
Masters Theses

Student Theses and Dissertations

Spring 2016

An IMU-based spacecraft navigation architecture using a robust multi-sensor fault detection scheme

Samuel J. Haberberger

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses



Part of the [Aerospace Engineering Commons](#)

Department:

Recommended Citation

Haberberger, Samuel J., "An IMU-based spacecraft navigation architecture using a robust multi-sensor fault detection scheme" (2016). *Masters Theses*. 7505.

https://scholarsmine.mst.edu/masters_theses/7505

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

AN IMU-BASED SPACECRAFT NAVIGATION ARCHITECTURE USING A
ROBUST MULTI-SENSOR FAULT DETECTION SCHEME

by

SAMUEL J. HABERBERGER

A THESIS

Presented to the Faculty of the Graduate School of the
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

2016

Approved by

Kyle DeMars, Advisor

Henry Pernicka

Serhat Hosder

Copyright 2016
SAMUEL J. HABERBERGER
All Rights Reserved

ABSTRACT

Redundant sensor networks of inertial measurement units (IMUs) provide inherent robustness and redundancy to a navigation solution obtained by dead reckoning the fused accelerations and angular velocities sensed by the IMU. However, IMUs have been known to experience faults risking catastrophic mission failure creating large financial setbacks and an increased risk of human safety. Different fusion methods are analyzed for a multi-sensor network using cost effective IMUs, including direct averaging and covariance intersection. Simulations of a spacecraft in low Earth orbit are used to baseline a typical expensive IMU and compare the navigation solution obtained from a network of several low-cost IMUs from fused data. Robust on-board fault detection schemes are developed and analyzed for a multi-sensor distributed network specifically for IMUs.

Simulations of a spacecraft are used to baseline several cases of sensor failure in a distributed network undergoing fusion to produce an accurate navigation solution. The presented results exhibit a robust fault identification scheme that successfully removes a failing sensor from the fusion process while maintaining accurate navigation solutions. In the event of a temporary sensor failure, the fault detection algorithm recognizes the sensors' return to nominal operating conditions and processes its sensor data accordingly.

ACKNOWLEDGMENTS

Firstly, I'd like to thank my advisor Dr. Kyle DeMars. You have given me the opportunity to envelop myself in the aerospace field of research, leaving me with a passion for navigation and estimation. You've held me to the highest expectations embedding me with a proud work ethic. Thank you for all of the priceless experience you have given me and for being a great advisor. I couldn't have made it to this point in my academic career without your guidance and knowledge.

Secondly I'd like to thank my committee members Dr. Hank Pernicka and Dr. Serhat Hosder. As an undergraduate student, you have both instilled invaluable knowledge to me along with a contagious enthusiasm for academia via your classes and the satellite research team. My desire and aspirations of pursuing research in this field are greatly accredited to both of you.

My deepest appreciation goes to my loving and supportive family. Without you I couldn't have made it to this point in my life and academic career. You have raised me with a work ethic that has given me the persistence, and will power needed in the engineering field. You will never know how truly thankful I am for all of the opportunities that you have provided for me.

Lastly, I'd like to thank my friends, classmates and academic peers James McCabe, Matthew Gualdoni, Jacob Darling, Levi Mallot, Matthew Glascock, John Schaefer, Keith LeGrand and the AREUS lab entity. I can't thank you all enough for the knowledge and technical advice that has been provided to me from you. I couldn't have asked for a better group of friends and academic peers to work with. It's truly been a pleasure working and going to school with all of you.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGMENTS	iv
LIST OF ILLUSTRATIONS	ix
LIST OF TABLES	xi
NOMENCLATURE	xii
 SECTION	
1 INTRODUCTION	1
1.1 MOTIVATION	3
1.2 OVERVIEW	5
2 SPACECRAFT DYNAMICS	7
2.1 IMU MODELING	7
2.2 CONTINUOUS TIME DYNAMICS	10
2.3 IMU INVERSION	11
2.4 MULTI-SENSOR TRANSLATIONAL COMPENSATION	13
2.5 DISCRETIZED DEAD-RECKONING EQUATIONS	14
2.6 MEAN AND COVARIANCE PROPAGATION	15
3 FUSION METHODOLOGY	18
3.1 DIRECT AVERAGING	18
3.2 COVARIANCE INTERSECTION	20
3.2.1 Implementation	24

3.2.2	Generalized Variance Weighting Selection	25
3.3	FAULT TOLERANT FUSION	26
3.4	FUSION EXAMPLE	28
3.5	FUSION METHODOLOGY SUMMARY AND CONCLUSION	31
4	FAULT DETECTION	34
4.1	PRINCIPAL COMPONENT ANALYSIS	34
4.2	FAULT DETECTION EXAMPLE	37
4.3	MODIFIED PRINCIPAL COMPONENT ANALYSIS	40
4.4	TRAINING VECTOR FAILURE	43
4.4.1	Covariance Intersection in the Feature Plane	43
4.4.2	Kullback-Leibler Divergence Covariance Threshold	45
4.4.3	Shannon Entropy Threshold	46
5	MEASUREMENT MODELING	48
5.1	POSITION MEASUREMENT MODELING	48
5.2	QUATERNION MEASUREMENT MODELING	50
5.3	RANGE MODELING	51
5.4	RANGE RATE MODELING	53
5.5	UNIT VECTOR STAR CAMERA MODELING	57
6	NAVIGATION ALGORITHM	60
6.1	THE DISCRETE EXTENDED KALMAN FILTER	60
6.1.1	Mean and Covariance Propagation	60
6.1.2	Mean and Covariance Update	64
6.1.3	Attitude Update	71

6.1.4	Extended Kalman Filter Summary	72
6.2	STATE ESTIMATE AND STATE ESTIMATION ERROR CO- VARIANCE PROPAGATION	73
6.2.1	Position and Velocity Error Covariance	74
6.2.2	Attitude Error Covariance	79
6.2.3	Error Covariance in a Fusion Network	81
6.3	MEASUREMENT PROCESSING	83
6.3.1	Position Measurement	83
6.3.2	Quaternion Measurement	84
6.3.3	Range and Range Rate	85
6.3.4	Unit Vector Star Camera	86
7	NAVIGATION/FAULT DETECTION SYSTEM ARCHITECTURE	88
7.1	SYSTEM IMPLEMENTATION	88
7.1.1	Fault Detection Architecture	88
7.1.2	Fusion Architecture	90
7.1.3	Navigation Architecture	90
7.2	SENSOR CONFIGURATION	91
8	SIMULATION RESULTS	93
8.1	SIMULATION CONFIGURATION	93
8.2	CASE 1: NOMINAL SENSOR OPERATION	97
8.3	CASE 2: SINGLE SENSOR FAILURE	104
8.4	CASE 3: MULTIPLE SENSOR FAILURES	109
8.5	CASE 4: TRAINING VECTOR SENSOR FAILURE	111

9	CONCLUSIONS	118
9.1	FUTURE CONSIDERATIONS.	120
APPENDICES		
A	IMU SPECIFICATIONS	121
B	FUSION ALGORITHM.	123
C	MATRIX DEFINITIONS	125
D	MONTE CARLO ANALYSIS	131
E	EKF CONSIDERATION: UNDERWEIGHTING	134
F	ERROR PARAMETER SIMULATION RESULTS	137
	BIBLIOGRAPHY	142
	VITA	144

LIST OF ILLUSTRATIONS

Figure	Page
1.1 x Position Standard Deviation versus Time with Plotted Errors.	4
2.1 IMU Errors vs. Time	9
3.1 IMU Data Fusion Computed by Directly Averaging the Measurements. . .	19
3.2 Direct Averaging Fusion ($N = 10, 100$ and 1000 Respectively)	19
3.3 IMU Data Fusion Computed by Covariance Intersection	24
3.4 CI Fusion Rule	25
3.5 Position [m] and Attitude [deg] RSS Error	30
3.6 $x - y$ Uncertainty Contours [m] at 50, 200, 300 [s], Respectively	31
3.7 Fusion Rule Uncertainty Comparison	32
4.1 Effects of PCA Undergoing Single Sensor Failure for Static and Dynamic Systems	38
4.2 First and Second Principal Components versus Time for a Static and Dynamic System Undergoing a Single Sensor Failure	39
4.3 First and Second Principal Components Plotted Over 3-Dimensional Zero- Mean Data Set Before and After Failure	40
7.1 Implementation Architecture Overview	88
7.2 Fault Detection in a Static System.	89
7.3 Fault Detection in a Dynamic System.	90
7.4 IMU Data Fusion Computed by Directly Averaging the Measurements. . .	91
7.5 Navigation Architecture	91
8.1 True Simulated Orbit Trajectory.	94
8.2 True Simulated \mathbf{a}_{ng}^i and $\boldsymbol{\omega}_{i/b}^i$	96

8.3	Time of Received Measurements	96
8.4	Case 1: Position Standard Deviation/Errors and RSS vs. Time	100
8.5	Case 1: Position Standard Deviation/Errors and RSS vs. Time	101
8.6	Case 1: Attitude Standard Deviation/Errors and RSS vs. Time	102
8.7	Case 1: Effects of MPCA with no Failures	103
8.8	Case 1: Feature Plane CI Training Vector Fault Detection	103
8.9	Case 2: Position Standard Deviation/Errors and RSS vs. Time	105
8.10	Case 2: Velocity Standard Deviation/Errors and RSS vs. Time	106
8.11	Case 2: Attitude Standard Deviation/Errors and RSS vs. Time	107
8.12	Case 2: Effects of MPCA with a Single IMU Failure	108
8.13	Case 2: Feature Plane CI Training Vector Fault Detection	108
8.14	Case 3: Effects of MPCA with Multiple IMU Failures	110
8.15	Case 3: Feature Plane CI Training Vector Fault Detection	110
8.16	Case 4: Training Vector Failure using Only MPCA	111
8.17	Case 4: Position Standard Deviation/Errors and RSS vs. Time	113
8.18	Case 4: Velocity Standard Deviation/Errors and RSS vs. Time	114
8.19	Case 4: Attitude Standard Deviation/Errors and RSS vs. Time	115
8.20	Case 4: Effects of MPCA with a Training Vector Failure	116
8.21	Case 4: Training Vector Fault Detection	117

LIST OF TABLES

Table	Page
3.1 Spacecraft IMU Configuration	29
3.2 Fusion Example Legend	29
8.1 Results Overview	97

NOMENCLATURE

$\mathbf{a}_{m,k}$	Measured non-gravitational acceleration outputted from the accelerometer
$\boldsymbol{\omega}_{m,k}$	Measured angular velocity outputted from the gyroscope
$\hat{\mathbf{n}}$	Estimated vector denoted by the hat
Δt	Time step between measurements
$\Delta \hat{\mathbf{v}}$	Measured non-gravitational acceleration multiplied by Δt
$\Delta \hat{\boldsymbol{\theta}}$	Measured angular velocity multiplied by Δt
$\bar{\mathbf{n}}$	Right handed, vector-first quaternion form
$\hat{\mathbf{r}}_k$	Position estimate
$\hat{\mathbf{v}}_k$	velocity estimate
$\hat{\mathbf{q}}_k$	Quaternion estimate
\mathbf{x}_k	State of the system
\mathbf{m}_k	Mean at some time k
\mathbf{P}_k	Covariance at some time k
$\boldsymbol{\sigma}_k$	Standard deviation at some time k
\mathbf{T}_b^i	Body to inertial attitude matrix
$\tilde{\mathbf{n}}$	Fused solution
N	Number of IMUs in system configuration
$p_g(\mathbf{x}; \mathbf{m}, \mathbf{P})$	Gaussian distribution in \mathbf{x} with a given mean and covariance
Φ	Feature plane mapping matrix
τ	Fault alarm threshold
\mathbf{y}	Principal components
\mathbf{x}_T	Training vector
Θ	Parity space mapping matrix
\mathbf{w}_k	IMU measurement noise at some time k

\mathbf{v}_k	External sensor noise at some time k
$\delta \mathbf{n}$	state or measurement deviation
\mathbf{z}_k	Received measurement at some time k
$\hat{\mathbf{z}}_k$	Estimated measurements as a function of the mean at some time k
\mathbf{F}	Dynamics Jacobian
\mathbf{M}	Mapping matrix that maps the noise into the system dynamics
\mathbf{Q}	Process noise covariance matrix
\mathbf{e}_k	State estimation error at some time k
\mathbf{m}_k^-	Predicted mean at some time k
\mathbf{m}_k^+	Corrected mean at some time k
\mathbf{P}_k^-	Predicted covariance at some time k
\mathbf{P}_k^+	Corrected covariance at some time k
$\mathbf{h}(\mathbf{m}_k^+)$	Estimated measurement as a function of the predicted mean
\mathbf{W}_k	Innovations or measurement residual covariance
\mathbf{C}_k	Cross covariance
\mathbf{K}_k	The Kalman gain
$\mathbf{H}(\mathbf{m}_k^-)$	Measurement Jacobian as a function of the predicted mean

1. INTRODUCTION

Spacecraft inertial measurement unit (IMU)-based navigation typically relies on a single, high-reliability, high-cost, tactical or strategic grade IMU to achieve accurate navigation solutions. In order to reduce cost while retaining accuracy of the navigation solution and improving the overall fault tolerance of the system, multiple lower-cost IMUs can be used. In order to combine the data from these multiple IMUs, several data fusion rules have been proposed. A direct averaging fusion rule of data from multiple IMUs was used in the demonstration and development of a micro-electrical-mechanical system (MEMS) IMU cluster [1]. Data fusion for IMUs in a decentralized distributed system has also been studied for enhanced pedestrian navigation, in which the direct averaging, the Centralized Filter, and the Federated Filter fusion rules [2] were used. The covariance intersection (CI) algorithm [3–5], has been cast under the more general logarithmic opinion pool framework and applied to the tracking of a space object using multiple ground-based optical sensors [5]. In order to improve upon the performance of the Federated Filter fusion rule, which equally weights each input solution to be fused, an intelligent weight selection scheme for the CI fusion rule is proposed. The navigation solutions input to the CI fusion rule with lower uncertainty are accepted with greater confidence than those with higher uncertainty. In order to evaluate the CI fusion rule with these intelligently selected weights, a simulation is constructed and analyzed in which this rule is compared to the direct averaging fusion rule for a distributed network of IMUs. This analysis determines the fusion rule considered in the analysis and construction of a fault detection algorithm for a cluster of IMUs.

In order to improve the capabilities of financially limited spacecraft, low-cost, high-performance, fault tolerant navigation systems are needed. Data fusion in a distributed network of low-cost IMUs can provide navigation system performance

comparable to those with a single, high-cost, tactical, or strategic grade IMU. The proposed fusion rules perform differently under certain circumstances; therefore, this thesis study investigates how intelligently selected weights for the CI fusion rule compares to the direct average fusion rule for the case when the system is operating nominally and the case when an IMU failure is present. The direct averaging fusion rule does not adapt for measurement degradation of each individual sensor; thus, it is expected that the CI fusion rule will be preferred to the direct averaging fusion rule when a sensor failure is present. This expectation can only hold if the CI fusion rule can down-weight the navigation solution of a degraded sensor producing off-nominal data. In order to perform the CI fusion rule, each IMU is used to propagate an independent navigation solution, and then the navigation solutions are fused together using weights that are selected to reflect the confidence in each navigation solution. The direct averaging rule, on the other hand, performs a simple average of the measurements from each IMU and propagates a single navigation solution forward in time.

Typical navigation solutions are comprised of two parts: determining the mean and determining the covariance. Inherently, covariance evolution is a function of the sensor error specifications that are provided by a data sheet or from *a priori* testing analysis. Therefore, with the proposed multi-sensor navigation architecture, a method of on-board fault detection is needed inside the navigation system. Fault detection algorithms must be constructed to ensure a robust fault tolerant distributed system of multiple redundant sensors, specifically IMUs in this case. One fault detection method that was examined in this thesis study is Principal Component Analysis (PCA), first invented by Karl Pearson [6]. PCA is an analytic, statistical procedure that orthogonally transforms correlated observations into a set of linearly uncorrelated values, known as principal components. This allows for the construction of

uncorrelated data patterns and data trends that can be used to identify a set of data that is not uniform with respect to the rest of the data sets.

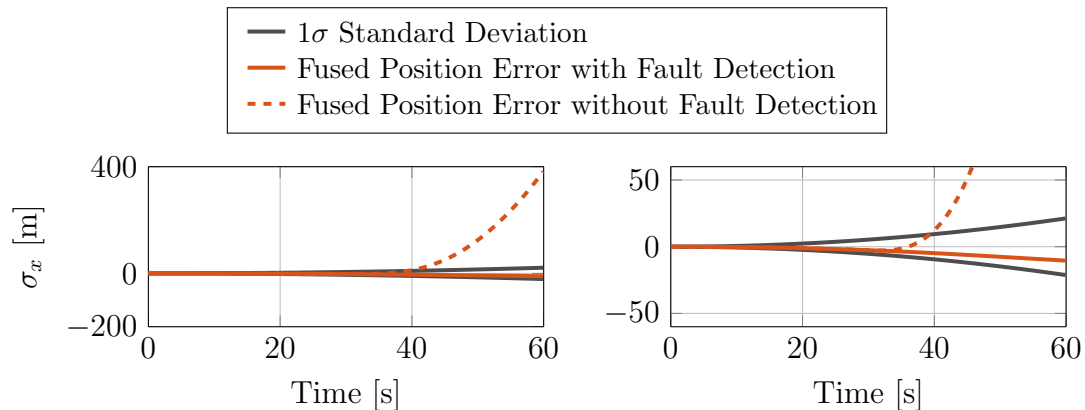
The issue at hand with the standard PCA method is that when a fault occurs in a dynamic environment, PCA cannot classify fault patterns in the sensor data due to similar patterns created by sensor movement. In order to classify faults in a dynamical system, a Modified Principal Component Analysis (MPCA) approach, proposed by Potter shown in Reference [7], is independent of sensor movement is considered. The MPCA approach introduces a null space matrix to compute a parity vector, which is a transformation into a data feature plane that can be used in testing for patterns in the comparable data. From Reference [8], this matrix cancels the dynamical movement on the PCA input by algebraic manipulation in which the procedure calculates the parity vector and then generates a fault pattern separate from the sensor data output. Using this modification of PCA, dynamic sensors, such as IMUs, can be configured and fused in a distributed network undergoing fault detection while using MPCA. This allows for a the construction of a robust fault tolerant algorithm that is capable of producing confident navigation solutions. This thesis provides an extension to the previous work found in Reference [9].

1.1. MOTIVATION

Data fusion, in parallel with fault detection, in a distributed network of low-cost IMUs can provide navigation system performance comparable to those with a single, high-cost, tactical, or strategic grade IMU, which is ideal for aerospace industries and even universities. Current budget limitations in aerospace industries provide the need for low cost multi-sensor data fusion prioritizing funding and feasibility on high-fidelity IMU-based mission payloads. In recent years, the industry's interest in small satellites has grown to become a large area of research. This includes university collaborations, which are directed toward making small spacecraft into powerful but

financially feasible space operations. Along with ameliorating budget limitations and advancing manned space flight missions in the near future, sensor redundancy and the detection of sensor failures are vital to mission success. While this current research is applied specifically to IMUs, the algorithms studied and implemented herein are designed to be robust and applicable to a wide variety of sensors on dynamic vehicles.

A motivating example is considered to demonstrate the need for on-board fault detection in a multi-sensor fusion network of IMUs. Suppose an arbitrary spacecraft with ten IMUs on board is undergoing a fusion process that feeds into the navigation solution. The motivation and importance of fault detection in a redundant sensor network is found in comparing the navigation solution of a system with and without on-board fault detection. A single sensor in the fusion network undergoes a failure halfway through the simulation characterized by an increase in bias and noise. Figure 1.1 shows the position standard deviation (1σ) along with the estimated position error with and without fault detection on-board along with a zoomed in view to better inspect the navigation solution.



(a) Position Standard Deviation

(b) Zoom in Position Standard Deviation

Figure 1.1. x Position Standard Deviation versus Time with Plotted Errors

It is clearly seen that even with a single IMU failure that the inability to detect a sensor fault can lead to large errors very quickly. Along with producing a poor estimate, the propagated uncertainty no longer accurately represents the true uncertainty of the system due to the uncertainty model being based off of set sensor specifications. These results clearly show the importance of fault detection on-board a distributed sensor network. For brevity, the velocity and attitude standard deviation/error plots are omitted but also show very similar results.

1.2. OVERVIEW

The current thesis has two main points of focus but contains necessary preliminary information about the system. In order to focus on data fusion and fault detection on-board a spacecraft, the governing spacecraft dynamics must be discussed in conjunction with a proposed IMU model defining non-negligible sensor errors. A brief discussion on mean and covariance propagation is then considered. The full derivation and explanation of mean and covariance is discussed in full detail later in the thesis, but is a necessary consideration in fusion methodology which is introduced directly after.

The first point of focus is a comparison and performance analysis of multiple data fusion rules applied to IMUs, i.e. direct averaging and Covariance Intersection (CI). This trade-study examines system implementation and a weighting selection enabling a more robust fusion rule method. In this comparison, it was deemed that direct averaging produces a more accurate navigation solution, i.e. mean and covariance, than that of CI. However, in the non-negligible consideration of a failing sensor in the system, with an intelligent weighting solution CI could effectively down-weight the failing sensor via *a priori* knowledge of a faulty sensor. While CI may seem more robust, it is not practical to assume that CI could obtain this knowledge of sensor degradation. A conclusion is made deeming direct averaging the appropriate fusion

rule to use for simulation with the assumption of an inherent fault detection method implemented.

The last main point of focus examines a fault detection algorithm applied to an IMU cluster, specifically that of Principal Component Analysis (PCA). A motivating example is then shown exposing problems in PCA for IMUs, so a modified version of this fault detection method is shown to account for these issues. Underlying fault detection processes are added into this modified fault detection method to completely allow for an autonomous, fault detection method.

The remainder of the thesis examines tools used in fusion and fault detection analysis. Measurement models for position and attitude updates in the extended Kalman filter (EKF) are shown followed by in-depth derivations of the discrete EKF. A full system implementation chapter is also shown in order to tie in and summarize all proposed methodologies. Finally, a spacecraft simulation is constructed and presented with the implementation of all subject matter discussed. This simulation produces results that merit fault detection and fusion necessary for homogeneous, multi-sensor navigation systems.

2. SPACECRAFT DYNAMICS

In order to model the spacecraft dynamics as a function of non-gravitational accelerations and inertial angular velocities, an IMU model is constructed. The continuous time dynamics contain these accelerations and angular velocities as a continuous function of time but are needed to be discretized to account for IMU measurement outputs. Using an inverted version of the proposed IMU model and analytical integration of the continuous time dynamics, the discretized dead-reckoning equations, which compute position, velocity and attitude, are constructed. This allows for mean and covariance analysis which inherently governs the needed navigation solution.

2.1. IMU MODELING

The acceleration and angular velocity measured by an inertial measurement unit (IMU) are corrupted by a variety of error sources. The IMU model, based on Reference [10], accounting for these error sources is given for the accelerometers and gyroscopes as

$$\mathbf{a}_{m,k} = \mathbf{a}_{Q,k} \left((\mathbf{I} + \mathbf{S}_a) (\mathbf{I} + \mathbf{N}_a + \mathbf{M}_a) (\mathbf{T}_i^{\text{imu}} \mathbf{a}_k^i + \mathbf{b}_{a,0} + \mathbf{b}_{a,k} + \mathbf{w}_{a,k}) \right) - \text{E} \{ \mathbf{b}_{a,0} \} \quad (2.1a)$$

$$\boldsymbol{\omega}_{m,k} = \boldsymbol{\omega}_{Q,k} \left((\mathbf{I} + \mathbf{S}_g) (\mathbf{I} + \mathbf{N}_g + \mathbf{M}_g) (\boldsymbol{\omega}_k^{\text{imu}} + \mathbf{b}_{g,0} + \mathbf{b}_{g,k} + \mathbf{w}_{g,k}) \right) - \text{E} \{ \mathbf{b}_{g,0} \} , \quad (2.1b)$$

where

\mathbf{a}_k^i is the true non-gravitational inertial acceleration experienced by the IMU expressed in the inertial frame at time t_k ,

$\mathbf{T}_i^{\text{imu}}$ is the rotation matrix representing the rotation from the inertial frame to the IMU frame,

$\mathbf{b}_{a,0}$ is the startup bias of the accelerometers,

$\mathbf{b}_{a,k}$ is the bias of the accelerometer at time t_k , which changes due to bias instability,

$\mathbf{w}_{a,k}$ is the thermo-mechanical zero-mean white noise present in the accelerometers,

\mathbf{S}_a is the scale factor error matrix of the accelerometers,

\mathbf{M}_a is the axes misalignment matrix of the accelerometers,

\mathbf{N}_a is the axes nonorthogonality matrix of the accelerometers, and

$\mathbf{a}_{Q,k}$ is the quantization affect caused by analog-to-digital conversion.

and similarly for the gyroscopes. The error sources are applied in the following order:

1. Startup bias, walking bias, and thermomechanical noise are applied first because they affect the sensor (accelerometer or gyroscope) regardless of how the sensor is mounted with respect to the defined IMU frame,
2. Axes nonorthogonality and misalignment errors are applied next to account for the mounting error between the sensors and the defined IMU frame,
3. A scale factor error is applied next to account for errant voltages, circuitry, etc. in converting the sensor output to a value that can be quantized,
4. Quantization error is applied last to emulate the Analog to Digital Conversion necessary for quantizing the sensor signal.

The mean of the startup bias of the sensor is subtracted after applying these errors as it is assumed known from sensor testing. It is necessary to add the startup bias before emulating the errors, then subtract its mean so the effect of the errors will have an appropriate effect on the startup bias.

The bias, $\mathbf{b}_{a,k}$, which changes due to bias instability, is given by the random walk model

$$\mathbf{b}_{a,k} = \mathbf{b}_{a,k-1} + \mathbf{w}_{a,BI,k}, \quad (2.2)$$

where $\mathbf{w}_{a,BI,k}$ is a white-noise process of known covariance, which is a function of the velocity random walk specification and time step of the IMU. The IMU errors discussed in the given model are graphically represented versus time in Figure 2.1. Note that the manufacturing errors shown in the figure are defined as the misalignment, nonorthogonality, and scale factor uncertainty errors.

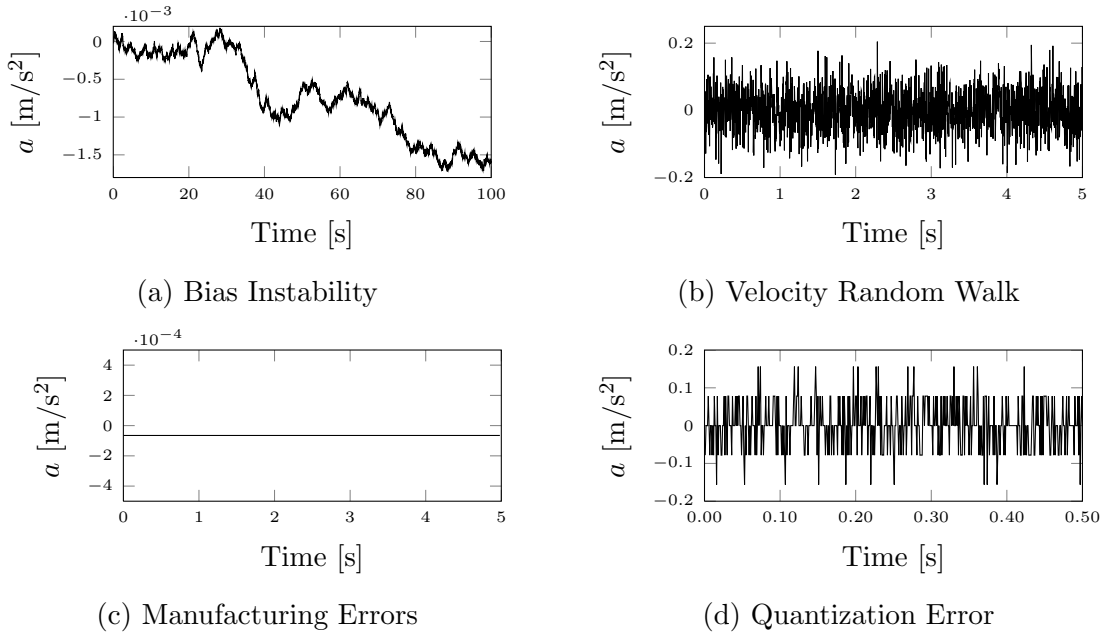


Figure 2.1. IMU Errors vs. Time

The sources of error used by this model with their respective graphics are assumed to affect the gyroscopes identically and thus their presentation is omitted for brevity.

2.2. CONTINUOUS TIME DYNAMICS

The continuous equations of motion for a vehicle with the aid of a strapdown IMU are given by [11]

$$\begin{aligned}\dot{\mathbf{r}}_{\text{fp}}^i(t) &= \mathbf{v}_{\text{fp}}^i(t) \\ \dot{\mathbf{v}}_{\text{fp}}^i(t) &= \mathbf{a}_g^i(\mathbf{r}_{\text{fp}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{cm}/\text{fp}}^c) + \mathbf{T}_c^i(t)\mathbf{a}_{\text{ng}}^c(t) \\ \dot{\bar{\mathbf{q}}}_i^c(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{c/i}^c(t) \otimes \bar{\mathbf{q}}_i^c(t),\end{aligned}$$

where the subscript “fp” represents a fixed point on the vehicle, such as the location of an IMU, and the subscript “cm” denotes the center of mass of the vehicle. \mathbf{r}_{fp}^i and \mathbf{v}_{fp}^i denote the position and velocity of the fixed point in the inertial frame, $\bar{\mathbf{q}}_i^c$ is the attitude quaternion that describes the orientation of the IMU case frame with respect to the inertial frame, $\bar{\boldsymbol{\omega}}_{c/i}^c$ is the pure quaternion representation of the angular velocity of the case frame with respect to the inertial frame and expressed in the case frame, \mathbf{a}_g^i and \mathbf{a}_{ng}^c are the gravitational and non-gravitational accelerations in the inertial and case frames, respectively, and \mathbf{T}_c^i is the transpose of the attitude matrix equivalent to the quaternion $\bar{\mathbf{q}}_i^c$. To simplify the nomenclature, let

$$\begin{aligned}\mathbf{r}_{\text{fp}}^i(t) &\rightarrow \mathbf{r}(t), & \mathbf{v}_{\text{fp}}^i(t) &\rightarrow \mathbf{v}(t), & \mathbf{a}_g^i(\cdot) &\rightarrow \mathbf{g}(\cdot), & \mathbf{T}_c^i(t) &\rightarrow \mathbf{T}^T(t), & \mathbf{a}_{\text{ng}}^c(t) &\rightarrow \mathbf{a}(t), \\ \bar{\mathbf{q}}_i^c(t) &\rightarrow \bar{\mathbf{q}}(t), & \bar{\boldsymbol{\omega}}_{c/i}^c(t) &\rightarrow \bar{\boldsymbol{\omega}}(t), & \mathbf{r}_{\text{cm}/\text{fp}}^c &\rightarrow \mathbf{d}, & \text{and } \mathbf{r}_{\text{fp}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{cm}/\text{fp}}^c &\rightarrow \mathbf{s}(t).\end{aligned}$$

With these substitutions, the equations of motion may be written succinctly as

$$\begin{aligned}\dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) &= \mathbf{g}(\mathbf{s}(t)) + \mathbf{T}^T(t)\mathbf{a}(t) \\ \dot{\bar{\mathbf{q}}}(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}(t) \otimes \bar{\mathbf{q}}(t).\end{aligned}$$

In this work, the extended Kalman filter approach to uncertainty propagation is used. As such, the state estimates are propagated by integration of the equations of motion with the dynamics evaluated at the current state estimate. Therefore, the estimates of position, velocity, and attitude have dynamics that are given by

$$\dot{\hat{\mathbf{r}}}(t) = \hat{\mathbf{v}}(t) \quad (2.3a)$$

$$\dot{\hat{\mathbf{v}}}(t) = \mathbf{g}(\hat{\mathbf{s}}(t)) + \hat{\mathbf{T}}^T(t)\hat{\mathbf{a}}(t) \quad (2.3b)$$

$$\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2}\hat{\boldsymbol{\omega}}(t) \otimes \hat{\mathbf{q}}(t). \quad (2.3c)$$

2.3. IMU INVERSION

In order to perform dead reckoning navigation, the measured non-gravitational acceleration and measured angular velocity must be “inverted” in order to solve for the true non-gravitational acceleration and true angular velocity. These relationships then form the basis for providing estimates of the true quantities that are employed in the discretization of Eqs. (2.3).

The measured non-gravitational acceleration and angular velocity are modeled using Eqs. (2.1). For some vector $\mathbf{v} = [v_x \ v_y \ v_z]^T$, define the matrices $[\mathbf{v}\setminus]$, $[\mathbf{v}\times]$, and $[\mathbf{v}\ast]$ to be

$$[\mathbf{v}\setminus] = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix}, \quad [\mathbf{v}\times] = \begin{bmatrix} 0 & v_z & -v_y \\ -v_z & 0 & v_x \\ v_y & -v_x & 0 \end{bmatrix}, \quad \text{and} \quad [\mathbf{v}\ast] = \begin{bmatrix} 0 & v_z & v_y \\ v_z & 0 & v_x \\ v_y & v_x & 0 \end{bmatrix}.$$

Omitting the effects due to quantization, the accelerometer model of Eq. (2.1a) becomes

$$\mathbf{a}_{m,k} = (\mathbf{I} + \mathbf{S}_a) (\mathbf{I} + \mathbf{N}_a + \mathbf{M}_a) (\mathbf{a}_k + \mathbf{b}_{a,0} + \mathbf{b}_{a,k} + \mathbf{w}_{a,k}) - \mathbf{E}\{\mathbf{b}_{a,0}\}, \quad (2.4)$$

where $\mathbf{a}_k = \mathbf{T}_i^{\text{imu}} \mathbf{a}_k^i$ is used for compactness. Equation (2.4) may be solved for \mathbf{a}_k in terms of the measured acceleration and the errors to yield

$$\mathbf{a}_k = (\mathbf{I} + \mathbf{N}_a + \mathbf{M}_a)^{-1} (\mathbf{I} + \mathbf{S}_a)^{-1} \bar{\mathbf{a}}_{m,k} - \mathbf{b}_{a,0} - \mathbf{b}_{a,k} - \mathbf{w}_{a,k}, \quad (2.5)$$

where $\bar{\mathbf{a}}_{m,k} = \mathbf{a}_{m,k} + \mathbb{E}\{\mathbf{b}_{a,0}\}$. Noting that $(\mathbf{I} + \mathbf{S}_a)(\mathbf{I} + \mathbf{M}_a + \mathbf{N}_a) \approx \mathbf{I} + \mathbf{\Lambda}_a$, applying the matrix inversion lemma to Eq. (2.5), and simplifying the resulting expression, it follows that the true non-gravitational acceleration written in terms of the measured non-gravitational acceleration as

$$\mathbf{a}_k = \mathbf{a}_{m,k} - [\bar{\mathbf{a}}_{m,k} \setminus] \mathbf{s}_a + [\bar{\mathbf{a}}_{m,k} \times] \mathbf{m}_a - [\bar{\mathbf{a}}_{m,k} *] \mathbf{n}_a - (\mathbf{b}_{a,0} - \mathbb{E}\{\mathbf{b}_{a,0}\}) - \mathbf{b}_{a,k} - \mathbf{w}_{a,k}, \quad (2.6)$$

from which one may obtain an estimate of the true non-gravitational acceleration as $\hat{\mathbf{a}}_k = \mathbb{E}\{\mathbf{a}_k\}$, which gives

$$\hat{\mathbf{a}}_k = \mathbf{a}_{m,k} - [\bar{\mathbf{a}}_{m,k} \setminus] \hat{\mathbf{s}}_a + [\bar{\mathbf{a}}_{m,k} \times] \hat{\mathbf{m}}_a - [\bar{\mathbf{a}}_{m,k} *] \hat{\mathbf{n}}_a - \hat{\mathbf{b}}_{a,k} - \hat{\mathbf{w}}_{a,k}. \quad (2.7)$$

An alternative expression for the estimate of the true non-gravitational acceleration is given by evaluating all of the error terms in Eq. (2.5) at their current estimates, such that

$$\hat{\mathbf{a}}_k = (\mathbf{I} + \hat{\mathbf{N}}_a + \hat{\mathbf{M}}_a)^{-1} (\mathbf{I} + \hat{\mathbf{S}}_a)^{-1} \bar{\mathbf{a}}_{m,k} - \hat{\mathbf{b}}_{a,0} - \hat{\mathbf{b}}_{a,k} - \hat{\mathbf{w}}_{a,k}. \quad (2.8)$$

If all of the error sources are zero-mean, it follows from either Eq. (2.7) or Eq. (2.8) that

$$\hat{\mathbf{a}}_k = \mathbf{a}_{m,k}.$$

Parallel results hold for expressing the true angular velocity in terms of the measured angular velocity. That is, following the same process used in arriving at Eq. (2.6), it can be shown that

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}_{m,k} - [\bar{\boldsymbol{\omega}}_{m,k} \setminus] \mathbf{s}_g + [\bar{\boldsymbol{\omega}}_{m,k} \times] \mathbf{m}_g - [\bar{\boldsymbol{\omega}}_{m,k} *] \mathbf{n}_g - (\mathbf{b}_{g,0} - \mathbb{E}\{\mathbf{b}_{g,0}\}) - \mathbf{b}_{g,k} - \mathbf{w}_{g,k}, \quad (2.9)$$

where all of the error sources are now for the gyro instead of the accelerometer, and $\bar{\boldsymbol{\omega}}_{m,k} = \boldsymbol{\omega}_{m,k} + \mathbb{E}\{\mathbf{b}_{g,0}\}$. It is then possible to determine an estimate of the true angular velocity as $\hat{\boldsymbol{\omega}}_k = \mathbb{E}\{\boldsymbol{\omega}_k\}$, which gives

$$\hat{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_{m,k} - [\bar{\boldsymbol{\omega}}_{m,k} \setminus] \hat{\mathbf{s}}_g + [\bar{\boldsymbol{\omega}}_{m,k} \times] \hat{\mathbf{m}}_g - [\bar{\boldsymbol{\omega}}_{m,k} *] \hat{\mathbf{n}}_g - \hat{\mathbf{b}}_{g,k} - \hat{\mathbf{w}}_{g,k}. \quad (2.10)$$

Additionally, an alternative expression for the estimate of the true angular velocity is given by

$$\hat{\boldsymbol{\omega}}_k = (\mathbf{I} + \hat{\mathbf{N}}_g + \hat{\mathbf{M}}_g)^{-1} (\mathbf{I} + \hat{\mathbf{S}}_g)^{-1} \bar{\boldsymbol{\omega}}_{m,k} - \hat{\mathbf{b}}_{g,0} - \hat{\mathbf{b}}_{g,k} - \hat{\mathbf{w}}_{g,k}. \quad (2.11)$$

Finally, as with the accelerometer, if all of the error sources are zero-mean, it follows from either Eq. (2.10) or Eq. (2.11) that

$$\hat{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_{m,k}.$$

2.4. MULTI-SENSOR TRANSLATIONAL COMPENSATION

When the IMU is displaced from the center of mass (CM) of the vehicle, non-gravitational acceleration effects are introduced due to the combined rotation of the vehicle and the displacement of the IMU. In order to transform the acceleration from an arbitrary fixed-point on the vehicle to the CM, translational compensation must be

performed. This step is not necessary when dead-reckoning navigation is performed at the IMU, but is required for navigation about any other point on the vehicle. The only significant effect is due to centripetal acceleration; therefore, compensation for the translational displacement of the IMU is performed as

$$\mathbf{a}_{\text{fp}}^i = \mathbf{a}_{\text{imu}}^i - \boldsymbol{\omega}_{c/i}^i \times \boldsymbol{\omega}_{c/i}^i \times (\mathbf{T}_c^i \mathbf{r}_{\text{fp}/\text{imu}}^c)$$

where the superscripts c , and i denote the case frame and the inertial frame respectively, \mathbf{a}_{imu} is the acceleration at the IMU, and \mathbf{a}_{fp} is the acceleration at some other fixed point on the vehicle.

2.5. DISCRETIZED DEAD-RECKONING EQUATIONS

Dead reckoning navigation is conducted using high-rate IMU data. Therefore, it is assumed that the non-gravitational acceleration and angular velocity are constant over a small time-step, which yields

$$\hat{\mathbf{a}}_k = \frac{\Delta \hat{\mathbf{v}}_k}{\Delta t_k} \quad \text{and} \quad \hat{\boldsymbol{\omega}}_k = \frac{\Delta \hat{\boldsymbol{\theta}}_k}{\Delta t_k}.$$

Applying analytical integration techniques to Eqs. (2.3) under the assumption of constant non-gravitational acceleration and angular velocity and following the derivations from References [11] and [12], it can be shown that the estimates for position, velocity, and attitude evolve according to

$$\hat{\mathbf{r}}_k = \hat{\mathbf{r}}_{k-1} + \hat{\mathbf{v}}_{k-1} \Delta t_k + \frac{1}{2} \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \hat{\boldsymbol{\theta}}_k \times] \right) \Delta \hat{\mathbf{v}}_k \Delta t_k \quad (2.12a)$$

$$+ \frac{1}{2} \left(\hat{\mathbf{g}}_{k-1} - \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_k \right) \Delta t_k^2$$

$$\hat{\mathbf{v}}_k = \hat{\mathbf{v}}_{k-1} + \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{2} [\Delta \hat{\boldsymbol{\theta}}_k \times] \right) \Delta \hat{\mathbf{v}}_k + \left(\hat{\mathbf{g}}_{k-1} - \frac{1}{2} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_k \right) \Delta t_k \quad (2.12b)$$

$$\hat{\mathbf{q}}_k = \bar{\mathbf{q}}(\Delta\hat{\boldsymbol{\theta}}_k) \otimes \hat{\mathbf{q}}_{k-1}, \quad (2.12c)$$

where

$$\bar{\mathbf{q}}(\Delta\hat{\boldsymbol{\theta}}_k) = \begin{bmatrix} \sin\left(\frac{1}{2}\|\Delta\hat{\boldsymbol{\theta}}_{m,k}\|\right) \Delta\hat{\boldsymbol{\theta}}_{m,k}/\|\Delta\hat{\boldsymbol{\theta}}_{m,k}\| \\ \cos\left(\frac{1}{2}\|\Delta\hat{\boldsymbol{\theta}}_{m,k}\|\right) \end{bmatrix}$$

The estimated non-gravitational acceleration and estimated angular velocity are computed from Eqs. (2.7) and (2.10) or Eqs. (2.8) and (2.11). Note that $\hat{\mathbf{G}}$ is defined as the partial derivative of gravity with respect to position or formally written as

$$\hat{\mathbf{G}}_{k-1} = \left[\frac{\partial \mathbf{g}(\mathbf{s})}{\partial \mathbf{s}} \right],$$

and $\hat{\mathbf{d}}$ is the estimated position vector from the center of mass of the vehicle to the IMU. For further information and readings on strapdown IMU modeling, see References [10] and [13].

2.6. MEAN AND COVARIANCE PROPAGATION

As previously mentioned, the dead-reckoning navigation considered in this thesis relies upon the treatment of uncertainty propagation in the same manner as is done for the extended Kalman filter (EKF). That is, the mean of the distribution is propagated using the nonlinear dynamical system and the covariance is propagated using a linearized dynamical system, where the linearization is performed about the current mean. The state vector is chosen to be the collection of the position, velocity, and attitude of the vehicle along with all of modeling parameters of the accelerometer and the gyro. This collection of states is denoted by

$$\mathbf{x}_k = \left[\mathbf{r}_k^T \quad \mathbf{v}_k^T \quad \bar{\mathbf{q}}_k^T \quad \mathbf{a}_{\text{param}}^T \quad \boldsymbol{\omega}_{\text{param}}^T \right]^T,$$

where

$$\mathbf{a}_{\text{param}} = \begin{bmatrix} \mathbf{b}_{a,0}^T & \mathbf{b}_{a,k}^T & \mathbf{s}_a^T & \mathbf{m}_a^T & \mathbf{n}_a^T \end{bmatrix}^T \quad \text{and} \quad \boldsymbol{\omega}_{\text{param}} = \begin{bmatrix} \mathbf{b}_{g,0}^T & \mathbf{b}_{g,k}^T & \mathbf{s}_g^T & \mathbf{m}_g^T & \mathbf{n}_g^T \end{bmatrix}^T .$$

Additionally, the noise terms for the accelerometer and gyro are concatenated into a single process noise as

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{a}_{\text{noise}}^T & \boldsymbol{\omega}_{\text{noise}}^T \end{bmatrix}^T ,$$

where

$$\mathbf{a}_{\text{noise}} = \begin{bmatrix} \mathbf{w}_{a,k}^T & \mathbf{w}_{a,\text{BI},k}^T \end{bmatrix}^T \quad \text{and} \quad \boldsymbol{\omega}_{\text{noise}} = \begin{bmatrix} \mathbf{w}_{g,k}^T & \mathbf{w}_{g,\text{BI},k}^T \end{bmatrix}^T .$$

Then, the dynamical system described by Eqs. (2.12) in conjunction with Eq. (2.2) for describing the evolution of the accelerometer and gyro walking biases may be expressed as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) ,$$

from which the propagation of the mean is obtained as

$$\mathbf{m}_k = \mathbf{f}(\mathbf{m}_{k-1}, \mathbf{0}) , \tag{2.13}$$

where \mathbf{m}_k represents the mean of the state at time t_k and the process noise is taken to be zero mean.

The error covariance is defined as

$$\mathbf{P}_k = \text{E} \{ \mathbf{e}_k \mathbf{e}_k^T \} . \tag{2.14}$$

While not shown here, by defining an error state to be the difference between the truth and the mean, the error can be shown to satisfy the discrete propagation equation

$$\mathbf{e}_k = \mathbf{F}_{k-1}\mathbf{e}_{k-1} + \mathbf{M}_{k-1}\mathbf{w}_{k-1}. \quad (2.15)$$

Substituting Eq. (2.15) into Eq. (2.14), expanding, and noting that the error and process noise are taken to be uncorrelated uncorrelated yields

$$\mathbf{P}_k = \mathbf{F}_{k-1}\mathbb{E}\{\mathbf{e}_{k-1}\mathbf{e}_{k-1}^T\}\mathbf{F}_{k-1}^T + \mathbf{M}_{k-1}\mathbb{E}\{\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T\}\mathbf{M}_{k-1}^T.$$

Defining $\mathbf{Q}_{k-1} \triangleq \mathbb{E}\{\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T\}$ and noting that $\mathbf{P}_{k-1} = \mathbb{E}\{\mathbf{e}_{k-1}\mathbf{e}_{k-1}^T\}$, the final form of the error covariance propagation for the covariance propagation is

$$\mathbf{P}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{M}_{k-1}\mathbf{Q}_{k-1}\mathbf{M}_{k-1}^T. \quad (2.16)$$

The mean and covariance propagation equations shown here are derived in full detail in the derivation of the discretized EKF contained in the Navigation Algorithm chapter. Equations 2.15 and 2.16 are necessary for the discussion of fusion methodology in the proceeding section. To formulate and construct the full $\mathbf{F}(\mathbf{m}_{k-1})$ and $\mathbf{M}(\mathbf{m}_{k-1})$ matrices, refer to Appendix C.

3. FUSION METHODOLOGY

The two main fusion methods investigated in the current work are the direct measurement averaging fusion rule and the covariance intersection fusion rule. While there are many different subsets of each fusion rule, along with different implementations, the two specific fusion methods considered herein allow for a comparison of the general trends to be expected of the methods.

3.1. DIRECT AVERAGING

The direct averaging fusion approach takes N non-gravitational accelerations, transformed by the translational compensation discussed earlier, \mathbf{a}_i , and angular velocities, $\boldsymbol{\omega}_i$, from N IMUs and averages them together to determine a fused non-gravitational acceleration, $\tilde{\mathbf{a}}$, and angular velocity, $\tilde{\boldsymbol{\omega}}$. These fused data are then implemented in the dead-reckoning equations to produce a fused mean, $\tilde{\mathbf{m}}$, and covariance, $\tilde{\mathbf{P}}$. For a set of L measurements \mathbf{x}_i , the fused data $\tilde{\mathbf{x}}$ is given by the direct average

$$\tilde{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad (3.1)$$

where $\mathbf{x}_i = \mathbf{a}_i$ or $\mathbf{x}_i = \boldsymbol{\omega}_i$. This method is the most simple fusion rule due to low computational costs and can be easily implemented with analog circuitry. There are multiple averaging schemes applicable to IMU fusion such as averaging the state estimates and covariances after propagation, but for simplicity, the result of the fused raw measurements will be propagated to provide a fused navigation solution. This method of direct averaging is shown as a block diagram in Figure 3.1.

From Reference [1], the expectation in the improvement in bias stability and noise is on the order of \sqrt{N} , where N is the number of IMUs. In using a sensor

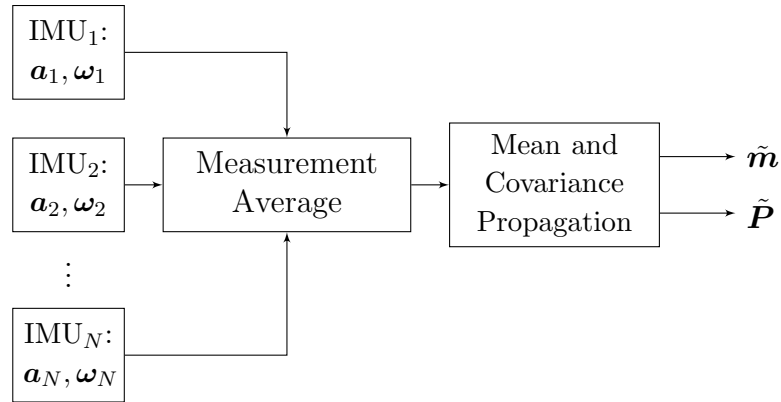


Figure 3.1. IMU Data Fusion Computed by Directly Averaging the Measurements

network of redundant, homogeneous sensors, the actual improvement is expected to be slightly less than the \sqrt{N} factor because all noise sources cannot be assumed to be completely uncorrelated. To visualize this factor of improvement, sensor networks of $N = 10, 100,$ and 1000 single degree of freedom accelerometers are simulated and averaged and are simply modeled in this case by injecting a bias and noise process into the true $a_i = 0$ measurements. The fused measurement is plotted over that of a single sensor in Figure 3.2. As expected it is seen that the noise is reduced by a

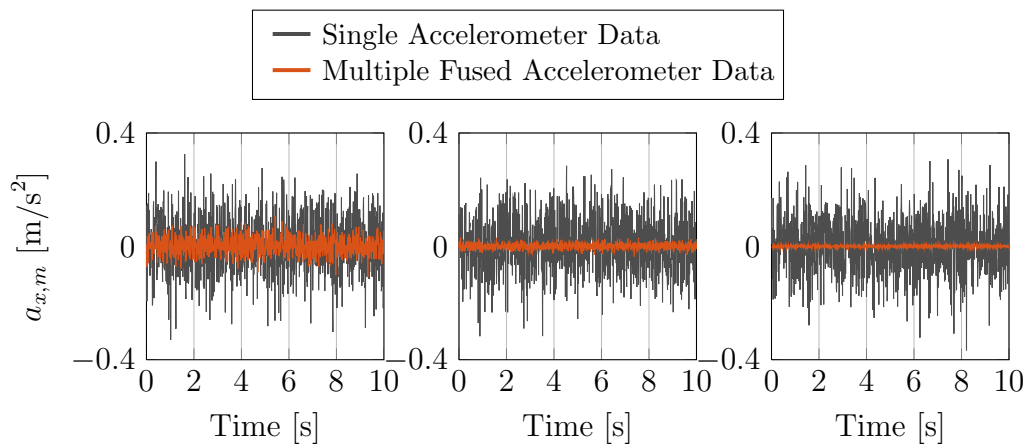


Figure 3.2. Direct Averaging Fusion ($N = 10, 100$ and 1000 Respectively)

significant factor as the number of sensors to be fused increases. This analysis of the direct averaging fusion rule exhibits simplicity; computationally and implementation wise. However, a more intelligent fusion rule is to be considered to examine fusion robustness.

3.2. COVARIANCE INTERSECTION

The covariance intersection (CI) rule is primarily known for fusing two Gaussian distributions by combining the means and covariances in order to produce consistent solutions. This approach can be described as a geometric representation of the Kalman filter because the form of the state estimate and covariance is identical to the standard Kalman filter, which can be found in Reference [3]. If the intersection of covariance is known exactly, the cross covariance will always lie within the intersection of each of the respective covariances from Reference [3]. In order to examine the full form of CI, a general case is implemented in order to fuse N distributions together to produce a more accurate mean and covariance of the system. To derive the CI fusion rule, the geometric mean density (GMD) fusion rule proposed in Reference [4] is applied to N Gaussian distributions. The properties of the GMD fusion rule and more details of the GMD fusion are given in Reference [4]. The most general form of the GMD fusion rule for fusing N arbitrary probability density functions (pdfs) is

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{1}{\eta} \prod_{i=1}^N p_g^{w_i}(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i) \quad (3.2)$$

where η is a normalization factor needed in order to have the distribution integrate to one and $p_g(\mathbf{x}; \mathbf{m}, \mathbf{P})$ represents a Gaussian distribution in \mathbf{x} with mean \mathbf{m} and covariance \mathbf{P} .

Raising a Gaussian distribution to the power w , it follows that

$$\begin{aligned} p_g^w(\mathbf{x}; \mathbf{m}, \mathbf{P}) &= \left[|2\pi\mathbf{P}|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \right\} \right]^w \\ &= |2\pi\mathbf{P}|^{-w/2} \exp \left\{ -\frac{w}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \right\} \end{aligned}$$

By defining $\bar{\mathbf{P}} \triangleq \mathbf{P}/w$, the Gaussian density raised to the power w may also be expressed as

$$p_g^w(\mathbf{x}; \mathbf{m}, \mathbf{P}) = |2\pi\mathbf{P}|^{-w/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^T \bar{\mathbf{P}}^{-1} (\mathbf{x} - \mathbf{m}) \right\}. \quad (3.3)$$

The term within the exponential is of the form of the Gaussian distribution with a scaled covariance, but the normalizing factor does not account for the scaled covariance. The normalizing factor should be $|2\pi\bar{\mathbf{P}}|^{-1/2}$; therefore, multiplying and dividing the raised Gaussian in Eq. (3.3) by the proper normalizing constant gives

$$p_g^w(\mathbf{x}; \mathbf{m}, \mathbf{P}) = k p_g(\mathbf{x}; \mathbf{m}, \bar{\mathbf{P}}), \quad (3.4)$$

where $k = w^{-n/2} |2\pi\mathbf{P}|^{(1-w)/2}$. That is, a Gaussian distribution raised to the power w is a scaled Gaussian with an inflated covariance (provided that $0 \leq w \leq 1$). Substituting the result of Eq. (3.4) into each term of the product in Eq. (3.2), it follows that the GMD fusion rule for Gaussian distributions is

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{1}{\eta} \prod_{i=1}^N k_i p_g(\mathbf{x}; \mathbf{m}_i, \bar{\mathbf{P}}_i), \quad (3.5)$$

where $k_i = w_i^{-n/2} |2\pi\mathbf{P}_i|^{(1-w_i)/2}$ and $\bar{\mathbf{P}}_i = \mathbf{P}_i/w_i$.

When the product is expanded out, the fused distribution of Eq. (3.5) becomes

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{\bar{k}}{\eta} [p_g(\mathbf{x}; \mathbf{m}_1, \bar{\mathbf{P}}_1) p_g(\mathbf{x}; \mathbf{m}_2, \bar{\mathbf{P}}_2) p_g(\mathbf{x}; \mathbf{m}_3, \bar{\mathbf{P}}_3) \cdots p_g(\mathbf{x}; \mathbf{m}_N, \bar{\mathbf{P}}_N)], \quad (3.6)$$

where $\bar{k} = \prod_{i=1}^N k_i$. The product of two Gaussian distributions is a well known procedure [14], but this process becomes more complicated when considering N pdfs. Given two Gaussian pdfs $p_g(\mathbf{x}; \mathbf{m}_a, \mathbf{P}_a)$ and $p_g(\mathbf{x}; \mathbf{m}_b, \mathbf{P}_b)$, their product is [14]

$$p_g(\mathbf{x}; \mathbf{m}_a, \mathbf{P}_a)p_g(\mathbf{x}; \mathbf{m}_b, \mathbf{P}_b) = \Gamma_{ab} p_g(\mathbf{x}; \mathbf{m}_c, \mathbf{P}_c) , \quad (3.7)$$

where

$$\mathbf{m}_c = \mathbf{P}_c (\mathbf{P}_a^{-1} \mathbf{m}_a + \mathbf{P}_b^{-1} \mathbf{m}_b) \quad (3.8a)$$

$$\mathbf{P}_c = (\mathbf{P}_a^{-1} + \mathbf{P}_b^{-1})^{-1} \quad (3.8b)$$

$$\Gamma_{ab} = |2\pi (\mathbf{P}_a + \mathbf{P}_b)^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{m}_a - \mathbf{m}_b)^T (\mathbf{P}_a + \mathbf{P}_b) (\mathbf{m}_a - \mathbf{m}_b) \right\} . \quad (3.8c)$$

Direct application of Eq. (3.7) to the product of $p_g(\mathbf{x}; \mathbf{m}_1, \bar{\mathbf{P}}_1)$ and $p_g(\mathbf{x}; \mathbf{m}_2, \bar{\mathbf{P}}_2)$ in Eq. (3.6) gives the GMD as

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{\bar{k}}{\eta} \Gamma_{12} [p_g(\mathbf{x}; \mathbf{m}_{12}, \bar{\mathbf{P}}_{12}) p_g(\mathbf{x}; \mathbf{m}_3, \bar{\mathbf{P}}_3) \cdots p_g(\mathbf{x}; \mathbf{m}_N, \bar{\mathbf{P}}_N)] , \quad (3.9)$$

where, from Eq. (3.5),

$$\mathbf{m}_{12} = \bar{\mathbf{P}}_{12} (\bar{\mathbf{P}}_1^{-1} \mathbf{m}_1 + \bar{\mathbf{P}}_2^{-1} \mathbf{m}_2) \quad (3.10a)$$

$$\bar{\mathbf{P}}_{12} = (\bar{\mathbf{P}}_1^{-1} + \bar{\mathbf{P}}_2^{-1})^{-1} \quad (3.10b)$$

and similarly for Γ_{12} . Now, apply Eq. (3.7) to the product of $p_g(\mathbf{x}; \mathbf{m}_{12}, \bar{\mathbf{P}}_{12})$ and $p_g(\mathbf{x}; \mathbf{m}_3, \bar{\mathbf{P}}_3)$ in Eq. (3.9) to obtain

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{\bar{k}}{\eta} \Gamma_{12} \Gamma_{123} [p_g(\mathbf{x}; \mathbf{m}_{123}, \bar{\mathbf{P}}_{123}) \cdots p_g(\mathbf{x}; \mathbf{m}_N, \bar{\mathbf{P}}_N)] , \quad (3.11)$$

where, from Eqs. (3.8), it follows that

$$\begin{aligned}\mathbf{m}_{123} &= \bar{\mathbf{P}}_{123} \left(\bar{\mathbf{P}}_{12}^{-1} \mathbf{m}_{12} + \bar{\mathbf{P}}_3^{-1} \mathbf{m}_3 \right) \\ \bar{\mathbf{P}}_{123} &= \left(\bar{\mathbf{P}}_{12}^{-1} + \bar{\mathbf{P}}_3 \right)^{-1},\end{aligned}$$

and similarly for Γ_{123} . From Eqs. (3.10), it follows that \mathbf{m}_{123} and $\bar{\mathbf{P}}_{123}$ are equivalently expressed as

$$\begin{aligned}\mathbf{m}_{123} &= \bar{\mathbf{P}}_{123} \left(\bar{\mathbf{P}}_1^{-1} \mathbf{m}_1 + \bar{\mathbf{P}}_2^{-1} \mathbf{m}_2 + \bar{\mathbf{P}}_3^{-1} \mathbf{m}_3 \right) \\ \bar{\mathbf{P}}_{123} &= \left(\bar{\mathbf{P}}_1^{-1} + \bar{\mathbf{P}}_2^{-1} + \bar{\mathbf{P}}_3 \right)^{-1}.\end{aligned}$$

It then follows that the GMD fusion rule for a set of N Gaussian distributions is given by continuing to reduce products of pairs of Gaussian pdfs; therefore, by induction, the GMD fusion rule becomes

$$\tilde{\mathbf{p}}(\mathbf{x}) = \frac{1}{\eta} \left[\bar{k} \Gamma_{12} \Gamma_{123} \cdots \Gamma_{12\dots N} \right] p_g(\mathbf{x}; \tilde{\mathbf{m}}, \tilde{\mathbf{P}}), \quad (3.12)$$

where

$$\tilde{\mathbf{m}} = \tilde{\mathbf{P}} \sum_{i=1}^N \bar{\mathbf{P}}_i^{-1} \mathbf{m}_i \quad \text{and} \quad \tilde{\mathbf{P}} = \left[\sum_{i=1}^N \bar{\mathbf{P}}_i^{-1} \right]^{-1}.$$

Since the integral of $\tilde{\mathbf{p}}(\mathbf{x})$ must evaluate to one and since the integral of $p_g(\mathbf{x}; \tilde{\mathbf{m}}, \tilde{\mathbf{P}})$ is guaranteed to evaluate to one, it follows that $\eta = \bar{k} \Gamma_{12} \Gamma_{123} \cdots \Gamma_{12\dots N}$. Additionally, recall that $\bar{\mathbf{P}}_i = \mathbf{P}_i/w_i$. Substituting for the normalizing factor and the scaled covariances, the GMD fusion rule in Eq. (3.12) is given in its final form as

$$\tilde{\mathbf{p}}(\mathbf{x}) = p_g(\mathbf{x}; \tilde{\mathbf{m}}, \tilde{\mathbf{P}}), \quad (3.13)$$

where

$$\tilde{\mathbf{m}} = \tilde{\mathbf{P}} \sum_{i=1}^N w_i \mathbf{P}_i^{-1} \mathbf{m}_i \quad \text{and} \quad \tilde{\mathbf{P}} = \left[\sum_{i=1}^N w_i \mathbf{P}_i^{-1} \right]^{-1}. \quad (3.14)$$

Thus, the GMD fusion rule produces a Gaussian distribution when applied to a set of N input Gaussian distributions; in fact, Eq. (3.13) is exactly the CI method for fusing a set of N means and covariances.

3.2.1. Implementation. The implementation using the CI fusion rule propagates the mean and covariance of each individual IMU and is followed by an application of CI at each time step to output a fused mean and covariance. This fusion rule is ideal for cheaper, fault-prone sensors in a distributed network by considering the weights correlated with CI. There are many different weighting techniques, although for brevity, only one will be examined. The process of the implementation of CI is outlined and shown in the block diagram in Figure 3.3.

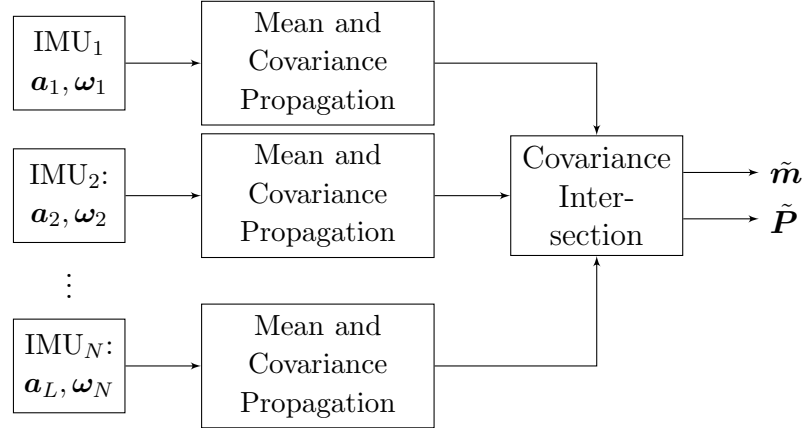


Figure 3.3. IMU Data Fusion Computed by Covariance Intersection

It is useful to visually see the effects of the CI fusion rule. Consider an example where two zero mean distributions, $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ respectively, with two

covariances and respective arbitrarily selected weights fused together using CI to produce the fused distribution $\tilde{p}(\mathbf{x})$. The uncertainty ellipses based on their respective distributions are plotted with the fused uncertainty ellipse in Figure 3.4. It is seen

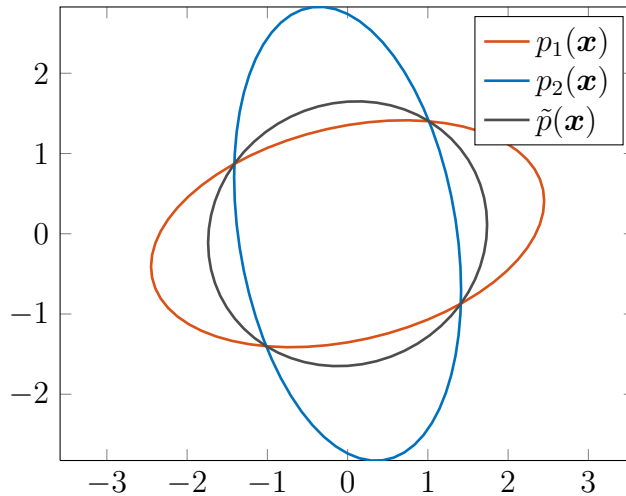


Figure 3.4. CI Fusion Rule

that the fused data produce a smaller uncertainty ellipse that geometrically is projected and drawn through the intersections of the two distributions undergoing fusion. However, it is not practical to arbitrarily select weights for each covariance; therefore, a weighting scheme is needed to be shown.

3.2.2. Generalized Variance Weighting Selection. The covariance intersection algorithm depends on weighting low uncertainties larger than high uncertainties. One proposed method of weighting is taken from the generalized variance method, proposed by Wilks, found in Reference [15], as a scalar measure of overall multidimensional scatter. The generalized variance is the trace of the reciprocal of the information matrix, which in the fusion case is the trace of the covariance matrix $\text{tr}\{\mathbf{P}\}$. With this knowledge, an algorithm for uncertainty weighting can be used to

intelligently weight the uncertain measurements. Noting that

$$\sum_{i=1}^L w_i = 1,$$

let P^* be the sum of the determinants of the covariance matrices, such that

$$P^* = \sum_{i=1}^L |\mathbf{P}_i|.$$

The trace weights can then be appropriately given by

$$w_i^* = \frac{\text{tr}\{\mathbf{P}_i\}}{P^*}.$$

However, this weighting scheme provides large weights for higher covariances which would allow for measurements with larger uncertainties to be trusted more than measurements with lower uncertainties. Noting that the above weight's value is inversely proportional to the desired weight, the final expression for generalized variance weight selection given as

$$w_i = \frac{1 - w_i^*}{\sum_{\ell=1}^N w_\ell^*}. \quad (3.15)$$

3.3. FAULT TOLERANT FUSION

IMU failures are typically very rare for most spacecraft due to the high fidelity nature of high-cost IMUs. In order to reduce the cost of using a single, high-cost, high-reliability tactical or strategic grade IMU, consider the case where multiple low-cost, commercial off-the-shelf (COTS) IMUs are used. In this case, the cost of a single high-cost, high-reliability IMU-based navigation system can be reduced by using a multiple low-cost IMU-based navigation system; however, this multiple IMU-based

system must be robust to faults in the IMUs as the lower-cost COTS IMUs have a non-negligible possibility of failure as compared to the single high-cost IMU.

Two fault tolerant frameworks are considered for this multiple IMU-based navigation system: direct averaging of the measured acceleration and angular velocity provided by each IMU and fusing the dead-reckoned state of each IMU using the CI method. The direct averaging method is expected to provide an estimate of the vehicle state with a lower uncertainty than is obtained by fusing the individual dead-reckoned navigation solutions using the covariance intersection method. When fusing the individual dead-reckoned navigation solutions using CI, the weights used for CI are selected as a function of the uncertainty in each dead-reckoned solution; thus, this fusion rule is predicted to be more robust than the direct averaging fusion rule. The uncertainty in the dead-reckoned solution corresponding to a failed sensor will become very large, which means that its weight will tend to zero, thereby effectively removing its influence on the fused navigation solution.

In order to take advantage of both the lower uncertainty obtained when direct averaging the measured accelerations and angular velocities and the robustness of fusing the individual navigation solutions using the CI method, a system is proposed that runs both algorithms in parallel and uses the dead-reckoned solution corresponding to the direct averaging method in the presence of no sensor failures. As a safe mode, the algorithm will switch to using the fused navigation solution given by the CI method in the case of sensor failure(s). This algorithm is summarized by Algorithm 1 in Appendix B. In order for the uncertainty of the individual dead-reckoned navigation solutions, $\mathbf{P}_{1,\dots,N}$, to propagate accurately, it is assumed that real-time values for the error parameters of each IMU are available from some type of real-time sensor calibration. This will cause \mathbf{P}_i to grow more quickly when IMU_{*i*} fails and produces more corrupted data than normal. If these data are blindly averaged, as is proposed in the direct averaging method, after a fault occurs in the system, then the

navigation solution corresponding to this method can no longer accurately represent the covariance. In the case where the weights used in CI fusion can down-weight failing sensors, the averaging fusion rule cannot be considered robust. As of now the covariances being fused are modeled as the true uncertainty of the system which is not the case in performing linear covariance. With this assumed knowledge of sensor and covariance degradation, CI is expected to down-weight dead-reckoned solutions with higher uncertainties acting as a pseudo fault detection method.

3.4. FUSION EXAMPLE

To demonstrate the performance of the proposed direct averaging and covariance intersection fusion methods, a simulation is performed in which the navigation solution fused from six low-cost IMUs is compared against the navigation solution obtained using a single tactical grade IMU. The true non-gravitational acceleration and angular velocities of the simulated IMUs are simulated for a spacecraft in LEO using AGI STK¹ with a primary spin about the body z -axis. The true accelerations and angular velocities are corrupted according to the error model given in Eqs. (2.1). The error parameters for these models are based off of the Microstrain 3DM-GX3-15² and the Epson M-G362PDC1³ IMUs and are summarized in Appendix A. Three of each IMU are assumed to be on the spacecraft, which is considered to be a $1.0 \times 1.0 \times 1.0$ meter cube. The locations of each of the six IMUs as well as the CM of the spacecraft are shown in Table 3.1.

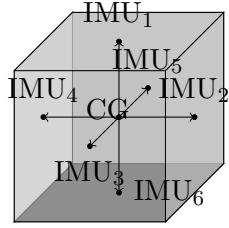
In order to compare the direct averaging and CI fusion rules, consider the case where one of the Microstrain 3DM-GX3-15 IMUs fails at $t = 100$ s, an arbitrarily chosen time, and begins to provide more corrupted data. Because the noise characteristics of this more corrupted data are assumed known, the uncertainty in

¹<http://www.agi.com/products/stk/>

²<http://www.microstrain.com/inertial/3dm-gx3-35>

³http://www.epsondevice.com/en/sensing_system/product/imu/g362/

Table 3.1. Spacecraft IMU Configuration

IMU	Position Vector [m]	
IMU ₁ : M-G362PDC1	$\mathbf{d}_1 = [0.0 \ 0.0 \ 0.5]$	
IMU ₂ : M-G362PDC1	$\mathbf{d}_2 = [0.0 \ 0.5 \ 0.0]$	
IMU ₃ : M-G362PDC1	$\mathbf{d}_3 = [0.5 \ 0.0 \ 0.5]$	
IMU ₄ : 3DM-GX3-15	$\mathbf{d}_4 = [0.0 \ -0.5 \ 0.0]$	
IMU ₅ : 3DM-GX3-15	$\mathbf{d}_5 = [-0.5 \ 0.0 \ 0.0]$	
IMU ₆ : 3DM-GX3-15	$\mathbf{d}_6 = [0.0 \ 0.0 \ -0.5]$	

the navigation solution associated with this IMU begins to grow very quickly. For brevity, only the position and attitude uncertainties are presented because the position and velocity uncertainties follow the same general trends and uncertainty is the most effective way to evaluate the fusion rules. The legend used for presenting the results is shown in Table 3.2. To provide information on the trend of uncer-

Table 3.2. Fusion Example Legend

—	Single 3DM-GX3-15 IMU Failure
---	Systron Donner: SDI-500 (Not Fused)
—	Epson: M-G362PDC1
—	Micro Strain: 3DM-GX3-15
—	Covariance Intersection Fusion Rule
---	Direct Averaging Fusion Rule

tainty growth, the position and attitude root-sum-square (RSS) are shown in Figure 3.5. The failure of the single Microstrain 3DM-GX3-15 IMU at $t = 100$ s causes an instantaneous change in how the uncertainty of the single IMU dead-reckoning navigation solution evolves. In the case of the position RSS, because the failure causes changes to the IMU acceleration, the effects are less immediate due to the necessity

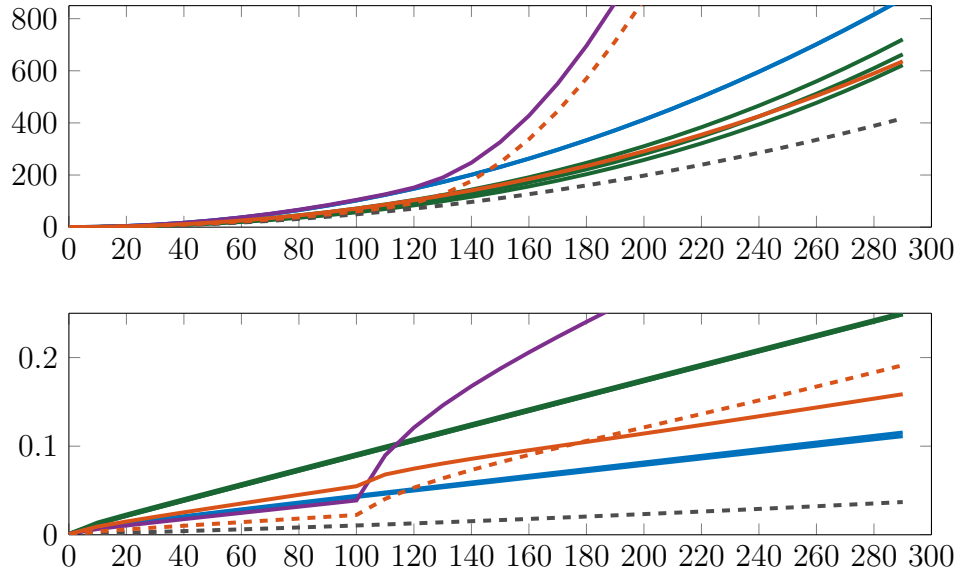


Figure 3.5. Position [m] and Attitude [deg] RSS Error

of doubly integrating the acceleration. In terms of the fusion rules, however, the position RSS demonstrates the quickest switch between the direct averaging fusion rule and the CI fusion rule. At approximately $t = 125$ s, the CI fusion rule begins to outperform the direct averaging fusion rule in position uncertainty. This reversal occurs only at approximately $t = 180$ s in the attitude uncertainty. In order to show the uncertainty in the position with less cluttered plots, a series of two dimensional uncertainty contour plots in the $x - y$ plane are shown at $t = 50$, 200 , and 300 s in Figure 3.6. The first time step shows the position uncertainty before any IMU failure, which illustrates that the direct averaging fusion rule provides a more confident navigation solution in the presence of no IMU failures. The next two time steps show the position uncertainty after the IMU failure, demonstrating that the CI fusion rule is preferred in the case of an IMU failure. While the direct averaging fusion rule clearly follows along with the failed IMU, the CI fusion rule downweights the failed IMU and is seen to be consistent with the remaining, nominally operating IMUs. However, the CI fusion rule has no knowledge of a sensor failure so a comparison of the two fusion

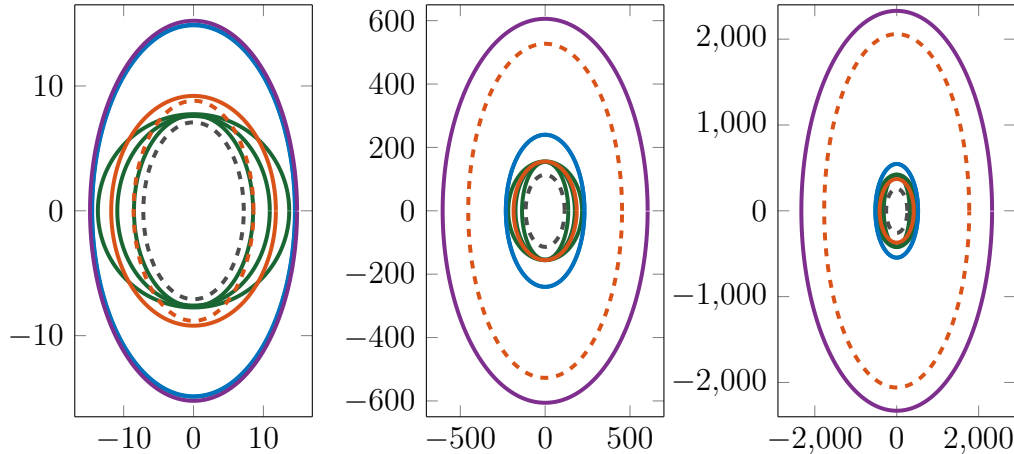


Figure 3.6. $x - y$ Uncertainty Contours [m] at 50, 200, 300 [s], Respectively

rules' true covariance outputs are shown for position and attitude in Figure 3.7. It can be determined that the true uncertainty of data averaging system undergoing no sensor failures can be characterized much closer to that of a tactical grade IMU. The CI fusion rule cannot obtain a covariance navigation solution lower than the lowest individual sensor which promotes direct averaging over CI.

3.5. FUSION METHODOLOGY SUMMARY AND CONCLUSION

In order to provide a lower-cost inertial measurement unit (IMU)-based navigation system as opposed to one that uses a typical high-cost, high-reliability tactical or strategic grade IMU, a fault-tolerant method was proposed that uses multiple low-cost IMUs and fuses their data together via a direct averaging method or fuses their individual navigation solutions together via the covariance intersection (CI) fusion rule. It is demonstrated that the direct averaging fusion rule tends to outperform the CI fusion rule during nominal operations where no hardware failures are present. On the other hand, the CI fusion rule is shown to outperform the direct averaging fusion rule when an IMU fails due to the ability of the CI fusion rule to selectively down-weight the navigation solution obtained from the failed IMU. The proposed algorithm

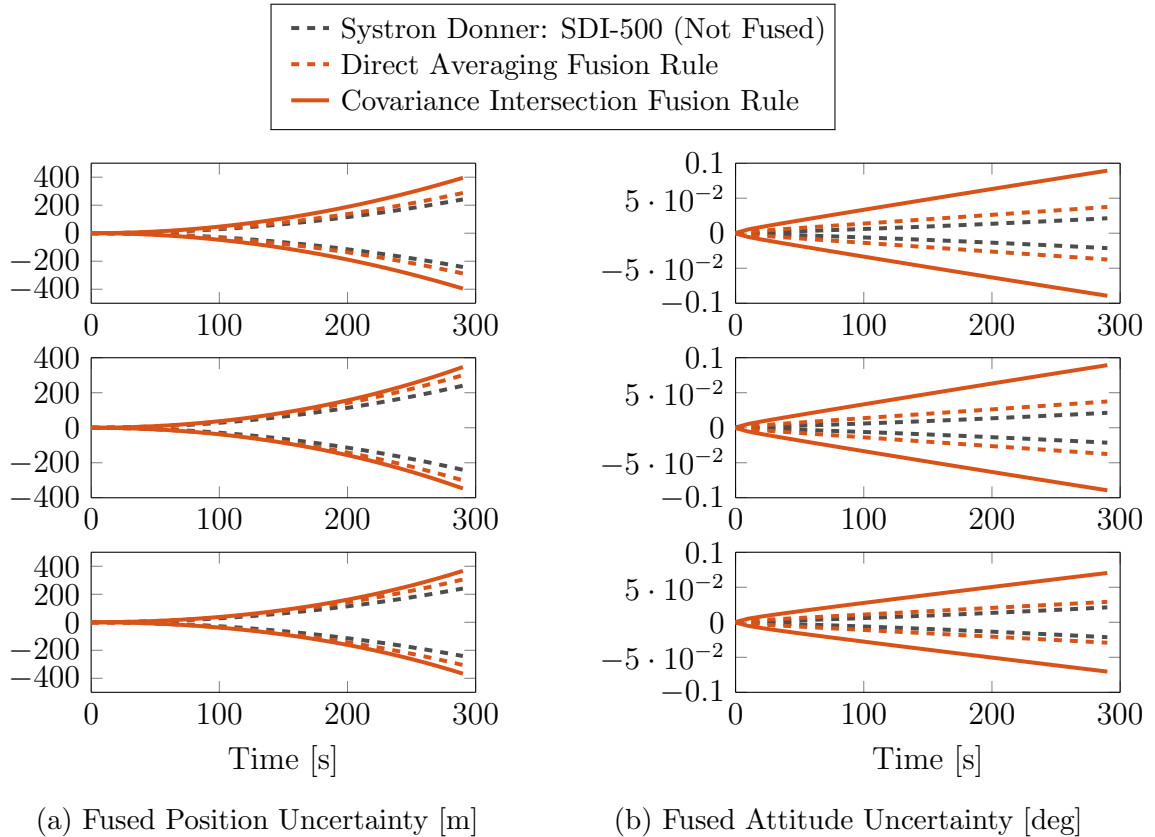


Figure 3.7. Fusion Rule Uncertainty Comparison

uses the navigation solution propagated using the direct averaged data when there are no IMU faults because it provides a more confident solution. In the case of an IMU failure, the algorithm will recognize the failure, and the navigation solution from the covariance intersection method will be used. However, the covariance propagation is a function of sensor errors given from data sheet specifications. The uncertainties undergoing failures were realized by adding in the sensor failure parameters into the IMU model forcing knowledge of the failure. If the covariance propagation could autonomously detect changes in sensor parameters, then CI would be the robust, fault tolerant fusion method. However, this error realization is not likely or realistic. It was seen in the results that a simple averaging fusion method produces a more accurate navigation solution than that of CI concluding that direct averaging is the best

option for IMU data fusion. Therefore, data averaging has no knowledge of faulty data, a fault detection architecture must be examined and implemented to provide a true fault tolerant multi-sensor network on-board a spacecraft.

4. FAULT DETECTION

To reiterate the conclusions of the comparison of direct averaging and CI, direct averaging out performs CI in the case of no faults. It was also seen that if a sensor failure is present, CI has the robust ability to recognize the fault if weighted correctly. Unfortunately, the proposed weighting solution is a function of covariance which is computed from sensor data sheet statistics and specifications. This leaves both fusion methods with the need for a robust, fault detection scheme, and due to the higher estimate direct averaging has to offer, the proposed fusion method in the considerations of fault detection is that of direct averaging. Principal Component Analysis (PCA) is the initial proposed fault detection method. However, an example is shown concluding in the need of a more desirable, modified version of PCA that better fits the dynamic nature of IMUs.

4.1. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) has many applications, such as dimensionality reduction, feature extraction and data visualization. While dimensionality reduction does not have much application to detecting faults in IMU data, feature extraction and data visualization play an important role in recognizing faults in an IMU sensor network. From Bishop [16], there are two commonly used definitions of PCA consisting of a maximum variance formulation and a minimum-error formulation. The maximum variance formulation orthogonally projects data onto a linear space with lower dimension such that the variance of the transformed data is maximized. This linear space is known as the principle subspace or the feature plane. The minimum-error formulation linearly projects the data while minimizing the average projection cost. Each method is equivalent, and the studied PCA definition will be that of the maximum variance formulation. In the case where there is no dynamical

movement, consider a multiple IMU sensor network with N number of IMUs. Following procedures from Lee [8] and Bishop [16], the first step is to fuse the data by directly averaging the measurements to get \mathbf{x}_{avg} , governed by

$$\mathbf{x}_{avg} = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{x}}_i, \quad (4.1)$$

where $\hat{\mathbf{x}}_i \in \mathbb{R}^m$ is the measured outputs from each IMU i.e. \mathbf{a}_i or $\boldsymbol{\omega}_i$. A data set is then constructed to produce zero mean data, $\mathbf{m}_{p,i}$, by subtracting the averaged output from each measurement given as

$$\mathbf{m}_{p,i} = \hat{\mathbf{x}}_i - \mathbf{x}_{avg}, \quad (4.2)$$

The next step is to compute the measurement covariance matrix. By definition the covariance is given by

$$\mathbf{P} = \mathbb{E} \left\{ (\mathbf{x} - \mathbb{E}[\mathbf{x}]) (\mathbf{x} - \mathbb{E}[\mathbf{x}])^T \right\},$$

which can be notationally written for a multi-sensor system as

$$\mathbf{P} = \frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{x}}_i - \mathbf{x}_{avg}) (\hat{\mathbf{x}}_i - \mathbf{x}_{avg})^T \quad (4.3)$$

$$= \frac{1}{N} \sum_{i=1}^N \mathbf{m}_{p,i} \mathbf{m}_{p,i}^T, \in \mathbb{R}^{m \times m}. \quad (4.4)$$

This covariance can be written in terms of its eigenvectors and eigenvalues, such that

$$\mathbf{P} = \mathbf{V} \boldsymbol{\Gamma} \mathbf{V}^T, \quad \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \quad (4.5)$$

where $\mathbf{V} \in \mathbb{R}^{m \times m}$ is the eigenvector matrix of \mathbf{P} composed of m unit eigenvectors \mathbf{v}_i , and $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times m}$ is the diagonal eigenvalue matrix of \mathbf{P} . This is found by solving

the characteristic equation

$$(\mathbf{P} - \lambda_i \mathbf{I}_{m \times m}) \mathbf{v}_i = \mathbf{0}, \quad (4.6)$$

where λ_i and \mathbf{v}_i denote the i^{th} eigenvalue and eigenvector respectively. The eigenvectors and eigenvalues are introduced as a representation of the variance in the data due to their orthogonality, which exposes patterns in the data. This pattern can be thought of as a relationship of the data sets that are related along a best-fit line. The most significant relationship between the data sets is known as the principle component and can be represented by the eigenvector with the largest eigenvalue. Once all eigenvector and eigenvalue pairs are found, it is important to list them from largest to smallest which will give an ordering of significance to the components known as a feature vector matrix Φ . Components of lesser significance can then be ignored; however, some information will be lost, but this can be considered negligible for most cases. The feature matrix can be written in terms of the most significant eigenvectors to lowest written as

$$\Phi = \begin{bmatrix} \phi_1^* & \phi_2^* & \cdots & \phi_s^* \end{bmatrix},$$

where the * superscript represents the eigenvectors in order of most significant to least, and the subscript s denotes the number of selected dimensions used where $s \leq m$. The feature vector constructed is then used to transform the original data into a new data set that shows the sensor's data patterns; the transformed data is

$$\mathbf{y} = \Phi^T \begin{bmatrix} \hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_1 & \cdots & \hat{\mathbf{x}}_N \end{bmatrix}^T, \quad (4.7)$$

where \mathbf{y} is the transformed data set that correlates the original data with a sensor pattern. While this case is only useful for a sensor with no movement, PCA can

be used for sensor calibration and fault detection when the system is at rest. This also lays the foundation for MPCA, which will provide fault tolerant algorithms for dynamical systems.

4.2. FAULT DETECTION EXAMPLE

A motivating example is provided to compare PCA for a static and dynamic accelerometer. Consider six ($L = 6$) 3-axis ($m = 3$) accelerometer systems assuming each accelerometer is at the center of mass. Throughout the example two selected dimensions for the feature matrix ($s = 2$) are used. The static measurements are simulated by corrupting a true acceleration of $\mathbf{a}_s = \mathbf{0}$ and the dynamic measurements are simulated from the equations of motion given by $\mathbf{a}_d = [\sin^2(\theta) \cos(\theta) 0.0]^T$ where $\theta = t\Delta t$ [rad]. The example is carried out with $\Delta t = 0.01$ [s], with a simulation time of 10 seconds, $t_f = 10$ [s]. At 5 seconds into the simulation, accelerometer three undergoes an increase in bias and noise, simulating a sensor fault. This example is used to prove and visualize the need for a modified version of PCA for dynamic systems. The measurements are then transformed using principal components based on Eq. (4.7). Figure 4.1 shows the effects and behavior of PCA applied on the static and dynamic systems.

Note that y_1 is the dominant information holder in the feature plane due to the feature vector matrix Φ giving preference to the most significant eigenvector. After a failure y_1 is dominated by the fault allows for the fault to expose itself while y_2 is dominated by the patterns of sensor noise. It is seen that in the static case, PCA is able to easily identify threshold outliers enabling a fault alarm to disregard the outlying measurements. The threshold value is selected based on the interpretation on what is and is not a failure. A fault alarm is determined by pulling a fault identification signal high to alert the system of a failure with a corresponding sensor ID ($i = 3$) in this case. This method is discussed in depth in the simulation results

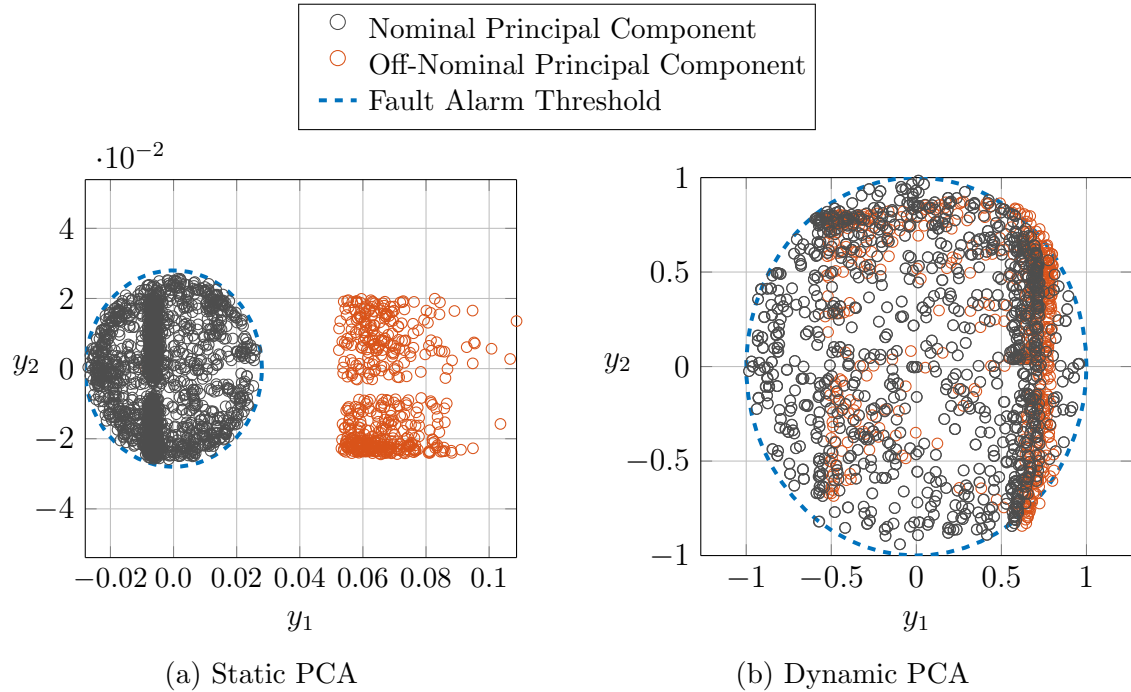


Figure 4.1. Effects of PCA Undergoing Single Sensor Failure for Static and Dynamic Systems

section. However, in the dynamic case, the principal components containing fault information cannot be distinguished from the nominal principal components proving a need to remove dynamic effects from the feature plane. This is the goal when modifying the principal component analysis method.

In Figure 4.1, it is seen that principal components tend to shrink in the first principal component direction. When plotted in two dimensions (y_1 vs. y_2), this decrease in the first principal component is not easily seen; therefore, each principal component is plotted separately versus time for the static case and the dynamic case and is shown in Figure 4.2.

While the clustered data points in the two dimensional plots become apparent from the one dimensional plots, it is desired to seek an explanation for the drastic decrease in the feature plane along the first principal component. The static case is examined in order to conclude a reasoning for this phenomena. It is known that the

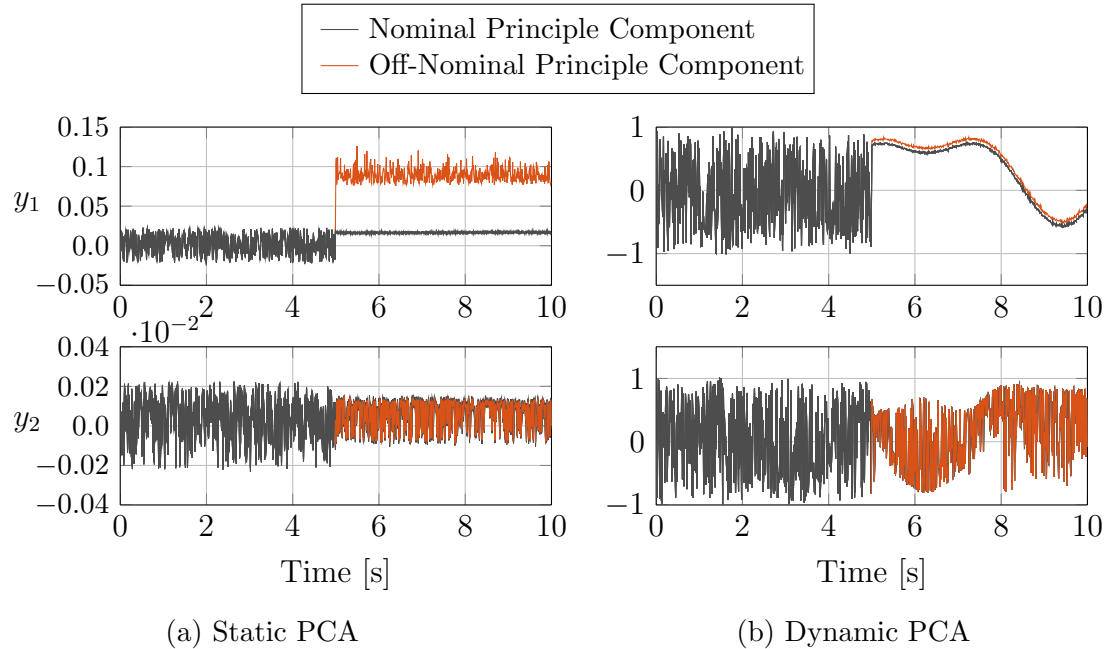


Figure 4.2. First and Second Principal Components versus Time for a Static and Dynamic System Undergoing a Single Sensor Failure

direction of the eigenvectors, used as the first and second principal components, are orthogonal and geometrically point along the best-fit lines through data along the first and second principal directions. The eigenvector pairs are plotted over a point cloud of zero-mean data in three dimensions for twenty sensors instead of the six sensors priorly used to emphasize the illustration. In the current discussion of fault detection, faulty data is denoted as off-nominal components, and the functional data is denoted as nominal components. These illustrations are plotted at three epochs prior-and post-failure and are shown in Figure 4.3.

From the PCA derivation, it is known that the feature plane is a function of the principal eigenvectors and the data set. Figures 4.3a-4.3c shows direction of the principal components pointing along the best fit line through the data. It is seen that both eigenvectors are pointing in drastically different directions at each epoch that accounts for the “noisy” feature plane patterns before the failure. Examining the

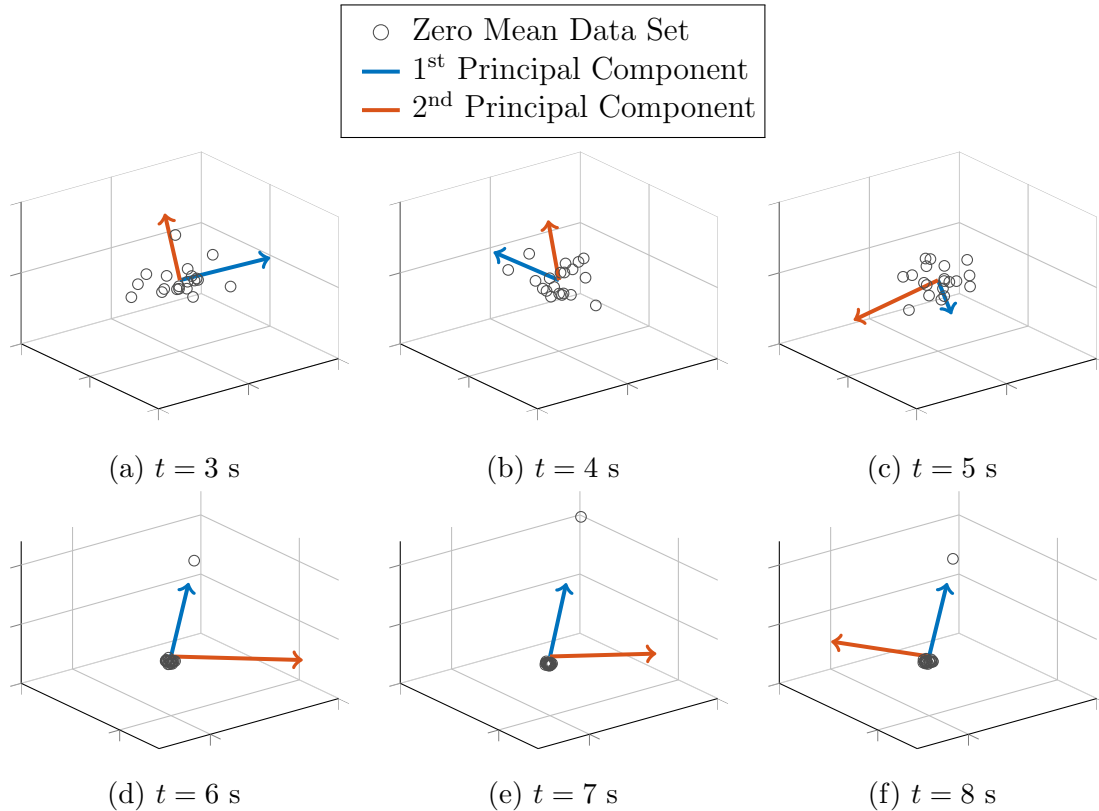


Figure 4.3. First and Second Principal Components Plotted Over 3-Dimensional Zero-Mean Data Set Before and After Failure

failure case in Figures 4.3d-4.3f, it is seen that the best fit line along the 1st principal component is along the direction of the faulty sensor data. The 1st principal component's variation remains relatively constant after the fault in order to geometrically satisfy the best fit line along the fault.

4.3. MODIFIED PRINCIPAL COMPONENT ANALYSIS

Recall that PCA can only be used for a non-moving sensor due to the fault-like data patterns caused by sensor motion. Hence, the Modified Principle Component Analysis (MPCA) is introduced to allow for fault detection in a dynamic system. This modified method uses a training data set $\hat{\mathbf{x}}_T$, which is the measured IMU outputs from a desired sensor to form the basis of a training set or training vector. From Reference

[8], the parity space concept is implemented, which helps remove the negative effects caused by sensor movement.

A matrix Θ is required in order to formulate the parity space, which is defined as the column space of Θ and is defined by the following conditions: Θ is a positive definite trapezoidal matrix that satisfies the conditions [8]

$$\Theta \mathbf{H} = \mathbf{0}, \quad \Theta \Theta^T = \mathbf{I} \in \mathbb{R}^{n-\zeta \times n-\zeta},$$

and

$$\Theta = \begin{bmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \cdots & \boldsymbol{\theta}_{n-\zeta} \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{\theta}_{c,1} & \boldsymbol{\theta}_{c,2} & \cdots & \boldsymbol{\theta}_{c,n} \end{bmatrix}$$

where \mathbf{H} is the sensor orientation matrix that defines the physical sensor placement and rotation for the distributed system of IMUs, n is the number of sensors used, and ζ is the rank of \mathbf{H} . The first step in computing the Θ matrix is to find the null space of \mathbf{H} , which will be denoted as \mathbf{N} ; that is,

$$\mathbf{N} = \text{null}(\mathbf{H}).$$

Then, to ensure that the needed conditions are satisfied, a QR factorization method [17] is applied to \mathbf{N} and the resulting \mathbf{R} matrix in the factorization is the Θ matrix. Using this method, \mathbf{N} can be decomposed into two matrices \mathbf{Q} and \mathbf{R} . The \mathbf{R} matrix formulated from this null space meets all of the conditions for Θ and is formally defined as

$$\mathbf{N} = \mathbf{Q}\mathbf{R} = \mathbf{Q}\Theta. \tag{4.8}$$

where \mathbf{Q} is an $m \times m$ unitary matrix and \mathbf{R} (or $\mathbf{\Theta}$) is an upper triangular matrix; hence, $\mathbf{\Theta}$ is trapezoidal in this case.

As stated before a training vector is used to remove the dynamic effects in the feature plane. A training parity matrix is found by multiplying a subset of the columns of $\mathbf{\Theta}$ and the training vector as

$$\mathbf{p}_T = \begin{bmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \cdots & \boldsymbol{\theta}_m \end{bmatrix} \hat{\mathbf{x}}_T.$$

This training parity matrix is now a function of sensor movement based on the training vector. The feature vector used to transform the data into the feature plane in the MPCA formulation \mathbf{Y}_T can then be calculated by finding the null space of \mathbf{p}_T to effectively create a matrix that negates dynamic effects and is given as

$$\mathbf{Y}_T = \text{null}(\mathbf{p}_T).$$

A parity matrix, which transforms all measurements into the same orientation space, is computed as

$$\mathbf{p} = \mathbf{\Theta} \begin{bmatrix} \hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_1 & \cdots & \hat{\mathbf{x}}_N \end{bmatrix}^T.$$

The columns of $\mathbf{\Theta}$ are defined as the projections of the direction of measurements onto the parity space. The parity matrix and dynamic null space matrix are used to create a sub-transformed data set, \mathbf{y}^* , which is not a function of sensor movement, and is given as

$$\mathbf{y}^* = \mathbf{Y}_T^T \boldsymbol{\Omega}. \quad (4.9)$$

Now, the typical PCA algorithm can be applied such that the covariance is written as

$$\mathbf{P} = \frac{1}{N} \sum_{i=1}^N (\mathbf{p}_i - \mathbf{p}_T)(\mathbf{p}_i - \mathbf{p}_T)^T.$$

Recall from PCA that the eigenvector decomposition of the covariance matrix is sorted so that the most significant eigenvector, eigenvalue pair rank from highest to lowest, which creates the principal components for the dynamic system. This sorted eigenvector decomposition matrix is the feature matrix, Φ , which gives the feature plane transformation with the utilization of Eq. (4.9) shown as

$$\mathbf{y} = \Phi^T \mathbf{y}^*. \quad (4.10)$$

4.4. TRAINING VECTOR FAILURE

MPCA is proven to work for a dynamic system under the assumption that the training vector does not fail. As stated before, the training vector output is a user-specified IMU that is considered to be the most trusted. However, in a redundant sensor network of homogeneous sensors, the training vector has the same probability of failure as the rest of the sensors. This gives rise to the need for a fault detection method applied specifically to the training vector. The proposed method takes advantage of the previously discussed CI fusion rule within the feature plane data set with reduced dimensionality. This allows for a measurement covariance fusion process that enables faults to be easily detected using a covariance distance thresholding scheme.

4.4.1. Covariance Intersection in the Feature Plane. Recall that the transformed feature plane measurements \mathbf{y}_i , are zero mean even for a dynamical system when using MPCA. This allows for the expected value of the feature plane

measurements to be calculated after using the previous j iterations, where j is a user-specified parameter. Recall from PCA analysis that the measurement covariance is defined as

$$\mathbf{P} = \text{E} \left\{ (\mathbf{x} - \text{E}[\mathbf{x}]) (\mathbf{x} - \text{E}[\mathbf{x}])^T \right\},$$

For MPCA, the training vector can be thought of as the expected value due to the governing dynamic principles. This allows for a covariance to be calculated based on the training vector itself. The expected value of each feature plane measurement can be thought of as the average of the previous j iterations, and the training vector set data that has been transformed into the feature plane is denoted by \mathbf{y}_{tv} . Therefore the feature plane covariance can be expressed as

$$\mathbf{P}_{y,i} = \frac{1}{j} \sum_{\ell=1}^j \left(\mathbf{y}_i^{(k-\ell)} - \mathbf{y}_{tv}^{(k-\ell)} \right) \left(\mathbf{y}_i^{(k-\ell)} - \mathbf{y}_{tv}^{(k-\ell)} \right)^T, \quad (4.11)$$

where j is the number of previous iterations desired to be used in the feature plane covariance calculation and the superscript $(k - \ell)$ denotes the ℓ^{th} previous iteration from the current iteration k . The feature plane covariance based on the training vector is then processed via CI. From Reference [18], the CI rule is derived from the geometric mean density (GMD) fusion rule [5] applied to L Gaussian distributions. The resulting general equations are given in Eqs. (4.12)

$$\tilde{\mathbf{m}}_y = \tilde{\mathbf{P}}_y \sum_{i=1}^L w_i \mathbf{P}_{y,i}^{-1} \mathbf{m}_i \quad \text{and} \quad \tilde{\mathbf{P}}_y = \left[\sum_{i=1}^L w_i \mathbf{P}_{y,i}^{-1} \right]^{-1}, \quad (4.12a)$$

where w_i is the associated weight for each mean and covariance matrix. Note that the fused mean is not needed for this training vector fault detection method as interest is only held in the uncertainty. It is known that the uncertainty for the training vector case $\mathbf{P}_{y,tv}$ will be equal to the null matrix. Assuming no failure in the system, the

uncertainty ellipses will maintain small uncertainty surrounding the $\mathbf{P}_{y,tv}$ uncertainty ellipse which happens to be zero. Therefore, if all failed sensors are accounted for and taken out of the system based on MPCA thresholding, the only time the uncertainty ellipses change in magnitude will be the result of a failure in the training vector. This brings up a need for a thresholding solution.

4.4.2. Kullback-Leibler Divergence Covariance Threshold. The proposed fault detection method for the training vector is left with a need for a failure threshold solution. The detection of a failure for this paper is governed by calculating the distance between covariance matrices by using the Kullback-Leibler divergence. The Kullback-Leibler divergence $d_{KL}(\mathbf{p}||\mathbf{q})$ between the two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined by the integral

$$d_{KL}(\mathbf{p}||\mathbf{q}) = \int_{-\infty}^{\infty} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}$$

For two multivariate normal distributions with means $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ and with covariance matrices \mathbf{A} and \mathbf{B} , the Kullback-Leibler divergence between the two distributions is [19]

$$d_{KL}(\mathbf{p}||\mathbf{q}) = \frac{1}{2} [\text{trace} \{ \mathbf{A}^{-1} \mathbf{B} \} + (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T \mathbf{A} (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B) - k + \ln (|\mathbf{B}|/|\mathbf{A}|)]$$

where k is the dimension of distributions. Assuming equal means, the Kullback-Leibler divergence can then be computed as

$$d(\mathbf{A}, \mathbf{B}) = d_{KL}(\mathbf{p}||\mathbf{q}) = \frac{1}{2} [\text{trace} \{ \mathbf{A}^{-1} \mathbf{B} \} - k + \ln (|\mathbf{B}|/|\mathbf{A}|)] .$$

Using this covariance measure, a fault alarm can be triggered using the following condition:

if $d(\mathbf{A}, \mathbf{B}) \leq \text{threshold}$, then $\mathbf{y}_{tv} = \text{nominal}$

if $d(\mathbf{A}, \mathbf{B}) > \text{threshold}$, then $\mathbf{y}_{tv} = \text{fail}$

If a fault alarm has been triggered, the system can then default to a different IMU to be considered as the training vector.

4.4.3. Shannon Entropy Threshold. The Kullback-Leibler divergence is the thresholding method used in the simulation results; however, it is sometimes useful to examine other thresholding methods. Another method proposed is to measure the Shannon entropy of each covariance matrix. The Shannon entropy is defined by the integral

$$d(\mathbf{A}) = - \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} = E\{-\log p(\mathbf{x})\}$$

This entropy is a measure of the organization or lack of organization of a random variable; i.e., the more spread out a random variable is, the higher its entropy is. For a Gaussian pdf with covariance \mathbf{A} it can be shown that the Shannon entropy is

$$d(\mathbf{A}) = \frac{1}{2} \log |2\pi e \mathbf{A}|.$$

Using this covariance measure, a fault alarm can be triggered using the following condition:

if $d(\mathbf{A}) \leq \text{threshold}$, then $\mathbf{y}_{tv} = \text{nominal}$

if $d(\mathbf{A}) > \text{threshold}$, then $\mathbf{y}_{tv} = \text{fail}$

As stated before, if a fault alarm for the training vector has been triggered, the system can then default to a different IMU to be considered as the training vector. Again, in a homogeneous network of sensors, the training vector is typically chosen arbitrarily, therefore if a training vector fails, a back-up training vector is arbitrarily chosen as well.

5. MEASUREMENT MODELING

Measurement models are developed for Kalman filtering applications needed to complete the navigation algorithm. The sensor models are to be used for spacecraft navigation applications per topics discussed in this section. Position updates will be determined by the processing of position measurements which are used to simulate the performance of a GPS position fix. The attitude updates will be determined by a quaternion measurement resembling a star camera quaternion measurement. The measurements that are modeled and processed here are not realistic in a spacecraft navigation environment or architecture, but are used as a tool to better aid the understanding of redundant IMU fault detection on-board.

5.1. POSITION MEASUREMENT MODELING

The position of the spacecraft, or sensor, in the inertial frame, \mathbf{r}_{sc}^i , is constructed by the addition position of the IMU in the inertial frame, \mathbf{r}_{imu}^i , and the position of the CM of the spacecraft with respect to the IMU in the body frame $\mathbf{r}_{s/imu}^b$. In addition, $\mathbf{r}_{s/imu}^b$ must be rotated from the body frame into the inertial frame by the quaternion rotation given as \mathbf{T}_b^i . This gives the final form of the position as

$$\mathbf{r}_{sc}^i = \mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b. \quad (5.1)$$

However, the position measurement will be corrupted by a zero-mean, Gaussian white-noise sequence \mathbf{v}_r . Therefore, the position of the spacecraft \mathbf{r}_{sc}^i and its corresponding estimate $\hat{\mathbf{r}}_{sc}^i$ are defined as

$$\mathbf{r}_{sc}^i = \mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b + \mathbf{v}_r$$

$$\hat{\mathbf{r}}_{sc}^i = \hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b.$$

Note that $\mathbf{r}_{s/imu}^b$ is assumed to be deterministic; therefore, it does not need to be estimated. The error in the estimate is then needed in the construction of the extended Kalman filter. The error estimate can be shown by the difference equation of \mathbf{r}_{sc}^i and $\hat{\mathbf{r}}_{sc}^i$ given as

$$\delta \mathbf{r}_{sc}^i = (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b + \mathbf{w}_r) - \left(\hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b \right).$$

Rearranging terms and defining the error of $\delta \mathbf{r}_{imu}^i$ as

$$\delta \mathbf{r}_{imu}^i = \mathbf{r}_{imu}^i - \hat{\mathbf{r}}_{imu}^i,$$

the error in the position of the spacecraft can be written as

$$\delta \mathbf{r}_{sc}^i = \delta \mathbf{r}_{imu}^i + \left\{ \mathbf{T}_b^i - \hat{\mathbf{T}}_b^i \right\} \mathbf{r}_{s/imu}^b + \mathbf{v}_r.$$

The true rotation matrix, \mathbf{T}_b^i , can be expanded using a first-order Taylor series expansion about its estimate as

$$\mathbf{T}_b^i = \hat{\mathbf{T}}_b^i - \hat{\mathbf{T}}_b^i \left[\delta \boldsymbol{\theta}_b^i \times \right]$$

where $\delta \boldsymbol{\theta}_b^i$ is defined as the attitude error and $[\delta \boldsymbol{\theta}_b^i \times]$ is defined as the skew symmetric cross product matrix of the attitude deviation. Plugging in this relation, and reversing the cross product matrix, the final position error is given as

$$\delta \mathbf{r}_{sc}^i = \delta \mathbf{r}_{imu}^i - \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta \boldsymbol{\theta}_b^i + \mathbf{v}_r. \quad (5.2)$$

It is then seen that the sensitivity of the position measurement has components associated with position error, attitude error and measurement noise.

5.2. QUATERNION MEASUREMENT MODELING

The quaternion attitude state of a spacecraft representing the orientation of the inertial frame with respect to the body frame, $\bar{\mathbf{q}}_i^b$, is considered to be given as a measurement by some means. The measured quaternion is taken to be the true quaternion corrupted by a zero-mean, Gaussian white-noise sequence, \mathbf{v}_θ , that is multiplicatively injected by the quaternion multiplication. Therefore, the quaternion and its respective estimate are given as

$$\bar{\mathbf{z}}_i^b = \bar{\mathbf{q}}(\mathbf{v}_\theta) \otimes \bar{\mathbf{q}}_i^b, \quad \text{and} \quad \hat{\mathbf{z}}_i^b = \hat{\mathbf{q}}_i^b$$

where $\bar{\mathbf{q}}(\mathbf{v}_\theta)$ is defined to as

$$\bar{\mathbf{q}}(\mathbf{v}_\theta) = \begin{bmatrix} \sin \frac{\|\mathbf{v}_\theta\|}{2} \frac{\mathbf{v}_\theta}{\|\mathbf{v}_\theta\|} \\ \cos \frac{\|\mathbf{v}_\theta\|}{2} \end{bmatrix}.$$

The difference between the true and estimated measurements in terms of its rotation angle is then desired and can be computed by taking twice the vector part of the quaternion multiplication between the quaternion measurement and the inverse of the estimate. The term $\bar{\mathbf{q}}_i^b \otimes \hat{\mathbf{q}}_i^b$ is well approximated as a small angle rotation quaternion giving the approximation

$$\bar{\mathbf{z}} \otimes (\hat{\mathbf{z}})^{-1} \approx \left[\frac{1}{2} \delta \mathbf{z} \right].$$

Therefore, taking twice the vector part of this assumption gives the attitude error estimate solely in terms of its error which is given as

$$\delta \mathbf{z} = 2\text{vec} \left(\bar{\mathbf{q}}_i^b \otimes (\hat{\mathbf{q}}_i^b)^{-1} \right). \quad (5.3)$$

5.3. RANGE MODELING

The range, ρ , by definition, is the magnitude of the difference between two position vectors. For this case, the two position vectors will be the vehicle's position in the inertial frame \mathbf{r}_s^i , and the position vector of a transmission receiving ground station \mathbf{r}_g^i in the inertial frame, which is assumed to be known. The position vector difference in the inertial frame is determined by

$$\begin{aligned} \delta \mathbf{r}^i &= \mathbf{r}_s^i - \mathbf{r}_g^i \\ &= \mathbf{r}_s^i - \mathbf{T}_f^i \mathbf{r}_g^f \\ &= \mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \end{aligned}$$

Then the range is shown as

$$\begin{aligned} h_\rho(\mathbf{x}) &= \|\delta \mathbf{r}^i\|_2 \\ &= \sqrt{(\delta \mathbf{r}^i)^T (\delta \mathbf{r}^i)}. \end{aligned}$$

The range equation can be expanded as

$$\rho = \sqrt{\left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right)^T \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right)}. \quad (5.4)$$

To simplify notation, let

$$\mathbf{r}_{s/g}^i = \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right)$$

where $\mathbf{r}_{s/g}^i$ is defined as the position vector of the range sensor to the ground station in the inertial frame. However the range sensor measurements will be corrupted by a zero-mean, Gaussian white-noise sequence, v_ρ . Therefore, the range measurement \mathbf{z}_ρ , and its corresponding estimate $\hat{\mathbf{z}}_\rho$, are defined as

$$\mathbf{z}_\rho = \sqrt{\left(\mathbf{r}_{s/g}^i\right)^T \left(\mathbf{r}_{s/g}^i\right)} + w_\rho \quad , \quad \hat{\mathbf{z}}_\rho = \sqrt{\left(\hat{\mathbf{r}}_{s/g}^i\right)^T \left(\hat{\mathbf{r}}_{s/g}^i\right)},$$

where

$$\hat{\mathbf{r}}_{s/g}^i = \left(\hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right).$$

Now, the difference between the true measurement and its estimate is considered as

$$\delta \mathbf{z}_\rho = \mathbf{z}_\rho - \hat{\mathbf{z}}_\rho.$$

Expanding the first term in a first-order Taylor series expansion about the estimated states gives

$$\delta \mathbf{z}_\rho = \left[\hat{\mathbf{z}}_\rho + \mathbf{f}_\rho \delta \mathbf{r}_{s/g}^i + v_\rho \right] - \hat{\mathbf{z}}_\rho = \mathbf{f}_\rho \delta \mathbf{r}_{s/g}^i,$$

where \mathbf{f}_ρ is the partial derivative of range with respect to the position of the range sensor with respect to the ground station, which is defined as

$$\mathbf{f}_\rho = \frac{\partial \rho}{\partial \mathbf{r}_{s/g}^i} = \frac{\left(\mathbf{r}_{s/g}^i\right)^T}{\sqrt{\left(\mathbf{r}_{s/g}^i\right)^T \left(\mathbf{r}_{s/g}^i\right)}}$$

with its respective deviation, $\delta \mathbf{r}_{s/g}^i$, defined as

$$\delta \mathbf{r}_{s/g}^i = \mathbf{r}_{s/g}^i - \hat{\mathbf{r}}_{s/g}^i = \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right) - \left(\hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right).$$

The range deviation expression is then simplified to be

$$\delta z_\rho = \mathbf{f}_\rho \delta \mathbf{r}_{imu}^i + \mathbf{f}_\rho \left[\mathbf{T}_b^i - \hat{\mathbf{T}}_b^i \right] \mathbf{r}_{s/imu}^b + v_\rho.$$

Then, using the relation to obtain the first-order approximation of the body to inertial transformation matrix in terms of its estimate, which is shown as

$$\mathbf{T}_b^i = \hat{\mathbf{T}}_b^i - \hat{\mathbf{T}}_b^i \left[\delta \boldsymbol{\theta}_b^i \times \right]$$

and plugging this relation back into δz_ρ gives the final form of the range deviation as

$$\begin{aligned} \delta z_\rho &= \mathbf{f}_\rho \delta \mathbf{r}_{imu}^i - \mathbf{f}_\rho \hat{\mathbf{T}}_b^i \left[\delta \boldsymbol{\theta}_b^i \times \right] \mathbf{r}_{s/imu}^b + v_\rho \\ &= \mathbf{f}_\rho \delta \mathbf{r}_{imu}^i + \mathbf{f}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta \boldsymbol{\theta}_b^i + v_\rho. \end{aligned} \quad (5.5)$$

It is then seen that the sensitivity of the range measurement has components associated with position error, attitude error and measurement noise

5.4. RANGE RATE MODELING

The range-rate, $\dot{\rho}$, is found by taking the time derivative of Equation 5.4. To simplify this derivation, both sides of the range equation are squared to give

$$\rho^2 = \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right)^T \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right).$$

Then, differentiating ρ^2 with respect to time yields

$$2\rho\dot{\rho} = 2 \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right)^T \frac{d}{dt} \left\{ \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f \right) \right\},$$

which can be simplified to

$$\rho\dot{\rho} = (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f)^T \frac{d}{dt} \{ (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f) \}. \quad (5.6)$$

Now, the variable $\mathbf{v}_{s/g}^i$ is defined to be

$$\mathbf{v}_{s/g}^i = \frac{d}{dt} \{ (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f) \} = \dot{\mathbf{r}}_{imu}^i + \dot{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b + \mathbf{T}_b^i \dot{\mathbf{r}}_{s/imu}^b - \dot{\mathbf{T}}_f^i \mathbf{r}_g^f - \mathbf{T}_f^i \dot{\mathbf{r}}_g^f,$$

where the time derivative of the two attitude matrices, defined in Reference [20], are given as

$$\begin{aligned} \dot{\mathbf{T}}_b^i &= \left(\dot{\mathbf{T}}_i^b \right)^T & \dot{\mathbf{T}}_f^i &= \left(\dot{\mathbf{T}}_i^f \right)^T \\ &= \left(-[\boldsymbol{\omega}_{b/i}^b \times] \mathbf{T}_i^b \right)^T & &= \left(-[\boldsymbol{\omega}_{f/i}^f \times] \mathbf{T}_i^f \right)^T \\ &= \left(\mathbf{T}_i^b \right)^T \left(-[\boldsymbol{\omega}_{b/i}^b \times] \right)^T & &= \left(\mathbf{T}_i^f \right)^T \left(-[\boldsymbol{\omega}_{f/i}^f \times] \right)^T \\ &= \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] & &= \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times]. \end{aligned}$$

Additionally, $\mathbf{r}_{s/imu}^b$ and \mathbf{r}_g^f are constant, known position vectors; therefore,

$$\dot{\mathbf{r}}_{s/imu}^b = \mathbf{0} \quad \text{and} \quad \dot{\mathbf{r}}_g^f = \mathbf{0}$$

and $\mathbf{v}_{s/g}^i$ may be written as

$$\mathbf{v}_{s/g}^i = \left(\mathbf{v}_{imu}^i + \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b + \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f \right)$$

The range rate can now be expressed as

$$\dot{\rho} = \frac{1}{\rho} (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f)^T \left(\mathbf{v}_{imu}^i + \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b + \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f \right)$$

Plugging back in all needed relations into the above equation yields the final form for range rate given by

$$\dot{\rho} = \frac{\left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f\right)^T \left(\mathbf{v}_{imu}^i + \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f\right)}{\sqrt{\left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f\right)^T \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f\right)}} \quad (5.7)$$

To simplify this, recall that

$$\begin{aligned} \mathbf{r}_{s/g}^i &= \left(\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f\right) \\ \mathbf{v}_{s/g}^i &= \left(\mathbf{v}_{imu}^i + \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f\right), \end{aligned}$$

where $\mathbf{r}_{s/g}^i$ is defined as the position vector of the range sensor to the ground station in the inertial frame and $\mathbf{v}_{s/g}^i$ is the velocity vector of the range sensor with respect to the ground station in the inertial frame. However, the sensor measurements will be corrupted by a zero-mean, Gaussian white-noise sequence, $v_{\dot{\rho}}$. Therefore, $\mathbf{z}_{\dot{\rho}}$ and its corresponding estimate $\hat{\mathbf{z}}_{\dot{\rho}}$, are defined as

$$\mathbf{z}_{\dot{\rho}} = \frac{\mathbf{r}_{s/g}^i \mathbf{v}_{s/g}^i}{\sqrt{\left(\mathbf{r}_{s/g}^i\right)^T \left(\mathbf{r}_{s/g}^i\right)}} + v_{\dot{\rho}} \quad , \quad \hat{\mathbf{z}}_{\dot{\rho}} = \frac{\hat{\mathbf{r}}_{s/g}^i \hat{\mathbf{v}}_{s/g}^i}{\sqrt{\left(\hat{\mathbf{r}}_{s/g}^i\right)^T \left(\hat{\mathbf{r}}_{s/g}^i\right)}}$$

where

$$\begin{aligned} \hat{\mathbf{r}}_{s/g}^i &= \left(\hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f\right) \\ \hat{\mathbf{v}}_{s/g}^i &= \left(\hat{\mathbf{v}}_{imu}^i + \hat{\mathbf{T}}_b^i [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f\right). \end{aligned}$$

Then the deviation of the range-rate measurement from its estimate can be shown by taking the difference between $\mathbf{z}_{\dot{\rho}}$ and $\hat{\mathbf{z}}_{\dot{\rho}}$ and expanding the first term in a first-order

Taylor series expansion about the estimate gives

$$\delta \mathbf{z}_{\dot{\rho}} = \mathbf{z}_{\dot{\rho}} - \hat{\mathbf{z}}_{\dot{\rho}} = [\hat{\mathbf{z}}_{\dot{\rho}} + \mathbf{f}_{\dot{\rho}} \delta \mathbf{r}_{s/g}^i + \mathbf{g}_{\dot{\rho}} \delta \mathbf{v}_{s/g}^i + v_{\dot{\rho}}] - \hat{\mathbf{z}}_{\dot{\rho}} = \mathbf{f}_{\dot{\rho}} \delta \mathbf{r}_{s/g}^i + \mathbf{g}_{\dot{\rho}} \delta \mathbf{v}_{s/g}^i + v_{\dot{\rho}}. \quad (5.8)$$

The partial of $\mathbf{r}_{s/g}^i$ with respect to range-rate can be solved in a rather straightforward manner as

$$\begin{aligned} \mathbf{f}_{\dot{\rho}} &= \frac{\mathbf{v}_{s/g}^i{}^T}{\sqrt{(\mathbf{r}_{s/g}^i)^T (\mathbf{r}_{s/g}^i)}} \left[\mathbf{I}_{3 \times 3} - \frac{\mathbf{r}_{s/g}^i \mathbf{r}_{s/g}^i{}^T}{(\mathbf{r}_{s/g}^i)^T (\mathbf{r}_{s/g}^i)} \right] \\ &= \frac{\mathbf{v}_{s/g}^i{}^T}{\rho} \left[\mathbf{I}_{3 \times 3} - \frac{\mathbf{r}_{s/g}^i \mathbf{r}_{s/g}^i{}^T}{\rho^2} \right]. \end{aligned}$$

Likewise, the partial of range-rate with respect to $\mathbf{v}_{s/g}$ is given by

$$\mathbf{g}_{\dot{\rho}} = \frac{\mathbf{r}_{s/g}^i{}^T}{\sqrt{(\mathbf{r}_{s/g}^i)^T (\mathbf{r}_{s/g}^i)}} = \frac{\mathbf{r}_{s/g}^i{}^T}{\rho}.$$

The terms $\delta \mathbf{r}_{s/g}^i$ and $\delta \mathbf{v}_{s/g}^i$ in Equation 5.8 are given as

$$\begin{aligned} \delta \mathbf{r}_{s/g}^i &= \mathbf{r}_{s/g}^i - \hat{\mathbf{r}}_{s/g}^i = (\mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f) - (\hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \mathbf{r}_g^f) \\ \delta \mathbf{v}_{s/g}^i &= \mathbf{v}_{s/g}^i - \hat{\mathbf{v}}_{s/g}^i \\ &= \left(\mathbf{v}_{imu}^i + \mathbf{T}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f \right) - \left(\hat{\mathbf{v}}_{imu}^i + \hat{\mathbf{T}}_b^i [\boldsymbol{\omega}_{b/i}^b \times] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i [\boldsymbol{\omega}_{f/i}^f \times] \mathbf{r}_g^f \right). \end{aligned}$$

Using the relation to obtain the first order approximation of the body to inertial transformation matrix in terms of its estimate, which is shown as

$$\mathbf{T}_b^i = \hat{\mathbf{T}}_b^i + \hat{\mathbf{T}}_b^i [\delta \boldsymbol{\theta}_b^i \times].$$

The term $\mathbf{v}_{s/g}^i$ can then be simplified to

$$\begin{aligned}\mathbf{v}_{s/g}^i &= (\hat{\mathbf{v}}_{imu}^i + \delta\mathbf{v}_{imu}^i) + \left(\hat{\mathbf{T}}_b^i + \hat{\mathbf{T}}_b^i [\delta\boldsymbol{\theta}_b^i \times] \right) \left[(\hat{\boldsymbol{\omega}}_{b/i}^b + \delta\boldsymbol{\omega}_{b/i}^b) \times \right] \mathbf{r}_{s/imu}^b - \mathbf{T}_f^i \left[\boldsymbol{\omega}_{f/i}^f \times \right] \mathbf{r}_g^f \\ &= (\hat{\mathbf{v}}_{imu}^i + \delta\mathbf{v}_{imu}^i) + \hat{\mathbf{T}}_b^i \left[\hat{\boldsymbol{\omega}}_{b/i}^b \times \right] \mathbf{r}_{s/imu}^b - \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{imu}^b \times \right] \delta\boldsymbol{\omega}_{b/i}^b - \hat{\mathbf{T}}_{b/i}^b \left[(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b) \right] \delta\boldsymbol{\theta}_b^i.\end{aligned}$$

Subtracting the estimate of the sensor velocity with respect to the ground station,

$\delta\mathbf{v}_{s/g}^i$ becomes

$$\delta\mathbf{v}_{s/g}^i = \delta\mathbf{v}_{imu}^i - \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta\boldsymbol{\omega}_{b/i}^b - \hat{\mathbf{T}}_b^i \left[(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b) \times \right] \delta\boldsymbol{\theta}_b^i.$$

The last step is to plug $\delta\mathbf{r}_{s/g}^i$ and $\delta\mathbf{v}_{s/g}^i$ back into the measurement variation. It is then shown that $\delta\dot{\rho}$ becomes

$$\delta\dot{z}_\rho = \left(\mathbf{R}_\rho \delta\mathbf{r}_{imu}^i + \mathbf{R}_\rho \left[\mathbf{T}_b^i - \hat{\mathbf{T}}_b^i \right] \mathbf{r}_{s/imu}^b \right) + \mathbf{V}_\rho \delta\mathbf{v}_{imu}^i - \mathbf{V}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta\boldsymbol{\omega}_{b/i}^b \quad (5.9)$$

$$- \mathbf{V}_\rho \hat{\mathbf{T}}_b^i \left[(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b) \times \right] \delta\boldsymbol{\theta}_b^i + w_\rho$$

$$= \left(\mathbf{R}_\rho \delta\mathbf{r}_{imu}^i - \mathbf{R}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta\boldsymbol{\theta}_b^i \right) + \left(\mathbf{V}_\rho \delta\mathbf{v}_{imu}^i - \mathbf{V}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta\boldsymbol{\omega}_{b/i}^b \right) \quad (5.10)$$

$$- \mathbf{V}_\rho \hat{\mathbf{T}}_b^i \left[(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b) \times \right] \delta\boldsymbol{\theta}_b^i + w_\rho. \quad (5.11)$$

5.5. UNIT VECTOR STAR CAMERA MODELING

By imaging stars and mapping them to a celestial reference, the unit vector to a star is measured as a function of the declination and right ascension of the star. A practical and common assumption is made that the stars are at infinity, meaning that the position of the star in the inertial frame is much greater than the position of the IMU in the inertial frame and the position of the star camera to the IMU i.e. $\mathbf{r}_s \gg \mathbf{r}_{imu}$ and $\mathbf{r}_s \gg \mathbf{r}_{sc/imu}$. Note that the imaged star and the star camera are denoted by the subscripts s and sc . This measurement model follows the derivation found Reference [12]. The position of the star from the star camera in the body frame

is then given by

$$\mathbf{r}_{s/sc}^b = \mathbf{T}_i^b \mathbf{r}_s^i.$$

A new reference frame is now introduced as the “celestial reference” frame (cr) in which the star’s coordinates are written. The celestial reference frame is considered to be at the center of the Earth. Therefore, the above equation can be written as

$$\mathbf{r}_{s/sc}^b = \mathbf{T}_i^b \mathbf{T}_{cr}^i \mathbf{r}_s^{cr}.$$

Also, note that $\|\mathbf{r}_{s/sc}^b\| = \|\mathbf{r}_s^{cr}\|$. The model of the star camera can then be written as

$$\mathbf{u}_{s/sc}^{sc} = \mathbf{T}_b^{sc} \mathbf{T}_i^b \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr}$$

However, the star camera measurements will be corrupted by a zero-mean, Gaussian, white-noise sequence, \mathbf{v}_{sc}^{sc} . Therefore, the practical use of the star camera measurement model is written as

$$\mathbf{z}_{sc} = \mathbf{T}_b^{sc} \mathbf{T}_i^b \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr} + \mathbf{v}_{sc}.$$

The estimated star camera measurement can be written as

$$\hat{\mathbf{z}}_{sc} = \mathbf{T}_b^{sc} \hat{\mathbf{T}}_i^b \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr}. \quad (5.12)$$

Now the deviation of the star camera measurement is examined and can be written as

$$\delta \mathbf{z}_{sc} = \mathbf{z}_{sc} - \hat{\mathbf{z}}_{sc} = \mathbf{T}_b^{sc} \left[\mathbf{T}_i^b - \hat{\mathbf{T}}_i^b \right] \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr} + \mathbf{v}_{sc}^{sc}.$$

Then applying a first-order Taylor series expansion and rearranging cross product terms introduces the measurement deviation as a function of attitude error, $\boldsymbol{\theta}_b^i$, as

$$\delta \mathbf{z}_{sc} = \mathbf{T}_b^{sc} \left[\hat{\mathbf{T}}_i^b \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr} \times \right] \delta \boldsymbol{\theta}_b^i + \mathbf{v}_{sc}^{sc} \quad (5.13)$$

Attitude error and star-camera measurement noise are the only sensitivity associated components of the star camera measurement.

6. NAVIGATION ALGORITHM

6.1. THE DISCRETE EXTENDED KALMAN FILTER

It is known that the Kalman filter operates on linear dynamical systems; however, spacecraft two-body dynamics obey nonlinear differential equations. Therefore the Kalman filter cannot be used, but the extended Kalman filter (EKF) makes use of linearization to handle these nonlinearities. The discrete EKF is derived due to the discrete nature of the dead-reckoning equations. The evolutionary equations for the propagation of the mean and covariance are to be examined, followed by the development of the mean and covariance update relationships. Consider a discrete dynamical system that is assumed to have the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1}$$

where $\mathbf{f}(\cdot)$ is the nonlinear dynamics that governs the transition of the state from \mathbf{x}_{k-1} to \mathbf{x}_k and \mathbf{M}_{k-1} maps the zero-mean white noise sequence \mathbf{w}_{k-1} into the dynamics. The initial state is taken to have mean \mathbf{m}_0 and covariance \mathbf{P}_0 , and \mathbf{w}_{k-1} has covariance \mathbf{Q}_{k-1} .

6.1.1. Mean and Covariance Propagation. First, the mean propagation is examined by taking the expected value of the the dynamical system form giving

$$\mathbf{m}_k = \mathbb{E}\{\mathbf{x}_k\} = \mathbb{E}\{\mathbf{f}(\mathbf{x}_{k-1} + \mathbf{M}_{k-1}\mathbf{w}_{k-1})\} = \mathbb{E}\{\mathbf{f}(\mathbf{x}_{k-1})\} + \mathbb{E}\{\mathbf{M}_{k-1}\mathbf{w}_{k-1}\}.$$

The mapping matrix \mathbf{M}_{k-1} is assumed to be deterministic, which allows it to be moved outside of the expectation. The mean can then be written as

$$\mathbf{m}_k = \mathbb{E}\{\mathbf{f}(\mathbf{x}_{k-1})\} + \mathbf{M}_{k-1}\mathbb{E}\{\mathbf{w}_{k-1}\}$$

It is also assumed that the process noise \mathbf{w}_{k-1} has zero-mean, i.e. $E\{\mathbf{w}_{k-1}\} = \mathbf{0}$, which gives the mean at time t_k as

$$\mathbf{m}_k = E\{\mathbf{f}(\mathbf{x}_{k-1})\}. \quad (6.1)$$

The nonlinear dynamics are then expanded by using a first-order Taylor series expansion about the current mean and higher-order terms are neglected, giving

$$\mathbf{f}(\mathbf{x}_{k-1}) = \mathbf{f}(\mathbf{m}_{k-1}) + \mathbf{F}(\mathbf{m}_{k-1})(\mathbf{x}_{k-1} - \mathbf{m}_{k-1})$$

where $\mathbf{F}(\mathbf{m}_{k-1})$ is the Jacobian of the nonlinear dynamics as a function of the prior mean, which linearizes the system, and is defined as

$$\mathbf{F}(\mathbf{x}_{k-1}) = \left[\frac{\partial \mathbf{f}(\mathbf{x}_{k-1})}{\partial \mathbf{x}_{k-1}} \Big|_{\mathbf{x}_{k-1}=\mathbf{m}_{k-1}} \right].$$

Using this expansion, Eq. (6.1) can be written as

$$\mathbf{m}_k = E\{\mathbf{f}(\mathbf{m}_{k-1})\} + E\{\mathbf{F}(\mathbf{m}_{k-1})(\mathbf{x}_{k-1} - \mathbf{m}_{k-1})\}.$$

Assuming that the mean at time $k-1$ is deterministic, $\mathbf{f}(\mathbf{m}_{k-1})$ is deterministic along with its Jacobian. Using these assumptions, the mean at time t_k can be written as

$$\mathbf{m}_k = \mathbf{f}(\mathbf{m}_{k-1}) + \mathbf{F}(\mathbf{m}_{k-1})E\{(\mathbf{x}_{k-1} - \mathbf{m}_{k-1})\}.$$

Finally, provided that the estimator is unbiased at time $k-1$ then $(\mathbf{x}_{k-1} - \mathbf{m}_{k-1})$ is zero-mean, i.e. $E\{(\mathbf{x}_{k-1} - \mathbf{m}_{k-1})\} = \mathbf{0}$, which gives the final form of the propagation equation for the mean as

$$\mathbf{m}_k = \mathbf{f}(\mathbf{m}_{k-1}). \quad (6.2)$$

Estimation errors are now defined at time $k - 1$ and k to be

$$\mathbf{e}_{k-1} = \mathbf{x}_{k-1} - \mathbf{m}_{k-1} \quad \text{and} \quad \mathbf{e}_k = \mathbf{x}_k - \mathbf{m}_k.$$

Starting with \mathbf{e}_k , a difference equation for the estimation error is given by considering the governing relationships for \mathbf{x}_k and \mathbf{m}_k in terms of \mathbf{x}_{k-1} , \mathbf{w}_{k-1} , and \mathbf{m}_{k-1} shown as

$$\begin{aligned} \mathbf{e}_k &= \mathbf{x}_k - \mathbf{m}_k \\ &= [\mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1}] - [\mathbf{f}(\mathbf{m}_{k-1})] \\ &= [\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{f}(\mathbf{m}_{k-1})] + \mathbf{M}_{k-1}\mathbf{w}_{k-1} \end{aligned}$$

which shows the evolution of the estimation error through discrete time steps. Noting that

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1}, \quad \text{and} \quad \mathbf{m}_{k-1} = \mathbf{f}(\mathbf{x}_{k-1}),$$

the estimation error at time k can be written as

$$\mathbf{e}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1} - \mathbf{F}(\mathbf{m}_{k-1})\mathbf{m}_{k-1}.$$

By combining like terms of $\mathbf{F}(\mathbf{m}_{k-1})$ it is seen that the estimation error at time $k - 1$ can be substituted into \mathbf{e}_k as

$$\mathbf{e}_k = \mathbf{F}(\mathbf{m}_{k-1})\mathbf{e}_{k-1} + \mathbf{M}_{k-1}\mathbf{w}_{k-1}. \quad (6.3)$$

From Eq. (6.3), the covariance propagation equation can be developed from the definitions of the state estimation error covariances at times $k - 1$ and k which are given

as

$$\mathbf{P}_{k-1} = \mathbb{E} \{ \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \} \quad \text{and} \quad \mathbf{P}_k = \mathbb{E} \{ \mathbf{e}_k \mathbf{e}_k^T \}.$$

The product of the propagated error with its transposed is found by substituting in Eq. (6.3) which gives

$$\begin{aligned} \mathbf{e}_k \mathbf{e}_k^T &= [\mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} + \mathbf{M}_{k-1} \mathbf{w}_{k-1}] [\mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} + \mathbf{M}_{k-1} \mathbf{w}_{k-1}]^T \\ &= \mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \mathbf{F}(\mathbf{m}_{k-1})^T + \mathbf{M}_{k-1} \mathbf{w}_{k-1} \mathbf{e}_{k-1}^T \mathbf{F}(\mathbf{m}_{k-1})^T \\ &\quad + \mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} \mathbf{w}_{k-1}^T \mathbf{M}_{k-1}^T + \mathbf{M}_{k-1} \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \mathbf{M}_{k-1}^T \end{aligned}$$

In order to formulate the error covariance, the expected value of $\mathbf{e}_k \mathbf{e}_k^T$ shown above is taken as

$$\begin{aligned} \mathbf{P}_k &= \mathbb{E} \{ \mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \mathbf{F}(\mathbf{m}_{k-1})^T \} + \mathbb{E} \{ \mathbf{M}_{k-1} \mathbf{w}_{k-1} \mathbf{e}_{k-1}^T \mathbf{F}(\mathbf{m}_{k-1})^T \} \\ &\quad + \mathbb{E} \{ \mathbf{F}(\mathbf{m}_{k-1}) \mathbf{e}_{k-1} \mathbf{w}_{k-1}^T \mathbf{M}_{k-1}^T \} + \mathbb{E} \{ \mathbf{M}_{k-1} \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \mathbf{M}_{k-1}^T \} \end{aligned}$$

and knowing that $\mathbf{F}(\mathbf{m}_{k-1})$ and \mathbf{M}_{k-1} are deterministic, they are pulled outside of the expectations shown as

$$\begin{aligned} \mathbf{P}_k &= \mathbf{F}(\mathbf{m}_{k-1}) \mathbb{E} \{ \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \} \mathbf{F}(\mathbf{m}_{k-1})^T + \mathbf{M}_{k-1} \mathbb{E} \{ \mathbf{w}_{k-1} \mathbf{e}_{k-1}^T \} \mathbf{F}(\mathbf{m}_{k-1})^T \\ &\quad + \mathbf{F}(\mathbf{m}_{k-1}) \mathbb{E} \{ \mathbf{e}_{k-1} \mathbf{w}_{k-1}^T \} \mathbf{M}_{k-1}^T + \mathbf{M}_{k-1} \mathbb{E} \{ \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \} \mathbf{M}_{k-1}^T. \end{aligned}$$

The process noise is assumed to be uncorrelated to the state; therefore,

$$\mathbb{E} \{ \mathbf{w}_{k-1} \mathbf{e}_{k-1}^T \} = \mathbf{0}, \quad \text{and} \quad \mathbb{E} \{ \mathbf{w}_{k-1} \mathbf{e}_{k-1} \} = \mathbf{0}$$

and using the knowledge that $\mathbf{P}_{k-1} = \text{E} \{ \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \}$, the covariance propagation equation arrives at its final form as

$$\mathbf{P} = \mathbf{F}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1} \mathbf{F}(\mathbf{m}_{k-1})^T + \mathbf{M}_{k-1} \mathbf{Q}_{k-1} \mathbf{M}_{k-1}^T. \quad (6.4)$$

where $\mathbf{Q}_{k-1} = \text{E} \{ \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \}$.

6.1.2. Mean and Covariance Update. Now, the measurement update is examined. The measurement is taken to be of the form

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

where \mathbf{v}_k is the measurement noise which is assumed to be a zero mean white-noise sequence with covariance \mathbf{R}_k ; that is,

$$\text{E} \{ \mathbf{v}_k \} = \mathbf{0} \quad \text{and} \quad \text{E} \{ \mathbf{v}_k \mathbf{v}_k^T \} = \mathbf{R}_k.$$

The *a priori* mean and covariance given before the addition of new information are denoted by \mathbf{m}_k^- and \mathbf{P}_k^- , respectively, which are defined to be the outputs of the propagation stage. For the Kalman filter, the *a posteriori* mean is given by a linear combination of the *a priori* data given as

$$\mathbf{m}_k^+ = \mathbf{N}_k \mathbf{m}_k^- + \mathbf{K} \mathbf{z}_k.$$

However, this equation is required to make use of the fact that the measurement is linear with respect to the state. To avoid this, an update is considered that effectively replaces the linear function of the prior mean with a constant vector \mathbf{a}_k , and has the form

$$\mathbf{m}_k^+ = \mathbf{a}_k + \mathbf{K}_k \mathbf{z}_k. \quad (6.5)$$

This still keeps a portion of the update as being a linear function of the data as well, but it allows nonlinear functions such as those used in the EKF to be considered. The prior and posterior errors are then defined to be

$$\mathbf{e}_k^- = \mathbf{x}_k - \mathbf{m}_k^- \quad \text{and} \quad \mathbf{e}_k^+ = \mathbf{x}_k - \mathbf{m}_k^+$$

which, in a rather straightforward manner, allows for the posterior mean to be written as

$$\mathbf{m}_k^+ = \mathbf{m}_k^- + \mathbf{e}_k^- - \mathbf{e}_k^+. \quad (6.6)$$

Now, the update equation is given by substituting Eq. (6.6) in to Eq. (6.5), which yields

$$\mathbf{m}_k^- + \mathbf{e}_k^- - \mathbf{e}_k^+ = \mathbf{a}_k + \mathbf{K} \mathbf{z}_k.$$

By taking the expected value of this equation as

$$\mathbb{E} \{ \mathbf{m}_k^- + \mathbf{e}_k^- - \mathbf{e}_k^+ \} = \mathbb{E} \{ \mathbf{a}_k + \mathbf{K} \mathbf{z}_k \}$$

and using the assumption of an unbiased prior mean, while enforcing a desired unbiased posterior and that \mathbf{K}_k is deterministic, yields

$$\mathbf{m}_k^- = \mathbf{a}_k + \mathbf{K}_k \hat{\mathbf{z}}_k \quad \text{where} \quad \hat{\mathbf{z}}_k = \mathbb{E} \{ \mathbf{z}_k \}.$$

Now, \mathbf{a}_k can be solved for such that for an unbiased posterior estimate is guaranteed, giving

$$\mathbf{a}_k = \mathbf{m}_k^- - \mathbf{K}_k \hat{\mathbf{z}}_k$$

and substituting this relationship for \mathbf{a}_k into Eq. (6.5) and simplifying gives the posterior mean as

$$\mathbf{m}_k^+ = \mathbf{m}_k^- + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k]. \quad (6.7)$$

Note that no measurement linearity needs to be specified for the mean update equation to hold. The covariance update is needed to be described, which follows from the manner in which the error gets updated. The *a posteriori* state estimation error is defined to be

$$\mathbf{e}_k^+ = \mathbf{e}_k^- - \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k),$$

and then the prior and posterior covariance can be defined respectively as

$$\mathbf{P}_k^- = \mathbb{E} \{(\mathbf{e}_k^-)(\mathbf{e}_k^-)^T\} \quad \text{and} \quad \mathbf{P}_k^+ = \mathbb{E} \{(\mathbf{e}_k^+)(\mathbf{e}_k^+)^T\}.$$

Using the definition of the posterior error shown above, the posterior covariance can be written as the expected value of the product of posterior error and its transpose, or

$$\begin{aligned} \mathbf{P}_k^+ &= \mathbb{E} \{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} - \mathbb{E} \{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \mathbf{K}_k^T\} \\ &\quad - \mathbb{E} \{\mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} + \mathbb{E} \{\mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \mathbf{K}_k^T\}. \end{aligned}$$

Again, assuming that the gain matrix is deterministic, the posterior covariance can then be written as

$$\begin{aligned} \mathbf{P}_k^+ &= \mathbb{E} \{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} - \mathbb{E} \{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T\} \mathbf{K}_k^T \\ &\quad - \mathbf{K}_k \mathbb{E} \{(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} + \mathbf{K}_k \mathbb{E} \{(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T\} \mathbf{K}_k^T. \end{aligned}$$

Recall that the prior covariance can be written as

$$\mathbf{P}_k^- = \text{E} \{ (\mathbf{e}_k^-)(\mathbf{e}_k^-)^T \} = \text{E} \{ (\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T \}$$

and then a cross-covariance \mathbf{C}_k , with the measurement, and a measurement covariance \mathbf{W}_k , are defined respectively as

$$\begin{aligned} \mathbf{C}_k &= \text{E} \{ (\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{x}_k - \mathbf{m}_k)^T \} \\ \mathbf{W}_k &= \text{E} \{ (\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \}. \end{aligned}$$

From these above relationships, the covariance update can then be written in its final form as

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{C}_k \mathbf{K}_k^T - \mathbf{K}_k \mathbf{C}_k^T + \mathbf{K} \mathbf{W}_k \mathbf{K}_k^T. \quad (6.8)$$

The remaining steps are to define the gain, the expected value of the measurement, the cross-covariance, and the measurement covariance. First, the gain matrix \mathbf{K}_k is examined. While any gain could be used for \mathbf{K}_k , it is desired to find a gain such that the mean square of the posterior state estimation error is minimized. The cost function, or performance index, to be minimized is given by

$$J = \text{E} \{ (\mathbf{e}_k^+)^T (\mathbf{e}_k^+) \} = \text{trace} \{ \text{E} \{ (\mathbf{e}_k^+)(\mathbf{e}_k^+)^T \} \} = \text{trace} \{ \mathbf{P}_k^+ \},$$

which is expanded using Eq. (6.8) and simplified as

$$\begin{aligned} J &= \text{trace} \{ \mathbf{P}_k^- \} - \text{trace} \{ \mathbf{C}_k \mathbf{K}_k \} - \text{trace} \{ \mathbf{K}_k \mathbf{C}_k^T \} + \text{trace} \{ \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^T \} \\ &= \text{trace} \{ \mathbf{P}_k^- \} - 2\text{trace} \{ \mathbf{K}_k \mathbf{C}_k^T \} + \text{trace} \{ \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^T \}. \end{aligned}$$

In order to minimize the performance index, the derivative of the performance index is taken term-by term with respect to the gain \mathbf{K}_k . The properties of the derivative of the trace of a matrix can be found in Appendix C. The derivative terms are found to be

$$\begin{aligned}\frac{\partial}{\partial \mathbf{K}_k} \text{trace} \{ \mathbf{P}_k^T \} &= \mathbf{0} \\ \frac{\partial}{\partial \mathbf{K}_k} \text{trace} \{ \mathbf{K}_k \mathbf{C}_k^T \} &= \mathbf{C}_k \\ \frac{\partial}{\partial \mathbf{K}_k} \text{trace} \{ \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^T \} &= \mathbf{K}_k [\mathbf{W}_k + \mathbf{W}_k^T].\end{aligned}$$

From the definition of a covariance matrix, it is known that \mathbf{W}_k is symmetric and therefore that $\mathbf{W}_k + \mathbf{W}_k = 2\mathbf{W}_k$, which gives the derivative of the performance index as

$$\frac{\partial J}{\partial \mathbf{K}_k} = -2\mathbf{C}_k + 2\mathbf{K}_k \mathbf{W}_k.$$

The gain that minimizes the performance index is the one that renders the performance index stationary, or

$$\frac{\partial J}{\partial \mathbf{K}_k} = -2\mathbf{C}_k + 2\mathbf{K}_k \mathbf{W}_k = \mathbf{0}.$$

Solving for \mathbf{K}_k yields the final form of the Kalman gain as

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{W}_k^{-1} \tag{6.9}$$

which successfully minimizes the cost function by the examination of the Hessian of the performance index, J , [21]. The expected value of the measurement is then examined. The case where the measurement is nonlinear in the state and is subjected

to additive measurement noise is considered as

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{L}_k \mathbf{v}_k.$$

where \mathbf{L}_k is a noise mapping matrix. Then the expected value of both sides of the measurement model is taken to be

$$\hat{\mathbf{z}}_k = \mathbb{E} \{ \mathbf{z}_k \} = \mathbb{E} \{ \mathbf{h}(\mathbf{x}_k^-) \} + \mathbb{E} \{ \mathbf{L}_k \mathbf{v}_k \} \quad (6.10)$$

which is expanded about the *a priori* mean, with neglecting higher order terms, via

$$\mathbf{h}(\mathbf{x}_k) = \mathbf{h}(\mathbf{m}_k^-) + \mathbf{H}(\mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-). \quad (6.11)$$

The measurement Jacobian, $\mathbf{H}(\mathbf{m}_k^-)$, is defined as

$$\mathbf{H}(\mathbf{m}_k^-) = \left[\frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big|_{\mathbf{x}_k = \mathbf{m}_k^-} \right].$$

Then, substituting first order Taylor series expansion, given by Eq. (6.11), into the measurement expected value, or Eq. (6.10), yields

$$\hat{\mathbf{z}}_k = \mathbb{E} \{ \mathbf{h}(\mathbf{m}_k^-) \} + \mathbb{E} \{ \mathbf{H}(\mathbf{m}_k^-) \} + \mathbb{E} \{ \mathbf{L}_k \mathbf{v}_k \},$$

and since $\mathbf{h}(\mathbf{m}_k^-)$, $\mathbf{H}(\mathbf{m}_k^-)$ and \mathbf{L}_k are taken to be deterministic, $\hat{\mathbf{z}}_k$ can be rewritten as

$$\hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{m}_k^-) + \mathbf{H}(\mathbf{m}_k^-) \mathbb{E} \{ \mathbf{x}_k \} + \mathbf{L}_k \mathbb{E} \{ \mathbf{v}_k \}.$$

Because the measurement noise $E\{\mathbf{v}_k\}$ is taken to be zero mean and knowing that $E\{\mathbf{x}_k\} = \mathbf{m}_k^-$, the final form of the expected value of the measurement is given by

$$\hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{m}_k^-) \quad (6.12)$$

The cross-covariance is now examined which is given as

$$\mathbf{C}_k = E\{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{z}_k - \hat{\mathbf{z}})^T\}.$$

Substituting Equation 6.12 into the measurement model and expected measurement it follows that

$$\mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k) - \mathbf{h}(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{v}_k,$$

and applying a first-order Taylor series expansion about $\mathbf{h}(\mathbf{x}_k)$, yields

$$\mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{H}(\mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-) + \mathbf{L}_k \mathbf{v}_k. \quad (6.13)$$

Substituting back into the cross-covariance and assuming that $\mathbf{H}(\mathbf{m}_k^-)$ and \mathbf{L}_k are still deterministic, \mathbf{C}_k becomes

$$\mathbf{C}_k = E\{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} \mathbf{H}(\mathbf{m}_k^-)^T + E\{(\mathbf{x}_k - \mathbf{m}_k^-)\mathbf{v}_k^T\} \mathbf{L}_k^T.$$

The assumption is then made that the state is not correlated to the measurement noise, i.e. $E\{(\mathbf{x}_k - \mathbf{m}_k^-)\mathbf{v}_k^T\} = \mathbf{0}$, and recognizing that $E\{(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T\} = \mathbf{P}_k^-$, the final form of the cross-covariance becomes

$$\mathbf{C}_k = \mathbf{P}_k^- \mathbf{H}_k^T(\mathbf{m}_k^-). \quad (6.14)$$

The final component needed is the measurement covariance (or innovations covariance) which is given as

$$\mathbf{W}_k = \text{E} \left\{ (\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \right\}.$$

Recall $(\mathbf{z}_k - \hat{\mathbf{z}}_k)$ defined in Equation 6.13, and following the same conditions as the cross covariance, i.e. $\mathbf{H}(\mathbf{m}_k^-)$ and \mathbf{L}_k are deterministic, the state is not correlated with the measurement noise, and the covariance of the measurement noise is given by \mathbf{R}_k . Using these assumptions as relationships, the innovations covariance is given by

$$\mathbf{W}_k = \mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T. \quad (6.15)$$

6.1.3. Attitude Update. The additive states, which are all of the states except the quaternion attitude, are updated according to

$$\mathbf{m}_k^+ = \mathbf{m}_k^- + \mathbf{K}_k [\mathbf{z}_k - \hat{\mathbf{z}}_k].$$

Define the deviation added to the *a priori* mean as $\delta\tilde{\mathbf{m}}_k$, such that

$$\mathbf{m}_k^+ = \mathbf{m}_k^- + \delta\tilde{\mathbf{m}}_k.$$

The *a priori* quaternion $\hat{\mathbf{q}}_k^-$ can then be updated according to a multiplicative update to obtain the posterior quaternion, $\hat{\mathbf{q}}_k^+$ as

$$\hat{\mathbf{q}}_k^+ = \bar{\mathbf{q}}(\delta\tilde{\mathbf{m}}_{k,\theta}) \otimes \hat{\mathbf{q}}_k^+, \quad (6.16)$$

where $\delta\tilde{\mathbf{m}}_{k,\theta}$ is the portion of $\delta\tilde{\mathbf{m}}_k$ correlated to the attitude mean. However, numerical error can cause slight deviations away from the unit norm quaternion and likewise with the symmetrical elements of the covariance matrix. A forced normal-

ization can be implemented on the attitude quaternion and symmetrization of the covariance matrix. The procedure for the normalization and symmetrization is given respectively as

$$\hat{\mathbf{q}}_{\text{norm},k}^+ = \frac{\hat{\mathbf{q}}_k^+}{\|\hat{\mathbf{q}}_k^+\|} \quad \text{and} \quad \mathbf{P}_{\text{sym},k}^+ = \frac{1}{2}(\mathbf{P}_k^+ + (\mathbf{P}_k^+)^T). \quad (6.17)$$

All components for the covariance updates are now solved for and can be used in Equation 6.8 to arrive at the posterior covariance of the system.

6.1.4. Extended Kalman Filter Summary. A summary of the EKF is provided below. First, recall that the system model and measurement model are given as

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{L}_k\mathbf{v}_k \end{aligned}$$

with initial conditions for the mean and covariance given by

$$\begin{aligned} \mathbf{m}_0 &= \text{E}\{\mathbf{x}_0\} \\ \mathbf{P}_0 &= \text{E}\{(\mathbf{x}_0 - \mathbf{m}_0)(\mathbf{x}_0 - \mathbf{m}_0)^T\}. \end{aligned}$$

The mean and covariance propagation evolve via

$$\begin{aligned} \mathbf{m}_k &= \mathbf{f}(\mathbf{m}_{k-1}) \\ \mathbf{P} &= \mathbf{F}(\mathbf{m}_{k-1})\mathbf{P}_{k-1}\mathbf{F}(\mathbf{m}_{k-1})^T + \mathbf{M}_{k-1}\mathbf{Q}_{k-1}\mathbf{M}_{k-1}^T. \end{aligned}$$

where \mathbf{F} is the Jacobian of the nonlinear system. The update step is listed as the expected measurement, innovations covariance, cross-covariance, Kalman gain, followed

with the mean and covariance update equations respectively and is shown as

$$\begin{aligned}
\hat{\mathbf{z}}_k &= \mathbf{h}(\mathbf{m}_k^-) \\
\mathbf{W}_k &= \mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T \\
\mathbf{C}_k &= \mathbf{P}_k^- \mathbf{H}_k^T(\mathbf{m}_k^-) \\
\mathbf{K}_k &= \mathbf{C}_k \mathbf{W}_k^{-1} \\
\mathbf{m}_k^+ &= \mathbf{m}_k^- + \mathbf{K}_k [\mathbf{z}_k - \hat{\mathbf{z}}_k] \\
\mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{C}_k \mathbf{K}_k^T - \mathbf{K}_k \mathbf{C}_k^T + \mathbf{K} \mathbf{W}_k \mathbf{K}_k^T
\end{aligned}$$

where $\mathbf{H}(\mathbf{m}_k^-)$ is the Jacobian of the nonlinear measurement. It is also important to note that in the case of a quaternion attitude update, the multiplicative update must be used to obtain the posterior quaternion. This multiplicative update is summarized in the previous section.

6.2. STATE ESTIMATE AND STATE ESTIMATION ERROR COVARIANCE PROPAGATION

The true state represents an analytical integration of the continuous time dynamics. It is important to note that these true equations are not exact, but process noise injected into the dynamical system can envelop this discrepancy. It is known that the true state of the system propagates according to Equations 2.12, which are shown as

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{v}_{k-1} \Delta t_k + \frac{1}{2} \mathbf{T}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \boldsymbol{\theta}_k \times] \right) \Delta \mathbf{v}_k \Delta t_k \quad (6.18a)$$

$$\begin{aligned}
&+ \frac{1}{2} \left(\mathbf{g}_{k-1} - \frac{1}{3} \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k \right) \Delta t_k^2 \\
\mathbf{v}_k &= \mathbf{v}_{k-1} + \mathbf{T}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{2} [\Delta \boldsymbol{\theta}_k \times] \right) \Delta \mathbf{v}_k + \left(\mathbf{g}_{k-1} - \frac{1}{2} \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k \right) \Delta t_k
\end{aligned} \quad (6.18b)$$

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}_{k-1}. \quad (6.18c)$$

The error estimation equations derived in the following section are used to formulate the needed Jacobian for the estimation error covariance propagation. The following derivations follow and are based on Reference [12].

6.2.1. Position and Velocity Error Covariance. Define the position and velocity errors to be

$$\mathbf{e}_{r,k} = \mathbf{r}_k - \hat{\mathbf{r}}_k \quad \text{and} \quad \mathbf{e}_{v,k} = \mathbf{v}_k - \hat{\mathbf{v}}_k.$$

Utilizing the mean position and velocity propagation equations shown in Equations 2.12a and 2.12b, the respective errors are expanded as

$$\begin{aligned} \mathbf{e}_{r,k} = & (\mathbf{r}_k - \hat{\mathbf{r}}_k) + (\mathbf{v}_k - \hat{\mathbf{v}}_k) \Delta t_k + \frac{1}{2} (\mathbf{g}_{k-1} - \hat{\mathbf{g}}_{k-1}) \Delta t_k^2 \\ & + \frac{1}{2} \left(\mathbf{T}_{k-1}^T \Delta \mathbf{v}_k - \hat{\mathbf{T}}_{k-1}^T \Delta \hat{\mathbf{v}}_{m,k} \right) \Delta t_k \\ & + \frac{1}{6} \left(\mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k - \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \Delta \hat{\mathbf{v}}_{m,k} \right) \Delta t_k \\ & - \frac{1}{6} \left(\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k - \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}} \right) \Delta t_k^2 \end{aligned} \quad (6.19a)$$

$$\begin{aligned} \mathbf{e}_{v,k} = & (\mathbf{v}_k - \hat{\mathbf{v}}_k) + (\mathbf{g}_{k-1} - \hat{\mathbf{g}}_{k-1}) \Delta t_k + \left(\mathbf{T}_{k-1}^T \Delta \mathbf{v}_k - \hat{\mathbf{T}}_{k-1}^T \Delta \hat{\mathbf{v}}_{m,k} \right) \\ & + \frac{1}{2} \left(\mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k - \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}} \times] \Delta \hat{\mathbf{v}}_{m,k} \right) \\ & - \frac{1}{2} \left(\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k - \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}} \right) \Delta t_k \end{aligned} \quad (6.19b)$$

In comparing the position and velocity estimation errors, many terms are identical, up to scaling. Therefore, it is needed to only expand the remaining four terms in order to linearize the estimation errors. The four terms to be linearized are given as

$$\mathbf{g}_{k-1}, \quad \mathbf{T}_{k-1}^T \Delta \mathbf{v}_k, \quad \mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k \quad \text{and} \quad \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k.$$

First, the attitude matrix is to be expanded. The quaternion is defined as a multiplicative error; therefore, the attitude matrices are multiplied in the same order as the quaternions, which gives

$$\delta \mathbf{T}_{k-1} = \mathbf{T}_{k-1} \hat{\mathbf{T}}_{k-1}^T. \quad (6.20)$$

This multiplicative attitude matrix error is expanded via a first-order Taylor series expansion, which is shown as

$$\delta \mathbf{T}_{k-1} = \mathbf{I}_{3 \times 3} - [\mathbf{e}_{\theta, k-1} \times].$$

where $\mathbf{e}_{\theta, k-1}$ is defined as the error associated with attitude at time t_{k-1} . Using the relationship given in Equation 6.20 and transposing the solution gives the attitude matrix expansion as

$$\mathbf{T}_{k-1}^T = \hat{\mathbf{T}}_{k-1}^T + \hat{\mathbf{T}}_{k-1}^T [\mathbf{e}_{\theta, k-1} \times]. \quad (6.21)$$

The next term to expand about the estimate is gravity as a function of position \mathbf{g}_{k-1} , such that

$$\mathbf{g}_{k-1} = \hat{\mathbf{g}}_{k-1} + \hat{\mathbf{G}}_{k-1} \mathbf{e}_{s, k-1}, \quad \text{where} \quad \mathbf{e}_{s, k-1} \triangleq \mathbf{s}_{k-1} - \hat{\mathbf{s}}_{k-1}. \quad (6.22)$$

where $\hat{\mathbf{G}}_{k-1}$ is defined as the gravity Jacobian and is given in Appendix C. The errors in $\Delta \mathbf{v}_k$ and $\Delta \boldsymbol{\theta}_k$ are then defined respectively, to be

$$\mathbf{e}_{\Delta v, k} \triangleq \Delta \mathbf{v}_k - \Delta \hat{\mathbf{v}}_{m, k}, \quad \text{and} \quad \mathbf{e}_{\Delta \theta, k} \triangleq \Delta \boldsymbol{\theta}_{k-1} - \Delta \hat{\boldsymbol{\theta}}_{m, k}.$$

Using these error definitions and eliminating higher-order terms, an expression for $\mathbf{T}_{k-1}^T \Delta \mathbf{v}_k$ is given as

$$\mathbf{T}_{k-1}^T \Delta \mathbf{v}_k = \hat{\mathbf{T}}_{k-1}^T \Delta \hat{\mathbf{v}}_k + \hat{\mathbf{T}}_{k-1}^T \mathbf{e}_{\Delta v, k} - \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m, k} \times] \mathbf{e}_{\theta, k-1}. \quad (6.23)$$

Now the term $\mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k$ is to be expanded and becomes

$$\mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k = \left(\hat{\mathbf{T}}_{k-1}^T + \hat{\mathbf{T}}_{k-1}^T [\mathbf{e}_{\theta, k-1} \times] \right) \left((\Delta \hat{\boldsymbol{\theta}}_{m, k} + \mathbf{e}_{\Delta \theta, k}) \times (\Delta \hat{\mathbf{v}}_{m, k} + \mathbf{e}_{\Delta v, k}) \right),$$

which, when expanded out and after eliminating higher-order terms yields

$$\begin{aligned} \mathbf{T}_{k-1}^T [\Delta \boldsymbol{\theta}_k \times] \Delta \mathbf{v}_k &= \hat{\mathbf{T}}_{k-1}^T (\Delta \hat{\boldsymbol{\theta}}_{m, k} \times \Delta \hat{\mathbf{v}}_{m, k}) + \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m, k} \times] \mathbf{e}_{\Delta v, k} \\ &\quad - \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m, k} \times] \mathbf{e}_{\Delta \theta, k} - \hat{\mathbf{T}}_{k-1}^T [(\Delta \hat{\boldsymbol{\theta}}_{m, k} \times \Delta \hat{\mathbf{v}}_{m, k}) \times] \mathbf{e}_{\theta, k-1}. \end{aligned} \quad (6.24)$$

Finally, the term $\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k$ is to be expanded and yields

$$\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k = \mathbf{G}_{k-1} \left(\hat{\mathbf{T}}_{k-1}^T + \hat{\mathbf{T}}_{k-1}^T [\mathbf{e}_{\theta, k-1} \times] \right) [(\hat{\mathbf{d}} + \mathbf{e}_d) \times] (\Delta \hat{\boldsymbol{\theta}}_{m, k} + \mathbf{e}_{\Delta \theta, k}),$$

where \mathbf{e}_d is defined as the error of the position vector from the IMU to the CM of the vehicle. If higher-order terms are eliminated from this expression, the terms containing both \mathbf{G}_{k-1} and an error source are already a first-order error source. With the elimination of higher-order terms, \mathbf{G}_{k-1} can then be written as its estimate $\hat{\mathbf{G}}_{k-1}$ due to any expansion of \mathbf{G}_{k-1} will result in second-order terms. Therefore, $\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k$ can then be written as

$$\begin{aligned} \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k &= \mathbf{G}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_{m, k} + \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T [\hat{\mathbf{d}} \times] \mathbf{e}_{\Delta \theta, k} \\ &\quad - \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m, k} \times] \mathbf{e}_d - \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T \left[(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m, k}) \times \right] \mathbf{e}_{\theta, k-1}. \end{aligned}$$

Now a vector \mathbf{p}_{k-1} is defined as

$$\mathbf{p}_{k-1} \triangleq \mathbf{G}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_{m,k}.$$

This vector can be expanded to first-order and is expressed as

$$\mathbf{p}_{k-1} = \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{U}}_{k-1} \mathbf{e}_{s,k-1} \quad \text{with} \quad \hat{\mathbf{p}}_{k-1} = \hat{\mathbf{G}} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_{m,k},$$

where $\hat{\mathbf{U}}_{k-1} \in \mathbb{R}^{3 \times 3}$, with the element in the i^{th} row and j^{th} column given by

$$\hat{U}_{k-1}(i, j) = \left[\sum_{\ell=1}^3 \frac{\partial g(i)}{\partial s(j) \partial s(\ell)} u(\ell) \Big|_{\mathbf{s}=\hat{\mathbf{s}}_{k-1}} \right].$$

Making use of \mathbf{p}_{k-1} , $\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \Delta \boldsymbol{\theta}_k$ can be expressed in its first form as

$$\begin{aligned} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \boldsymbol{\theta}_k &= \mathbf{G}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_{m,k} + \hat{\mathbf{U}}_{k-1} \mathbf{e}_{s,k-1} + \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T [\hat{\mathbf{d}} \times] \mathbf{e}_{\Delta \theta, k} \\ &\quad - \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{e}_d - \hat{\mathbf{G}}_{k-1} \mathbf{T}_{k-1}^T \left[(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m,k}) \times \right] \mathbf{e}_{\theta, k-1}. \end{aligned} \quad (6.25)$$

Only error expression for $\mathbf{e}_{s,k-1}$, $\mathbf{e}_{\Delta v, k}$ and $\mathbf{e}_{\Delta \theta, k}$ are needed to obtain the final position and velocity error estimation expressions. From the definitions of $\mathbf{e}_{s,k-1}$ and \mathbf{s}_{k-1} gives

$$\mathbf{e}_{s,k-1} = \mathbf{s}_{k-1} - \hat{\mathbf{s}}_{k-1} = (\mathbf{r}_{k-1} - \hat{\mathbf{r}}_{k-1}) + \left(\mathbf{T}_{k-1}^T \mathbf{d} - \hat{\mathbf{T}}_{k-1}^T \hat{\mathbf{d}} \right),$$

which is then expressed in first order form as

$$\mathbf{e}_{s,k-1} = \mathbf{e}_{r,k-1} + \hat{\mathbf{T}}_{k-1}^T \mathbf{e}_d - \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \mathbf{e}_{\theta, k-1}, \quad (6.26)$$

and the error terms $\mathbf{e}_{\Delta v,k}$ and $\mathbf{e}_{\Delta\theta,k}$ are defined respectively to be

$$\mathbf{e}_{\Delta v,k} = \Delta \mathbf{v}_k - \Delta \hat{\mathbf{v}}_{m,k}$$

$$\mathbf{e}_{\Delta\theta,k} = \Delta \boldsymbol{\theta}_k - \Delta \hat{\boldsymbol{\theta}}_{m,k}$$

and using the substitutions of $\Delta \mathbf{v}_k$ and $\Delta \boldsymbol{\theta}_k$ from Equations 2.7 and 2.10 yields

$$\mathbf{e}_{\Delta v,k} = [\Delta \hat{\mathbf{v}}_{m,k} \times] \mathbf{m}_a - [\Delta \hat{\mathbf{v}}_{m,k}^*] \mathbf{n}_a - [\Delta \hat{\mathbf{v}}_{m,k} \setminus] \mathbf{s}_a - \mathbf{b}_a - \mathbf{w}_{v,k} \quad (6.27a)$$

$$\mathbf{e}_{\Delta\theta,k} = [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g - [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g - \mathbf{b}_g - \mathbf{w}_{\theta,k}. \quad (6.27b)$$

The final form of the position and velocity error covariance is obtained by substituting Equations 6.21-6.27 into Equations 6.19a and 6.19b yielding

$$\mathbf{e}_{r,k} = \left[\mathbf{I}_{3 \times 3} + \frac{1}{2} \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \Delta t_k^2 \right] \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1} \Delta t_k \quad (6.28a)$$

$$\begin{aligned} & - \frac{1}{2} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \frac{1}{3} \hat{\mathbf{T}}_{k-1}^T [(\Delta \hat{\boldsymbol{\theta}}_{m,k} \times \Delta \hat{\mathbf{v}}_{m,k}) \times] \right) \Delta t_k \mathbf{e}_{\theta,k-1} \\ & - \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] - \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1} [(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m,k}) \times] \right] \Delta t_k^2 \mathbf{e}_{\theta,k-1} \\ & + \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1} + \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right] \Delta t_k^2 \mathbf{e}_d \\ & - \hat{\mathbf{R}}_a \mathbf{b}_g + \hat{\mathbf{R}}_a [\Delta \hat{\mathbf{v}}_{m,k} \times] \mathbf{m}_a - \hat{\mathbf{R}}_a [\Delta \hat{\mathbf{v}}_{m,k}^*] \mathbf{n}_a - \hat{\mathbf{R}}_a [\Delta \hat{\mathbf{v}}_{m,k} \setminus] \mathbf{s}_a \\ & - \hat{\mathbf{R}}_g \mathbf{b}_g + \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g - \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g \\ & - \hat{\mathbf{R}}_a \mathbf{w}_{v,k} - \hat{\mathbf{R}}_g \mathbf{w}_{\theta,k} \end{aligned}$$

$$\mathbf{e}_{v,k} = \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \Delta t_k \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1} \quad (6.28b)$$

$$\begin{aligned} & - \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \frac{1}{2} \hat{\mathbf{T}}_{k-1}^T [(\Delta \hat{\boldsymbol{\theta}}_{m,k} \times \Delta \hat{\mathbf{v}}_{m,k}) \times] \right) \mathbf{e}_{\theta,k-1} \\ & - \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] - \frac{1}{2} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1} [(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m,k}) \times] \right] \Delta t_k \mathbf{e}_{\theta,k-1} \\ & + \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{2} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1} + \frac{1}{2} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right] \Delta t_k \mathbf{e}_d \end{aligned}$$

$$\begin{aligned}
& - \hat{\mathbf{V}}_a \mathbf{b}_g + \hat{\mathbf{V}}_a [\Delta \hat{\mathbf{v}}_{m,k} \times] \mathbf{m}_a - \hat{\mathbf{V}}_a [\Delta \hat{\mathbf{v}}_{m,k}^*] \mathbf{n}_a - \hat{\mathbf{V}}_a [\Delta \hat{\mathbf{v}}_{m,k} \setminus] \mathbf{s}_a \\
& - \hat{\mathbf{V}}_g \mathbf{b}_g + \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g - \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g \\
& - \hat{\mathbf{V}}_a \mathbf{w}_{v,k} - \hat{\mathbf{V}}_g \mathbf{w}_{\theta,k}
\end{aligned}$$

where the matrices $\hat{\mathbf{R}}_a$, $\hat{\mathbf{R}}_g$, $\hat{\mathbf{V}}_a$ and $\hat{\mathbf{V}}_g$ are defined to be,

$$\begin{aligned}
\hat{\mathbf{R}}_a &= \frac{1}{2} \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right) \Delta t_k \\
\hat{\mathbf{R}}_g &= \frac{1}{6} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta t_k \right) \Delta t_k \\
\hat{\mathbf{V}}_a &= \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right) \\
\hat{\mathbf{V}}_g &= \frac{1}{2} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta t_k \right).
\end{aligned}$$

To formulate and construct the full $\mathbf{F}(\mathbf{m}_{k-1})$ and $\mathbf{M}(\mathbf{m}_{k-1})$ matrices, refer to Appendix C.

6.2.2. Attitude Error Covariance. While the position and velocity error are defined in a straightforward manner, the attitude error must be defined as a multiplicative attitude estimation error as

$$\delta \bar{\mathbf{q}}_k \triangleq \bar{\mathbf{q}}_k \otimes \hat{\bar{\mathbf{q}}}_k^{-1}.$$

Equation 2.12c is substituted in for $\hat{\bar{\mathbf{q}}}_k$ and Equation 6.18c for $\bar{\mathbf{q}}_k$ which gives

$$\begin{aligned}
\delta \bar{\mathbf{q}}_k &= \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}_{k-1} \otimes \hat{\bar{\mathbf{q}}}_{k-1}^{-1} \otimes \bar{\mathbf{q}}(\Delta \hat{\boldsymbol{\theta}}_{m,k})^{-1} \\
&= \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_k) \otimes \delta \bar{\mathbf{q}}_{k-1} \otimes \bar{\mathbf{q}}(\Delta \hat{\boldsymbol{\theta}}_{m,k})^{-1} \\
&= \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_{m,k})^{-1} \otimes \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_{m,k}) \otimes \delta \bar{\mathbf{q}}_{k-1} \otimes \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_{m,k})^{-1}.
\end{aligned}$$

It can then be shown that

$$\bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_k) \otimes \delta\bar{\mathbf{q}}_{k-1} \otimes \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_k)^{-1} = \begin{bmatrix} \mathbf{T}(\Delta\boldsymbol{\theta}_{m,k}\delta\bar{\mathbf{q}}_{k-1}) \\ \delta q_{k-1} \end{bmatrix}$$

where

$$\mathbf{T}(\Delta\boldsymbol{\theta}_{m,k}) = \mathbf{I}_{3 \times 3} - \sin \|\Delta\boldsymbol{\theta}_{m,k}\| [\Delta\hat{\boldsymbol{\theta}}_{m,k} \times] + (1 - \cos \|\Delta\boldsymbol{\theta}_{m,k}\|) [\Delta\boldsymbol{\theta}_{m,k} \times][\Delta\boldsymbol{\theta}_{m,k} \times].$$

The term $\bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k})^{-1}$ is well approximated as a small angle rotation quaternion giving the approximation as

$$\bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k})^{-1} \approx \begin{bmatrix} \frac{1}{2}\mathbf{e}_{\Delta\theta,k} \\ 1 \end{bmatrix},$$

where the error term $\mathbf{e}_{\Delta\theta,k}$ is defined as $\Delta\boldsymbol{\theta}_k - \Delta\hat{\boldsymbol{\theta}}_{m,k}$. These relationships give the attitude estimation error as

$$\delta\bar{\mathbf{q}}_k = \begin{bmatrix} \frac{1}{2}\mathbf{e}_{\Delta\theta,k} \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{T}(\Delta\hat{\boldsymbol{\theta}}_{m,k})\delta\bar{\mathbf{q}}_{k-1} \\ \delta q_{k-1} \end{bmatrix}.$$

With the small angle assumption, the vector part of the quaternion then fully represents the attitude, so approximating to first order, the above expression becomes

$$\delta\bar{\mathbf{q}}_k = \mathbf{T}(\Delta\hat{\boldsymbol{\theta}}_{m,k})\delta\bar{\mathbf{q}}_{k-1} + \frac{1}{2}\mathbf{e}_{\Delta\theta,k}$$

Letting $\delta\bar{\mathbf{q}}_k$ change notationally to $\mathbf{e}_{\theta,k}$, and knowing that the rotation vector $\boldsymbol{\theta}$ is approximately twice the vector part of the quaternion, the attitude estimation error

is given by

$$\mathbf{e}_{\theta,k} = \mathbf{T}(\Delta\hat{\boldsymbol{\theta}}_{m,k})\mathbf{e}_{\theta,k-1} + \mathbf{e}_{\Delta\theta,k}.$$

Then substituting Equation 2.9 into $\mathbf{e}_{\Delta\theta,k}$, the final form of the attitude estimation error propagation is given by

$$\mathbf{e}_{\theta,k} = \mathbf{T}(\Delta\hat{\boldsymbol{\theta}}_{m,k})\mathbf{e}_{\theta,k-1} + [\Delta\hat{\boldsymbol{\theta}}_{m,k}\times]\mathbf{m}_g - [\Delta\hat{\boldsymbol{\theta}}_{m,k}*]\mathbf{n}_g - [\Delta\hat{\boldsymbol{\theta}}_{m,k}\setminus]\mathbf{s}_g - \mathbf{b}_g - \mathbf{w}_g. \quad (6.29)$$

6.2.3. Error Covariance in a Fusion Network. In the construction of a multi-IMU fusion network of N IMUs with direct averaging as the applied fusion rule, the increased performance follows a \sqrt{N} factor. However, in the derivation of the position, velocity and attitude estimation error covariance, this fusion performance index is not accounted for. It is desired to construct a set of error covariance equations that contain and account for this \sqrt{N} parameter to allow for the statistical quantification of N IMUs in its covariance propagation. The proposed method is to inject this parameter into all of the IMU error terms given as bias \mathbf{b} , misalignment \mathbf{m} , nonorthogonality \mathbf{n} , scale factor uncertainty \mathbf{s} , and its noise \mathbf{w} for accelerometers and gyroscopes. Per this discussion, the fused estimation error covariance for position and velocity can be written as a function of \sqrt{N} and is given as

$$\begin{aligned} \mathbf{e}_{r,k} = & \left[\mathbf{I}_{3\times 3} + \frac{1}{2} \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \Delta t_k^2 \right] \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1} \Delta t_k \quad (6.30a) \\ & - \frac{1}{2} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \mathbf{v}_{m,k} \times] + \frac{1}{3} \hat{\mathbf{T}}_{k-1}^T [(\Delta \boldsymbol{\theta}_{m,k} \times \Delta \mathbf{v}_{m,k}) \times] \right) \Delta t_k \mathbf{e}_{\theta,k-1} \\ & - \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] - \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1} [(\hat{\mathbf{d}} \times \Delta \boldsymbol{\theta}_{m,k}) \times] \right] \Delta t_k^2 \mathbf{e}_{\theta,k-1} \\ & + \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1} + \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right] \Delta t_k^2 \mathbf{e}_d \end{aligned}$$

$$\begin{aligned}
& - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_a \mathbf{b}_g + \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_a [\Delta \mathbf{v}_{m,k} \times] \mathbf{m}_a - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_a [\Delta \mathbf{v}_{m,k}^*] \mathbf{n}_a - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_a [\Delta \mathbf{v}_{m,k} \setminus] \mathbf{s}_a \\
& - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_g \mathbf{b}_g + \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g \\
& - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_a \mathbf{w}_{v,k} - \frac{1}{\sqrt{N}} \hat{\mathbf{R}}_g \mathbf{w}_{\theta,k} \\
\mathbf{e}_{v,k} = & \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \Delta t_k \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1} \\
& - \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \mathbf{v}_{m,k} \times] + \frac{1}{2} \hat{\mathbf{T}}_{k-1}^T [(\Delta \hat{\boldsymbol{\theta}}_{m,k} \times \Delta \mathbf{v}_{m,k}) \times] \right) \mathbf{e}_{\theta,k-1} \\
& - \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] - \frac{1}{2} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1} [(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m,k}) \times] \right] \Delta t_k \mathbf{e}_{\theta,k-1} \\
& + \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{2} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1} + \frac{1}{2} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right] \Delta t_k \mathbf{e}_d \\
& - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_a \mathbf{b}_g + \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_a [\Delta \mathbf{v}_{m,k} \times] \mathbf{m}_a - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_a [\Delta \mathbf{v}_{m,k}^*] \mathbf{n}_a - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_a [\Delta \mathbf{v}_{m,k} \setminus] \mathbf{s}_a \\
& - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_g \mathbf{b}_g + \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_g [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g \\
& - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_a \mathbf{w}_{v,k} - \frac{1}{\sqrt{N}} \hat{\mathbf{V}}_g \mathbf{w}_{\theta,k}
\end{aligned} \tag{6.30b}$$

and the attitude error covariance for an IMU fusion network can then be written as

$$\mathbf{e}_{\theta,k} = \mathbf{T}(\Delta \hat{\boldsymbol{\theta}}_{m,k}) \mathbf{e}_{\theta,k-1} + \frac{1}{\sqrt{N}} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \mathbf{m}_g - \frac{1}{\sqrt{N}} [\Delta \hat{\boldsymbol{\theta}}_{m,k}^*] \mathbf{n}_g \tag{6.31}$$

$$- \frac{1}{\sqrt{N}} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \setminus] \mathbf{s}_g - \frac{1}{\sqrt{N}} \mathbf{b}_g - \frac{1}{\sqrt{N}} \mathbf{w}_g. \tag{6.32}$$

Recall that the terms $\hat{\mathbf{R}}_a$, $\hat{\mathbf{R}}_g$, $\hat{\mathbf{V}}_g$ and $\hat{\mathbf{V}}_g$ are defined in the prior section in the derivation of the estimation error covariance equations. With this set of equations, the propagated uncertainty is now sensitive to the number of sensors being fused in accordance with the fusion performance increase model from Reference [1]. It is important to note that this error model is only for a homogeneous sensor network and the sensor specifications are taken from one sensor. With the current proposed fault detection scheme, it is also necessary to note that N can change over time in the event of a sensor failure.

6.3. MEASUREMENT PROCESSING

6.3.1. Position Measurement. The measurement for the position of a spacecraft is given by its non-linear measurement model as

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}_r$$

and recalling that the position estimation of a spacecraft is given by

$$\mathbf{h}(\mathbf{x}) = \mathbf{r}_{imu}^i + \mathbf{T}_b^i \mathbf{r}_{s/imu}^b.$$

The estimated position measurement is the measurement model computed at the mean, i.e.,

$$\mathbf{h}(\hat{\mathbf{x}}) = \hat{\mathbf{r}}_{imu}^i + \hat{\mathbf{T}}_b^i \mathbf{r}_{s/imu}^b.$$

The measurement deviation, $\delta\mathbf{z}$, of the position sensor is computed with the difference equation of the measurement of the position at some time k , and its respective estimate evaluated at the current mean as

$$\delta\mathbf{z} = \mathbf{h}(\mathbf{x}) - \mathbf{h}(\hat{\mathbf{x}}) + \mathbf{v}_r = \mathbf{H}(\hat{\mathbf{x}})\mathbf{e} + \mathbf{v}_r.$$

The measurement Jacobian, $\mathbf{H}(\hat{\mathbf{x}})$, for a position sensor is derived from the position deviation that has the form

$$\mathbf{H}(\mathbf{m}_k) = \begin{bmatrix} \mathbf{H}_r & \mathbf{0}_{3 \times 3} & \mathbf{H}_\theta & \mathbf{0}_{3 \times 27} \end{bmatrix}$$

from Equation 5.2 that is given as

$$\delta\mathbf{z} = \delta\mathbf{r}_{sc}^i - \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] \delta\boldsymbol{\theta}_b^i.$$

The components associated with the position and attitude error define the specific elements of the measurement Jacobian and are given as

$$\mathbf{H}_r = \mathbf{I}_{3 \times 3}, \quad \text{and} \quad \mathbf{H}_\theta = -\hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times].$$

6.3.2. Quaternion Measurement. Recall that the quaternion estimation of a spacecraft's attitude is given by

$$h(\bar{\mathbf{q}}) = \hat{\mathbf{q}}_i^b.$$

The measurement residual \mathbf{y}_θ, k of the quaternion sensor is computed by a multiplicative difference equation of the quaternion measurement $\bar{\mathbf{q}}_i^b$ at some time k and its estimate $\hat{\mathbf{q}}_i^b$ as a function of the mean quaternion which yields

$$\mathbf{y}_{\theta,k} = 2\text{vec}(\bar{\mathbf{q}}_{m,i}^b \otimes \hat{\mathbf{q}}_i^b).$$

Recall that the term $\bar{\mathbf{q}}_{m,i}^b \otimes \hat{\mathbf{q}}_i^b$ is well approximated as a small angle rotation quaternion. The measurement Jacobian for a quaternion sensor is derived from the attitude deviation that has the form

$$\mathbf{H}(\mathbf{m}_k) = \begin{bmatrix} \mathbf{0}_{3 \times 6} & \mathbf{H}_\theta & \mathbf{0}_{3 \times 27} \end{bmatrix}$$

from Equation 5.3 given again as

$$\delta \boldsymbol{\theta}_i^b = 2\text{vec}(\bar{\mathbf{q}}_i^b \otimes \hat{\mathbf{q}}_i^b) = \mathbf{e}_\theta.$$

The components associated with the attitude error define the specific element of the measurement Jacobian and is given as

$$\mathbf{H}_\theta = \mathbf{I}_{3 \times 3}$$

6.3.3. Range and Range Rate. Recall that the estimation range and range rate measurement is given by

$$\hat{z}_\rho = \sqrt{\left(\hat{\mathbf{r}}_{s/g}^i\right)^T \left(\hat{\mathbf{r}}_{s/g}^i\right)}, \quad \text{and} \quad \hat{z}_{\dot{\rho}} = \frac{\hat{\mathbf{r}}_{s/g}^i \hat{\mathbf{v}}_{s/g}^i}{\sqrt{\left(\hat{\mathbf{r}}_{s/g}^i\right)^T \left(\hat{\mathbf{r}}_{s/g}^i\right)}}.$$

The measurement residual $\mathbf{y}_{rv,k}$ of the range and rangerate sensor is computed with the measurement of the range and rangerate at some time by ρ , $\dot{\rho}$ and their estimates \hat{z}_ρ , $\hat{z}_{\dot{\rho}}$ given as

$$\mathbf{y}_{rv,k} = [z_\rho \ z_{\dot{\rho}}]^T - [\hat{z}_\rho \ \hat{z}_{\dot{\rho}}]^T$$

The measurement Jacobian for a range and range rate sensor is derived from the range and range rate deviations which has the form

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{\rho,r} & \mathbf{0}_{3 \times 3} & \mathbf{H}_{\rho,\theta} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 12} \\ \mathbf{H}_{\dot{\rho},r} & \mathbf{H}_{\dot{\rho},v} & \mathbf{H}_{\dot{\rho},\theta} & \mathbf{H}_{\dot{\rho},b} & \mathbf{H}_{\dot{\rho},m} & \mathbf{H}_{\dot{\rho},n} & \mathbf{H}_{\dot{\rho},s} & \mathbf{0}_{3 \times 12} \end{bmatrix}.$$

from Equations 5.5 and 5.9 which are given as

$$\begin{aligned} \delta z_\rho &= \mathbf{R}_\rho \delta \mathbf{r}_{imu}^i - \mathbf{R}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta \boldsymbol{\theta}_b^i + w_\rho \\ \delta z_{\dot{\rho}} &= \left(\mathbf{R}_\rho \delta \mathbf{r}_{imu}^i - \mathbf{R}_\rho \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta \boldsymbol{\theta}_b^i \right) + \left(\mathbf{V}_{\dot{\rho}} \delta \mathbf{v}_{imu}^i - \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i \left[\mathbf{r}_{s/imu}^b \times \right] \delta \boldsymbol{\omega}_{b/i}^b \right. \\ &\quad \left. - \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i \left[\left(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b \right) \times \right] \delta \boldsymbol{\theta}_b^i \right) + w_{\dot{\rho}}. \end{aligned}$$

The components of associated with the position, velocity and attitude error define the specific elements of the measurement Jacobian and are shown as

$$\begin{aligned}
\mathbf{H}_{\rho,r} &= \mathbf{R}_\rho & \mathbf{H}_{\rho,\theta} &= \mathbf{R}_\rho \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] \\
\mathbf{H}_{\dot{\rho},r} &= \mathbf{R}_\rho & \mathbf{H}_{\dot{\rho},v} &= \mathbf{V}_{\dot{\rho}} \\
\mathbf{H}_{\dot{\rho},b} &= \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] \hat{\mathbf{b}}_{g,k} & \mathbf{H}_{\dot{\rho},m} &= -\mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] [\bar{\boldsymbol{\omega}}_{m,k} \times] \\
\mathbf{H}_{\dot{\rho},n} &= \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] [\bar{\boldsymbol{\omega}}_{m,k} \setminus] & \mathbf{H}_{\dot{\rho},s} &= \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i [\mathbf{r}_{s/imu}^b \times] [\bar{\boldsymbol{\omega}}_{m,k}^*] \\
\mathbf{H}_{\dot{\rho},\theta} &= \mathbf{V}_{\dot{\rho}} \hat{\mathbf{T}}_b^i [(\hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{r}_{s/imu}^b) \times].
\end{aligned}$$

Note that the \mathbf{V} and \mathbf{R} matrices for range and range rate are defined in their respective modeling sections.

6.3.4. Unit Vector Star Camera. Recall that the estimated unit vector star camera measurement is given by Equation 5.12 as

$$\hat{\mathbf{z}}_{sc} = \mathbf{T}_b^{sc} \hat{\mathbf{T}}_i^b \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr}.$$

where the the transformation matrix from the inertial-to-body frame estimate is a function the estimated inertial-to-body quaternion which is defined to be

$$\hat{\mathbf{T}}_{i,k}^b = \mathbf{T}(\hat{\mathbf{q}}_{i,k}^b) = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_3q_1 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & 1 - 2q_3^2 - 2q_1^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_3q_1 - q_2q_4) & 2(q_2q_3 - q_1q_4) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

The measurement residual $\mathbf{y}_{r,k}$ of the unit vector star camera is computed with the measurement of a star unit vector with respect to the star camera at some time by $\mathbf{u}_{s/sc,k}^{sc}$ and its estimate $\hat{\mathbf{u}}_{s/sc,k}^{sc}$, and is given as

$$\mathbf{y}_{r,k} = \mathbf{z}_{sc}^{sc} - \hat{\mathbf{z}}_{sc}.$$

The measurement Jacobian for a unit vector star vector camera is derived from Eq. (5.13) which has the form

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}_\theta & \mathbf{0}_{3 \times 12} & \mathbf{0}_{3 \times 12} \end{bmatrix}.$$

From Equation Eq. (5.13), which is given as

$$\delta \mathbf{z}_{sc} = \mathbf{T}_b^{sc} \left[\hat{\mathbf{T}}_i^{bc} \mathbf{T}_{cr}^i \mathbf{z}_{sc} \times \right] \delta \boldsymbol{\theta}_b^i + \mathbf{v}_{sc}, \quad (6.33)$$

the component associated with the attitude error define the specific elements of the measurement Jacobian and is shown as

$$\mathbf{H}_\theta = \mathbf{T}_b^{sc} \left[\hat{\mathbf{T}}_i^{bc} \mathbf{T}_{cr}^i \mathbf{u}_{s/sc}^{cr} \times \right].$$

7. NAVIGATION/FAULT DETECTION SYSTEM ARCHITECTURE

7.1. SYSTEM IMPLEMENTATION

The implementation and construction of a distributed network of multiple redundant inertial sensors with robust fault detection is outlined in this section. The architecture hierarchy in this overview is divided into three sections: sensor fault detection, data fusion, and navigation. This overview is shown in a block diagram in Figure 7.1.

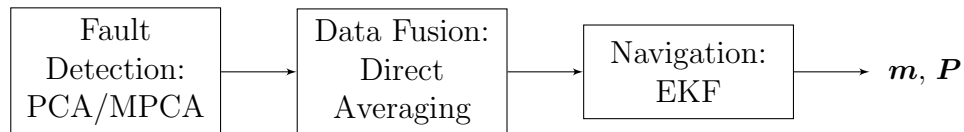


Figure 7.1. Implementation Architecture Overview

Throughout this chapter, each block is individually broken down into a more in-depth description and detailed systematic diagrams.

7.1.1. Fault Detection Architecture. Simulation implementation of fault detection can become convoluted if not carefully outlined. For a static system, two underlying processes are needed for a closed-loop fault detection algorithm: fusion/navigation and PCA. The PCA algorithm is an underlying process that is running at a desired frequency. After the L IMUs are transformed into the feature plane, the transformed data are compared to a user specified fault threshold, τ . If a sensor output exceeds this threshold, a fault alarm is signaled by attaching a fault identification to the poor or failing measurement. This fault identification label prevents the faulty IMU from being introduced in the fusion of the sensor outputs. If all of the transformed sensor outputs are within the specified threshold, the navigation algorithm

proceeds as follows. The fusion and navigation algorithm starts by subtracting the centripetal acceleration from each sensor output, followed by the simple direct average fusion rule for all sensor outputs within the fault threshold condition. The fused data can then be used to dead reckon a navigation solution, in this case the mean, \mathbf{m} , and covariance, \mathbf{P} , of the system. This procedure is summarized and visualized in block diagram form in Figure 7.2.

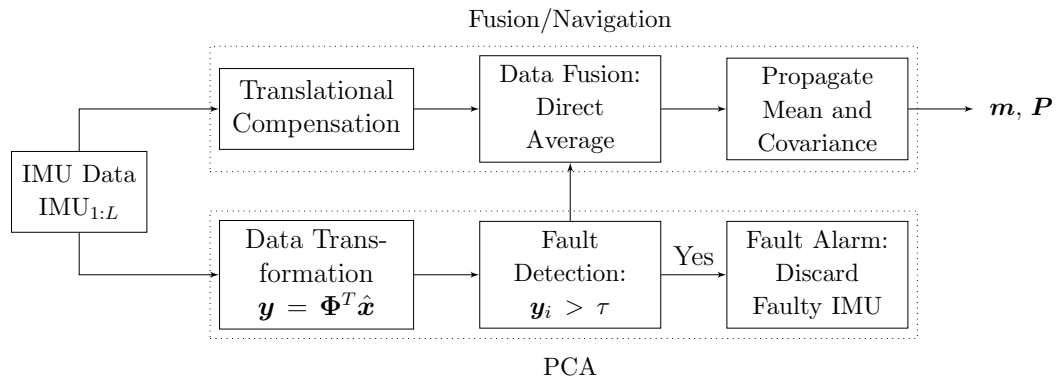


Figure 7.2. Fault Detection in a Static System

A dynamic system has a similar architecture regarding fusion/navigation and measurement fault detection using MPCA with the addition of a training vector fault detection algorithm. Recall that when using MPCA, a trusted measurement set, or the training vector, is used in the process of removing the system dynamics. As stated before, to ensure that the training vector does not fail, a CI fault detection scheme is implemented in the loop along with fusion/navigation and MPCA. This scheme applies CI relative to the transformed training vector in the feature plane allowing for a fault analysis by computing distances of the fused covariance matrices. A distance threshold is set to determine a fault in the training vector. This algorithmic addition must be computed at a lower frequency than MPCA and navigation in order

to store previous feature plane data sets to compute a fused covariance relative to the transformed training vector. This procedure is summarized and visualized in block diagram form in Figure 7.3.

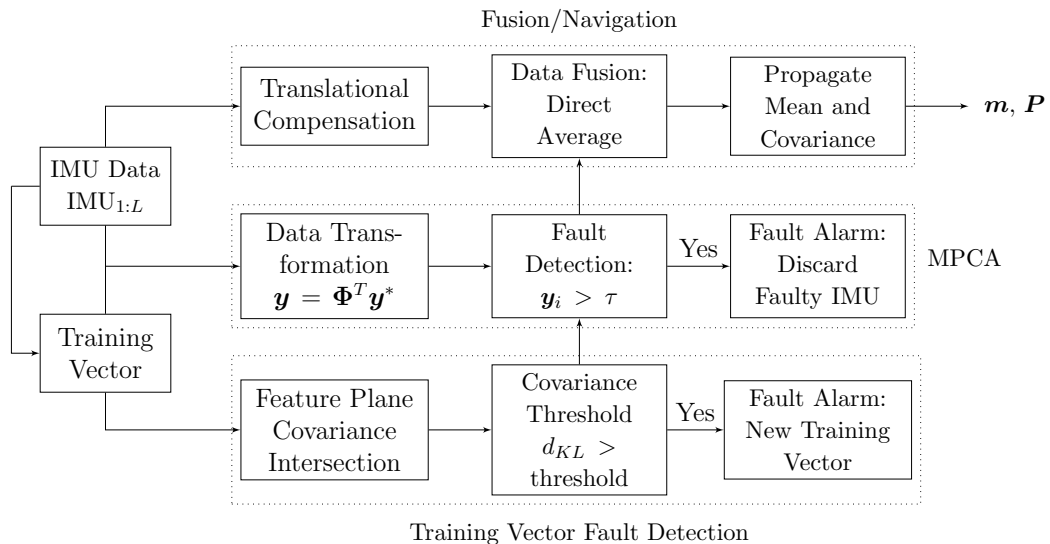


Figure 7.3. Fault Detection in a Dynamic System

7.1.2. Fusion Architecture. The proposed data fusion process is reiterated here for systematic overview purposes. The fusion process undergoes a direct averaging algorithm producing fused IMU measurements from N sensors. The fused measurements that have been passed through a needed fault detection algorithm are then passed into the navigation algorithm. As seen in the Fusion Methodology chapter, the direct averaging process can be seen in block diagram form in Figure 7.4. As seen in the fusion block diagram, this process is computationally and architecturally simple while producing measurements with a noise reduction factor of \sqrt{N} .

7.1.3. Navigation Architecture. The navigation architecture is defined as a predictor and corrector stage. The predictor computes the predicted mean m^- and covariance P^- . The predicted mean is computed as a function of the fused

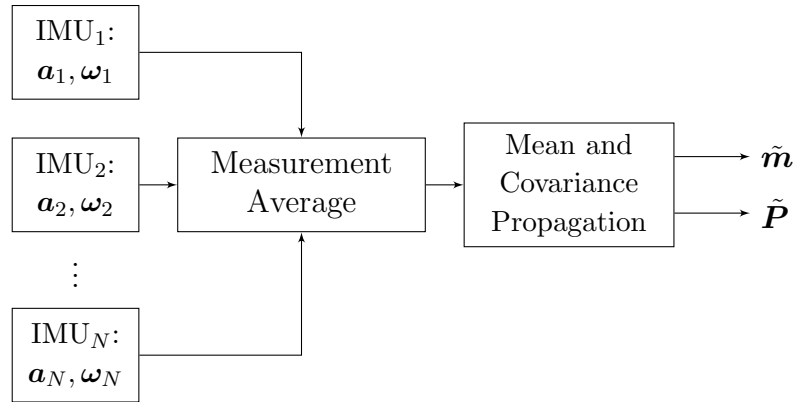


Figure 7.4. IMU Data Fusion Computed by Directly Averaging the Measurements

non-gravitational acceleration, $\tilde{\mathbf{a}}$, and angular velocity, $\tilde{\boldsymbol{\omega}}$, given by Eq. (2.12). This predicted solution is then fed into the corrector stage. The corrector takes data from external measurements \mathbf{z}_m and the predicted mean and covariance to compute the final navigation solution, i.e. mean \mathbf{m}^+ and covariance \mathbf{P}^+ . This process is summarized and shown in block diagram form in Figure 7.5.

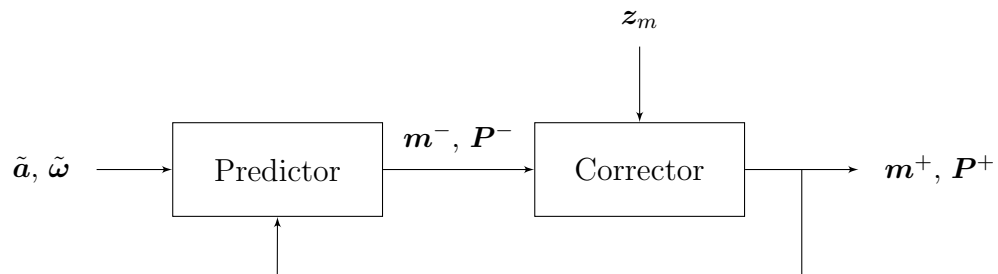


Figure 7.5. Navigation Architecture

7.2. SENSOR CONFIGURATION

In constructing a distributed network of redundant inertial sensors, orientation and configuration of the network brings to light the manner in which the network of

sensors is configured. In some cases, the overall system configuration may limit the placement of the individual sensors, such that they are placed to meet certain mass distribution or wiring constraints. Other situations, however, may allow for more freedom in the placement of the individual sensors. In this latter case, which is the case considered in this thesis, the issue becomes how to intelligently place and orient the sensors. In Reference [22], the problem of orienting a set of single-axis sensors is considered, and it is shown that by minimizing the cost function, defined as the trace of the measurement covariance, an optimal sensor configuration for navigation performance is found. The resulting optimal sensor configuration, denoted by \mathbf{H} , is found in Eq. (7.1) as

$$\mathbf{H}^T \mathbf{H} = \frac{n}{3} \mathbf{I}_{3 \times 3}, \quad (7.1)$$

where n is the number of sensors in the network. While this is only demonstrated for single-axis sensors, the same approach is used herein for three-axis sensors in order to emulate an intelligently designed sensor network. It should also be noted that nothing about the optimal sensor network implies that optimality is preserved when a fault is encountered. The design of fault-tolerant optimal placement and orientation, as well as the analysis of the sensor configuration, is not considered in this thesis; it is a topic for future study.

8. SIMULATION RESULTS

8.1. SIMULATION CONFIGURATION

To demonstrate the performance of the proposed direct averaging fusion rule and while running an underlying fault detection method, i.e. MPCA, a spacecraft navigation solution is examined. This simulation consists of a spacecraft with ten low-cost Microstrain 3DM-GX3-35¹ IMUs onboard, $N = 10$, and where all sensor axis orientations are assumed to be perfectly aligned, which gives the sensor orientation matrix to be

$$\mathbf{H} = \mathbf{0} \in \mathbb{R}^{N \times N}.$$

The initial conditions of the simulated spacecraft are given in orbital element form, i.e. semimajor axis a , eccentricity e , inclination i , argument of periapsis ω , right ascension of the ascending node Ω , and true anomaly ν . This set of orbital elements is given as

$$[a \ e \ i \ \omega \ \Omega \ \nu] = [1.05 \times 10^4 \text{ [km]} \ 0.200 \ 28.4989^\circ \ 0.00^\circ \ 0.00^\circ \ 0.00^\circ]$$

or, when converted to Cartesian coordinates shown as

$$\mathbf{r}_{t,0} = \begin{bmatrix} 8.400080 \times 10^3 & 0.00 & 0.00 \end{bmatrix}^T \text{ [km]}, \quad \mathbf{v}_{t,0} = \begin{bmatrix} 0.00 & 6.6316 & 3.6006 \end{bmatrix}^T \text{ [km/s]}.$$

With these true initial conditions, the true position for one orbit \mathbf{x}_t , along with the initial position $\mathbf{x}_{t,0}$ and initial velocity direction vector $\mathbf{v}_{t,0}$, are plotted and are shown in Figure 8.1. Note that the axes are scaled to the radius of the Earth, denoted by [ER]. An initial covariance is specified for the position, velocity, and attitude states,

¹<http://www.microstrain.com/inertial/3dm-gx3-35>

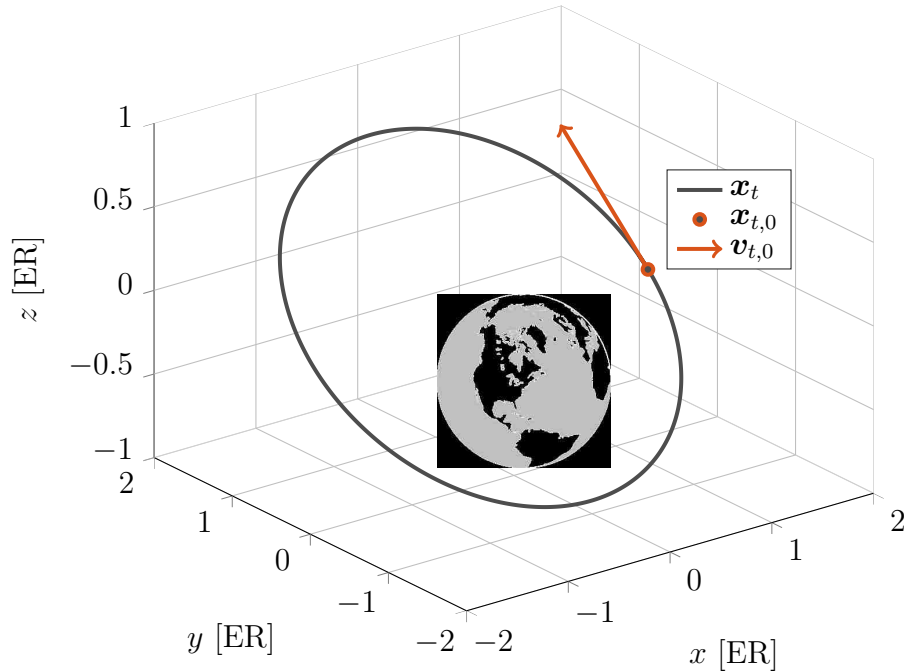


Figure 8.1. True Simulated Orbit Trajectory

while the respective standard deviations of the accelerometer $\sigma_{a,param}$, and gyroscope $\sigma_{g,param}$, are governed by the IMU data sheet shown in Appendix A. This initial covariance is given as

$$P_0 = S_0 S_0^T$$

The term S_0 is defined as the diagonal matrix of the state standard deviations shown as

$$S_0 = \text{diag} \left\{ \left[\sigma_r \quad \sigma_v \quad \sigma_\theta \quad \sigma_d \quad \sigma_{a,param} \quad \sigma_{g,param} \right] \right\},$$

where σ_d is defined as the standard deviation of the position of the IMU with respect to the spacecraft's CM. The initial position, velocity, and attitude standard deviations

are specified as

$$\boldsymbol{\sigma}_r = \begin{bmatrix} 20^2 \\ 20^2 \\ 20^2 \end{bmatrix} [m], \quad \boldsymbol{\sigma}_v = \begin{bmatrix} 3^2 \\ 3^2 \\ 3^2 \end{bmatrix} [m/s], \quad \boldsymbol{\sigma}_\theta = \begin{bmatrix} 0.05^2 \\ 0.05^2 \\ 0.05^2 \end{bmatrix} [rad].$$

For simplification purposes, this position vector is assumed to be perfectly known; therefore, $\boldsymbol{\sigma}_d = \mathbf{0}$. While this creates a non-positive definite initial covariance matrix, it does not create any singularities in the system. This initial covariance is used to generate a Gaussian initial condition distribution centered on the initial truth \mathbf{x}_0 , which is distributed as

$$\mathbf{m}_0 \sim \mathcal{N}(\mathbf{x}_0, \mathbf{P}_0)$$

The true non-gravitational acceleration and angular velocities of the simulated IMUs are simulated at 100 Hz using AGI STK¹ with a constant angular velocity along each axis. These true non-gravitational accelerations $\mathbf{a}_{t,ng}$ and inertial angular velocities $\boldsymbol{\omega}_{i/b}^i$, both outputted respectively from STK, are plotted over time and are shown in Figure 8.2. As discussed prior in the navigation section, a position and a quaternion measurement update are used in the overall navigation process via the EKF. Position and quaternion measurements are received at 0.2 Hz and 0.1 Hz, respectively. The times at which measurements are received throughout the simulation are shown in Figure 8.3, which are denoted when \mathbf{z}_r and \mathbf{z}_q are pulled to the value of one. The simulation and navigation solutions are examined for only 500 seconds, $t_f = 500$ [s], as seen in the time axis of the received measurements in Figure 8.3. Multiple cases are examined to better understand the characteristics and behavior of MPCA during the event of sensor failures. The first case examined contains no sensor failures,

¹<http://www.agi.com/products/stk/>

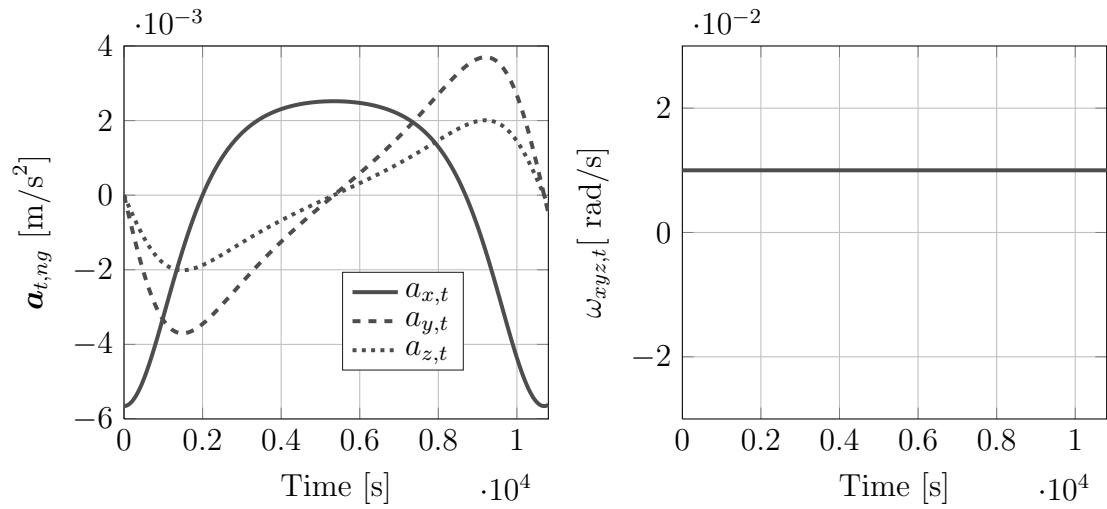


Figure 8.2. True Simulated \mathbf{a}_{ng}^i and $\omega_{i/b}^i$

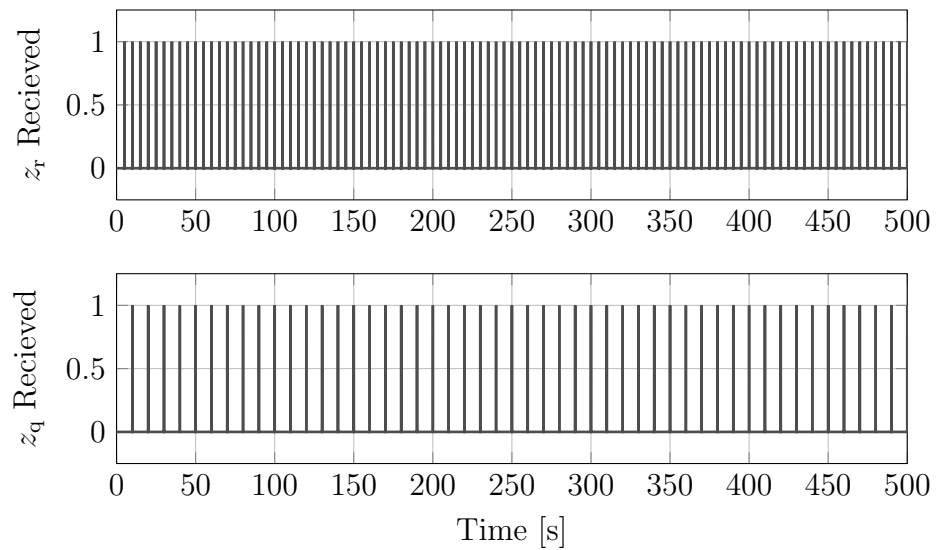


Figure 8.3. Time of Received Measurements

but provides a baseline navigation solution that is used to compare against that of off-nominal sensor cases. The next three cases examine different sensor failure scenarios; the first being a single IMU failure, the second being multiple IMU failures, and the last containing a failure in the crucial training vector. The overview of all four cases examined in the results is found in Table 8.1.

Table 8.1. Results Overview

Case 1	Nominal Operating Conditions
Case 2	Single IMU Failure
Case 3	Multiple IMU Failure
Case 4	Training Vector Failure

8.2. CASE 1: NOMINAL SENSOR OPERATION

The first simulated case study examines the behavior of a homogeneous multi-IMU network undergoing nominal operating conditions; i.e., no sensor failures. The navigation results of this fusion network create a performance baseline, providing useful information in the examination of the sensor failure cases. A Monte Carlo analysis is performed on this nominal case to verify the fusion filter algorithms. The position and velocity $3 - \sigma$ standard deviations, and their respective root sum square (RSS) errors and mean errors are plotted versus time. These errors are defined respectively as

$$\sigma_{RSS,k} = \sqrt{\text{trace}(\mathbf{P}_k)} \quad \text{and} \quad \mathbf{e}_k = \|\mathbf{x}_k - \mathbf{m}_k\|.$$

and are shown in Figures 8.4 and 8.5, respectively.

It is seen that the position and velocity $3 - \sigma$ standard deviations accurately envelop their respective errors. It is important to note that the Monte Carlo simula-

tion is only run for 100 samples, due to computational time, which accounts for the slight deviation in the Monte Carlo and fusion filter uncertainty. Note that the Monte Carlo algorithm is shown in Appendix D. The attitude $3 - \sigma$ standard deviation, and its respective RSS error and mean errors, are plotted versus time and are shown in Figure 8.6. Note that all of the IMU error parameter standard deviations/errors are shown in Appendix F.

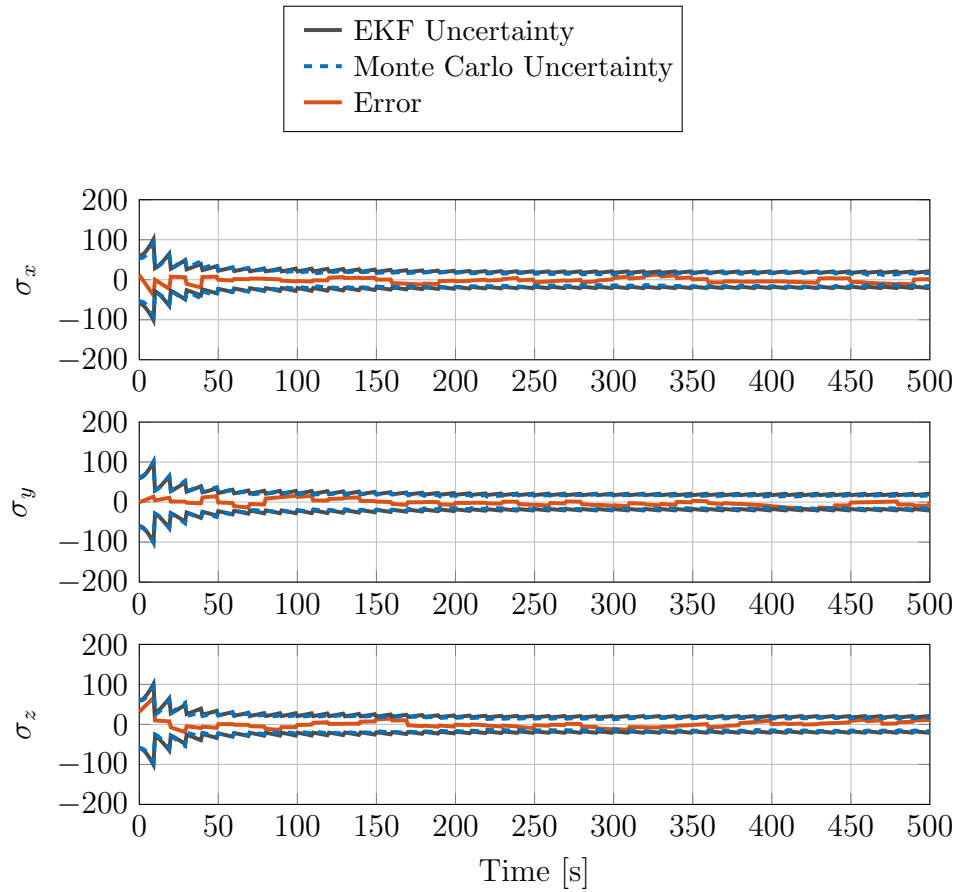
Recall that due to the multiplicative properties of the quaternion state, the rotation error is defined as twice the vector part of the quaternion error, which yields

$$\mathbf{e}_\theta = 2\text{vec}(\bar{\mathbf{q}} \otimes \hat{\mathbf{q}}).$$

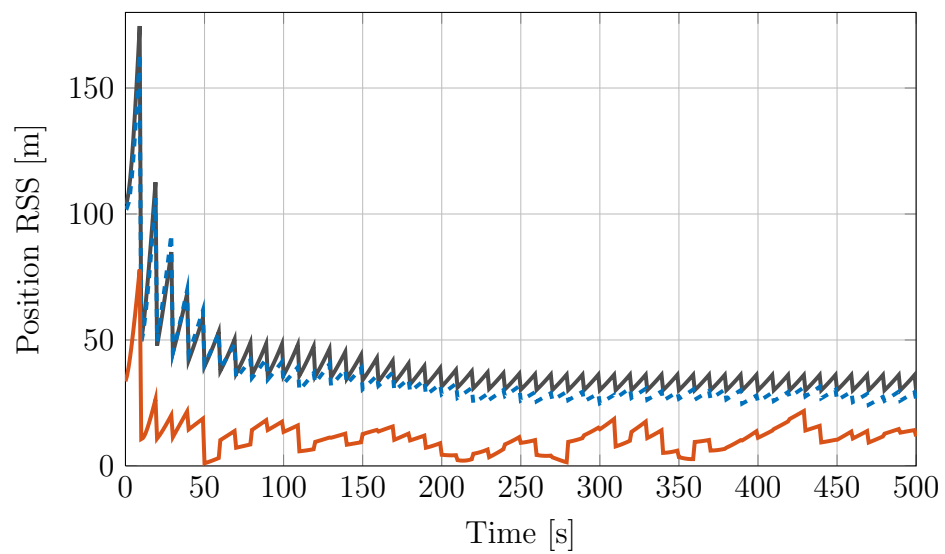
MPCA is now examined for the nominal operation case, where the first two principle components, i.e. y_1 and y_2 , are considered in the process of fault detection with a fault threshold of $\tau = 1.5 \times 10^{-3}$. A point cloud representation of the principal components is plotted along with y_1 versus time and can be found in Figures 8.7a and 8.8, respectively.

It can be seen that both principal components stay within the specified fault threshold, as expected from a sensor cluster without failure, signaling no fault flags in the fusion process. The system can then determine if it is needed to switch training vectors to that of a nominal sensor, in the case of a failing training vector, if the feature plane CI solution exceeds its expected threshold, $\tau = 0.008$. Note that the threshold values are predetermined, user specified constants in this case and future study will examine more intelligent selections of these thresholds. Figure 8.8 exhibits the feature plane CI solution, σ_{CI} , time history along with its N associated principal component standard deviations, σ_{y1} . Note that \mathbf{y}_1 consisted of N transformed data points that correspond to its respective sensor. As expected, in the event of no training vector failure, the feature plane Covariance Intersection solution maintains a

continuous nominal status defined by the given threshold. Note that the threshold is arbitrarily selected in this case which could lead to inaccurate thresholding solutions if walking biases become evident in the transformed linear subspace. Thresholding propagation and intelligent selection are to be a topic of future study.

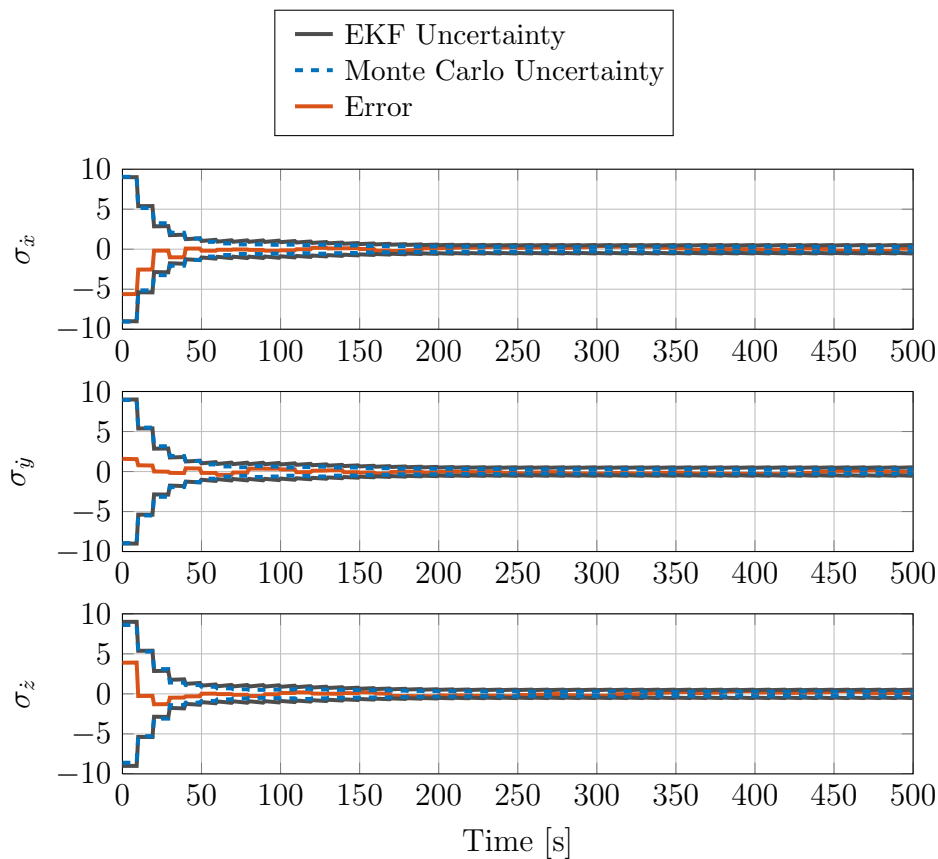


(a) Position Uncertainty [m]

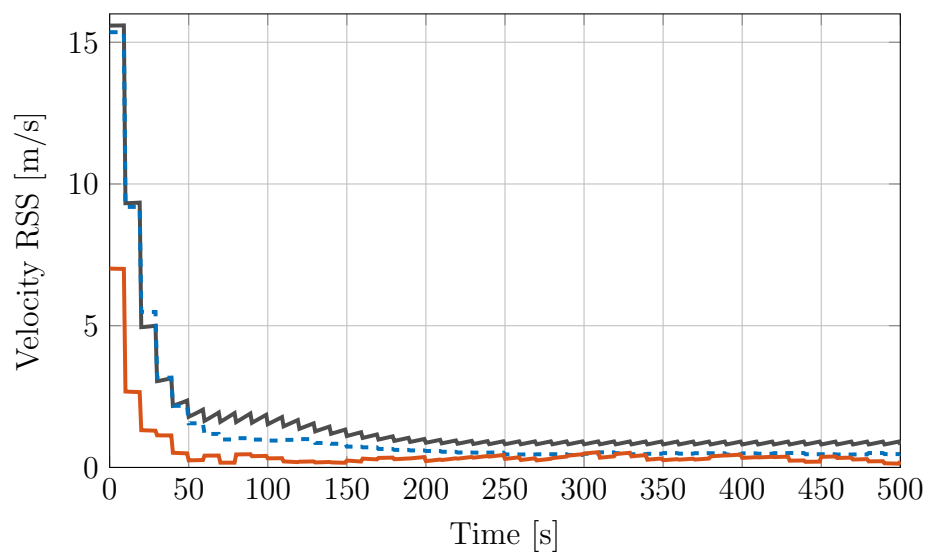


(b) Position RSS [m]

Figure 8.4. Case 1: Position Standard Deviation/Errors and RSS vs. Time



(a) Velocity Uncertainty [m/s]



(b) Velocity RSS [m/s]

Figure 8.5. Case 1: Position Standard Deviation/Errors and RSS vs. Time

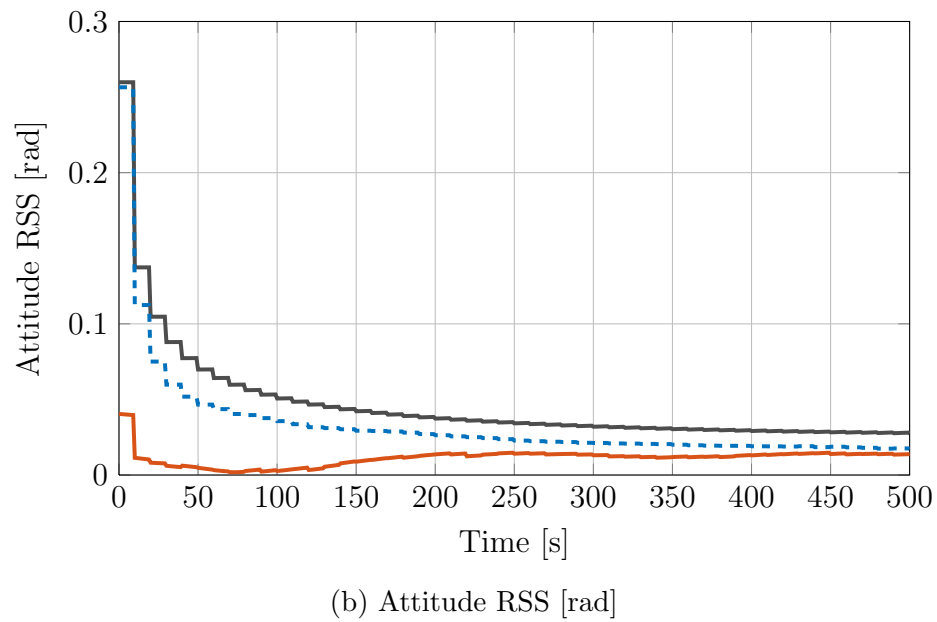
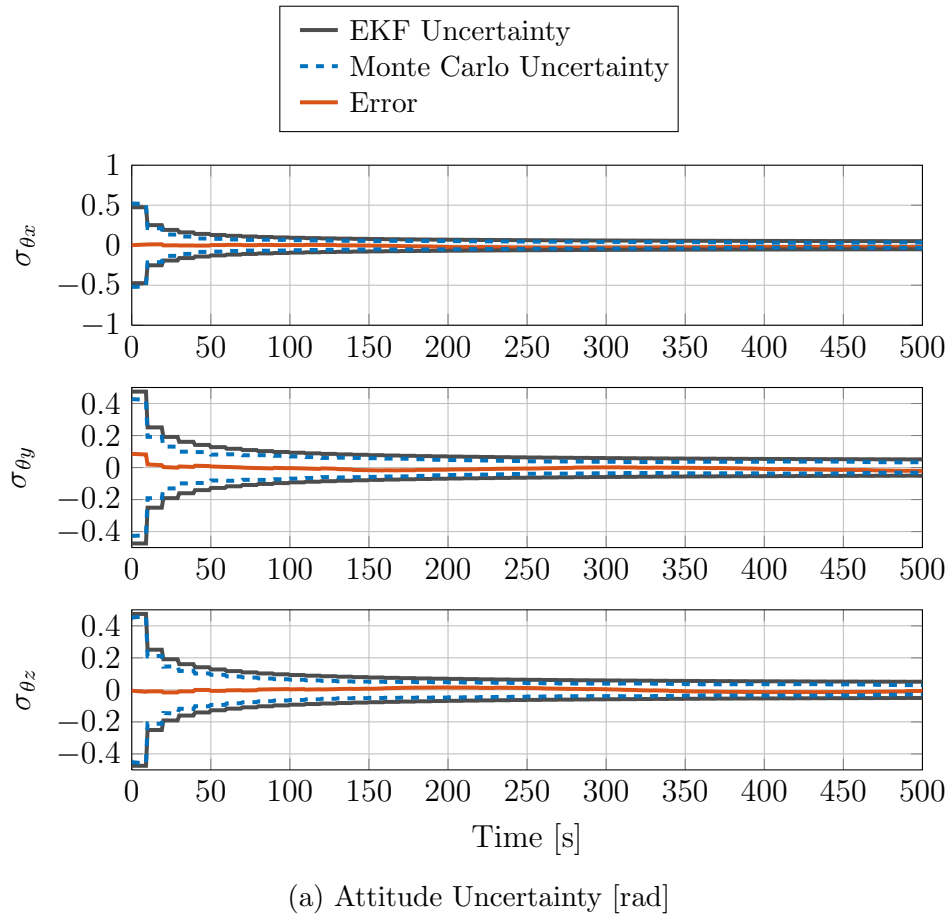


Figure 8.6. Case 1: Attitude Standard Deviation/Errors and RSS vs. Time

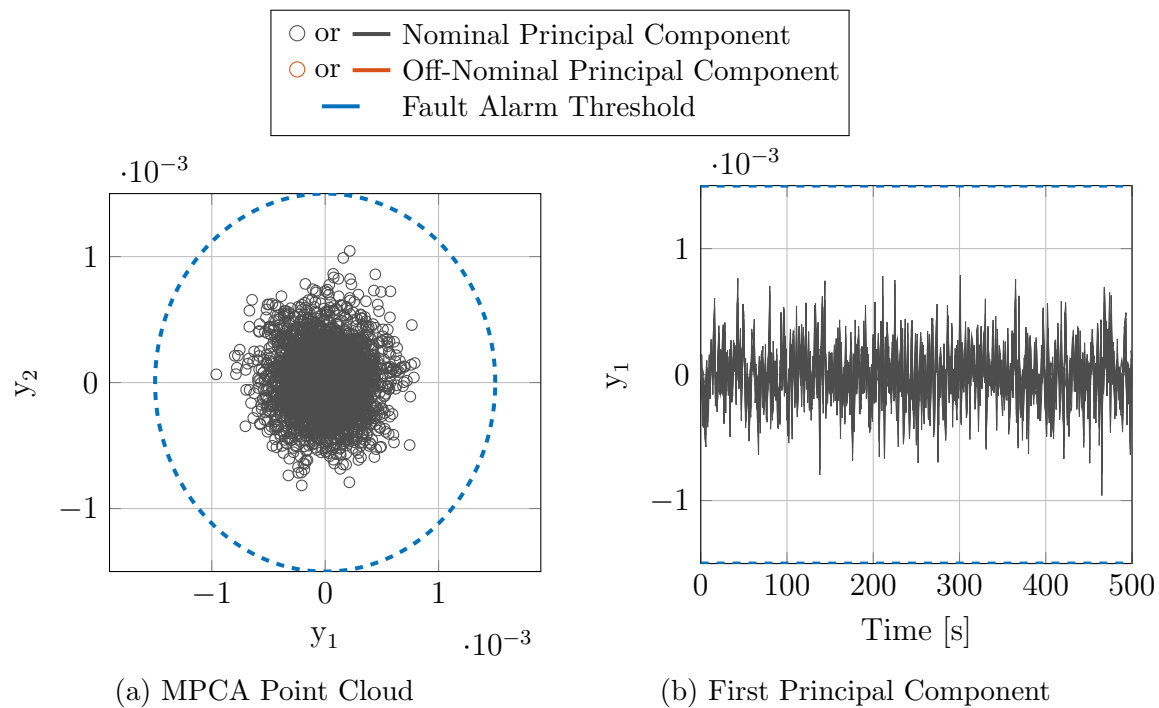


Figure 8.7. Case 1: Effects of MPCA with no Failures

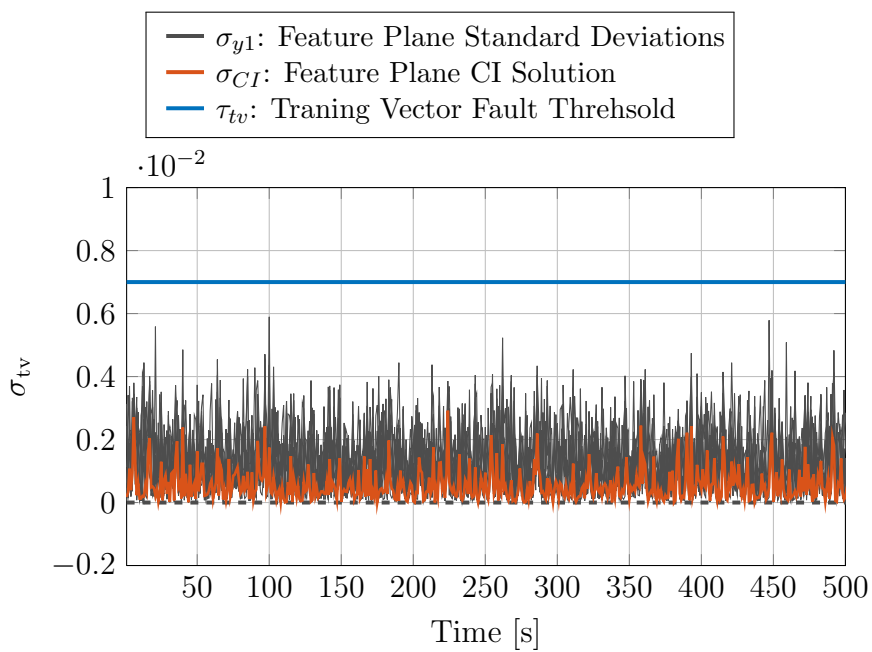


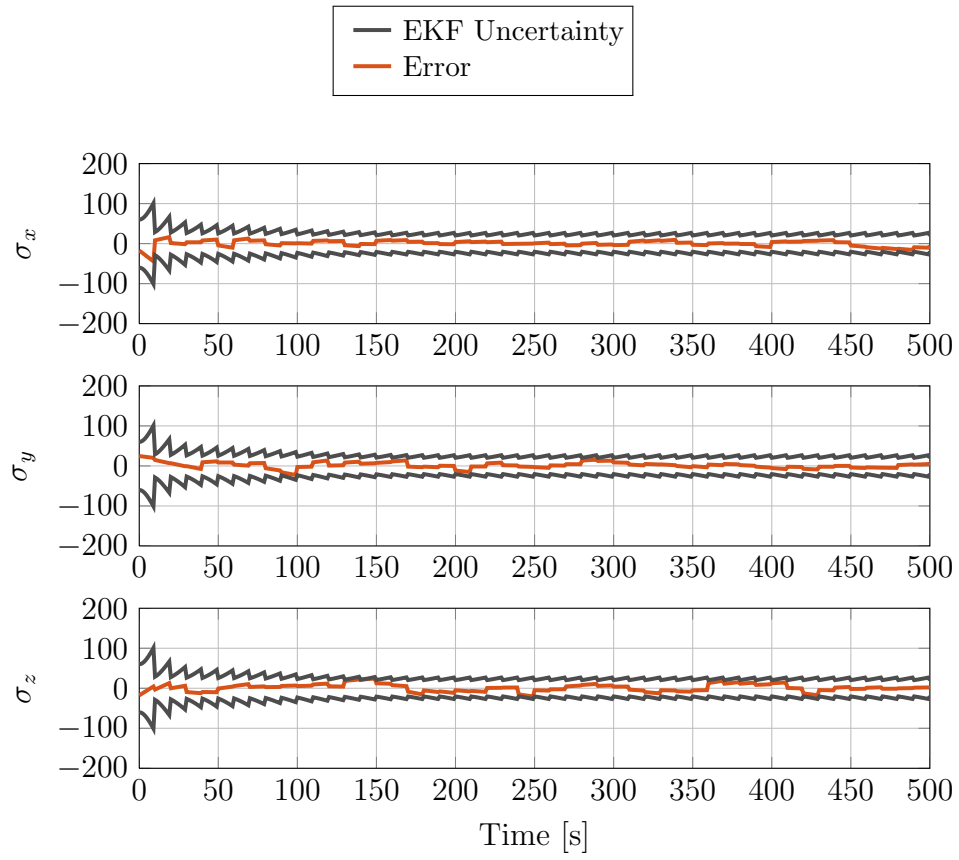
Figure 8.8. Case 1: Feature Plane CI Training Vector Fault Detection

8.3. CASE 2: SINGLE SENSOR FAILURE

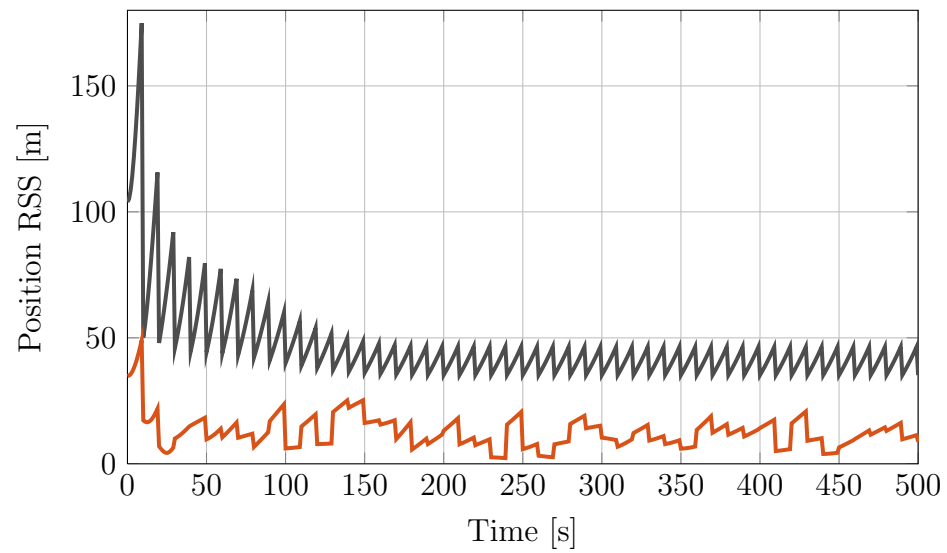
The second simulated case study examines the behavior of a homogeneous multi-IMU network undergoing a single IMU failure at $t = 250$ seconds into the simulation. The sensor fault is modeled as a sudden increase in bias and noise. The position and velocity $3 - \sigma$ standard deviations, and their respective root sum square (RSS) errors and mean errors, are plotted versus time and are shown in Figures 8.9 and 8.10, respectively. It is seen that the position and velocity $3 - \sigma$ standard deviations accurately envelop their respective errors with a single failure. The attitude $3 - \sigma$ standard deviation, and its respective RSS error and mean errors, are plotted versus time and is shown in Figure 8.11.

MPCA is now examined for the case of a single failure where the first two principle components, i.e. y_1 and y_2 , are considered in the process of fault detection with a fault threshold of $\tau = 1.5 \times 10^{-3}$. A point cloud representation of the principal components is plotted along with y_1 versus time and can be found in Figures 8.12a and 8.13, respectively.

It can be seen that MPCA successfully detects and isolates the failing sensor and eliminates the outliers from the data fusion process. In certain cases [18], if the sensor is re-calibrated or noisy faulty data points re-enter the threshold, the data could then be processed again; however, it is practical to completely eliminate a failure from the system in the case of a failure. As before, the underlying training vector fault detection method was processed at 1 Hz, and the feature plane deviations away from the training vector's transformed data are shown in Figure 8.13.

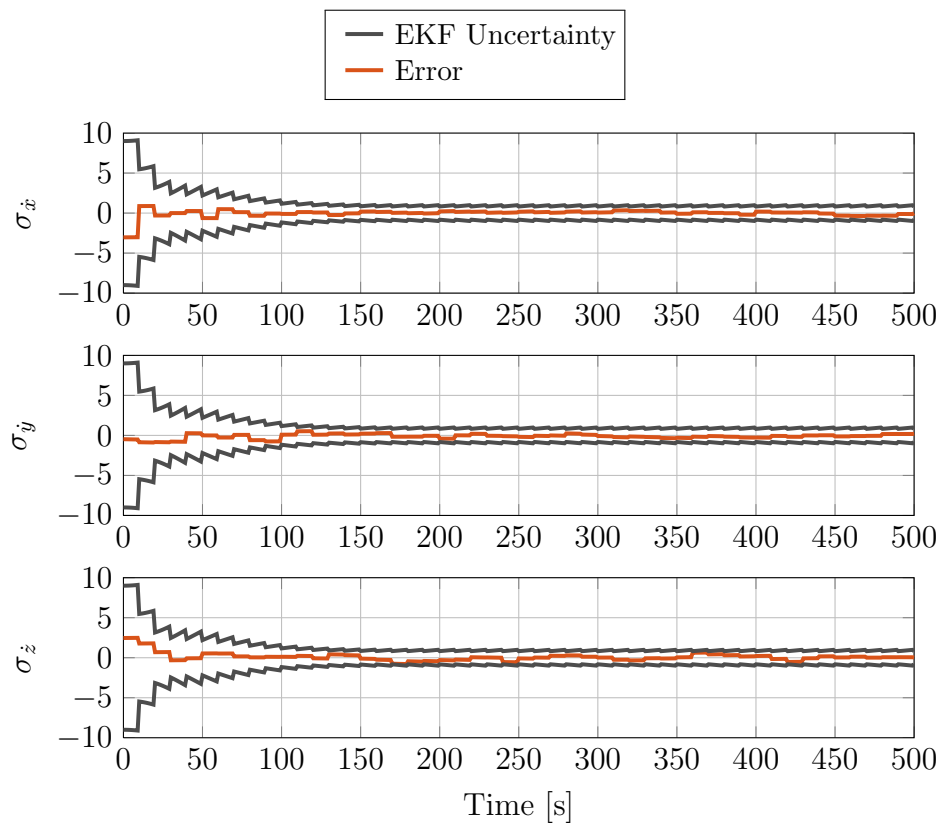


(a) Position Uncertainty [m]

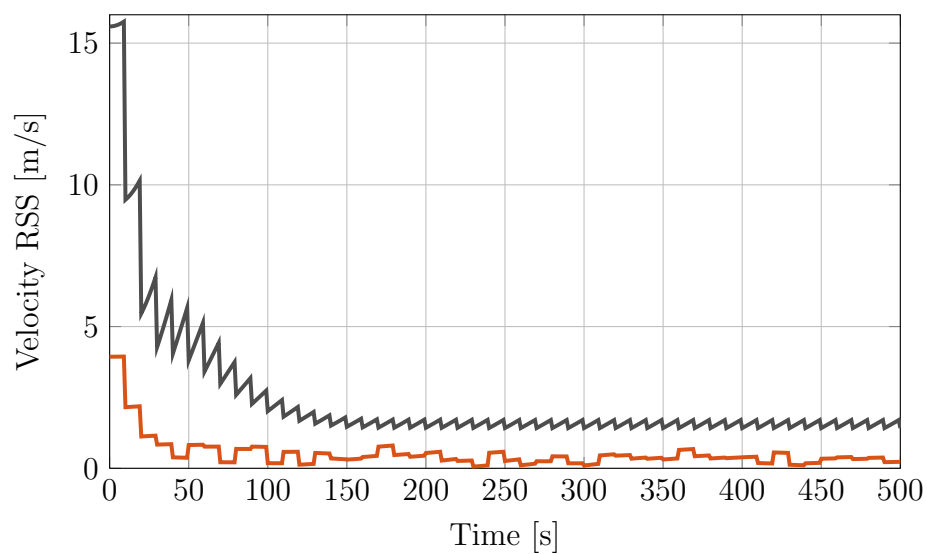


(b) Position RSS [m]

Figure 8.9. Case 2: Position Standard Deviation/Errors and RSS vs. Time

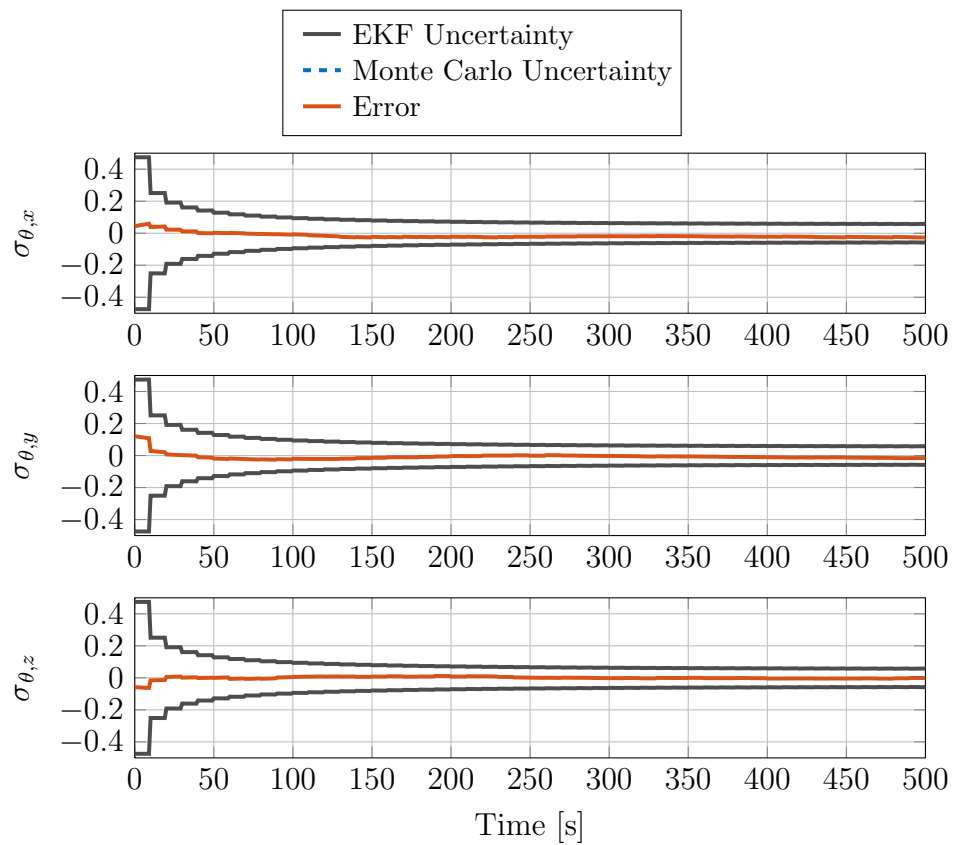


(a) Velocity Uncertainty [m/s]

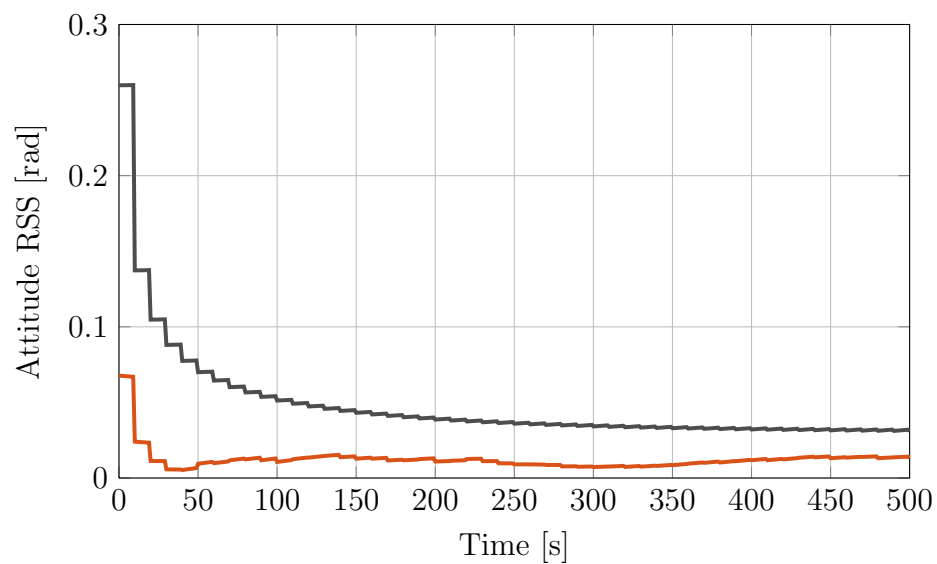


(b) Velocity RSS [m/s]

Figure 8.10. Case 2: Velocity Standard Deviation/Errors and RSS vs. Time



(a) Attitude Uncertainty [rad]



(b) Attitude Magnitude Uncertainty

Figure 8.11. Case 2: Attitude Standard Deviation/Errors and RSS vs. Time

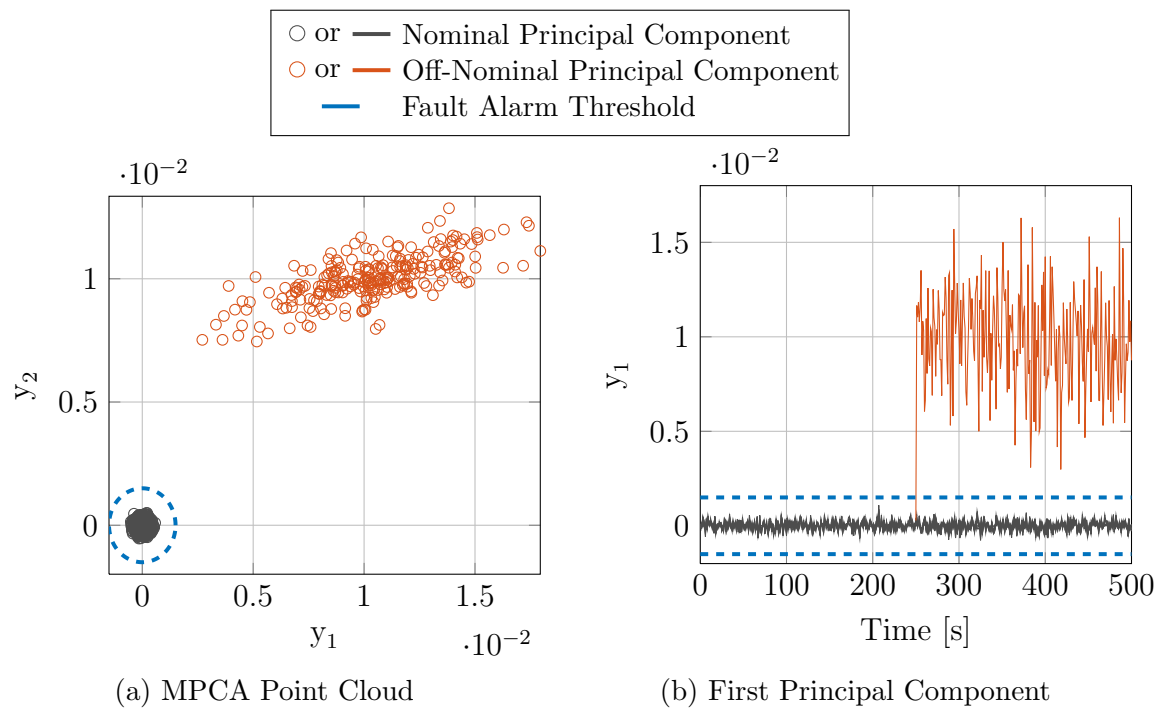


Figure 8.12. Case 2: Effects of MPCA with a Single IMU Failure

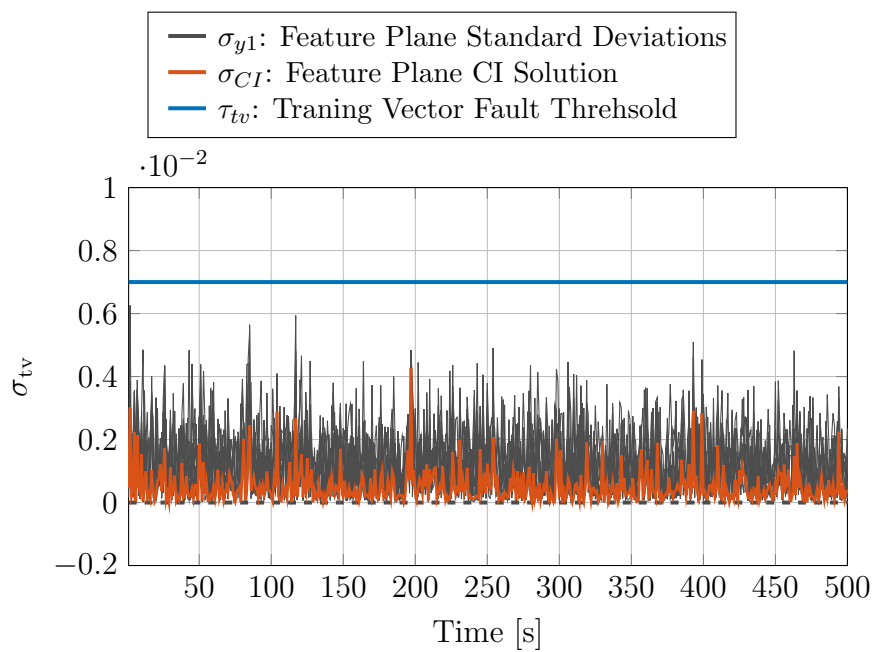


Figure 8.13. Case 2: Feature Plane CI Training Vector Fault Detection

8.4. CASE 3: MULTIPLE SENSOR FAILURES

The third case examined looks at a system with multiple sensor failures, specifically three failures in this case. All three sensors fail at the same time, $t = 250$ seconds, and it is desired to examine the performance of the fault detection process with multiple failures. The navigation solutions for this case are omitted for brevity due to performance similarities to that of the previous two cases. However, it can be seen that MPCA successfully detects and isolates the failing sensors and eliminates the outliers from the data fusion process. A point cloud representation of the principal components is plotted along with y_1 versus time and can be found in Figures 8.14a and 8.15, respectively. While there is no failure in the training vector, the underlying training vector fault detection method is examined, which is processed at 1 Hz. The feature plane deviations away from the training vector's transformed data are shown in Figure 8.13. It is important to note that even in the case of three sensor failures, the training vector fault detection process remains undisturbed and is nominally operating as expected.

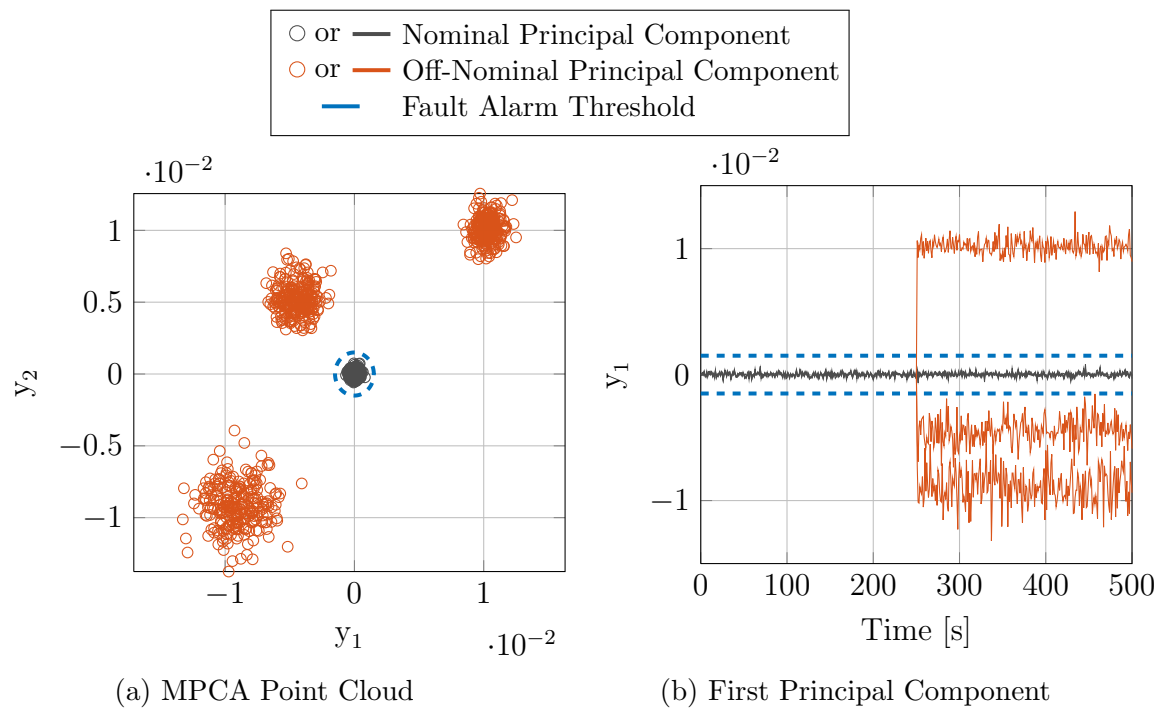


Figure 8.14. Case 3: Effects of MPCA with Multiple IMU Failures

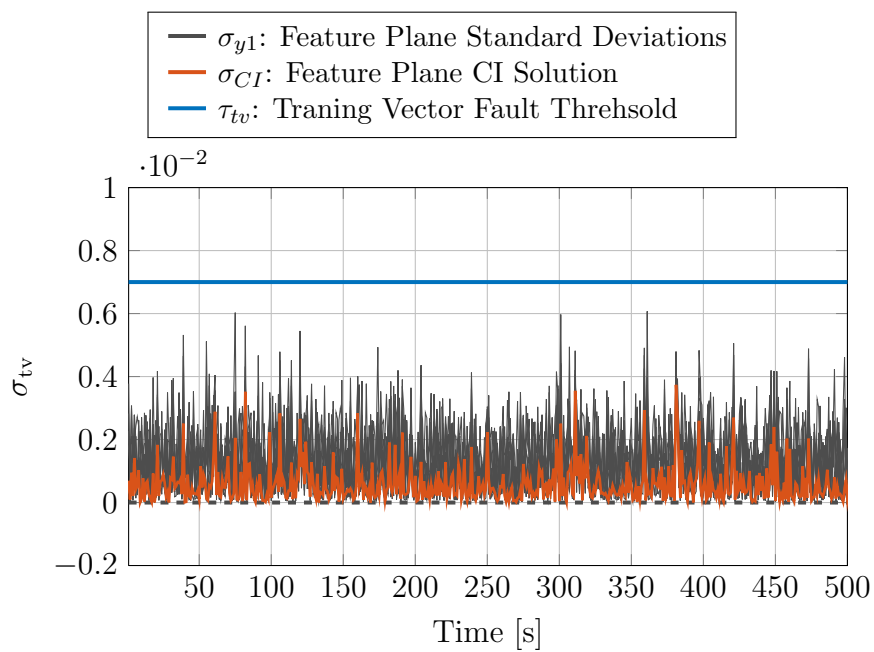


Figure 8.15. Case 3: Feature Plane CI Training Vector Fault Detection

8.5. CASE 4: TRAINING VECTOR SENSOR FAILURE

The last simulated case study examines the behavior of a homogeneous multi-IMU network undergoing a sensor failure at $t = 250$ seconds. In this case, the failure occurs in the IMU that is being used as the training vector for MPCA. It is desired to examine the performance of the proposed underlying fault detection method for the training vector within MPCA along with the respective navigation solutions. Before examining performance results of this method, an illustration is shown to display the need for this method. First, an examination of the system without training vector fault detection is considered. Figure 8.16 shows the feature plane during a training vector failure without using this feature plane training vector fault detection method.

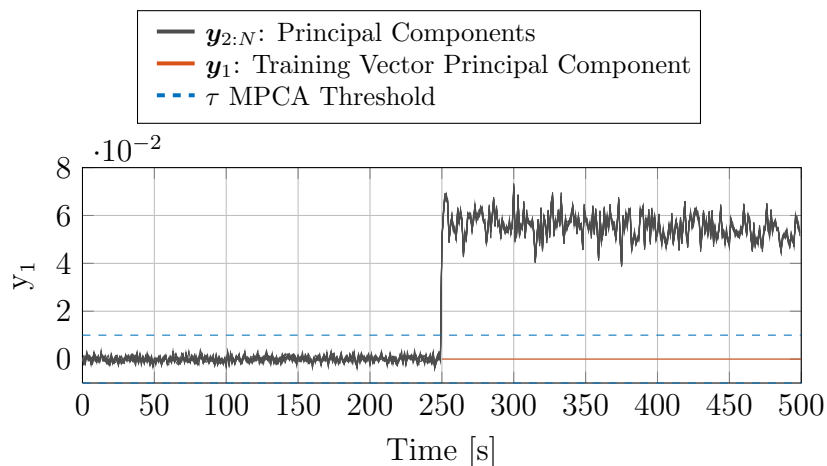
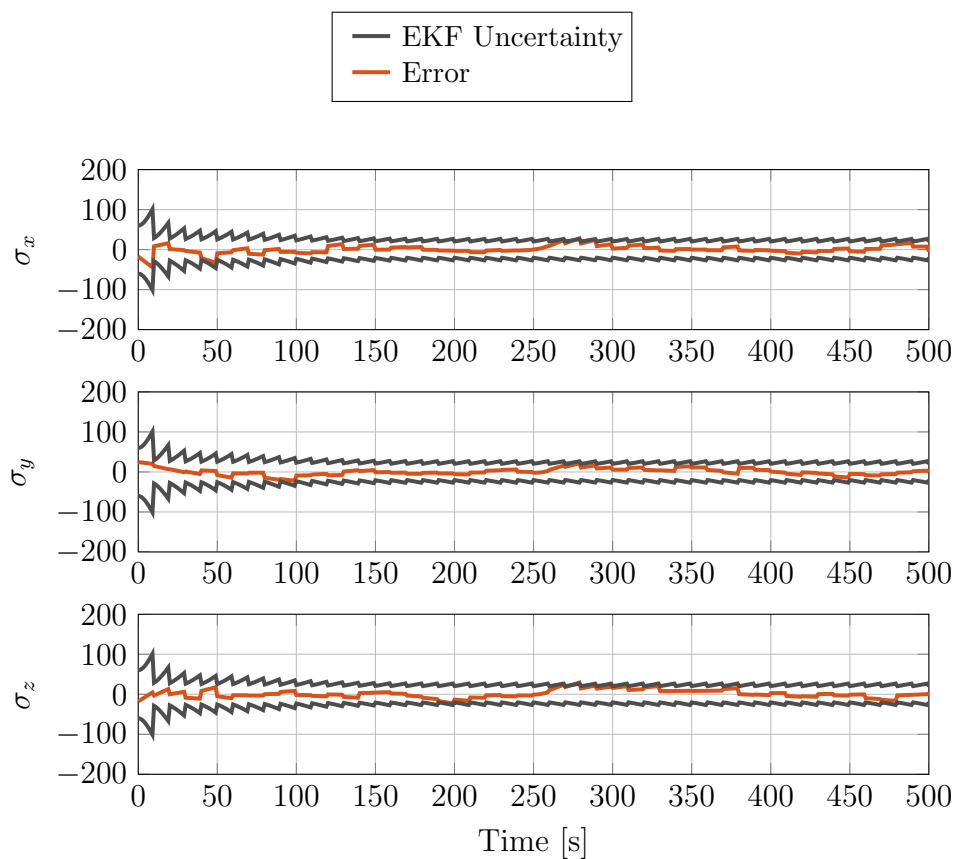


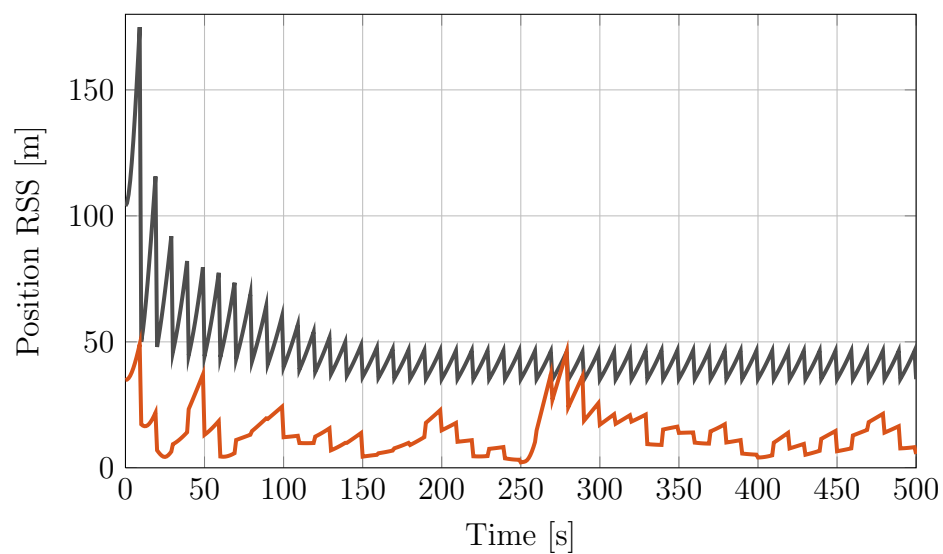
Figure 8.16. Case 4: Training Vector Failure using Only MPCA

Now, the underlying fault detection method of the training vector is implemented. The position and velocity $3 - \sigma$ standard deviations, and their respective root sum square (RSS) errors and mean errors, are plotted versus time and are shown in Figures 8.17 and 8.18, respectively. The attitude $3 - \sigma$ standard deviation, and

its respective RSS error and mean errors, are plotted versus time and are shown in Figure 8.19. It is seen that at the time of failure, the position, velocity, and attitude increase in error, unlike the other failure cases. This is due to the training vector fault detection frequency. Since this training vector fault detection method uses the previous j iterations of the feature plane covariance time history, there is a brief period where the filter believes the measurements of the faulty training vector, which results in a non-negligible error. However, as shown in Figures 8.17, 8.18, and 8.19, the filter does not diverge and is able to maintain a nominal status due to this underlying feature plane fault detection. The training vector is then switched based on the large increases in the feature plane deviations, which is shown later. A point cloud representation of the principal components are plotted along with the y_1 component versus time and can be seen in Figures 8.20a and 8.20b respectively. Note that the gold points denote the initial believed failing sensors. In the event of a training vector failure, the system maintains its belief that the training vector is correct even if all other sensors have failed. The system can then determine when to toggle to a nominal training vector once the feature plane CI solution exceeds its expected threshold, $\tau = 0.008$. Figure 8.21a exhibits the feature plane CI solution, σ_{CI} , time history along with its N corresponding transformed data point standard deviations relative to the training vector, σ_{y_1} . The same graphic is zoomed in, shown in Figure 8.21b, to illustrate the behavior of this method in a clearer manner. Note that the feature plane CI solution maintains a nominal condition after the training vector is toggled from the failing sensor to a new nominal sensor. From this, it is seen that MPCA can handle training vector failures without suffering filter divergence in the event where this underlying training vector fault detection method is performed.

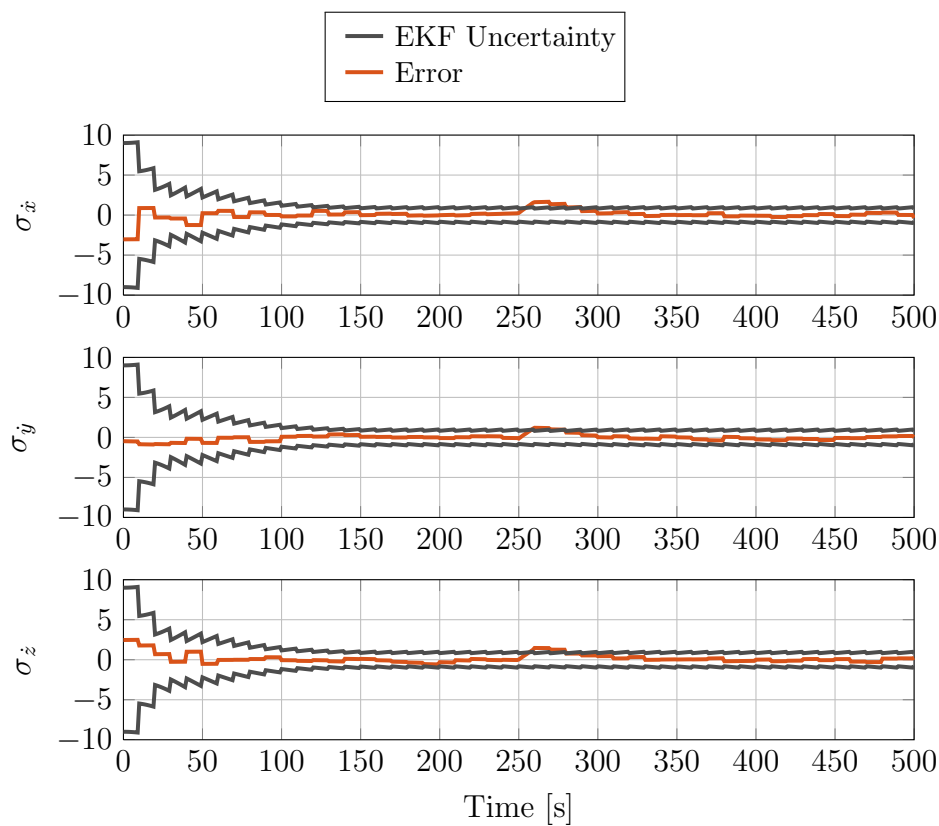


(a) Position Uncertainty [m]

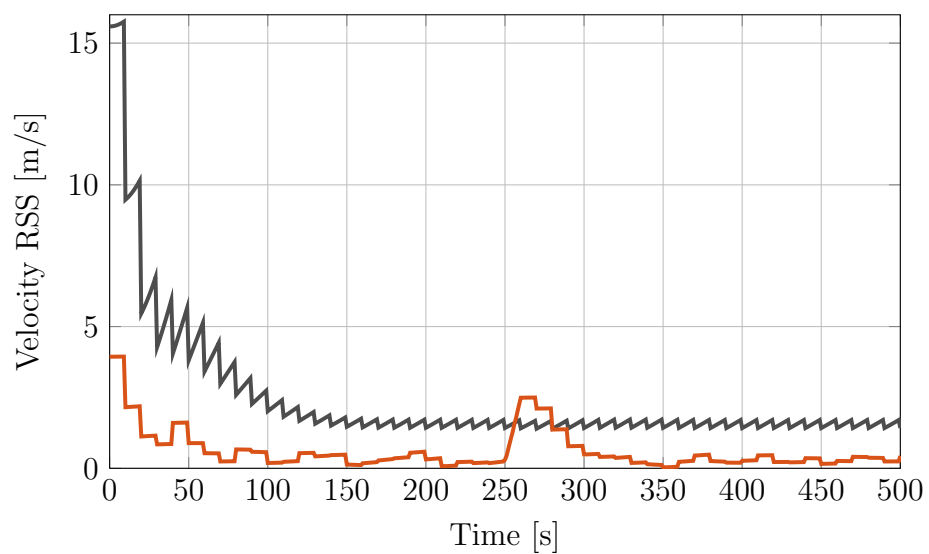


(b) Position RSS [m]

Figure 8.17. Case 4: Position Standard Deviation/Errors and RSS vs. Time

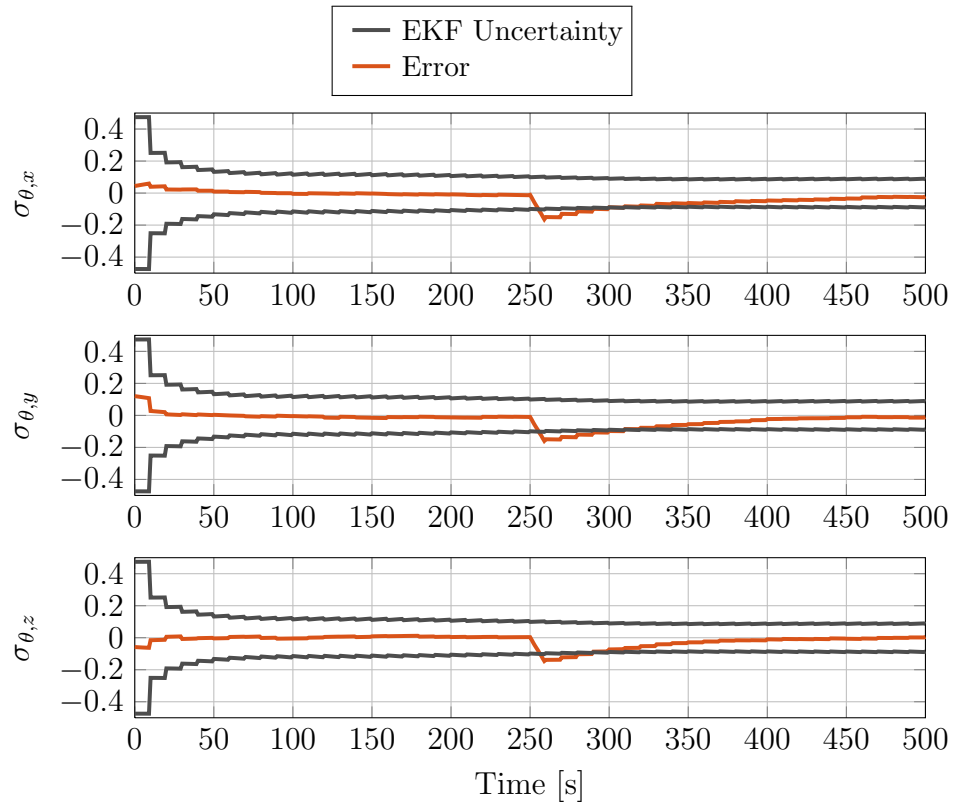


(a) Velocity Uncertainty [m/s]

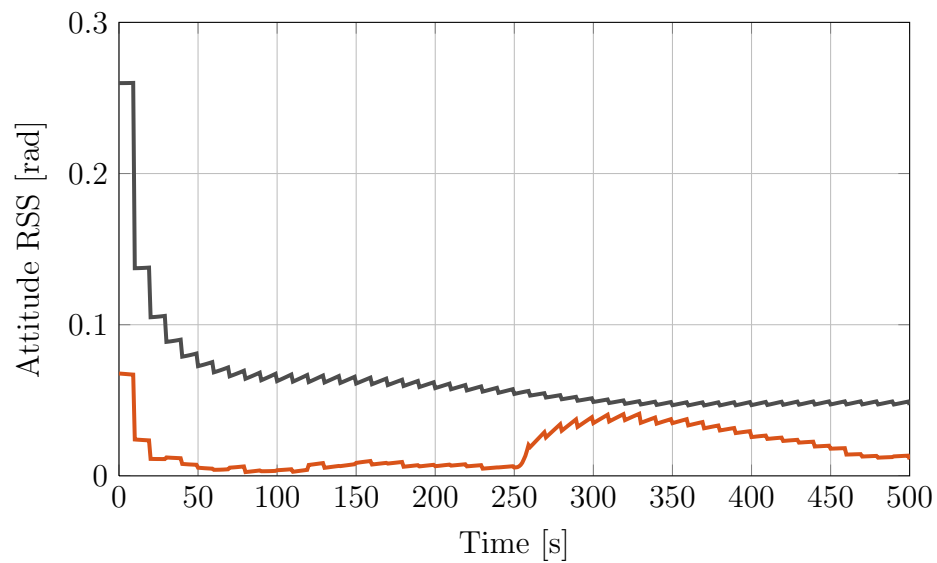


(b) Velocity RSS [m/s]

Figure 8.18. Case 4: Velocity Standard Deviation/Errors and RSS vs. Time



(a) Attitude Uncertainty [rad]



(b) Attitude RSS [rad]

Figure 8.19. Case 4: Attitude Standard Deviation/Errors and RSS vs. Time

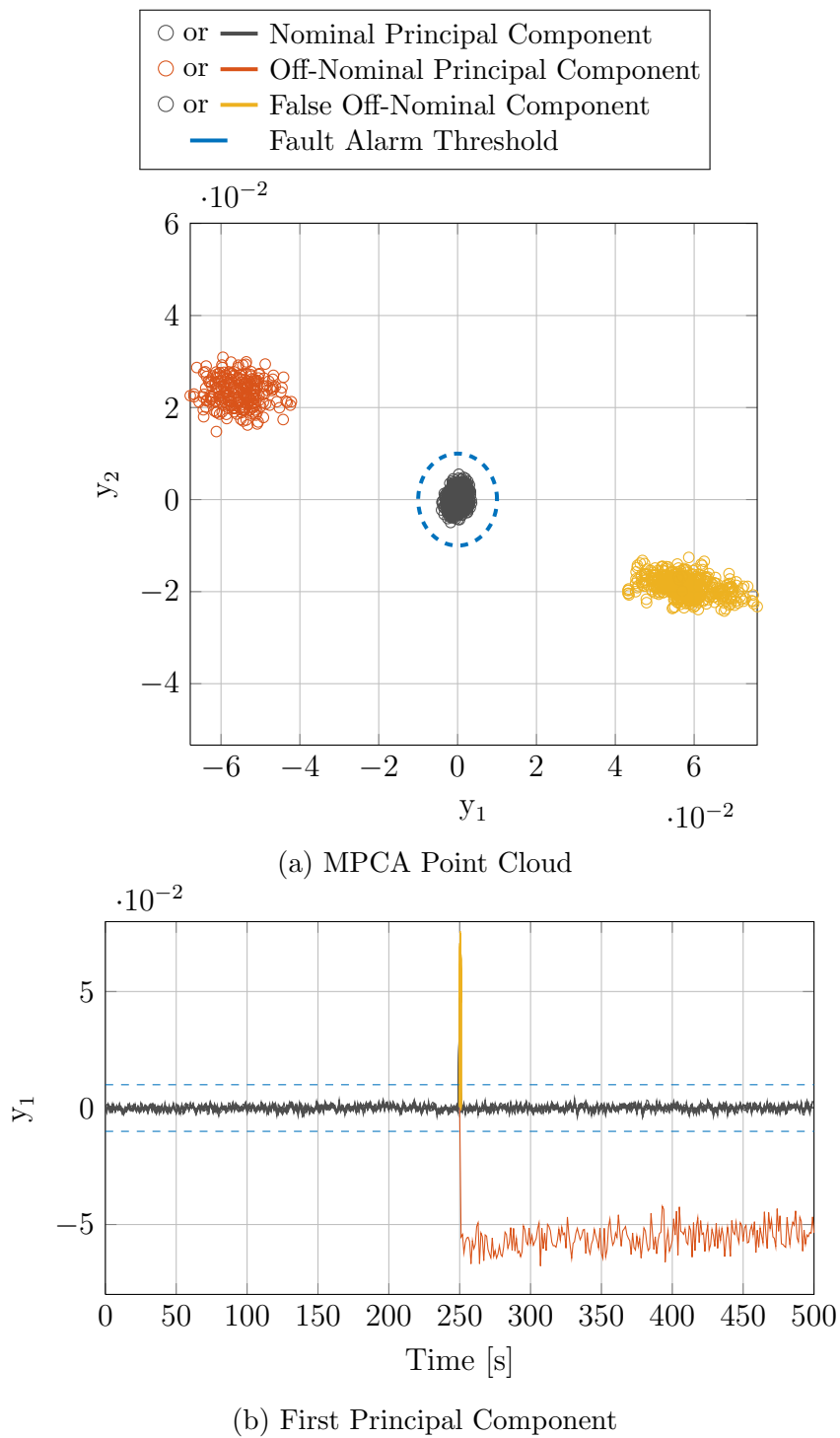
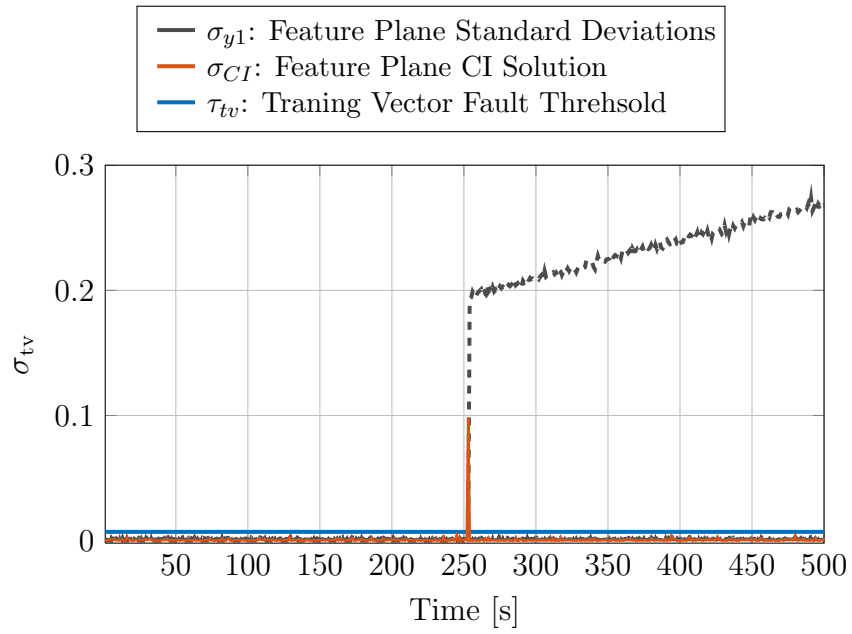
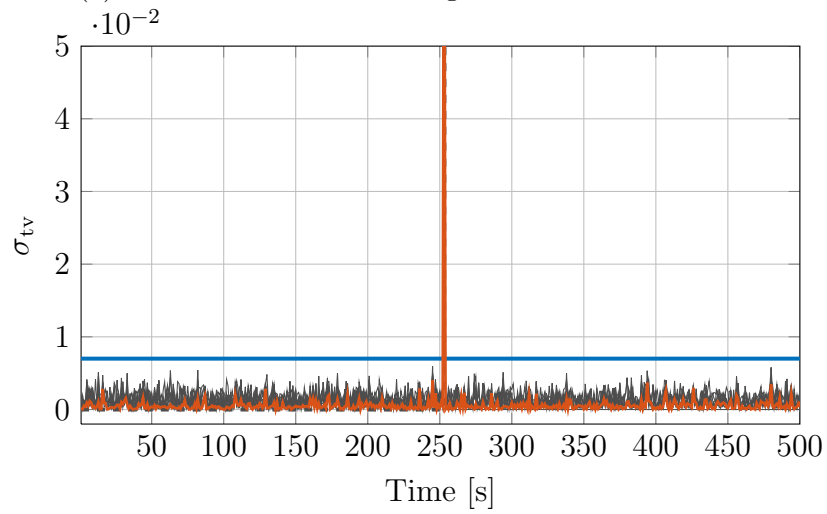


Figure 8.20. Case 4: Effects of MPCA with a Training Vector Failure



(a) Feature Plane CI Training Vector Fault Detection



(b) Feature Plane CI Training Vector Fault Detection

Figure 8.21. Case 4: Training Vector Fault Detection

9. CONCLUSIONS

In order to provide a lower-cost inertial measurement unit (IMU)-based navigation system as opposed to one that uses a typical high-cost, high-reliability tactical or strategic grade IMU, a fault-tolerant method is proposed that uses multiple low-cost IMUs and fuses their data together via a direct averaging method or their individual navigation solutions together via the covariance intersection (CI) fusion rule. It is demonstrated that the direct averaging fusion rule tends to outperform the CI fusion rule during nominal operations where no hardware failures are present. On the other hand, the CI fusion rule is shown to outperform the direct averaging fusion rule when an IMU fails due to the ability of the CI fusion rule to selectively downweight the navigation solution obtained from the failed IMU. The proposed algorithm uses the navigation solution propagated using the direct averaged data when there are no IMU faults since it provides a more confident solution. In the case of an IMU failure, the algorithm will recognize the failure, and the navigation solution from the covariance intersection method will be used. This distributed IMU network can provide a comparable navigation solution to that of a single, high-cost, high reliability IMU at significantly reduced cost. However, CI will only outperform direct averaging if there is knowledge of a sensor failure due to the limitation of the weighting scheme proposed. Since both fusion methods suggest needs for sensor fault detection methods, the direct averaging fusion rule is implemented. The direct averaging rule produces more accurate navigation solutions, i.e. mean and covariance, while also remaining computationally efficient.

In a distributed network of low-cost IMUs, sensor failures have been known to happen and were dealt with in this thesis using principal component analysis (PCA) for static systems and a modified version of principal component analysis (MPCA) for dynamical systems. Three different failure cases were examined to determine the

robustness of the proposed fault detection algorithms, which were compared to a baseline nominal (no-failure) case. The first failure case took into account a single IMU failure after an arbitrarily set time. The MPCA fault detection algorithm realized the fault and isolated the faulty measurements from the fusion process allowing the navigation solution and EKF to maintain nominal status. A similar case was set up for the event of multiple IMU failures; in this case, three failures occurred at the same time. Like the first failure case, at the fault occurrences, the algorithm isolated the faulty measurements from the fusion process and also maintained nominal filter status. The final case examined the failure of the vital training vector. An underlying fault detection process was run at a slower frequency than the IMU frequency due to calculating feature plane measurement covariances relative to the training vector. In the case of a training vector failure, it was seen that the system believes all sensors to be failing and the training vector to be the only nominal sensor. By implementing Covariance Intersection into the feature plane covariances, a thresholding fault detection scheme was constructed based on the overall fused feature plane covariance. Due to timing considerations of this underlying fault detection method, the system believes that the nominal sensors are failing for only a brief period of time. Once the time history of the feature plane CI solution shows a discrepancy in trend, the training vector is toggled to that of a nominal sensor. During this time of falsely believed off-nominal sensors, the EKF suffers a spike in error but does not diverge which allows the filter to remain operational even in the event of a training vector failure. In all of the cases considered in this paper, it was shown that the faults could be properly detected and isolated, which allowed the navigation solution to produce a covariance that is still consistent with its estimated state and provided for a robust, fault-tolerant method for performing spacecraft navigation.

9.1. FUTURE CONSIDERATIONS

In the current thesis, a complete, robust multi-sensor fault detection navigation network was proposed and outlined. However, there are still a few minor topics of future study. The thresholds that were selected to determine fault characteristics were chosen arbitrarily enabling a threshold regime that remains constant, encompassing all nominal principal components. While this may seem to work during a 500 second simulation, walking biases may eventually cause the transformed data points to obtain similar random walks as well. Due to the linear subspace transformation performed on the sensor outputs, these walking biases do not present themselves directly in the feature plane. To further understand and handle these errors that come through in the feature plane, further studies are needed to propagate a sensor fault threshold forward in time instead of an arbitrary selection. Along with examining thresholding trends, another area of future discussion is that of sensor placement in a system. If a system is constructed where the sensors are decentralized throughout a spacecraft, optimal sensor configuration needs to be examined. It could also be proposed that instead of multiple axes IMUs, such as proposed in this thesis, single axes IMUs would be used and distributed optimally throughout the system. It could then be shown that by minimizing a cost function defined as the trace of the measurement covariance, an optimal sensor configuration could be found.

APPENDIX A
IMU SPECIFICATIONS

	Specification	3DM-GX3-15	M-G362PDC1
Accelerometers	Analog-to-Digital Converter Bitrate	16	16
	Range	$\pm 156.96 \frac{\text{m}}{\text{s}^2}$	$\pm 176.58 \frac{\text{m}}{\text{s}^2}$
	Scale Factor Error	$\mathcal{U}(-500, 500)$ ppm	$\mathcal{U}(-125, 125)$ ppm
	Axes Nonorthogonality Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-108, 108)$ arcsec
	Axes Misalignment Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-108, 108)$ arcsec
	Velocity Random Walk	$0.0007848 \frac{\text{m/s}}{\sqrt{\text{hr}}}$	$0.04 \frac{\text{m/s}}{\sqrt{\text{hr}}}$
	Bias Instability	$0.0003924 \frac{\text{m}}{\text{s}^2}$	$0.000981 \frac{\text{m}}{\text{s}^2}$
	Bias Instability Time	100 s	100 s
	Startup Bias	$\mathcal{U}(-0.0196, 0.0196) \frac{\text{m}}{\text{s}}$	$\mathcal{U}(-0.0785, 0.0785) \frac{\text{m}}{\text{s}}$
Gyroscopes	Analog-to-Digital Converter Bitrate	16	16
	Range	$\pm 600 \frac{\circ}{\text{s}}$	$\pm 300 \frac{\circ}{\text{s}}$
	Scale Factor Error	$\mathcal{U}(-500, 500)$ ppm	$\mathcal{U}(-5000, 5000)$ ppm
	Axes Nonorthogonality Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-360, 360)$ arcsec
	Axes Misalignment Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-360, 360)$ arcsec
	Angular Random Walk	$0.03 \frac{\circ}{\sqrt{\text{hr}}}$	$0.1 \frac{\circ}{\sqrt{\text{hr}}}$
	Bias Instability	$18 \frac{\circ}{\text{hr}}$	$3 \frac{\circ}{\text{hr}}$
	Bias Instability Time	100 s	100 s
	Startup Bias	$\mathcal{U}(-0.25, 0.25) \frac{\circ}{\text{s}}$	$\mathcal{U}(-0.5, 0.5) \frac{\circ}{\text{s}}$

APPENDIX B
FUSION ALGORITHM

Algorithm 1 Robust, Fault Tolerant Multiple IMU Fusion.

1. Draw measured $\mathbf{a}_{1,\dots,N}$ and $\boldsymbol{\omega}_{1,\dots,N}$ from IMU_{1,\dots,N}.
 2. Propagate the individual dead-reckoned solutions, $\mathbf{m}_{1,\dots,N}$ and $\mathbf{P}_{1,\dots,N}$, which correspond to IMU_{1,\dots,N} according to Eqs. (2.13) and (2.14).
 3. Determine $w_{1,\dots,N}$ from Eq. (3.15) for the CI fusion method.
 4. Fuse $\mathbf{m}_{1,\dots,N}$ and $\mathbf{P}_{1,\dots,N}$ using the CI fusion method according to Eq. (4.12) to obtain $\tilde{\mathbf{m}}_{CI}$ and $\tilde{\mathbf{P}}_{CI}$.
 5. Apply the direct average fusion to $\mathbf{a}_{1,\dots,N}$ and $\boldsymbol{\omega}_{1,\dots,N}$ according to Eq. (3.1), to obtain $\tilde{\mathbf{a}}_{DA}$ and $\tilde{\boldsymbol{\omega}}_{DA}$.
 6. Propagate the dead-reckoned solution from $\tilde{\mathbf{a}}_{DA}$ and $\tilde{\boldsymbol{\omega}}_{DA}$ according to equations Eqs. (2.13) and (2.14), to obtain $\tilde{\mathbf{m}}_{DA}$ and $\tilde{\mathbf{P}}_{DA}$.
 7. **if** $\text{tr } \tilde{\mathbf{P}}_{CI} < \text{tr } \tilde{\mathbf{P}}_{DA}$
 - Use $\tilde{\mathbf{m}}_{CI}$ and $\tilde{\mathbf{P}}_{CI}$ as navigation solution.
 - else**
 - Use $\tilde{\mathbf{m}}_{DA}$ and $\tilde{\mathbf{P}}_{DA}$ as navigation solution
 - end**
-
-

APPENDIX C

MATRIX DEFINITIONS

DERIVATIVE OF A TRACE MATRIX

For the derivation of the Kalman gain matrix, it is needed to take the derivative of the trace of a matrix. The derivative properties can be shown as

$$\begin{aligned}\frac{\partial}{\partial \mathbf{A}} \text{trace} \{ \mathbf{BAC} \} &= \mathbf{B}^T \mathbf{C}^T \\ \frac{\partial}{\partial \mathbf{A}} \text{trace} \{ \mathbf{ABA}^T \} &= \mathbf{A} [\mathbf{B} + \mathbf{B}^T].\end{aligned}$$

MATRIX NOTATION

For some vector $\mathbf{v} = [v_x \ v_y \ v_z]^T$, define the matrices $[\mathbf{v}\setminus]$, $[\mathbf{v}\times]$, and $[\mathbf{v}\ast]$ to be

$$[\mathbf{v}\setminus] = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix}, \quad [\mathbf{v}\times] = \begin{bmatrix} 0 & v_z & -v_y \\ -v_z & 0 & v_x \\ v_y & -v_x & 0 \end{bmatrix}, \quad \text{and} \quad [\mathbf{v}\ast] = \begin{bmatrix} 0 & v_z & v_y \\ v_z & 0 & v_x \\ v_y & v_x & 0 \end{bmatrix}.$$

DEAD-RECKONING MATRIX DEFINITIONS

The equations used follow Reference [12]. The following intermediate equations are used in order to calculate the \mathbf{F} and \mathbf{M} matrices. \mathbf{T}_{k-1}^T is the coordinate transformation from the inertial frame to the IMU frame and is calculated using the quaternion portion of the state vector \mathbf{x} .

$$\mathbf{T}_{k-1}^T = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_3q_1 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & 1 - 2q_3^2 - 2q_1^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_3q_1 - q_2q_4) & 2(q_2q_3 + q_1q_4) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

\mathbf{G}_{k-1} represents the Jacobian of the acceleration due to gravity with respect to the vehicle position at t_{k-1} and is expressed for a central body spherical gravity field as

$$\mathbf{G}_{k-1} = \mathbf{G}(\mathbf{s}_{k-1}) = \begin{bmatrix} -\frac{\mu}{|\mathbf{s}_{k-1}|^3} + \frac{3\mu s_x^2}{|\mathbf{s}_{k-1}|^5} & \frac{3\mu s_x s_y}{|\mathbf{s}_{k-1}|^5} & \frac{3\mu s_x s_z}{|\mathbf{s}_{k-1}|^5} \\ \frac{3\mu s_y s_x}{|\mathbf{s}_{k-1}|^5} & -\frac{\mu}{|\mathbf{s}_{k-1}|^3} + \frac{3\mu s_y^2}{|\mathbf{s}_{k-1}|^5} & \frac{3\mu s_y s_z}{|\mathbf{s}_{k-1}|^5} \\ \frac{3\mu s_z s_x}{|\mathbf{s}_{k-1}|^5} & \frac{3\mu s_z s_y}{|\mathbf{s}_{k-1}|^5} & -\frac{\mu}{|\mathbf{s}_{k-1}|^3} + \frac{3\mu s_z^2}{|\mathbf{s}_{k-1}|^5} \end{bmatrix}$$

\mathbf{U}_{k-1} is the Hessian of the acceleration due to gravity and is given by

$$\mathbf{U} = \frac{\mu}{s^7} [(s^2 \mathbf{J}_x - 5x \mathbf{J}) u_x + (s^2 \mathbf{J}_y - 5y \mathbf{J}) u_y + (s^2 \mathbf{J}_z - 5z \mathbf{J}) u_z]$$

where

$$\mathbf{J} = \begin{bmatrix} 3x^2 - s^2 & 3xy & 3xz \\ 3xy & 3y^2 - s^2 & 3yz \\ 3xz & 3yz & 3z^2 - s^2 \end{bmatrix}$$

which can be broken down into the following components:

$$\mathbf{J}_x = \begin{bmatrix} 4x & 3y & 3z \\ 3y & -2x & 0 \\ 3z & 0 & -2x \end{bmatrix}, \quad \mathbf{J}_y = \begin{bmatrix} -2y & 3x & 0 \\ 3x & 4y & 3z \\ 0 & 3z & -2y \end{bmatrix}, \quad \mathbf{J}_z = \begin{bmatrix} -2z & 0 & 3x \\ 0 & -2z & 3y \\ 3x & 3y & 4z \end{bmatrix}.$$

Now, define n as $\boldsymbol{\omega}_{m,k} \Delta t_k$ and if $\|n\| > 0$, $n = n/\|n\|$, $\mathbf{T}(\boldsymbol{\omega}_{m,k} \Delta t_k)$ is then defined to be

$$\mathbf{T}(\boldsymbol{\omega}_{m,k} \Delta t_k) = \mathbf{I}_{3 \times 3} - \sin \|\boldsymbol{\omega}_{m,k} \Delta t_k\| [\mathbf{n} \times] + (1 - \cos \|\boldsymbol{\omega}_{m,k} \Delta t_k\|) [\mathbf{n} \times][\mathbf{n} \times].$$

THE F MATRIX

The \mathbf{F} matrix at time t_{k-1} is defined as

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & \mathbf{F}_{r\theta} & \mathbf{F}_{rd} & \mathbf{F}_{rb_a} & \mathbf{F}_{rm_a} & \mathbf{F}_{rn_a} & \mathbf{F}_{rs_a} & \mathbf{F}_{rb_g} & \mathbf{F}_{rm_g} & \mathbf{F}_{rn_g} & \mathbf{F}_{rs_g} \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & \mathbf{F}_{v\theta} & \mathbf{F}_{vd} & \mathbf{F}_{vb_a} & \mathbf{F}_{vm_a} & \mathbf{F}_{vn_a} & \mathbf{F}_{vs_a} & \mathbf{F}_{vb_g} & \mathbf{F}_{vm_g} & \mathbf{F}_{vn_g} & \mathbf{F}_{vs_g} \\ 0 & 0 & \mathbf{F}_{\theta\theta} & 0 & 0 & 0 & 0 & 0 & \mathbf{F}_{\theta b_g} & \mathbf{F}_{\theta m_g} & \mathbf{F}_{\theta n_g} & \mathbf{F}_{\theta s_g} \\ 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I}_{3 \times 3} \end{bmatrix}$$

with the following entries:

$$\mathbf{F}_{rr} = \mathbf{I}_{3 \times 3} + \frac{1}{2} \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \Delta t_k^2$$

$$\mathbf{F}_{rv} = \mathbf{I}_{3 \times 3} \Delta t_k$$

$$\begin{aligned} \mathbf{F}_{r\theta} = & -\frac{1}{2} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \frac{1}{3} \hat{\mathbf{T}}_{k-1}^T \left[(\Delta \hat{\boldsymbol{\theta}}_{m,k} \times \Delta \hat{\mathbf{v}}_{m,k}) \times \right] \right) \Delta t_k \\ & - \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] - \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T \left[(\hat{\mathbf{d}} \times \Delta \hat{\boldsymbol{\theta}}_{m,k}) \times \right] \right] \Delta t_k^2 \end{aligned}$$

$$\mathbf{F}_{rd} = \frac{1}{2} \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{3} \hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T + \frac{1}{3} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T \left[\Delta \hat{\boldsymbol{\theta}}_{m,k} \times \right] \right] \Delta t_k^2$$

$$\mathbf{F}_{rb_a} = -\hat{\mathbf{R}}_a \Delta t$$

$$\mathbf{F}_{rm_a} = \hat{\mathbf{R}}_a [\Delta \hat{\mathbf{v}}_{m,k} \times]$$

$$\mathbf{F}_{rn_a} = -\hat{\mathbf{R}}_a [\Delta \hat{\mathbf{v}}_{m,k} *]$$

$$\mathbf{F}_{rsa} = -\hat{\mathbf{R}}_a[\Delta\hat{\mathbf{v}}_{m,k}\searrow]$$

$$\mathbf{F}_{rmg} = \hat{\mathbf{R}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}\times]$$

$$\mathbf{F}_{rng} = -\hat{\mathbf{R}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}*]$$

$$\mathbf{F}_{rsg} = -\hat{\mathbf{R}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}\searrow]$$

$$\mathbf{F}_{vr} = \left(\hat{\mathbf{G}}_{k-1} - \frac{1}{2}\hat{\mathbf{U}}_{k-1} \right) \Delta t_k$$

$$\mathbf{F}_{vv} = \mathbf{I}_{3\times 3}$$

$$\begin{aligned} \mathbf{F}_{v,\theta} = & - \left(\hat{\mathbf{T}}_{k-1}^T [\Delta\hat{\mathbf{v}}_{m,k}\times] + \frac{1}{2}\hat{\mathbf{T}}_{k-1}^T \left[(\Delta\hat{\boldsymbol{\theta}}_{m,k}\times\Delta\hat{\mathbf{v}}_{m,k})\times \right] \right) \\ & - \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{2}\hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}}\times] - \frac{1}{2}\hat{\mathbf{G}}_{k-1}\hat{\mathbf{T}}_{k-1}^T \left[(\hat{\mathbf{d}}\times\Delta\hat{\boldsymbol{\theta}}_{m,k})\times \right] \right] \Delta t_k \end{aligned}$$

$$\mathbf{F}_{vd} = \left[\left(\hat{\mathbf{G}}_{k-1} - \frac{1}{2}\hat{\mathbf{U}}_{k-1} \right) \hat{\mathbf{T}}_{k-1}^T + \frac{1}{2}\hat{\mathbf{G}}_{k-1}\hat{\mathbf{T}}_{k-1}^T \left[\Delta\hat{\boldsymbol{\theta}}_{m,k}\times \right] \right] \Delta t_k$$

$$\mathbf{F}_{vba} = -\hat{\mathbf{V}}_a\Delta t$$

$$\mathbf{F}_{vma} = \hat{\mathbf{V}}_a[\Delta\hat{\mathbf{v}}_{m,k}\times]$$

$$\mathbf{F}_{vna} = -\hat{\mathbf{V}}_a[\Delta\hat{\mathbf{v}}_{m,k}*]$$

$$\mathbf{F}_{vsa} = -\hat{\mathbf{V}}_a[\Delta\hat{\mathbf{v}}_{m,k}\searrow]$$

$$\mathbf{F}_{vbg} = \hat{\mathbf{V}}_g\Delta t$$

$$\mathbf{F}_{vmg} = \hat{\mathbf{V}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}\times]$$

$$\mathbf{F}_{vng} = -\hat{\mathbf{V}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}*]$$

$$\mathbf{F}_{vsg} = -\hat{\mathbf{V}}_g[\Delta\hat{\boldsymbol{\theta}}_{m,k}\searrow]$$

$$\mathbf{F}_{\theta\theta} = \mathbf{T}(\Delta\hat{\boldsymbol{\theta}}_{m,k})$$

$$\mathbf{F}_{\theta bg} = -\mathbf{I}_{3\times 3}\Delta t$$

$$\mathbf{F}_{\theta mg} = [\Delta\hat{\boldsymbol{\theta}}_{m,k}\times]$$

$$\mathbf{F}_{\theta ng} = -[\Delta\hat{\boldsymbol{\theta}}_{m,k}*]$$

$$\mathbf{F}_{\theta sg} = -[\Delta\hat{\boldsymbol{\theta}}_{m,k}\searrow]$$

where $\Delta \hat{\mathbf{v}}_{m,k} = \mathbf{a}_{m,k} \Delta t$ and $\Delta \hat{\boldsymbol{\theta}}_{m,k} = \boldsymbol{\omega}_{m,k} \Delta t$.

THE M MATRIX

The \mathbf{M} matrix at time t_{k-1} is defined as

$$\mathbf{M}_{k-1} = \begin{bmatrix} -\hat{\mathbf{R}}_a \Delta t & \hat{\mathbf{R}}_g \Delta t & 0 & 0 \\ -\hat{\mathbf{V}}_a \Delta t & \hat{\mathbf{V}}_g \Delta t & 0 & 0 \\ 0 & -\mathbf{I}_{3 \times 3} \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I}_{3 \times 3} \Delta t & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_{3 \times 3} \Delta t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} \hat{\mathbf{R}}_a &= \frac{1}{2} \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right) \Delta t_k \\ \hat{\mathbf{R}}_g &= \frac{1}{6} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta t_k \right) \Delta t_k \\ \hat{\mathbf{V}}_a &= \hat{\mathbf{T}}_{k-1}^T \left(\mathbf{I}_{3 \times 3} + \frac{1}{2} [\Delta \hat{\boldsymbol{\theta}}_{m,k} \times] \right) \\ \hat{\mathbf{V}}_g &= \frac{1}{2} \left(\hat{\mathbf{T}}_{k-1}^T [\Delta \hat{\mathbf{v}}_{m,k} \times] + \hat{\mathbf{G}}_{k-1} \hat{\mathbf{T}}_{k-1}^T [\hat{\mathbf{d}} \times] \Delta t_k \right). \end{aligned}$$

APPENDIX D
MONTE CARLO ANALYSIS

MONTE CARLO ALGORITHM

First, the “average” quaternion, labeled $\bar{\mathbf{q}}^*$, is found according to Reference [23] as the eigenvector corresponding to the largest eigenvalue of the matrix \mathbf{M} , where

$$\mathbf{M} = \sum_{i=1}^N \bar{\mathbf{q}}_i \bar{\mathbf{q}}_i^T$$

and N is the number of Monte Carlo Samples. The error quaternion is then found for each state vector in the cell \mathbf{X} as

$$\delta \bar{\mathbf{q}}_i = \bar{\mathbf{q}}^* \otimes \bar{\mathbf{q}}_i^{-1}$$

for the i th state vector in N . Next, a cell of augmented vectors corresponding to the cell of state vectors, labeled \mathbf{Y} , is defined in terms of a rotation vector (as opposed to an error quaternion) as

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{r}_i & \mathbf{v}_i & 2\delta \mathbf{q}_i & \mathbf{d} & \mathbf{a}_{param} & \boldsymbol{\omega}_{param} \end{bmatrix}^T$$

where

$$\delta \bar{\mathbf{q}}_i \approx \begin{bmatrix} \delta \mathbf{q}_i \\ 1 \end{bmatrix}$$

for small error angles. The mean is then

$$\mathbf{Y}^* = \frac{1}{N} \sum_{i=1}^N \mathbf{Y}_i$$

and the covariance is

$$\mathbf{P}_{mc} = \frac{1}{N} \sum_{i=1}^N (\mathbf{Y}_i - \mathbf{Y}^*) (\mathbf{Y}_i - \mathbf{Y}^*)^T$$

This augmented state vector, which is expressed in terms of a rotation vector for the vehicle attitude, is necessary to compute the mean and covariance because the error in the rotation vector representation of the vehicle attitude is approximately additive under the assumption of small angles. Note that the error quaternion representation is multiplicative.

APPENDIX E

EKF CONSIDERATION: UNDERWEIGHTING

In the construction and simulation of the EKF, it is sometimes needed to “soften” measurement updates to avoid over-convergence in the filter. The accelerometer biases were seen to over-converge in the current simulation. To avoid this, the position measurements were underweighted. Recall that the update equations used to incorporate measurements in the EKF are given by

$$\begin{aligned}\mathbf{m}_k^+ &= \mathbf{m}_k^- + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k] \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{C}_k \mathbf{K}_k^T - \mathbf{K}_k \mathbf{C}_k^T + \mathbf{K} \mathbf{W}_k \mathbf{K}_k^T\end{aligned}$$

where the Kalman gain, \mathbf{K}_k , is defined as

$$\begin{aligned}\mathbf{K}_k &= \mathbf{C}_k \mathbf{W}_k^{-1} \\ &= \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) [\mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T]^{-1}.\end{aligned}$$

This optimal gain can be applied to the state estimation error covariance which allows the covariance update equation to be rewritten as

$$\begin{aligned}\mathbf{P}_k^+ &= \mathbf{P}_k^- \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^T \\ &= \mathbf{P}_k^- - \mathbf{K}_k [\mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T + \mathbf{I}] \mathbf{K}_k^T.\end{aligned}$$

It is seen from this rewritten covariance update that by lowering the Kalman gain, a smaller update occurs in both the state estimate and the state estimation error covariance. Using this information, the covariance update can be reduced in order to slow down the convergence of the filter, preventing over convergence of the filter. The Kalman gain is then augmented with an underweighting factor \mathbf{U}_k as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) [\mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T + \mathbf{U}_k]^{-1}.$$

While this underweighting factor can be determined by computing second-order effects in the residual covariance, an ad-hoc selection can be made as a tuning parameter. This ad-hoc selection of the underweighting factor is given as a scaled form of the uncertainty mapped into the measurement uncertainty, such that

$$\mathbf{U}_k = \frac{1-p}{p} \mathbf{H}_k(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}_k^T(\mathbf{m}_k^-) \quad (9.1)$$

where $0 < p < 1$. Knowing that the innovations covariance is defined as

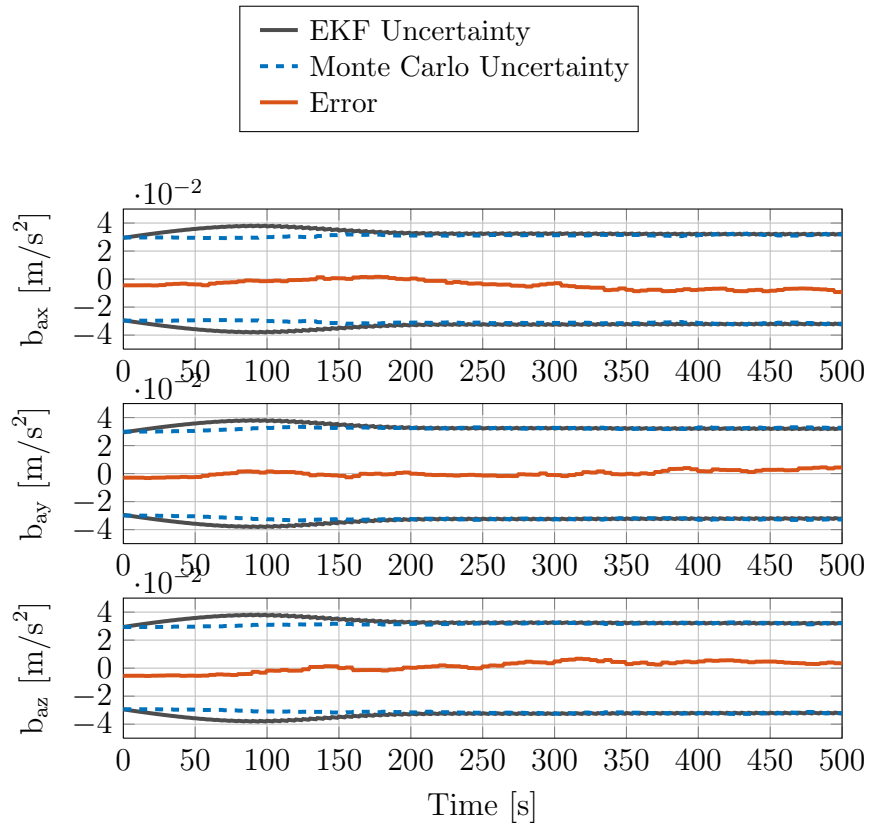
$$\mathbf{W}_k = \mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T,$$

and using the ad-hoc relationship in Eq. (9.1), the final form of the underweighting consideration is given in terms of the innovations covariance as

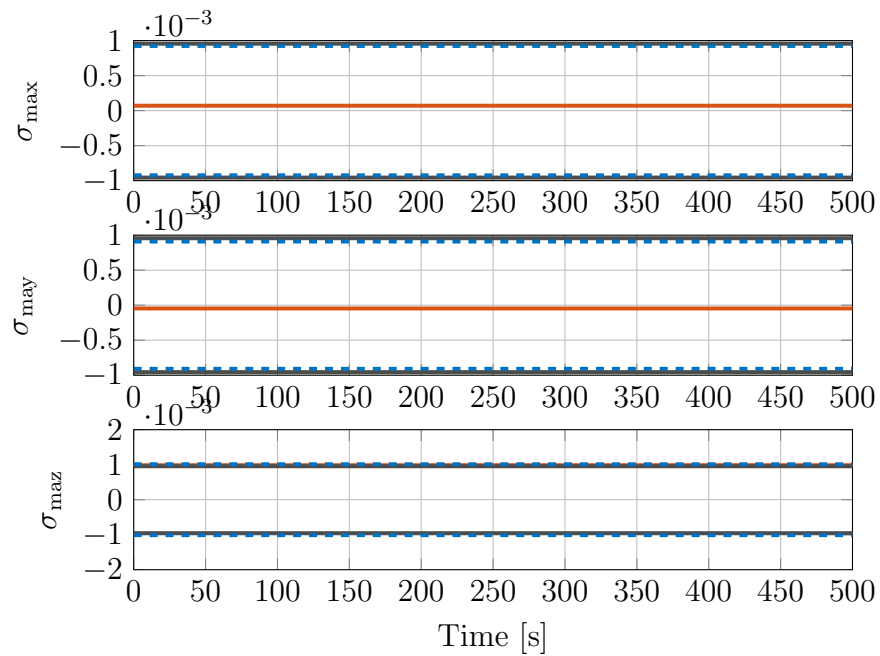
$$\mathbf{W}_k = \frac{1-p}{p} \mathbf{H}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}^T(\mathbf{m}_k^-) + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T. \quad (9.2)$$

APPENDIX F

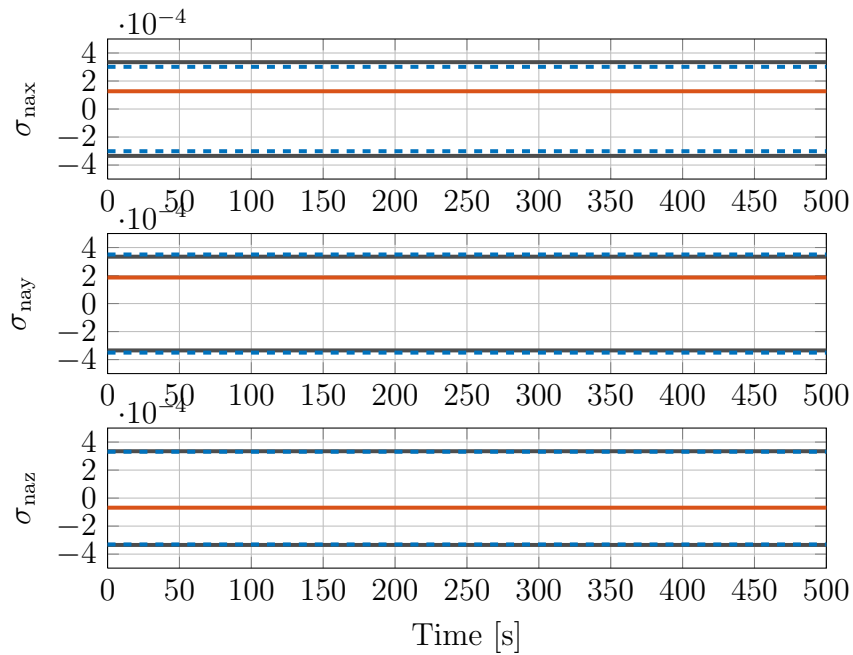
ERROR PARAMETER SIMULATION RESULTS



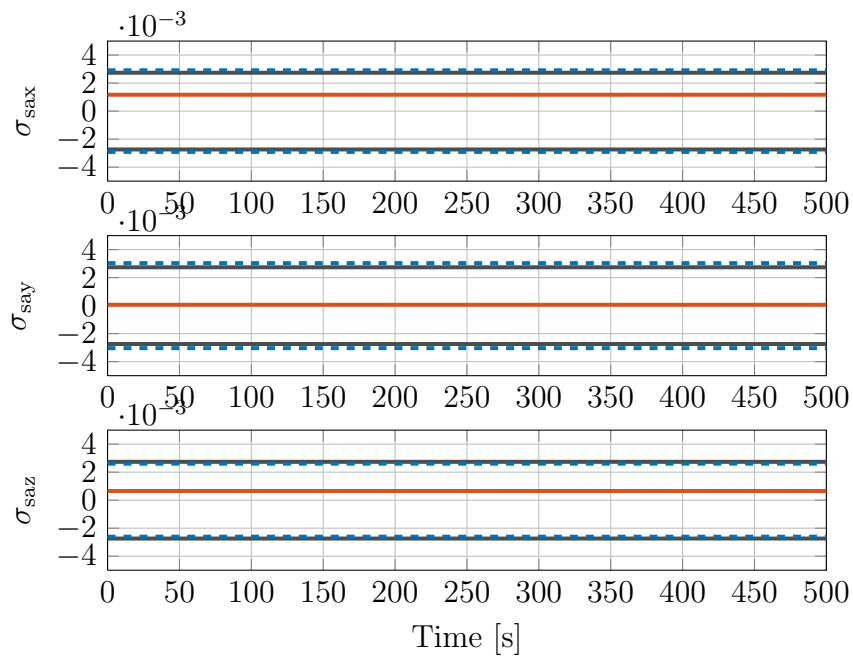
Case 1: Accelerometer Bias Standard Deviation/Errors vs. Time



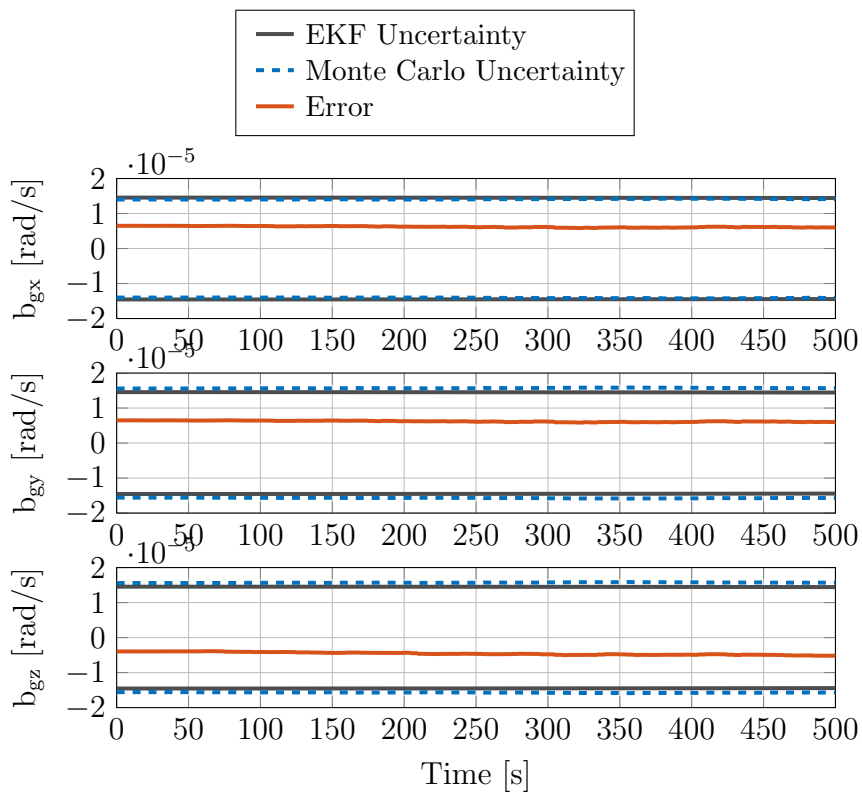
Case 1: Accelerometer Misalignment Standard Deviation/Errors vs. Time



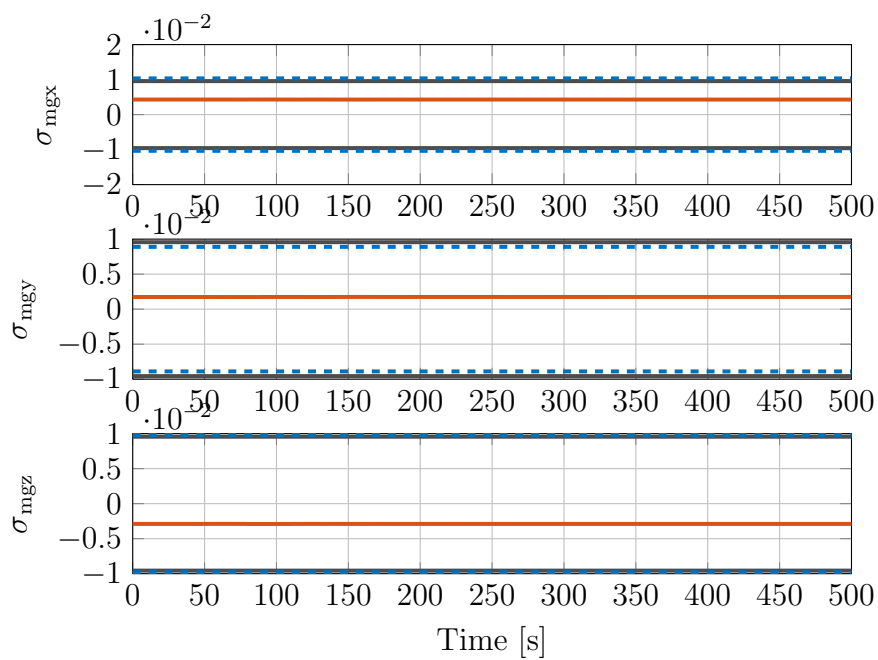
Case 1: Accelerometer Nonorthogonality Standard Deviation/Errors vs. Time



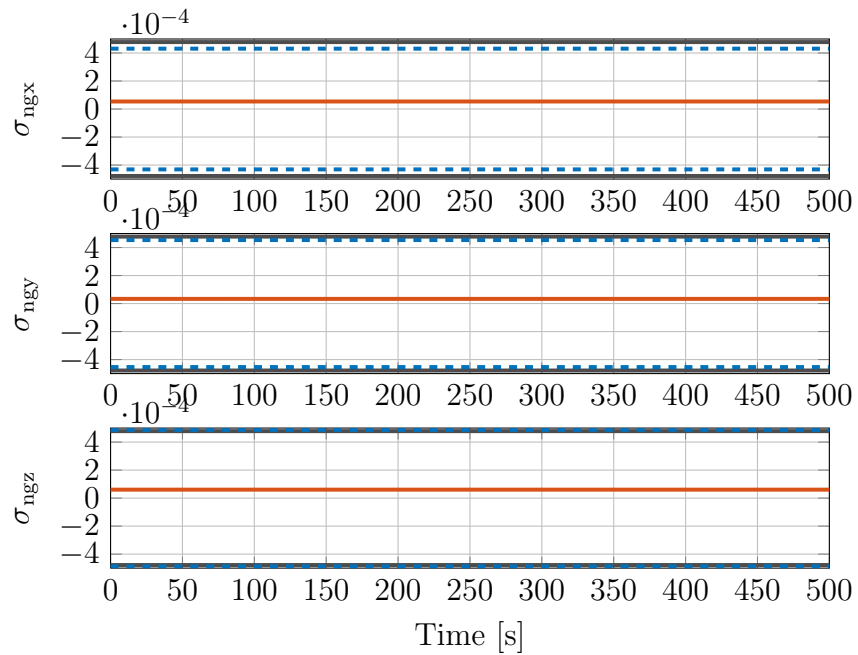
Case 1: Accelerometer Scale Factor Standard Deviation/Errors vs. Time



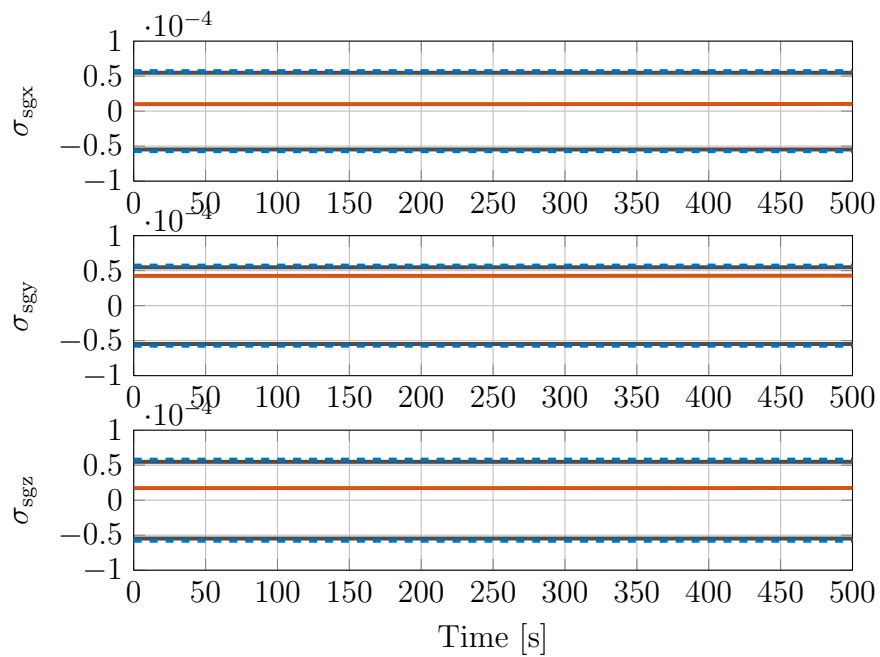
Case 1: Gyroscope Bias Standard Deviation/Errors vs. Time



Case 1: Gyroscope Misalignment Standard Deviation/Errors vs. Time



Case 1: Gyroscope Nonorthogonality Standard Deviation/Errors vs. Time



Case 1: Gyroscope Scale Factor Standard Deviation/Errors vs. Time

BIBLIOGRAPHY

- [1] Daniel Greenheck, Robert Bishop, Eric Jonardi, and John Christian. Design and testing of a low-cost MEMS IMU cluster for SmallSat applications. In *Proceedings of the 2013 Small Satellite Conference*, 2013.
- [2] Jared B. Bancroft and Gerard Lachapelle. Data fusion algorithms for multiple inertial measurement units. *ISSN 1424-8220*, June 2011.
- [3] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, New York, 2nd edition, 2011.
- [4] Tim Bailey, Simon Julier, and Gabriel Agamennoni. On conservative fusion of information with unknown non-gaussian dependence. In *15th International Conference on Information Fusion (FUSION)*, pages 1876–1883. IEEE, July 9–12 2012.
- [5] Kyle J. DeMars and James S. McCabe. Multi-sensor data fusion in non-gaussian orbit determination. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Advances in the Astronautical Sciences, 2014.
- [6] Karl Pearson. On lines and planes closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.
- [7] J. E. Potter and M. C. Suman. Thresholdless redundancy management with arrays of skewed instruments. In *Integrity in Electronic Flight Control Systems*, pages 15–25, 1997.
- [8] Lee Wonhee and Chan Gookpark. A fault detection method of redundant imu using modified principal component analysis. *International Journal of Aeronautical and Space Science*, 13(3):298–404, 2012.
- [9] Samuel J. Haberberger and Kyle J. DeMars. Spacecraft navigation using a robust multi-sensor fault detection scheme. In *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, 2016.
- [10] Kenneth R. Britting. *Inertial Navigation Systems Analysis*. Wiley Interscience, New York, 1st edition, 1971.
- [11] Renato Zanetti. *Advanced Navigation Algorithms for Precision Landing*. PhD thesis, The University of Texas at Austin, Austin, TX, December 2007.
- [12] DeMars, Kyle J. Precision navigation for lunar descent and landing. Master’s thesis, The University of Texas at Austin, Austin, TX, May 2007.
- [13] Woodman, Oliver J. *An Introduction to Inertial Navigation*. Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007.

- [14] Jason L. Williams and Peter S. Maybeck. Cost-function-based Gaussian mixture reduction for target tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, pages 1047–1054, 2003.
- [15] S. S. Wilks. Certain generalizations in the analysis of variance. *Biometrika*, 24 (3/4):471 – 594, November 1932.
- [16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, New York City, NY, 3rd edition, 2006.
- [17] Walter Gander. Algorithms for the qr-decomposition. In *Research Report No. 80-02*, 1980.
- [18] Jacob E. Darling Samuel J. Haberberger and Kyle J. DeMars. Distributed network navigation using multi-sensor data fusion. In *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, pages 3427–3444, 2015.
- [19] Duchi J. Derivations of linear algebra and optimization.
- [20] Malcolm D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993.
- [21] A. Gelb. *Applied Optimal Estimation*. M.I.T. Press, 1974. ISBN 9780262700085.
- [22] Duk-Sun Shim and Cheol-Kwang. Yang. Optimal configuration of redundant inertial sensors for navigation and fdi performance. *Sensors ISSN*, (1424-8220), July 2010.
- [23] Crassidis J . Oshman Y. Markley F., Cheng Y. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197.

VITA

Samuel Haberberger was born in St. Louis, Missouri on November 22, 1991. He began his aerospace engineering studies at Missouri University of Science and Technology in the fall of 2010. As an undergraduate student, Samuel began doing research with Dr. Kyle DeMars on IMU-based navigation and multi sensor data fusion systems. Samuel graduated with his Bachelor of Science degree in the spring of 2014. The proceeding Summer, Samuel entered and started his graduate aerospace engineering program at Missouri University of Science and Technology to study fault detection methods in multi-sensor networks applied to IMU-based navigation. As a graduate student, Samuel was the recipient of a NASA Missouri Space Grant Consortium Fellowship and worked as a research assistant under Dr. Kyle DeMars. He received his MS degree in Aerospace Engineering in May of 2016 from Missouri University of Science and Technology.