Integrity Monitoring for Multiple Errors in Vision Navigation Systems

Nicholas Allen Baine
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Engineering Commons

Repository Citation
Browse all Theses and Dissertations. 729.
https://corescholar.libraries.wright.edu/etd_all/729

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-core scholar@wright.edu.
Integrity Monitoring for Multiple Errors in Vision Navigation Systems

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

Nicholas A. Baine
B.S., Wright State University, 2007
M.S., Wright State University, 2012

2013
Wright State University

May 17, 2013

Kuldip Rattan, Ph.D.
Dissertation Director

Ramana Grandhi, Ph.D.
Director, Ph.D. in Engineering Program

R. William Ayres, Ph.D.
Interim Dean, School of Graduate Studies

Committee on Final Examination

Kuldip S. Rattan, Ph.D.

John Gallagher, Ph.D.

Xiaodong Zhang, Ph.D.

Jacob Campbell, Ph.D.

Meir Pachter, Ph.D.
ABSTRACT


In aviation applications, navigation integrity is paramount. Integrity of GPS systems is well established with set standards. Vision-based navigation systems have been found to be an adequate substitute for GPS when it is unavailable but are unlikely to be utilized until there is a measure for system integrity. Work has been done to detect the effect of a single measurement pair being corrupted with a bias; however, the measurement geometry varies greatly with the environment. The environment could be sparse in visual features to track, or the environment could be rich with features. With more features, there is a greater probability of having multiple corrupted measurements. It is essential that multiple corrupt measurements are detected and excluded to assure the integrity and reliability of the system. In addition, misalignment errors in the camera system result in systematic errors that are undetectable by current methods. This dissertation focuses on understanding the existing integrity monitoring methods and using them for the detection of multiple errors in vision-based navigation systems, as well as, developing a technique for detecting systematic errors due to camera misalignment and scaling. These methods are developed analytically and verified by using simulations. These simulations serve to demonstrate the usefulness of these methods in achieving the goal of this research to further the area of integrity monitoring for vision systems, so it could eventually be used as a trusted system and as a back-up for GPS navigation.
# Contents

1 Introduction ........................................... 1
   1.1 Motivation ........................................... 1
   1.2 Contributions ...................................... 5
   1.3 Outline .............................................. 7
   1.4 Mathematical Notation ............................... 7

2 Background ............................................ 9
   2.1 Random Variables .................................... 9
      2.1.1 Normal Distribution ............................ 9
      2.1.2 Chi-Squared Distribution ..................... 10
      2.1.3 Non-Central Chi-Squared Distribution ...... 11
      2.1.4 Chi and Non-Central Chi Distributions ..... 13
      2.1.5 Binomial Distribution ........................ 18
      2.1.6 Hypergeometric Distribution ................. 18
   2.2 Least Squares ....................................... 20
   2.3 Kalman Filtering .................................... 25
   2.4 Extended Kalman Filter ............................ 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Observability of Kalman Filter</td>
<td>29</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Navigation Frames</td>
<td>30</td>
</tr>
<tr>
<td>2.6</td>
<td>Attitude/Orientation Representation</td>
<td>31</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Euler Angles</td>
<td>31</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Direction Cosine Matrix $C_{n}^{b}$</td>
<td>34</td>
</tr>
<tr>
<td>2.6.3</td>
<td>$C_{b}^{n}$ to Euler angle conversion</td>
<td>35</td>
</tr>
<tr>
<td>2.6.4</td>
<td>$C_{b}^{n}$ Integration</td>
<td>37</td>
</tr>
<tr>
<td>2.6.5</td>
<td>Inertial Navigation Equations</td>
<td>39</td>
</tr>
<tr>
<td>2.7</td>
<td>INS Aiding</td>
<td>40</td>
</tr>
<tr>
<td>2.8</td>
<td>Global Positioning System</td>
<td>42</td>
</tr>
<tr>
<td>2.8.1</td>
<td>Pseudorange Measurements</td>
<td>43</td>
</tr>
<tr>
<td>2.8.2</td>
<td>Dilution of Precision</td>
<td>45</td>
</tr>
<tr>
<td>2.9</td>
<td>Vision Based Navigation</td>
<td>48</td>
</tr>
<tr>
<td>2.9.1</td>
<td>Projection Model</td>
<td>49</td>
</tr>
<tr>
<td>2.9.2</td>
<td>Measurement Model</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>GPS Integrity Monitoring</td>
<td>56</td>
</tr>
<tr>
<td>3.1</td>
<td>Receiver Autonomous Integrity Monitoring</td>
<td>56</td>
</tr>
<tr>
<td>3.2</td>
<td>Parity Vector</td>
<td>57</td>
</tr>
<tr>
<td>3.3</td>
<td>Least Square Residuals</td>
<td>61</td>
</tr>
<tr>
<td>3.4</td>
<td>Slope Method</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>Current Vision Integrity Monitoring</td>
<td>66</td>
</tr>
<tr>
<td>4.1</td>
<td>Parity Vector and Slope</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>Detection and Isolation Multiple Random Errors in Vision Systems</td>
<td>70</td>
</tr>
<tr>
<td>5.1</td>
<td>Test Statistic</td>
<td>70</td>
</tr>
</tbody>
</table>
5.2 Bayes Algorithm for Isolating Corrupt Measurements ........................................ 71
5.3 Results .................................................................................................................. 73

6 Dilution of Precision for Vision Systems ................................................................. 83
   6.1 Derivation of Dilution of Precision for Vision Systems ...................................... 83
   6.2 Effect of DOP on Integrity Monitoring Performance ........................................... 86
   6.3 Numeric Examples ............................................................................................. 91
   6.4 Summary ............................................................................................................ 96

7 Detection of Systematic Errors in Vision Systems .................................................. 99
   7.1 Test Statistic Development .............................................................................. 99
   7.2 Relationship between Test Statistic and Errors ............................................... 103
      7.2.1 X-Axis Misalignment Error (Pan) .............................................................. 109
      7.2.2 Y-Axis Misalignment Error (Tilt) .............................................................. 113
      7.2.3 Z-Axis Misalignment Error (Rotation) ..................................................... 118
      7.2.4 Scaling Error (Focus/Zoom) .................................................................... 122

8 Conclusion ............................................................................................................. 126
   8.1 Discussion ......................................................................................................... 126
   8.2 Future Work .................................................................................................... 127
   8.3 Summary ......................................................................................................... 129
List of Figures

1.1 Graphical depiction of horizontal alert level (HAL) and vertical alert level (VAL). 3

2.1 Plot of the probability distribution function (pdf) of a normal distribution with standard deviation $\sigma = 1, 2, 3, 4$. 10

2.2 Plot of the probability distribution function (pdf) of a $\chi^2$ distribution with degrees of freedom $dof = 2, 4, 6, 8$. 12

2.3 Plot of the probability distribution function (pdf) of a non-central $\chi^2$ distribution with ten degrees of freedom ($dof = 10$) and non-centrality parameter $\lambda = 2, 4, 6, 8$. 14

2.4 Plot of the probability distribution function (pdf) of a non-central $\chi$ distribution with ten degrees of freedom ($dof = 10$) and non-centrality parameter $\lambda = 2, 4, 6, 8$. 16

2.5 Plot of the probability distribution function (pdf) of a $\chi$ distribution with ten degrees of freedom, $d.o.f. = 2, 4, 6, 8$. 17

2.6 Plot of the probability mass function (pmf) of a binomial distribution with 50 degrees-of-freedom ($k = 50$) and probability of positive $p = 0.1, 0.2, 0.4, 0.8$. 19

2.7 Plot of the probability mass function (pmf) of a hypergeometric distribution with population size, $N = 100$, varying positive result population $M = 10, 20, 30, 40$, and varying number of samples taken $k = 50$. 21
2.8 Plot of the Probability Mass Function (pmf) of a Hypergeometric Distribution with population size, \( N = 100 \), positive result population \( M = 30 \), and varying number of samples taken \( k = 10, 20, 30, 40 \). ........................................ 22

2.9 Nomenclature and the Steps of the Kalman Filter. ........................................ 26

2.10 Heading rotation following right hand rule. ........................................ 32

2.11 Pitch rotation following right hand rule. ........................................ 33

2.12 Roll rotation following right hand rule. ........................................ 34

2.13 Layouts of different types of GPS aided INS: (a) Uncoupled Integration, (b) Loosely Coupled Integration, and (c) Tightly Coupled Integration. .............................. 41

2.14 Thin lens camera model. ........................................ 50

2.15 Pinhole Camera Model. ........................................ 50

2.16 Camera projection model. ........................................ 51

2.17 Target to image transformation geometry. ........................................ 53

4.1 Chart showing the areas of Vision Integrity Monitoring. ................................. 66

5.1 Probability of getting a good subset of measurements assuming that there is a total of 100 measurements and 5 measurements in a given subset. ......................... 74

5.2 Average sum of the error probability vector \( \vec{P} \) vs. test iteration, given a 50 run Monty Carlo experiment with 5 of 100 measurements bad. ........................................ 75

5.3 Average sum of the error probability vector \( \vec{P} \) vs. test iteration, given a 50 run Monty Carlo experiment with 10 of 100 measurements bad. ........................................ 76

5.4 Average sum of the error probability vector \( \vec{P} \) vs. test iteration, given a 50 run Monty Carlo experiment with 15 of 100 measurements bad. ........................................ 77

5.5 Steady-state average sum of the error probability vector \( \vec{P} \) vs. the number of bad measurements out of 100 total measurements. ........................................ 78
5.6 Steady-state average sum of the error probability vector $\bar{P}$ vs. the probability of getting a good subset of data with 100 total measurements and a varying number of bad measurements. .............................................................. 79

5.7 Predicted and simulated steady-state average sum of the error probability vector $\bar{P}$ vs. the number of bad measurements out of 100 total measurements. ...................... 82

6.1 Example of Good Measurement Geometry (1000m): Measurements in the camera frame with pixel coordinates. ................................................................. 91

6.2 Example of Good Measurement Geometry (1000m): Target features (blue) and vehicle location (red) in local navigation frame. ................................. 92

6.3 Case 1: Good measurement geometry with a low distance from the targets (1000m). 93

6.4 Case 2: Good measurement geometry with a larger distance from the targets (5000m). 94

6.5 Case 3: Bad measurement geometry with a larger distance from the targets (5000m). 95

6.6 Detector operating characteristics (DOC) for good geometry at 1000m. ............ 96

6.7 Detector operating characteristics (DOC) for good geometry at 5000m. ............ 97

6.8 Detector operating characteristics (DOC) for bad geometry at 5000m. .............. 97

7.1 Target to image transformation geometry with two camera frames. .................. 100

7.2 Sensor platform and feature locations. The platform location is red and the features for tracking are blue. ................................................................. 103

7.3 Image projection in pixel frame of tracked targets as seen by the camera on the sensor platform. ................................................................. 104

7.4 Image projection of features given that an unwanted bias is introduced (red) and the correct projection (blue). ................................................................. 105

7.5 Distribution of the Sum of Squared Differences, $SSD$, given a 1000 run Monte Carlo experiment with (red) and without (blue) a systematic error. ............ 106
7.6 Plot comparing the SSD ($\chi^2$ test statistic) and a roll (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.

7.7 Comparison of $\beta$ ($\chi$ test statistic) and a roll (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.

7.8 Example of feature locations on an image with a pan (x-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.

7.9 Comparison of the test statistic and a pan (x-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the angle error.

7.10 Geometry for relating the horizontal view angle, focal length, and width of the image plane.

7.11 Example of feature locations on an image with a tilt (y-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.

7.12 Comparison of the test statistic and a tilt (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.

7.13 Geometry for relating the vertical view angle, focal length, and height of the image plane.

7.14 Example of feature locations on an image with a rotation (z-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.
7.15 Comparison of the test statistic and a rotation (z-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.

7.16 Geometry for determining the displacement of a feature location resulting from image rotation.

7.17 Example of feature locations on an image with a scaling error. Blue markers represent the correct location and red markers represent the measured location.

7.18 Comparison of the test statistic and a scaling error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Scaling Error.
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Important WGS-84 Parameters</td>
<td>31</td>
</tr>
<tr>
<td>6.1</td>
<td>DOP for example cases</td>
<td>95</td>
</tr>
</tbody>
</table>
Acknowledgment

I would like to take this opportunity to extend my thanks to Dr. Jacob Campbell and the rest of the Sensors Directorate of the Air Force Research Laboratory who have helped me in my research. I would also like to thank DAGSI and AFRL for funding my research, and my advisor Dr. Kuldip Rattan and my other committee members, Dr. John Gallagher, Dr. Xiaodong Zhang, and Dr. Meir Pachter for their guidance and support.
Dedicated to

My fiancé, Elaine Knepp, and my family

for giving me the support and encouragement to chase my dreams.
Introduction

1.1 Motivation

Modern aerospace operations require accurate navigation systems. These operations include the basic need of civilian aircraft for approach, departure, and en route navigation. Additional military operations requiring navigation include precision bombing, aerial refueling, formation flying, and unmanned missions. Remarkable levels of precision and accuracy, for both civilian and military navigation users, resulted from the creation and development of the Global Positioning System (GPS). With the installation of additional Global Navigation Satellite Systems (GNSS), the performance level should continue to improve; however, the accuracy achievable by this—or any other system—is dependent on its ability to estimate the true position, given current operating conditions and the quality of sensor measurements. Unfortunately, modeling errors, environmental factors, and equipment limitations prevent a perfect estimation of the true position, regardless of the system.

An Inertial Navigation System (INS) uses specific forces and angular rate measurements to estimate the position, velocity, and orientation of a vehicle. The measurements for an INS are produced by an Inertial Measurement Unit (IMU), which consists of accelerometers and gyroscopes. This type of navigation has the benefit of being independent of external signals, making it
impossible to jam or spoof; consequently, it is a very useful system for military applications and for critical civilian systems. Because this system is very dependent on the quality of sensors, it is influenced by sensor misalignments, drifts, and biases resulting from limitations in the materials and design of sensors. In addition, the specific forces measured by the accelerometers include gravity and vehicle accelerations, thus a gravity model is needed. Errors in the gravity model used to convert the specific forces into accelerations also result in system errors. An INS works by integrating measurements of acceleration over time to determine position. This process causes errors in the measurements to accumulate and grow over time, resulting in system accuracy that correspondingly degrades with time. Because of this limitation, a stand-alone INS is often used for short-term navigation. For long-term navigation, an INS is coupled with another navigation system such as GPS. By integrating two systems, the INS is able to operate with its error bounded, and the combined result is better than either system can produce operating on its own. This still leaves the navigation system vulnerable because of the dependence upon external signals employed by GPS, because external radio-frequency (RF) signals can be either jammed or spoofed, rendering the system inoperative or resulting in false navigation solutions.

This vulnerability caused by the use of external signals has led to research investigating alternatives. One alternative actively being explored to remove the dependence upon GNSS is the fusion of an INS and a vision system [10, 11, 15, 20, 25, 37, 38, 44, 47, 54, 57–62]. Such a system is not reliant upon external signals and is completely passive, emitting no trackable signals. This feature is beneficial for stealth applications and eliminates concerns regarding spectrum allocation if such a system is used on a large scale. Such a system can also be relatively small and inexpensive, requiring little power to operate relative to its GPS counterparts [44].

Earlier research has shown the feasibility of a vision-aided navigation system in many different configurations. These proposed systems are shown to provide reasonable accuracy on the
level needed for aviation applications. The accuracy available is dependent on the camera system used, the INS measurement errors mentioned above, the feature-tracking algorithm, the vehicle trajectory, and the image scene [10].

![Graphical depiction of horizontal alert level (HAL) and vertical alert level (VAL).](image)

Figure 1.1: Graphical depiction of horizontal alert level (HAL) and vertical alert level (VAL).

With vision-aided inertial navigation systems quickly developing into viable alternatives to GPS, some research has been done to develop integrity monitoring [29–31]. The goal of integrity monitoring is to determine when the position being calculated by the navigation system lacks sufficient accuracy to be used. Figure 1.1 illustrates the concept of alert levels: The green marker represents the true position of the aircraft, the yellow marker represents an acceptable position navigation solution, and the red marker represents a navigation solution with an unacceptable amount of error.
There is no perfect navigation system; therefore, all navigation solutions have some error. It is essential to determine if a solution contains an acceptable amount of error with respect to the true position. This is done by breaking the error into two parts and, consequently, two specifications. To be certified for use in any particular application, the system is required to operate with less than a specified amount of error in the horizontal and vertical position solution. For integrity monitoring purposes the maximum allowable position ambiguity is referred to as the horizontal alert level (HAL) and vertical alert level (VAL), respectively as shown in Figure 1.1. In the event that errors in measurements render the system unable to provide a sufficiently accurate position solution (position certainty worse than HAL or VAL), an integrity monitoring technique is required to detect it.

In the previous research, the methods used in GPS integrity monitoring were redefined and used for vision systems. This work began solving the problem by exploring methods with the assumption assumed that only one bad measurement would occur at any given time. Though this assumption is generally valid for GPS, the probability of multiple errors in vision systems increases with the number of features tracked. The possibility of multiple errors necessitates further development of these methods to accommodate multiple errors. With vision systems, there is also the possibility of a camera misalignment or bad calibration, which will result in corruption of all measurements. Any such misalignment would pass the integrity tests laid out in the preexisting methods.

This dissertation develops methods to detect multiple errors that are either independent of each other or when all the measurements are corrupted by a systematic error. This is done with the assumption that the features being tracked by the vision system are at known locations, which is sufficient for applications where there are convenient features that are mappable, such as mid-air
refueling, formation flying, and automated landing.

1.2 Contributions

The previous section has described the motivation for further development of vision-based navigation integrity. This dissertation extends the work done by Larson [29–31], who laid a framework for image based navigation using methods from GPS integrity monitoring. Common among integrity monitoring schemes in general, is the ability to both detect and isolate a bad measurement. The work done by Larson developed a test statistic used to detect a fault in the system. Using this method, it is possible to isolate a single error using existing parity space methods. However, as mentioned in the previous section, multiple errors are more likely to occur in a vision-based navigation system.

The first contribution of this work is the development of a Bayes-inspired algorithm for detecting and isolating multiple bad measurements. The work done by Larson is effective in detecting multiple random errors in vision systems, and this algorithm is an expansion of that work, utilizing his method for detection and providing a method for the identification/isolation of multiple measurement errors.

The second contribution of this dissertation is the development of the Dilution of Precision (DOP) for vision navigation systems that is linked to the performance of the integrity monitoring system. The cited work by Larson was accompanied by an analysis of how measurement geometry effects the relationship between the test statistic and the horizontal position error. This analysis provided a sense of how the integrity monitoring method would perform in different situations. This contribution is an expansion of that work and provides a metric that can be used to determine
the ability of the system to detect errors. DOP for vision systems is similar to DOP for GPS systems and is used to determine if the system can detect the fault.

The third and final contribution of this dissertation is the development of a method to detect systematic errors in a vision-based navigation system. Earlier methods work by creating a test statistic that looks for agreement in the measurements from an image. If one of the measurements indicates something different than the others, it sets off an alarm; however, if all measurements are at fault due to a systematic error, the error will not be detected. The method developed in this dissertation looks for agreement between two separate vision-based systems. If there is disagreement, then the test statistic increases above the threshold and an alarm is sounded. An analysis is subsequently done to show the effect of misalignment and scale errors with regard to the test statistic.

Together, these contributions make it possible to address multiple errors in an vision-based navigation system. Work done by Larson was the first work done in this area and only provided for the scenario involving a single fault. This dissertation sets aside the assumption that there is only one fault and searches for multiple faults. The first two contributions expand the cited work, which is useful for detecting multiple independent faults, by providing a metric to determine the sensitivity of the method and allowing for isolation of multiple errors. In the event that an error is systematic, the third contribution is a method for its detection. Together, these contributions allow for a vision system that is more robust to errors, which can easily exist in an vision-based navigation system.
1.3 Outline

Chapter 2 introduces the relevant background needed to understand this work in introduced, and chapter 3 discusses an overview of the current methods used in GPS integrity monitoring. Chapter 4, reviews the current literature on integrity monitoring of vision navigation systems. Chapter 5 develops the method for isolating multiple random errors in vision-based navigation systems. This is followed by the development of Dilution of Precision for vision systems in Chapter 6. Then in Chapter 7, a method is introduced and analyzed for use in detecting systematic errors in vision-navigation systems. The final chapter of this dissertation serves as a conclusion and discussion of contributions and future research avenues.

1.4 Mathematical Notation

For clarity, the following is a description of the mathematical notation that will be used in this proposal:

**Scalars:** Scalars are represented by italic characters such as $a$ or $b$.

**Vectors:** Vectors are represented by bold lower case letters, shown as $a$ or $b$, and are usually in column form.

**Matrices:** Matrices are represented by bold upper case letters, such as $A$ or $B$ and the scalar values of a matrix can be referenced as $A_{ij}$ with the $i^{th}$ row and $j^{th}$ column element.

**Transpose:** A vector or matrix transpose is denoted by a superscript $T$, as in $a^T$.

**Estimated Variables:** Estimates of random variables are denoted by adding a ”hat” symbol, such as $\hat{a}$. 
Calculated and Measured Variables: Variables that contain errors due to their being measured are distinguished by a tilde symbol, as in \( \tilde{\mathbf{a}} \).

Reference Frame: Navigation vectors are defined with respect to reference frames; a superscript letter is used to denote the current frame of reference, as in \( \mathbf{x}^a \).

Direction Cosine Matrix: Direction Cosine Matrices are matrices that rotate vectors from one frame of reference to another, as in \( \mathbf{C}^b_a \) which, when premultiplied to a vector, converts the vector from the \( a \)-frame to the \( b \)-frame.

Identity Matrix: Identity matrices are denoted by a bold capital letter \( \mathbf{I} \), as in \( \mathbf{I} \).

Relative Position or Motion: When a vector represents relative position or motion, subscripts are combined, as in \( \mathbf{p}^c_{ab} \) is the position of the \( a \)-frame with respect to the \( b \)-frame expressed in \( c \)-frame coordinates.
Background

2.1 Random Variables

When working with random events, it is necessary to quantify them. A random variable is defined as a function from a sample space, \( S \), into real numbers. Given a sample space, \( S = \{s_1, \ldots, s_n\} \), where \( s_i \) is a possible outcome with a probability function \( P \) and the random variable \( X \) with range \( \chi = \{\chi_1, \ldots, \chi_m\} \), it is possible to define a probability function \( P_X \) on \( \chi \). \( X = \chi_i \) will be observed if and only if the output of the random event is such that \( X(s_j) = \chi_i \). Therefore, \( P_X(X=\chi_i) = P(\{s_j \in S : X(s_j) = \chi_i\}) \).

2.1.1 Normal Distribution

The normal distribution, also known as the Gaussian distribution, is one of the best known distributions in statistics because it is tractable analytically. In addition, the Central Limit Theorem can be used to show that the normal distribution is a fair approximation for a variety of distributions with large samples. The normal distribution has two parameters, which are the mean, \( \mu \), and the variance, \( \sigma^2 \), and is denoted by \( N(\mu, \sigma^2) \). The probability distribution function (pdf) of a normal distribution, \( N(\mu, \sigma^2) \) is

\[
P(X = \chi) = f(\chi|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(\chi-\mu)^2}{(2\sigma^2)}}. \tag{2.1}
\]
Further, if $X \sim N(\mu, \sigma^2)$, then the random variable $Z = (X - \mu)/\sigma$ has a standard normal distribution, $N(0, 1)$ with mean, $\mu = 0$, and variance, $\sigma^2 = 1$. Figure 2.1 is a plot of probability density functions of a normal distribution. The plot displays four examples with zero mean, $\mu = 0$, and with the standard deviation, $\sigma$, equal to values 1, 2, 3, and 4. Additional details about the normal distribution can be found in [12, 22, 28].

![Probability Density Function of Normal Distribution](image)

Figure 2.1: Plot of the probability distribution function (pdf) of a normal distribution with standard deviation $\sigma = 1, 2, 3, 4$.

2.1.2 Chi-Squared Distribution

The sum of the squares of $k$ standard normal random variables yields a new random variable, $Q$, as
\begin{equation}
Q = \sum_{i=1}^{k} Z_i^2. \tag{2.2}
\end{equation}

\(Q\) is described by the chi-squared distribution with \(k\) degrees-of-freedom and is denoted as \(Q \sim \chi^2(k)\). The chi-squared distribution has only one parameter, the degrees of freedom, \(k\). Its probability density function is

\[ P(X = x) = f(x, k) = \begin{cases} 
 x^{(k/2)-1}e^{x/2} & , \quad x \geq 0 \\
 2^{(k/2)}\Gamma\left(\frac{k}{2}\right) & , \quad x < 0 
\end{cases} \tag{2.3} \]

where the gamma function, \(\Gamma(k/2)\), is defined as

\[ \Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}dt, \tag{2.4} \]

and can be expressed in a closed form given integer values of \(n > 0\) taking the form

\[ \Gamma(n) = (n - 1). \tag{2.5} \]

Figure 2.2 is a plot of probability density functions of a \(\chi^2\) distribution. The plot displays four examples with different degrees-of-freedom equal to 2, 4, 6, and 8. Additional details about the chi-squared and gamma distributions can be found in [12, 22, 28].

### 2.1.3 Non-Central Chi-Squared Distribution

In the event that \(Q\) is a sum of squares of normal random variables with a non-zero mean, \(\mu\), and variance, \(\sigma^2\), then \(Q\) is from a non-central chi-squared distribution with \(k\) degrees of freedom and non-centrality parameter, \(\lambda\), which is based on the mean of the random variables. The random variable \(Q\) is denoted as \(Q \sim \chi^2(k; \lambda)\) with the following definition
Figure 2.2: Plot of the probability distribution function (pdf) of a $\chi^2$ distribution with degrees of freedom $dof = 2, 4, 6, 8$. 
\[ Q = \sum_{i=1}^{k} \left( \frac{X_i}{\sigma_i} \right)^2 \]  

(2.6)

where \( X_i \) is a normally distributed random variable with mean, \( \mu_i \) and variance, \( \sigma_i \). The non-centrality parameter is

\[ \lambda = \sum_{i=1}^{k} \left( \frac{\mu_i}{\sigma_i} \right)^2 \]  

(2.7)

The probability density function of a non-central chi-squared distribution is

\[
P(X = x) = f(x, k, \lambda) = \begin{cases} 
\frac{1}{2} e^{-(x+\lambda)/2} \left( \frac{x^{k/2}}{\lambda^{k/2}} \right) I_{k/2-1}(\sqrt{\lambda x}), & x \geq 0 \\
0, & x < 0
\end{cases}
\]  

(2.8)

where \( I_a(y) \) is a modified Bessel function of the first kind and can be calculated as

\[
I_a(y) = (y/2)^a \sum_{j=0}^{\infty} \frac{(y^2/4)^j}{j! \Gamma(a+j+1)}.
\]  

(2.9)

Figure 2.3 is a plot of probability density functions of a non-central \( \chi^2 \) distribution. The plot displays four examples with ten degrees-of-freedom, d.o.f.=10, and varying non-centrality parameter, \( \lambda \), equal to 2, 4, 6, and 8. Additional details about this distribution can be found in [12, 23, 28].

### 2.1.4 Chi and Non-Central Chi Distributions

The chi and non-central chi distribution are very similar to the chi squared and non-central chi squared distributions. In both cases, the random variable is defined as
Figure 2.3: Plot of the probability distribution function (pdf) of a non-central $\chi^2$ distribution with ten degrees of freedom ($dof = 10$) and non-centrality parameter $\lambda = 2, 4, 6, 8$. 
\[ Q = \sqrt{\sum_{i=1}^{k} \left( \frac{X_i}{\sigma_i} \right)^2 } \]  

(2.10)

where \( X_i \) is a normally distributed random variable with mean, \( \mu_i \) and variance, \( \sigma_i \).

In the case of the non-central chi distribution, there are two parameters, the degrees of freedom, \( k \) and the non-centrality parameter \( \lambda \). The non-centrality parameter is the same as in the non-central chi-squared distribution and can be calculated using equation (2.7). The probability density function of a non-central chi distribution is nearly the same as that for the chi-squared, with the exception of the \( x^2 \) terms and is expressed as

\[
P(X = x) = f(x, k, \lambda) = \begin{cases} 
\frac{1}{2} e^{-\left(\frac{x^2+\lambda}{\lambda}\right)} \left( \frac{x^2}{\lambda} \right)^{k/4-1/2} I_{k/2-1}(\sqrt{\lambda x^2}), & x \geq 0 \\
0, & x < 0 
\end{cases}
\]  

(2.11)

where \( I_a(y) \) is a modified Bessel function seen in equation (2.9).

Figure 2.4 is a plot of probability density functions of a non-central \( \chi^2 \) distribution. The plot displays four examples with ten degrees-of-freedom, d.o.f.=10 and varying non-centrality parameter, \( \lambda \), equal to 2, 4, 6, and 8.

If the random variable, \( Q \) is the square root of the sum of \( X_i \) random variables with zero mean (\( \mu = 0 \)), then they will have a non-centrality parameter, \( \lambda \) equal to zero. This results in a standard chi distribution, such that the distribution has only one parameter, the degrees-of-freedom, \( k \). The resulting probability density function is

\[
P(X = x) = f(x, k) = \begin{cases} 
\frac{x^{(k)-1} e^{x^2/2}}{2^{k/2} \Gamma \left( \frac{k}{2} \right)}, & x \geq 0 \\
0, & x < 0 
\end{cases}
\]  

(2.12)
Figure 2.4: Plot of the probability distribution function (pdf) of a non-central $\chi$ distribution with ten degrees of freedom ($dof = 10$) and non-centrality parameter $\lambda = 2, 4, 6, 8$. 
with $\Gamma(k/2)$ denoting the gamma function such that

$$
\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} \, dt
$$

and can be expressed in a closed form given integer values of $n > 0$ taking the form

$$
\Gamma(n) = (n - 1).
$$

Figure 2.5 is a plot of four examples of the probability density function of a $\chi^2$ distribution. The plot displays four examples with varying degrees-of-freedom, d.o.f., equal to 2, 4, 6, and 8.

Figure 2.5: Plot of the probability distribution function (pdf) of a $\chi$ distribution with ten degrees of freedom, $d.o.f. = 2, 4, 6, 8$. 
2.1.5 Binomial Distribution

The binomial distribution is a discrete distribution. It describes the probability of obtaining \( x \) positive results with a sequence of \( n \) independent attempts with probability of positive \( p \). The probability mass function (PMF) of the binomial distribution is

\[
P(X = x | p, k) = f(x, p, k) = \binom{k}{x} p^x (1 - p)^{k-x},
\]

(2.15)

where,

\[
\binom{k}{x} = \frac{k!}{x!(k-x)!},
\]

(2.16)

is the binomial coefficient or 'choose' function. The binomial distribution can be used to model the probability of a number of positive attempts with a sample size of \( n \) if there is a replacement (each attempt has a probability independent of previous successes). If sampling is carried out without replacement (each attempt is not independent) then the distribution is called the hypergeometric distribution. Additional details about this distribution can be found in [12, 24, 28].

Figure 2.6 is a plot of probability mass functions of a binomial distribution. The plot displays four examples with fifty degrees-of-freedom (\( k = 50 \)) and varying probability of positive result \( p \) equal to 10%, 20%, 40% and 60%.

2.1.6 Hypergeometric Distribution

The hypergeometric distribution is a discrete distribution. It describes the probability of obtaining \( x \) positive results with a sequence of \( k \) samples from a population of size \( N \) with \( M \) positive results. This distribution makes the assumption that each sample is done without replacement, meaning that each sample affects the next and are not independent. The probability mass function
Figure 2.6: Plot of the probability mass function (pmf) of a binomial distribution with 50 degrees-of-freedom ($k = 50$) and probability of positive $p = 0.1, 0.2, 0.4, 0.8$. 
(pmt) of the hypergeometric distribution is

$$P(X = x|N, M, k) = \binom{M}{x} \binom{N-M}{k-x} \binom{N}{k}$$

where,

$$\binom{N}{k} = \frac{N!}{k!(N-k)!}$$

is the same choose function used for the binomial distribution.

Figure 2.7 is a plot of probability mass functions of a hypergeometric distribution. The plot displays four examples with pollution size ($N = 100$), constant number of samples taken ($k = 50$), and varying positive results available ($M = 10, 20, 30, 40$).

Figure 2.8 is a plot of additional probability mass functions of a hypergeometric distribution with varying degrees of freedom rather than positive result population as in Figure 2.7. The plot displays four examples with pollution size ($N = 100$), thirty positive results ($M = 30$), and varying number of samples taken (degrees-of-freedom), $k = 10, 20, 30, 40$. Additional details about this distribution can be found in [12, 24, 28].

### 2.2 Least Squares

This section serves as the background on the use of least-squares to find an optimal solution to a set of linear equations. Integrity monitoring is calculated by developing a measure of the quality of a solution and the measurements used to compute it. Navigation systems can make use of models of
Figure 2.7: Plot of the probability mass function (pmf) of a hypergeometric distribution with population size, $N = 100$, varying positive result population $M = 10, 20, 30, 40$, and varying number of samples taken $k = 50$. 
Figure 2.8: Plot of the Probability Mass Function (pmf) of a Hypergeometric Distribution with population size, $N = 100$, positive result population $M = 30$, and varying number of samples taken $k = 10, 20, 30, 40$. 

Figure 2.8: Plot of the Probability Mass Function (pmf) of a Hypergeometric Distribution with population size, $N = 100$, positive result population $M = 30$, and varying number of samples taken $k = 10, 20, 30, 40$. 

Probability Mass Function of Hypergeometric Distribution

- $n=10$
- $n=20$
- $n=30$
- $n=40$
both vehicle dynamics and sensor measurements. Given the relative linearity around an estimated operating point with respect to the error, it is common practice to linearize these models to take advantage of well established mathematical techniques such as least squares. The deterministic measurement model, relating a state vector $\mathbf{x}$ to a measurement vector $\mathbf{z}$ is given in linear form as

$$\mathbf{z} = \mathbf{H}\mathbf{x},$$

(2.19)

where the matrix $\mathbf{H}$ is a linear measurement model. The ability to solve for the states $\mathbf{x}$ despite noise and corrupted values for the measurements $\mathbf{z}$ is at the core of a navigation system. In most navigation systems there are more measurements than needed to find a unique solution, making the system overdetermined (with $m$ equations and $n$ unknowns, $m > n$). The fact the that the system is overdetermined results in $\mathbf{z} = \mathbf{H}\mathbf{x}$ being inconsistent unless the measurements are perfect and free from noise. Since measurements are never perfect, $\mathbf{z} - \mathbf{H}\mathbf{x} \neq 0$. Least squares is a method used to find an estimate of $\mathbf{x}$, written as $\hat{\mathbf{x}}$, such that the value of $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ is minimized. The least square solution can be found by premultiplying equation (2.19) by $\mathbf{H}^T$ such that

$$\mathbf{H}^T(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) = 0.$$  

(2.20)

After distributing and rearranging terms, the normal equation from statistics is found [50] as

$$\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}} = \mathbf{Hz}$$

(2.21)

If the columns of $\mathbf{H}$ are independent then $\mathbf{H}^T\mathbf{H}$ is invertible [50] and the best estimate of $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z},$$

(2.22)

which is the least-squares solution. This makes the assumption that all measurements are of equal
quality. If it is known that some measurements are of better quality, then it is best to use a weighted least squares method [50]. This is done by premultiplying both sides of equation (2.19) by a weighted matrix $W$, which yields

$$Wz = WHx.$$  

(2.23)

The corresponding weighted least squares solution is expressed as

$$H^TW^TWz = H^TW^WH\hat{x}$$  

(2.24)

This can be simplified by letting $C = W^TW$ such that

$$H^TC\hat{x} = H^TCz$$  

(2.25)

Premultiplying both sides of equation (2.25) by the inverse of $H^TC\hat{x}$ produces the weighted estimate of the states

$$\hat{x}_W = (H^TC)^{-1}H^TCz$$  

(2.26)

Typically it is best to choose the weighted matrix $C$ as the inverse of the covariance matrix of the measurements. The covariance matrix is typically represented as the matrix $R$ with diagonal elements equal to the variance of the noise in each corresponding measurement in $z$ and off-diagonal elements equal to the cross-covariance of measurement errors [50]. If the weights are chosen such that $C = R^{-1}$, then the weighted estimate of the states is

$$\hat{x}_W = (H^TR^{-1}H)^{-1}H^TR^{-1}z.$$  

(2.27)
2.3 Kalman Filtering

A Kalman filter is an effective means of combining various types of sensor information and system knowledge in the form of a model and generate an optimal estimate of the states of the system. The name filter is often used when something is being purified (rid of unwanted contaminants). In essence, the Kalman Filter is a filter for measurements, filtering out unwanted uncertainty (measurement noise and model noise). [17]

The Kalman filter has two distinct steps that are repeated in discrete time at each instance. The first step is the prediction/extrapolation, which utilizes a model of the system to predict the states of the system after one time interval. The second step is the observation/update, during which measurements are used in combination with the prediction to estimate the states of the system at that time interval.

A linear system is often modeled using state-space representation, as seen in equations (2.28) and (2.29) for a continuous time system, and equations (2.30) and (2.31) for a discrete time system.

**Continuous Time System:**

\[
\dot{x}(t) = Fx(t) + w(t)
\]  \hspace{1cm} (2.28)

\[
y(t) = Cx(t) + v(t)
\]  \hspace{1cm} (2.29)

**Discrete Time System:**

\[
x(k + 1) = \Phi x(k) + w(k)
\]  \hspace{1cm} (2.30)

\[
z(k) = Hx(k) + v(k)
\]  \hspace{1cm} (2.31)
Figure 2.9: Nomenclature and the Steps of the Kalman Filter.

System Model and Measurement Noise / Uncertainty:

\[ E\{w(k)\} = 0 \]  \hspace{1cm} (2.32)
\[ E\{w(k)w^T(k)\} = Q(k) \]  \hspace{1cm} (2.33)
\[ E\{v(k)\} = 0 \]  \hspace{1cm} (2.34)
\[ E\{v(k)v^T(k)\} = R(k) \]  \hspace{1cm} (2.35)
\[ E\{v(k)v^T(k)\} = R(k) \]  \hspace{1cm} (2.36)

• **Prediction / Extrapolation:** This step of the Kalman filter extrapolates the state estimate and error covariance matrix, using equations (2.37) and (2.38). Equation (2.37) predicts the states of the system by using a rough state space model $F$. The error covariance matrix is then updated with equation (2.38) using the model $F$ and $Q$, which describes the uncertainty of the model in terms of variance.

**State Estimate Extrapolation:**

\[ \hat{x}_{k+1}(-) = \Phi \hat{x}_k(+) \]  \hspace{1cm} (2.37)
Error Covariance Extrapolation:

\[ P_{k+1}(-) = \Phi P_k(+) \Phi^T + Q_k \quad (2.38) \]

- **Observation / Update:** This step of the Kalman filter uses equations (2.39), (2.40) and (2.41) to update the state estimate, \( \hat{x}_{k+1} \), with a measurement/observation \( z_{k+1} \), the error covariance matrix, \( P_{k+1} \), and the Kalman gain matrix \( K \). The combining of the observation and the prediction is done using a special gain, known as the kalman gain, \( K \). The Kalman gain is based on the knowledge of the uncertainties.

State Estimate Update:

\[ \hat{x}_{k+1}(+) = \hat{x}_{k+1}(-) + K_{k+1} \left[ z_{k+1} - H_{k+1} \hat{x}_{k+1}(-) \right] \quad (2.39) \]

Error Covariance Update:

\[ P_{k+1}(+) = [I - K_{k+1} H_{k+1}] P_{k+1}(-) \quad (2.40) \]

Kalman Gain:

\[ K_k = P_k(-) H_k^T \left[ H_k P_k(-) H_k^T + R_k \right]^{-1} \quad (2.41) \]

### 2.4 Extended Kalman Filter

The original Kalman filter is an excellent method for state estimation of linear systems. Unfortunately, not all systems are linear. For those cases, there is the extended Kalman filter. The extended Kalman filter (EKF) works by using the non-linear model to predict the states and measurements,
and linearizes the model about the state-estimate for computing the covariance $P$. [17]

The **discrete extended kalman filter** is implemented using the following equations:

- **Computing the predicted state estimate:**

  $$\hat{x}_k(-) = \Phi_{k-1}(\hat{x}_{k-1}(+))$$  \hspace{1cm} (2.42)

- **Computing the predicted measurement:**

  $$\hat{z}_k = h_k(\hat{x}_k(-))$$  \hspace{1cm} (2.43)

- **Linear approximation:**

  $$\Phi[1]_{k-1} \approx \frac{\partial \phi_k}{\partial x} |_{x=\hat{x}_k(+)}$$  \hspace{1cm} (2.44)

- **Conditioning the predicted estimate on the measurement:**

  $$\hat{x}_k(+) = \hat{x}_k(-) + K_k(z_k - \hat{z}_k),$$  \hspace{1cm} (2.45)

- **Linear approximation:**

  $$H[1]_{k-1} \approx \frac{\partial h_k}{\partial x} |_{x=\hat{x}_k(-)}$$  \hspace{1cm} (2.46)

- **Computing a priori covariance matrix:**

  $$P_k(-) = \Phi[1]_{k-1}P_{k-1}(+)^T \Phi[1]_{k-1}^T + Q_{k-1}$$  \hspace{1cm} (2.47)
• Computing the Kalman Gain:

\[ K_k = P_k(-)H_k^{[1]^T}[H_k^{[1]}P_k(-)H_k^{[1]^T} + R_k]^{-1} \]  

(2.48)

• Computing the a posteriori covariance matrix:

\[ P_k(+) = \{I - K_kH_k^{[1]}\}P_k(-) \]  

(2.49)

2.5 Observability of Kalman Filter

The Kalman filter for an INS often contains many states (15 or more), with each error modeled as a state in the filter. More often than not, there are more states than there are measurements, which makes it possible for some states to be unobservable. This happens when some states do not affect the observations or if the observations are influenced by multiple states in the same manner. Determining the observability of the system can be done using an observability matrix [16] and [46]. If the observability matrix is full rank, then all the states are observable. The observability can be easily checked using MATLAB during simulation.

The observability of the error states is dependent on the orientation of the INS and consequently, the direction cosine matrix relating the body frame to the navigation frame [21]. If the body frame is perfectly aligned or close to the navigation frame then the observability is no longer full rank. However, this will not occur when a vehicle is in dynamic motion. Although observability is not the focus of this dissertation, it is taken in to consideration.

29
2.5.1 Navigation Frames

A navigation frame is a coordinate system in which navigation calculations are often performed. Often, multiple coordinate systems / navigation frames.

A few common frames used in navigation are:

- **Body Frame**: This is the frame for the various inertial sensors. This frame moves with the vehicle that is navigating, often with the x-axis pointing through the front of the vehicle, y-axis to the right, and the z-axis pointing down, completing a right hand coordinate system. The origin is usually placed at the center of gravity to simplify calculations for aerial vehicles.

- **Earth Surface NED (North-East-Down)**: The NED frame is a useful form for vehicles trying to navigate. The North and East axis form a plane tangential to the surface of the earth and the down axis points in the direction of gravity. This frame works well when operating near the origin, but it does not take into account the curvature of the earth. Therefore, it is not good for global navigation, but works well for local navigation.

- **World Geodetic System-1984 (WGS-84)**: WGS-84 is the elliptical mapping of the earth in an Earth fixed frame, with a location given in terms of latitude, longitude, and altitude. Latitude ($\phi$) is zero at the equator and reaches 90° and −90° at the geographic North and South Poles, respectively. Longitude ($\lambda$) is equal to zero at the Greenwich meridian and goes to 180° both East and West. Altitude ($h$) is the height in meters above the ellipsoid.

- **Earth Centered Earth Fixed (ECEF)**: The ECEF frame is an Earth fixed frame similar to WGS-84, but is in cartesian coordinates. The origin of this coordinate frame is located at the Earth’s center of mass. The x-axis points through the Greenwich meridian where the latitude and longitude are equal to zero ($\lambda = \phi = 0$). The z-axis is pointing from the origin to the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semimajor Axis</td>
<td>a</td>
<td>6378137 m</td>
</tr>
<tr>
<td>Semiminor Axis</td>
<td>b</td>
<td>6356752.314 m</td>
</tr>
<tr>
<td>Angular Velocity of the Earth</td>
<td>( \omega_e )</td>
<td>7292115.1467 ( \text{rad} ) ( \text{sec}^{-1} )</td>
</tr>
<tr>
<td>Earth’s Gravity Constant</td>
<td>( \mu )</td>
<td>3986005 ( E8 ) ( \text{m}^3 \text{ s}^{-2} )</td>
</tr>
</tbody>
</table>

Table 2.1: Important WGS-84 Parameters.

North Pole and is parallel to the axis of rotation of the Earth. The y-axis is at right angles to the x- and z-axis completing the right hand orthogonal system.

- **Earth Centered Inertial (ECI):** The ECI frame is similar to the ECEF frame, but with one major difference. The ECI frame does not rotate with the Earth. The x-axis is instead pointing to a distant star called the Vernal Equinox. This axis lies in the equatorial plane (\( \lambda = 0 \)). The z-axis remains along the axis of rotation of the Earth and the y-axis lies orthogonal to the x- and z- axis.

## 2.6 Attitude/Orientation Representation

There are two different ways that a vehicle’s attitude will be represented in this thesis. There are the Euler angles and the direction cosine matrix (\( C_{bn}^b \)). On the frame of a vehicle, a strapdown IMU will be used. The data from it will be in terms of the body frame and will need to be transformed into the navigation frame. That is exactly what the matrix \( C_{bn}^b \) does. It rotates vectors from the the body frame (\( b \)) to the navigation frame (\( n \)). This rotation is done with three angles called Euler angles (roll \( \phi \), pitch \( \theta \), and heading \( \psi \)) and are performed relative to the body frame.

### 2.6.1 Euler Angles

- **Heading \( \psi \):** The rotation about the z-axis is known as the heading and yaw of a vehicle. The z-axis is pointing down through the bottom of the vehicle and to rotate around it would
change the direction of travel in the navigation frame. Heading and yaw are usually the same unless the vehicle is in a sideslip, which can be caused by wind or dynamics of the airframe [45].

\[
R(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = C^n_1 \tag{2.50}
\]

- **Pitch \( \theta \)**: The rotation about the y-axis of the body attached frame (frame-1 in this case) is referred to as the pitch angle in avionics. This is because the y-axis typically points down the wings of an aircraft. To rotate about the y-axis would “pitch” the nose of the vehicle up or down relative to the horizon [45].
$\begin{bmatrix}
\cos(\theta) & 0 & -\sin(\theta) \\
0 & 1 & 0 \\
\sin(\theta) & 0 & \cos(\theta)
\end{bmatrix} = C^1_2 \tag{2.51}$

- **Roll $\phi$:** The rotation about the x-axis of the body attached frame (frame-2 in this case) is known as the roll when referring to avionics. This is because the x-axis goes from an origin at the center of mass of a vehicle through the front of the vehicle in the direction the vehicle usually travels. For aircraft, this rotation would be about the center-line of the airframe. Since this a right hand coordinate system, a clockwise rotation would be a positive roll angle [45].

$\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi) & \sin(\phi) \\
0 & -\sin(\phi) & \cos(\phi)
\end{bmatrix} = C^2_6 \tag{2.52}$
Figure 2.12: Roll rotation following right hand rule.

### 2.6.2 Direction Cosine Matrix $C^n_b$

In this representation, the rotations are maintained in a matrix form. The matrix can be calculated using the Euler angles and vice-versa. To get a direction cosine matrix, the Euler angle rotations need to be put in a sequence, which will create a rotation matrix that rotates a vector from the navigation frame to the body frame. One such sequence and the one used in this thesis can be seen in equation (2.53) [45].

$$C^n_b = R(\phi) R(\theta) R(\psi) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$  \hspace{1cm} (2.53)

The direction cosine matrix ($C^n_b$) is a 3x3 orthonormal matrix; therefore the matrix has the following properties:

$$C^n_b C^n_b^T = I^{3 \times 3}$$  \hspace{1cm} (2.54)

$$\det(C^n_b) = 1$$  \hspace{1cm} (2.55)
Equation (2.54) represents a very useful property of orthogonal matrices, the transpose of the
direction cosine matrix is equivalent to its inverse. Equation (2.55) represents the normality of the
direction cosine matrix, which ensures that when multiplied by a vector, the result will only be
rotated and not scaled. [53]

\[
C^n_b = \begin{bmatrix}
\cos(\theta) \cos(\psi) & \sin(\phi) \sin(\theta) \cos(\psi) - \sin(\psi) \cos(\phi) & \cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi) \\
\cos(\theta) \sin(\psi) & \cos(\phi) \sin(\theta) \sin(\psi) - \cos(\psi) \cos(\phi) & \cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi) \\
-\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta)
\end{bmatrix}
\] (2.56)

\(C^n_b\) matrix with small angles

For small angles, the first order Taylor series expansion can be used for the trigonometric functions
in equation (2.56) (i.e. \(\cos(\beta) \approx 1\) and \(\sin(\beta) \approx \beta\)). Applying this to equation (2.56) yields a skew
symmetric rotation matrix as seen in equation (2.57). [53]

\[
C^n_b \approx \begin{bmatrix}
1 & -\psi & \theta \\
\psi & 1 & -\phi \\
-\theta & \phi & 1
\end{bmatrix} = I^{3 \times 3} + I^{3 \times 3} \times \begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix}
\] (2.57)

2.6.3 \(C^n_b\) to Euler angle conversion

If a direction cosine matrix \((C^n_b)\) is given, then the Euler angles can be easily be derived from
equations (2.56) and (2.53), resulting in equations (2.58), (2.59), and (2.60). [53]

\[
\phi = \tan^{-1} \left( \frac{c_{32}}{c_{33}} \right) = \tan^{-1} \left( \frac{\sin(\phi) \cos(\theta)}{\cos(\phi) \cos(\theta)} \right)
\] (2.58)
\[ \theta = \sin^{-1} (-c_{31}) = \sin^{-1} (\sin(\theta)) \quad (2.59) \]

\[ \psi = \tan^{-1} \left( \frac{c_{21}}{c_{11}} \right) = \tan^{-1} \left( \frac{\cos(\theta) \sin(\psi)}{\cos(\theta) \cos(\psi)} \right) \quad (2.60) \]

Equations (2.58), (2.59), and (2.60) become numerically unstable when the pitch angle (\(\theta\)) is large. In the case of a large pitch angle it is better to use equations (2.69) and (2.70) for \(\psi\) and \(\phi\) respectively [53].

\[ c_{23} - c_{12} = \sin(\psi - \phi)(\sin(\theta + 1)) \quad (2.61) \]

\[ c_{13} + c_{22} = \cos(\psi - \phi)(\sin(\theta + 1)) \quad (2.62) \]

\[ c_{23} + c_{12} = \sin(\psi + \phi)(\sin(\theta - 1)) \quad (2.63) \]

\[ c_{13} - c_{22} = \sin(\psi + \phi)(\sin(\theta - 1)) \quad (2.64) \]

Dividing equation (2.61) with equation (2.62) yields equations (2.65) & (2.67), and dividing equation (2.63) with equation (2.64) yields equations (2.66) & (2.68). Solving for \(\phi\) and \(\psi\), gives the final two equations ((2.69) & (2.70)) [53]

\[ \frac{c_{23} + c_{12}}{c_{13} - c_{22}} = \frac{\sin(\psi + \phi)(\sin(\theta - 1))}{\cos(\psi + \phi)(\sin(\theta - 1))} = \tan(\psi + \phi) \quad (2.65) \]

\[ \frac{c_{23} - c_{12}}{c_{13} + c_{22}} = \frac{\sin(\psi - \phi)(\sin(\theta - 1))}{\cos(\psi - \phi)(\sin(\theta - 1))} = \tan(\psi - \phi) \quad (2.66) \]
\[
\tan^{-1}\left(\frac{c_{23} + c_{12}}{c_{13} - c_{22}}\right) = \psi + \phi \tag{2.67}
\]

\[
\tan^{-1}\left(\frac{c_{23} - c_{12}}{c_{13} + c_{22}}\right) = \psi - \phi \tag{2.68}
\]

\[
\psi = \frac{1}{2} \left[ \tan^{-1}\left(\frac{c_{23} + c_{12}}{c_{13} - c_{22}}\right) + \tan^{-1}\left(\frac{c_{23} - c_{12}}{c_{13} + c_{22}}\right) \right] \tag{2.69}
\]

\[
\phi = \frac{1}{2} \left[ \tan^{-1}\left(\frac{c_{23} - c_{12}}{c_{13} - c_{22}}\right) - \tan^{-1}\left(\frac{c_{23} - c_{12}}{c_{13} + c_{22}}\right) \right] \tag{2.70}
\]

### 2.6.4 $C^n_b$ Integration

For a strapdown inertial navigation system, the $C^n_b$ rotation matrix is constantly changing based on rotation rates. If the vector $[p, q, r]$ represents the rotation rate about each axis, then the continuous change rate of the $C^n_b$ matrix is as seen in equation (2.71). [53]

\[
\dot{C}^n_b = C^n_b \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} = C^n_b \Omega^n_{nb} \tag{2.71}
\]

Integrating equation (2.72) over time gives the rotation matrix $C^n_b$

\[
C^n_b = \int \dot{C}^n_b \, dt = \int C^n_b \Omega^n_{nb} \, dt \tag{2.72}
\]

Equation (2.72) is for continuous time. To implement this in software, the integration needs to be discretized. For discrete time, simple Euler integration, equation (2.73), will suffice given that the time between samples/updates ($\Delta T$) is small.
\[ C^a_b(k+1) = C^a_b(k) + \dot{C}^a_b(k) \Delta T = C^a_b(k)(I^{3\times3} + \Omega^n_{nb}) \Delta T = C^a_b(k) \left[ \begin{array}{ccc} 1 & -r & q \\ r & 1 & -p \\ -q & p & 1 \end{array} \right] \Delta T \] (2.73)

For a more accurate integration in cases where the sample time is less than ideal, the Taylor series expansion of the integration is used as seen in equations (2.74) through (2.79).

\[ C^a_b(k+1) = C^a_b(k) \exp \left( \int_{t_k}^{t_{k+1}} \Omega^b_{nb} dt \right) \] (2.74)

Assuming that the rotation rates are constant over the sample period, we can write:

\[ \int_{t_k}^{t_{k+1}} \Omega^b_{nb} dt = \Omega^b_{nb} \Delta T \] (2.75)

Applying equation (2.75) to equation (2.74) yields

\[ C^a_b(k+1) = C^a_b(k) \exp(\Omega^b_{nb} \Delta T) \] (2.76)

If \( \sigma \times = \Omega^b_{nb} \Delta T \) then the following is the Taylor series expansion of the exponential in equation (2.76).

\[ \exp(\Omega^b_{nb} \Delta T) = \exp(\sigma \times) = I^{3\times3} + [\sigma \times] + \frac{[\sigma \times]^2}{2!} + \frac{[\sigma \times]^3}{3!} + \frac{[\sigma \times]^4}{4!} \ldots \] (2.77)

After sufficient manipulation, as seen in [2] and [53], the following equations are obtained.

\[ \exp(\Omega^b_{nb} \Delta T) = \exp(\sigma \times) = I^{3\times3} + \frac{\sin \sigma}{\sigma} [\sigma \times] + \frac{(1 - \cos \sigma)}{\sigma^2} [\sigma \times]^2 \] (2.78)
\[ \sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} = \sqrt{(\omega_x \Delta T)^2 + (\omega_y \Delta T)^2 + (\omega_z \Delta T)^2} \quad (2.79) \]

### 2.6.5 Inertial Navigation Equations

The following equations of navigation are general equations used for inertial systems in the global (earth) frame [14, 53].

\[
\dot{v}^e = C_b^e \mathbf{f}^b - 2\omega_{ie}^e \times v^e + g_l^e \quad (2.80)
\]

where, \(v^e\) is a \([3 \times 1]\) vector with the components of the vehicles velocity in the earth frame, \(C_b^e\) is a \([3 \times 3]\) direction cosine rotation matrix from the body frame to the earth frame, \(\omega_{ie}^e\) is a \([3 \times 1]\) vector with the components of the earth’s rotation in the earth inertial frame, \(\mathbf{f}^b\) is a \([3 \times 1]\) vector with the specific forces from the accelerometers in the body frame, and \(g_l^e\) is a \([3 \times 1]\) vector with the components of the local gravity vector in the earth frame.

Integrating equation (2.80) yields the velocity of the vehicle and integrating the second time yields the position of the vehicle as seen in equations (2.81) and (2.82), respectively.

\[
v^e = \int_0^t f^e dt - \int_0^t 2\omega_{ie}^e \times v^e dt + \int_0^t g_l^e dt \quad (2.81)
\]

\[
x^e = \int_0^t v^e dt \quad (2.82)
\]

where \(f^e = C_b^e \mathbf{f}^b\). To implement these equations, they must first be discretized, using Euler integration as

\[
v^e(k + 1) = v^e(k) + \dot{v}^e(k) \Delta T \quad (2.83)
\]
\[ x^e(k + 1) = x^e(k) + v^e(k)\Delta T \] (2.84)

### 2.7 INS Aiding

There are several different approaches to couple other navigation systems with an INS. The most popular navigation system combination is GPS coupled with an INS. The following are examples of three different techniques for integrating GPS with an INS: uncoupled, loosely coupled, and tightly coupled [26].

- **Uncoupled**: The uncoupled filter, seen in Figure 2.13(a), consists only of an algorithm that combines the IMU measurements \((\hat{\mathbf{f}}, \hat{\mathbf{\omega}})\) with the GPS measurements \((\hat{\mathbf{r}}, \hat{\mathbf{v}})\). The two systems essentially run separately each providing a navigation solution. These navigation solutions are then combined. This can be done different ways including integrating the INS results and then averaging these with the GPS, or by using a Kalman filter.

- **Loosely Coupled**: The loosely coupled filter, seen in Figure 2.13(b), is more accurate than the uncoupled filter. IMU and GPS still act as separate units. The GPS receiver calculates a complete navigation solution and provides a computed position and velocity. This is fed into a Kalman Filter along with uncorrected navigation states from the INS. The Kalman filter uses dynamic models for both GPS and INS to estimate the errors in the navigation states. This can be implemented in either a feedback or feedforward configuration. Figure 2.13(b) shows the use of a feedback scheme where corrections are sent back to the INS.

- **Tightly Coupled**: The tightly coupled filter, seen in Figure 2.13(c), is the best among the three, but it is also the most complex. Rather than working as separate units the GPS and INS are working tightly integrated with one another. Both devices send raw data into a Kalman
Figure 2.13: Layouts of different types of GPS aided INS: (a) Uncoupled Integration, (b) Loosely Coupled Integration, and (c) Tightly Coupled Integration.

filter, which tracks and feeds back the errors in the INS and provides tracking aid to the GPS.
2.8 Global Positioning System

The Global Positioning System is a space-based navigation system comprised of satellites in medium-earth orbit. It provides accurate three-dimensional position and timing information globally. The GPS system has excellent long-term accuracy, but has low short term precision due to high frequency noise errors, which effect short-term performance. This is one reason why it is often coupled with an INS, which has excellent short term accuracy, but suffers from drift caused by sensor errors. Coupling the systems provides a better solution than either could produce separately and can result in a reduction in performance requirements for the independent systems while operating.

GPS utilizes time-of-arrival measurements made from signals sent by satellites. Since users only receive signals, they operate passively, allowing for an unlimited number of users simultaneously [36]. The signals are received by the user from the satellites, which are at known locations. The time difference between when the signal was sent and received is multiplied by the speed of light to determine the range to a given satellite. The time is known precisely on the satellite by use of redundant atomic clocks. However, the receiver is not equipped with an accurate clock. This results in the receive time not being known precisely. Therefore, time is one of the variables which is solved for in addition to position. Since the time is not precisely known, the measurement made by the receiver is called a pseudorange, because it is not the true range due to the receiver time being unknown.

Since this research is not focused on GPS, the actual signal is not described here, but for details on the GPS signal, see [19, 26, 36, 48]. There are four different measurements that can be made from the signal from the satellites. They are pseudorange, doppler, carrier-phase, and carrier-to-noise density. These measurements are raw and should not be confused with the computed
outputs of position and velocity generated by the receiver. Access to the raw measurements from a receiver are required for most GPS aided INS methods. The most commonly used measurement is the pseudorange and is often the only measurement used.

2.8.1 Pseudorange Measurements

As mentioned in the previous section, the pseudorange is the true range between a user and a satellite plus a bias caused by the uncertainty in time along with other error sources. The main source of the bias is the receiver clock, but the other contributors are the satellite clock, atmospheric effects, and multipath interference. The pseudorange equation is given by

\[ \rho = r + c(\delta t_r - \delta t_s) + c\delta t_{tropo} + c\delta t_{iono} + c\delta t_{mp} + v \]  

(2.85)

where, \( \rho \) is the GPS pseudorange (meters), \( r \) is the true range from the user to the satellite (meters), \( c \) is the speed of light (meters/second), \( \delta t_r \) is the receiver clock error (seconds), \( \delta t_s \) is the satellite clock error (seconds), \( \delta t_{tropo} \) is the error due to tropospheric delay (seconds), \( \delta t_{iono} \) is the error due to ionospheric delay (seconds), \( \delta t_{mp} \) is the error due to multipath interference (seconds), and \( v \) is the error due to receiver noise (meters).

The range, \( r \), is the true line-of-sight (LOS) range between the satellite and the receiver. As the signal travels through the atmosphere, the path of the signal is often distorted resulting in the errors from the ionosphere, \( \delta t_{iono} \), and the troposphere, \( \delta t_{tropo} \). Atmospheric modeling and forecasting can be used to mitigate the impact of \( \delta t_{iono} \) and \( \delta t_{tropo} \). When the signal is reflected off objects and the ground, it results in multiple copies of the same signal being received. Receiver and antenna design are used to reduce the impact of multipath and block all signals that are not the true LOS signal. By reducing these errors, the dominate term left is from the receiver clock. This error can be modeled as a clock bias term and solved for when computing the position solution. The other remaining errors are assumed to be noise-like.
Since range \( r \) is a non-linear measurement of position, the receivers calculate the position by linearizing about an initial approximated guess of the position and then solving iteratively. A full description of this method of solving for position is given in [36]. The pseudorange can be expressed in simplified form as seen in equation (2.86).

\[
\rho = \sqrt{(x_{m_i} - x_t)^2 + (y_{m_i} - y_t)^2 + (z_{m_i} - z_t)^2} + b + \epsilon \quad (2.86)
\]

where \((x_{m_i}, y_{m_i}, z_{m_i})\) is the Location of the \(i^{th}\) satellite (meters), \((x_t, y_t, z_t)\) is the true location of the receiver (meters), \(b\) is the receiver clock bias (meters), and \(\epsilon\) is the error in measurement (meters).

If the true position, \(x_t\), and bias term are expressed as \(x_t = x_0 + \delta x\) and \(b = b_0 + \delta b\), the error terms \(\delta x\) and \(\delta b\) represent the correction to be applied to the initial estimates \(x_0\) and \(b_0\). If \(\rho_c\) is a pseudorange with the corrections \(\delta x\) and \(\delta b\) applied, then the linearized equation is created using a first order Taylor Series approximation [9, 36, 40, 41] as

\[
\delta \rho = \rho_c - \rho_0 = \|x_t - x_0 - \delta x\| - \|x_t - x_0\| + (b - b_0) + \epsilon \quad (2.87)
\]

\[
\approx - \frac{(x_t - x_0)}{\|x_t - x_0\|} \delta x + \delta b + \epsilon \quad (2.88)
\]

Equation (2.89) is written in the matrix form \(z = Hx + v\), as

\[
\delta \rho = \begin{bmatrix}
\delta m_1 \\
\delta m_2 \\
\vdots \\
\delta m_n
\end{bmatrix} = \begin{bmatrix}
\frac{(x_{m_1} - x_0)}{\|x_t - x_0\|} & -\frac{(y_{m_1} - y_0)}{\|x_t - x_0\|} & -\frac{(z_{m_1} - z_0)}{\|x_t - x_0\|} \\
\frac{(x_{m_2} - x_0)}{\|x_t - x_0\|} & -\frac{(y_{m_2} - y_0)}{\|x_t - x_0\|} & -\frac{(z_{m_2} - z_0)}{\|x_t - x_0\|} \\
\vdots & \vdots & \vdots \\
\frac{(x_{m_n} - x_0)}{\|x_t - x_0\|} & -\frac{(y_{m_n} - y_0)}{\|x_t - x_0\|} & -\frac{(z_{m_n} - z_0)}{\|x_t - x_0\|}
\end{bmatrix} \begin{bmatrix}
\delta x \\
\delta b
\end{bmatrix} + \epsilon \quad (2.90)
\]
The solution of equation 2.90 can now be found using linear numerical methods such as least-squares. The least-squares solution for the overdetermined system is given by [36]

\[
\begin{bmatrix}
\delta x \\
\delta b
\end{bmatrix} = (H^T H)^{-1} H^T \delta \rho
\]

(2.91)

where

\[
H = \begin{bmatrix}
\frac{(x_{m1} - x_0)}{\|x_t - x_0\|} & \frac{(y_{m1} - y_0)}{\|x_t - x_0\|} & \frac{(z_{m1} - z_0)}{\|x_t - x_0\|} & 1 \\
\frac{(x_{m2} - x_0)}{\|x_t - x_0\|} & \frac{(y_{m2} - y_0)}{\|x_t - x_0\|} & \frac{(z_{m2} - z_0)}{\|x_t - x_0\|} & 1 \\
\vdots & \vdots & \vdots & \vdots \\
\frac{(x_{mn} - x_0)}{\|x_t - x_0\|} & \frac{(y_{mn} - y_0)}{\|x_t - x_0\|} & \frac{(z_{mn} - z_0)}{\|x_t - x_0\|} & 1
\end{bmatrix}
\]

(2.92)

These equations are solved iteratively in a method called iterative least-squares. The process is repeated until the correction is below a desired threshold.

### 2.8.2 Dilution of Precision

Dilution of precision (DOP) is a standard measure of the effect satellite geometry has on the precision of location and timing solution [1]. The observation equation for the \(i^{th}\) GPS satellite with known position \((x_i, y_i, z_i)\) and receiver at location \(p^n\) with clock bias \(c_b\) is given as the pseudorange as

\[
Z_{\rho_i} = \rho_i = \sqrt{(x_i - p^n_x)^2 + (y_i - p^n_y)^2 + (z_i - p^n_z)^2 + c_b}
\]

(2.93)

The location and clock bias of the receiver are represented in a four element state vector \(x\).

\[
x = \begin{bmatrix} p^n \\ c_b \end{bmatrix}
\]

(2.94)
Equation (2.93), the observation equation, can be written as \( Z_\rho = h(x) \), where \( h(x) \) is a non-linear measurement equation and H.O.T. stands for "higher order terms".

\[
Z_\rho = h(x) = \hat{x} + \frac{\partial h(x)}{\partial x} \bigg|_{x=\hat{x}} \delta x + \text{H.O.T.} \tag{2.95}
\]

The error in estimate of the state vector, \( \delta x \), is expressed as

\[
\delta x = x - \hat{x} \tag{2.96}
\]

Similarly, the error in the measurement, \( \delta Z_\rho \), is expressed as

\[
\delta Z_\rho = h(x) - h(\hat{x}) \tag{2.97}
\]

Combining equations (2.95) and (2.97) yields

\[
\delta Z_\rho = \frac{\partial h(x)}{\partial x} \bigg|_{x=\hat{x}} \delta x = H \delta x \tag{2.98}
\]

where \( H \) is the first order Taylor series expansion of \( h(x) \).

To calculate the DOP, it is assumed that \( \delta Z_\rho \) and \( H \) are known given a pseudorange, satellite position and estimate of the receiver’s position and that \( \delta x \) and \( \delta Z_\rho \) are normal random variables with zero mean. Premultiplying equation (2.98) by \((H^T H)^{-1}H^T\) yields

\[
\delta x = (H^T H)^{-1}H^T \delta Z_\rho \tag{2.99}
\]

The error covariance of \( x \) is given by
\[
E \langle (\delta x)(\delta x)^T \rangle = E \langle \left[ H^T H \right]^{-1} \left[ H^T \delta Z_\rho \right] \left[ H^T \delta Z_\rho \right]^T \rangle = (H^T H)^{-1} H^T \langle \delta Z_\rho \delta Z_\rho^T \rangle H \left[ (H^T H)^{-1} \right]^T
\]

(2.100)

(2.101)

The measurement covariance is assumed to be uncorrelated between satellites with variance \( \sigma^2 \) given by

\[
E \langle (\delta Z_\rho)(\delta Z_\rho)^T \rangle = \sigma^2 I_4
\]

(2.102)

Substituting equation (2.102) into (2.101) yields

\[
E \langle (\delta x)(\delta x)^T \rangle = \sigma^2 (H^T H)^{-1} H^T \left[ (H^T H)^{-1} \right]^T
\]

(2.103)

\[
= \sigma^2 \left[ (H^T H)^{-1} \right]^T = \sigma^2 (H^T H)^{-1}
\]

(2.104)

Equation (2.104) relates the noise in measurements to noise in the state vector. The dilution of precision is found in the diagonal elements of \( (H^T H)^{-1} \) in equation (2.105).

\[
(H^T H)^{-1} = \begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} \\
A_{21} & A_{22} & A_{23} & A_{24} \\
A_{31} & A_{32} & A_{33} & A_{34} \\
A_{41} & A_{42} & A_{43} & A_{44}
\end{bmatrix}
\]

(2.105)

The geometric dilution of precision (GDOP), position dilution of precision (PDOP), horizontal dilution of precision (HDOP), vertical dilution of precision (VDOP), the time dilution of precision (TDOP) are given in equations (2.106)-(2.110), representing the navigation solution sen-
sivities to pseudorange measurement errors.

\[
GDOP = \sqrt{A_{11} + A_{22} + A_{33} + A_{44}} \quad (2.106)
\]

\[
P DOP = \sqrt{A_{11} + A_{22} + A_{33}} \quad (2.107)
\]

\[
H DOP = \sqrt{A_{11} + A_{22}} \quad (2.108)
\]

\[
V DOP = \sqrt{A_{33}} \quad (2.109)
\]

\[
T DOP = \sqrt{A_{44}} \quad (2.110)
\]

2.9 Vision Based Navigation

The objective of the proposed research is to further investigate integrity monitoring techniques in vision aided/based navigation systems. The emphasis is therefore on the measurements made by vision systems in the form of pixel coordinates in an image of a known feature. The following includes essential background relating a real world target/feature and the measurement from the vision system. The focus of this research is on the measurement provided by the imaging system and not the process that created it. Therefore, the following assumptions are being made.

- A feature tracker generates measurements in the form of pixel coordinates of known features at a suitable rate.

- The vision system is calibrated in a manner that allows for the relationship between pixel coordinates and position in the camera frame to be known with any lens distortion already corrected for.
The relationship between the camera frame and vehicle body frame is known.

The use of GPS integrated with an INS has been well established. The two systems complement each other well, but there are environments and conditions that can result in GPS signals being unavailable. This led to research into the use of optical systems to aid in navigation \cite{10, 25, 38, 54, 56, 59, 60, 62}. Vision-based navigation can be done without the use of an INS, but the vision system performance is based on the quality of measurements that can be made given the environment and availability of features for tracking. The vision system can be used in a similar manner to GPS when tightly coupled with an INS to bound the errors that grow with time \cite{60}. Together, an INS and vision system have the potential to reliably provide accuracy on the level of GPS.

Optical navigation can be used in many environments including those, which are unknown \cite{18}. When the environment is unknown without features at known locations, a process called Simultaneous Location and Mapping (SLAM) is used to estimate the location of trackable features at the same time, solving for the navigation states of the vehicle. The research proposed here focuses on navigation in environments with known features at known locations. These will be tracked by a vision system that is passive, taking in a three-dimensional (3-D) scene and projecting it onto a two-dimensional (2-D) image plane.

2.9.1 Projection Model

The optical properties of a camera govern the relationship between a scene and its projection onto an image. Optics seldom exhibit ideal properties allowing for a simple model. However, many calibration and correction techniques exist to reduce and correct for non-linear optical effects \cite{15, 27, 33, 39, 60, 64}. These corrections allow for the projection to be modeled internally of an ideal thin lens. For an ideal thin lens, the projection onto the image plane is a function of the focal
Figure 2.14: Thin lens camera model.

Figure 2.15: Pinhole Camera Model.
length and the distance to the lens as shown in Figure 2.9.1. The thin lens directs the parallel light rays toward the focus resulting in an inverted image beyond the focus. This is expressed as the fundamental equation for a thin lens equation as

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f}$$

(2.111)

where $Z$ is the distance from an object in the scene to the lens, $z$ is the distance from the lens to the image plane, and $f$ is the focal length of the lens [34]. If the aperture of the lens is decreased, it can be modeled as a pinhole camera. Given the pinhole camera model depicted in Figure 2.15, all light must pass through the aperture and projects an image on a plane located at the focal length $f$ from the aperture [34].

![Camera projection model](image)

Figure 2.16: Camera projection model.

If the image plane is placed in front of the optical center, the model is further simplified as seen in Figure 2.16. This results in a non-inverted image. Given a point source location, $s^c$ relative to the optical center the resulting location on the image plane is given by
\[ s^{\text{proj}} = \left( \frac{f}{s^c_z} \right) s^c \]  

(2.112)

with \( s^c_z \) as the distance from the optical center of the camera in the \( z_c \) direction [60]. This camera projection is then converted into a digital image. The image plane coordinates need to be mapped to a coordinate system based on pixels. Assuming a rectangular \((M \times N)\) pixel grid with a height \(H\) and a width \(W\), the transformation from projection coordinates to pixel coordinates is given by

\[
\begin{bmatrix}
    -M/H & 0 & 0 \\
    0 & 0 & N \quad W \\
    0 & N/W & 0 \\
\end{bmatrix}
\begin{bmatrix}
    s^{\text{proj}}_x \\
    s^{\text{proj}}_y \\
    1 \\
\end{bmatrix} + \begin{bmatrix}
    \frac{M+1}{2} \\
    \frac{N+1}{2} \\
    0 \\
\end{bmatrix}
\]

(2.113)

Combining equations (2.112) and (2.113) yields the transformation from the camera frame to the pixel frame:

\[
\begin{bmatrix}
    -f \quad M/H \\
    0 & 0 \\
    0 & f \quad N/W \\
    0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
    s^c_x \\
    s^c_y \\
    s^c_z \\
\end{bmatrix} + \begin{bmatrix}
    \frac{M}{2} \\
    \frac{N}{2} \\
    0 \\
\end{bmatrix}
\]

(2.114)

\[
\begin{bmatrix}
    -f \quad M/H \\
    0 & 0 \\
    0 & f \quad N/W \\
    0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
    s^c_x \\
    s^c_y \\
    s^c_z \\
\end{bmatrix} = \begin{bmatrix}
    \frac{M}{2} \\
    \frac{N}{2} \\
    0 \\
\end{bmatrix}
\]

(2.115)

where \( T^\text{pix}_c \) is the homogeneous transformation matrix from camera to pixel frame [60]. To convert pixel coordinates back to camera coordinates, the inverse transformation can be used as
\[ T^{c}_{\text{pix}} = (T^{\text{pix}}_{c})^{-1} \]  
(2.116)

\[ T^{c}_{\text{pix}} = \begin{bmatrix} -\frac{H}{fM} & 0 & \frac{H(M+1)}{2fM} \\ 0 & \frac{W}{fN} & -\frac{W(N+1)}{2fN} \\ 0 & 0 & 1 \end{bmatrix} \]  
(2.117)

The coordinates are still in terms of a camera model and need to be related to the navigation frame as seen in Figure 2.17. The relationship between navigation frame and the camera frame are given by

\[ \mathbf{s}^n = \mathbf{t}^n - \mathbf{p}^n \]  
(2.118)
\[ s^c = C_n^b C_b^c s^n \]  

where \( s^n \) and \( s^c \) are the line-of-sight vectors from the camera to the target in the navigation and camera frames respectively, \( p^n \) is the position of the camera in the navigation frame, and \( t^n \) is the location of the target in the navigation frame [60].

### 2.9.2 Measurement Model

Before measurements can be used or analyzed, it is necessary to have a linearized measurement model. This research makes use of the model created by Veth [59]. For this model, a minimal error state vector for a vision aided inertial system is used, as given by

\[ \delta x = \begin{bmatrix} \delta p^n \\ \delta \psi \end{bmatrix} \]  

(2.120)

where \( \delta x \) is the error state vector, \( \delta p^n \) is the 3-dimensional error vector in position of the platform, and \( \delta \psi \) are the tilt error states [59, 60].

The measurement model for the \( i^{th} \) image feature is given by

\[ z_i = s_i^{pix} = T_{pix} s_i^c \]  

(2.121)

where \( z_i \) is the measurement vector from the \( i^{th} \) feature, \( T_{pix} \) is the homogeneous transformation matrix from the camera frame to the pixel frame, and \( s_i^c \) is the line-of-sight vector from the camera to the \( i^{th} \) feature target. This is a non-linear relationship and is expressed as a non-linear measurement equation \( h(x) \) as
\[ h(x) = \frac{1}{s_c} T_c^{pix} s_c^c \] 

(2.122)

The measurement model matrix is found by taking the first order Taylor series expansion of \( h(x) \) [29]. The measurement matrix is by with \( \mu = 1/s_c \) and \( \beta = [0 \ 0 \ 1] \).

\[
H = \frac{\partial h(x)}{\partial x} \bigg|_{x=\hat{x}} = \begin{bmatrix} \frac{\partial h}{\partial p^n} & \frac{\partial h}{\partial \psi} \end{bmatrix} 
\] 

(2.123)

\[
\frac{\partial h}{\partial p^n} = \mu T_c^{pix} \left( s_i^c \beta C^b_c C^b_n - C^c_b C^b_n \right) 
\]

(2.124)

\[
\frac{\partial h}{\partial \psi} = \mu T_c^{pix} \left( \frac{\partial s_i^c}{\partial \psi} - s_i^c \beta \frac{\partial s_i^c}{\partial \psi} \right) 
\]

(2.125)

\[
\frac{\partial s_i^c}{\partial \psi} = -C^c_b C^b_n \left[ (t^n - p^n) \times \right] 
\]

(2.126)

with \( \mu = 1/s_c \) and \( \beta = [0 \ 0 \ 1] \).
GPS Integrity Monitoring

This section provides an overview of integrity monitoring methods used in GNSS systems. It begins with a discussion on Receiver Autonomous Integrity Monitoring (RAIM). That is followed by the parity vector, and the least square residual methods for obtaining a test statistic. Lastly slope is discussed, which is the relationship between the test statistic used and the error limits for the navigation solution.

3.1 Receiver Autonomous Integrity Monitoring

The GPS system has become the system of choice for navigation due to its performance and reliability. Even though it is a fairly reliable system, its use in safety critical systems required that the reliability be guaranteed. This resulted in a considerable amount of research and development of integrity monitoring algorithms, the foremost of which is Receiver Autonomous Integrity Monitoring (RAIM) [51]. RAIM is the most useful method developed to date in that it is passive and localized to the GPS receiver without a large and complicated infrastructure of additional sensors. RAIM algorithms are not standardized among receivers, but they primarily rely on least squares residuals from a particular instant of data or similar method using a parity vector [41]. These methods have their limitations in availability of being able to detect and exclude bad measurements, and they make the assumption that there is a single measurement error, which is a valid assumption for GPS [6, 9, 40].
3.2 Parity Vector

The GPS system has become the system of choice for navigation due to its performance and reliability. Even though it is a fairly reliable system, its use in safety critical systems required that the reliability be guaranteed. This resulted in a considerable amount of research and development of integrity monitoring algorithms, the foremost of which is Receiver Autonomous Integrity Monitoring (RAIM) [51]. RAIM is the most useful method developed to date, in that, it is passive and localized to the GPS receiver without a large and complicated infrastructure of additional sensors. RAIM algorithms are not standardized among receivers, but they primarily rely on least squares residuals from a particular instant of data or similar method using a parity vector [41]. Both methods have been found to be equivalent [5] and have performance limitations in the ability to detect and exclude bad measurements. They also make the assumption that there is a single measurement error, which is a valid assumption for GPS [6, 9, 40].

The parity vector method for integrity monitoring was first presented by Potter [43] for monitoring inertial navigation systems. It was then reintroduced as a method of integrity monitoring for GPS by Sturza [51]. The following is a derivation that mirrors the one presented in mathematical detail by Sturza.

The parity vector method is dependent upon the presence of redundant measurements. In other words, the number of measurements \( m \), must exceed the number of states \( n \) being estimated, such that \( m - n \geq 1 \). A linearized measurement model is given by:

\[
z = Hx + w + b
\]

where \( z \) is the \((m \times 1)\) measurement vector that results from the product of the \((m \times n)\) measurement matrix \( H \) and the \((n \times 1)\) state vector \( x \) plus the \((m \times 1)\) vector of measurement noise \( w \) with
diagonal covariance of $\sigma^2 I$ and the $(m \times 1)$ bias vector $b$ that represents faults in the measurements. Assuming that the measurements are independent and that there are redundant measurements, the measurement matrix $H$ is not square and is consisting of independent column vectors, so that it can be successfully decomposed using QR decomposition as

$$z = QRx + w + b$$

where the resulting $Q$ has dimensions $m \times m$ and is orthogonal, meaning $Q^{-1} = Q^T$, and $R$ is an upper triangular matrix with dimensions $m \times n$ where the last $m - n$ rows contain only zeros. Premultiplying equation (3.1) by $Q^T$ yields

$$Q^Tz = Rx + Q^T(w + b).$$

The $Q^T$ and $R$ matrices can be subdivided into $Q_x^T$ and $U$ representing the first $n$ rows of $Q^T$ and $R$ respectively, and $Q_p^T$ representing the last $m - n$ rows of $Q^T$.

$$\begin{bmatrix} Q_x^T \\ Q_p^T \end{bmatrix} z = \begin{bmatrix} U \\ 0 \end{bmatrix} x + \begin{bmatrix} Q_x^T \\ Q_p^T \end{bmatrix} (w + b).$$

If regarded as a separate system of equations, the last $m - n$ rows of equation (3.3) correspond to equations that do not contain the navigation states and only contain the noise and the bias vectors. The matrix $Q_p^T$ is defined as the parity matrix $P$ with rows that are orthogonal to $z$ and columns that span the parity space of $H$ \cite{6, 55}. This allows for measurements with unobservable biases to be transformed into the parity space, where they can be observed in the form of the parity vector as

$$p_p = Q_p^Tz = Pz = P(w + b)$$
The resulting elements of the parity vector are normally distributed with mean \( \mu = Pb \) and covariance \( \sigma^2 \mathbf{I} \). The parity vector does make the assumption that \( p_p \) and \( x \) are independent and that the noise \( w \) is of zero mean allowing for \( p_p \) to be of zero mean when no faults are present. The inner product

\[
D = p_p^T p_p = (Pz)^T (Pz) = z^T P^T P z = z^T S z
\]

can be used as a test statistic for fault detection, where \( S = P^T P \). The decision variable \( D \) has a chi-square distribution based upon the distribution of the elements of the parity and fault vectors. In the event that there is a fault, the distribution for \( D \) will become a non-central chi-square distribution allowing for a threshold test to be used to indicate whether or not a fault has occurred, i.e.,

\[
H_0 : D < \gamma \quad \text{(no fault)}
\]

\[
H_1 : D > \gamma \quad \text{(fault)}
\]

This decision variable is subjected to a dual hypothesis test where \( H_0 \) represents no fault and \( H_1 \) indicates a fault. This is done by comparing \( D \) with a threshold \( \gamma \), which is based upon a desired probability of false alarm \( P_{fa} \), number of redundant measurements \( m - n \), and the covariance of the measurement noise \( \sigma^2 \). The performance of the test statistic is characterized by the probability of false alarm \( (P_{fa}) \) and the probability of missed detection \( (P_{md}) \). The operating characteristics of the test statistic can be obtained by plotting \( P_{md} \) vs. \( P_{fa} \) for various parameter combinations.
\[ P_{fa} = P[D > \gamma|H_0] \]
\[ P_{md} = P[D < \gamma|H_1] \]

Assuming that all measurement faults are equally likely, the hypothesis test is characterized by

\[ P_{fa} = P[D > \gamma|b = 0] \]
\[ P_{md} = \frac{1}{m} \sum_{i=1}^{m} P[D < \gamma|b = b_i] . \]

If there is no fault, \( b = 0 \), then \( E[p_p] = 0 \) and \( D/\sigma^2 \) has a chi-squared distribution with \( m - n \) degrees-of-freedom. Therefore,

\[ P_{fa} = Q(\gamma/\sigma^2|m - n) = 1 - P(\chi^2|r) \]

where \( Q(\chi^2|r) = 1 - P(\chi^2|r) \) and \( P(\chi^2|r) \) is the chi-squared distribution function given by

\[ P(\chi^2|\sigma) = \left[ 2^{r/2} \Gamma \left( \frac{r}{2} \right) \right]^{-1} \int_0^{\chi^2} t^{r/2-1} e^{-t/2} dt. \]

The probability of false alarm, \( P_{fa} \), depends on the number of redundant measurements \( m - n \), and is independent of the measurement geometry \( H \). Therefore, the required threshold can be found as a function of \( P_{fa}, m - n, \) and \( \sigma^2 \) as

\[ \gamma = f \left( P_{FA}, m - n, \sigma^2 \right) = \sigma^2 Q^{-1} (P_{FA}|m - n) , \]

where \( Q^{-1}(P|r) \) is the inverse of \( Q(\chi^2|r) \).
Given that there is a fault, then \( b = b_i \), \( E[p_p] = Pb_i \), and \( D/\sigma^2 \) is a random variable from a non-central chi-squared distribution with \( m - n \) degrees of freedom and non-centrality parameter

\[
\theta_i = \left( \frac{b^2}{\sigma^2} \right) S_{ii}
\]

where \( S_{ii} \) is the \( i^{th} \) diagonal element of \( S = P^T P \). When written as a distribution, this yields

\[
P_{md} = \frac{1}{m} \sum_{i=1}^{m} P \left( \frac{\gamma}{\sigma^2} | m - n, \frac{b^2}{\sigma^2} S_{ii} \right),
\]

(3.4)

where \( P(\chi^2| r, \theta) \) is the non-central chi-square probability function.

### 3.3 Least Square Residuals

Work done by [4,13,32,35,42] laid the foundation for GPS integrity monitoring using least-square residuals. Least squares residuals makes the same assumption as parity space in that there are redundant measurements available making the system overdetermined with the number of measurements \( m \) exceeding the number of states \( n \) such that \( m - n \geq 1 \). For GPS, \( n = 4 \) since the states being solved for include the three dimensional position and a clock bias term. In the original work that developed the least-square residual method, the measurement equation was given in terms of the pseudorange:

\[
\rho_i = d_i - \begin{bmatrix} e_i^T & 1 \end{bmatrix} x - \epsilon_i
\]

(3.5)

where \( \rho_i \) is the pseudorange of the \( i^{th} \) satellite, \( d_i \) is the distance between the user and the \( i^{th} \) satellite, \( e_i \) is the unit vector from the user to the \( i^{th} \) satellite, \( x \) is the state vector including position and the clock bias term, and \( \epsilon_i \) is normally distributed measurement noise with mean \( \mu_i \)
and variance $\sigma_i^2$. The vector representation of the measurement equation is [42]:

$$z = d - \rho = \begin{bmatrix} e_i^T & 1 \\ \vdots & \vdots \\ \vdots & \vdots \\ e_m^T & 1 \end{bmatrix} x + \epsilon = Hx + \epsilon \quad (3.6)$$

The least-square estimate is expressed as:

$$\hat{x}(H^TH)^{-1}H^Tz = \bar{H}z \quad (3.7)$$

with the estimated measurement is given as:

$$\hat{z} = H\hat{x} = H\bar{H}z \quad (3.8)$$

The difference between the actual measurements and the predicted measurements yields the vector of residual errors:

$$\hat{\epsilon} = z - \hat{z} = (I - H\bar{H})z \quad (3.9)$$

Substituting equation (3.6) for $z$ yields the following:
\[
\hat{\epsilon} = (I - HH)(Hx + \epsilon) \quad (3.10)
\]
\[
= Hx - HHHx + \epsilon - HH\epsilon
\quad (3.11)
\]
\[
= H(I - HH)x + (I - HH)\epsilon
\quad (3.12)
\]
\[
= H(I - (H^TH)^{-1}H^TH)x + (I - HH)\epsilon
\quad (3.13)
\]
\[
= H(I - I)x + (I - HH)\epsilon
\quad (3.14)
\]
\[
= (I - HH)\epsilon
\quad (3.15)
\]

The sum of squares error (SSE) is defined as the inner product of \(\hat{\epsilon}^T\hat{\epsilon}\) and makes a useful test statistic. In the case where the noise has a zero mean (no fault bias), the SSE has a chi-square distribution just like the decision variable from the parity vector method. With \(m - 4\) degrees of freedom the test statistic used is:

\[
r = \sqrt{\frac{\hat{\epsilon}^T\hat{\epsilon}}{m - 4}} = \sqrt{\frac{SSE}{m - 4}} \quad (3.16)
\]

In the event that the measurements are affected by a non-zero mean in the noise, the test statistic will come from a non-central chi-square distribution with a non-centrality parameter \(\lambda\) given as [6]:

\[
\lambda = \left(\frac{\rho_{bias}}{\sigma}\right)^2 \quad (3.17)
\]

The non-central chi-square distribution cannot be expressed in closed form, but can be approximated using numeric integration [6, 42]. The integrity checking process is performed by comparing the test statistic \(r\) to a threshold \(\gamma\). The threshold \(\gamma\) is generated through Monte Carlo simulation and is selected based on desired false alarm and missed detection requirements.
3.4 Slope Method

The slope method is used in conjunction with either the parity vector or least-squares methods, and is useful in relating the test statistic to the error in position caused by a biased measurement. The ”slope” is the ratio between the horizontal position error and the test statistic. For GPS, the ”slope” is a linear relationship that approximates the effect of a growing pseudorange bias and its effect on the horizontal position error. For any given satellite, the ”slope” is a function between the horizontal position space and the parity space [63] and is given by

\[
\text{Slope}_i = \frac{\sqrt{\bar{H}_{1,i}^2 + \bar{H}_{2,i}^2}}{\sqrt{S_{i,i}}} \quad (3.18)
\]

where \( S = P^T P \). This method assumes that there is no noise and considers the effect of the bias only; therefore, the parity vector is \( p = P(b + 0) \). The resulting horizontal position error for the \( i^{th} \) measurement is [63]:

\[
H_{ei} = \text{Slope}_i \| P \| \quad (3.19)
\]

The slope of each measurement will be different based upon the measurement geometry. A worst case upper bound can be estimated using the measurement with the largest slope and is given as [63]:

\[
|H_{bias}| = \max_{i=1:m} [\text{Slope}_i] \| p \| \quad (3.20)
\]

Since the process is not truly deterministic, and there is measurement noise in the system, this estimate does not reflect the true horizontal position error. However, the measurement noise is assumed to be zero-mean additive white gaussian noise (AWGN), making the deterministic method a reasonable method for approximating the expected value of the parity vector [51, 55]. The slope method allows for the projection of position error onto the parity space allowing for
easy visualization of thresholds relative to protection levels.
Current Vision Integrity Monitoring

The area of vision navigation can be subdivided into two categories: methods based on tracking known features and methods based on tracking unknown features. Consequently, the measurements need to be treated differently when performing integrity monitoring. Figure 4.1 shows a break down of the areas for integrity monitoring of vision systems.

Figure 4.1: Chart showing the areas of Vision Integrity Monitoring.

There has been little research done in the area of integrity monitoring for vision navigation.
systems. The beginning framework was laid out by Larson in [31], [29], and [30]. In these works, GPS integrity monitoring methods using parity vectors and slope were introduced to vision navigation systems. This work focuses on detecting a single pixel pair error relating to a known feature as seen in Figure 4.1. The following is a summary of that work.

4.1 Parity Vector and Slope

This section includes a derivation of the work done in [31], [29], and [30]. This derivation is provided in the same manner as it was in [29]. In that work, four assumptions were made:

- Tracked features are known and do not need to be estimated.
- An image-based measurement is considered a two element set, consisting of an (x,y) pair.
- The bias is multidimensional in that it is a magnitude times sinusoidal components of the angle of the error in the x and y directions.
- Noise is assumed to be zero mean additive white gaussian noise.

This derivation makes use of the fact that the x and y elements of a pixel pair are measurements linked by a single observation and hold adjacent positions i and j in the measurement vector. The components of the bias vector $b$ are $b_i = \|b\| \sin \theta$ and $b_j = \|b\| \cos \theta$, where $\|b\|$ is the magnitude of the pixel error and $\theta$ is the angle of the error in the x-y pixel frame. The slope method described in [6] is a ratio of the square vector norm of the horizontal position error $\|\delta x\|^2$ and the square vector norm of the parity vector $\|p\|^2$ or residual vector if the residual method is used. If using the parity vector method, the resulting relationship is given by

$$\frac{\|\delta x_h\|^2}{\|p\|^2} = \frac{\delta x^T \delta x}{p^T p} = \frac{b^T \tilde{H}^T \tilde{H} b}{b^T \tilde{P}^T \tilde{P} b}$$

(4.1)
with \( \bar{H} = (H^T H)^{-1} H^T \), which is the Moore-Penrose pseudo-inverse of \( H \) and \( P \) is the parity matrix described in the previous section. The subscript \( h \) on \( \delta x_h \) and \( \bar{H}_h \) indicates that it only includes the horizontal position elements of \( \delta x \) and corresponding rows of \( \bar{H} \). Equation (4.1) can be simplified using \( G = \bar{H}_h^T \bar{H}_h \) and \( S = P^T P \) as

\[
\frac{\|\delta x\|^2}{\|p\|^2} = \frac{b^T G b}{b^T S b} \tag{4.2}
\]

Following the assumption that there is only one error and making the bias vector zero for all elements except \( b_i \) and \( b_j \), the numerator of the ratio is given by

\[
b^T G b = b_i^2 G_{ii} + b_i b_j (G_{ij} + G_{ji}) + b_j^2 G_{jj} \tag{4.3}
\]

Since \( G \) is symmetric, i.e. \( G_{ij} = G_{ji} \), and substituting the sinusoidal definitions of \( b_i \) and \( b_j \) yields

\[
b^T G b = \|b\|^2 \sin^2(\theta) G_{ii} + 2\|b\|^2 \sin(\theta) \cos(\theta) G_{ji} + \|b\|^2 \cos^2(\theta) G_{jj} \tag{4.4}
\]

Now using the double angle identity for sine, \( \sin(2\theta) = 2 \sin(\theta) \cos(\theta) \), yields the final form of the numerator as

\[
\|\delta x_h\|^2 = \|b\|^2 \left[ \sin^2(\theta) G_{ii} + \sin(2\theta) G_{ij} + \cos^2(\theta) G_{jj} \right] \tag{4.5}
\]

The denominator is found in a similar manner taking advantage of the symmetry of \( S \):

\[
\|p\|^2 = \|b\|^2 \left[ \sin^2(\theta) S_{ii} + \sin(2\theta) S_{ji} + \cos^2(\theta) S_{jj} \right] \tag{4.6}
\]

Canceling the bias term in the numerator and denominator results in the following expression
\[ \frac{\| \delta x \|}{\| p \|} = \left[ \frac{\sin^2(\theta) G_{ii} + \sin(2\theta) G_{ji} + \cos^2(\theta) G_{jj}}{\sin^2(\theta) S_{ii} + \sin(2\theta) S_{ji} + \cos^2(\theta) S_{jj}} \right]^{\frac{1}{2}} \quad (4.7) \]

Using the pythagorean identity, \( \cos^2(\theta) = 1 - \sin^2(\theta) \), the expression can be rewritten in terms of sine only as

\[ \frac{\| \delta x \|}{\| p \|} = \left[ \frac{\sin^2(\theta)(G_{ii} - G_{jj}) + \sin(2\theta) G_{ij} + G_{jj}}{\sin^2(\theta)(S_{ii} - S_{jj}) + \sin(2\theta) S_{ij} + S_{jj}} \right]^{\frac{1}{2}} \quad (4.8) \]

Larson showed the use of a slope method, whereby the decision variable \( D = p^T p \) is related to error in horizontal position as a ratio. This method is useful in estimating the effect of a bias in a measurement on the horizontal position for setting up a detection threshold. This method is also useful in the event that there are multiple measurement errors, but becomes less accurate as the number of errors increases. Further analysis can be seen in [29].

69
Detection and Isolation Multiple Random Errors in Vision Systems

The research summarized in section 4 [29], [30], and [31] converted the GPS integrity monitoring techniques using parity space and slope to vision measurements and provided a framework common to both navigation systems. However, the assumptions made for GPS are not applicable for vision systems. In GPS systems, it is highly unlikely that there will be more than one bad measurement at a time. The GPS constellation is closely monitored and robust. In the case of vision measurements, there is a much higher likelihood of having more than one bad measurement. The previous research does not address this possibility. The previous work also is only focused on the detection of an error and does not address isolating the error. Section 5.1, summarizes the test statistic discussed in the previous section, which is designed to detect a bad measurement. This method is then used in section 5.2 as a test to determine if a subset of data is good and an iterative method is used to isolate the bad measurements in the data.

5.1 Test Statistic

In [31], [29], and [30], Larson used the slope method from GPS integrity monitoring, whereby the decision variable \( D = p^T p \) is related to error in horizontal position as a ratio of the squared vector norm of both the horizontal position error and the parity vector:
\[
\frac{\|\delta x\|^2}{\|p\|^2} = \frac{\delta x^T \delta x}{p^T p} = \frac{b^T G b}{b^T S b} \tag{5.1}
\]

with \( G = \bar{H}_h^T \bar{H}_h \), \( S = P^T P \), and \( \bar{H} = (H^T H)^{−1} H^T \), which is the Moore-Penrose pseudo-inverse of \( H \). Assuming that the bias vector \( b \) is zero except for the \( b_i \) and \( b_j \) components (corresponding to a bias in a singular set of pixel coordinates with corresponding error magnitude and direction \( \theta \)), equation (5.1) can be written as (For full details regarding the derivation see [29]):

\[
\frac{\|\delta x\|^2}{\|p\|^2} = \left[ \frac{\sin^2(\theta)(G_{ii} - G_{jj}) + \sin(2\theta)G_{ij} + G_{jj}}{\sin^2(\theta)(S_{ii} - S_{jj}) + \sin(2\theta)S_{ij} + S_{jj}} \right]^{\frac{1}{2}} \tag{5.2}
\]

### 5.2 Bayes Algorithm for Isolating Corrupt Measurements

The algorithm for isolating faulty measurements is based on Bayes’ Rule given by equation (5.3) and discussed in many books on probability and statistics [8, 49].

\[
P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{\infty} P(B|A_j)P(A_j)} \tag{5.3}
\]

When the complete set of measurements fails the test described by section 5.1, it is assumed that there is at least one faulty measurement in the set and that each measurement is equally likely to have the error. Therefore, all the elements of vector \( \vec{P} \) represent the probability of error in each of the measurements and are initialized as \( 1/m \) where \( m \) is the number of measurements.

Multiple random subsets of data are created from the original set and tested. If they pass, equation (5.5) is used to update the corresponding elements of \( \vec{P} \) related to the measurements in the subset. If they fail, equation (5.4) is used to update the corresponding elements of \( \vec{P} \). After several tests on different subset combinations of the measurements, \( \vec{P} \) converges, given a high enough probability of having a passing subset of data.
\[
\bar{P}\{\text{Error} = 1 | \text{Alarm}\}(k + 1) = \frac{P_{md}\bar{P}(k)}{\sum(P_{md}\bar{P}(k)) + P_{fa}\bar{P}_e} \quad (5.4)
\]

\[
\bar{P}\{\text{Error} = 1 | \text{Pass}\}(k + 1) = \frac{P_{md}\bar{P}(k)}{\sum(P_{md}\bar{P}(k)) + P_{md}\bar{P}_e} \quad (5.5)
\]

where \( P_{md} \) is the probability of a missed detection, \( P_{fa} \) is the probability of false alarm for the test and \( P_e \) is the probability of an error existing in the subset measurements given the subset of \( \bar{P} \) and a bar over a probability is the reverse, \( \bar{P}_e = 1 - P_e \).

The probability of obtaining a random subset of data that passes is based on a hypergeometric distribution given by

\[
P(X = x | N, M, n) = \frac{\binom{M}{x} \binom{N - M}{n - x}}{\binom{N}{n}} \quad (5.6)
\]

with \( N \) as the total number of measurements, \( n \) as the number of samples in a subset for testing, \( M \) as the number of bad measurements, and \( x \) as the number of bad measurements in a subset.

Assuming that it only takes one bad measurement to result in a failed test, the probability of a passing subset is given by
\[ P(\text{GoodSet}) = P(X = 0|N, M, n) \]  
\[ = \binom{N - M}{n} \binom{N}{n} \]  

Figure 5.1 shows a plot of the probability of passing the test relative to the number of bad measurements given that five measurements are taken at a time for testing.

### 5.3 Results

This algorithm is demonstrated using a 50 run Monte Carlo experiment. Each experiment is performed with a total of 100 measurements and a varying number of bad measurements. Figure 5.2 shows that the sum of \( \bar{P} \) converges to 4.5 after 100 iterative tests. After convergence, all five erroneous measurements can be isolated without any false positives relating to other measurements. It should be noted that the algorithm does not need to run to full convergence to isolate the bad measurements, but it can be run with fewer iterations assuming that every measurement has been included in a test. However, if it has not converged, the likelihood of isolating good measurements is decreased, but may be acceptable as a trade off for computation time required for additional tests.

As the number of bad measurements is increased, the sum of \( \bar{P} \) increases as seen in Figures 5.3 and 5.4 show that \( \sum \bar{P} \) converging to 8 and 10.8, with 10 and 15 bad measurements, respec-
If many more experiments are performed and the steady-state value of $\sum \vec{P}$ is plotted relative to the number of bad measurements in Figure 5.5, a non-linear relationship is seen. This relationship is similar to the probability of passing a test vs. the number of bad measurements as seen in Figure 5.1.

A linear relationship is found when the steady-state value of $\sum \vec{P}$ is plotted relative to the probability of selecting a good subset as seen in Figure 5.6. This relationship varies depending on the total number of measurements $N$ and the number of measurements in a testing subset $n$ but remains linear. This provides a simple tool to uncover the number of bad measurements.
Figure 5.2: Average sum of the error probability vector $\vec{P}$ vs. test iteration, given a 50 run Monty Carlo experiment with 5 of 100 measurements bad.
Figure 5.3: Average sum of the error probability vector $\vec{P}$ vs. test iteration, given a 50 run Monty Carlo experiment with 10 of 100 measurements bad.
Figure 5.4: Average sum of the error probability vector $\vec{P}$ vs. test iteration, given a 50 run Monty Carlo experiment with 15 of 100 measurements bad.
Figure 5.5: Steady-state average sum of the error probability vector $\vec{P}$ vs. the number of bad measurements out of 100 total measurements.
Figure 5.6: Steady-state average sum of the error probability vector $\vec{P}$ vs. the probability of getting a good subset of data with 100 total measurements and a varying number of bad measurements.
It is possible to determine the value that $\sum \bar{P}$ will converge to given a set number of bad measurements, total number of measurements, and size of the subset that is tested at random. The values of $\bar{P}$ associated with good measurements converge to zero. If there is only one bad measurement, the value associated with it will converge to one, but this is not true when there are multiple errors. Starting with the equation used when there is an alarm we have

$$\bar{P}\{\text{Error} = 1|\text{Alarm}\}(k + 1) = \frac{\bar{P}_{md}\bar{P}(k)}{\sum(\bar{P}_{md}\bar{P}(k)) + P_{fa}\bar{P}_e}. \quad (5.9)$$

As the process iterates, the value for $\bar{P}_e$, which is the probability that there is no error, goes to zero. This simplifies equation (5.9) to yield

$$\bar{P}\{\text{Error} = 1|\text{Alarm}\}(k + 1) = \frac{\bar{P}_{md}\bar{P}(k)}{\sum(\bar{P}_{md}\bar{P}(k))}. \quad (5.10)$$

Looking at the value of a single element of $\bar{P}(k)$, which corresponds to a bad measurement yields

$$P\{\text{Error} = 1|\text{Bad}\} = \frac{\bar{P}_{md}P\{\text{Error} = 1|\text{Bad}\}}{\sum(\bar{P}_{md}\bar{P}\{\text{Error} = 1|\text{Alarm}\})}. \quad (5.11)$$

The scalar value, $\bar{P}_{md}$ cancels so that

$$P\{\text{Error} = 1|\text{Bad}\} = \frac{P\{\text{Error} = 1|\text{Bad}\}}{\sum(\bar{P}\{\text{Error} = 1|\text{Alarm}\})}. \quad (5.12)$$

It is assumed that all bad measurements are equally likely to be detected and that the final value of $\bar{P}(k)$ that corresponds to each individual bad measurement will be the same. Therefore, the denominator will contain an integer multiple of the value $P\{\text{Error} = 1|\text{Bad}\}$. There is assumed to be guaranteed one bad measurement if the alarm is triggered. Additional bad measurements in the tested subset are assumed to occur with a probability described by the hypergeometric distribution. For example, the probability that there would be two errors given an alarm is
\[ P(X = 1, N - 1, M - 1, k - 1) = \binom{M - 1}{1} \binom{N - M}{k - 2} \binom{N}{k - 1} \]  

(5.13)

where \( N \) is the size of the measurement population, \( M \) is the number of bad measurements in the population, and \( k \) is the number of measurements used in the test. It is assumed that one measurement is bad and this represents the likelihood of selecting only one more bad measurement. If this is applied to equation (5.12), it becomes

\[
P\{\text{Error} = 1|\text{Bad}\} = \frac{P\{\text{Error} = 1|\text{Bad}\}}{P\{\text{Error} = 1|\text{Bad}\}(1 + \sum_{x=1}^{k-1}(x \ast P(X = x, N - 1, M - 1, k - 1)))}
\]  

(5.14)

The \( P\{\text{Error} = 1|\text{Bad}\} \) terms cancel yielding

\[
P\{\text{Error} = 1|\text{Bad}\} = \frac{1}{1 + \sum_{x=1}^{k-1}(x \ast P(X = x, N - 1, M - 1, k - 1))}
\]  

(5.15)

Given that the good measurement converge to zero and the bad measurements converge to \( P\{\text{Error} = 1|\text{Bad}\} \), the sum of \( \bar{P}\{\text{Error} = 1\} \) is

\[
SSoP = \frac{M}{1 + \sum_{x=1}^{k-1}(x \ast P(X = x, N - 1, M - 1, k - 1))}
\]  

(5.16)

Figure 5.7 shows a plot comparing the values that the sum of \( \bar{P}\{\text{Error} = 1\} \) converge to compared with values predicted using equation (5.16). The result is a close match. This makes it possible to estimate the number of bad measurements after several iterations of the test.
Figure 5.7: Predicted and simulated steady-state average sum of the error probability vector $\vec{P}$ vs. the number of bad measurements out of 100 total measurements.
Dilution of Precision for Vision Systems

This section introduces the concept of Dilution of Precision (DOP) for vision systems. For GPS, the DOP is a measure of the quality of the measurement geometry, relating how an error in measurement is either amplified or attenuated when it is mapped to the solution. The DOP is very important for integrity monitoring, since high DOP, which corresponds to a bad geometry, indicates that the solution may be of poor quality and may conceal the existence of and amplify the effect of bad measurements. The following is derived in a manner similar to DOP for GPS systems.

6.1 Derivation of Dilution of Precision for Vision Systems

To calculate the DOP of a vision system, it is assumed that \( \delta Z_{pix} \) and \( H \) are calculated, using a given camera model, feature positions, and estimate of the camera’s position as seen in equations (2.123)-(2.126). It is also assumed that the errors, \( \delta x \) and \( \delta Z_{pix} \), are normal random variables with zero mean. As with GPS, solving for the error in the state vector, \( \delta x \), by least-squares yields

\[
\delta x = (H^T H)^{-1} H^T \delta Z_{pix}.
\]

The error covariance of \( x \) is, therefore, given by
\[ E \left\langle (\delta x)(\delta x)^T \right\rangle = E \left\langle \left[ (H^T H)^{-1} H^T \delta Z_{\text{pix}} \right] \left[ (H^T H)^{-1} H^T \delta Z_{\text{pix}} \right]^T \right\rangle \]
\[ = (H^T H)^{-1} H^T E \left\langle \delta Z_{\text{pix}} \delta Z_{\text{pix}}^T \right\rangle H \left[ (H^T H)^{-1} \right]^T. \] (6.1)

The measurement covariance is assumed to be uncorrelated between different sets of pixel coordinates with variance \( \sigma^2 \). Each measurement is a set of pixel coordinates with both a horizontal measurement and a vertical measurement. Although it is likely that errors in the \( x \)-coordinate and \( y \)-coordinate will occur together, they are still orthogonal and uncorrelated. Therefore,

\[ E \left\langle (\delta Z_{\text{pix}})(\delta Z_{\text{pix}})^T \right\rangle = \sigma^2 I_6 \] (6.2)

Substituting (6.2) into (6.1) yields

\[ E \left\langle (\delta x)(\delta x)^T \right\rangle = \sigma^2 (H^T H)^{-1} H^T H \left[ (H^T H)^{-1} \right]^T \]
\[ = \sigma^2 \left[ (H^T H)^{-1} \right]^T = \sigma^2 (H^T H)^{-1}, \]

which relates noise in measurements to noise in the state vector. The dilution of precision is found in the diagonal elements of the dilution of precision matrix.
\[
G = (H^T H)^{-1} = \begin{bmatrix}
G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\
G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \\
G_{31} & G_{32} & G_{33} & G_{34} & G_{35} & G_{36} \\
G_{41} & G_{42} & G_{43} & G_{44} & G_{45} & G_{46} \\
G_{51} & G_{52} & G_{53} & G_{54} & G_{55} & G_{56} \\
G_{61} & G_{62} & G_{63} & G_{64} & G_{65} & G_{66}
\end{bmatrix}.
\]

(6.3)

The dilution of precision matrix, \( G \), is different for vision systems. Since the state error vector for a vision system contains both 3D position and orientation, the types of DOP are different. In comparison to GPS, vision systems have no time state, but rather, they contain three states for orientation, which include roll, pitch, and yaw. The different types of DOP available are

\[
GDOP = \sqrt{G_{11} + G_{22} + G_{33} + G_{44} + G_{55} + G_{66}}
\]

(6.4)

\[
PDOP = \sqrt{G_{11} + G_{22} + G_{33}}
\]

(6.5)

\[
HDOP = \sqrt{G_{11} + G_{22}}
\]

(6.6)

\[
VDOP = \sqrt{G_{33}}
\]

(6.7)

\[
ODOP = \sqrt{G_{44} + G_{55} + G_{66}}
\]

(6.8)

where \( GDOP \) is the geometric dilution of precision, \( PDOP \) is the position dilution of precision, \( HDOP \) is the horizontal dilution of precision, \( VDOP \) is the vertical dilution of precision, and \( ODOP \) is the orientation dilution of precision. They represent the navigation solution sensitivities to feature measurement errors in pixel coordinates. Despite the fact that \( GDOP \) can be calculated, it has no practical purpose, due to the different units among the states.
In GPS systems, the DOP is unitless. Both the measurement and states are distances evaluated in meters. Even the clock bias is calculated in meters, but remains easily converted to seconds using the speed-of-light. This has the convenience of making the units of all GPS DOP terms \( \text{m/m} \). Since vision systems can be used to calculate both orientation and position, which have units of radians and meters respectively. This fact combined with the measurement units being in pixels results in a complete mismatch between DOP terms. Therefore, the \( GDOP \) term is a combination of different terms and will be dominated by only one or two of those terms and have no meaningful units.

In vision systems, the DOP has units and can only be used with separate terms for position and orientation. \( PDOP \), \( VDOP \), and \( HDOP \) have the units \( \text{m/pixel} \), and \( ODOP \) has the units \( \text{rad/m} \). These terms serve the same use as in GPS systems. For a given variance in the measurements \( \sigma^2 \), the variance in the navigation states representing horizontal position can be found by multiplying the measurement variance by \( HDOP \).

In GPS systems, horizontal and vertical DOP have clear meaning. Looking back at the measurement model described in equations (2.120)-(2.126), the measurements are relative to pixels in an image and the navigation states are relative to a navigation frame. In this case, horizontal is referring to errors relative to the position in x- and y-direction in the navigation frame, and vertical is referring to errors relative to the position state in the z-direction in the navigation frame. Since the navigation frame can be either local or global and have a variety of orientations, it is important to note that these terms are relative to the navigation frame as it is defined for a system.

### 6.2 Effect of DOP on Integrity Monitoring Performance

For Receiver Autonomous Integrity Monitoring (RAIM) algorithms, the probability of missed detection of a fault is a function of multiple factors, which include the number of measurements,
Finding this probability of missed detection is useful in determining if the integrity of the navigation solution can be assured to be within specifications given the performance of the system and the measurement geometry.

In the event of poor measurement geometry, the DOP becomes large and the navigation system performance degrades. A similar effect occurs with RAIM algorithms. Such algorithms include parity space [43, 51] or least squares residual [4, 13, 32, 35, 42] methods. Detection of a fault in either method is done with hypothesis testing and the test statistics from both methods are equivalent. [5]

When navigation measurement geometry is varying, the relationship between measurement errors and the navigation states is also varying. Investigating the effect of errors on navigation states based on the geometry of measurements was first considered by Brown and Sturza [3]. The following is based on their initial work. The equations have been rederived and modified for use in vision-based navigation.

Given that
\[
\tilde{z} = z + b = Hx + b
\]
where \(\tilde{z}\) is the measurement, \(z\), with a bias, \(b\), added, then the least square approximation of the state vector \(\hat{x}\) is
\[
\hat{x} = (H^T H)^{-1} H^T \tilde{z} = GH^T \tilde{z} = x + \tilde{H}_1 b.
\]
where \((H^T H)^{-1}\) is the dilution of precision matrix \(G\), and \(\tilde{H} = GH^T\). The error in the state
estimate is calculated by subtracting the true state yielding

\[
\delta x = \hat{x} - x = \bar{H}b_i = \bar{h}_ib_i = G^{T\bar{H}}b_i
\]

where \( b_i \) represents the \( i^{th} \) value of the error vector \( b \) corresponding to the measurement failure, and \( h_i \) and \( \bar{h}_i \) represent the \( i^{th} \) row and column of the measurement matrix \( H \) and \( \bar{H} \), respectively. With this it is possible to calculate the n-dimensional state error due to the single error \( b_i \) yielding

\[
GE^2 = E \left[ \delta x^T \delta x \right] = \bar{h}_i^T\bar{h}_ib_i^2. \tag{6.9}
\]

Although \( GE \) has no practical use in vision systems due to the combination of dissimilar units, it is shown for completeness in reference to works done in GPS. Similar expressions are found for the spherical position error (\( SPE \)), the horizontal position error (\( HPE \)), and the orientation error (\( OE \)) given by equations (6.10), (6.11) and (6.12), respectively. Unlike \( GE \), these terms are useful in determining the effect of the geometry on detection of errors in the measurements that result in navigation state errors, which exceed tolerances.

\[
SPE^2 = (\bar{H}_1^2 + \bar{H}_2^2 + \bar{H}_3^2) b_i^2 \tag{6.10}
\]

\[
HPE^2 = (\bar{H}_1^2 + \bar{H}_2^2) b_i^2 \tag{6.11}
\]

\[
OE^2 = (\bar{H}_4^2 + \bar{H}_5^2 + \bar{H}_6^2) b_i^2 \tag{6.12}
\]

Given the expressions for the error in equations (6.10), (6.11), and (6.12), the probability of missed detection (\( P_{md} \)) as a function of \( SPE, HPE \) or \( OE \) is given as
\[ P_{md} = \begin{cases} \frac{1}{m} \sum_{i=1}^{m} P \left( \frac{\gamma}{\sigma^2} | m - n, \frac{SPE^2}{\sigma^2} \left( \frac{S_{ii}}{(H_{1i}^2 + H_{2i}^2 + H_{3i}^2)} \right) \right) ; & \text{for Spherical Position Error} \\
\frac{1}{m} \sum_{i=1}^{m} P \left( \frac{\gamma}{\sigma^2} | m - n, \frac{HPE^2}{\sigma^2} \left( \frac{S_{ii}}{(H_{1i}^2 + H_{2i}^2)} \right) \right) ; & \text{for Horizontal Position Error} \\
\frac{1}{m} \sum_{i=1}^{m} P \left( \frac{\gamma}{\sigma^2} | m - n, \frac{OE^2}{\sigma^2} \left( \frac{S_{ii}}{(H_{4i}^2 + H_{5i}^2 + H_{6i}^2)} \right) \right) ; & \text{for Orientation Error} \end{cases} \]

In the cases of poor measurement geometry, the performance of integrity monitoring techniques degrades and large navigation errors can occur without detection. The following is an introduction to a set of integrity geometry parameters: \( \Delta P_i, \Delta H_i, \text{and} \Delta O_i \). These parameters are similar to the ones used for GPS RAIM \([3,7,52]\) and can be used to determine the effectiveness of RAIM-style integrity monitoring algorithms for detecting failures in vision navigation systems.

If \( G \) is the DOP matrix and \( G_i \) is the DOP matrix with the \( i^{th} \) measurement omitted, then

\[
G_i = G + \frac{G_h h_i^T G}{1 - h_i^T G h_i}
\]

where \( h_i \) is the \( i^{th} \) row of the measurement matrix \( H \) corresponding to the omitted measurement.

The DOP can then be calculated from \( G_i \) given the measurement geometry without the \( i^{th} \) measurement.

The difference in DOP is represented by the parameters \( \Delta P_i, \Delta H_i, \text{and} \Delta O_i \) such that
\[
\Delta P_i^2 = \text{PDOP}_i^2 - \text{PDOP}^2 = \frac{H_{1i}^2 + H_{2i}^2 + H_{3i}^2}{S_{ii}} 
\]  
(6.14)

\[
\Delta H_i^2 = \text{HDOP}_i^2 - \text{HDOP}^2 = \frac{H_{1i}^2 + H_{2i}^2}{S_{ii}} 
\]  
(6.15)

\[
\Delta O_i^2 = \text{ODOP}_i^2 - \text{ODOP}^2 = \frac{H_{4i}^2 + H_{5i}^2 + H_{6i}^2}{S_{ii}} 
\]  
(6.16)

If horizontal position in the navigation frame is the primary concern, then \( \Delta H_i \) would be used. The least detectable measurement failure corresponds to the largest value of \( \Delta H_i \); therefore, the worst case is obtained by using

\[
\Delta H_{\text{max}} = \text{MAX} (\Delta H_i). 
\]

However, if the interest is in the probability of missed detection and all measurement faults are equally likely, then equations (6.14)-(6.16) can be substituted into equation (6.13) yielding equation (6.17).

\[
P_{md} = \begin{cases} 
\frac{1}{m} \sum_{i=1}^{M} P \left( \frac{\gamma}{\sigma^2} \left| m - n, \left( \frac{\text{SPE}}{\Delta P_i \sigma} \right)^2 \right| \right); & \text{for Spherical Position Error} \\
\frac{1}{m} \sum_{i=1}^{M} P \left( \frac{\gamma}{\sigma^2} \left| m - n, \left( \frac{\text{HPE}}{\Delta H_i \sigma} \right)^2 \right| \right); & \text{for Horizontal Position Error} \\
\frac{1}{m} \sum_{i=1}^{M} P \left( \frac{\gamma}{\sigma^2} \left| m - n, \left( \frac{\text{OE}}{\Delta O_i \sigma} \right)^2 \right| \right); & \text{for Orientation Error} 
\end{cases} 
\]  
(6.17)

This relationship shows the connection between the DOP and the performance of RAIM style integrity monitoring algorithms, allowing for performance prediction of integrity monitoring for vision navigation systems.
6.3 Numeric Examples

As in GPS, dilution of precision is a good measure for the quality of the measurement geometry. In
the following example, the geometry is ideal with several well distributed reference points with an
average distance from the references, \( s_c = 1000 \text{m} \) and a focal distance \( f = 100 \text{m} \). These points
can be seen in the local navigation frame in Figure 6.3(a) and in the pixel frame in Figure 6.3(b).
The resulting matrix \((H^TH)^{-1}\) is given by equation (6.18).

\[
(H^TH)^{-1} = \begin{bmatrix}
0.6233 & 0.1095 & 0.005 & 0.000 & 0.000 & 0.000 \\
0.1095 & 0.3227 & 0.001 & 0.000 & 0.000 & 0.000 \\
0.005 & 0.001 & 0.000 & 0.000 & 0.000 & 0.000 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\
\end{bmatrix}
\]

(6.18)

Figure 6.1: Example of Good Measurement Geometry (1000m): Measurements in the camera
frame with pixel coordinates.
Figure 6.2: Example of Good Measurement Geometry (1000m): Target features (blue) and vehicle location (red) in local navigation frame.

The resulting position, horizontal, vertical, and orientation dilution of precision are given by

\[
PDOP = 0.9726 \frac{m}{\text{pixel}} \tag{6.19}
\]
\[
HDOP = 0.9726 \frac{m}{\text{pixel}} \tag{6.20}
\]
\[
VDOP = 7.6589 \times 10^{-4} \frac{m}{\text{pixel}} \tag{6.21}
\]
\[
ODOP = 7.6105 \times 10^{-4} \frac{\text{rad}}{\text{pixel}} \tag{6.22}
\]

It is interesting to note that the error is focused in the horizontal position. The vertical position is less affected by the noise in the optical measurements (pixel coordinates). This is due to the orientation of the camera. The fact that the camera is pointed downward, relative to the navigation frame, results in excellent \( VDOP \). The distance in the z-axis of the camera frame is measured by the scale of the image, whereas the measurement of distance in the x- and y-axis of the camera are measured by the translation of the image features. The \( ODOP \) is lower in value for two reasons,
one is that it is independent of distance to features. As the distance to features increases, the error in position increases, but not orientation. The other cause is the difference in units. One radian of orientation error is far worse than one meter of position error.

The following is a presentation of three different cases to show how the DOP changes depending on measurement geometry starting with Case 1, which is from the previous example.

**Case 1**

In this first case, the geometry is ideal with several well-distributed reference points with an average distance from the references, \( s_c = 1000m \), and focal distance \( f = 100m \). These points can be seen in the local navigation frame in Figure 6.3(a) and in the pixel frame in Figure 6.3(b). The DOP for this case is expected to be low indicating a good measurement geometry.
Case 2

In the second case, the geometry is still ideal with several well-distributed reference points with respect to the pixel frame. However, the target features are farther away with an average distance from the references, $s_c^z = 5000\text{m}$. These points can be seen in the local navigation frame in Figure 6.4(a) and in the pixel frame in Figure 6.4(b). The DOP for this case is still expected to be low indicating a good measurement geometry; however, they will be higher given the increased range from the target references.

Case 3

In the third and final example case, The geometry is poor with the measurements clustered and farther away with an average distance $s_c^z = 5000\text{m}$. These points can be seen in the local navigation frame in Figure 6.5(a) and in the pixel frame in Figure 6.5(b). This geometry is expected to produce a high DOP indicating that it is a poor measurement geometry.
(a) Measurements in the camera frame with pixel coordinates.

(b) Target features (blue) and vehicle location (red) in local navigation frame.

Figure 6.5: Case 3: Bad measurement geometry with a larger distance from the targets (5000m).

Results

Table 6.1 summarizes the DOP for all three of the example cases. Both the first and second case had good measurement geometry and consequently had low DOP, but it is important to note that the effect of the increase in distance to the references can be seen in the DOP. The third case had poor measurement geometry, and the corresponding effect on the DOP is obvious.

<table>
<thead>
<tr>
<th>DOP</th>
<th>Example Case 1</th>
<th>Example Case 2</th>
<th>Example Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDOP</td>
<td>0.9726</td>
<td>3.7077</td>
<td>79.8148</td>
</tr>
<tr>
<td>HDOP</td>
<td>0.9726</td>
<td>3.7077</td>
<td>79.8148</td>
</tr>
<tr>
<td>VDOP</td>
<td>$7.6589 \times 10^{-4}$</td>
<td>$7.6397 \times 10^{-4}$</td>
<td>0.0206</td>
</tr>
<tr>
<td>ODOP</td>
<td>$7.6105 \times 10^{-4}$</td>
<td>$7.6283 \times 10^{-4}$</td>
<td>0.0160</td>
</tr>
</tbody>
</table>

Table 6.1: DOP for example cases.

In regard to integrity monitoring performance, it is possible to compare the three cases by looking at the Detector Operating Characteristics (DOC), which is a plot of the probability of false alarm ($P_{fa}$) vs. the probability of missed detection ($P_{md}$) seen in Figures 6.6, 6.7, and 6.8. In all cases, there is a clear trade off between the $P_{fa}$ and $P_{md}$. Each figure plots three different navigation error to measurement noise ratios ($ENR = GE/\sigma$). Larger ENR results in a curve
with lower $P_{md}$ and $P_{fa}$. In Figures 6.6 and 6.7, the examples of good geometry, the ENR is 2, 4, and 6 from top to bottom. The integrity monitoring performance is clearly better when the measurements are made at a lower altitude with the targets closer, than at a high altitude. For the bad geometry case in Figure 6.8, the ratios are increased because the geometry is bad enough to make errors of smaller size impractical to detect. Even with the increased ENR, the bad geometry case has worse performance relative to the good geometry.

![Graph showing detector operating characteristics (DOC) for good geometry at 1000m.](image)

**Figure 6.6: Detector operating characteristics (DOC) for good geometry at 1000m.**

### 6.4 Summary

In summary, this chapter developed the dilution of precision (DOP) concept for vision systems. Vision systems differ from GPS in that they are not based on range measurements, but rather pixel coordinates in an image. This measurement has information regarding the orientation of the vehicle, but lacks the time component and results in the existence of the orientation dilution of precision
Figure 6.7: Detector operating characteristics (DOC) for good geometry at 5000m.

Figure 6.8: Detector operating characteristics (DOC) for bad geometry at 5000m.
(ODOP). The effect of measurement noise on orientation is low; however, in the event of a bias or camera parameter fault, the effect on orientation could still be considerable.

In the examples above, the camera was assumed to be pointing down and tracking features below. This resulted in most of the DOP to be from the horizontal position states, indicating that they are the most effected by the noise in the measurements. Given the DOP relationships, it is possible to layout known features, predict performance, and optimize for a low DOP, improving the performance of the navigation system and integrity monitoring of that system.
Detection of Systematic Errors in Vision Systems

Previous methods make use of redundant information within an image to detect faults within measurements made from the image. This redundancy is ideal for identifying random measurement errors; however, if all the measurements have a common bias, they will be in agreement with each other and the error will go undetected. This chapter develops a method to detect and isolate systematic errors in a camera system. As with other integrity monitoring techniques, the method described in this chapter relies on redundant measurements. The method shown utilizes measurements made by additional cameras. With two cameras, it is possible to detect a systematic error, but three or more cameras are needed to isolate the bad camera. The following section develops a test statistic to detect systematic errors, by comparing measurements seen by two cameras. The relationship between different alignment errors and the test statistic are also shown.

7.1 Test Statistic Development

Figure 7.1 shows an exaggerated separation of two camera frames to make the geometry easier to see. It is expected that all the cameras are mounted on the same sensor platform with their position and orientation known relative to each other.

Using the coordinate system shown in Figure 7.1, the following vectors can be defined:
Figure 7.1: Target to image transformation geometry with two camera frames.

\[
\begin{align*}
\mathbf{p}_{21}^c &= C_b^c C_n^b (\mathbf{p}_2^n - \mathbf{p}_1^n) \\
\mathbf{p}_{12}^c &= C_b^c C_n^b (\mathbf{p}_1^n - \mathbf{p}_2^n) \\
\mathbf{s}_{12}^c &= \mathbf{s}_2^c - \mathbf{p}_{12}^c \\
\mathbf{s}_{12}^c &= \mathbf{s}_1^c - \mathbf{p}_{21}^c
\end{align*}
\]

The vector \( \mathbf{p}_{12}^c \) is the position of the first camera relative to the second in camera frame \( c2 \), and \( \mathbf{p}_{21}^c \) is the position of the second camera relative to the first in camera frame \( c1 \). \( \mathbf{s}_{12}^c \) and \( \mathbf{s}_{21}^c \) represent the target location in terms of the coordinate frame origin of the other camera, respectively. From these equations, it is possible to take a pixel coordinate from one camera and map it to the other. Assuming that the cameras have overlapping views, they should see the same targets, and the pixel coordinates in the first camera should correspond to the pixel coordinates of the second camera given the geometry relating the two camera frames.

The pixel coordinates in the first and second camera frame for the targets seen by both camera
and camera 2 are given by

\[
\begin{align*}
\mathbf{s}_{1}^{c_1\text{pix}} &= \frac{1}{s_{c_1}^{z}} \mathbf{T}_{c_1}^{\text{pix}} \mathbf{s}_{1}^{c_1} \\
\mathbf{s}_{2}^{c_2\text{pix}} &= \frac{1}{s_{c_2}^{z}} \mathbf{T}_{c_2}^{\text{pix}} \mathbf{s}_{2}^{c_2}
\end{align*}
\]  

(7.5)  

(7.6)

where \(s_{1}^{c_1\text{pix}}\) is a measurement made by the first camera. To compare this measurements with the same measurement made by the second camera, \(s_{2}^{c_2\text{pix}}\), they need to be mapped to the other camera’s pixel coordinate frame. The first step is to rotate the vector \(\mathbf{s}_{1}^{c_2}\) in equation (7.3) so that it is in the camera frame \(c_1\). This is done by premultiplying by the direction cosine matrix, \(\mathbf{C}_{c_2}^{c_1} = \mathbf{C}_{n}^{c_1} \mathbf{C}_{c_2}^{n}\) such that

\[
\mathbf{s}_{1}^{c_1} = \mathbf{C}_{c_2}^{c_1} \mathbf{s}_{1}^{c_2} = \mathbf{C}_{c_2}^{c_1} (\mathbf{s}_{2}^{c_2} - \mathbf{p}_{12}^{c_2}).
\]  

(7.7)

Then \(\mathbf{s}_{1}^{c_1}\) is converted to pixel coordinates by substituting equation (7.7) into equation (7.5) which results in

\[
\mathbf{s}_{1}^{c_1\text{pix}} = \frac{1}{s_{c_1}^{z}} \mathbf{T}_{c_1}^{\text{pix}} \mathbf{C}_{c_2}^{c_1} (\mathbf{s}_{2}^{c_2} - \mathbf{p}_{12}^{c_2}).
\]  

(7.8)

The position of the object measured by camera 2 in its own coordinate system can be obtained from equation (7.6) as

\[
\mathbf{s}_{2}^{c_2} = \mathbf{s}_{2}^{c_2\text{pix}} \mathbf{T}_{c_2}^{\text{pix}} \mathbf{s}_{2}^{c_2\text{pix}}.
\]  

(7.9)

Substituting equation (7.9) into equation (7.8) yields

\[
\mathbf{s}_{1}^{c_1\text{pix}} = \frac{1}{s_{c_1}^{z}} \mathbf{T}_{c_1}^{\text{pix}} \mathbf{C}_{c_2}^{c_1} (\mathbf{s}_{2}^{c_2\text{pix}} \mathbf{T}_{c_2}^{\text{pix}} \mathbf{s}_{2}^{c_2\text{pix}} - \mathbf{p}_{12}^{c_2}).
\]  

(7.10)
Following the same procedure, the measurements from Camera 1 can be mapped to Camera 2 as

\[
\mathbf{s}_{1}^{c_{1}pix} = \frac{s_{c_{1}}^{c_{2}}}{{s_{c_{2}}}^{c_{1}}} \mathbf{T}_{c_{1}}^{pix} C_{c_{2}}^{c_{1}} T_{c_{2}}^{pix} \mathbf{s}_{2}^{c_{2}pix} - \frac{1}{{s_{c_{1}}}^{c_{2}}} T_{c_{1}}^{pix} C_{c_{2}}^{c_{1}} \mathbf{p}_{12}^{c_{1}}
\]  

(7.11)

In the following equations measurements are designated by a tilde (\(\tilde{s}\)) and transformed measurements from another camera are designated with an accent (\(\hat{s}\)). The difference between the measurement from Camera 1, \(\tilde{s}_{1}^{c_{1}pix}\), and the transformed measurements from Camera 2, \(\hat{s}_{2}^{c_{2}pix}\) is given by

\[
D = \tilde{s}_{1}^{c_{1}pix} - \hat{s}_{2}^{c_{2}pix}
\]  

(7.13)

The sum of squared differences, \(SSD\), between the measurements from Camera 1 and the transformed measurements from Camera 2, can be used as a metric for determining the disagreement between the two cameras. If either one is not aligned properly or is out of calibration, the SSD, will be increased and is given by

\[
SSD = \sum D^{2}
\]  

(7.14)

For the purposes of relating the test statistic to camera alignment errors or scaling errors, the test statistic chosen is

\[
\beta = \sqrt{SSD} = \sqrt{\sum D^{2}}
\]  

(7.15)

Taking the square root of the SSD has an effect on the distribution of the test statistic; SSD has a Chi-squared distribution and \(\beta\) has a Chi distribution. Using a Chi distribution allows for a linear
relationship between the types of errors and the mean of the test statistic. This is discussed in detail in the following section.

### 7.2 Relationship between Test Statistic and Errors

Figures 7.2 and 7.3 provide an example of the test statistic, used to detect a small systematic error in the form of an alignment error. Figure 7.2 shows the location of the sensor platform relative to the features that it is tracking visually. The projection of those features into measurements can be seen in Figure 7.3. The features were designed to provide good measurement geometry.

![Figure 7.2: Sensor platform and feature locations. The platform location is red and the features for tracking are blue.](image)

Figure 7.4 shows the resulting projection when a bias is introduced and is mapped to the second camera. If the bias did not exist, the red and blue dots, which represent measurements would
Figure 7.3: Image projection in pixel frame of tracked targets as seen by the camera on the sensor platform.
be at the same locations. This translation of the measurements is a result of a small misalignment in one of the cameras.

Figure 7.4: Image projection of features given that an unwanted bias is introduced (red) and the correct projection (blue).

Figure 7.5 shows the results of a 1000 run Monte Carlo experiment both with and without a misalignment present. Without a misalignment present, the result is a chi-squared distribution. When there is a misalignment, the result is shifted and can be easily distinguished from the other distribution. The width of these distributions is dependent upon the performance of the sensors used. The better the quality of the sensor, the smaller the error that can be detected.

Figure 7.6 shows a plot of the SSD versus an alignment error. The SSD test statistic increases quadratically with respect to the alignment error (angle) and has a chi-squared distribution. This clearly shows the relationship between the misalignment and the test statistic; however, there is a more convenient test statistic that can be used. Figure 7.7 shows a plot of the square root of the
Figure 7.5: Distribution of the Sum of Squared Differences, $SSD$, given a 1000 run Monte Carlo experiment with (red) and without (blue) a systematic error.
SSD with respect to the misalignment. This test statistic has a linear relationship, rather than a quadratic relationship, relative to the error and is referred to as $\beta$. The variance is constant rather than increasing as is with the chi-squared distribution.

\[ \chi^2 \text{ Test Statistic v. Y-Axis Angle Error} \]

![Graph showing SSD (chi-squared) test statistic vs. y-axis angle error.](image)

Figure 7.6: Plot comparing the SSD ($\chi^2$ test statistic) and a roll (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.

The mean of the test statistic can be calculated for both SSD (chi-squared) and $\beta$ (chi). For chi-squared distributions, the mean is defined as

\[ \mu = k + \lambda \] (7.16)

where $k$ is the degrees-of-freedom (two for each pixel coordinate pair) and $\lambda$ is the non-centrality
Figure 7.7: Comparison of $\beta$ ($\chi$ test statistic) and a roll (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.
parameter.

\[ \lambda = \sum_{i=1}^{k} \left( \frac{\mu_i}{\sigma_i} \right)^2 \]  \hspace{1cm} (7.17)

With the test statistics defined, it is possible to detect errors in both misalignment and scaling due to changes in the camera parameters and alignment. Each type of error has a different effect on the test statistic. For the purposes of comparing different errors, the \( \beta \) test statistic is used. Using the same non-centrality parameter as seen in equation (7.17), the mean of the \( \beta \) test statistic is

\[ \mu = \sqrt{k + \lambda} \]  \hspace{1cm} (7.18)

In the following section, the relationship between the \( \beta \) test statistic and misalignments about each axis of the camera as well as scaling error are explored.

### 7.2.1 X-Axis Misalignment Error (Pan)

If the camera has a misalignment, which results in a rotation about the x-axis of the camera frame, the image will be panned making the measurements shift in the x-direction. An example of this is shown in Figure 7.8. The result of 10,000 simulations with varying alignment errors about the x-axis is shown in Figure 7.9. Regardless of the measurement geometry, it is possible to determine the effect of a misalignment error by perturbing equations (7.11) and (7.12). However, for most applications, the misalignment errors will be small or easily detectable by other means. For both small misalignment angles and for measurements based on targets that are far away (distance to target is in excess of focal distance, which is common in aviation applications), all pixel targets will translate an equal distance.
Figure 7.8: Example of feature locations on an image with a pan (x-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.
Figure 7.9: Comparison of the test statistic and a pan (x-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the angle error.
The mean of the test statistic is calculated using equation (7.18). The non-centrality parameter, \( \lambda \), for the example in Figures 7.8 and 7.9, is given by

\[
\lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2.
\]  

(7.19)

where \( b_i \) is the bias in the location of each pixel measurement due to the misalignment and \( k = 10 \).

In the case of a small angle, the bias is a function of the misalignment angle, the resolution of the camera and the horizontal viewing angle of the camera. The ratio of the bias (displacement of the feature locations in pixels) to the resolution of the camera is approximately the same as the ratio of the misalignment angle and the viewing angle such that

\[
\frac{b_x}{N} = \frac{\alpha}{\text{viewing angle}}
\]  

(7.20)

\[
b_x = \frac{N \alpha}{\text{viewing angle}}
\]  

(7.21)

where \( b_x \) is a displacement in the x-direction, \( N \) is the horizontal camera resolution, and \( \alpha \) is the angle error. Since this misalignment only causes displacement in the x-direction, \( \lambda \) is found by substituting \( b_x \) from equation (7.21) as half of the values represented by \( b_i \) in equation (7.19) such that

\[
\lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2 = k \left( \frac{N \alpha}{\sigma(\text{viewing angle})} \right)^2.
\]  

(7.22)

Given the camera geometry discussed in Section 2.9.1, the horizontal view angle is found using geometry as seen in Figure 7.10, where \( f \) is the focal length, \( W \) is the width of the image plane in the x-direction, and \( \theta \) is half of the viewing angle. Using right-triangle geometry, shown in Figure 7.10, the angle \( \theta \) can be written as
\[ \theta = \tan^{-1}\left(\frac{W}{2f}\right) \]  

(7.23)

and the viewing angle of the camera in the horizontal direction is

\[ \text{viewing angle} = 2\theta = 2\tan^{-1}\left(\frac{W}{2f}\right). \]  

(7.24)

Therefore, the mean of the test statistic is given by

\[ \mu = \sqrt{k + \frac{k}{2} \left(\frac{N\alpha}{2\sigma \tan^{-1}\left(\frac{W}{2f}\right)}\right)^2}. \]  

(7.25)

In Figure 7.9, the mean was plotted as a yellow line on top of the results of the simulation with various angle errors. The sensitivity of this test statistic to rotation errors about the x-axis of the camera is highly dependent on the number of measurements. The more features present, the higher the effect on the test statistic. It should be noted that in this example all the features remained within the image. If a feature is near the edge of the image, or an angle is large, it is possible to lose features for comparison, which is also a good indicator of a fault.

### 7.2.2 Y-Axis Misalignment Error (Tilt)

If the camera has a misalignment that results in a rotation about the y-axis of the camera frame, the image will be tilted making the measurements shift in the y-direction. An example of this is shown in Figure 7.11. The result of 10,000 simulations with varying alignment errors about the y-axis is shown in Figure 7.12. As before, it is possible to determine the effect of a misalignment error by perturbing equations (7.11) and (7.12). However, for most applications, the misalignment errors will be small or easily detectable by other means. For both small misalignment angles and for
measurements based on targets that are far away (distance to target is in excess of focal distance, which is common in aviation applications), all pixel targets will translate an equal distance.

For the example in Figures 7.11 and 7.12, the test statistic relationship is similar to a misalignment about the x-axis. Equations (7.18) and (7.19) are used to calculate the mean and non-centrality parameter, $\lambda$, with $k = 10$. In the case of a small angle the bias is a function of the misalignment angle, the resolution of the camera and the vertical viewing angle of the camera. The ratio of the bias (displacement of the feature locations in pixels) to the resolution of the camera is approximately the same as the ratio of the misalignment angle and the viewing angle such that

$$\frac{b_y}{M} = \frac{\alpha}{\text{viewing angle}}$$  \hspace{1cm} (7.26)

$$b_y = \frac{M\alpha}{\text{viewing angle}}$$  \hspace{1cm} (7.27)
Figure 7.11: Example of feature locations on an image with a tilt (y-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.
Figure 7.12: Comparison of the test statistic and a tilt (y-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.
where $b_y$ is a displacement in the $y$-direction, $M$ is the vertical camera resolution, and $\alpha$ is the angle error. Since this misalignment only causes displacement in the $y$-direction, $\lambda$ is found by substituting $b_y$ from equation (7.27) as half of the values represented by $b_i$ in equation (7.19) such that

$$\lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2 = \frac{k}{2} \left( \frac{M\alpha}{\sigma(\text{viewing angle})} \right)^2$$

(7.28)

Given the camera geometry discussed in Section 2.9.1, the horizontal view angle is found using geometry as seen in Figure 7.13, where $f$ is the focal length, $H$ is the height of the image plane in the $y$-direction, and $\theta$ is half of the viewing angle. Using right-triangle geometry,

$$\theta = \tan^{-1} \left( \frac{H}{2f} \right)$$

(7.29)

and the viewing angle of the camera in the vertical direction is

$$\text{viewing angle} = 2\theta = 2 \tan^{-1} \left( \frac{H}{2f} \right).$$

(7.30)

The resulting mean for the test statistic is

$$\mu = \sqrt{\frac{k}{2} + \frac{k}{2} \left( \frac{M\alpha}{2\sigma \tan^{-1} \left( \frac{H}{2f} \right)} \right)^2}.$$ 

(7.31)

In Figure 7.11, the mean is plotted as a yellow line on top of the results of the simulation with various angle errors. The sensitivity of this test statistic to rotation errors about the $y$-axis of the camera is highly dependent on the number of measurements. The more features present, the higher the effect on the test statistic. It should be noted that in this example all the features remained within the image. As with the previous case, any feature near the edge of the image can be lost for comparison depending on the magnitude of the misalignment. Any loss of features may
also be an indicator of a fault.

![Figure 7.13: Geometry for relating the vertical view angle, focal length, and height of the image plane.](image)

7.2.3 Z-Axis Misalignment Error (Rotation)

If the camera has a misalignment that results in a rotation about the z-axis of the camera frame, the image will be rotated making the measurements rotate around the center of the image. An example of this is shown in Figure 7.15. The result of 10,000 simulations with varying alignment errors about the z-axis is shown in Figure 7.14. As before, it is possible to determine the effect of a misalignment error by perturbing equations (7.11) and (7.12). However, the misalignment errors will be small or easily detectable by other means. Starting with the mean of the \( \beta \) test statistic

\[
\mu = \sqrt{k + \lambda}. \tag{7.32}
\]
Figure 7.14: Example of feature locations on an image with a rotation (z-axis) misalignment error. Blue markers represent the correct location and red markers represent the measured location.
Figure 7.15: Comparison of the test statistic and a rotation (z-axis) misalignment error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Angle Error.
For the example in Figures 7.15 and 7.14, there are ten degrees-of-freedom \( k = 10 \) (two for each measurement pair) and \( \lambda \) is

\[
\lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2.
\]  

(7.33)

where \( b_i \) is the bias in the location of each pixel measurement due to the misalignment. Since the feature locations are rotating around the center of the image, features near the center have a smaller bias than features located near the center of the image. Consequently, the bias in a feature location due to the rotation is a function of the distance of the feature from the center of the image and the angle the image was rotated. Figure 7.16 displays the displacement, \( \delta \) of the feature as it is rotated by an angle \( \theta \) around the center of the image with a distance \( r \) from the center of the image. Since all pixel coordinates are positive integers, the distance of the \( i^{th} \) feature from the center of the image is

\[
r = \sqrt{\left( x_i - \frac{N}{2} \right)^2 + \left( y_i - \frac{M}{2} \right)^2}.
\]  

(7.34)

where \((x_i, y_i)\) is the location of the \( i^{th} \) feature, \( N \) is the horizontal resolution, and \( M \) is the vertical resolution. The bias in the features location can be found using simple right triangles yielding

\[
\sin \left( \frac{\alpha}{2} \right) = \frac{\delta/2}{r}.
\]  

(7.35)

Solving for \( \delta \) yields

\[
\delta = 2r \sin \left( \frac{\alpha}{2} \right).
\]  

(7.36)

Since the non-centrality parameter, \( \lambda \), is the sum of the biases in both x- and y-directions squared \((b_x^2 \text{ and } b_y^2)\), it is unnecessary to separate \( \delta \) into its components because
\[ \delta^2 = b_x^2 + b_y^2 \]  
\[ \lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2 = \sum_{i=1}^{k/2} \left( \frac{2r_i \sin(\alpha)}{\sigma_i} \right)^2 \]  
(7.38)

The resulting mean for the test statistic is given by

\[ \mu = \sqrt{k + \sum_{i=1}^{k/2} \left( \frac{2r_i \sin(\alpha)}{\sigma_i} \right)^2}. \]  
(7.39)

In Figure 7.15 the mean was plotted as a yellow line on top of the results of the simulation with various angle errors. The sensitivity of this test statistic to rotation errors about the z-axis of the camera is highly dependent on the number of measurements and their distance to the center of the image. Features near the center of the image contribute very little to the increase in the test statistic compared to features near the edge.

### 7.2.4 Scaling Error (Focus/Zoom)

If the camera has a fault that results in a scaling error, the features in the image will either move toward or away from the center of the image. An example of this is shown in Figure 7.18. The result of 10,000 simulations with varying scale factors is shown in Figure 7.17. It is possible to determine the effect of a scaling error by perturbing equations (7.11) and (7.12). However, the effect on the test statistic is easily calculated by other means. Equations (7.18) and (7.19) are used to calculate the mean and non-centrality parameter, \( \lambda \), with \( k = 10 \).

Since the feature locations moving either toward or away from the center based on the per-
Figure 7.16: Geometry for determining the displacement of a feature location resulting from image rotation.

Figure 7.17: Example of feature locations on an image with a scaling error. Blue markers represent the correct location and red markers represent the measured location.
Figure 7.18: Comparison of the test statistic and a scaling error. Blue ‘x’ represent simulated results and the yellow line represents the predicted mean value for the test statistic given the Scaling Error.
percentage of the scaling factor, features near the center will be near the vanishing point of the image and move very little, and features near the outer edge of the image will move more. Consequently, the bias in a feature location due to the scaling effect is a function of the distance of the feature from the center of the image and the percentage of the scaling effect. The distance, \( r \), between the feature and the center is calculated using equation (7.34), and the displacement in the features location is found with simple multiplication such that

\[
\delta = r \times \text{scale}\%.
\]  

(7.40)

Since the non-centrality parameter, \( \lambda \), is the sum of the biases in both x- and y-directions squared \( (b_x^2 \text{ and } b_y^2) \), this makes it unnecessary to separate \( \delta \) into its components because, as before,

\[
\delta^2 = b_x^2 + b_y^2
\]  

(7.41)

and \( \lambda \) can be written as

\[
\lambda = \sum_{i=1}^{k} \left( \frac{b_i}{\sigma_i} \right)^2 = \sum_{i=1}^{k/2} \left( \frac{r \times \text{scale}\%}{\sigma_i} \right)^2.
\]  

(7.42)

The resulting mean for the test statistic is given as

\[
\mu = \sqrt{k + \sum_{i=1}^{k/2} \left( \frac{r \times \text{scale}\%}{\sigma_i} \right)^2}.
\]  

(7.43)

In Figure 7.18 the mean was plotted as a yellow line on top of the results of the simulation with various angle errors. The sensitivity of this test statistic to scaling effects is highly dependent on the number of measurements and their distance to the center of the image. Features near the center of the image contribute very little to the increase of the test statistic compared to features near the edge.
Conclusion

This dissertation focuses on understanding existing integrity monitoring methods and using them for the detection of multiple errors in vision-based navigation systems, as well as, developing a technique for detecting systematic errors due to camera misalignment and scaling. This chapter first presents a discussion of key contributions of this research. This is then followed by a discussion of future related research and a brief summary.

8.1 Discussion

The research in this dissertation established methods for the detection and isolation of multiple errors in vision-based navigation systems. Prior to this dissertation, there has been only a preliminary effort by researchers to create a framework for research in the area of vision-based navigation systems. That framework was created by adapting methodology from GNSS integrity monitoring for use in vision systems. This adaptation came with a mathematical model relating a test statistic to the horizontal position error. However, this work also maintained the assumption from GNSS integrity monitoring that there will only be one error at any moment in time. This is a good assumption for GNSS systems with a closely monitored satellite network, but it is far less likely in a vision-based system, which can have errors associated with bad image registration or camera calibration.
The three key contributions of this dissertation addressed those issues. The first contribution, developed a method to isolate multiple independent bad measurements. This method was a Bayes-inspired algorithm for detecting and isolating multiple bad measurements. The second contribution of this dissertation was the development of the Dilution of Precision (DOP) for vision-based navigation systems, which was linked to the performance of the integrity monitoring system. DOP for vision systems is similar to DOP for GPS systems and can be used to determine if the system can detect the fault.

The third and final contribution of this dissertation was the development of a method to detect systematic errors in a vision-based navigation system. The earlier methods created a test statistic that looks for agreement in the measurements from an image. If one of the measurements indicates something different than the others, it sets off an alarm; however, if all measurements are at fault due to a systematic error, the error will not be detected. The method developed in this dissertation looks for agreement between two separate vision-based systems. If there is disagreement, then the test statistic increases above the threshold and an alarm is sounded. The subsequent analysis showed the effect of misalignment errors and scale errors with regard to the test statistic. It was found that both misalignment error and scale errors can be detected. This method is independent of the physical location of the features, but is dependent upon the location of the measurements in the image/pixel frame. Consequently, this method can be used for features at both known and unknown locations.

8.2 Future Work

This dissertation developed methods for detecting multiple measurement errors in vision-based navigation systems. As mentioned earlier, GNSS integrity has been an active research area since the advent of the GPS system more than twenty years ago. This dissertation made significant ad-
Advances to the state-of-the-art in this relatively new area. To advance this area further, constraints used in this research should be removed. Some of these constraints and subsequent research avenues are described in this section, detailing future work possibilities for integrity monitoring in vision-based navigation systems.

This dissertation and other work done by [29] focused on an epoch-by-epoch least-squares approach to vision-based navigation. This works well for identifying errors in measurements. For systems that integrate both a vision and inertial navigation systems, there has not been any research done relating the navigation error and a test statistic. As mentioned earlier in this dissertation, work done in [60] uses vision to aid an inertial navigation system. To perform integrity monitoring on this type of system, further research is needed. This future research direction may use a filter approach, using an Extended Kalman Filter (EKF), similar to an approach used in GPS aided inertial systems, or possibly use a particle filter.

This dissertation makes the assumption that the error in measurements is independent and gaussian. For many camera systems and image registration methods, the uncertainty in measurements is dependent upon the location of the feature in the image frame. This effects the integrity methods developed in this dissertation as well as the DOP. Future work can be done to integrate additional knowledge of the uncertainty into the measurement noise model.

The work in this dissertation made the assumption that observed objects used for measurements are at known locations. This assumption is acceptable for applications such as formation flying, mid-air refueling, and automated landing systems, where features of interest are on a nearby aircraft or markers on a runway. Future work could investigate the impact of measurements at unknown (estimated) locations. This would make the integrity monitoring system more versatile allowing for its use in more environments.
The work in this dissertation is theoretical and does not address the amount of work needed to apply this to a working navigation system. With GNSS systems such as GPS, the system is well defined. Vision systems can include a variety of systems, which use different navigation techniques. Without defined specifications, developing integrity monitoring techniques will remain difficult. Once a system is designed, it will be possible to design integrity monitoring algorithms, which address the needs of that system, as has been done for GPS.

8.3 Summary

Vision-aided inertial navigation systems have shown the necessary performance needed to be a back-up for GPS. However, such a system will need an integrity monitoring system before it is trusted for critical operations. Work has already been completed that converts techniques used in GPS integrity monitoring for use in vision systems. These methods create a framework to begin work in the area of integrity monitoring for vision systems but are not designed to detect and isolate multiple random or systematic errors in vision systems. This research analyzed and further developed methods for detecting and isolating multiple measurement errors. This dissertation also developed a method to detect systematic errors, which were undetectable by existing techniques. The ultimate goal of this research was to further the area of integrity monitoring for vision systems, so it could eventually be used as a trusted system and as a back-up for GPS navigation.
Bibliography


