matrix in (6.6) may be intuitively determined to be null matrices. This occurs when there is no defined relationship between a partitioned system innovation element and a partitioned system state element. In particular, the following elements of the partitioned design matrix will be null matrices:

- $C_{G,imu i} = \mathbf{0} \forall i = 1, \dots, n$: The GNSS observations are modeled independently of the IMU systematic error parameters.
- $C_{G,a} = \mathbf{0}$: When the GNSS rover receiver is considered to be located at the origin of the vehicle frame, the GNSS observations are reflective only of the system's position and are not related to the system attitude parameters.
- $C_{gi,\lambda} = \mathbf{0} \forall i = 1, \dots, n$, $C_{si,\lambda} = \mathbf{0} \forall i = 1, \dots, n$: The gyroscope and specific force observations for a given IMU sensor are not dependent upon the GNSS ambiguity parameter estimates.
- $C_{gi,imu j} = \mathbf{0} \forall i \neq j, i \in 1, \dots, n, j \in 1, \dots, n$, $C_{si,imu j} = \mathbf{0} \forall i \neq j, i \in 1, \dots, n, j \in 1, \dots, n$: The systematic error parameters for a given IMU sensor are modeled independently of observations from any other IMU sensor in the MIMU array.

Taking these into account, the partitioned design matrix in (6.6) may be expressed as

$$C = \begin{bmatrix} C_{g1,p} & C_{g1,a} & 0 & C_{g1,imu1} & 0 & \cdots & 0 \\ C_{s1,p} & C_{s1,a} & 0 & C_{s1,imu1} & 0 & \cdots & 0 \\ C_{g2,p} & C_{g2,a} & 0 & 0 & C_{g2,imu2} & \cdots & 0 \\ C_{s2,p} & C_{s2,a} & 0 & 0 & C_{s2,imu2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{gn,p} & C_{gn,a} & 0 & 0 & 0 & \cdots & C_{gn,imun} \\ C_{sn,p} & C_{sn,a} & 0 & 0 & 0 & \cdots & C_{sn,imun} \\ C_{G,p} & 0 & C_{G,\lambda} & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (6.7)$$

Here, the definition of the design matrix elements relating to the GNSS observations remain identical to those defined for the alternative attitude models in Section 5.5. The modeling of the individual IMU sensors in the MIMU array is described in Section 6.4.

6.4 MIMU Array Observation Equations in the GMIS

The core of the observation models for the MIMU array are identical to the single IMU observation model presented in Section 2.4.2.4. The core of these equations remains the same when modeling IMU sensors in an array, and the specific changes necessary to model these observations for a MIMU array are presented in the subsequent Sections.

6.4.1 Modeling Accelerometer Observations in the MIMU Array

One important practical consideration that must be considered when modeling multiple IMU sensors that does not need consideration for a single IMU sensor is that each sensor in a MIMU array must be properly aligned to integrate their data reliably. This requires individual sensor calibration to account for the relative orientation between IMU sensors in the MIMU array and bring them into a common frame of reference.

Assuming a standard three axis accelerometer triad for each constituent IMU sensor, the accelerometer observations for the i th IMU sensor in the MIMU array may be modeled as

$$\mathbf{R}_i(\mathbf{f}_{ib}^b)_i = \mathbf{C}_n^b \mathbf{a} + \mathbf{C}_n^b \mathbf{g} - \mathbf{b}_{ai} - \mathbf{S}_{ai} \mathbf{f}_{ib}^b + \mathbf{C}_n^b \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} + \mathbf{C}_n^b \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i + \boldsymbol{\Delta}_{f_{ib}^b, i} \quad (6.8)$$

where $(\mathbf{f}_{ib}^b)_i$ denotes the specific force the body frame of the i th IMU experiences with respect to the inertial frame, expressed in the body frame; $\mathbf{a}$ denotes the acceleration the body frame experiences in the navigation frame; $\mathbf{g}$ denotes the gravity vector in the navigation frame; $\mathbf{v}$ denotes the velocity of the body frame in the navigation frame; $\mathbf{r}_i$ denotes the lever arm vector for the i th IMU; $\mathbf{R}_i$ is the rotation matrix that aligns the i th IMU sensor in the MIMU array to the right-forward-up orientation with respect

to the vehicle frame; and $\Delta_{f_{ib,i}}^b$ denotes the observation noise vector for the accelerometer triad of the i th IMU in the MIMU array.

The observation model in (6.8) differs from that in (2.46) in several key ways:

- The accelerometer biases and scale factor errors are specific to each constituent IMU sensor in the MIMU array. Since each sensor is expected to have different biases and scale factor errors, particularly for MEMS IMUs, this additional parameterization is necessary to account for the sensors' systematic errors without loss of accuracy.
- The lever arm vector $\mathbf{r}_i$ is considered specific to each constituent IMU sensor in the MIMU array. While the lever arm vector is not included in the state vector in this research, the elements of the lever arm vector could be included as state elements for a rigorous calibration procedure, or for applications where the lever arm vectors might change over time (e.g. large-scale MIMU arrays distributed over an aircraft).
- The inclusion of $\mathbf{R}_i$ to bring all IMU sensors into a common reference frame. This again stresses the importance of having accurate knowledge of the relative geometry between constituent IMU sensors in a MIMU array. As with the separately-defined lever arm vectors, $\mathbf{R}_i$ is considered to already be determined prior to data collection, but could be included as elements in the system state vector for a rigorous calibration procedure, or in situations where the relative orientation of constituent IMU sensors may change significantly over time.
- The separate definition of the observation noise vector for every constituent IMU sensor in the MIMU array. This is core to one of the main research contributions of this chapter, namely in the investigation of and accounting for the different stochastic properties of individual IMU sensors in the MIMU array. This would also be essential to consider when developing an adaptive system that uses multiple IMUs with different Full-Scale Ranges (FSRs) to extend the effective FSR of the MIMU array.

Taking the partial derivatives of (6.8) with respect to the relevant elements of the state vector results in the following definition of the non-null design matrix submatrices for the accelerometer observations from the i th IMU:

$$\mathbf{C}_{si,p} = [\mathbf{0}_3 \quad \mathbf{0}_3 \quad \mathbf{C}_n^b] \quad (6.9)$$

$$\mathbf{C}_{si,a} = \left[\frac{\partial \mathbf{C}_n^b}{\partial \mathbf{x}_a} (\mathbf{a} + \mathbf{g} + \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i) \right] \quad (6.10)$$

$$\mathbf{C}_{si,imui} = \begin{bmatrix} & f_{ib,measx}^b & 0 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & 0 & f_{ib,measy}^b & 0 \\ & 0 & 0 & f_{ib,measz}^b \end{bmatrix} \quad (6.11)$$

Equations (6.9) to (6.11) may be numerically evaluated and substituted into the design matrix in (6.7) to fully realize the IMU accelerometer observation models in the GMIS.

6.4.2 Modeling Gyroscope Observations in the MIMU Array

As with the accelerometer observations, it is very important to account for the relative geometry between different sensors' gyroscopes in the MIMU array, since it is essential to be combining the different sensors' observations within the same frame of reference. This necessitates several key changes to the standard IMU gyroscope observation model in (2.47).

Assuming a standard three axis gyroscope triad for each constituent IMU sensor in the MIMU array, the gyroscope observations for the i th IMU in the array may be modeled as

$$\mathbf{R}_i(\omega_{ib}^b)_i = \omega_{nb}^b - \mathbf{b}_{gi} - \mathbf{S}_{gi}\omega_{ib,meas}^b + \mathbf{C}_n^b(\omega_{ie}^n + \omega_{en}^n) + \Delta_{\omega_{ib},i}^b \quad (6.12)$$

where $(\omega_{ib}^b)_i$ denotes the gyroscope measurement vector describing the body frame's angular rate with respect to the inertial frame, observed in the body frame; ω_{ie}^n is the angular velocity vector of Earth with respect to the inertial frame, expressed in the navigation frame; ω_{en}^n is the angular velocity vector of the navigation frame with respect to Earth, expressed in the navigation frame; $\mathbf{R}_i$ is the rotation matrix that aligns the i th IMU sensor in the MIMU array to the right-forward-up orientation with respect to the vehicle frame; and $\Delta_{\omega_{ib},i}^b$ is the observation noise vector for the i th IMU's gyroscope observations.

As with the accelerometer observations, there are several key differences between the observation model presented in (6.12) and that in (2.47):

- The gyroscope biases and scale factor errors are specific to each constituent sensor in the MIMU array. As with the accelerometer model in (6.8), the additional parameterization is necessary to account for the fact that each sensor is expected to have different systematic errors, and that MEMS sensors in particular are prone to unstable behaviour in these errors through time and between turn-ons.
- The inclusion of $\mathbf{R}_i$ to bring all sensors into a common reference frame. It is worth noting that, assuming each IMU to have a consistently defined body frame, $\mathbf{R}_i$ will remain the same across both accelerometer and gyroscope observations. While $\mathbf{R}_i$ is considered *a priori* knowledge for

the purposes of this research, this fact may be leveraged as part of a rigorous calibration procedure.

- The separate definition of the observation noise vector for each gyroscope in the MIMU array. As with the accelerometers, this is core to the goal of separately characterizing and accounting for the different stochastic properties of different constituent sensors in the MIMU array.

Following a similar procedure as was followed in Section 5.5.1, the following definitions of the non-null design matrix submatrices may be determined for the gyroscope observations of the i th IMU:

$$\mathbf{C}_{gi,a} = \begin{bmatrix} 0 & \cos \alpha & \sin \alpha \cos \beta \\ \mathbf{0}_3 & 1 & -\sin \beta \\ 0 & \sin \alpha & -\cos \alpha \cos \beta \end{bmatrix} \quad (6.13)$$

$$\mathbf{C}_{gi,imui} = \begin{bmatrix} \omega_{ib,measx}^b & 0 & 0 \\ \mathbf{I}_3 & 0 & \omega_{ib,measy}^b & 0 & \mathbf{0}_3 & \mathbf{0}_3 \\ 0 & 0 & \omega_{ib,measz}^b \end{bmatrix} \quad (6.14)$$

Equations (6.13) and (6.14) may be numerically evaluated and substituted into the design matrix in (6.7) to fully realize the IMU gyroscope observation models in the GMIS.

6.4.3 Modeling Atypical IMU Sensors

One particular strength of the GMIS' ability to account for individual sensors in a MIMU array lies in its ability to intuitively account for atypical IMU sensors, such as four-axis accelerometers/gyroscopes. While this is not a focus in this research, it is worth noting that Section 6.4.1 and Section 6.4.2 both assume three-axis accelerometer and gyroscope observations, with the sensing axes all being designed to be mutually orthogonal. This section puts forth a basis for accounting for IMU observations that do not conform to this paradigm. The specific process and rationale for doing this is laid out in detail for the IMU accelerometer observations, with the gyroscope observations following a very similar rationale.

Consider that the i th IMU sensor in a MIMU array consists of:

- m_s accelerometers arranged with their sensing axes oriented in the directions of the unit vectors $\mathbf{u}_j, j \in 1, \dots, m_s$ in the IMU body frame; and
- m_g gyroscopes arranged with their sensing axes oriented in the directions of the unit vectors $\mathbf{v}_j, j \in 1, \dots, m_g$ in the IMU body frame.

Each accelerometer and gyroscope in the IMU sensor will have an associated bias and scale factor error, leading to a total of m_s accelerometer biases and scale factor errors, and m_g gyroscope biases and scale factor errors. The corresponding elements of the partitioned system state vector may therefore be defined as

$$\mathbf{x}_{imui} = \left[(b_{gi})_1 \quad \cdots \quad (b_{gi})_{m_g} \quad (s_{gi})_1 \quad \cdots \quad (s_{gi})_{m_g} \quad (b_{si})_1 \quad \cdots \quad (b_{si})_{m_s} \quad (s_{si})_1 \quad \cdots \quad (s_{si})_{m_s} \right]^T \quad (6.15)$$

Since there are m_s accelerometers in the i th IMU, the specific force observation vector may be expressed as

$$(\mathbf{f}_{ib}^b)_i = [f_1 \quad \cdots \quad f_{m_s}]^T \quad (6.16)$$

For the specific force observations, each element of the specific force vector f_j corresponds to the acceleration that the system is experiencing along the IMU's sensing axis $\mathbf{u}_j$. Intuitively, the acceleration the system experiences along $\mathbf{u}_j$ may be expressed as the projection of the system's acceleration onto the direction of the sensing axis. This projection may be practically determined by taking the dot product of the sensing axis direction in the IMU body frame with the system's acceleration, expressed in the IMU body frame. This allows each element of the specific force vector to be expressed as

$$f_j = \mathbf{C}_n^b \mathbf{a} \cdot \mathbf{u}_j + \mathbf{C}_n^b \mathbf{g} \cdot \mathbf{u}_j + \mathbf{C}_n^b \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} \cdot \mathbf{u}_j + \mathbf{C}_n^b \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i \cdot \mathbf{u}_j - (b_{si})_j - (s_{si})_j f_j + \Delta_{f_j} \quad (6.17)$$

This allows for the overall specific force vector of the i th IMU to be modeled as

$$\begin{aligned} (\mathbf{f}_{ib}^b)_i &= [f_1 \quad \cdots \quad f_{m_s}]^T \\ &= \begin{bmatrix} \mathbf{C}_n^b \mathbf{a} \cdot \mathbf{u}_1 + \mathbf{C}_n^b \mathbf{g} \cdot \mathbf{u}_1 + \mathbf{C}_n^b \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} \cdot \mathbf{u}_1 + \mathbf{C}_n^b \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i \cdot \mathbf{u}_1 \\ \vdots \\ \mathbf{C}_n^b \mathbf{a} \cdot \mathbf{u}_{m_s} + \mathbf{C}_n^b \mathbf{g} \cdot \mathbf{u}_{m_s} + \mathbf{C}_n^b \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} \cdot \mathbf{u}_{m_s} + \mathbf{C}_n^b \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i \cdot \mathbf{u}_{m_s} \end{bmatrix} \\ &\quad - \mathbf{b}_{si} - \mathbf{S}_{s_i} (\mathbf{f}_{ib}^b)_i + \Delta_{(\mathbf{f}_{ib}^b)_i} \end{aligned} \quad (6.18)$$

The partial derivatives of (6.18) may be used to populate the corresponding non-null submatrices of the observation design matrix as follows:

$$\mathbf{C}_{si,p} = \begin{bmatrix} \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{u}_1^T \mathbf{C}_n^b \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{u}_{m_s}^T \mathbf{C}_n^b \end{bmatrix} \quad (6.19)$$

$$\mathbf{C}_{si,a} = \begin{bmatrix} \frac{\partial \mathbf{C}_n^b}{\partial \mathbf{x}_a} (\mathbf{a} + \mathbf{g} + \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i) \cdot \mathbf{u}_1 \\ \vdots \\ \frac{\partial \mathbf{C}_n^b}{\partial \mathbf{x}_a} (\mathbf{a} + \mathbf{g} + \{2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\} \times \mathbf{v} + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{\omega}_{ib}^b \times \mathbf{r}_i) \cdot \mathbf{u}_{m_s} \end{bmatrix} \quad (6.20)$$

$$\mathbf{C}_{si,imui} = \begin{bmatrix} & & & f_1 & \cdots & 0 \\ \mathbf{0}_{m_s, m_g} & \mathbf{0}_{m_s, m_g} & \mathbf{I}_{m_s} & \vdots & \ddots & \vdots \\ & & & 0 & \cdots & f_{m_s} \end{bmatrix} \quad (6.21)$$

Since there are m_g gyroscopes in the i th IMU, the angular rate observation may be expressed as

$$(\boldsymbol{\omega}_{ib}^b)_i = [\omega_1 \quad \cdots \quad \omega_{m_g}]^T \quad (6.22)$$

As was the case with the specific force observations, the element of the angular rate vector ω_j corresponds to the angular rate that the system is experiencing along the IMU's sensing axis $\mathbf{v}_j$. This may again be practically determined via the dot product of the system's angular velocity vector, expressed in the body frame, with the sensing axis direction in the body frame. This allows for each element of the angular rate vector to be expressed as

$$\omega_j = \boldsymbol{\omega}_{nb}^b \cdot \mathbf{v}_j + \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \cdot \mathbf{v}_j - (b_{gi})_j - (s_{gi})_j \omega_j + \Delta_{\omega_j} \quad (6.23)$$

and the overall angular rate vector may be expressed as

$$(\boldsymbol{\omega}_{ib}^b)_i = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_{m_g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_{nb}^b \cdot \mathbf{v}_1 + \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \cdot \mathbf{v}_1 \\ \vdots \\ \boldsymbol{\omega}_{nb}^b \cdot \mathbf{v}_{m_g} + \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \cdot \mathbf{v}_{m_g} \end{bmatrix} - \mathbf{b}_{gi} - \mathbf{S}_{gi} (\boldsymbol{\omega}_{ib}^b)_i + \boldsymbol{\Delta}_{(\boldsymbol{\omega}_{ib}^b)_i} \quad (6.24)$$

As per Section 5.5.1, the differential equation connecting $\boldsymbol{\omega}_{nb}^b$ to the attitude parameters is defined after [Zhao, 2016] for the RPH attitude model using (5.14). The partial derivatives of (6.24) may be used to populate the corresponding non-null submatrices of the design matrix as follows for the RPH attitude model:

$$\mathbf{C}_{gi,a} = \begin{bmatrix} \mathbf{0}_{1,3} & \mathbf{v}_1^T \mathbf{D} \\ \vdots & \vdots \\ \mathbf{0}_{1,3} & \mathbf{v}_{m_g}^T \mathbf{D} \end{bmatrix} \quad (6.25)$$

$$\mathbf{C}_{gi,imui} = \begin{bmatrix} & \omega_1 & \cdots & 0 & & \\ \mathbf{I}_{m_g} & \vdots & \ddots & \vdots & \mathbf{0}_{m_g, m_s} & \mathbf{0}_{m_g, m_s} \\ & 0 & \cdots & \omega_{m_g} & & \end{bmatrix} \quad (6.26)$$

The equations developed through this Section are significant in that they allow for natural and intuitive modeling of atypical IMU sensors, regardless of whether they are included as sensors in a MIMU array or if they are being used as a single IMU in a PosAtt system.

6.4.3.1 *Demonstrating Compatibility Between the Atypical IMU Models and the Standard IMU Models*

One way of supporting the validity of the models for an atypical IMU sensor is by evaluating the derived equations for the special case of an IMU with standard three axis accelerometer and three axis gyroscope triads (i.e. the standard IMU configuration). For this special case, the sensing axes for the accelerometers and gyroscopes are coincident, and are defined as

$$\mathbf{u}_1 = \mathbf{v}_1 = [1 \quad 0 \quad 0]^T \quad (6.27)$$

$$\mathbf{u}_2 = \mathbf{v}_2 = [0 \quad 1 \quad 0]^T \quad (6.28)$$

$$\mathbf{u}_3 = \mathbf{v}_3 = [0 \quad 0 \quad 1]^T \quad (6.29)$$

Taking these sensing axis definitions, the specific force vector for the i th IMU will simply be $(\mathbf{f}_{ib}^b)_i = [f_x \quad f_y \quad f_z]^T$, which is identical to the definition of the specific force vector for a standard three axis accelerometer triad. Additionally, the angular rate vector for the i th IMU will be $(\boldsymbol{\omega}_{ib}^b)_i = [\omega_x \quad \omega_y \quad \omega_z]^T$, which is identical to the definition of the angular rate vector for a standard three axis gyroscope triad.

The overall modeled specific force vector, as defined in (6.18), may be evaluated using the sensing axis definition from (6.27) to (6.29) to arrive at an observation model for the specific force vector which is identical to the standard IMU observation model in (6.8). Additionally, substituting the sensing axis definitions into the equations for the design matrix submatrices from (6.19) to (6.21) yields identical equations to the corresponding design matrix submatrices from (6.9) to (6.11).

The overall modeled angular rate vector, as defined in (6.24), may also be evaluated using the sensing axis definitions from (6.27) to (6.29) to arrive at an observation model for the angular rate vector which is identical to the standard IMU observation model in (6.12). Again, the sensing axis

definitions may be substituted into the equations for the design matrix submatrices in (6.25) and (6.26) to arrive at identical equations to the corresponding design matrix submatrices in (6.13) and (6.14).

This demonstrated compatibility between the model for the general, atypical sensor configuration and the standard three axis accelerometer/gyroscope triad IMU sensors supports the validity of the proposed observation models for the atypical IMU sensors.

6.5 Comprehensive Error Analysis

The measurement, process noise, and predicted system state residual vectors are estimable via the alternative derivation from [Wang, 1997] presented in Section 2.5.1.1 in the absence of state constraints, or via the alternative derivation presented in Section 3.3 in the presence of state constraints. These three residual vectors, and their associated covariances and redundancy contributions are valuable in providing comprehensive information regarding the performance of individual IMU sensors in an MIMU array without requiring a ground truth solution or simulated data.

When focusing on comparing the performance of the constituent sensors in a MIMU array, the observation residual vector is the most obviously useful quantity. The GMIS is unique in its ability to directly estimate the observation residuals for the raw IMU gyroscope and accelerometer measurements, and these quantities are readily used for Variance Component Estimation (VCE) to characterize individual sensor performance in a PosAtt system. The modeling performed in Section 6.4 is well-suited to leveraging VCE to characterize individual IMU sensor performance for a MIMU array.

As with every use of VCE, the specific application of VCE must inform the selection of the subsets of the residual vectors being used for VCE in order maximize the amount of meaningful information that is provided. This chapter's primary focus is on characterizing individual IMU performance in a MIMU array, so the following sets of variance components are estimated:

For the measurement residuals:

- i. An overall variance component describing the performance of the entire observation vector taken as a whole. This provides an overview of where there may generally be issues in the observations, regardless of their specific source.
- ii. Separate variance components dedicated to characterizing the performance of the raw gyroscope and accelerometer measurements for each IMU sensor in the MIMU array. This provides overall performance measures for the accelerometer/gyroscope triads, which can be

directly compared across different IMUs to evaluate their relative performance over the observation period.

- iii. Variance components dedicated to characterizing the performance of the GPS double-differenced observations, including ones for the L1 C/A, L1 Carrier Phase, and L2 Carrier Phase observations. These quantities are not tied to the quality of the raw IMU observations, but provide contextual information that may help clarify any particular issues in the dataset.

For the process noise residuals, an overall variance component describing the performance of the entire process noise vector as a whole is used. This provides information about the general performance of the kinematic models used over the course of the observation period.

Since the overall DOF for a Kalman Filter is equal to the number of recorded observations plus the number of system state constraints at any given point in time, and the sum of the redundancy contributions across *all three* residual vectors will be equal to the overall DOF, some elements of each residual vector will generally have smaller contributions to the overall system redundancy at an observation epoch. Practically speaking, this means that the ability to reliably determine the associated variance components over a short time window may be limited, particularly for the process noise variance component since the process noise redundancy contribution is typically very small. This chapter follows the convention set in previous chapters of estimating the variance components over a period of time using the cumulative sum of both the residual subsets' quadratic forms and their corresponding redundancy contributions. This results in more reliable estimates of the variance components, while sacrificing their time domain resolution.

6.6 Processing Workflow

At its core, the processing workflow used in this chapter is identical to the workflow that was developed for the PosAtt Engine in Section 4.6 and that was expanded to accommodate multiple attitude models in Section 5.7. The workflow for the processing module required slight modification to accommodate multiple IMU sensors in the PosAtt system, and the modified workflow developed for this chapter is shown in Figure 6.6.1. It is worth noting that this workflow also includes the functionality of the workflow developed for Section 5.7, so this processing workflow accommodates both MIMU array data and multiple attitude models. The combination of these functionalities is not explored further in this chapter but could be explored in future work.

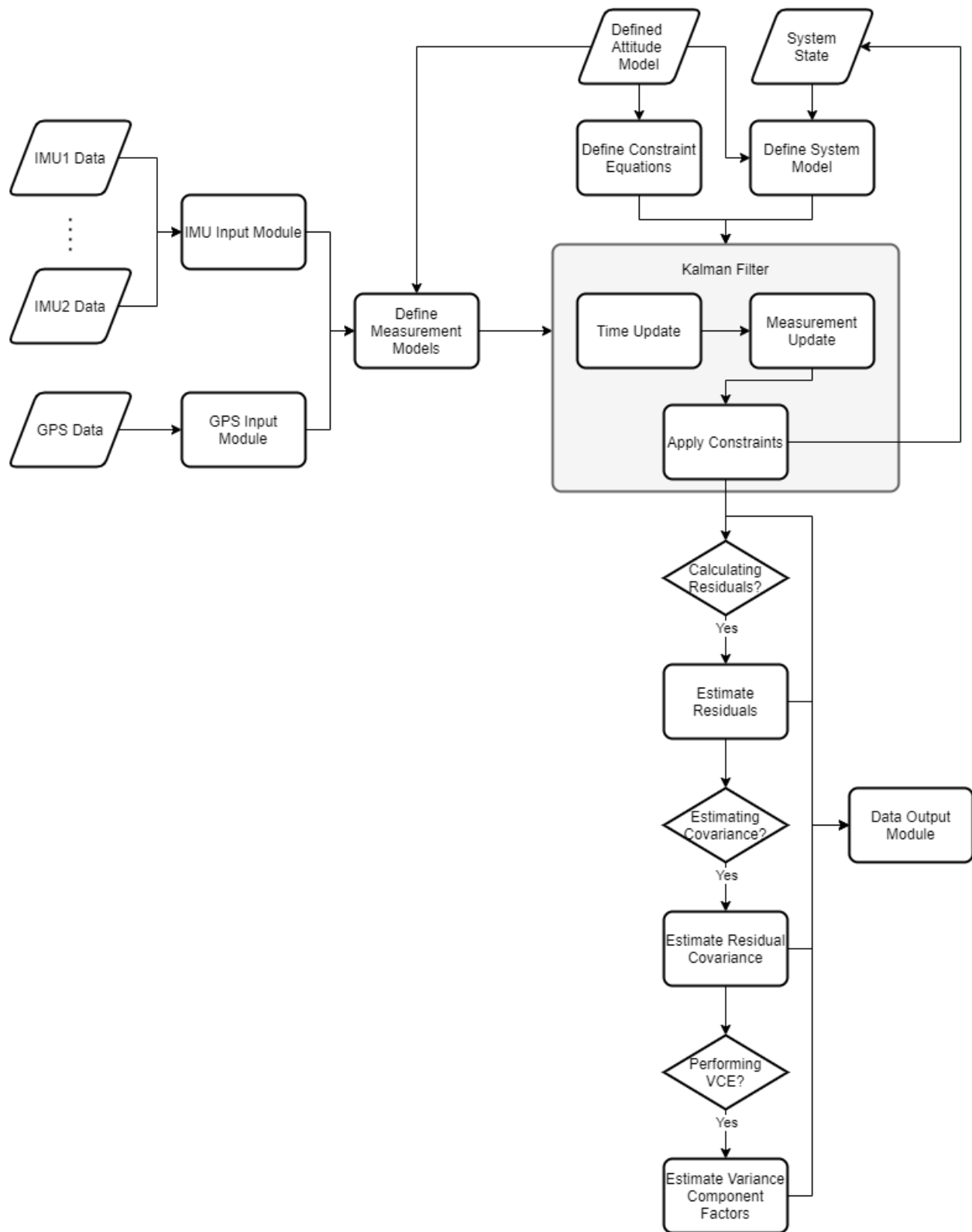


Figure 6.6.1: The modified processing workflow to accommodate an arbitrary number of IMU sensors in a MIMU array.

6.7 Land Vehicle Test Dataset and Analysis of Results

This section presents the road test dataset used to validate the MIMU array modeling techniques used in this research, and provides an analysis of the performance of each IMU in the MIMU array.

6.7.1 Definition of the Road Test Dataset

The test dataset used in this chapter was collected using an in-house integrated navigation system that consists of the following:

- Three VectorNav VN-100 IMUs operating at 100 Hz.
- Two high-rate NovAtel OEM6 receivers operating at 100 Hz, with one receiver mounted as part of the PosAtt system and the other operating at a fixed location to act as a base station for the calculation of the double-differenced GPS observations.

The ground vehicle used to collect the test dataset remained stationary for the first 5 minutes of the dataset, after which it spent approximately 5 minutes moving in circles about a residential court. This was followed by another 5 minute stationary period, after which the vehicle was driven in kinematic for approximately 30 minutes.

A top-down view of the trajectory is shown in Figure 6.7.1, with the corresponding velocity and acceleration profiles displayed in Figure 6.7.2 and Figure 6.7.3, respectively. The estimated roll, pitch, and heading profiles are shown in Figure 6.7.4, and their estimated first-order time derivatives are presented in Figure 6.7.5.

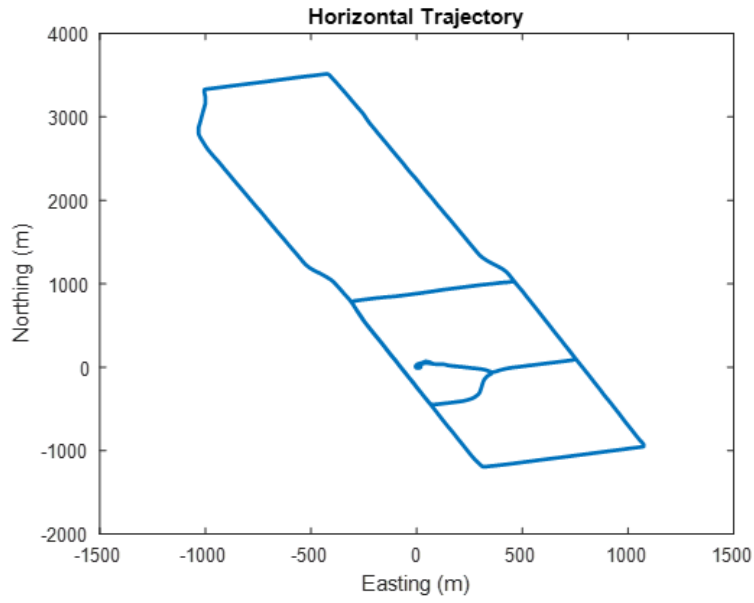


Figure 6.7.1: A top-down view of the trajectory of the kinematic dataset. Coordinates are expressed in the local geodetic coordinate system with an origin coincident to the vehicle's starting location.

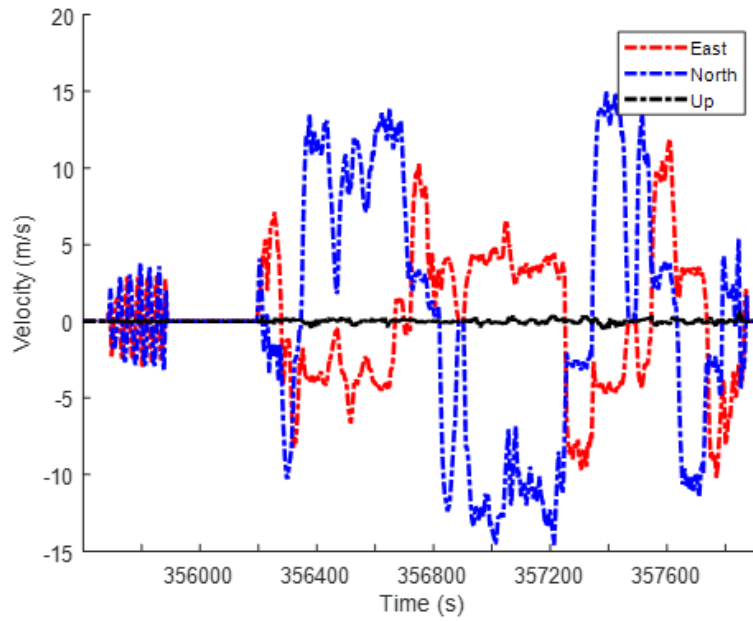


Figure 6.7.2: The velocity profile for the kinematic dataset. The velocity components are expressed in the navigation frame's local geodetic coordinates.

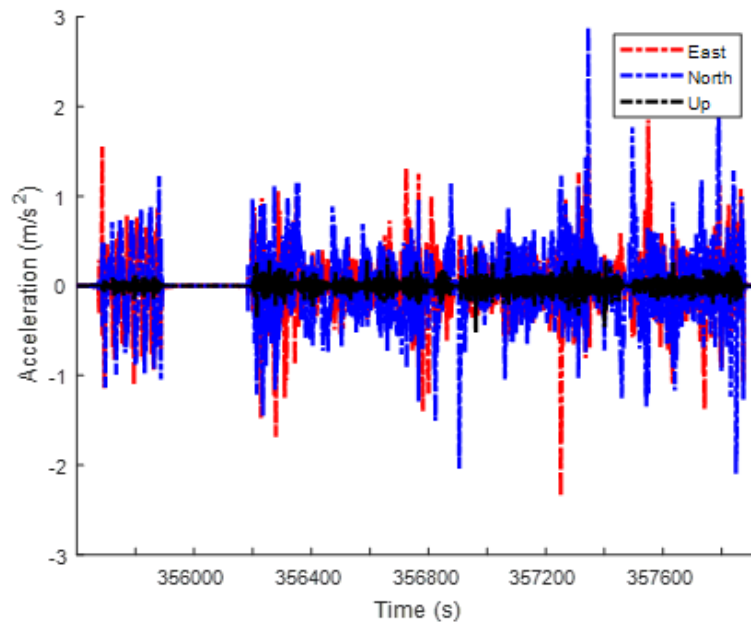


Figure 6.7.3: The acceleration profile for the kinematic dataset. The acceleration components are expressed in the navigation frame's local geodetic coordinates.

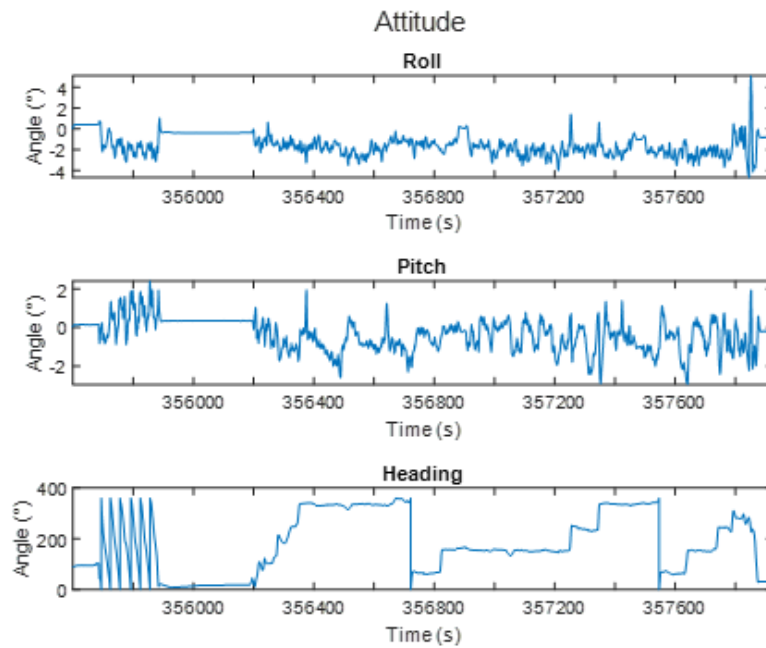


Figure 6.7.4: The roll, pitch, and heading profiles for the kinematic dataset. The attitude is presented relative to the navigation frame's local geodetic coordinates.

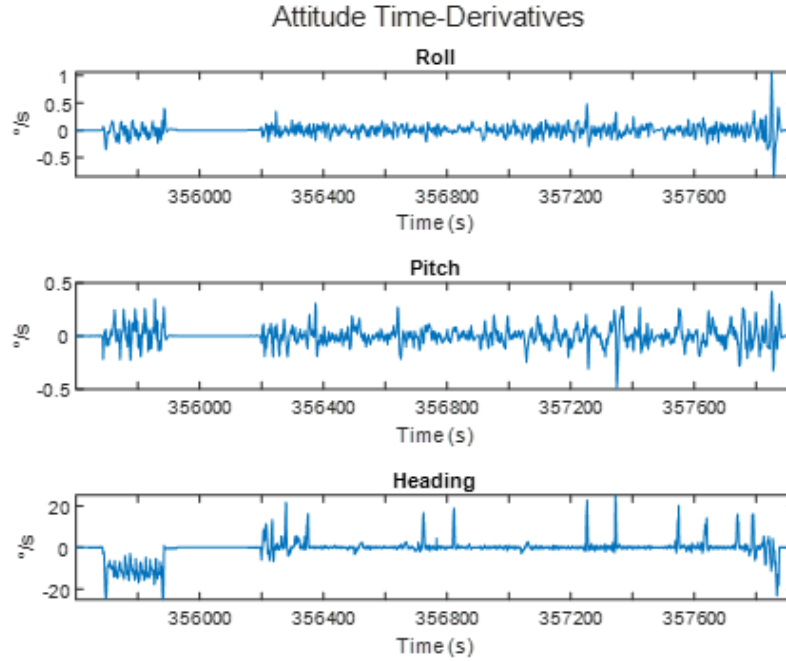


Figure 6.7.5: The estimated first-order time derivatives for the roll, pitch, and heading values for the kinematic dataset. These values are expressed relative to the navigation frame's local geodetic coordinates.

Using the GMIS, the data from all three constituent IMU sensors were able to be directly integrated via the EKF measurement update without requiring the generation of a “fused” dataset. This novel approach to accounting for the MIMU array output provides unique opportunities for further analysis, with the following opportunities explored in more detail in this chapter:

- Directly estimating the individual systematic errors for each sensor in the MIMU array, without any assumption of their consistency across different IMU sensors. This is further explored in Section 6.7.2.
- Using VCE to characterize the performance of each sensor in the MIMU array by using the estimated raw observation residuals along with their contributions to the overall system redundancy. This is further explored in Section 6.7.3.
- Using *a posteriori* estimated solution accuracy to directly compare the position/orientation estimation performance of the MIMU array data with that of a Single IMU (SIMU), without the need for a simulated dataset or a ground truth solution. This is further explored in Section 6.7.4.

6.7.2 Estimating Sensor Systematic Errors and Measurement Residuals

A unique factor to the GMIS lies in its ability to directly model raw IMU measurements. This allows for observation residuals to be estimated for the raw specific force and angular rate observations after the EKF's measurement update, rather than requiring the use of the system innovation vector to characterize the sensors' performance. For data from a MIMU array, the IMUs are modeled individually in the positioning EKF under the GMIS, and this allows for separate sets of raw IMU measurement residuals to be estimated for all sensors comprising the MIMU array.

For the kinematic test dataset, the estimated accelerometer specific force residuals for each IMU sensor in the MIMU array are shown in Figure 6.7.6, and the gyroscope angular rate residuals for each IMU sensor in the MIMU array are shown in Figure 6.7.7. These quantities are not estimable under the traditional sensor integration strategy. Histograms of the individual IMUs' standardized residual values are provided in Figure 6.7.8 for the accelerometer specific force observations, and in Figure 6.7.9 for the gyroscope angular rate observations. While not a focus in this chapter, the standardized GPS observation residual histograms are provided in Figure 6.7.10 for the sake of completeness and to provide context for some of the overall challenges that the kinematic test dataset presents.

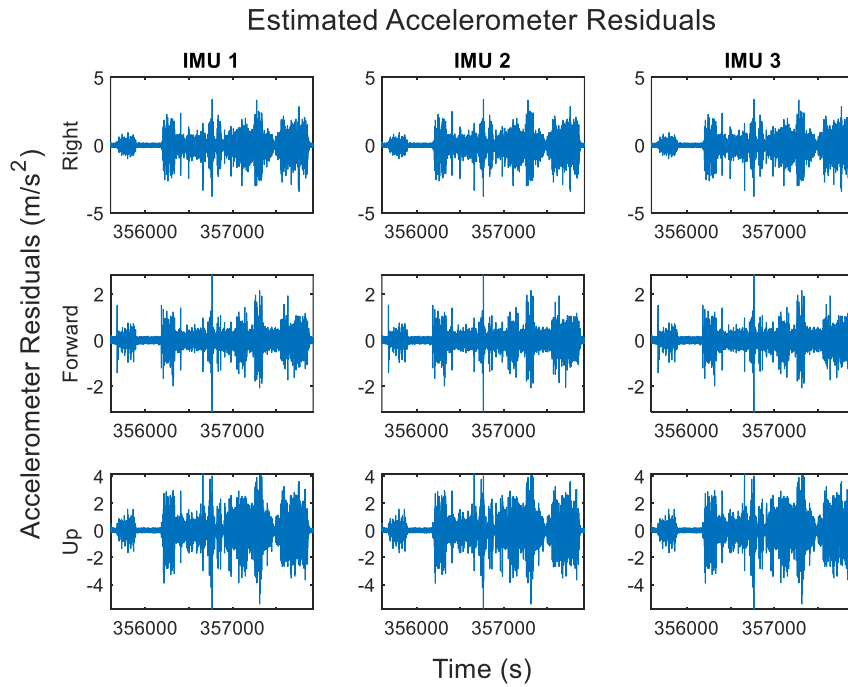


Figure 6.7.6: The estimated accelerometer specific force residuals for each IMU in the MIMU array over the kinematic test dataset.

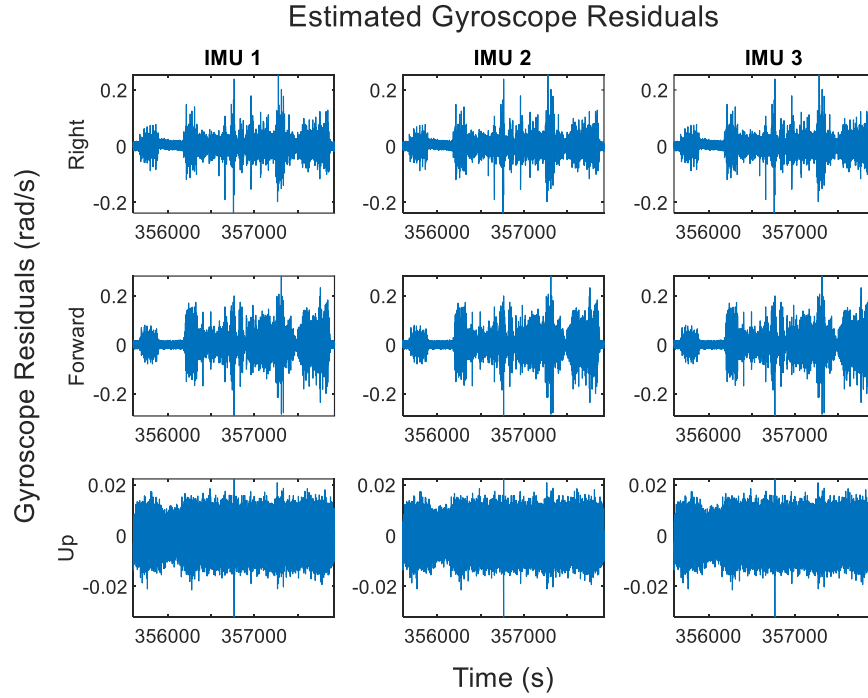


Figure 6.7.7: The estimated gyroscope angular rate residuals for each IMU in the MIMU array over the kinematic test dataset.

There are two main conclusions that may be drawn from the plots of the time-varying accelerometer and gyroscope residuals:

- i. The IMU accelerometer/gyroscope residuals' performance is very consistent across the three different IMU sensors in the MIMU array. Each of the sensors comprising the MIMU array in this test dataset are of the same model and expected to have similar performance. The consistency in the behaviour of their estimated residuals suggests that the IMU observations have been appropriately modeled and successfully integrated through the GMIS.
- ii. The behaviour of the IMU accelerometer/gyroscope residuals through time is highly dependent upon the time-varying dynamics that the PosAtt system is experiencing over the course of the test dataset. The static portions of the dataset are clearly identifiable in Figure 6.7.6 and Figure 6.7.7 as the periods of time with generally much smaller residual values, and the times in the dataset with the highest IMU random errors occur when the system is undertaking a maneuver. This phenomenon is not surprising, especially considering that confounding factors such as vibration of the PosAtt system also vary with the system dynamics.

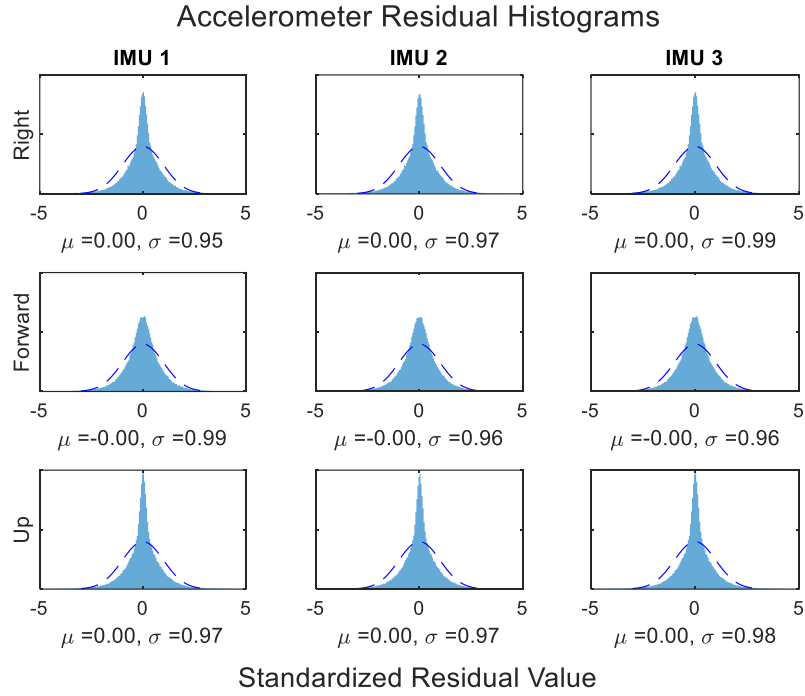


Figure 6.7.8: Histograms of the standardized residual values for the accelerometer specific force observations. The standard normal distribution is superimposed for reference.

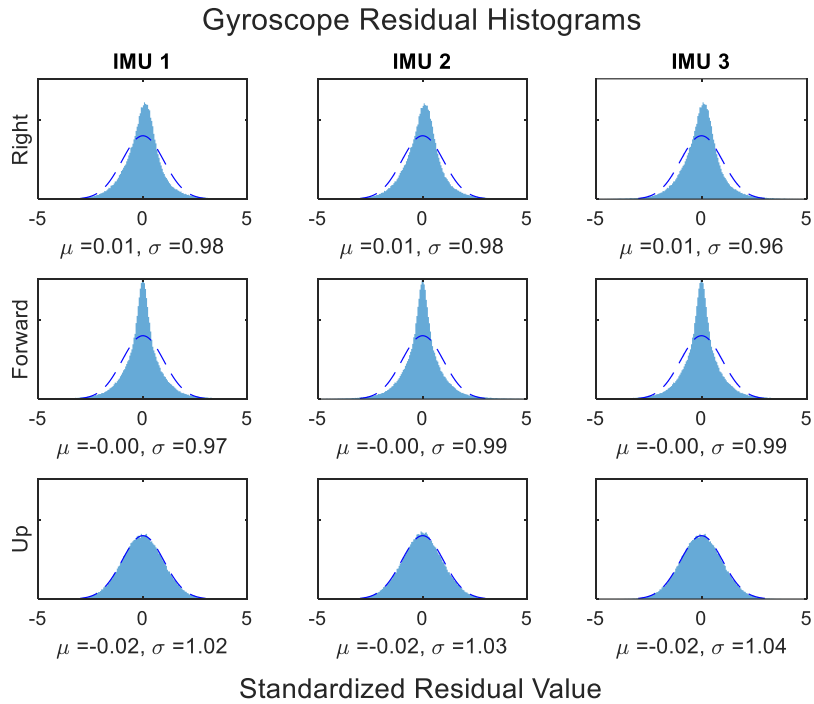


Figure 6.7.9: Histograms of the standardized residual values for the gyroscope angular rate observations. The standard normal distribution is superimposed for reference.

The IMU measurement residual histograms in Figure 6.7.8 and Figure 6.7.9 show that the IMU accelerometer/gyroscope observations behaved very well over the course of the kinematic test dataset. Many of the histograms shown in these two figures have sharp peaks near zero, which may be explained by the fact that the IMU sensors experienced lower random noise when the system was stationary/experiencing very low dynamics, and provides further evidence to support this conclusion.

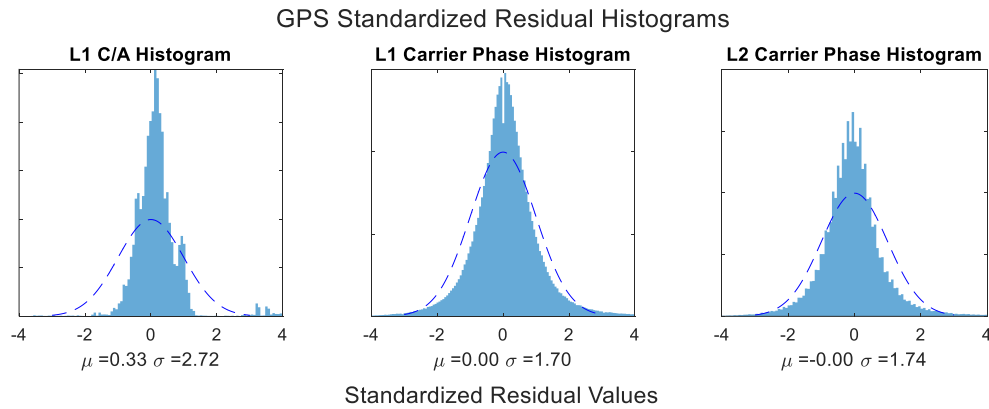


Figure 6.7.10: Histograms of the standardized L1 C/A, L1 Carrier Phase, and L2 Carrier Phase residuals over the entire study period. The standard normal distribution is superimposed for reference.

It is worth noting that the GPS measurement residuals show that the GPS measurements generally behaved more poorly than would have been expected over the kinematic test dataset. The test dataset was collected in a residential neighbourhood with many visual obstructions of the sky, and as a result satellites were dropped frequently over the duration of the test dataset. At several points, this required changing the “base” satellite for generating the GPS double-differenced observations, and whenever this happened, there was an associated period of estimating the new float ambiguity parameters. There were even several points in time that the number of common visible satellites between the base OEM6 receiver and the rover OEM6 receiver fell below the requisite four for reliable position estimation. The L1 C/A observations in particular had a relatively high proportion of values that were not quite flagged as outliers but had large standardized residual values.

Modeling the MIMU array in the GMIS also allows for individualized sensor systematic error estimation, i.e., the estimation of the individual biases/scale factor errors associated with each accelerometer and gyroscope. These estimates are shown in Figure 6.7.11 and Figure 6.7.12 for the accelerometer biases and scale factor errors, respectively, and in Figure 6.7.13 and Figure 6.7.14 for the gyroscope biases and scale factor errors, respectively.

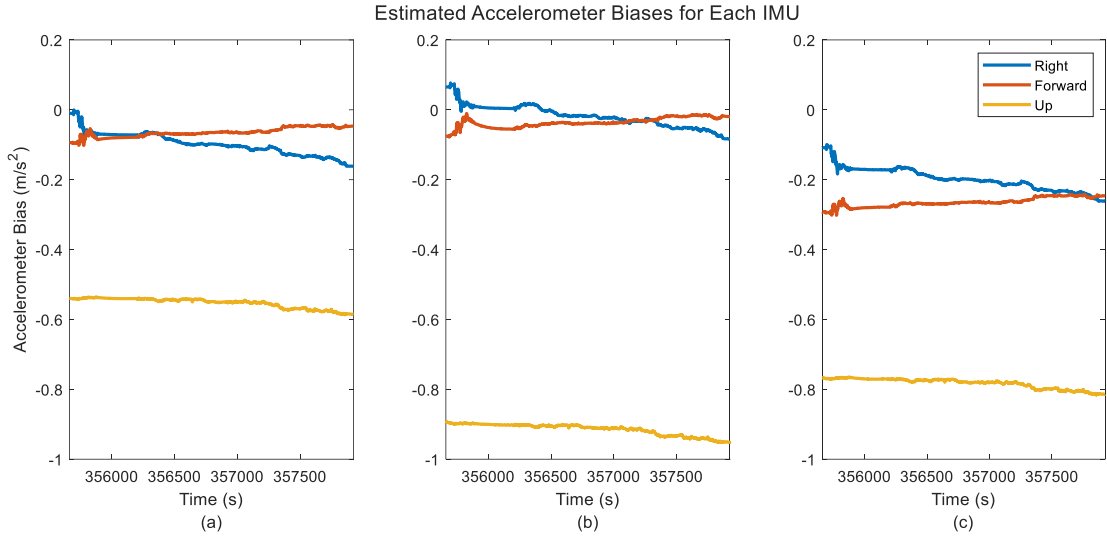


Figure 6.7.11: The estimated accelerometer biases for the individual accelerometers in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.

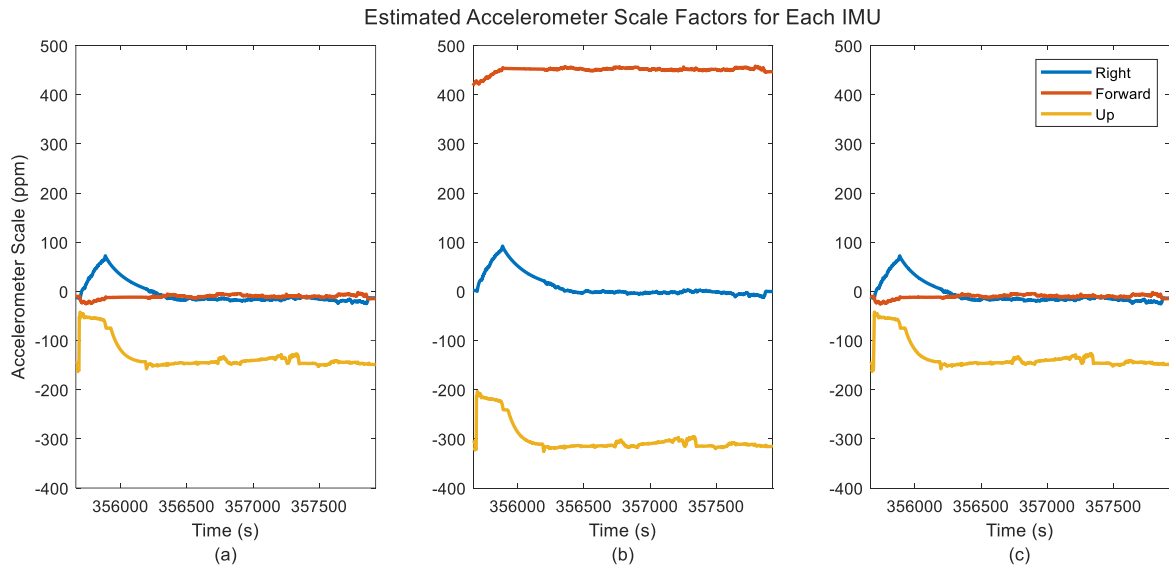


Figure 6.7.12: The estimated accelerometer scale factor errors for the individual accelerometers in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.

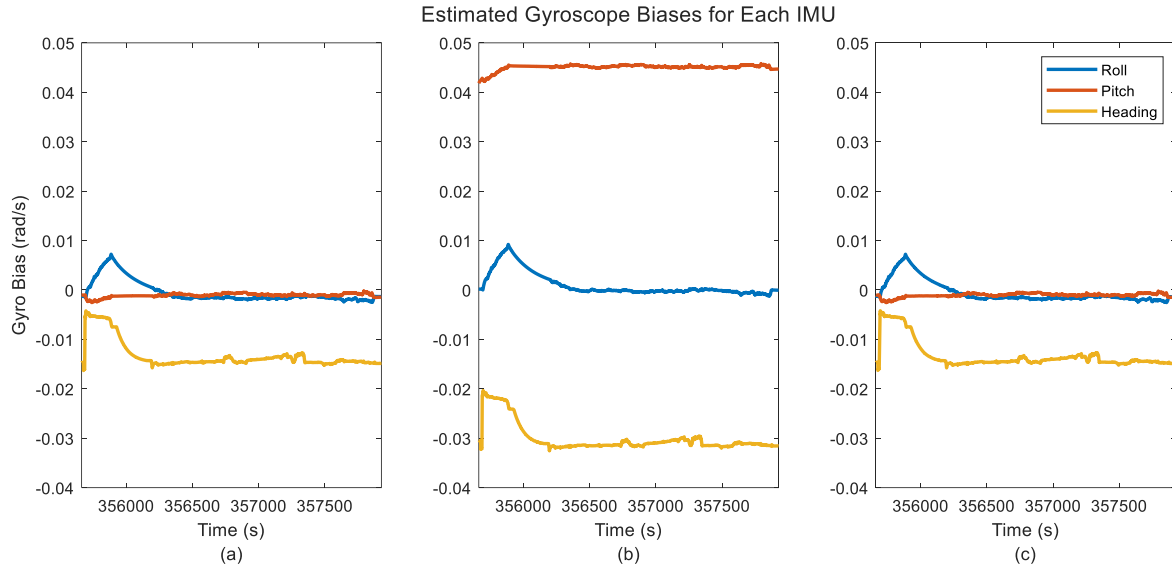


Figure 6.7.13: The estimated gyroscope biases for the individual gyroscopes in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.

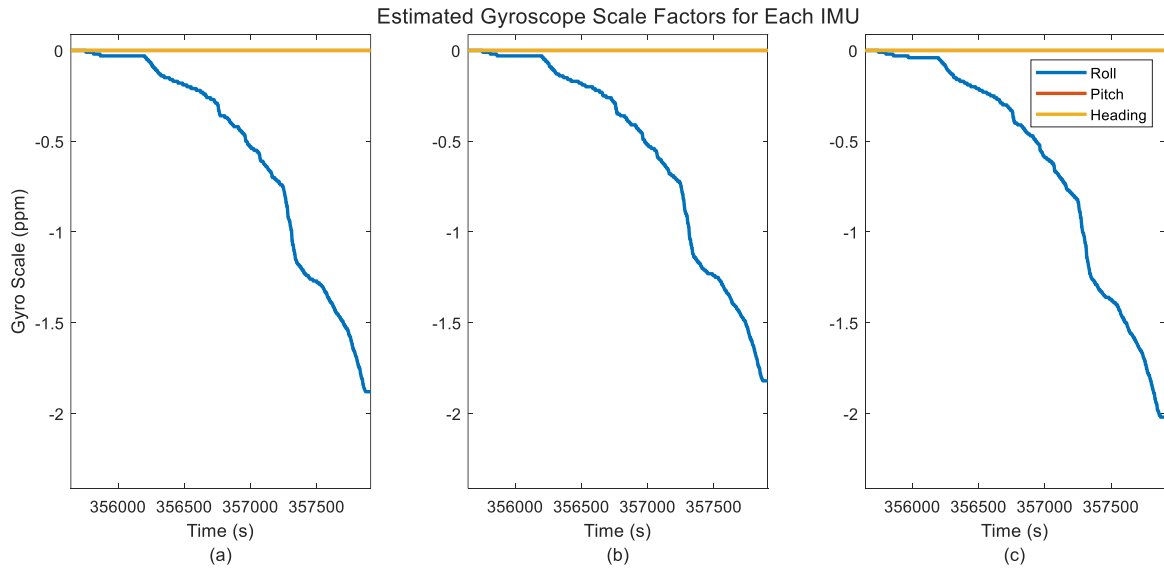


Figure 6.7.14: The estimated gyroscope scale factor errors for the individual gyroscopes in (a) the first IMU in the array, (b) the second IMU in the array, and (c) the third IMU in the array.

In Figure 6.7.11 and Figure 6.7.13, it is clear that the individual IMU sensor biases remain relatively stable through time over the kinematic dataset, with the accelerometer biases experiencing more apparent sensor drift than the gyroscope biases. This is to be expected for the tactical-grade IMUs used in this research, even though they are MEMS sensors. It is also clear that the bias estimates are different for each IMU accelerometer and gyroscope in the array, which clearly suggests that they have

been accounted for properly and supports the assertion that their systematic errors are generally not homogeneous.

In Figure 6.7.12 and Figure 6.7.14, the individual IMU sensor scale error estimates generally remain consistent over the course of the test dataset, with very little if any scale error wander over the observation period.

The scale error estimates in Figure 6.7.14 remain at or near zero for the entire duration of the dataset. The initial value of each of the gyroscope scale error state elements was zero, so this indicates that these systematic errors did not change significantly over the course of the dataset. This is explained by the fact that scale error represents an error that increases with the magnitude of the observations being recorded. Considering an observation precision of approximately $1^\circ/\text{s}$ in the gyroscope observations (this precision is based on the number of significant digits provided in the IMU output), and a maximum gyroscope observation magnitude of approximately $25.5^\circ/\text{s}$ in the test dataset, a scale error of at least 650 ppm would be necessary to even produce a potentially detectable effect in the observations. This suggests that the gyroscope scale factor errors are not observable over the test dataset due to the fact that there were not highly varying attitude dynamics over the kinematic test dataset.

As was the case with the accelerometer and gyroscope bias estimates, the estimated scale factor errors for the accelerometer observations are generally different for different sensors in the MIMU array, which suggests that the sensors should be modeled individually in the EKF, rather than using a group of shared states for multiple IMU sensors.

6.7.3 Characterizing Sensor Performance Using Variance Component Estimation

The modeling of individual IMU raw accelerometer and gyroscope observations allows for the estimation of their residual estimates as was done in Section 6.7.2. The calculation of these residual estimates additionally allows for the use of VCE to characterize the performance of different parts of the positioning EKF with the aid of the associated measurement redundancy indices. The variance components associated with each type of observable quantity provides an overview of how well the actual measurements fit their associated observation models on the whole. Practically, there are several types of variance component factors that may be calculated:

1. An overall standard error factor that includes all observation types. This standard error factor provides a sense of the overall performance of the observation models over the course of the dataset.
2. Variance component factors of the individual observation types. These variance components provide separate estimates of the performance of each observation type in the PosAtt system. They provide more granular information than the variance factor of unit weight, but at a lower redundancy that makes it more challenging to determine a reliable estimate of each variance component.
3. Variance component factors for individual observations of the same quantity through time. For example, this could include separate variance component factor estimates for each component of an IMU accelerometer/gyroscope triad. These provide the highest degree of detail into sensor performance with the trade-off of relying on individual observations' redundancy contributions, which are generally low compared to those for the overall variance factor of unit weight and the observation type variance component factors.

The overall standard error factor for the entire EKF as a LS system is shown in Figure 6.7.15, estimated over a 20 second moving window for the kinematic test dataset. From the plot, it is clear that the observations' overall quality was better when the system was stationary or experiencing less dynamic motion. For the non-stationary portions of the dataset, the standard error factors hovered at a value of approximately 1, indicating that the overall performance of the positioning sensors is in agreement with their assumed *a priori* observation accuracies. Note that this estimate uses information from all IMU observation residuals in addition to the GPS observation residuals.

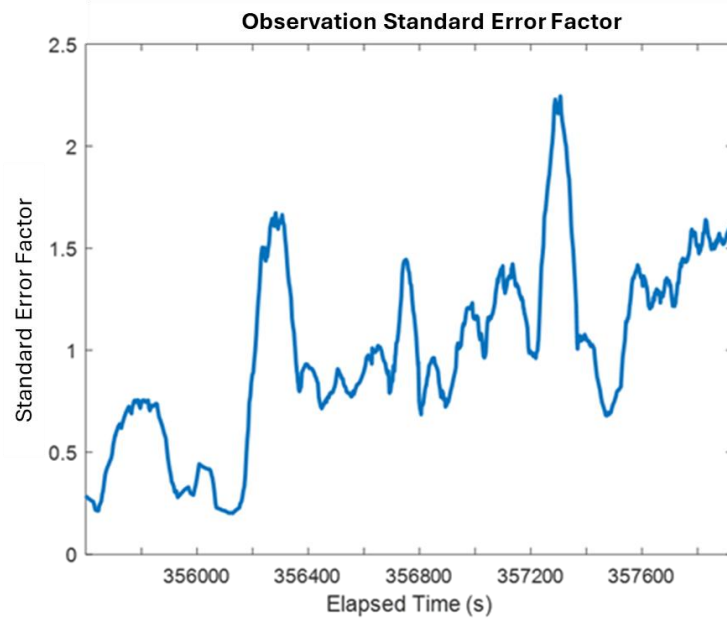


Figure 6.7.15: The estimated overall standard error factor for the MIMU-integrated system (moving window: 20s).

Looking more granularly at the dataset and estimating the observation standard error factors for each observation type in the MIMU array (i.e. accelerometers and gyroscope observations) provides the standard error component plots shown in Figure 6.7.16 for the accelerometer observations and Figure 6.7.17 for the gyroscope observations.

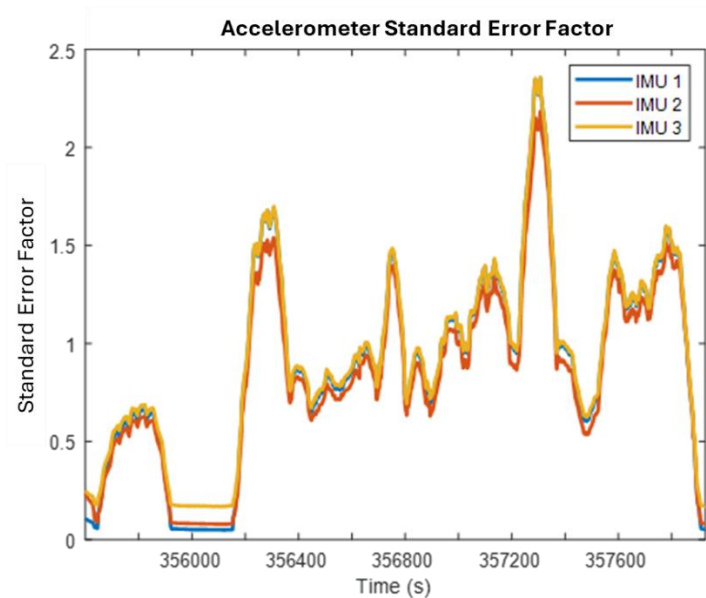


Figure 6.7.16: The estimated standard error factors for the specific force measurements of each IMU's accelerometer triad in the MIMU-integrated system (moving window: 20s).

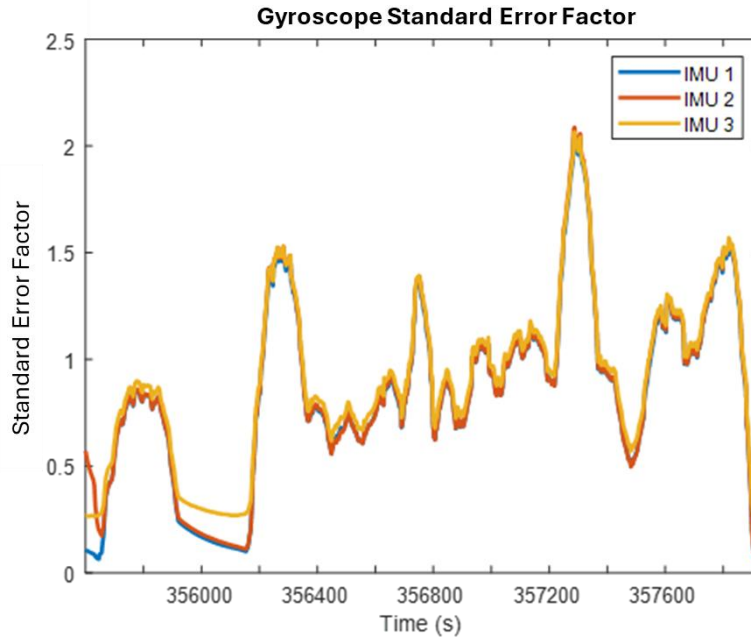


Figure 6.7.17: The estimated standard error factors for the specific force measurements of each IMU's gyroscope triad in the MIMU-integrated system (moving window: 20s).

The accelerometer standard error factor plots in Figure 6.7.16 and the gyroscope standard error factor plots in Figure 6.7.17 provide further evidence to support the conclusion that IMU performance varies with the system dynamics. During the stationary periods in the dataset, both of these plots indicate that the IMU sensor performance is better than expected for all constituent IMU sensors. During the kinematic portions of the dataset, both plots indicate that the IMU sensors performed as expected, with spikes in the standard error estimates at times that significant maneuvers were undertaken.

All three IMUs exhibited very similar behaviour in all three plots over the course of the kinematic dataset, with nearly identical standard error factor estimates over the entire observation period. This is encouraging, since all IMUs are of the same model, and it is very instructive to see them exhibit such similar random error behaviours when their systematic error estimates are generally quite different from one another. The similarity between their standard error factor estimates also indicates the extent to which the system dynamics define the observation noise for each sensor. These results indicate that IMU observations in particular would benefit from being integrated via an adaptive filter that uses the system dynamics to weight the IMU observations' importance in the positioning EKF.

All of the standard error factor plots presented in this section use windowed estimates of the standard error factors for each quantity being studied. It is also possible to estimate global values of

these quantities over the course of the entire dataset to estimate the *a posteriori* observation noise for each observation type in the dataset. These *a posteriori* estimates are provided in Table 6.7.1 for the IMU observation accuracies.

Table 6.7.1: The global a posteriori standard deviations for each observation type in the MIMU array. These values were calculated by scaling the a priori standard deviation estimate for the accelerometer and gyroscope observations by the corresponding global estimate of the standard error factor. Values were estimated separately for each IMU in the MIMU array.

IMU Sensor Number	<i>A Posteriori</i> Standard Deviations	
	Accelerometer [m/s²]	Gyroscope [°/s]
1	0.110	0.803
2	0.108	0.806
3	0.105	0.799

6.7.4 Comparing Multi-IMU Results to Single-IMU Results

The *a posteriori* standard deviations also provide an opportunity to draw direct comparisons between the performance of different configurations of sensors over the same dataset, without requiring a ground truth solution or simulated data. Practically speaking, different subsets of the available positioning sensors may be used to estimate the system's trajectory, and then their *a posteriori* standard errors may be used to determine the *a posteriori* positioning/orientation solution accuracy using that particular sensor configuration.

This section compares the quality of the solutions between the positioning/orientation solution using all sensors in the MIMU array, and the positioning/orientation solution for the Single-IMU (SIMU) solution using IMU Sensor 1. The time-varying *a posteriori* standard deviations for the system state elements for both the MIMU and SIMU solutions may be compared directly using their ratio over the course of the dataset to characterize their relative performance. Taking the MIMU standard deviation as the numerator and the SIMU standard deviation as the denominator, the difference between the calculated ratio and one provides information about the average accuracy improvement that the MIMU solution offers over the SIMU solution.

A comparison between the MIMU and SIMU solutions' *a posteriori* position accuracies is shown in Figure 6.7.18. Comparisons between the MIMU and SIMU solutions' *a posteriori* attitude accuracies and attitude time derivative accuracies are shown in Figure 6.7.19 and Figure 6.7.20, respectively.

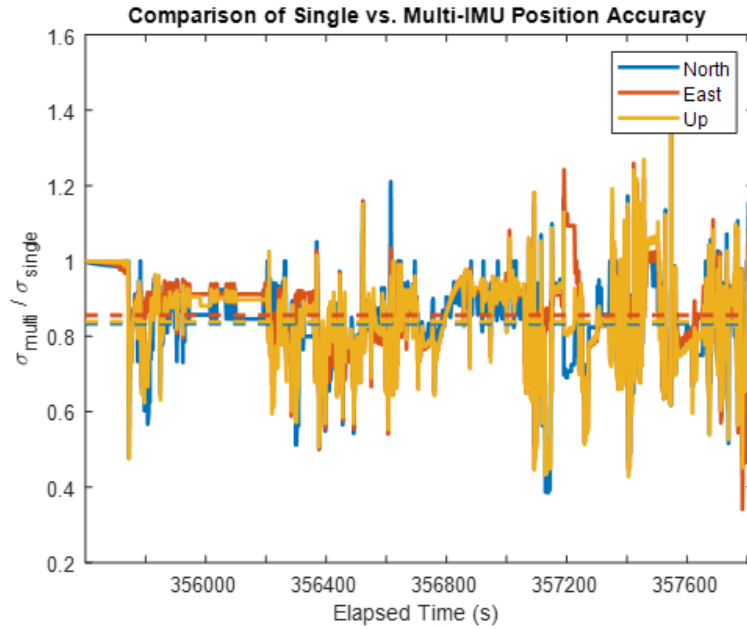


Figure 6.7.18: The ratio of the *a posteriori* MIMU position standard deviations to the *a posteriori* SIMU position standard deviations. The horizontal dashed lines show the average ratios for the component of the associated colour.

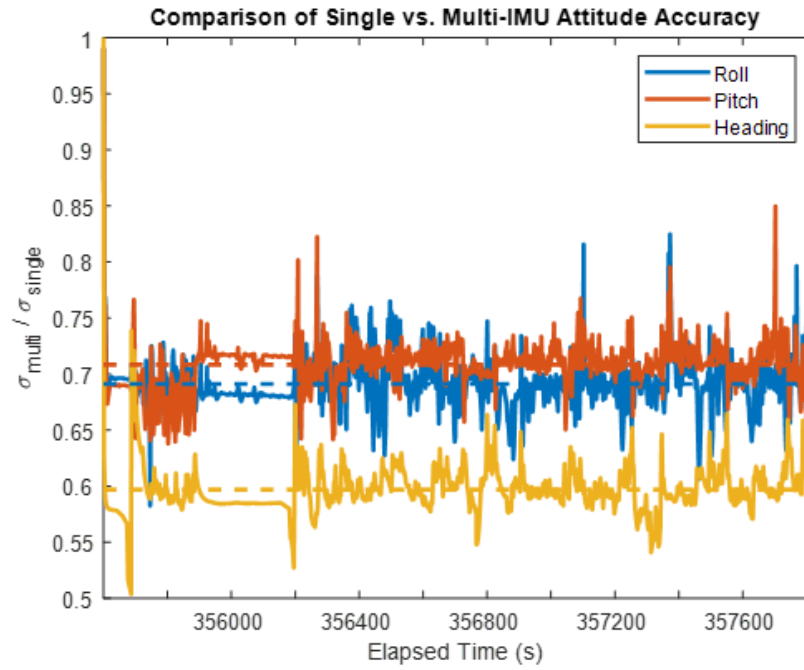


Figure 6.7.19: The ratio of the a posteriori MIMU attitude standard deviations to the a posteriori SIMU attitude standard deviations. The horizontal dashed lines show the average ratios for the quantity of the associated colour.

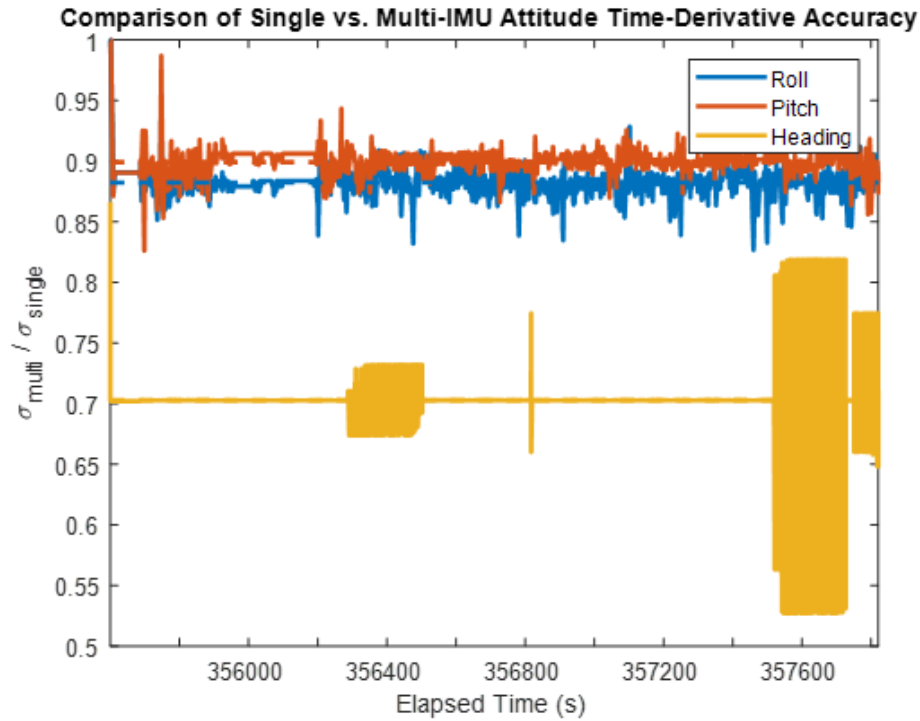


Figure 6.7.20: The ratio of the a posteriori MIMU attitude time derivative standard deviations to the a posteriori SIMU attitude time derivative standard deviations. The horizontal dashed lines show the average ratios for the quantity of the associated colour.

From these three figures, it is clear that the MIMU generally presents a significant improvement in the accuracy of the system's position, attitude, and attitude rate-of-change over the course of the kinematic test dataset. This is to be expected, since the addition of extra IMU sensors to the PosAtt system adds redundancy to the system.

In Figure 6.7.18, the data from the MIMU array provides more significant improvements in the overall estimated position accuracy when the system is in motion than when it is stationary. The position accuracy comes primarily from the GPS observations, so the MIMU and SIMU solutions are much closer in their accuracy than they are for the attitude and attitude rate-of-change.

Figure 6.7.19 shows a significant improvement in the attitude accuracy for the MIMU solution over the attitude accuracy for the SIMU solution, in fact showing the most dramatic improvement of all three of these figures. There are several important factors that contribute to the MIMU solution's dramatic improvement of the attitude-related state elements and how this improvement changes over time:

- Since the constituent IMU sensors in the MIMU array are assumed to have a constant relative geometry with respect to one another (i.e. they are assumed to conform to rigid body motion). This results in more reliable ongoing estimates of the individual sensor biases than would be possible for the SIMU solution, which also improves the quality of the data coming from each of the individual IMU sensors in the MIMU array. As long as the assumed rigid body motion holds true, even the individual sensors in the MIMU array can be expected to outperform a single IMU being used on its own.
- The roll and pitch can generally be estimated reliably using just IMU accelerometer data, but the system heading cannot be. This means that the MIMU array provides a more significant improvement to the heading estimation than to the roll and pitch estimation.
- The kinematic dataset used in this chapter was land vehicle-based, with the roll and pitch generally experiencing much lower system dynamics. This results in the time-varying improvements of attitude accuracy remaining much more consistent through time than for the positioning solution, and these improvements vary much less over the course of the dataset.

The average improvement of the *a posteriori* MIMU positioning solution accuracy is presented in Table 6.7.2.

Table 6.7.2: Characterization of the MIMU solution performance improvement compared to the SIMU solution performance, expressed as a percentage improvement in the estimated a posteriori standard deviations of corresponding quantities.

	MIMU Performance Improvement over the SIMU		
	East	North	Up
Position	14.4%	16.6%	16.1%
Attitude	Roll	Pitch	Heading
	29.6%	31.1%	40.2%
Attitude Rate-of-Change	Roll	Pitch	Heading
	11.7%	10.1%	29.7%

It is also valuable to compare the estimated overall standard error components for the SIMU solution with those of the MIMU solution. The SIMU-derived overall standard error components are

shown in Figure 6.7.21. As expected, this plot exhibits very similar behaviours to the overall standard error plot in Figure 6.7.15, indicating that the overall system performance relative to the *a priori* assumed observation accuracies is consistent for both the MIMU and SIMU solutions.

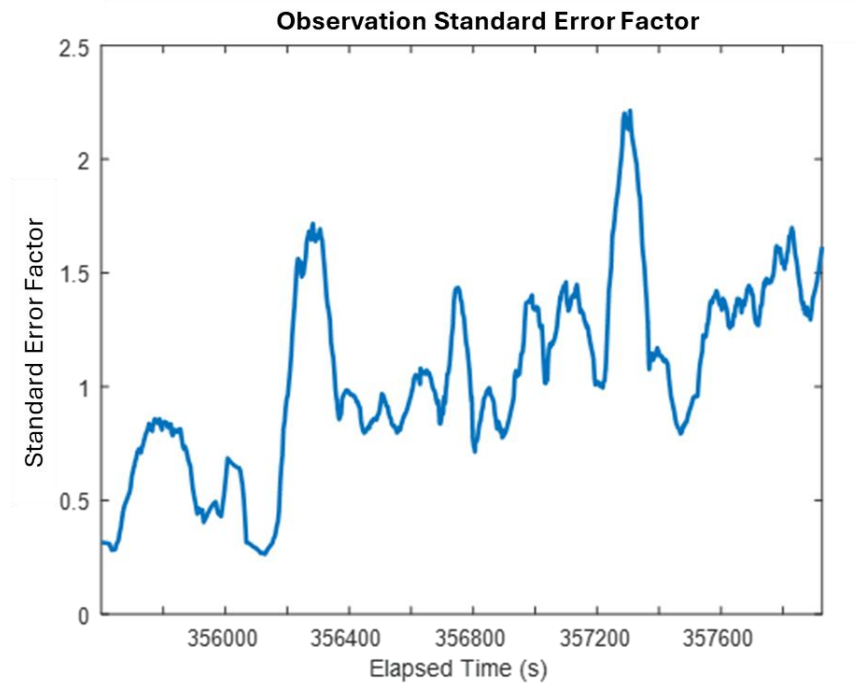


Figure 6.7.21: The estimated overall standard error factors for the SIMU-integrated system (moving window: 20s).

6.8 Summary

This chapter presents a general realization of the GMIS to accommodate MIMU array data. The methods developed in this chapter accommodate MIMU arrays of arbitrary size and configuration, and they fit neatly within the standard GMIS workflow without requiring significant modifications to it. This further cements the GMIS as a robust processing strategy that can readily accommodate any configuration of position/orientation sensors.

With the proposed modeling techniques in this chapter, it is possible to model individual IMUs in the MIMU array and calculate individual, time-varying estimates of their systematic error parameters. This opens up opportunities for further analysis, namely:

- Rigorous Fault Detection and Exclusion (FDE) for each sensor in the MIMU array. The individual modeling of IMU sensors allows for direct estimates of their raw observation residuals to be

calculated, and these provide an intuitive way to statistically evaluate the observations from constituent IMU sensors for outliers.

- Evaluating the stability of each constituent IMU sensor in the MIMU array over time. The fact that the IMU systematic errors are included as states in the positioning EKF allows for their estimates to change with time to reflect the current behaviour of an IMU sensor at any given observation epoch. Knowing whether conditions are such that the MEMS IMUs are displaying unstable behaviour through time is valuable for high-accuracy applications.
- Evaluating the performance of each constituent IMU sensor in the MIMU array using Variance Component Estimation (VCE). Specified sensor accuracy typically presents a best case scenario for observation quality, and it is important to evaluate how well a given IMU is actually performing over a kinematic dataset. This is especially true for MEMS IMU sensors, that have their stochastic properties particularly strongly influenced by factors such as environmental conditions.

Additionally, the modeling in this chapter enables more intuitive modeling of atypical IMUs that do not have the standard arrangement of mutually-orthogonal accelerometer/gyroscope triads. Under the traditional sensor integration strategy, the data from an IMU sensor is integrated via its INS dead-reckoning solution. This requires the differences in atypical sensors to be accounted for in the IMU mechanization equations, and this makes it much harder to practically use the data from these sensors. From the generalized IMU accelerometer/gyroscope models in this chapter, it is possible to directly integrate atypical sensors' data through the positioning EKF under the GMIS knowing only the direction of their sensing axes in the IMU's body frame. This makes incorporating unconventional IMU sensors into a PosAtt system much more intuitive and accessible when using the GMIS.

All of this additional flexibility and robust processing capability does come with significant performance costs. Considering the sensor biases and scale factor errors to be included in the system states for each IMU individually, 12 additional state vector elements are produced per IMU in the MIMU array. This significantly increases the dimension of the optimization problem being solved in the positioning EKF, which normally would have just 15 core states outside of the IMU systematic error parameters. This presents several potential avenues for improvement of the GMIS' computational efficiency:

- Pre-filtering the IMU data to reduce the overall data rate from each sensor could potentially drastically reduce performance costs during periods of time where the system is experiencing low dynamics.
- Since the IMU systematic error parameters are not expected to change rapidly, new estimates of the sensor biases and scale factor errors do not need to be calculated at every observation epoch. If these parameters were updated through a secondary Kalman Filter that was run at a lower rate than the positioning EKF, this could potentially negate the associated performance cost of using IMU array data in the GMIS.

7 Analytical Framework for Online Calibration of Sensor Systematic Errors Under the Generic Multisensor Integration Strategy

This chapter is based upon the published work:

Brunson, B., and Wang, J. (2025): Analytical Framework for Online Calibration of Sensor Systematic Errors Under the Generic Multisensor Integration Strategy. *Sensors*, 25(10), 3239.

7.1 Preamble

When using low-cost sensors in high-accuracy environments, it is crucial to properly account for the systematic errors that impact observation quality. This is doubly important when using MEMS sensors, since they exhibit unstable behaviour through time and are susceptible to environmental effects such as changes in ambient temperature. Systematic errors generally fall into one of two categories:

- i. Intrinsic, or internal to the sensor. This includes factors such as sensor biases and scale factor errors, as well as environmental factors such as the effects of the ambient temperature on a sensor's electrical or mechanical systems.
- ii. Extrinsic, or external to the sensor. These systematic errors typically arise from the geometric relationship between different constituent sensors in a PosAtt system, with the most common examples being lever arm vectors between different position/orientation sensors, and sensor boresight angles.

Ensuring the proper calibration of these error sources is essential in any sensor fusion application, but rigorous calibration of these errors often requires specialized equipment and calibration regimens. For example, IMU sensors often require precise turntables to calibrate their gyroscopes. Most sensor calibration is conducted offline, but is also cumbersome for end users of PosAtt systems. Offline sensor calibration can also be insufficient for MEMS sensors, that can have their sensor systematic errors vary significantly from turn-on to turn-on or even over the course of a single operating period. If a rigorous sensor calibration is required every time a PosAtt system is being operated, the reduced costs associated with using MEMS sensors can be quickly offset by increased operational costs.

There has been an ongoing trend in positioning technology towards the in-run, or online, calibration of IMU sensors in particular. Such procedures incorporate the estimation of IMU systematic errors into the position/attitude estimation process, often requiring general calibration maneuvers to be undertaken at the start of the operating period of the PosAtt system.

One of the biggest challenges in online IMU calibration is in the observability of the IMU systematic error parameters. Many applications are ill-suited to determining error parameters related to specific sensing axes. For instance, land vehicle motion is generally constrained to a locally planar surface, so systematic errors impacting the local vertical are generally unobservable due to the system's lack of vertical motion under regular operating conditions. Even aerial vehicles do not typically experience large dynamic changes to their system roll and pitch under normal operating conditions, which impacts the observability of sensor errors relating to these quantities.

Consequently, there has been a recent shift towards “observability-aware” systems that use observability measures such as the observability Gramian to guide the online calibration process as in [Lv et al, 2022; Yang et al, 2022]. While the observability Gramian provides a clear sense of whether or not the system is observable, it presents an all-or-nothing metric in that it cannot distinguish between an observable system and an observable but poorly-posed system [Krener and Ide, 2009]. Recent literature has shown a desire to shift away from this all-or-nothing perspective, with Lv et al [2022] notably using a truncated Singular Value Decomposition of a LS system to provide reliable calibration of observable states, even under degenerate conditions.

One essential aspect that is missing from IMU online calibration so far in the literature is a general analytical framework for evaluating the potential of an online calibration process over an arbitrary trajectory and even what specific maneuvers may deliver better observability for online calibration. Such a framework can provide significant assistance in the design and planning of real-time operations and data collection missions, i.e. in pre-analysis tasks for routine positioning and mapping applications. This chapter proposes an analytical framework based on the combined strengths of evaluating the following:

- The degree of observability of individual states or a group of states, after [Ham and Brown, 1983]. Pre-analysing the observability of specific states allows for the identification of which sensor systematic errors are essential to model, and also allows for an assessment of the impact that specific calibration maneuvers have on their estimation.

- The reliability in Kalman Filters, after [Wang, 1997; Caspary and Rüeger, 2000; Wang, 2009], which can provide insight into the minimum detectable values of specific systematic errors as outliers in the associated observations.
- The estimatability of augmented sensor systematic error states in Kalman Filters via Hypothesis testing, after [Wang, 1997, 2008, 2009; Caspary and Rüeger, 2000; Wang et al, 2021], which predicts how well the augmented states can be estimated.

These three avenues for analysis may be explored together to provide an intensive assessment of the potential performance of the augmented systematic error states being studied.

This chapter proposes a pre-analysis framework for the online calibration of sensor systematic errors within the GMIS. This framework specifically builds upon the principles of reliability analysis in Kalman filtering put forth by [Wang, 2009, 1997] and some of the reliability measures described in [Caspary and Rüeger, 2000], adapting them to the positioning Kalman Filter used in the GMIS. Additionally, this framework is specifically realized for an INS/GNSS integrated system and used to evaluate the relative value of several calibration maneuvers. The proposed framework is specifically realized under the GMIS, to allow for the direct modeling of IMU sensor systematic errors.

7.2 Analytical Framework for Online Sensor Calibration

In the first instance, it is important to clarify what information is necessary to conduct the pre-analysis for a kinematic positioning and orientation trajectory. This chapter uses the same system models developed and presented in Sections 5.3 and 5.5.1. IMU measurement models are the same as those presented in Sections 2.4.2.4.1 and 2.4.2.4.2, but the GNSS measurements are modeled as direct observations of the system's position and velocity.

Once the system models and observation models are defined, the *a priori* covariance matrices of the observation vector, process noise vector, and initial state vector may be propagated through a pre-defined trajectory without requiring any actual observations. The following information is necessary to conduct this error propagation:

- The system's three-dimensional positions, velocities, and accelerations.
- The system's attitude angles and their time derivatives.
- The anticipated covariance matrices for the observation and process noise vectors.
- The initial state vector's *a priori* covariance matrix.

From a design or pre-analysis perspective, it is valuable to predict how an online calibration procedure is expected to perform in Kalman Filtering. This chapter presents an analytical framework for online sensor calibration in a multisensor integrated kinematic positioning and navigation system, with a focus on the following aspects:

- i. Characterizing the time-varying observability of the augmented sensor systematic states to evaluate the feasibility of specific maneuvers and the merits of including specific systematic error states in the state vector.
- ii. Determining the minimum estimable values of the sensor systematic error states to confirm their minimum significant identifiable values in relation to the sensors' stochastic properties.
- iii. Optimizing the anticipated trajectory designs to produce the best performing candidate trajectories.
- iv. Performing analysis on the redundancy contributions of observations in a designed trajectory to determine the minimum detectable systematic error as outliers in the sensor observations.

7.2.1 Observability and Degree of Observability of States

The observability of system states is an essential measure of the quality of a linear system. When discussing observability, it is useful to define three “levels” of observability that a system may have, in increasing order of strictness [Rhee et al, 2004]:

- i. A system is considered completely observable over a specific time interval (t_0, t_k) if and only if the initial system state vector may be calculated solely from the observations collected over (t_0, t_k) .
- ii. A system is considered differentially observable over a specific time interval (t_0, t_k) if and only if it is completely observable over every sub-interval on (t_0, t_k) .
- iii. A system is considered instantaneously observable if and only if it is completely observable over the interval (t_k, t_{k+1}) .

It is typically simpler to determine whether a system is instantaneously observable than it is to determine whether a system is differentially or completely observable, and as a result many methods of assessing a system's observability start by determining if it is instantaneously observable. Once this information is available, a system may be concluded to be differentially observable over (t_0, t_k) if it is instantaneously observable for every observation epoch in the interval [Rhee et al, 2004], and a system may be concluded to be completely observable over (t_0, t_k) if it is also differentially observable over

that interval (although this is not a necessary condition for complete observability) [Rhee et al, 2004]. Additionally, a system is completely observable over (t_0, t_k) if it is instantaneously observable at any point over that interval [Rhee et al, 2004].

Although these three quantities are essential properties in characterizing how a linear system may be expected to perform, they also have inherent limitations. Each of these three quantities fall into all-or-nothing criteria that can only express whether the overall system is observable, and cannot characterize which states are the most/least observable among the observable states. Generally, the observation quality does not factor into characterizing system observability, so it is possible to have a system that is completely observable due to the geometry of the observations, but is practically unobservable due to the observations being of too low a quality to reliably estimate the system states. Given these limitations, it is valuable to characterize the “degree of observability” for individual states in a Kalman Filter. There are many ways to accomplish this, and some of these methods even take observation accuracy into account as part of their criteria. This subsection specifically describes three measure of observability that are used in this research: the observability index, which does not identify the degree of observability of a system, and the observability Gramian and eigenvalues of the state covariance matrix, which do.

7.2.1.1 The Observability Matrix

The instantaneous local observability index is determined by evaluating the observability matrix [Rhee et al, 2004]

$$\mathcal{O}(t) = \begin{bmatrix} N_0 \\ N_1 \\ \vdots \\ N_{n-1} \end{bmatrix} \quad (7.1)$$

wherein $N_0 = C$, and $N_i = N_{i-1}A + \frac{d}{dt}N_{i-1}$ and n is the dimension of the state vector. The observability matrix has $(n - 1)m$ rows, where m is the number of observations at the current epoch, and n columns.

If the rank of the observability matrix is equal to n , the system is instantaneously observable. If its rank is always equal to n over (t_0, t_k) , then the system is both differentially observable and completely observable over that interval.

7.2.1.2 The Observability Gramian

A system's observability can be characterized by estimating the impact that a true error in a given state has on the overall observation vector [Krener and Ide, 2009]. Generally, if an error in a state has little to no impact on the observation vector, that state may be considered to be unobservable. If a state x_i has an associated true error of ϵ , the true error in the overall state vector is given by $\mathbf{dx}^{\pm i} = \pm \epsilon \mathbf{e}_i$, where $\mathbf{e}_i$ is the i th basis vector. The resultant impact on the observation vector is determined as follows

$$\mathbf{z}^{\pm i} = \mathbf{C} \mathbf{dx}^{\pm i} \quad (7.2)$$

(7.2) may be used to populate each element of the empirical local observability Gramian, which is a matrix $\mathbf{P}$ such that

$$\mathbf{P}_{ij} = \frac{1}{4\epsilon^2} \int_{t_0}^{t_k} (\mathbf{z}^{+i} - \mathbf{z}^{-i})^T (\mathbf{z}^{+j} - \mathbf{z}^{-j}) dt \quad (7.3)$$

The observability Gramian is generally evaluated over an interval, but may technically be estimated at a specific observation epoch. The longer the interval that is used to estimate the observability Gramian, the better information it provides regarding the overall observability of the system. In this chapter, the observability Gramian is always evaluated over the entire candidate trajectory.

The observability Gramian may also be used to estimate several measures of the system's overall observability: the reciprocal of the smallest eigenvalue of $\mathbf{P}$ is called the local unobservability index, and larger values of this index indicate a less observable system; and the ratio of the square root of the largest to the square root of the smallest eigenvalue of $\mathbf{P}$ is called the local estimation condition number, with larger values suggesting poorly-posed systems.

7.2.1.3 Observability after Eigenvalues of the State Covariance Matrix

The eigenvalues and eigenvectors of the state covariance matrix may be used to characterize the degree of observability of linear combinations of the system states [Ham and Brown, 1983]. Generally, the largest eigenvalue corresponds with the linear combination of states that has the lowest observability, and the smallest eigenvalue corresponds with the linear combination that has the highest observability. Conducting an observability analysis using the state covariance matrix allows for the

impact of both the observation geometry and observation accuracies to be accounted for and characterized.

Usually, the significance of the eigenvalues/eigenvectors of a covariance matrix is not easily interpreted on its own. To mitigate this, the covariance matrix will be normalized prior to its eigen analysis. The normalization adopted in this work makes use of the diagonal matrix $\mathbf{D}_x(0)$ that is composed of the variances of the initial state vector, and is calculated as follows [Ham and Brown, 1983]

$$\mathbf{D}'_x(k) = \left(\sqrt{\mathbf{D}_x(0)}\right)^{-1} \mathbf{D}_x(k) \left(\sqrt{\mathbf{D}_x(0)}\right)^{-1} \quad (7.4)$$

where $\sqrt{\mathbf{M}}$ is the square root of the diagonal matrix, $\mathbf{D}_x(k)$ is the state covariance matrix at epoch k , and $\mathbf{D}'_x(k)$ is the normalized state covariance matrix at epoch k . The normalized covariance matrix in (7.4) is further scaled using its own trace as follows

$$\mathbf{D}_x^N(k) = \frac{n}{\text{tr}(\mathbf{D}'_x(k))} \mathbf{D}'_x(k) \quad (7.5)$$

which ensures that all eigenvalues lie between 0 and n , which makes them easier to interpret.

7.2.1.4 Practical Considerations in Observability Analysis

Eigenvalue-based observability analysis for both the observability Gramian and the state covariance matrix is difficult to plot and interpret, since both eigenvalues and eigenvectors are liable to change through time. Practically, one may make some concessions on these observability measures to more clearly express how the observability of specific states evolves through time.

Neglecting the correlation between states (i.e. ignoring the off-diagonal elements of the observability Gramian/state covariance matrix) produces corresponding eigenvectors that are simply a set of n -dimensional basis vectors, with the corresponding eigenvectors being the diagonal elements of the matrix being analyzed.

For the local observability Gramian, neglecting the correlation between states in (7.3) results in the following integral

$$\mathbf{P}_{ii} = \frac{1}{4\epsilon^2} \int_{t_0}^{t_k} (\mathbf{z}^{+i} - \mathbf{z}^{-i})^T (\mathbf{z}^{+i} - \mathbf{z}^{-i}) dt \quad (7.6)$$

to approximate the observability of the i th state $\forall i \in (1, 2, \dots, n)$. They can also potentially be used to approximate the local estimation condition number and unobservability index, with an associated loss of accuracy.

For the state covariance matrix, neglecting the correlation between states in (7.5) results in the normalized element

$$(\mathbf{D}_x^N)_{ii} = \frac{n}{\text{tr}(\mathbf{D}_x')} (\mathbf{D}_x')_{ii} \quad (7.7)$$

to approximate the unobservability of state element i .

7.2.2 Minimum Estimable Values of Systematic Error States

Modeling systematic errors in measurements is only meaningful if the corresponding augmented states are significantly estimable in Kalman Filtering during the online calibration process. Practically, their minimum estimable values (i.e. their significance from zero) over a specific portion of or an entire kinematic trajectory depend on their standard errors. They or their effects may count for little in Kalman filtering if their compensations for the corresponding errors are statistically too coarse as compared with the standard errors associated with their corresponding measurements.

7.2.2.1 Hypothesis Testing of Individual Augmented Systematic Error States

The epoch-wise covariance matrix for the system states estimates the accuracy of the state vector. Specifically, the square root of each diagonal element in $\mathbf{P}(k + 1)$ is the anticipated standard deviation of its corresponding system state [Caspary and Rüeger, 2000]

$$\sigma_{x_i} = \sqrt{\mathbf{P}_{ii}(k + 1)}, i \in (1, 2, \dots, n) \quad (7.8)$$

When evaluating the viability of a systematic error model or of a trajectory for online sensor errorcalibration, the estimate $\hat{x}_i(k + 1)$ of a systematic error state $x_i(k + 1)$ needs to be statistically significantly different from zero, which is formulated via the following null hypothesis H_0 and its alternative hypothesis H_a :

$$H_0: \tilde{x}_i = 0, H_a: \tilde{x}_i \neq 0, i \in (1, 2, \dots, n) \quad (7.9)$$

Under the assumption that the measurements and process noises conform to normal distributions, the test statistic for $x_i(k + 1)$ is

$$z_i = \frac{\hat{x}_i}{\sigma_{x_i}} \sim N(0, 1) \quad (7.10)$$

When using measurements, the test statistic is compared to a critical value z_{crit} after the standard normal distribution at the specified test significance. If it satisfies

$$\frac{|\hat{x}_i|}{\sigma_{x_i}} \geq z_{crit} \quad (7.11)$$

the null hypothesis is statistically rejected and the alternative hypothesis is accepted at the specified test significance level.

In system design without using measurements, (7.11) is used to determine a minimum estimable value for $\hat{x}_i$ as follows

$$|\hat{x}_i| \geq \sigma_{x_i} z_{crit} \quad (7.12)$$

wherein z_{crit} is a critical value determined by using the specified test significance level α (Type I Error) and power $1 - \beta$ (Type II Error) [Caspary and Rüeger, 2000]. As long as the minimum estimable value of $\hat{x}_i$ satisfies the design specification, it should be included in the state vector.

7.2.2.2 Hypothesis Testing of a Group of Augmented Systematic Error States

Generally, the estimate of the augmented system states for modeling the sensor systematic errors are correlated with one another in Kalman Filtering. A hypothesis test for a group of the correlated states can be introduced analogously to the deformation analysis in surveying [Caspary, 2000]. Usually, as a part of the statistical tests, hypothesis testing of a group of the correlated states is conducted first (as global or regional tests), and then hypothesis testing of each individual state of interest follows (as local tests). Generally, complications arise when coordinating the significance levels for tests on a group of states, as opposed to individual states, to ensure that results remain consistent [Caspary, 2000].

Straightforwardly, a multidimensional linear hypothesis testing of a group of correlated systematic error states of interest, $\mathbf{x}_g(k+1)$ is expressed through a null hypothesis and its alternative as follows

$$H_0: \mathbf{H}(k+1)\tilde{\mathbf{x}}_g = \mathbf{0}, H_a: \mathbf{H}(k+1)\tilde{\mathbf{x}}_g \neq \mathbf{0} \quad (7.13)$$

Assuming the multistates that are being tested follow a multivariate normal distribution with a zero mean vector and its covariance matrix, the test statistic corresponding to (7.13) is given by

$$\chi_g^2 = \hat{\mathbf{x}}_i^T \mathbf{H}^T(k+1) \left(\mathbf{H}(k+1) \mathbf{P}(k+1) \mathbf{H}^T(k+1) \right)^{-1} \mathbf{H}(k+1) \hat{\mathbf{x}}_g \cdot h \quad (7.14)$$

which conforms to a chi-squared distribution with h degrees of freedom. If data were available, this test statistic could be evaluated and compared to the critical value of the chi-squared distribution for a specified test significance/power. To accept the null hypothesis, the following condition would need to be satisfied:

$$\hat{\mathbf{x}}_g^T \mathbf{H}^T(k+1) \left(\mathbf{H}(k+1) \mathbf{P}(k+1) \mathbf{H}^T(k+1) \right)^{-1} \mathbf{H}(k+1) \hat{\mathbf{x}}_g \cdot h \leq \chi_{crit}^2 \quad (7.15)$$

Since this work is focused on pre-analysis, the special case where $\hat{\mathbf{x}}_g$ is an eigenvector of $\mathbf{H}^T(k+1) \left(\mathbf{H}(k+1) \mathbf{P}(k+1) \mathbf{H}^T(k+1) \right)^{-1} \mathbf{H}(k+1)$ is considered. Under this circumstance,

$$\hat{\mathbf{x}}_g^T \mathbf{H}^T(k+1) \left(\mathbf{H}(k+1) \mathbf{P}(k+1) \mathbf{H}^T(k+1) \right)^{-1} \mathbf{H}(k+1) \hat{\mathbf{x}}_g = \hat{\mathbf{x}}_g^T \hat{\mathbf{x}}_g \lambda = \lambda |\hat{\mathbf{x}}_g|^2 \quad (7.16)$$

Depending upon the choice of eigenvalues, the quadratic form in (7.16) either represents a best-case or worst-case scenario for the statistical test. Rearranging (7.16) to isolate $|\hat{\mathbf{x}}_g|$ yields

$$|\hat{\mathbf{x}}_g| \leq \sqrt{\frac{\chi_{crit}^2}{h\lambda}} \quad (7.17)$$

Here, the largest value of λ provides the strictest statistical test.

When the null hypothesis is instead framed as

$$H_0: \mathbf{H}\tilde{\mathbf{x}}_i \neq \mathbf{0}, H_a: \mathbf{H}\tilde{\mathbf{x}}_i = \mathbf{0} \quad (7.18)$$

the condition in (7.17) instead becomes

$$|\hat{\mathbf{x}}_g| \geq \sqrt{\frac{\chi_{crit}^2}{h\lambda}} \quad (7.19)$$

and the smallest value of λ provides the strictest statistical test.

Specifically, a hypothesis test for the significance of a group of augmented system states is a special case of (7.13) where $\mathbf{H}(k+1) = \mathbf{I}$, leading to the null hypothesis and associated alternative hypothesis of

$$H_0: \tilde{\mathbf{x}}_i \neq \mathbf{0}, H_a: \tilde{\mathbf{x}}_i = \mathbf{0} \quad (7.20)$$

and the same condition as (7.19) to define the minimum estimable length of the system state subvector. Here, the value of λ is given by the smallest eigenvalue of $\mathbf{P}^{-1}$.

7.2.3 Directly Comparing Covariance Matrices

For improving the performance of online sensor calibration in the design phase, it is useful to be able to directly compare covariance matrices estimated over different trajectories to determine what provides the “most accurate” results. The simplest method of accomplishing this is by directly comparing the anticipated standard deviations for each system state element between each candidate solution, but this approach neglects the correlation between system state elements.

A covariance matrix $\mathbf{D}_a$ may be considered to be “better” than another covariance matrix $\mathbf{D}_b$ of a random vector when their quadratic forms satisfy [Caspary, 2000]

$$\mathbf{f}^T \mathbf{D}_a \mathbf{f} \leq \mathbf{f}^T \mathbf{D}_b \mathbf{f} \quad \forall \mathbf{f} \in \mathbb{R}^n \quad (7.21)$$

or alternatively,

$$\mathbf{f}^T (\mathbf{D}_a - \mathbf{D}_b) \mathbf{f} \leq 0 \quad \forall \mathbf{f} \in \mathbb{R}^n \quad (7.22)$$

The condition in (7.22) is satisfied when $\mathbf{D}_a - \mathbf{D}_b$ is a negative definite or negative semi-definite matrix. So $\mathbf{D}_a$ may be considered “better” than $\mathbf{D}_b$ under this circumstance.

Of course, there are many circumstances under which $\mathbf{D}_a - \mathbf{D}_b$ will be an indefinite matrix. This corresponds to a situation where some portions of $\mathbf{D}_a$ outperform $\mathbf{D}_b$ and other portions underperform compared to $\mathbf{D}_b$. Under this situation, it is useful to define a measure that estimates the degree to which $\mathbf{D}_a$ is better than $\mathbf{D}_b$. This research proposes a measure based on the estimated eigenvalues of $\mathbf{D}_a - \mathbf{D}_b$:

$$c_{ab} = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^n |\lambda_i|} \quad (7.23)$$

The comparison measure in (7.23) is a real number that is bounded between -1 and 1. When all of the eigenvalues are positive, $c_{ab} = 1$ and $\mathbf{D}_a$ may be considered “better” than $\mathbf{D}_b$ for all states. When all of the eigenvalues are negative, $c_{ab} = -1$ and $\mathbf{D}_b$ may be considered to outperform $\mathbf{D}_a$ for all states. When c_{ab} is positive, $\mathbf{D}_a$ has some elements that perform “worse” than $\mathbf{D}_b$, but $\mathbf{D}_a$ may be considered “better” on the whole.

7.2.4 Minimum Detectable Values of Systematic Errors as Outliers after the Internal Reliability

On the grounds of the internal reliability in pre-analysis in Kalman Filtering [Wang, 2008, 1997], the redundancy indices are retrieved from the following idempotent matrix

$$\mathbf{R}_z = \mathbf{D}_{v_z} \mathbf{R}^{-1} \quad (7.24)$$

for the observation vector and

$$\mathbf{R}_w = \mathbf{D}_{v_w} \mathbf{Q}^{-1} \quad (7.25)$$

for the process noise vector, wherein $\mathbf{D}_{v_z}$ and $\mathbf{D}_{v_w}$ are the covariance matrices for the observation and process noise residuals, respectively.

The diagonal elements of the matrices in (7.24) and (7.25) provide the redundant indices for each of the independent observations and process noise factors, respectively, such that $r_{z_i} = (\mathbf{R}_z)_{ii}$ and $r_{w_i} = (\mathbf{R}_w)_{ii}$. r_{z_i} and r_{w_i} describe how much redundant information is held by the individual elements in the measurement vector and in the process noise vector. A redundancy value of 0 indicates no redundant information (equivalent to high leverage or full necessity), and a value of 1 indicates completely redundant information (either its standard error is too large or the element is geometrically irrelevant). Generally, the redundant indices should have intermediate values, or be evenly distributed as their total is equal to the total number of redundant measurements.

In the case that there exists a constant bias in a specific measurement series, the constant bias may be occasionally identified epoch-wise as an outlier when it is not augmented in the state vector through a shaping filter. An analysis of its internal reliability may be able epoch-wise to determine its minimum detectable value for a specific test significance/power. As long as the bias conforms to a random constant, it may later be estimated via a shaping filter. The objective of this section is to investigate how the average of such minimum detectable outlier values may relate to the minimum estimable systematic error states defined in Section 7.2.2 over a period of time, or even across the whole dataset.

With this pre-analysis framework, it is possible to assess the minimum detectable constant errors in a positioning sensor by evaluating the attainable precision in the associated residual vector. This is accomplished via the relationship between the true error vector $\boldsymbol{\varepsilon}$ and the associated residual vector $\mathbf{v}$

$$\mathbf{v} = -(\mathbf{D}_v \mathbf{D}_l^{-1}) \boldsymbol{\varepsilon} = -\mathbf{R}_l \boldsymbol{\varepsilon} \quad (7.26)$$

where $\mathbf{D}_l$ is the covariance matrix describing the *a priori* stochastic properties of the associated observation vector.

Considering only the effects on each residual element from varying its associated observation error allows (7.26) to be given as

$$v_i = -r_{l_i} \varepsilon_i, i \in (1, 2, \dots, m) \quad (7.27)$$

and propagating the error from ε_i to v_i yields

$$\sigma_{v_i} = \sqrt{r_{l_i}} \sigma_{l_i} \quad (7.28)$$

Assuming that a specific sensor systematic error is characterized as constant throughout its entire measurement series, it is possible to detect it successively as an outlier. The significance of the mean residual value over a time period of s observation epochs is tested using the following test statistic [Wang, 1997; Wang, 2009]:

$$\delta_i(\alpha, 1 - \beta) = \frac{\sum r_{l_j}}{\sigma_{l_i} \sqrt{s}} \varepsilon_i \quad (7.29)$$

under the specified significance level of α (Type I Error) and power of $1 - \beta$ (Type II Error), from which the minimum detectable error is determined as

$$\varepsilon_{i,min} = \delta_{i,crit}(\alpha, 1 - \beta) \frac{\sum \sqrt{r_{l_j}}}{\sigma_{l_i} \sqrt{s}} \quad (7.30)$$

The minimum detectable error in (7.30) is used to estimate the potential detectable value of the constant systematic errors in the observation vector. Practically, this means that the minimum detectable errors for the measurement series from a specific sensor over a period of time may be estimated.

Minimum detectable errors can also be estimated for the process noise vector using a similar process. This information can be used to identify the minimum resolvable sensor error drift over the test trajectory.

It is worth noting that the estimated minimum detectable error presents a measure of the system's potential precision, rather than of the system's potential accuracy. In other words, it estimates

the minimum consistent error in the observations/process noises that could produce a detectable output through the Kalman Filter.

7.2.5 Workflow of the Analytical Pre-Analysis Framework

The analytical pre-analysis framework presented in this section provides a valuable basis for evaluating the anticipated performance of an online calibration procedure, both with global and local performance measures. A workflow diagram for the framework is presented in Figure 7.2.1. The observability matrix and instantaneous observability Gramian are notably separate from the other quantities that are estimated as a part of the pre-analysis. This is because these two quantities are defined purely from the observation and system models, and do not rely on the output from the Kalman Filtering error propagation.

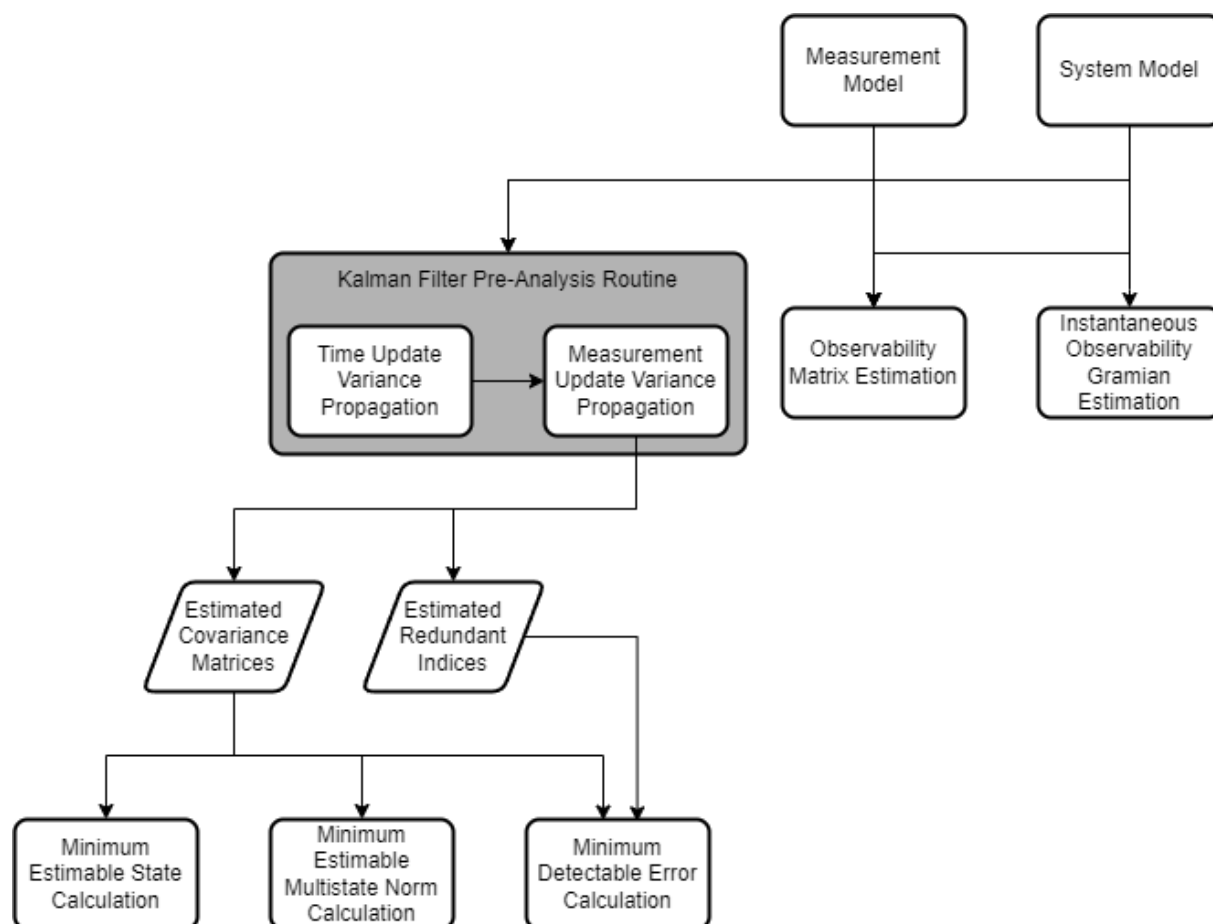


Figure 7.2.1: Workflow for estimating the specific quantities used in the proposed framework using the measurement models, system models, and output of the Kalman Filter pre-analysis routine.

7.3 Case Studies

The pre-analysis framework for online sensor calibration proposed in Section 3 allows for a comparative assessment of different candidate trajectories to evaluate how well they are suited for the online calibration of an IMU sensor and can quantitatively compare the calibration performance of different realizations of the same basic calibration maneuver. To practically apply this framework and illustrate its utility, three particular scenarios are considered in this research:

- i. Case #1: No intentional calibration maneuvers. Here, we pre-analyze a trajectory derived from a real dataset without any special intention of calibration maneuvers to demonstrate how the proposed framework is practically implemented.
- ii. Case #2: Circular motion. A common maneuver is considered here, in which a ground vehicle travels in circles for a certain period of time. The pre-analysis is focused on how different factors, such as the average vehicle speed and the radius of circular motion, impact the online calibration.
- iii. Case #3: Varying system roll and pitch. Another common maneuver is considered, in which the PosAtt system is held approximately stationary and is tilted so that its roll and pitch move through a specified range. Here, the focus is on how different factors, such as the maximum allowed roll and pitch and the rotational speed of the tilting, impact the online calibration. This maneuver is not practical for most ground vehicles, but is possible for UAV-mounted systems.

The three case studies are explored in more detail in the subsequent sections. Moreover, in all these scenarios, the same system technical specifications were applied for all constituent positioning/orientation sensors. Table 7.3.1 summarizes the anticipated accuracies of the applied measurements. All statistical tests in this section use a significance level of $\alpha = 0.01$ and a power level of $\beta = 0.80$ unless stated otherwise.

Table 7.3.1: The technical specifications of the sensor measurements used in the case studies.

Measurement	Standard Deviation	Unit
GNSS-Derived Position	± 2	cm
GNSS-Derived Velocity	± 3	cm/s

Accelerometer Specific Force	± 10	cm/s^2
Gyroscope Angular Rate	± 40	$^{\circ}/\text{s}$

7.3.1 Case #1: No Calibration Maneuvers

Figure 7.3.1 shows the top-down view of the trajectory that was adapted from a real road dataset, as well as the trajectory's associated velocity profile. Figure 7.3.2 shows the trajectory's associated acceleration and attitude profiles. All calculations in this section were performed using only information derived from the PosAtt system's trajectory, unless indicated otherwise.

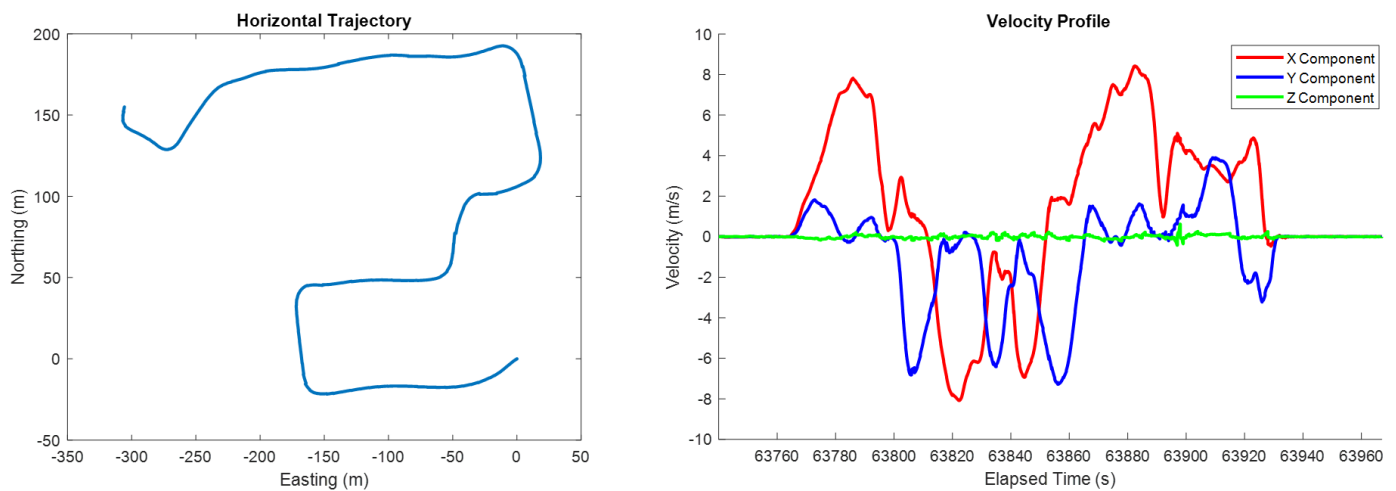


Figure 7.3.1: The top-down view of the trajectory used in Case #1 (left) and the velocity profile of the trajectory used in Case #1 (right). The components of the velocity are provided in the forward-right-down vehicle frame.

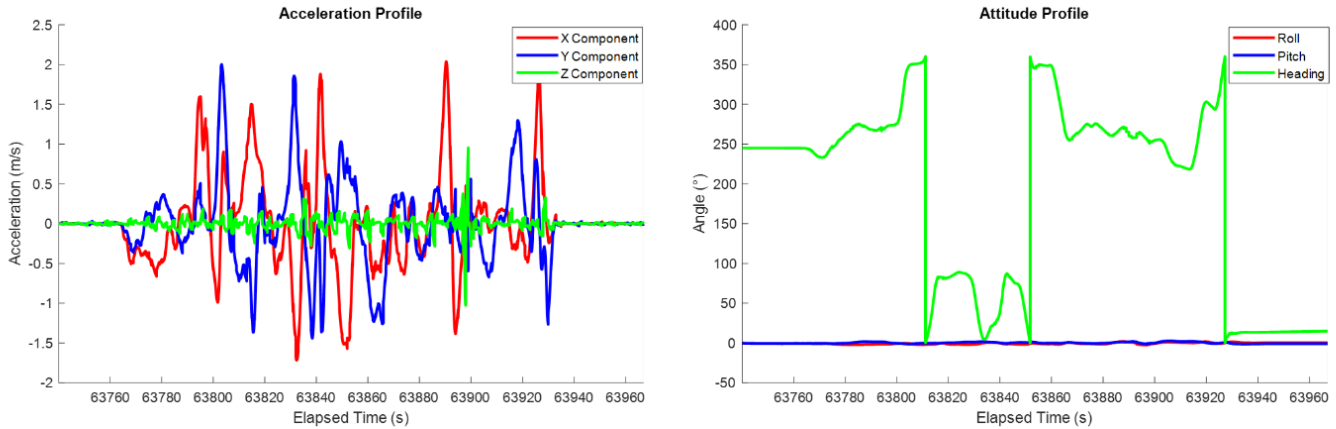


Figure 7.3.2: The acceleration profile of the trajectory used in Case #1 (left) and the attitude profile of the trajectory used in Case #1 (right). The components of the acceleration are provided in the forward-right-down vehicle frame.

7.3.1.1 Observability Analysis

One major challenge in online sensor calibration, particularly for MEMS IMUs, is the observability of their associated systematic error states. Most practical applications have some implicit constraints on the vehicle's motion; for example, ground vehicles generally do not naturally move vertically or significantly change their roll and pitch angles. As a result, most datasets have some sensor axes associated with practically unobservable systematic errors. While unobservable systematic error states may be included in the Kalman Filter without introducing numerical instability into the state estimation process, their inclusion as estimated states is unnecessary and increases the computational load of the online calibration process.

The test trajectory used in this section is defined for a car traveling on a relatively flat road and therefore has no significant vertical motion or roll/pitch dynamics, as can be observed in Figure 7.3.1 and Figure 7.3.2. The systematic errors relating to the vertical system dynamics or the dynamics in the roll and pitch are therefore intuitively expected to be unobservable in practice.

This section explores the more rigorous methods of determining the observability of the overall Kalman Filtering system and the individual systematic error states that are being modeled.

The rank of the observability matrix calculated via (7.1) for each observation epoch over the entire dataset was 22 for the 30 states that are being estimated. It may therefore be inferred that there are a total of eight unobservable states over this dataset, although there is no indication of which states are unobservable.

Estimating the observability Gramian over the entire dataset yields an unobservability index of approximately 133,172 and an estimation condition number of 2,972,916,621. The large magnitude of these numbers indicates that at least some systematic error states are completely unobservable over this dataset. The unobservability index and estimation condition number are valuable measures of overall observability, but they are also difficult to interpret since they are unbounded estimates and there is subjectivity in the definition of what constitutes a “large” issue with system observability.

To identify which states are unobservable, the eigenvalues and eigenvectors of the normalized state covariance matrix are estimated. Since eight states were determined to be unobservable from the rank of the observability matrix, the eight eigenvectors with the largest associated eigenvalues describe the systematic error states that are least observable. Table 7.3.2 presents an overview of the eight largest eigenvalues of the normalized state covariance matrix and their associated eigenvectors.

The eigenvectors with the largest associated eigenvalues indicate the quantities that are the least observable. These include the IMU lever arm X, Y, and Z components, the X and Y components of the gyroscope scale factor errors, and the X, Y, and Z components of the accelerometer scale factor error. The elements contributing the most to the lack of observability of the system are the lever arm X and Y components. Note that there is often repetition in the dominant components of the eigenvectors presented in Table 7.3.2. For example, the X and Y components of the gyroscope scale factor error are dominant in two of the eigenvectors. This indicates that there is generally poor observability of horizontal gyroscope scale factor errors, as opposed to this poor observability being limited to a single direction in the horizontal plane.

Table 7.3.2: Results of evaluating the eigenvalues and eigenvectors of the normalized state covariance matrix after (7.5).

Dominant Eigenvector Components	Associated Eigenvalue	Contribution to $\text{tr}(D_x^N)$ (%)
GPS-IMU Lever Arm – X and Y	29.4769	98.256
GPS-IMU Lever Arm – Z	0.2414	0.805
GPS-IMU Lever Arm – X and Y	0.1882	0.627
Gyroscope Scale – X and Y	0.0403	0.134

Gyroscope Scale – X and Y	0.0329	0.110
Accelerometer Scale – Z	0.0022	0.007
Accelerometer Scale – X	0.0021	0.007
Accelerometer Scale – Y	0.0020	0.007
TOTAL	29.986	99.953

The eigenvalues and eigenvectors in Table 7.3.2 were evaluated for a single state covariance matrix at the end of the trajectory. The eigenvector analysis provides clear indication of what states are lacking observability in the system, but the evolution of the dominant components of the eigenvectors through time is challenging to visualize and interpret.

To characterize how the observability of individual states evolves through time, the normalized covariance elements in (7.7) are used to approximate the observability index for individual state elements throughout the entire trajectory (Figure 7.3.3 to Figure 7.3.5). These plots confirm the intuitive conclusions that the lever arm X and Y components and the augmented sensor scale factor error states are generally less observable than the other systematic error states over the test trajectory. This generally agrees with the results presented in Table 7.3.2, although the more rigorous eigenvector analysis does additionally identify the Z component of the lever arm vector to be less observable.

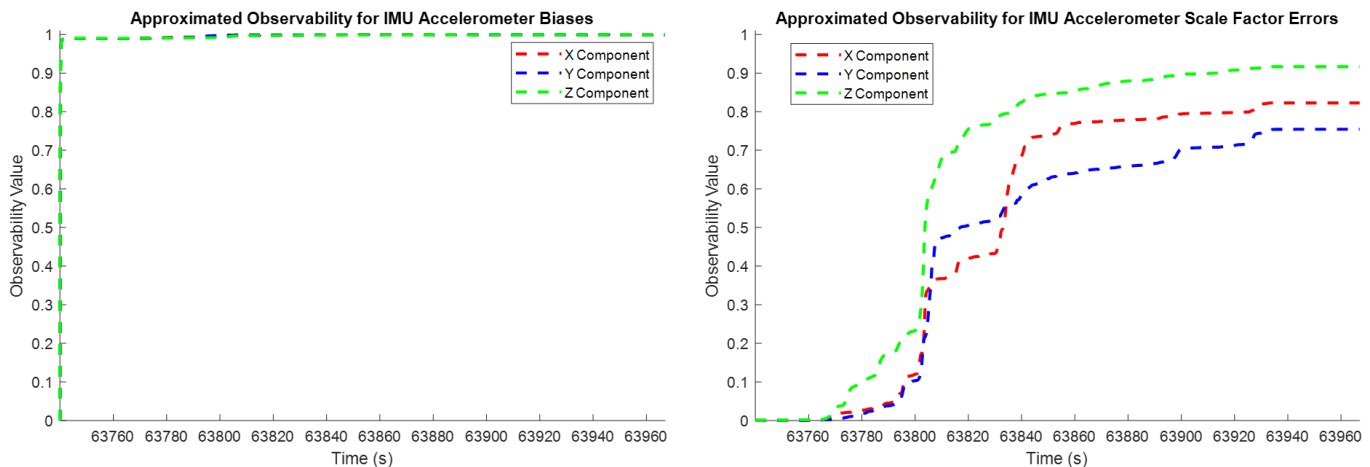


Figure 7.3.3: The observability of the systematic errors in the accelerometers (left: biases; right: scale factor errors).

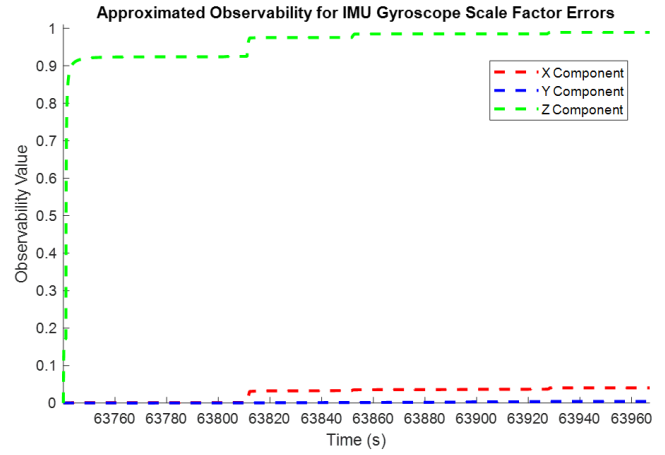
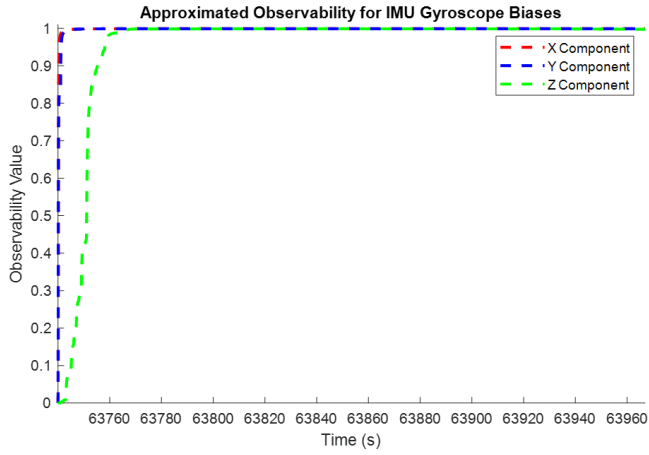


Figure 7.3.4: The observability of the systematic errors in the gyroscopes (left: biases; right: scale factor errors).

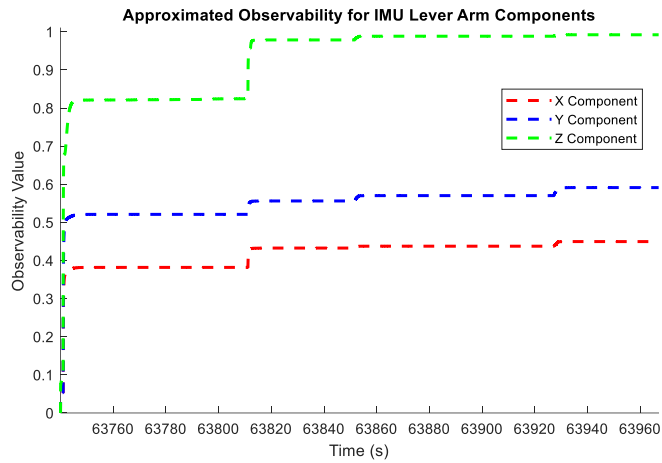


Figure 7.3.5: The observability of the lever arm components.

7.3.1.2 Minimum Estimable Systematic Error States

The minimum estimable values of the augmented systematic error states, based on (7.12) in Section 7.2.2 are plotted for the accelerometers in Figure 7.3.6, for the gyroscopes in Figure 7.3.7, and for the lever arm vector between the IMU sensor and the GNSS receiver in Figure 7.3.8.

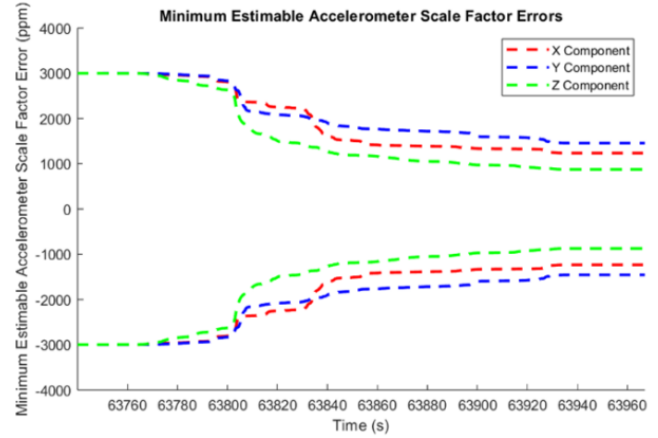
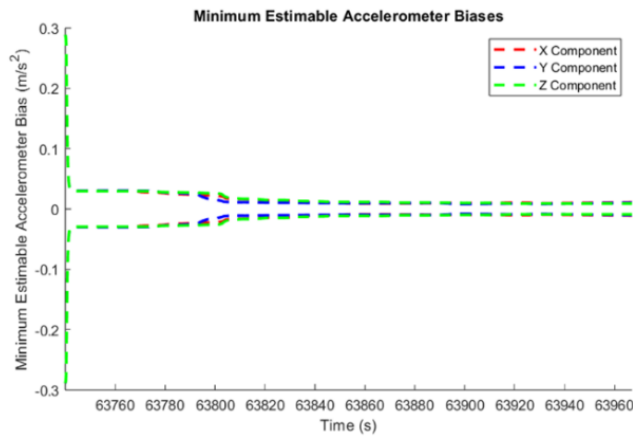


Figure 7.3.6: The minimum estimable systematic errors in the accelerometers (left: biases; right: scale factor errors).

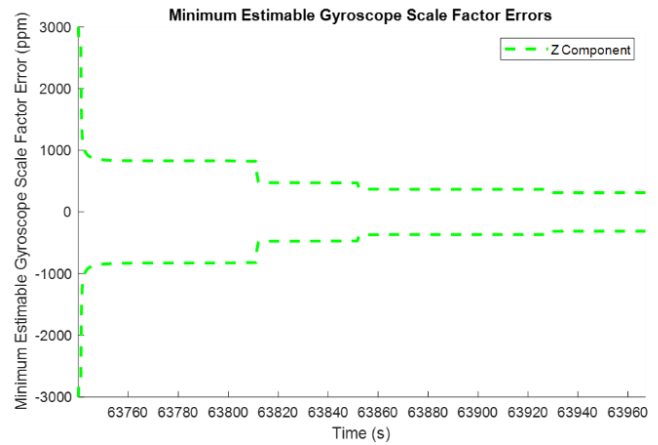
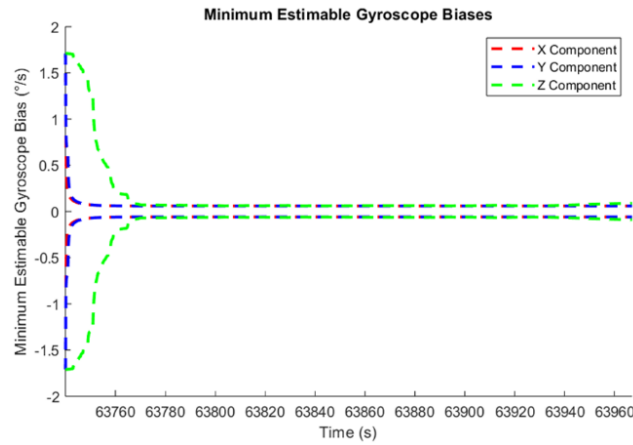


Figure 7.3.7: The minimum estimable systematic errors in the gyroscopes (left: biases; right: scale factor errors).

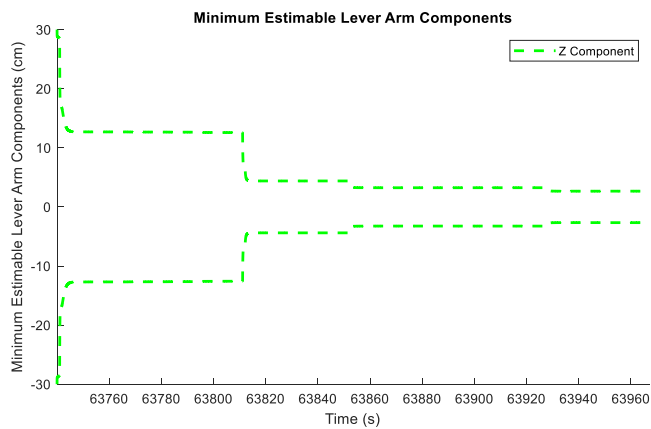


Figure 7.3.8: The minimum estimable lever arm components for the trajectory.

As can be seen in Figure 7.3.6 and Figure 7.3.7, the propagated standard errors of the IMU sensor bias states clearly have converged quickly, and remained stable until the end of the dataset, except for the bias associated with the Z-gyroscope, which took approximately 20 seconds longer to converge as the Z-gyroscope was vertically-aligned and could not be oriented while the vehicle was in stationary. However, the estimate of the Z-gyroscope bias converged quickly once the system began to move.

The trajectory in Case #1 did not experience high accelerations. The accelerometers therefore did not experience the high dynamics that would have produced larger associated scale factor errors. This means that, although Figure 7.3.6 exhibits an improvement in the estimated scale factor error states of the three accelerometers through time, this improvement is very limited over the entire trajectory.

Table 7.3.3 summarizes the minimum estimable values of the IMU systematic error states after (7.12). The minimum estimable values of both the accelerometer and gyroscope biases are significantly smaller than their associated observation standard errors (10 cm/s² and 40 °/s, respectively). Of course, the impact of scale factor errors on sensor observations depends upon the range of the sensor observations. For the accelerometers, the maximum linear acceleration the test trajectory reaches is ~2.3 m/s², which corresponds to an impact of 0.2 – 0.3 cm/s² on the sensor observations for the minimum estimable accelerometer scale factor errors. For the gyroscopes, the maximum heading rate-of-change the test trajectory reaches is ~21.3°/s, which corresponds to an impact of 0.4°/s on the sensor observations for the minimum estimable z-gyro scale factor error.

Table 7.3.3: Summary of the minimum estimable values of the IMU systematic error states (Case #1). Augmented states that were not estimated due to their unobservability are marked with “-”.

Systematic Error State	Minimum Estimable Systematic Error		
	X Component	Y Component	Z Component
Accelerometer Bias [cm/s ²]	1.1	1.0	0.9
Accelerometer Scale Factor Error [ppm]	1235	1455	875
Gyroscope Bias [°/s]	3.5	3.5	3.9

Gyroscope Scale Factor Error[ppm]	-	-	311
IMU-GPS Lever Arm [cm]	-	-	2.6

7.3.1.3 Minimum Detectable Values of Sensor Systematic Errors as Observation Outliers

The minimum detectable observation errors can vary significantly depending upon the time window used to evaluate them. The minimum detectable observation errors are estimated over a variety of time windows using equation (7.30) and presented in Figure 7.3.9. Notably, all three components of the accelerometer and gyroscope minimum detectable outliers have identical values for every time window used. These quantities all converge well as the time window increases in size, and their final converged values are provided in Table 7.3.4.

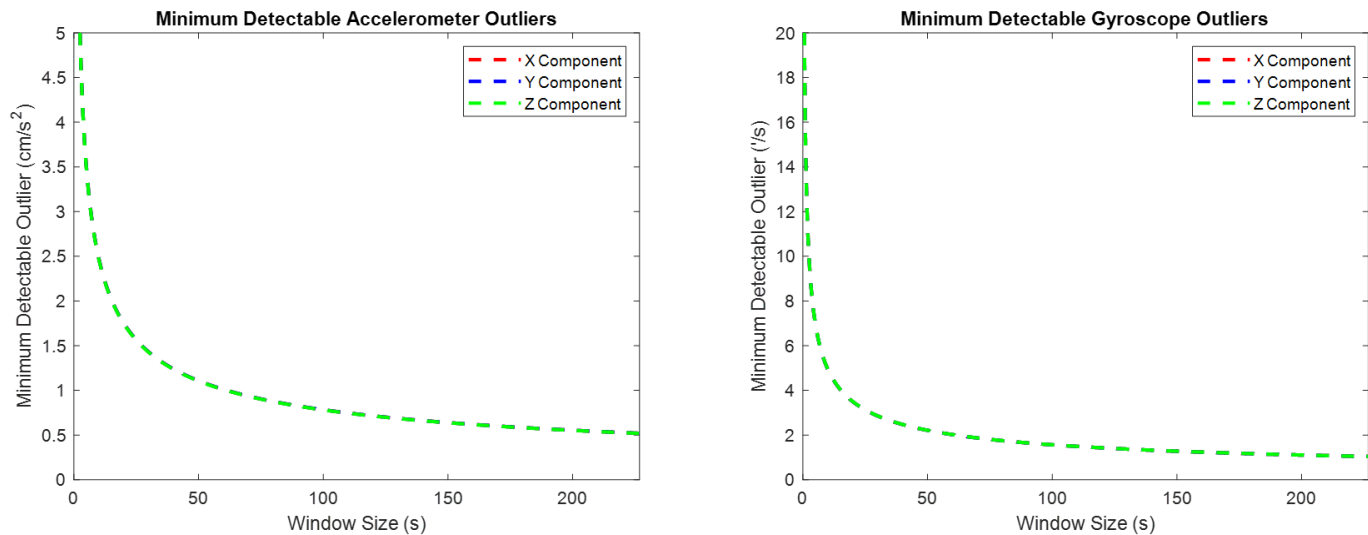


Figure 7.3.9: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, using a variety of window sizes, ranging from 0.4 seconds to the entire dataset.

Table 7.3.4: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.

Observation Type	Minimum Detectable Observation Outliers		
	X Component	Y Component	Z Component

Accelerometer [cm/s ²]	0.52	0.52	0.52
Gyroscope [°/s]	1.05	1.05	1.05

Comparing the results from Table 7.3.3 and Table 7.3.4, the minimum detectable observation outliers in the accelerometers/gyroscopes are significantly smaller than the associated minimum estimable sensor biases. This difference draws their relative roles in the pre-analysis framework into focus:

- i. The minimum estimable sensor systematic error states provide a prediction of the attainable accuracy of the system.
- ii. The minimum detectable observation errors provide a prediction of the system's precision via estimating the minimum consistent observation error that would produce a detectable impact on the system's output.

7.3.1.4 Comparison of Pre-Analysis Results with Road Test Data

Since the pre-analysis conducted in Sections 7.3.1.1 to 7.3.1.3 uses a trajectory derived from a real road test dataset, the estimated minimum estimable systematic error states may be compared to the *a posteriori* minimum estimable systematic error states. The *a posteriori* values are determined via scaling the assumed sensor accuracies by their associated time-varying variances of unit weight. The process of estimating these variances of unit weight is described in detail in Chapter 3, and Chapter 5 shows these estimated quantities for the road test dataset used in this chapter [Brunson and Wang, 2023].

The road test dataset was collected from a land vehicle equipped with York University's in-house integrated navigation system, consisting of an IMU 440 sensor operating at 100 Hz and a NovAtel OEM4 receiver operating at 1 Hz. A second NovAtel OEM4 receiver operating at 1 Hz was operated as a base station at a fixed location for the purposes of generating double-differenced GPS observations.

The plots of the minimum estimable systematic error states for the real dataset appear almost identical to the corresponding plots in Section 7.3.1.2. While this is encouraging in terms of the consistency between the pre-analysis results and the road test's results, providing these plots here would not provide meaningful insight outside of establishing this consistency. Instead, the ratio between the *a priori* and *a posteriori* minimum estimable systematic error states is provided to give insight into the actual performance of the system.

The comparison between the *a priori* and *a posteriori* minimum estimable accelerometer systematic error states are presented in Figure 7.3.10, the comparison between the *a priori* and *a posteriori* minimum estimable gyroscope systematic error states are presented in Figure 7.3.11, and the comparison between the *a priori* and *a posteriori* minimum estimable lever arm components are presented in Figure 7.3.12.

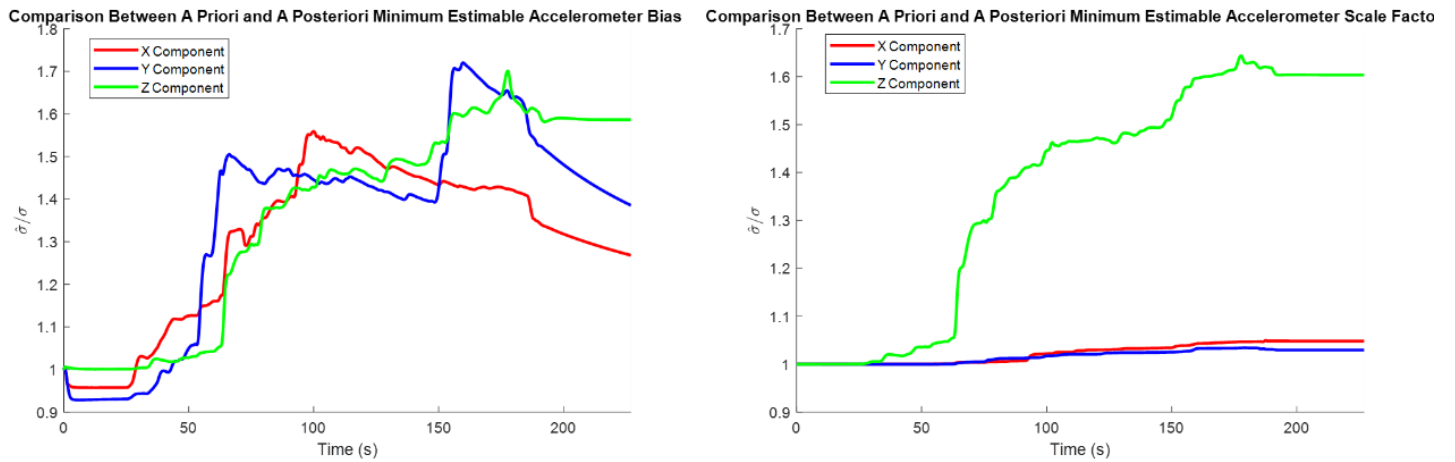


Figure 7.3.10: Ratio between the *a priori* and *a posteriori* standard deviations of the accelerometer bias estimates (left) and scale factor estimates (right).

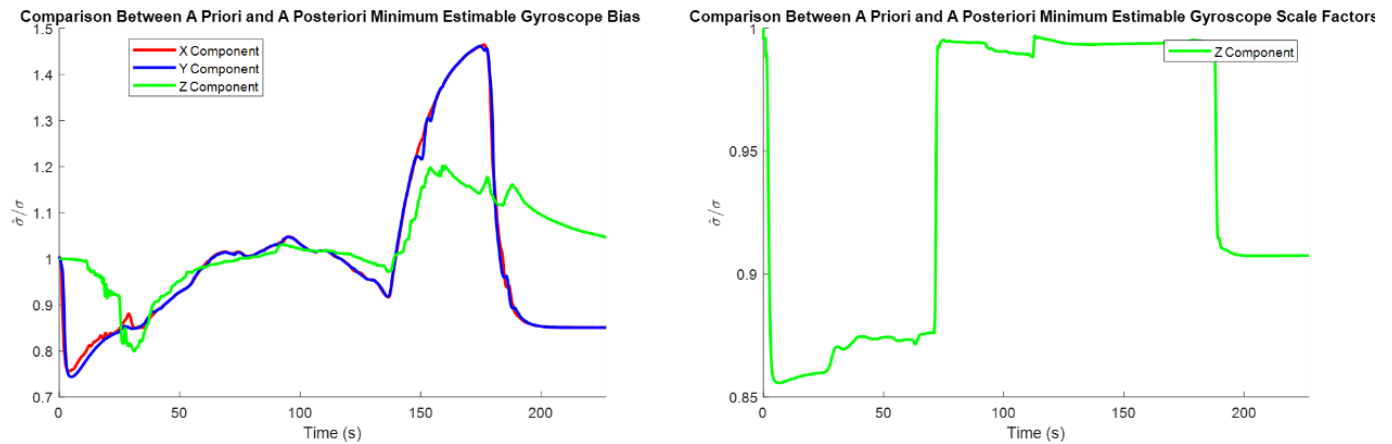


Figure 7.3.11: Ratio between the *a priori* and *a posteriori* standard deviations of the gyroscope bias estimates (left) and scale factor error estimates (right).

Comparison Between A Priori and A Posteriori Minimum Estimable Lever Arm Components

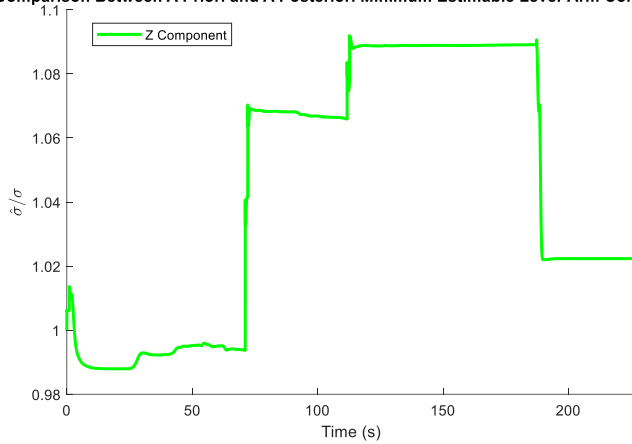


Figure 7.3.12: Ratio between the *a priori* and *a posteriori* standard deviations of the lever arm component estimates.

In the plots of the ratio of the *a posteriori* to *a priori* standard deviations, the calculated ratio is typically larger than one, meaning that the actual observation accuracy did not exceed the sensor specifications. Notably, the *a posteriori* minimum estimable Z-axis component of the gyroscope scale factor error outperformed the *a priori* standard deviation estimates consistently for the entire observation period.

The final values for the *a posteriori* minimum estimable systematic error states are provided in Table 7.3.5. These values may be directly compared to the corresponding values provided in Table 7.3.3.

Table 7.3.5: Final values of the *a posteriori* minimum estimable IMU systematic error estimates for the road test dataset. Augmented states that were not estimated due to their unobservability are marked with “-”.

Systematic Error State	Minimum Estimable Systematic Error		
	X Component	Y Component	Z Component
Accelerometer Bias [cm/s ²]	1.3	1.3	1.4
Accelerometer Scale [ppm]	1663	1963	1444
Gyroscope Bias [°/s]	1.4	1.4	4.7
Gyroscope Scale [ppm]	-	-	302

Lever Arm Vector [cm]	-	-	4.1
------------------------------	---	---	-----

7.3.2 Case #2: Simulated Trajectories with a Circular Motion Maneuver

Driving the vehicle carrying the PosAtt system in a circle is a common calibration maneuver to undertake prior to data collection procedures for an online calibrated system. With such a maneuver, there are three defining parameters that may impact the quality of the online calibration: the radius of the circular motion, the tangent velocity along the circle, and the number of times the circular motion is repeated. The number of repetitions of the maneuver is assumed to have similar impact on all radii/linear velocity combinations, so we first identify the speed and radius that yield the lowest minimum estimable systematic errors.

7.3.2.1 Identifying Optimum Parameters for the Circular Motion Maneuver

The minimum estimable accelerometer and gyroscope biases/scale factor errors at the end of the maneuver are shown from Figure 7.3.13 to Figure 7.3.14. The blank areas in the plots represent sets of parameters that would constitute a potentially unsafe maneuver, due to the chance of skidding in a ground vehicle.

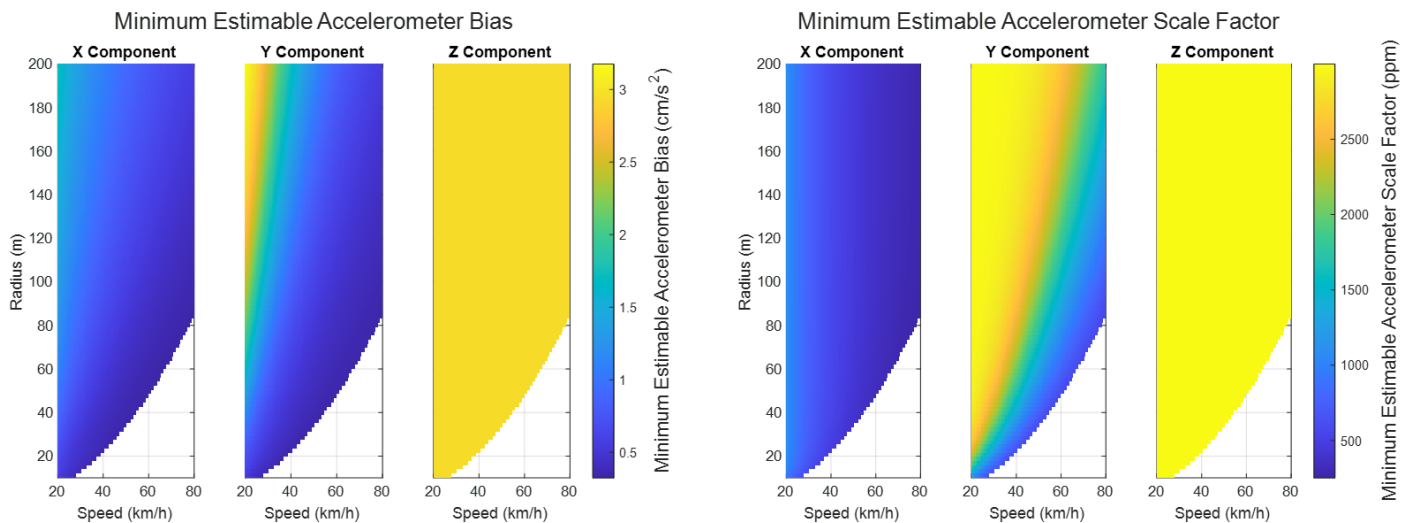


Figure 7.3.13: The minimum estimable values of the augmented accelerometer bias states (left) and scale factor error states (right) under circular motion (speeds ranging from 20 - 80 km/h and radii ranging from 10 - 200 m).

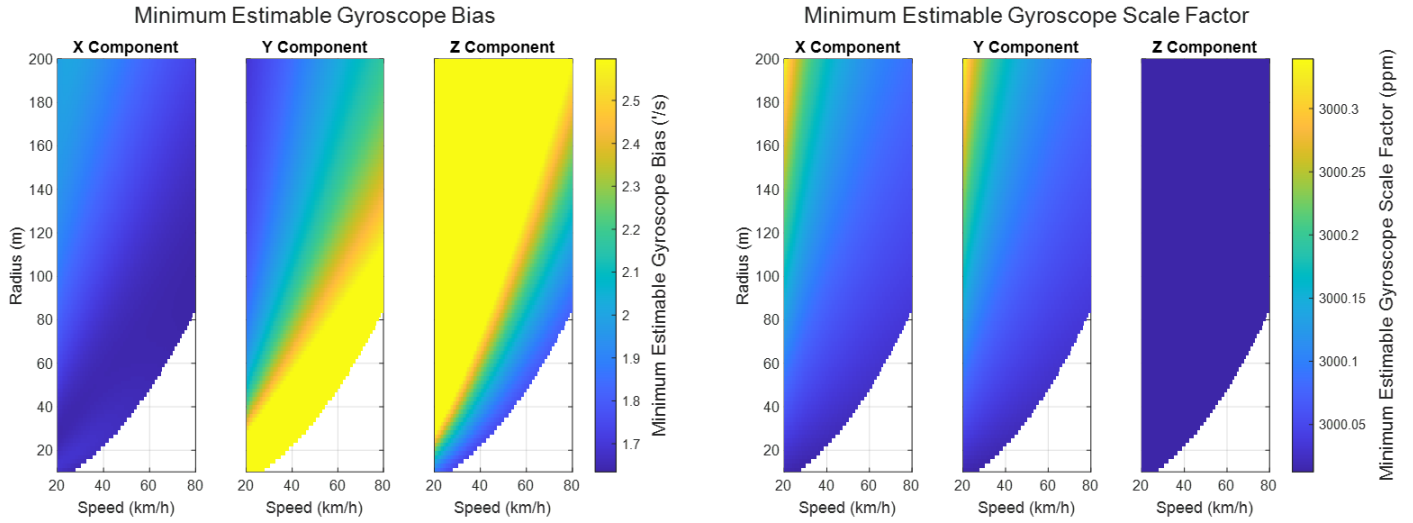


Figure 7.3.14: The minimum estimable values of the augmented gyroscope bias states (left) and scale factor error states (right) under circular motion (speeds ranging from 20 – 80 km/h and radii ranging from 10 – 200 m).

As shown in Figure 7.3.13 and Figure 7.3.14, the overall minimum estimable values for all of the augmented sensor bias states have reached a level better than 1σ of their associated sensor standard errors for both the accelerometers and gyroscopes.

The minimum estimable value of the augmented z-accelerator's systematic error states is not sensitive to the speed and radius of the circular motion as the other two components are. This result is intuitive, considering that changing the parameters of the horizontal motion the system experiences has no impact on the vertical acceleration of the system. Specifically, the biases and scale factor errors in the specific force measurements are better estimated when the circular motion is at a higher speed with a smaller radius. This effect is particularly pronounced for the accelerometer's Y-component (i.e. the axis pointing in the lateral direction), which can be easily interpreted according to $a_r = v_t^2/R$ (a_r , v_t , and R are the centripetal acceleration, tangent velocity, and radius, respectively).

Additionally, the minimum estimable gyroscope bias and scale factor errors are also generally less influenced by the parameters of the circular motion. This result is also intuitive, since the dynamics of the system attitude remains similar for different parameters of circular motion.

For the subsequent sections, a circular motion with a radius of 30 metres and a tangent speed of 40 km/h is considered. The observability of the specific augmented systematic error states for this trajectory is shown in Section 7.3.2.2, and the minimum estimable systematic error states are shown as the maneuver is repeated in Section 7.3.2.3.

7.3.2.2 Observability Analysis

For the circular motion trajectory with a radius of 30 metres and a tangent velocity of 40 km/h, the rank of the observability matrix is 24, indicating that there are a total of six unobservable states in the Kalman Filter. Estimating the observability Gramian over the entire dataset yields an unobservability index of approximately 274,881 and an estimation condition number of 6,802,290,162. The large magnitude of these numbers indicates that at least some of the systematic error states are unobservable over this dataset.

The 8 largest eigenvalues of the normalized state covariance matrix and the dominant components of their corresponding eigenvectors are provided in Table 7.3.6. Generally, the least observable quantities under circular motion are the 3 components of the GPS-IMU lever arm vector and all IMU scale factor errors except for the Z-component of the gyroscope, which changes significantly with the heading.

Table 7.3.6: Results of evaluating the eigenvalues and eigenvectors of the normalized state covariance matrix.

Dominant Components	Associated Eigenvalue	Contribution to $\text{tr}(D_x^N)$ (%)
GPS-IMU Lever Arm – X and Y	29.4769	98.256
GPS-IMU Lever Arm – Z	0.2414	0.805
GPS-IMU Lever Arm – X and Y	0.1882	0.627
Gyroscope Scale – X and Y	0.0403	0.134
Gyroscope Scale – X and Y	0.0329	0.110
Accelerometer Scale – Z	0.0022	0.007
Accelerometer Scale – X	0.0021	0.007
Accelerometer Scale – Y	0.0020	0.007
TOTAL	29.986	99.953

The diagonal elements of the normalized state covariance matrix are used to approximate the unobservability index for the individual state elements in Figure 7.3.15, Figure 7.3.16, and Figure 7.3.17. These plots may be used to successfully identify the accelerometer Z-component scale factor, the gyroscope X- and Y-component scale factors, and the lever arm components as unobservable.

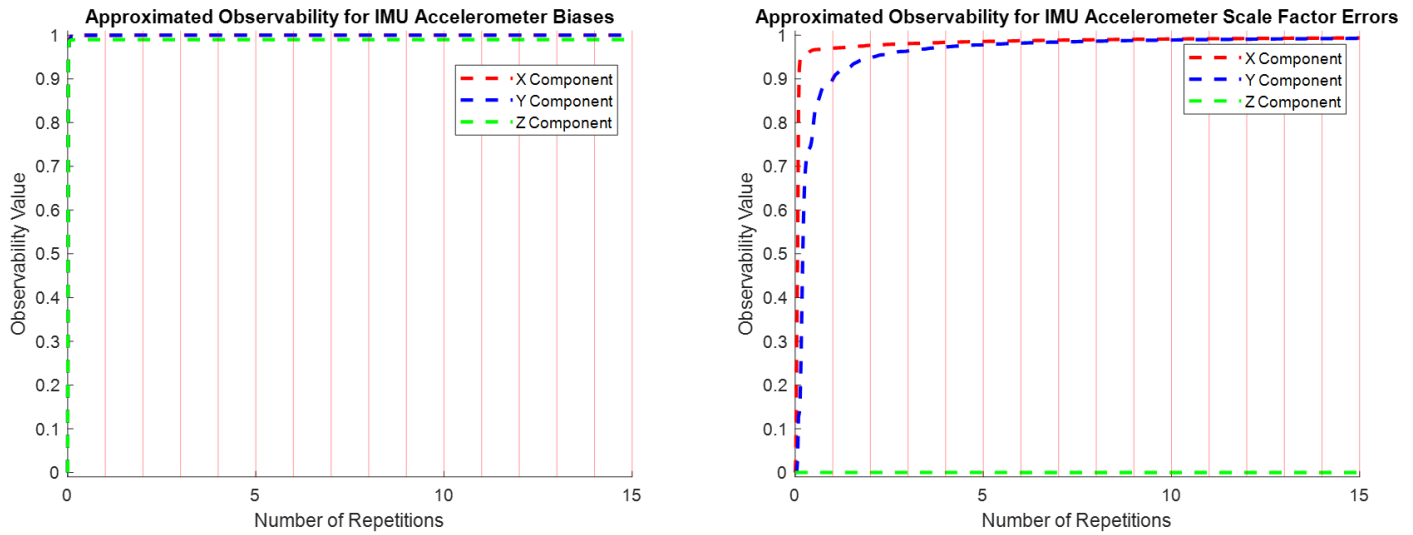


Figure 7.3.15: The observability of the systematic errors in the accelerometers (left: biases; right: scale factor errors).

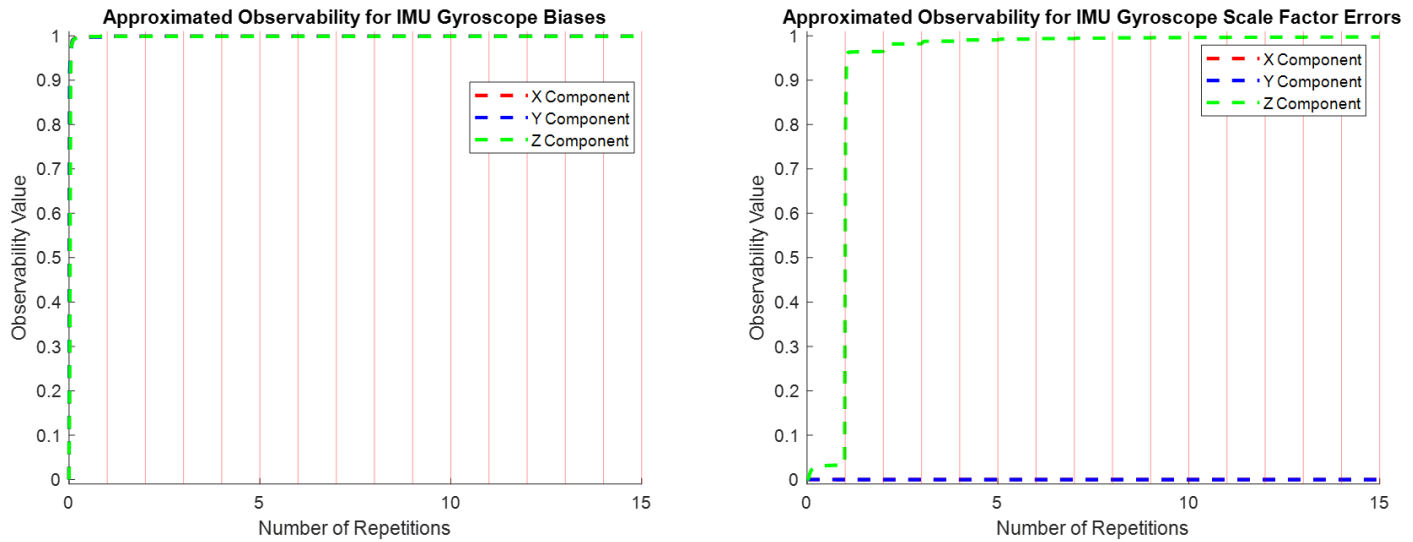


Figure 7.3.16: The observability of the systematic errors in the gyroscopes (left: biases; right: scale factor errors).

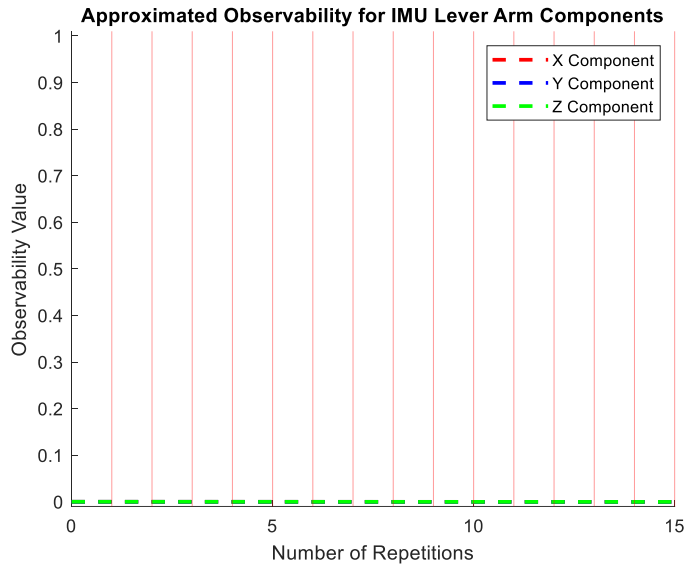


Figure 7.3.17: The observability of the lever arm components.

7.3.2.3 Minimum Estimable Systematic Errors

For the circular motion maneuver with a radius of 30 metres and a tangent velocity of 40 km/h, the number of full rotations taken by the vehicle is modified to determine the optimal amount of time to devote to this maneuver. The plots of each systematic error through time are shown in Figure 7.3.18 and Figure 7.3.19.

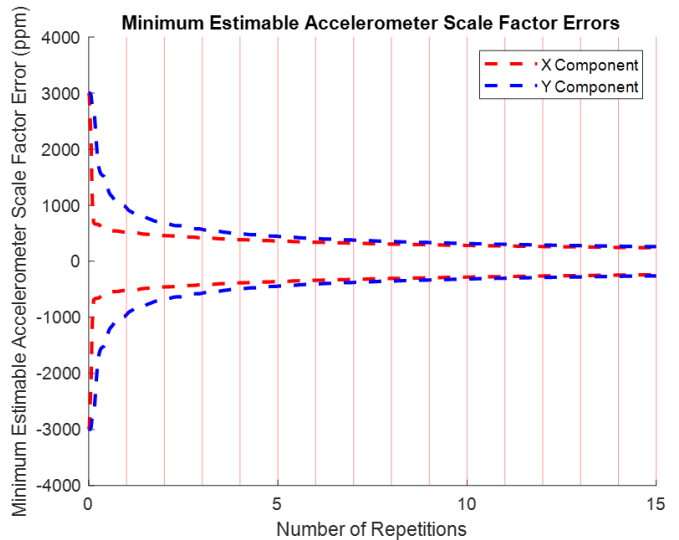
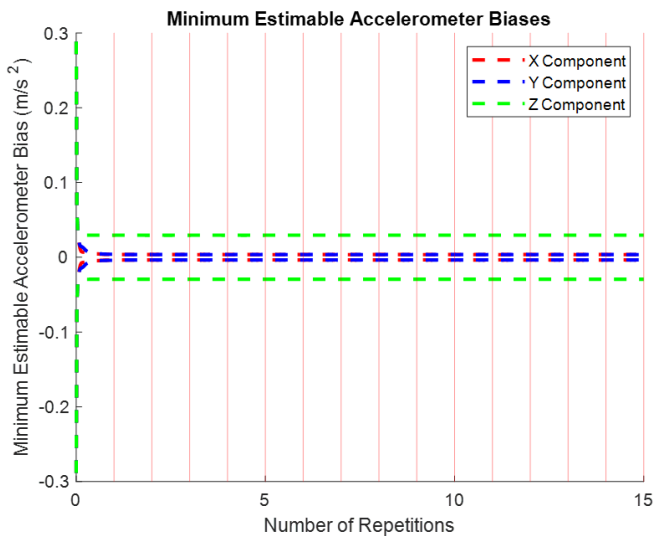


Figure 7.3.18: Minimum estimable values of the augmented accelerometer bias states (left) and scale factor states (right) for circular motion with a radius of 30 m and a tangent velocity of 40 km/h. Vertical red grid lines are aligned to the starting position of each circle.

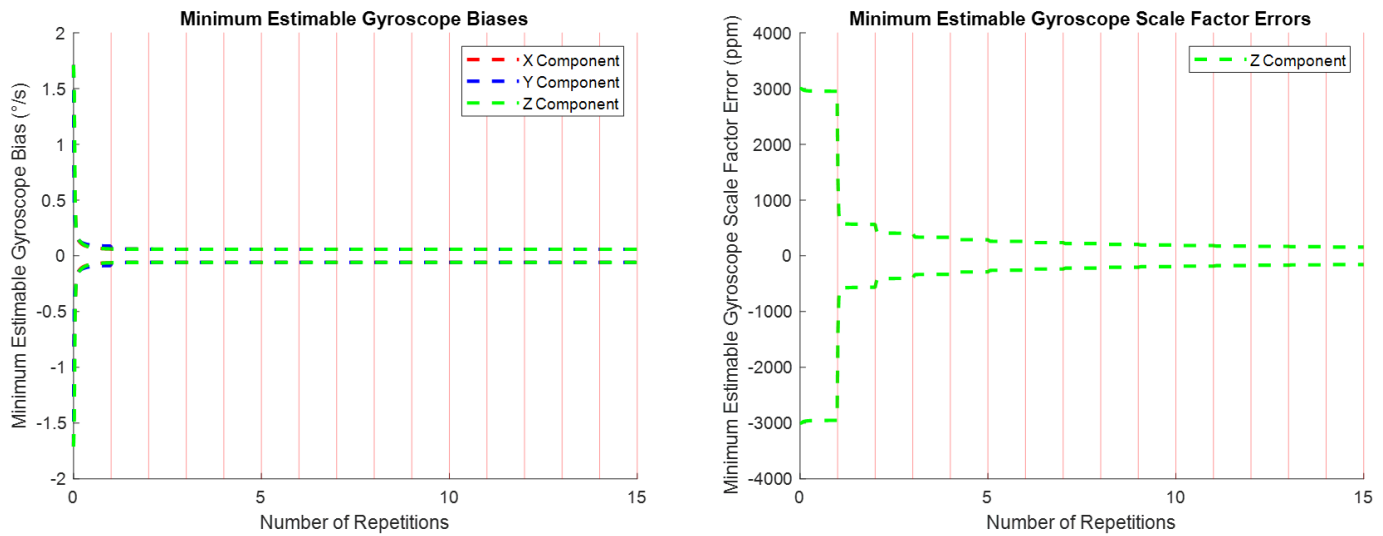


Figure 7.3.19: Minimum estimable values of the gyroscope bias states (left) and scale factor states (right) for circular motion with a radius of 30 m and a tangent velocity of 40 km/h. Vertical red grid lines are aligned to the starting position of each circle.

Clearly, both Figure 7.3.18 and Figure 7.3.19 demonstrate that the accelerometer and gyroscope biases converge very well even before the system experiences one full revolution. Subsequent revolutions yield vastly diminishing returns on information about the IMU biases.

In contrast, the minimum estimable scale factor error states exhibit a much more gradual improvement, taking multiple full revolutions to reach a point of diminishing returns on further maneuvers. It is worth noting that since the circular motion is horizontal, the vertical scale factor error state in the accelerometers and the X- and Y-axes' scale factor error states in the gyroscopes are not observable in this situation, and therefore their accuracies do not improve over time above their initial assumed accuracy.

The minimum estimable values of the augmented error states for the IMU are summarized in Table 7.3.7. Values are provided after 3 repetitions of the circular motion and 15 repetitions of the circular motion.

Table 7.3.7: Minimum estimable values of the augmented systematic error states under circular motion with a radius of 30 m and a tangent velocity of 40 km/h (3 repetitions of the circular motion and 15 repetitions). Augmented states that were not estimated due to their unobservability are marked with “-”.

Systematic Error State	Minimum Estimable Systematic Error					
	3 Repetitions			15 Repetitions		
	X	Y	Z	X	Y	Z
Accelerometer Bias [cm/s ²]	0.4	0.4	2.9	0.4	0.4	2.9
Accelerometer Scale [ppm]	411	544	-	233	250	-
Gyroscope Bias [°/s]	1.7	1.7	1.9	1.7	1.7	1.9
Gyroscope Scale [ppm]	-	-	88	-	-	42
Lever Arm [cm]	-	-	-	-	-	-

7.3.2.4 Directly Comparing Covariance Matrices

Comparing the relative quality of two candidate maneuvers by comparing their minimum estimable systematic error states can be inconclusive when each candidate maneuver produces some more reliable systematic error state estimates than the other. Conducting a pair-wise comparison between two candidate maneuvers can provide a clearer measure of which maneuver performs better on the whole.

To evaluate the impact vehicle speed has on the overall quality of IMU online calibration, a pair-wise comparison is conducted between different circular trajectories that have different vehicle speeds, but the same circular radius of 30 metres. The pair-wise comparison is accomplished via the

methodology provided in Section 7.2.3, and a plot of the normalized comparison measures is provided in Figure 7.3.20.

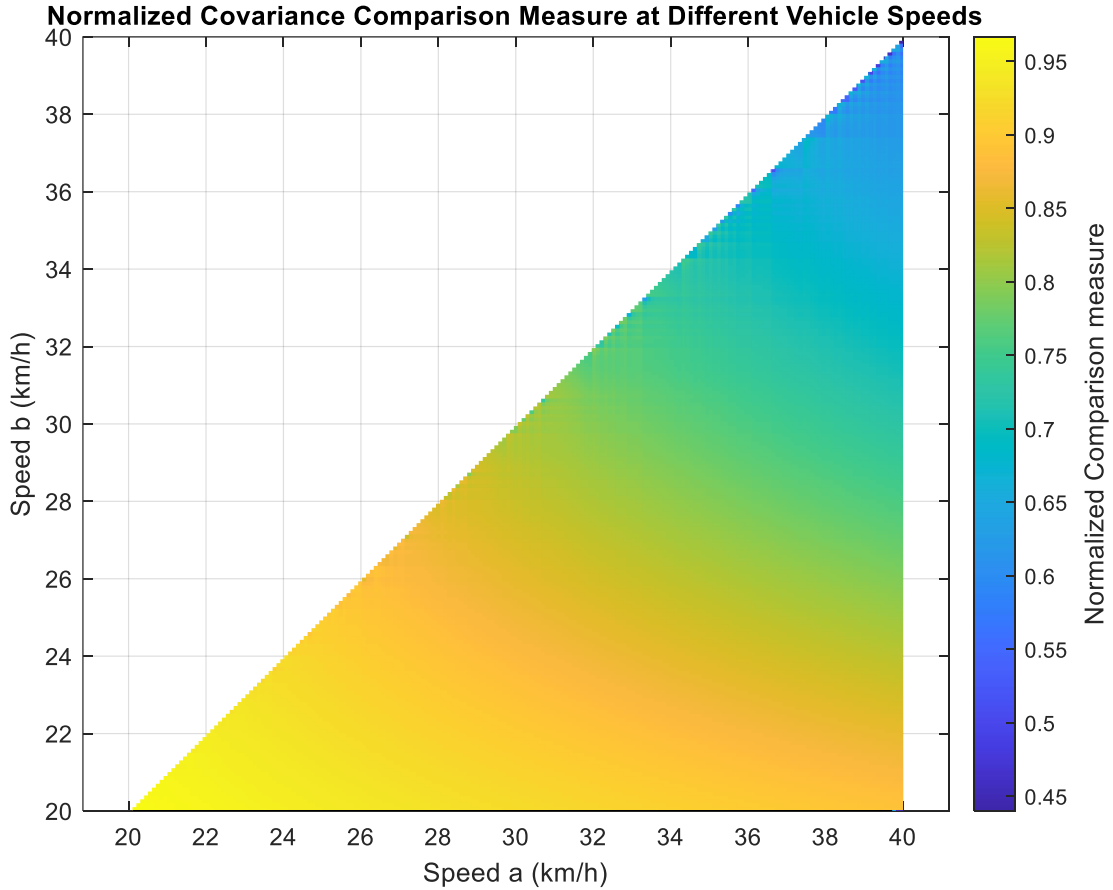


Figure 7.3.20: The estimated normalized covariance comparison measures for circular motion with a radius of 30 metres at different vehicle speeds. The covariance comparison measure is c_{ab} , with $\mathbf{D}_a$ determined using Speed a (X-axis) and $\mathbf{D}_b$ determined using Speed b (Y-axis).

The normalized comparison measure indicates that higher speeds generally correspond to improvements in the overall quality of the state covariance matrix, since the normalized comparison measure is consistently positive. This improvement does yield diminishing returns as the speed increases further, and this behaviour is particularly evident at speeds above 30 km/h.

7.3.2.5 Minimum Detectable Observation Errors

The minimum detectable observation errors over the entire test trajectory are provided in Table 7.3.8.

Table 7.3.8: The minimum detectable observation errors for the IMU accelerometers and gyroscopes, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.

Observation Type	Minimum Detectable Systematic Errors as Observation Outliers		
	X Component	Y Component	Z Component
Accelerometer [cm/s^2]	0.50	0.50	0.50
Gyroscope [$^\circ/\text{s}$]	1.00	1.00	1.00

7.3.3 Case #3: Simulated Trajectories with Varying System Roll and Pitch Maneuver

This maneuver consists of holding the kinematic system approximately stationary and rotating it through a series of planned tilt angles, varying both its roll and pitch with time. To optimize the contribution of this maneuver to the online calibration process, there are three defining parameters that may impact the online calibration quality: the maximum tilt angle the system reaches, the rotation rate of the system, and the repetitions of the attitude maneuver. Analogously to the circular motion maneuver, the number of maneuver repetitions is assumed to have a similar impact on all maximum tilt angle and rotation rate combinations. So, we first identify an optimal set of maximum tilt angles and rotation rates.

7.3.3.1 Identifying Optimum Parameters for the Attitude Maneuver

The minimum estimable values of the IMU biases and scale factor errors in the augmented state vector at the end of the maneuver are shown in Figure 7.3.21 to Figure 7.3.22. The minimum estimable values for the lever arm components and the minimum estimable value of the absolute lever arm length are shown in Figure 7.3.23.

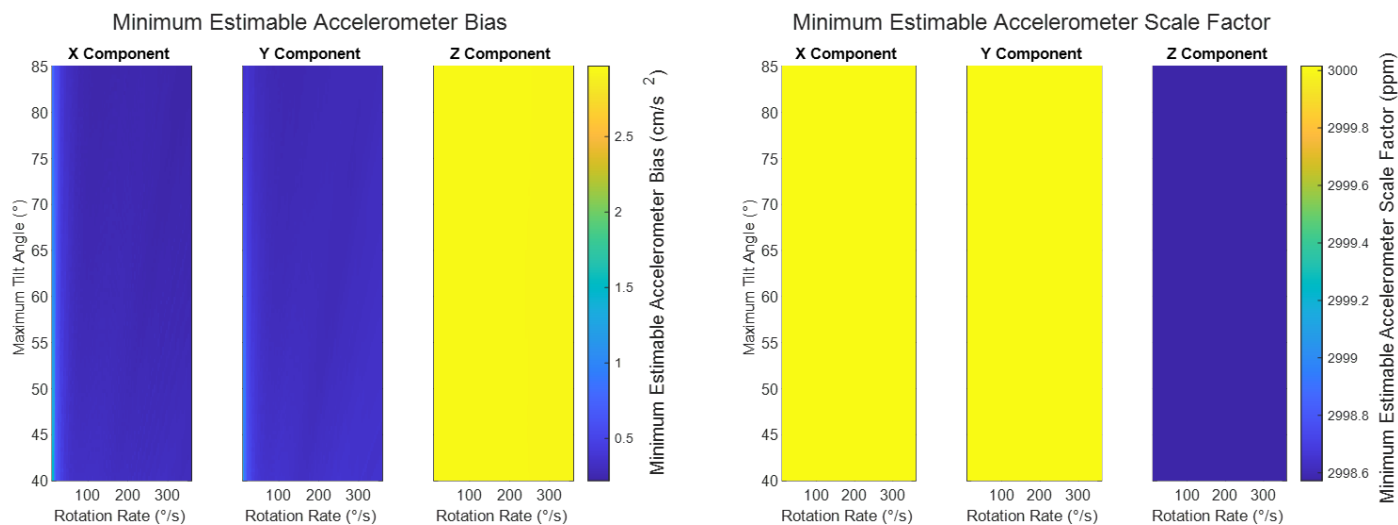


Figure 7.3.21: The minimum estimable values of the augmented accelerometer bias states (left) and scale factor error states (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).

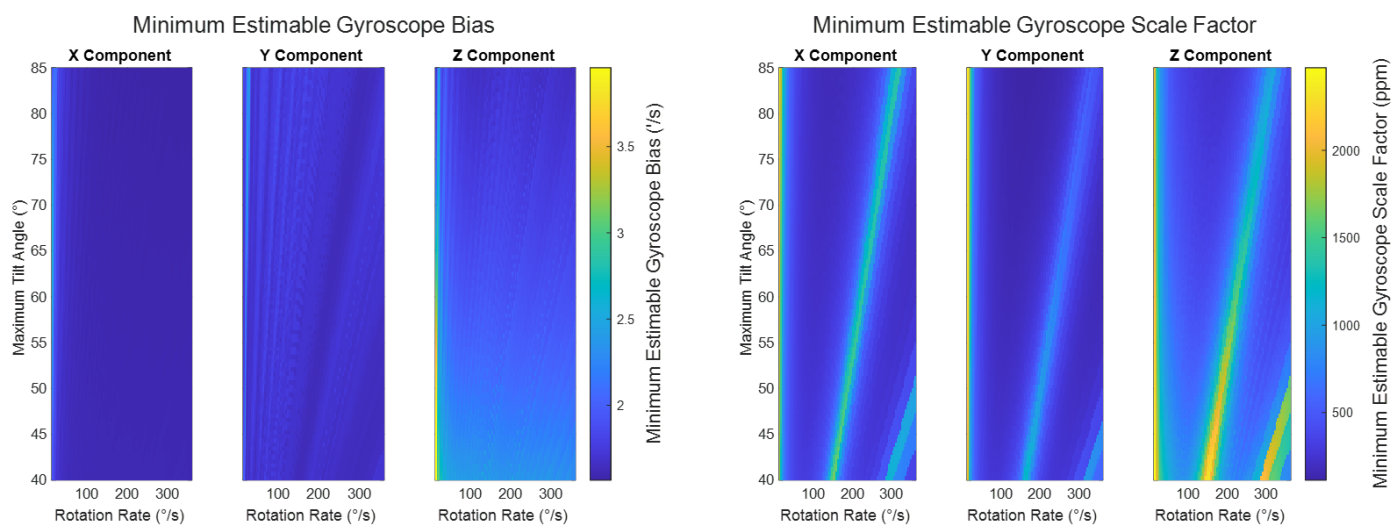


Figure 7.3.22: The minimum estimable values of the augmented gyroscope bias states (left) and scale factor error states (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).

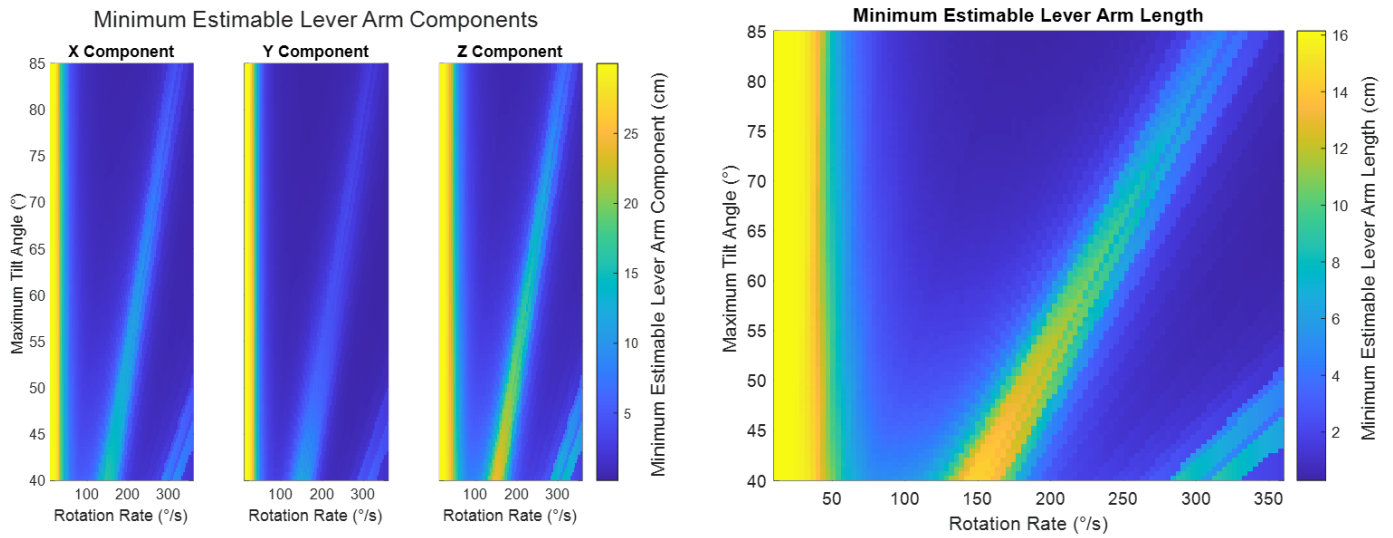


Figure 7.3.23: The minimum estimable values of the augmented lever arm states (left) and the absolute lever arm length (right) after rotating the PosAtt system through a series of tilt angles (rotation rate varies from 10 to 360 °/s, and the maximum tilt varies from 40 to 85°).

From Figure 7.3.21, the rotation rate and maximum system tilt angle did not have an apparent influence on the estimation of the accelerometer biases and scale factor errors. This is an intuitive result, since this maneuver does not involve any linear motion of the system.

The minimum estimable values of the augmented gyroscope bias states were not visibly influenced by the system's rotation rate and maximum tilt angle (Figure 7.3.22). Both the x- and y-gyroscope bias estimates benefit the most from using this maneuver, consistently reaching a minimum estimable value of about 2°/s ($\sigma_{gyro} = \pm 40' / s$) across most rotation rates and maximum system tilt values.

The minimum estimable values of the augmented gyroscope scale factor error states in Figure 7.3.22 were significantly impacted by the system's rotation rate and maximum tilt angle for this maneuver. It is important to note the degenerate pairings of rotation rate and maximum tilt angle that result in a lower accuracy of the augmented gyroscope scale factor error states. Outside of these degenerate combinations, the maximum tilt angle showed a clear impact on the accuracy of the augmented gyroscope scale factor error states than the rotation rate, where the larger maximum tilt angles generally yielded better estimates of the augmented scale factor error states.

This maneuver is particularly valuable in identifying the lever arm components and is much more effective in estimating the lever arms than any planar motion-based maneuvers. This is very intuitive when one considers that for a kinematic PosAtt system in planar motion, the vector $C_n^b \omega_{ib}^b$ will be collinear with ω_{ib}^b . The cross product between them is therefore equal to zero and the term $C_n^b \omega_{ib}^b \times \omega_{ib}^b \times r$ is therefore also equal to zero so that the lever arm components become unobservable for “purely” planar motion.

As with the circular motion maneuver, it is instructive to extract a set of promising tilting parameters from the outcomes of Figure 7.3.21 to Figure 7.3.23 and conduct a more thorough analysis and characterize how the minimum estimable values of the error states can improve as the tilting maneuver is repeated multiple times. Here, a rotation rate of 100°/s and a maximum tilt angle of 45° were used. Section 7.3.3.2 investigates the observability of the individual systematic error states under this maneuver and Section 7.3.3.3 investigates how the minimum estimable values of the systematic error states evolve as the maneuver is repeated.

7.3.3.2 Observability Analysis

From the simulated trajectory using a tilting maneuver with a maximum tilt angle of 45° and a rotation rate of 100°/s, the rank of the observability matrix is 27, indicating that there are three unobservable states in the system. Estimating the observability Gramian over the entire dataset yields an unobservability index of approximately 387,128 and an estimation condition number of 5,645,117,909. Again, the large magnitude of these quantities indicates that at least some of the systematic error states are unobservable over this dataset.

The 8 largest eigenvalues of the normalized state covariance matrix and the dominant components of the corresponding eigenvectors are provided in Table 7.3.9. The least observable states under the tilting maneuver include the accelerometer scale factor errors, GPS-IMU lever arm vector, and gyroscope scale factor errors.

Table 7.3.9: The observability results after the eigenvalues and eigenvectors of the normalized state covariance matrix.

Dominant States	Eigenvalue	Contribution to $\text{tr}(D_x^N)$ (%)
Accelerometer Scale – Z	20.5832	68.611

Accelerometer Scale – X and Y	7.6080	25.360
Accelerometer Scale – X and Y	0.7245	2.415
GPS-IMU Lever Arm – Y	0.7057	2.352
GPS-IMU Lever Arm – X	0.2341	0.780
GPS-IMU Lever Arm – Z	0.1442	0.481
Gyroscope Scale – Z	0.0001	0.000
Gyroscope Scale – X	0.0001	0.000
TOTAL	29.9999	99.999

The estimated observability indices for the individual state elements are plotted in Figure 7.3.24 to Figure 7.3.26, which have been used to successfully identify all systematic error states as observable, except for those related to the accelerometer scale factor errors.

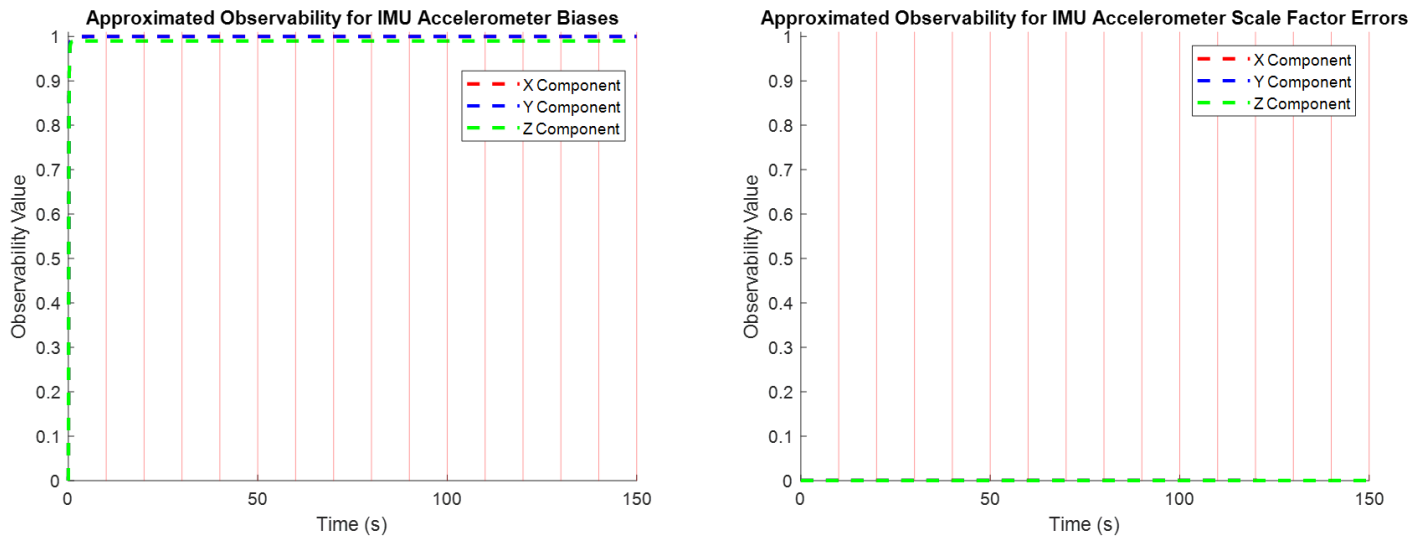


Figure 7.3.24: The observability of the augmented systematic error states for the accelerometers (left: biases; right: scale factor errors).

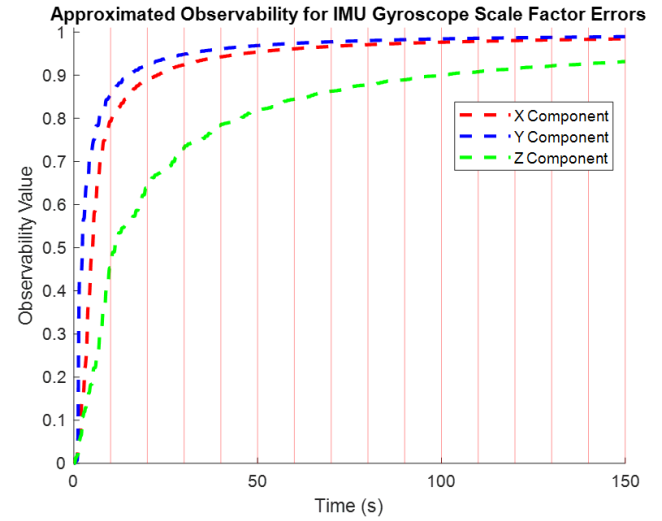
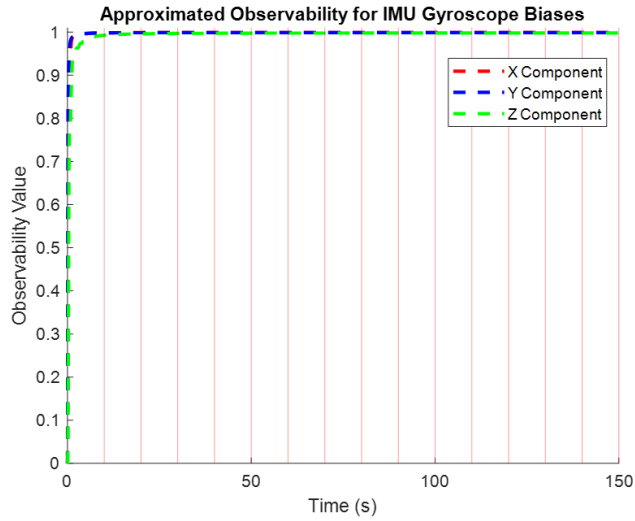


Figure 7.3.25: The observability of the augmented systematic error states for the gyroscopes (left: biases; right: scale factor errors).

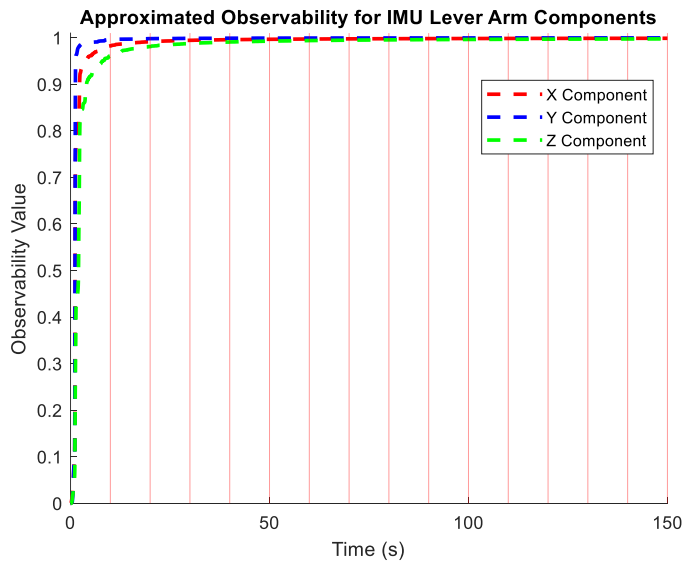


Figure 7.3.26: The observability of the augmented lever arm states.

7.3.3.3 Minimum Estimable Systematic Errors

Under the tilting maneuver trajectory with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° , the minimum estimable values of the accelerometer and gyroscope bias and scale factor error states are shown in Figure 7.3.27 and Figure 7.3.28, and the minimum estimable lever arm components and lever arm length are shown in Figure 7.3.29.

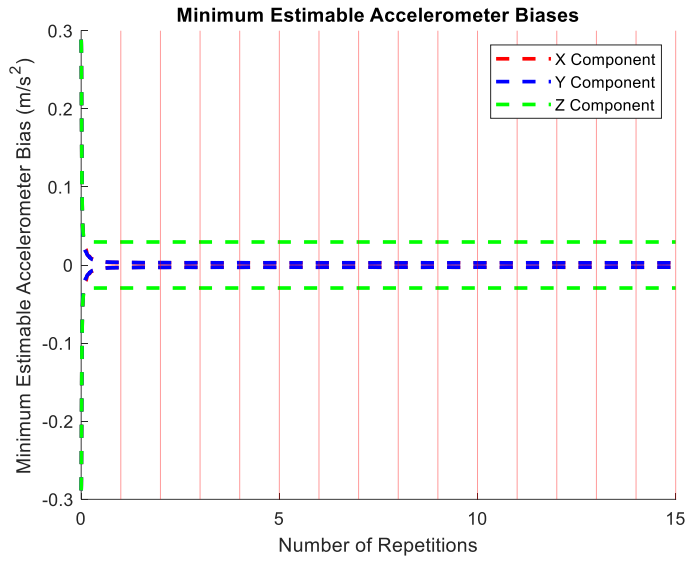


Figure 7.3.27: The minimum estimable values of the augmented accelerometer systematic error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).

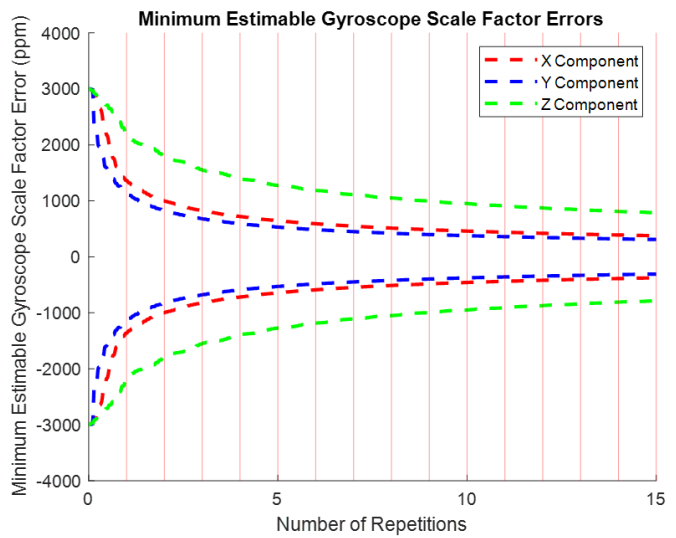
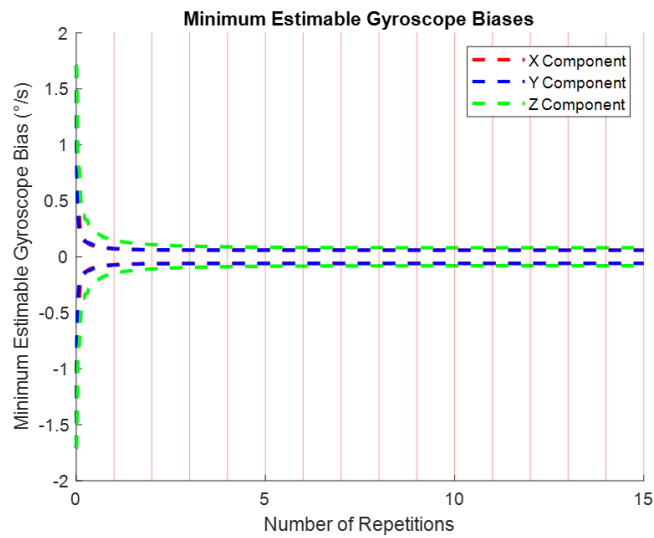


Figure 7.3.28: The minimum estimable values of the augmented gyroscope error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).

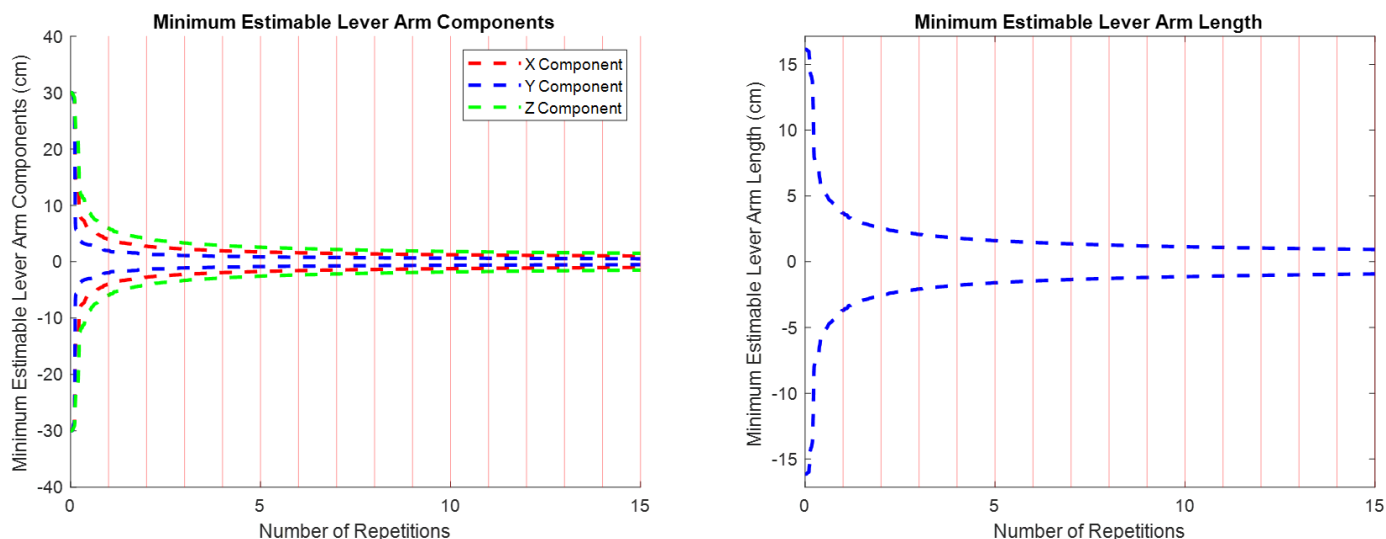


Figure 7.3.29: The minimum estimable values of the augmented lever arm states (left) and length of the absolute lever arm vector (right) under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (vertical red grid lines are shown whenever the PosAtt system returns to its starting orientation i.e. for each full repetition of the calibration maneuver).

As expected, these plots show a marked improvement in the estimable values of the observable systematic error states through time, and the values generally converged after a maximum of three repetitions of the calibration maneuver and then exhibited stable behaviour through time.

It is worth noting that the minimum estimable lever arm vector takes longer to converge than many of the other systematic error states, and that the lever arm components in particular benefit from repetitions of the calibration maneuver.

The converged minimum estimable IMU systematic error estimates for the tilting maneuver are provided in Table 7.3.10. Values are provided after 3 repetitions of the tilting maneuver and after 15 repetitions of the tilting maneuver.

Table 7.3.10: The minimum estimable values of the augmented IMU systematic error states under the tilting maneuver with a rotation rate of $100^\circ/\text{s}$ and a maximum tilt angle of 45° (after 3 repetitions and after 15 repetitions of the tilting maneuver). Augmented states that were not estimated due to their unobservability are marked with “-”.

Systematic Error State	Minimum Estimable Value					
	3 Repetitions			15 Repetitions		
	X	Y	Z	X	Y	Z

Accelerometer Bias [cm/s²]	0.3	0.3	2.9	0.3	0.3	2.9
Accelerometer Scale [ppm]	-	-	-	-	-	-
Gyroscope Bias [°/s]	3.6	3.6	5.5	3.6	3.6	5.4
Gyroscope Scale [ppm]	172	136	416	79	64	188
Lever Arm [cm]	2.2	1.1	3.3	1.0	0.5	1.5

7.3.3.4 Directly Comparing Covariance Matrices

To evaluate the impact that rotation rate has on the overall quality of IMU online calibration, a pair-wise comparison is conducted between different trajectories that have different rotation rates, but the same maximum tilt angle of 50°. A plot of the normalized covariance comparison measures is provided in Figure 7.3.30.

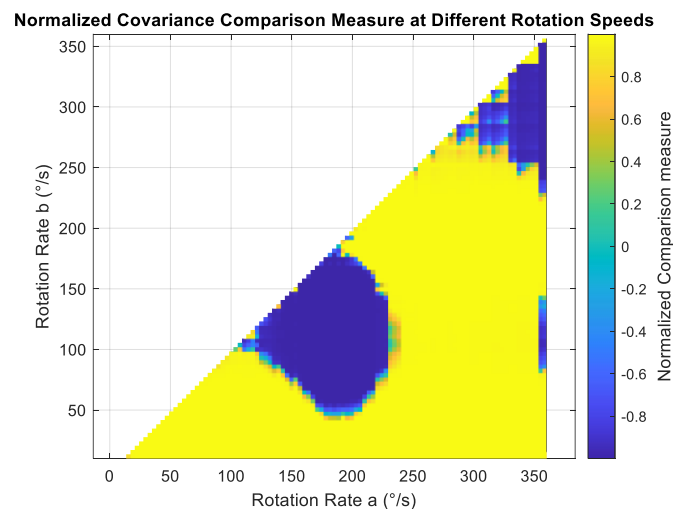


Figure 7.3.30: The estimated normalized covariance comparison measures for the tilting maneuver with a maximum tilt angle of 50° at different rotation rates. The covariance comparison measure is c_{ab} , with D_a determined using Rotation Rate a (X-axis) and D_b determined using Rotation Rate b (Y-Axis).

The normalized comparison measure indicates that generally, higher rotation rates correspond to improvements in the overall quality of the state covariance matrix, outside of some degenerate rotation rates that reduce the state covariance matrix's overall quality.

7.3.3.5 Minimum Detectable Observation Errors

The minimum detectable observation errors over the entire test trajectory are provided in Table 7.3.11.

Table 7.3.11: The minimum detectable values of the IMU accelerometer and gyroscope systematic errors as observation outliers, estimated over the entire test trajectory. Note that the detectable biases are identical across all three different sensor axes.

Observation Type	Minimum Detectable Observation Errors		
	X Component	Y Component	Z Component
Accelerometer [cm/s^2]	0.64	0.64	0.64
Gyroscope [$^\circ/\text{s}$]	1.26	1.26	1.26

7.4 Summary

This chapter proposes a novel framework of trajectory pre-analysis for the online sensor calibration of positioning/orientation sensors. The proposed framework allows for the performance of the following tasks in a pre-analysis environment:

- Evaluating minimum estimable values of the augmented systematic error states for any intrinsic/extrinsic errors being modeled in the Kalman Filter.
- Evaluating the minimum estimable length of a vector containing a group of intrinsic/extrinsic errors being modeled as part of the positioning Kalman Filter.
- Comparing the overall accuracy of multiple potential trajectories, to determine which provides an overall more reliable set of system states.
- Evaluating the redundancy contributions of different positioning/orientation measurements for arbitrary sensor configurations over a potential trajectory.
- Evaluating the minimum detectable values of the anticipated constant biases in the measurement and process noise vectors.

The proposed framework is robust and independent of specific sensor configuration. This allows for a wide range of applications in a system design environment, including:

- Testing different calibration maneuvers to determine an optimal set of guided maneuvers for users to undertake before beginning regular data collection procedures. The information gleaned from this analysis will be essential in ensuring that the observability of important sensor systematic error state elements is maximized when guiding end users through appropriate application-specific field procedures.
- Testing different sensor configurations over a test trajectory to determine which sensor configuration provides the best basis for online sensor calibration. This system design application has particularly interesting potential, especially when combined with the MIMU array modeling work accomplished in this research. This framework would allow for the determination of optimal sensor placement in a MIMU array, optimal sensor array size, and optimal combinations of varying-quality/dynamic range IMUs in a MIMU array, before needing to invest in physical systems for testing.

The proposed framework was realized for an INS/GNSS integrated system with the GNSS integrated in a loosely-coupled manner and used to evaluate three test scenarios: one where no specific calibration maneuvers were undertaken; one where circular motion was used as a calibration maneuver at the start of data collection; and one where the PosAtt system underwent a tilting maneuver at the start of data collection. All of these scenarios were pre-processed without any actual observation values, which allowed for their relative value in the context of online IMU sensor calibration to be evaluated.

8 Conclusions and Future Work

The TMIS has gaps in both system design and post-processing environments, where flexible and robust sensor modeling and comprehensive error analysis show particular utility. This research fills these gaps, and the contributions of this research are broadly centred on (1) extending the framework for comprehensive error analysis in Kalman Filtering, and (2) improving the ability to fully characterize sensor errors and system model performance in Kalman Filtering.

This research achieves the objectives specified in Section 1.2 by innovatively enhancing the capabilities and completeness of the GMIS in three main areas: (1) system attitude modeling (Chapter 5), (2) Multi-IMU (MIMU) array modeling and integration (Chapter 6), and (3) pre-analysis for online sensor calibration in Kalman filtering (Chapter 7). As a standard formulation, the framework of comprehensive error analysis in the state constrained Kalman Filter (Chapter 3) plays a very essential role in the relevant research. All of this was accomplished using the PosAtt Engine (Chapter 4), which realizes the GMIS for integrating IMU and GNSS data.

8.1 Conclusions

This research takes significant strides towards establishing the GMIS as a fully matured sensor integration strategy. This is accomplished by standardizing the system and measurement models under the GMIS, and fully leveraging the comprehensive error analysis tools that are enabled by the GMIS. The only work remaining to be done before the GMIS can see widespread adoption is to address its computational complexity. Specific future work relating to this task is discussed in Section 8.2.

The specific contributions made in this dissertation are summarized as follows:

First, Chapter 3 proposes an alternate derivation of the system state equality constrained Kalman Filter that allows for the development of equations to estimate the residuals, residual covariance matrices, and redundancy contributions for the observation, process noise, and predicted system state vectors, which were previously not analytically available. Chapter 3 demonstrates the equivalence of the state constrained solution to those available in the existing literature, and also delivers a standard formulation of how to apply the linearized constraints to the unconstrained solution strictly after the Least Squares Principle. Its significance is summarized as follows:

- i. It allows for the estimation of the residual vectors and their associated covariances and redundancy contributions to be integrated into existing workflows without requiring any broader changes.
- ii. It also opens the door to other avenues for analysis for state constrained Kalman Filters that are further explored in other chapters of this dissertation. Chief among these is Variance Component Estimation (VCE), which allows for an evaluation of the *a posteriori* performance of a system relative to its assumed accuracy.
- iii. It describes the concept of the time-varying VCE for state constrained Kalman Filters, and explores the effects of changing the window size used for VCE in a state constrained Kalman Filter. Generally, larger window sizes were found to produce less noisy estimates of any estimated variance components, at the cost of reducing their time resolution. These effects were stronger for variance components related to the system's process noise, since the process noise vector typically contributes significantly less towards the overall system redundancy than the observation vector does.

Second, attitude modeling is systematically studied in Chapter 5, and the GMIS was enhanced using the following attitude models: the Roll-Pitch-Heading (RPH) model, the Direction Cosine Matrix (DCM) model, and the Quaternion Model. The formulation of these models is unique in multiple respects:

- i. All three representations were structured in a similar way, considering the attitude parameters' first-order time-derivatives in the state vector and taking their second-order time-derivatives as white process noise. This development of the attitude models broadly standardizes the process of establishing system models for attitude kinematics.
- ii. The state constraint equations are developed not only for the attitude parameters in the DCM and Quaternion attitude models, but also for their first-order time derivatives. The formulation of the constraints on these attitude parameters is unique and opens up opportunities for further exploration of alternative models governing the attitude dynamics of the system. Some of these opportunities are discussed in Section 8.2.
- iii. The IMU measurements are directly modeled as functions of the corresponding attitude states in each attitude model. This makes it feasible to directly estimate the residuals of the IMU observations under each attitude model, which is normally not possible under the TMIS.

- iv. The use of VCE is explored to directly compare the performance of each attitude model without requiring access to a ground truth or reference solution. The variance components related to the process noise were of especial interest, since the attitude components of the process noise vector were considered to be indicative of the performance of the attitude system models. The VCE standard error estimates were unitless, and this provided an opportunity to directly compare the performance of each attitude model, despite the fact that they each used different units and even had a different number of associated elements in the process noise vector.

Chapter 6 provides a novel and innovative realization of the GMIS to accommodate data from a Multi-IMU (MIMU) array of arbitrary size and configuration, accounting for the individual IMU sensors' systematic errors and stochastic properties for every sensor in the MIMU array. There are several unique aspects to the method of MIMU array integration developed and adopted through this research:

- i. All data from the MIMU array are modeled directly in the positioning Kalman Filter without requiring the simulation of a "fused" sensor or INS output. Other methods for MIMU array data fusion are limited by the TMIS's requirement of a single "core" IMU sensor, and using the GMIS for MIMU array data processing completely removes this limitation.
- ii. All IMU systematic errors are modeled individually in the positioning Kalman Filter, including all gyroscope and accelerometer biases and scale factor errors. Other methods for MIMU array data fusion typically rely on assuming homogeneous behaviour of these systematic errors across different sensors in the MIMU array, and this research's approach is novel in modeling these sensor errors independently from one another.
- iii. All IMU stochastic properties are accounted for individually in the positioning Kalman Filter. Practically, this means that the MIMU array may more naturally integrate data from IMU sensors that have different Full Scale Range (FSR) values to effectively increase the FSR of the integrated PosAtt system, even when the IMU sensors' observations are generally of different qualities.

The unique approach to MIMU array integration adopted in this research allowed Chapter 6 to take a generalized approach to modeling IMU sensor observations in the positioning Kalman Filter that even accommodates atypical sensors that could not be readily integrated via the TMIS. Section 6.4.3 presented observation models for the individual sensing axes of an IMU sensor in the GMIS. Taking this generalized observation model as part of the positioning Kalman Filter allows for atypical sensors, such as four axis tetrahedrally arranged accelerometers, to be integrated naturally into the workflow for

processing MIMU array data. This level of flexibility is unparalleled in the TMIS and provides significant avenues for further development and improvement.

The successful realization of unique MIMU modeling analytics in Chapter 6 also provided a means to explore more detailed statistical analysis for MIMU array data. This detailed analysis includes:

- i. Rigorous Fault Detection and Exclusion (FDE) for each sensor in the MIMU array. Since the observation residuals may be directly estimated through the positioning Kalman Filter in the GMIS, they may be used to evaluate individual observations from each constituent IMU sensor to rigorously identify outliers.
- ii. Evaluation of how each constituent IMU sensor's systematic errors evolve over time. For MEMS sensors, which notoriously exhibit biases and scale factor errors that shift with time and over changing environmental conditions, this detailed characterization of time-varying sensor systematic errors is valuable.
- iii. Characterization of each constituent IMU sensor's overall performance through time using VCE. This allows for a thorough analysis of IMU observation quality with varying system dynamics, and opens opportunities for further exploration into adaptive methods that use VCE to adaptively scale the IMU observation covariance matrices.

An analytical framework for pre-analysing online sensor calibration of a multi-sensor system was proposed (Chapter 7), which allows for the identification of optimal procedures and sensor configurations in mission planning and system design. This pre-analysis framework allows for a holistic assessment of the potential performance of online sensor calibration over a candidate trajectory, including:

- i. Minimum significant values for individual systematic error parameters, and for the norm of groups of correlated systematic error parameters.
- ii. Minimum detectable values for observation biases from a particular sensor.
- iii. Redundancy contributions of individual observations in the positioning Kalman Filter.
- iv. Observability analysis for individual additional parameters in the positioning Kalman Filter.
- v. Direct comparison between the covariance matrices for different candidate trajectories/system configurations.

The proposed pre-analysis framework provides a solid foundation for assessing the relative value of different calibration maneuvers and sensor configurations in a PosAtt system, and presents the most holistic and comprehensive approach to pre-analysis of PosAtt systems in the literature.

8.2 Future Work

This dissertation opens up many avenues for further research and investigation with the GMIS that fall outside the scope of this research.

Many of the processing techniques used in this research place significantly more computational burden on the positioning Kalman Filter used in the GMIS, primarily through the inclusion of additional elements in the system state vector. The GMIS in general is less computationally efficient than the TMIS, and there are three factors in particular that increase the computational complexity of the positioning Kalman Filter in this research:

- Since the GMIS models all position/orientation observations from all constituent sensors, the positioning Kalman Filter needs its measurement update evaluated at every observation epoch. This is more frequent than would be necessary under the TMIS, which only requires measurement updates when there are aiding observations available.
- Since the IMU systematic errors are accounted for using additional state elements, the dimension of the positioning Kalman Filter is increased relative to what would be necessary under the TMIS.
- Since measures of solution quality are made using residual vectors instead of the system innovation, there are additional calculations required to estimate these residual quantities after the measurement update in the positioning Kalman Filter. This issue is not specific to the GMIS, but is necessarily introduced when conducting comprehensive error analysis.

Each of these factors presents a barrier to applying the GMIS in real-time applications, but there are avenues to address them and make the GMIS suitable even for embedded real-time systems.

To reduce the impact of applying the Kalman Filter measurement update at the typically higher data rate of constituent IMU sensors, pre-filtering the IMU data is a potentially valuable approach to artificially reduce the data rate during periods of time where the system is experiencing lower dynamics.

To reduce the impact of the added systematic error states from sensors being modeled in the positioning Kalman Filter, these added states could instead be modeled as part of a secondary Kalman

Filter that is not updated at the same rate as the core positioning Kalman Filter. This would drastically improve the performance of GMIS, especially for MIMU array data, and is considered an important next step in this research.

Beyond improving the efficiency of the positioning Kalman Filter used in the GMIS, a valuable avenue for future work with the GMIS is to extend the positioning filter beyond just using the Extended Kalman Filter (EKF). To date, all development of the GMIS has used either a Linear Kalman Filter or the EKF. Given the improvements that have been made to the GMIS' modeling capabilities and error analysis tools, a logical next step would be to incorporate other filters, such as the Particle Filter and the Unscented Kalman Filter, into the GMIS workflow. AI-based estimation techniques have recently seen broader adoption in the field of positioning and navigation, and there are also opportunities to incorporate these techniques into the GMIS to investigate the benefits they can bring to this integration strategy.

One major deliverable of this research is the PosAtt Engine, which realizes the GMIS for general GPS/IMU data integration. There are several future work items planned for the PosAtt Engine to generally improve its efficiency, robustness, and solution quality:

- Currently, the PosAtt Engine has placed the limit on the use of the double-differenced GPS observations between a stationary base receiver and a rover receiver due to project time restrictions without loss of generality. The further inclusion of measurements from other GNSS constellations such as Galileo, Beidou, and GLONASS, since using multi-constellation GNSS observables will definitely help this work reach its full potential.
- The PosAtt Engine currently only accommodates an EKF for position and attitude estimation. Building support for other filter types, such as the UKF, is another major priority for further development.
- The kinematic datasets that have been used to test the PosAtt Engine have provided very encouraging results, but gaining access to a broader range of datasets using a variety of different vehicle dynamics would allow for a more thorough testing of the PosAtt Engine's capabilities and limits.
- To facilitate more direct comparison with the TMIS, implementing the TMIS in the PosAtt Engine is another major priority for further work.

Outside of the general improvements that may be made to the GMIS and the PosAtt Engine, there are many avenues for further development that arise specifically from the research conducted in this dissertation.

The attitude modeling work conducted in Chapter 5 is an important starting point for analyzing attitude models using the GMIS. The available testing data was unfortunately limited to ground vehicle data, with very limited dynamics in the system roll and pitch. It would be valuable to perform the sort of comparative analysis that was accomplished in Chapter 5 for airborne data that has more dynamic shifts in the system roll and pitch, to properly evaluate how each attitude model performs, even under degenerate conditions. Additionally, adaptively applying the results of VCE to ongoingly tune the KF observation and process noise covariance matrices is a particularly impactful avenue for further work, and could be especially impactful for real-time analysis.

The method of applying state constraints on the first-order time derivatives of the attitude parameters in Chapter 5 is also readily extendable to higher-order time derivatives of the attitude parameters. This allows for future exploration of more complex system models of the system attitude, which may be valuable for applications where the system's attitude is subject to highly varying dynamics.

The MIMU array integration approach taken in Chapter 6 has significant potential, especially in modeling atypical IMU sensors. One avenue for further research is to collect data using an atypical IMU and explore the relative value of using atypical IMUs for sensor integration compared to using data from multiple IMUs.

There are many sources of systematic errors that are exacerbated by large lever arms between the origin of the navigation frame and IMU sensors. The approach to MIMU integration taken in this research not only gives more insight into the individual performance of IMUs in a MIMU array, but also has potential in modeling errors that are impacted by the IMU lever arms.

The pre-analysis framework proposed in Chapter 7 provides a valuable perspective to evaluate the reliability that a given candidate trajectory provides to online sensor calibration. This is a framework that can be built upon and further explored to, for example, provide detailed information on the relative quality of different sensor configurations. This has particular promise in evaluating MIMU array configurations, where the size of the MIMU array and relative geometry of the sensors in the array may

be modified to determine optimal sensor placements without requiring constant reconfiguration of a physical system and repeated data collection.

References

- Abeyaratne, R. (1998). Continuum mechanics. Lecture Notes on The Mechanics of Elastic Solids.
- Advanced Navigation. (2025). *Inertial Measurement Unit (IMU) – an introduction*.
<https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/#h-interpreting-inertial-measurement-unit-specifications>
- Aggrey, J., Bisnath, S., Naciri, N., Shinghal, G., & Yang, S. (2020). Multi-GNSS precise point positioning with next-generation smartphone measurements. *Journal of Spatial Science*, 65(1), 79–98.
<https://doi.org/10.1080/14498596.2019.1664944>
- Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems*, 1(2), 256-262.
- Alexiev, K. (2019). Algorithms for imu navigation—a review. In *Bulgarian Academy of Sciences*.
- Alouani, A. T., & Blair, W. D. (1991). Use of a kinematic constraint in tracking constant speed, maneuvering targets. *Proceedings of the 30th IEEE Conference on Decision and Control*.
<https://doi.org/10.1109/cdc.1991.261780>
- An, X., Meng, X., & Jiang, W. (2020). Multi-constellation GNSS precise point positioning with multi frequency raw observations and dual-frequency observations of ionospheric-free linear combination. *Satellite Navigation*, 1(1), 7. <https://doi.org/10.1186/s43020-020-0009-x>
- Anderson, J. M., & Mikhail, E. M. (2015). " Surveying, Theory and Practice", McGraw Hill, 2001.
- Aqel, M.O.A., Marhaban, M.H., Saripan. (2016). M.I. *et al.* Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* 5, 1897. <https://doi.org/10.1186/s40064-016-3573-7>
- Azzam, R., Kong, F. H., Taha, T., & Zweiri, Y. (2021). Pose-graph neural network classifier for global optimality prediction in 2D SLAM. *IEEE Access*, 9, 80466-80477.
- Backman, J., Oksanen, T., & Visala, A. (2012). Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. *Computers and Electronics in Agriculture*, 82, 32–43.
<https://doi.org/10.1016/j.compag.2011.12.009>
- Bahadur, B., & Nohutcu, M. (2021). Integration of variance component estimation with robust Kalman filter for single-frequency multi-GNSS positioning. *Measurement*, 173, 108596.

- Bai, F., Vidal-Calleja, T., & Grisetti, G. (2021). Sparse pose graph optimization in cycle space. *IEEE Transactions on Robotics*, 37(5), 1381-1400.
- Balaban, E., Saxena, A., Bansal, P., Goebel, K. F., & Curran, S. (2009). Modeling, Detection, and Disambiguation of Sensor Faults for Aerospace Applications. *IEEE Sensors Journal*, 9(12), 1907-1917. *IEEE Sensors Journal*. <https://doi.org/10.1109/JSEN.2009.2030284>
- Bancroft, J.B. and G. Lachapelle (2011): Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors* 2011, 11, 6771-6798, doi.org/10.3390/s110706771
- Beatty, M. F. (1986). Kinematics of Rigid Body Motion. In M. F. Beatty (Ed.), *Principles of Engineering Mechanics: Kinematics—The Geometry of Motion* (pp. 85–149). Springer US. https://doi.org/10.1007/978-1-4899-7285-9_2
- Becker, M., Bestmann, U., Schwithal, A., Hecker, P., Bott, R., Kohl, P., Knedlik, S., Scheyer, H., & Hinüber, E. v. (2010). Evaluation of a double equipped MEMS IMU based on real flight trial scenarios. *IEEE/ION Position, Location and Navigation Symposium*, 202–213. <https://doi.org/10.1109/PLANS.2010.5507129>
- Beran, T., Langley, R. B., Bisnath, S. B., & Serrano, L. (2007). High-Accuracy Point Positioning with Low-Cost GPS Receivers. *NAVIGATION*, 54(1), 53–63. <https://doi.org/10.1002/j.2161-4296.2007.tb00395.x>
- Bernardes, E., & Viollet, S. (2022). Quaternion to Euler angles conversion: A direct, general and computationally efficient method. *Plos one*, 17(11), e0276302.
- Bernstein, D. (2013). Geometry, kinematics, statics and dynamics. *University of Michigan*.
- Bertiger, W., Desai, S. D., Haines, B., Harvey, N., Moore, A. W., Owen, S., & Weiss, J. P. (2010). Single receiver phase ambiguity resolution with GPS data. *Journal of Geodesy*, 84(5), 327–337 <https://doi.org/10.1007/s00190-010-0371-9>
- Biswas, B. N., Chatterjee, S., Mukherjee, S. P., & Pal, S. (2013). A discussion on Euler method: A review. *Electronic Journal of Mathematical Analysis and Applications*, 1(2), 2090-2792.
- Blewitt, G. (1997). Basics of the GPS technique: observation equations. *Geodetic applications of GPS*, 1, 46.

- Breuer, M., Konrad, T., & Abel, D. (2016). *High Precision Localisation in Customised GNSS Receiver for Railway Applications*. 779–787. <https://doi.org/10.33012/2016.14696>
- Brossard, M., S. Bonnabel and A. Barrau (2020). "Denoising IMU gyroscopes with deep learning for open loop attitude estimation", *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4796-4803.
- Brossard, M., & Bonnabel, S. (2019, May). Learning wheel odometry and IMU errors for localization. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 291-297). IEEE.
- Brunker, A., Wohlgemuth, T., Frey, M., & Gauterin, F. (2018). Odometry 2.0: A Slip-Adaptive EIF-Based Four-Wheel-Odometry Model for Parking. *IEEE Transactions on Intelligent Vehicles*, 1–1. doi:10.1109/tiv.2018.2886675
- Brunson, B., & Wang, J. (2025). Analytical Framework for Online Calibration of Sensor Systematic Errors Under the Generic Multisensor Integration Strategy. *Sensors*, 25(10), 3239. <https://doi.org/10.3390/s25103239>
- Brunson, B., Wang, J., & Ma, W. (2024). Innovative Modeling of IMU Arrays Under the Generic Multi Sensor Integration Strategy. *Sensors*, 24(23), 7754.
- Brunson, B., & Wang, J. (2023). Three Attitude Models and their Characterization in the Generic Multisensor Integration Strategy for Kinematic Positioning and Navigation. *The Journal of Global Positioning Systems*, 19(1), 79-98.
- Bryson, M., & Sukkarieh, S. (2004, December). Vehicle model aided inertial navigation for a UAV using low-cost sensors. In *Proceedings of the Australasian Conference on Robotics and Automation* (pp. 1-9). Australian Robotics and Automation Association.
- Butcher, J. C. (1987). *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. J. Wiley.
- Capriglione, D., Carratù, M., Catelani, M., Ciani, L., Patrizi, G., Pietrosanto, A., Singuaroli, R., & Sommella, P. (2021). Analysis of MEMS devices under temperature stress test. *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 63–68. <https://doi.org/10.1109/MetroAeroSpace51421.2021.9511744>
- Cao, D., Rakheja, S., & Su, C. Y. (2008). Heavy vehicle pitch dynamics and suspension tuning. Part I: unconnected suspension. *Vehicle System Dynamics*, 46(10), 931–953. <https://doi.org/10.1080/00423110701732899>

- Carrera Villacrés, J. L., Zhao, Z., Braun, T., & Li, Z. (2019). A Particle Filter-Based Reinforcement Learning Approach for Reliable Wireless Indoor Positioning. *IEEE Journal on Selected Areas in Communications*, 37(11), 2457–2473. IEEE Journal on Selected Areas in Communications. <https://doi.org/10.1109/JSAC.2019.2933886>
- Carlin, L., Hauschild, A., & Montenbruck, O. (2021). Precise point positioning with GPS and Galileo broadcast ephemerides. *GPS Solutions*, 25(2), 77. <https://doi.org/10.1007/s10291-021-01111-4>
- Carlone, L., Calafiore, G. C., Tommolillo, C., & Dellaert, F. (2016). Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3), 545-565.
- Carlone, L., Aragues, R., Castellanos, J. A., & Bona, B. (2014). A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, 33(7), 965–987. <https://doi.org/10.1177/0278364914523689>
- Carlsson, H., et al. (2022): "Inertial navigation using an inertial sensor array." *arXiv preprint arXiv:2201.11983*
- Caspary, W., & Wang, J. (1998). Redundanzanteile und Varianzkomponenten im Kalman Filter. *Zeitschrift für Vermessungswesen*, 123(4), 121-128.
- Caspary, W., & Rüeger, J. M. (2000). Concepts of network and deformation analysis. School of Geomatic Engineering, University of New South Wales.
- Chandrasekharaiah, D. S., & Debnath, L. (2014). *Continuum mechanics*. Elsevier.
- Chang, L., Di, J., & Qin, F. (2022). Inertial-Based Integration With Transformed INS Mechanization in Earth Frame. *IEEE/ASME Transactions on Mechatronics*, 27(3), 1738–1749. IEEE/ASME Transactions on Mechatronics. <https://doi.org/10.1109/TMECH.2021.3090428>
- Chen, C., & Pan, X. (2025). Deep learning for inertial positioning: A survey. *IEEE transactions on intelligent transportation systems*, 25(9), 10506-10523.
- Chen, L., Wang, Z., Zheng, F., Gong, X., Tao, J., & Shi, C. (2023). Characteristic analysis and calibration of CDMA and FDMA intra-system and inter-system biases in multi-GNSS precise point positioning. *GPS Solutions*, 27(4), 203.

- Chen, G., Wei, N., Tao, J., & Zhao, Q. (2024). Comparison between GPS network analysis with undifferenced and double differenced integer ambiguity resolution: A practical perspective. *Advances in Space Research*, 74(2), 582–595. <https://doi.org/10.1016/j.asr.2024.03.076>
- Chepur, S. P., Simonetto, A., Leus, G., & van der Veen, A. J. (2013, December). Tracking position and orientation of a mobile rigid body. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (pp. 37-40). IEEE.
- Choo, H., Lee, B., Kim, H., & Choi, B. (2023). Automated detection of construction work at heights and deployment of safety hooks using IMU with a barometer. *Automation in Construction*, 147, 104714. <https://doi.org/10.1016/j.autcon.2022.104714>
- Civera, J., Grasa, O. G., Davison, A. J., & Montiel, J. M. M. (2009, October). 1-point RANSAC for EKF-based structure from motion. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3498-3504). IEEE.
- Civera, J., Grasa, O. G., Davison, A. J., & Montiel, J. M. (2010). 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of field robotics*, 27(5), 609-631.
- Cocheteux, M., Moreau, J., & Davoine, F. (2025). Uncertainty-Aware Online Extrinsic Calibration: A Conformal Prediction Approach (No. arXiv:2501.06878). arXiv. <https://doi.org/10.48550/arXiv.2501.06878>
- Collins, P., Bisnath, S., Lahaye, F., & Héroux, P. (2010). Undifferenced GPS Ambiguity Resolution Using the Decoupled Clock Model and Ambiguity Datum Fixing. *NAVIGATION*, 57(2), 123–135. <https://doi.org/10.1002/j.2161-4296.2010.tb01772.x>
- Crampin, M. (1986). On the concept of angular velocity. *European Journal of Physics*, 7(4), 287. <https://doi.org/10.1088/0143-0807/7/4/014>
- Crandall, D., Owens, A., Snavely, N., & Huttenlocher, D. (2011, June). Discrete-continuous optimization for large-scale structure from motion. In *CVPR 2011* (pp. 3001-3008). IEEE.
- Crocoll, P., Seibold, J., Scholz, G., & Trommer, G. F. (2014). Model-Aided Navigation for a Quadrotor Helicopter: A Novel Navigation System and First Experimental Results. *NAVIGATION*, 61(4), 253-271. <https://doi.org/10.1002/navi.68>

- Deepika, M.G., and A. Arun (2018). "Analysis of INS Parameters and Error Reduction by Integrating GPS and INS Signals." In *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, 18–23. doi.org/10.1109/ICDI3C.2018.00013.
- De Geeter, J., Van Brussel, H., De Schutter, J., & Decréton, M. (1997). A smoothly constrained Kalman filter. *IEEE transactions on pattern analysis and machine intelligence*, 19(10), 1171-1177.
- Dellenbach P, Deschaud JE, Jacquet B et al (2022) CT-ICP: real-time elastic LiDAR odometry with loop closure. In: 2022 international conference on robotics and automation (ICRA). IEEE, pp 5580–5586
- De Paula, P. A., Dos Santos, M. F., Mercorelli, P., & Schettino, V. B. (2024). PID Controller Application in a Gimbal Construction for Camera Stabilization and Tracking. *2024 25th International Carpathian Control Conference (ICCC)*, 01–06. https://doi.org/10.1109/ICCC62069.2024.10569310
- Ding, W., Sun, W., Gao, Y., & Wu, J. (2021). Carrier Phase-Based Precise Heading and Pitch Estimation Using a Low-Cost GNSS Receiver. *Remote Sensing*, 13(18), Article 18. https://doi.org/10.3390/rs13183642
- Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. *IEEE Signal Processing Magazine*, 20(5), 19–38.
- Doer, C., & Trommer, G. F. (2020, September). An ekf based approach to radar inertial odometry. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (pp. 152-159). IEEE.
- Du, S., Sun, W., & Gao, Y. (2016). MEMS IMU Error Mitigation Using Rotation Modulation Technique. *Sensors*, 16(12), Article 12. https://doi.org/10.3390/s16122017
- Elfring, J., Torta, E., & van de Molengraft, R. (2021). Particle Filters: A Hands-On Tutorial. *Sensors*, 21(2), Article 2. https://doi.org/10.3390/s21020438
- El-Sheimy, N., Chiang, K. W., & Noureldin, A. (2006). The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments. *IEEE Transactions on instrumentation and measurement*, 55(5), 1606-1615.
- Engel J, Schops T, Cremers D (2014) LSD-SLAM: large-scale direct monocular SLAM. In: European Conference on computer vision, Zurich, p 834–849

- Fan, H. (2024). Attitude Calculation Algorithm for Quadrotor UAV Based on Acceleration Correction Model. *International Journal of Computer Science and Information Technology*, 3(2), Article 2. <https://doi.org/10.62051/ijcsit.v3n2.42>
- Fang, H., Haile, M. A., & Wang, Y. (2021). Robust Extended Kalman Filtering for Systems with Measurement Outliers. *IEEE Transactions on Control Systems Technology*, 1–8. doi:10.1109/tcst.2021.3077535
- Fang, K., Cai, T., & Wang, B. (2024). The Kinematic Models of the SINS and Its Errors on the SE(3) Group in the Earth-Centered Inertial Coordinate System. *Sensors*, 24(12), Article 12. <https://doi.org/10.3390/s24123864>
- Fehlberg, E. (1969). Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten Kontrolle. *Computing*, 4(2), 93–106. <https://doi.org/10.1007/BF02234758>
- Fehlberg, E. (1969). *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems* (Vol. 315). National aeronautics and space administration.
- Feng, D., Wang, C., He, C., Zhuang, Y., & Xia, X.-G. (2020). Kalman-Filter-Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and Navigation. *IEEE Internet of Things Journal*, 7(4), 3133–3146. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2020.2965115>
- Förstner, W. (1979). Ein verfahren zur schätzung von varianz-und kovarianzkomponenten. *Allgemeine Vermessungsnachrichten*, 86(11-12), 446-453.
- Friess, P. (1990). *Kinematische Positionsbestimmung für die Aerotriangulation mit dem NAVSTAR Global Positioning System*. Verlag der Bayerischen Akademie der Wissenschaften in Kommission bei der CH Beck'schen Verlagsbuchhandlung.
- Gagnon, E.; Vachon, A. and Y. Beaudoin (2018): Data Fusion Architectures for Orthogonal Redundant Inertial Measurement Units. *Sensors* **2018**, 18, 1910, doi.org/10.3390/s18061910
- Gao, L., Xia, X., Zheng, Z., & Ma, J. (2023). GNSS/IMU/LiDAR fusion for vehicle localization in urban driving environments within a consensus framework. *Mechanical Systems and Signal Processing*, 205(Complete). <https://doi.org/10.1016/j.ymssp.2023.110862>
- Gao, Z., Shen, W., Zhang, H., Ge, M., & Niu, X. (2016). Application of Helmert variance component based adaptive Kalman filter in multi-GNSS PPP/INS tightly coupled integration. *Remote Sensing*, 8(7), 553.

- Gebre-Egziabher, D., & Gleason, S. (2009). *GNSS applications and methods*. Artech House.
- Geiger, W., Bartholomeyczik, J., Breng, U., Gutmann, W., Hafen, M., Handrich, E., Huber, M., Jackle, A., Kempfer, U., Kopmann, H., Kunz, J., Leinfelder, P., Ohmberger, R., Probst, U., Ruf, M., Spahlinger, G., Rasch, A., Straub-Kalthoff, J., Stroda, M., ... Zimmermann, S. (2008). MEMS IMU for AHRS applications. *2008 IEEE/ION Position, Location and Navigation Symposium*, 225–231. <https://doi.org/10.1109/PLANS.2008.4569973>
- Ghanbarpourasl, H., & Zobar, S. (2022). Kalman Filtering With Linear State Equality Constraints: A MethodBased on Separating Variables. *IEEE Access*, 10, 35460-35468.
- Ghanizadegan, K., & Hashim, H. A. (2024). Quaternion-based Unscented Kalman Filter for 6-DoF Vision-based Inertial Navigation in GPS-denied Regions. *IEEE Transactions on Instrumentation and Measurement*.
- Ghilani, C. D. (2017). *Adjustment computations: spatial data analysis*. John Wiley & Sons.
- Giroux, R., Gourdeau, R., & Landry, R. (2005, April). Extended Kalman filter implementation for low-cost INS/GPS Integration in a Fast Prototyping Environment. In *Symposium on Navigation of the Canadian Navigation Society* (pp. 1-11).
- Godha, S., & Lachapelle, G. (2008). Foot mounted inertial system for pedestrian navigation. *Measurement Science and Technology*, 19(7), 075202. doi:10.1088/0957-0233/19/7/075202
- Gong, W., Li, J., & Yang, J. (2009). A Novel Positioning Algorithm with Pseudo Range and Carrier Phase Based on Kalman Filter. *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, 1, 417–420. <https://doi.org/10.1109/NSWCTC.2009.75>
- Gopaul, N., Wang, J. G., & Scherzinger, B. (2009). Simplified algorithms of variance component estimation for static and kinematic GPS single point positioning. *Journal of Global Positioning Systems*, 8(1), 43-52.
- Gopaul, N. S., Wang, J., & Guo, J. (2010). On posteriori variance and covariance components estimation in GPS relative positioning. *Proceedings of CPGPS*, 141-148.
- Grammenos, A., Mascolo, C., & Crowcroft, J. (2018). You Are Sensing, but Are You Biased? A User Unaided Sensor Calibration Approach for Mobile Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), 11:1-11:26. <https://doi.org/10.1145/3191743>

- Greenheck, D. (2015). Design and Characterization of a Low Cost MEMS IMU Cluster for Precision Navigation. *Master's Theses (2009 -)*. https://epublications.marquette.edu/theses_open/325
- Grewal, M., Weill, L., & Andrews, A. (2007). *Global positioning systems, inertial navigation, and integration, second edition*. Wiley-Interscience.
- Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., & Hertzberg, C. (2010, May). Hierarchical optimization on manifolds for online 2D and 3D mapping. In *2010 IEEE International Conference on Robotics and Automation* (pp. 273-278). IEEE.
- Grisetti, G., Stachniss, C., & Burgard, W. (2009). Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10(3), 428-439.
- Groves P., 2013. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*, Artech.
- Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7), 53–82.
- Ham, F. M., & Brown, R. G. (1983). Observability, Eigenvalues, and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(2), 269–273. *IEEE Transactions on Aerospace and Electronic Systems*. <https://doi.org/10.1109/TAES.1983.309446>
- Hamidi, H., E.S. Abdolkarimi, and M. R. Mosavi (2020). "Prediction of MEMS-Based INS Error Using Interval Type-2 Fuzzy Logic System in INS/GPS Integration." In *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*, 1–5. doi.org/10.1109/CSICC49403.2020.9050081.
- Hanson, R., & Pachter, M. (2005, August). Optimal gyro-free IMU geometry. In *AIAA guidance, navigation, and control conference and exhibit* (p. 6151).
- Heller, W. G. (1975). Free-Inertial and Damped-Inertial Navigation Mechanization and Error Equations. *TASC No. TR-312-1-1, The Analytical Sciences Corporation, Reading, MA*.
- Hemingway, E.G., O'Reilly, O.M. (2018). Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody Syst Dyn* 44, 31–56. <https://doi.org/10.1007/s11044-018-9620-0>

- Herman, L. K. (1995). The history, definition and peculiarities of the Earth centered inertial (ECI) coordinate frame and the scales that measure time. *1995 IEEE Aerospace Applications Conference. Proceedings*, 2, 233–263 vol.2. <https://doi.org/10.1109/AERO.1995.468933>
- Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2012). Observability-constrained vision aided inertial navigation. University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep, 1, 6.
- Höflinger, F., Müller, J., Zhang, R., Reindl, L. M., & Burgard, W. (2013). A Wireless Micro Inertial Measurement Unit (IMU). *IEEE Transactions on Instrumentation and Measurement*, 62(9), 2583–2595. *IEEE Transactions on Instrumentation and Measurement*. <https://doi.org/10.1109/TIM.2013.2255977>
- Huang, F., Z. Wang, L. Xing and C. Gao (2022). "A MEMS IMU gyroscope calibration method based on deep learning", *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1-9.
- Hwangbo, M., Kim, J. S., & Kanade, T. (2013). IMU self-calibration using factorization. *IEEE Transactions on Robotics*, 29(2), 493-507.
- Janssen, V. (2001). Optimizing the Number of Double-Differenced Observations for GPS Networks in Support of Deformation Monitoring Applications. *GPS Solutions*, 4(3), 41–46. <https://doi.org/10.1007/PL00012854>
- Jekeli, C. (2001). *Inertial Navigation Systems with Geodetic Applications*. Berlin, Boston: De Gruyter. <https://doi.org/10.1515/9783110800234>
- Jiang, B., Gao, H., Han, F., & Dong, H. (2021). Recursive filtering for nonlinear systems subject to measurement outliers. *Science China Information Sciences*, 64(7), 172206.
- Jiang, C., et al. (2018). "A MEMS IMU de-noising method using long short term memory recurrent neural networks (LSTM-RNN)", *Sensors*, vol. 18, no. 10, pp. 3470.
- Johnson, B., Albrecht, C., Braman, T., Christ, K., Duffy, P., Endean, D., Gnerlich, M., & Reinke, J. (2021). Development of a Navigation-Grade MEMS IMU. *2021 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, 1–4. <https://doi.org/10.1109/INERTIAL51137.2021.9430466>
- Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. *Signal Processing, Sensor Fusion, and Target Recognition VI*, 3068, 182–193.

<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3068/1/New-extension-of-the-Kalman-filter-to-nonlinear-systems/10.1117/12.280797.short>

- Juillard, J. (2020). *Sensors & MEMS: Metrology, Electronics, Physics, etc.* Group of Electrical Engineering of Paris.
- Jung, J. H., Heo, S., & Park, C. G. (2020). Observability analysis of IMU intrinsic parameters in stereo visual-inertial odometry. *IEEE Transactions on Instrumentation and Measurement*, 69(10), 7530-7541.
- Junoh, S. A., Subedi, S., & Pyun, J.-Y. (2021). Floor Map-Aware Particle Filtering Based Indoor Navigation System. *IEEE Access*, 9, 114179–114191. IEEE Access. <https://doi.org/10.1109/ACCESS.2021.3102992>
- Jwo, D. J., Biswal, A., & Mir, I. A. (2023). Artificial neural networks for navigation systems: A review of recent research. *Applied Sciences*, 13(7), 4475.
- Kalogiratou, Z., Monovasilis, Th., Psihoyios, G., & Simos, T. E. (2014). Runge–Kutta type methods with special properties for the numerical integration of ordinary differential equations. *Physics Reports*, 536(3), 75–146. <https://doi.org/10.1016/j.physrep.2013.11.003>
- Kanschat, G., & Scheichl, R. (2021, July 20). Numerical Analysis of ordinary differential equations. https://katana.iwr.uni-heidelberg.de/pdfs/NumODE21_notes.pdf
- Kasdin, N. J. (1995). Runge-Kutta algorithm for the numerical integration of stochastic differential equations. *Journal of Guidance, Control, and Dynamics*, 18(1), 114–120. <https://doi.org/10.2514/3.56665>
- Kastella, K., Yearly, B., Zadra, T., Brouillard, R., & Frangione, E. (2000). Bias modeling and estimation for GMTI applications. *Proceedings of the Third International Conference on Information Fusion*, 1, TUC1/7-TUC112 vol.1. <https://doi.org/10.1109/IFIC.2000.862677>
- Kempe, V. (2011). *Inertial MEMS - Principles and Practice - 7.1 General Measurement Objectives*. (pp. 283). Cambridge University Press. Retrieved from <https://app.knovel.com/hotlink/pdf/id:kt008MUGC1/inertial-mems-principles/general-measurement-objectives>
- Khan, M., Kai, Y. D., & Gul, H. U. (2017). Indoor Wi-Fi positioning algorithm based on combination of Location Fingerprint and Unscented Kalman Filter. *2017 14th International Bhurban Conference*

- on *Applied Sciences and Technology (IBCAST)*, 693–698.
<https://doi.org/10.1109/IBCAST.2017.7868127>
- Kirkko-Jaakkola, M., Collin, J., & Takala, J. (2012). Bias Prediction for MEMS Gyroscopes. *IEEE Sensors Journal*, 12(6), 2157–2163. IEEE Sensors Journal. <https://doi.org/10.1109/JSEN.2012.2185692>
- Kitt, B., Geiger, A., & Lategahn, H. (2010, June). Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *2010 IEEE intelligent vehicles symposium* (pp. 486–492). IEEE.
- Knight, N. L., Wang, J., & Rizos, C. (2010). Generalised measures of reliability for multiple outliers. *Journal of geodesy*, 84, 625–635.
- Ko, S., & Bitmead, R. R. (2007). State estimation for linear systems with state equality constraints. *Automatica*, 43(8), 1363–1368. <https://doi.org/10.1016/j.automatica.2007.01.017>
- Koifman, M., & Bar-Itzhack, I. Y. (1999). Inertial navigation system aided by aircraft dynamics. *IEEE transactions on control systems technology*, 7(4), 487–493.
- Kong, X. (2000). *Inertial Navigation System Algorithms for Low Cost IMU*. Department of Mechanical and Mechatronic Engineering, Graduate School of Engineering, University of Sydney.
- Konrad, T., Breuer, M., Engelhardt, T., & Abel, D. (2017). State Estimation for a Multirotor using Tight Coupling of GNSS and Inertial Navigation *. *IFAC-PapersOnLine*, 50(1), 11683–11688. <https://doi.org/10.1016/j.ifacol.2017.08.1684>
- Krener, A. J., & Ide, K. (2009). Measures of unobservability. Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference. doi:10.1109/cdc.2009.5400067
- Kuga, H. K., Siqueira, R. H., Carrara, V., & De Oliveira, E. J. (2012). On ground calibration of tetrahedron gyro package for attitude determination. *Advances in the Astronautical Sciences*, 145, 353–364.
- Kuipers, J. B. (1999). *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press.
- Kuzmanovic, V., & Pajic, S. (2021). MEMS based IMU adaptive 3D calibration. *Industry 4.0*, 6(3), 96–99.
- Langley, R. B., Teunissen, P. J., & Montenbruck, O. (2017). Introduction to GNSS. *Springer handbook of global navigation satellite systems*, 3–23.

- Lashley, M., & Bevy, D. M. (2013). Performance Comparison of Deep Integration and Tight Coupling. *Navigation*, 60(3), 159–178. doi:10.1002/navi.43
- Le, B. L., Ahmed, K., & Tsuji, H. (2003, March). Mobile location estimator with NLOS mitigation using Kalman filtering. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*. (Vol. 3, pp. 1969-1973). IEEE.
- Lee, D., Jung, M., Yang, W., & Kim, A. (2024). Lidar odometry survey: recent advancements and remaining challenges. *Intelligent Service Robotics*, 17(2), 95-118.
- Lee, J., Hanley, D., & Bretl, T. (2022). Extrinsic calibration of multiple inertial sensors from arbitrary trajectories. *IEEE Robotics and Automation Letters*, 7(2), 2055-2062.
- Loianno, G., Brunner, C., McGrath, G., & Kumar, V. (2017). Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU. *IEEE Robotics and Automation Letters*, 2(2), 404–411. IEEE Robotics and Automation Letters. <https://doi.org/10.1109/LRA.2016.2633290>
- Li, F., Xiao, J., Bao, Z., Li, R., & Cai, S. (2021). Research on GPS satellite pseudo-range single point positioning algorithm. *2021 International Conference on Networking, Communications and Information Technology (NetCIT)*, 316–321. <https://doi.org/10.1109/NetCIT54147.2021.00070>
- Li, Y., Ushiku, Y., & Harada, T. (2019, May). Pose graph optimization for unsupervised monocular visual odometry. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 5439 5445). IEEE.
- Li, X., Huang, W., Zhu, X., & Zhao, Z. (2022). MEMS-IMU Error Modelling and Compensation by 3D turntable with temperature chamber. *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–5. <https://doi.org/10.1109/ISNCC55209.2022.9851744>
- Li, X., & Ling, H. (2021). Pogo-net: Pose graph optimization with graph neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5895-5905).
- Li, X., Zhang, P., Huang, G., Zhang, Q., Guo, J., Zhao, Y., & Zhao, Q. (2019). Performance analysis of indoor pseudolite positioning based on the unscented Kalman filter. *GPS Solutions*, 23(3), 79. <https://doi.org/10.1007/s10291-019-0870-y>
- Li, Y., Chen, R., Niu, X., Zhuang, Y., Gao, Z., Hu, X., & El-Sheimy, N. (2020). Inertial sensing meets artificial intelligence: Opportunity or challenge?. *arXiv preprint arXiv:2007.06727*.

- Li, Z., Liu, Z., & Zhao, L. (2021). Improved robust Kalman filter for state model errors in GNSS-PPP/MEMS IMU double state integrated navigation. *Advances in Space Research*, 67(10), 3156–3168. <https://doi.org/10.1016/j.asr.2021.02.010>
- Libero, Y. and I. Klein (2022): A Unified Filter for Fusion of Multiple Inertial Measurement Units. 2022, doi.org/10.48550/arXiv.2208.14524
- Libero, Y. and I. Klein (2024): Augmented Virtual Filter for Multiple IMU Navigation, in *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-12, 2024, Art no. 8501812, doi.org/10.1109/TIM.2024.3370767.
- Lin, X., Bar-Shalom, Y., & Kirubarajan, T. (2004). Exact multisensor dynamic bias estimation with local tracks. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2), 576–590. IEEE Transactions on Aerospace and Electronic Systems. <https://doi.org/10.1109/TAES.2004.1310006>
- Liu, H., X. Wei, M. Perusquía-Hernández, N. Isoyama, H. Uchiyama and K. Kiyokawa (2023). "DUET: Improving Inertial-Based Odometry via Deep IMU Online Calibration," in *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-13, Art no. 2525513, doi: 10.1109/TIM.2023.3306833
- Liu, W. I., & Li, Y. (2019). Error modeling and extrinsic–intrinsic calibration for LiDAR-IMU system based on cone cylinder features. *Robotics and Autonomous Systems*, 114, 124-133.
- Liu, P., Zhao, S., Qing, L., Ma, Y., Lu, B., & Hou, D. (2019). A Calibration Method for DVL Measurement Errors Based on Observability Analysis. *2019 Chinese Control Conference (CCC)*, 3851–3856. <https://doi.org/10.23919/ChiCC.2019.8865920>
- Lotfy, A., Abdelfatah, M. A., & Hosny, H. (2020). Improving the performance of GNSS kinematic point positioning technique by simulated data. *NRIAG Journal of Astronomy and Geophysics*, 9(1), 491–498. <https://doi.org/10.1080/20909977.2020.1783633>
- Lv, Y., Liu, S., Gao, Y., Dai, J., Ren, Z., & Liu, Y. (2024). An Ultra-Wideband Indoor Localization Algorithm with Improved Cubature Kalman Filtering Based on Sigmoid Function. *Applied Sciences*, 14(6), 2239.
- Lv, J., Zuo, X., Hu, K., Xu, J., Huang, G., & Liu, Y. (2022). Observability-aware intrinsic and extrinsic calibration of LiDAR-IMU systems. *IEEE Transactions on Robotics*, 38(6), 3734-3753.

- Maling, D. H. (1992). *Coordinate systems and map projections*. Pergamon.
- Marques Filho, E. A., Neto, A. R., & Kuga, H. K. (2010). A LOW COST INS/GPS NAVIGATION SYSTEM INTEGRATED WITH A MULTILAYER FEED FORWARD NEURAL NETWORK. *Journal of Aerospace Engineering*, 2(2), 26.
- Martin, H. F. S. (2016). Overcoming the challenges of low-cost inertial navigation [Doctoral, UCL (University College London)]. In *Doctoral thesis, UCL (University College London)*. UCL (University College London). <https://discovery.ucl.ac.uk/id/eprint/1524153/>
- Martin, H., Groves, P., & Newman, M. (2016). The Limits of In-Run Calibration of MEMS Inertial Sensors and Sensor Arrays. *NAVIGATION*, 63(2), 127–143. <https://doi.org/10.1002/navi.135>
- Martin, H., Groves, P. D., Newman, M., & Faragher, R. (2013). A New Approach to Better Low-cost MEMS IMU Performance Using Sensor Arrays. 2125–2142. <http://www.ion.org/publications/abstract.cfm?jp=p&articleID=11378>
- Martinec, D., & Pajdla, T. (2007, June). Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1-8). IEEE.
- Maybeck, P. S. (1982). *Stochastic models, estimation, and control*. Academic press.
- Mercurio, M., & Reyes, R. (2024). THE APPLICABILITY OF LOW-COST MULTI-FREQUENCY MULTI-GNSS RECEIVERS FOR GEODETIC CONTROL SURVEYING. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4-W8-2023*, 371–377. ISPRS ICWG IV/III
Philippine Geomatics Symposium (PhilGEOS) 2023 - 6–7 December 2023, Diliman, Quezon City, Philippines. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W8-2023-371> 2024
- Milligan, T. (1999). More applications of Euler rotation angles. *IEEE Antennas and Propagation Magazine*, 41(4), 78–83. IEEE Antennas and Propagation Magazine. <https://doi.org/10.1109/74.789738>
- Mishra, S., Pandey, G., & Saripalli, S. (2021). Target-free Extrinsic Calibration of a 3D-Lidar and an IMU. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (pp. 1-7). IEEE.

- Mirzaei, F. M. (2013). Extrinsic and intrinsic sensor calibration (Doctoral dissertation, University of Minnesota).
- Miyazaki, F., Tomioka, T., Ogawa, J., Ono, D., Masunishi, K., Uchida, K., Murase, H., Ogawa, E., Ishibashi, F., & Tomizawa, Y. (2024). Demonstration of Multi-axis Sensitivity and Temperature Compensation for a Physically Tightly-Coupled MEMS IMU Integrating a Mode-Matched Gyroscope and Two Differential Resonant Accelerometers. *2024 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, 1–4. <https://doi.org/10.1109/INERTIAL60399.2024.10502096>
- Moreira, G., Marques, M., & Costeira, J. P. (2021). Fast pose graph optimization via Krylov-Schur and Cholesky factorization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1898-1906).
- Mráz, E., Trizuljak, A., Rajchl, M., Sedláček, M., Štec, F., Stanko, J., & Rodina, J. (2024). Multi-sensor fusion for robust indoor localization of industrial UAVs using particle filter. *Journal of Electrical Engineering*, 75(4), 304–316. <https://doi.org/10.2478/jee-2024-0037>
- Mukundan, R. (2012). Quaternions. In: *Advanced Methods in Computer Graphics*. Springer, London. https://doi.org/10.1007/978-1-4471-2340-8_5
- Mur-Artal R, Montiel JM, Tardós JD et al (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans Robot* 31(5):1147–1163
- Nassar, S. (2003). *Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications* (pp. 10-32). Calgary, AB, Canada: University of Calgary, Department of Geomatics Engineering.
- Noureldin, A., Karamat, T. B., & Georgy, J. (2012). *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media.
- Olson, E., Leonard, J., & Teller, S. (2006, May). Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. (pp. 2262-2269). IEEE.
- Ovenden, M. W. (1972). The Significance of Newton's Laws of Motion. *Journal of the British Astronomical Association*, Vol. 82, p. 106-111, 82, 106-111.
- Petovello, M. (2015). How does a GNSS receiver estimate velocity? *Inside GNSS*, 14.

- Pietrzykowski, Z., Wołęjsza, P., Nozdrzykowski, Ł., Borkowski, P., Banaś, P., Magaj, J., Chomski, J., Mąka, M., Mielniczuk, S., Pańka, A., Hatłas-Sowińska, P., Kulbiej, E., & Nozdrzykowska, M. (2022). The autonomous navigation system of a sea-going vessel. *Ocean Engineering*, 261, 112104. <https://doi.org/10.1016/j.oceaneng.2022.112104>
- Pizzinga, A. (2010). Constrained Kalman Filtering: Additional Results. *International Statistical Review*, 78(2), 189–208. <https://doi.org/10.1111/j.1751-5823.2010.00098.x>
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2002). *Numerical recipes in C: The Art of Scientific Computing* (2nd ed.). Cambridge University Press.
- Prikhodko, I. P., Bearss, B., Merritt, C., Bergeron, J., & Blackmer, C. (2018). Towards self-navigating cars using MEMS IMU: Challenges and opportunities. *2018 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, 1–4. <https://doi.org/10.1109/ISISS.2018.8358141>
- Qi, H., & Moore, J. B. (2002). Direct Kalman filtering approach for GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2), 687–693.
- Qian, K. (2017). *Generic Multisensor Integration Strategy and Innovative Error Analysis for Integrated Navigation*. <http://hdl.handle.net/10315/33627>
- Qian, K., Wang, J. G., & Hu, B. (2015). Novel integration strategy for GNSS-aided inertial integrated navigation. *Geomatica*, 69(2), 217–230.
- Quinchia, A. G., G. Falco, E. Falletti, F. Dovis, and C. Ferrer (2013). “A Comparison between Different Error Modeling of MEMS Applied to GPS/INS Integrated Systems.” *Sensors* 13, no. 8. doi.org/10.3390/s130809549.
- Rahman, F., Uwineza, J.-B., & Farrell, J. A. (2020). *ECEF Position Accuracy and Reliability: Inertial Navigation with GNSS Precise Point Positioning (PPP)*. <https://escholarship.org/uc/item/4km0d6pz>
- Ramanandan, A., Chen, A., & Farrell, J. A. (2011). Inertial navigation aiding by stationary updates. *IEEE Transactions on Intelligent Transportation Systems*, 13(1), 235–248.
- Rao, B. R. (2013). *GPS/GNSS Antennas*. Artech House.

- Rhee, I., Abdel-Hafez, M. F., & Speyer, J. L. (2004). Observability of an integrated GPS/INS during maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2), 526–535. *IEEE Transactions on Aerospace and Electronic Systems*. <https://doi.org/10.1109/TAES.2004.1310002>
- Rogers, R. M. (2003). *Applied mathematics in integrated navigation systems* (Vol. 1). AIAA.
- Rosales, J. J., & Colomina, I. (2005). A flexible approach for the numerical solution of the INS mechanization equations. *Proceedings of the 6th Geomatic week, Barcelona*.
- Rouzaud, D., & Skaloud, J. (2011). Rigorous Integration of Inertial Navigation with Optical Sensors by Dynamic Networks. *NAVIGATION*, 58(2), 141–152. <https://doi.org/10.1002/j.2161-4296.2011.tb01797.x>
- Sahawneh, L. R., Al-Jarrah, M. A., Assaleh, K., & Abdel-Hafez, M. F. (2010). Real-Time Implementation of GPS Aided Low-Cost Strapdown Inertial Navigation System. *Journal of Intelligent & Robotic Systems*, 61(1–4), 527–544. <https://doi.org/10.1007/s10846-010-9501-0>
- Salychev, O. S. (1998) “Inertial Systems in Navigation and Geophysics” *Bauman MSTU Press*. pp. 11-98, 118-132, 243-266.
- Salytcheva, A. O. (2004). Medium accuracy INS/GPS integration in various GPS environments (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>. doi:10.11575/PRISM/22042
- Scherzinger, B. M. (2000, September). Precise robust positioning with inertial/GPS RTK. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)* (pp. 155-162).
- Schmidt, S. F., Bjorkman, W. S., & Conrad, B. (1973). *New mechanization equations for aided inertial navigation systems* (No. NASA-CR-2352). NASA.
- Schmidt, D. K., & Raney, D. L. (2012). Modeling and Simulation of Flexible Flight Vehicles. *Journal of Guidance, Control, and Dynamics*. <https://doi.org/10.2514/2.4744>
- Searle, S. R. (1995). An overview of variance component estimation. *Metrika*, 42(1), 215-230.
- Seeber, G. (2003). Satellite Geodesy, Walter de Gruyter. *Berlin Shen Y, Chen Q, Hsu H, Zhang X, Lou L (2013) A modified short arc approach for recovering gravity eld model.*

- Shahri, A. M., & Rasoulzadeh, R. (2017). Implementation of a Low- Cost Multi- IMU by Using Information Form of a Steady State Kalman Filter. *AUT Journal of Electrical Engineering*, 49(2), 195–204. doi.org/10.22060/eej.2017.12045.5028
- Sharp, G. C., Lee, S. W., & Wehe, D. K. (2004). Multiview registration of 3D scenes by minimizing error between coordinate frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8), 1037-1050.
- Shen, Y., Hwang, B., & Jeong, J. P. (2020). Particle Filtering-Based Indoor Positioning System for Beacon Tag Tracking. *IEEE Access*, 8, 226445–226460. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3045610
- Shen, Q., Chang, H., Wu, Y., & Xie, J. (2019). Turn-on bias behavior prediction for micromachined Coriolis vibratory gyroscopes. *Measurement*, 131, 380-393.
- Shi, L.-F., Feng, B.-L., Dai, Y.-F., Liu, G.-X., & Shi, Y. (2023). Pedestrian Indoor Localization Method Based on Integrated Particle Filter. *IEEE Transactions on Instrumentation and Measurement*, 72, 1–10. IEEE Transactions on Instrumentation and Measurement. https://doi.org/10.1109/TIM.2023.3235426
- Sibai, F. N., Trigui, H., Zanini, P. C., & Al-Odail, A. R. (2012). Evaluation of indoor mobile robot localization techniques. 2012 International Conference on Computer Systems and Industrial Informatics. doi:10.1109/iccsii.2012.6454560
- Simon, D. (2006). Optimal State Estimation. *John Wiley & Sons*, 2006.
- Simon, D., & Chia, T. L. (2002). Kalman filtering with state equality constraints. *IEEE transactions on Aerospace and Electronic Systems*, 38(1), 128-136.
- Simon, D. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8), 1303-1318.
- Sirtkaya, S., B. Seymen and A. A. Alatan, "Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation," *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 2013, pp. 219-226.
- Souza, E. M. D., Monico, J. F. G., & Pagamisse, A. (2009). GPS satellite kinematic relative positioning: analyzing and improving the functional mathematical model using wavelets. *Mathematical Problems in Engineering*, 2009(1), 934524.

- Spencer, A. J. M. (2004). *Continuum mechanics*. Courier Corporation.
- Sun, W., Zhao, J., Ding, W., & Sun, P. (2023). Robust UKF Relative Positioning Approach for Tightly Coupled Vehicle Ad Hoc Networks Based on Adaptive M-Estimation. *IEEE Sensors Journal*, 23(9), 99599971. IEEE Sensors Journal. <https://doi.org/10.1109/JSEN.2023.3262656>
- Suzuki, T. (2016). *Integration of GNSS Positioning and 3D Map using Particle Filter*. 1296–1304. <https://doi.org/10.33012/2016.14857>
- Tanigawa M., H. Luinge, L. Schipper and P. Slycke, "Drift-free dynamic height sensor using MEMS IMU aided by MEMS pressure sensor," *2008 5th Workshop on Positioning, Navigation and Communication*, Hannover, Germany, 2008, pp. 191-196, doi: 10.1109/WPNC.2008.4510374.
- Tan, T. N., Khenchaf, A., Comblet, F., Franck, P., Champeyroux, J. M., & Reichert, O. (2020). Robust extended kalman filter and long short-term memory combination to enhance the quality of single point positioning. *Applied Sciences*, 10(12), 4335.
- Tan, T. D., Ha, L. M., Long, N. T., Tue, H. H., & Thuy, N. P. (2007, October). Feedforward structure of Kalman filters for low cost navigation. In *International Symposium on Electrical-Electronics Engineering (ISEE2007)* (pp. 1-6).
- Tao, B., Qiu, W., & Yao, Y. (Eds.). (2019). *Error theory and foundation of surveying adjustment*. Wuhan University Press.
- Teskey, B. (1979). *Geodetic Aspects of Engineering Surveys Requiring High Accuracy*. University of New Brunswick. <https://unbscholar.lib.unb.ca/bitstreams/c5f9d47d-6428-4c6f-8a14-eee2591c776a/download>
- Teunissen, P. J. G., De Jonge, P. J., & Tiberius, C. C. J. M. (1997). Performance of the LAMBDA Method for Fast GPS Ambiguity Resolution. *NAVIGATION*, 44(3), 373–383. <https://doi.org/10.1002/j.2161-4296.1997.tb02355.x>
- Teunissen, P. J. G., Joosten, P., & Odijk, D. (1999). The Reliability of GPS Ambiguity Resolution. *GPS Solutions*, 2(3), 63–69. <https://doi.org/10.1007/PL00012758>
- Teunissen, P. J. G., & Amiri-Simkooei, A. R. (2007). Least-squares variance component estimation. *Journal of Geodesy*, 82(2), 65–82. doi:10.1007/s00190-007-0157-x

- Thomas, N., Cruickshank, D. G. M., & Laurensen, D. I. (2000, September). A robust location estimator architecture with biased Kalman filtering of TOA data for wireless systems. In *2000 IEEE Sixth International Symposium on Spread Spectrum Techniques and Applications. ISSTA 2000*.
- Thrun, S., & Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6), 403-429.
- Titterton, D., J. Weston, 2004. 'Strapdown Inertial Navigation Technology' (Radar, Sonar and Navigation) <https://digital-library.theiet.org/content/books/ra/pbra017e>
- Trawny, N., & Roumeliotis, S. I. (2005). Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep, 2*, 2005.
- Turner, L., & Sherlock, C. (2013). An introduction to particle filtering. *Lancaster University, Lancaster*.
- Turkoglu, M. O., Brachmann, E., Schindler, K., Brostow, G. J., & Monszpart, A. (2021, December). Visual camera re-localization using graph neural networks and relative pose supervision. In *2021 International Conference on 3D Vision (3DV)* (pp. 145-155). IEEE.
- Vanicek, P., & Krakiwsky, E. (1986). *Geodesy: The Concepts*. North-Holland.
- Van Sickle, J. (2023). *GPS and GNSS for Land Surveyors*. CRC Press.
- VectorNav. (2025). *Introduction to Inertial Sensors*. <https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-inertial>
- Voight, J. (2021). *Quaternion algebras*. Springer International Publishing.
- Waegli, Adrian, et al. (2008): "Redundant MEMS-IMU Integrated with GPS for Performance Assessment in Sports." *2008 IEEE/ION Position, Location and Navigation Symposium*, IEEE, 2008, pp. 1260-1268. doi.org/10.1109/PLANS.2008.4570079.
- Wagner, J. F. (2004). Adapting the principle of integrated navigation systems to measuring the motion of rigid multibody systems. *Multibody System Dynamics*, 11, 87-110.
- Wagner, J. F., & Wieneke, T. (2003). Integrating satellite and inertial navigation—Conventional and new fusion approaches. *Control Engineering Practice*, 11(5), 543–550. [https://doi.org/10.1016/S0967-0661\(02\)00043-6](https://doi.org/10.1016/S0967-0661(02)00043-6)
- Wang, J. (1999). Stochastic Modeling for Real-Time Kinematic GPS/GLONASS Positioning. *Navigation*, 46(4), 297-305.

- Wang, J. (1997): Filtering Methods for Error Tolerant Kinematic Positioning, PhD Diss., No. 52, Schriftenreihe Studiengang Vermessungswesen der Universität der Bundeswehr München, ISSN 0173 1009, Neubiberg, Germany, 1997.
- Wang, J., Gopaul, S. N., & Guo, J. (2010, August). Adaptive Kalman filtering based on posteriori variance covariance components estimation. In *Proceeding of the CPGPS Technical Forum Satellite Navigation and Positioning, Shanghai, China* (Vol. 1720, p. 111).
- Wang, J. G., Qian, K., & Hu, B. (2015). An unconventional full tightly-coupled multi-sensor integration for kinematic positioning and navigation. In *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III* (pp. 753-765). Springer Berlin Heidelberg.
- Wang, J., Boda, A., & Hu, B. (2021). Comprehensive error analysis beyond system innovations in Kalman filtering. In *Learning Control* (pp. 59-92). Elsevier.
- Wang, J.; Brunson, B. and Hu, B. (2022): Innovative Formulation in Discrete Kalman Filtering with Constraints – A Generic Framework for Comprehensive Error Analysis, *Journal of Global Positioning Systems*, vol. 18, issue 1, 56–69.
- Wang, L., Tang, H., Zhang, T., Chen, Q., Shi, J., & Niu, X. (2021). Improving the Navigation Performance of the MEMS IMU Array by Precise Calibration. *IEEE Sensors Journal*, 21(22), 26050–26058. IEEE Sensors Journal. <https://doi.org/10.1109/JSEN.2021.3118455>
- Wang, P., Liu, C., Zhang, J., Wang, L., & Zhang, M. (2024). DS-UKF-Based Positioning Method for Intelligent Connected Vehicles in Urban Intersection Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 25(6), 6118–6132. IEEE Transactions on Intelligent Transportation Systems. <https://doi.org/10.1109/TITS.2023.3336770>
- Wendel, J., Schlaile, C., & Trommer, G. F. (2001, June). Direct Kalman filtering of GPS/INS for aerospace applications. In *International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS2001)*.
- Wu, Y., Wang, P., & Hu, X. (2003). Algorithm of Earth-centered Earth-fixed coordinates to geodetic coordinates. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4), 1457–1461. IEEE Transactions on Aerospace and Electronic Systems. <https://doi.org/10.1109/TAES.2003.1261144>

- Xu, P., Shi, C., & Liu, J. (2012). Integer estimation methods for GPS ambiguity resolution: An applications-oriented review and improvement. *Survey Review*, 44(324), 59–71. <https://doi.org/10.1179/1752270611Y.0000000004>
- Yang, Y., He, H., & Xu, G. (2001). Adaptively robust filtering for kinematic geodetic positioning. *Journal of Geodesy*, 75, 109-116.
- Yang, Y., Geneva, P., Zuo, X., & Huang, G. (2020). Online imu intrinsic calibration: Is it necessary?. 2020 Robotics: Science and Systems.
- Yang, K., Bar-Shalom, Y., Willett, P., & Inou, H. (2020, December 1). *Bias Estimation for Collocated Sensors: Model Identification and Measurement Fusion*. | *Journal of Advances in Information Fusion* | EBSCOhost. <https://openurl.ebsco.com/contentitem/gcd:148726319?sid=ebsco:plink:crawler&id=ebsco:gcd:148726319>
- Yang, C., & Blasch, E. (2009). Kalman filtering with nonlinear state constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 45(1), 70-84.
- Yang, Z., van Beijnum, B.-J. F., Li, B., Yan, S., & Veltink, P. H. (2020). Estimation of Relative Hand-Finger Orientation Using a Small IMU Configuration. *Sensors*, 20(14), Article 14. <https://doi.org/10.3390/s20144008>
- Yang, Z., & Shen, S. (2016). Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1), 39-51.
- Yang, Y., & Xu, T. (2003). An Adaptive Kalman Filter Based on Sage Windowing Weights and Variance Components. *Journal of Navigation*, 56(2), 231–240. doi:10.1017/s0373463303002248
- Yi, D., S. Yang, and S. Bisnath (2022). Native Smartphone Single- and Dual-Frequency GNSS PPP/IMU Solution in Real-World Driving Scenarios. *Remote Sensing* 14, no. 14. doi.org/10.3390/rs14143286.
- Yi, D., Bisnath, S., Naciri, N., & Vana, S. (2021). Effects of ionospheric constraints in Precise Point Positioning processing of geodetic, low-cost and smartphone GNSS measurements. *Measurement*, 183, 109887. <https://doi.org/10.1016/j.measurement.2021.109887>

- Yokozuka M, Koide K, Oishi S et al (2020) LiTAMIN: LiDAR-based tracking and mapping by stabilized ICP for geometry approximation with normal distributions. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 5143–5150
- Yu, F., Zhu, M., Xiao, S., & Wang, J. (2018, April). A designed AKF algorithm applied to unconventional GPS and multiple low-cost IMUs integration strategy. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)* (pp. 708-714). IEEE.
- Yu, F., Zhu, M., Xiao, S., & Wang, J. (2018). Practical Mechanisms to Realize Smooth Transitions for Unconventional Multi-sensor Integrated Kinematic Positioning and Navigation. *Engineering Letters*, 26(3).
- Zhang J, Singh S (2014) LOAM: LiDAR odometry and mapping in real-time. In: Robotics: science and systems. Berkeley, CA, pp 1–9
- Zhao, Long (2016): Modern Navigation Technologies, Lecture Slides, Graduate Course of Beijing University of Aeronautics and Astronautics, accessed in April 2023, dnc.buaa.edu.cn/info/1051/1433.htm.
- Zhong, Q., & Groves, P. D. (2022). Multi-epoch 3D-mapping-aided positioning using Bayesian filtering techniques. *NAVIGATION: Journal of the Institute of Navigation*, 69(2). <https://navi.ion.org/content/69/2/navi.515.abstract>
- Zhong, Q., & Groves, D. (2022). *Outlier detection for 3D-mapping-aided GNSS positioning*. <https://discovery.ucl.ac.uk/id/eprint/10157939/>