The thesis of Zachary Hall was reviewed and approved* by the following:

Puneet Singla  
Associate Professor of Aerospace Engineering  
Thesis Advisor

David B. Spencer  
Professor of Aerospace Engineering

Amy R. Pritchett  
Professor of Aerospace Engineering  
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.
Abstract

The uncertain Lambert problem and computation of reachability sets are in many ways complimentary problems with important applications in Space Situational Awareness (SSA). Formulating the solution to these problems in a stochastic framework is an important intermediate step to enabling the next generation of collision assessment, satellite tracking and characterization capabilities. Traditionally, accurately computing solutions to these problems in a probabilistic manner was either computationally infeasible due to the large number of simulations required to capture the statistical properties of the solution, or was only accurate near the nominal solution due to the inability of linear variational methods to capture higher order statistical moments of the solution. Using a Taylor series expansion of the deterministic solution, a polynomial approximation method is proposed to calculate the higher order sensitivity matrices. An orthogonal polynomial basis function expansion is used to represent the solution of the uncertain Lambert and reachability problems. The coefficients of these orthogonal polynomials represent the higher order sensitivity matrices. The Conjugate Unscented Transformation (CUT) method is utilized for the purposes of computing the multi-dimensional expectation integrals required to determine the least squares coefficients, and offers extensive computational savings compared to other quadrature methods. Additionally, the CUT method allows the computation of these sensitivity matrices without the cumbersome need to explicitly take higher order partial derivatives, as well as having the advantage of achieving higher accuracy over a larger input domain than was previously available using other numerical integration methods. The mathematical framework for the polynomial approximation method is laid out and test case simulations for both the uncertain Lambert problem and the reachability set problem are presented. Discussion of results, as well as the relative advantages and limitations of the method for each simulation is included.
# Table of Contents

List of Figures vi

List of Tables viii

Acknowledgments ix

Chapter 1
   Introduction 1
      1.1 Background ............................................. 1
      1.2 Thesis Outline ........................................... 3

Chapter 2
   Mathematical Background 5
      2.1 The Taylor Series Expansion ............................. 5
      2.2 The Least Squares Method ............................... 9
      2.3 Orthogonal Polynomials .................................. 12
      2.4 Numerical Integration Methods ............................ 15
         2.4.1 The Monte Carlo Method ............................. 16
         2.4.2 Gaussian Quadrature ................................. 18
      2.5 Example Higher Order Sensitivity Problem ............... 22
         2.5.1 Alternatives to Gaussian Quadrature ................. 24

Chapter 3
   The Uncertain Lambert Problem 27
      3.1 Problem Background ...................................... 27
      3.2 Simulation Description .................................... 28
         3.2.1 Simulation Results .................................. 31
         3.2.2 Test Case 1: LEO .................................... 35
         3.2.3 Test Case 2: GTO .................................... 39
# List of Figures

2.1 Taylor Series Approximations for \( \sin(x) \) .............................. 7  
2.2 Inner Product of 2D Vectors ................................................. 13  
2.3 Error in Monte Carlo Approximation vs Number of MC Points ...... 17  
2.4 Polynomial Approximation for \( N = 50 \) and Varying \( d \) ............ 22  
2.5 Polynomial Approximation for \( d = 7 \) and Varying \( N \) ............... 23  
2.6 A Schematic of CUT Axes .................................................... 25  
2.7 Comparison of Quadrature Schemes for Same Order of Accuracy ... 25  
3.1 Uncertain Lambert Problem Diagram ....................................... 30  
3.2 Polynomial Approximation Method Summary ............................ 31  
3.3 Error in Expectation Values of \( x \) vs. \( N \) ............................... 33  
3.4 Plot of Coefficient Order of Magnitude for Varying Approximation Orders ................................................................. 36  
3.5 Test Case 1: \( V_1 \) Spatial Representation ................................ 37  
3.6 Test Case 1: \( V_1 \) Mahalanobis Distance Representation .............. 38  
3.7 Plot of Coefficient Order of Magnitude for Varying Approximation Orders ................................................................. 39  
3.8 Test Case 2: \( V_1 \) Spatial Representation ................................ 40  
3.9 Test Case 2: \( V_1 \) Mahalanobis Distance Representation .............. 41  
4.1 Satellite Reachability Problem Diagram .................................... 45  
4.2 Diagram of Maneuver Geometry VNC Frame .............................. 46  
4.3 Test Case 1: Nominal Orbit .................................................. 47  
4.4 Test Case 2: Diagram of Arrival Geometry ............................... 48  
4.5 Test Case 2: Nominal LOI Maneuver ..................................... 50  
4.6 Test Case 1: 4th Order Reachability Set Over Time .................... 53  
4.7 Test Case 1: Spatial Position Error Plot .................................. 54  
4.8 Test Case 1 Mahalanobis Distance Position Error .................... 55  
4.9 Test Case 2 LNL: 4th Order Reachability Set Over Time ................ 56  
4.10 Test Case 2 LNL: Spatial Position Error Plot .......................... 57
4.11 Test Case 2 LNL: Mahalanobis Distance Position Error Plot . . . . 58
4.12 Test Case 2 HNL: 4th Order Reachability Set Over Time . . . . . 59
4.13 Test Case 1 HNL: Spatial Position Error Plot . . . . . . . . . . . 59
4.14 Test Case 1 HNL: Mahalanobis Distance Position Error Plot . . . 60
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Table of Common Orthogonal Polynomials</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>Test Case 1 LEO State Vector</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Test Case 2: GTO State Vector</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>Statistical Moments Calculated Using CUT8</td>
<td>34</td>
</tr>
<tr>
<td>3.4</td>
<td>Test Case 1: Covariance $\Sigma$ Using CUT8 Method</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Test Case 1: 1(^{st}) Order Approximation Coefficients</td>
<td>35</td>
</tr>
<tr>
<td>3.6</td>
<td>Test Case 1: Initial Velocity Error</td>
<td>38</td>
</tr>
<tr>
<td>3.7</td>
<td>Test Case 1: Initial Velocity Error</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>Test Case 1: Nominal Transfer Orbit Parameters</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Test Case 1: Nominal Burn Parameters</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Nominal Maneuver Uncertainties</td>
<td>47</td>
</tr>
<tr>
<td>4.4</td>
<td>Test Case 2: Nominal Orbit Elements</td>
<td>49</td>
</tr>
<tr>
<td>4.5</td>
<td>Test Case 2: Input Uncertainty</td>
<td>50</td>
</tr>
<tr>
<td>4.6</td>
<td>Simulation Uncertainties</td>
<td>52</td>
</tr>
<tr>
<td>4.7</td>
<td>Test Case 1: Error Summary</td>
<td>53</td>
</tr>
<tr>
<td>4.8</td>
<td>Test Case 2 LNL: Error Summary</td>
<td>56</td>
</tr>
<tr>
<td>4.9</td>
<td>Test Case 2 HNL: Error Summary</td>
<td>58</td>
</tr>
</tbody>
</table>
Acknowledgments

I would like to recognize and thank those who have helped me throughout my academic career, without whom this thesis would not be possible. First, I would like to express my thanks and deepest gratitude to Dr. Puneet Singla for providing constant support and guidance throughout my first few years as a graduate student. Working as advisee to Dr. Singla has been an incredible learning experience and a true pleasure due to his depth of understanding, enthusiasm for engineering, and willingness to help his students day or night. I would also like to extend thanks to Dr. David Spencer and Dr. Amy Pritchett for reviewing and providing feedback on my thesis. My sincere thanks must also be extended to the Air Force Research Laboratory (AFRL), and the Science Mathematics and Research for Transformation (SMART) scholarship program for funding my graduate studies.

Additionally, I would like to thank the other members of the Control and Analysis of Stochastic Systems (CASS) Laboratory. It has been an invaluable help to have colleagues and friends who are always willing to provide advice and a fresh perspective on our mutual research interests. Furthermore, I would like to thank all of the professors and students affiliated with the Astrodynamics Research Group of Penn State (ARGoPS) for providing an academically stimulating environment and sense of community within our department.

Lastly, I would like to extend thanks to my friends and family. To Emily; Thank you for your constant love and support and for providing me happiness throughout my graduate career. To my friends; Thank you all for the good times we’ve had, and for all the good times to come. To my brother Nate; Thank you for being my confidant, my friend, and most importantly my role model. Finally, to my parents Laura and Gerry; Thank you both for your continuous love and encouragement, and for making me the person I am today.
Chapter 1
Introduction

1.1 Background

Since the beginning of the space age in 1957 with the launch of Sputnik 1 by the Soviet Union, humanity has slowly pushed out into space and established a constant presence in near-Earth orbit. Whether or not we realize it on a day to day basis, space has become an essential part of everyday life for most modern societies. Some examples of space technologies that we use benefit from every day include the Global Positioning Satellite (GPS) constellation, telecommunication networks, scientific and environmental satellites for weather and disaster preparedness, and military support satellites. In addition to direct space technologies, there are a myriad of spinoff technologies that stemmed from the National Aeronautics and Space Administration’s (NASA) space exploration program which include everything from medicine to industrial advances to direct consumer products [1]. As humans expand their presence in space, certain commonly used orbits become crowded and the problem of monitoring and managing the objects in space becomes exponentially more difficult.

Space Situational Awareness (SSA) is a term that has gained recent popularity in regards to managing this problem, and was underscored as an essential component to protect national security and other interests in a report by the National Research Council [2]. According to the U.S. Air Force Doctrine Document 3-14, SSA is defined as 'the requisite foundational, current, and predictive knowledge and characterization of space objects and the operational environment upon which space operations depend- including physical, virtual, information, and human
dimensions— as well as all factors, activities, and events of all entities conducting or preparing to conduct space operations’ [3]. In a broad sense, the goal of SSA can be summarized as the maximization of knowledge about past, present, and future orbital states, physical characteristics, and capabilities of all natural and artificial satellites in near-Earth orbit [4]. One major facet of SSA is debris tracking for the purposes of collision risk assessment. As of July 2018 the NASA Orbital Debris Program Office Reports that there are 19,137 man-made objects (active/defunct satellites and other debris) being tracked by the U.S Space Surveillance Network (SSN) [5]. Coupled with limited sensor availability, these man-made objects as well as the many thousands of additional natural satellites pose a real problem for any entity with space assets it wishes to protect.

SSA becomes particularly important in the military domain, where knowledge of the capabilities needed to protect friendly assets, and to potentially disable or destroy enemy satellites, become matters of national security [6]. In January 2007, the Chinese intentionally destroyed one of their own defunct weather satellites with an Anti-Satellite (ASAT) missile test, creating a cloud of at least 2,087 pieces of debris trackable by the U.S. SSN, as well as an estimated 35,000 pieces as small as 1cm in size [7]. This missile test was the single largest space debris creating event in history, and the resulting debris continues to threaten the International Space Station (ISS) and other Low Earth Orbit (LEO) satellites to this day. Although there has never been a publically documented military strike in which one country destroys the satellite of another country, the 2007 Chinese ASAT test and similar tests from both the United States and Russia, demonstrates that such capabilities currently exist.

Unfortunately, the problem of tracking and maintaining a catalog of satellites is exacerbated by the presence of uncertainty in almost all aspects of space operations. Uncertainty in the dynamic model, measurements, orbital state, and maneuver capabilities can all lead to errors in the ability to predict future states of the satellite. These uncertainties can dramatically limit or completely eliminate the ability to perform SSA tasks such as collision assessment, satellite tracking and characterization. In light of this, there is an evident need to formulate some of the most fundamental space problems in a stochasitic framework which can quantify uncertainty and provide probabilistic solutions to inform decision making.

The Lambert problem and the solution to a given orbit dynamical model
represent two fundamental and complimentary problems in astrodynamics, which have been rigorously studied for decades. The Lambert problem is a two-point boundary value problem, where the initial and final position of a satellite and the time of flight are given, and the solution is the initial and final velocity which satisfies those conditions. This problem is complimentary to the solution of a dynamic model, which solves for the final position and velocity of a satellite given the initial position and velocity. In the simplest two body model, there are analytic expressions for the relationship between initial and final satellite state; however, for more complex dynamic models, numerical integration is frequently used to integrate the equations of motion forward in time to the final state. One similarity between both of these problems is that the given variables are assumed to be deterministic and known to infinite precision. When we consider the input variables to have some uncertainty associated with them, as would be the case in realistic satellite operation scenarios, the solution to these problems becomes stochastic. The stochastic formulation of the Lambert problem is called the uncertain Lambert problem, and the stochastic formulation of a given dynamical model is known as the reachability set problem.

Traditionally, any non-deterministic solutions to these two problems use first order sensitivity approaches to determine how the solution varies linearly with respect to the deviation around a nominal solution. These approaches provide a means to determine the solution in a small region around a nominal trajectory, however they quickly break down due to high non-linearities present in the problem dynamics. The goal of this thesis is to develop and implement the framework needed to develop a probabilistic solution to the uncertain Lambert Problem and the reachability set problem using higher order sensitivity matrices which is valid not only around a nominal trajectory but over the domain of a probability distribution function (pdf).

1.2 Thesis Outline

This section provides a brief summary of the chapters and overall structure of this thesis.

Chapter 2 provides the mathematical background necessary for the development of the higher order sensitivity method proposed in this thesis. The background
begins with a discussion of the Taylor series expansion and how polynomials can be used to approximate non-linear functions. The least squares method and how to formulate minimum error polynomial coefficients with respect to a given pdf by leveraging orthogonal polynomials is presented. Next, orthogonal polynomials, their properties, and how to compute them using Gram-Schmidt Orthogonalization is discussed. Finally, a summary is given on numerical integration techniques and methods to compute least squares polynomial approximation coefficients.

Chapter 3 presents the uncertain Lambert problem. Background literature and gaps in research are identified, and the improvements this thesis provides are discussed. Two simulation test cases are described in detail with all parameters provided so that they may be reproduced in the future, and the results of these simulations for varying sensitivity matrix approximation orders are presented and discussed.

Chapter 4 presents the reachability set problem, using a similar analysis format as in Chapter 3. Two test cases taken from literature are described in detail and the results are presented and discussed for varying sensitivity matrix approximation orders.

Chapter 5 summarizes the thesis and provides concluding remarks on the relative advantages and limitations of the higher order sensitivity matrix method proposed in this thesis. This chapter also makes suggestions for future research to expand upon the work in this thesis.
Chapter 2  
Mathematical Background

This chapter will present the preliminary mathematics underlying the proposed polynomial approximation method. The concepts presented in this chapter will be used in numerical analyses in later chapters. This chapter will cover topics including the Taylor Series Expansion, The Least Squares Method, Orthogonal Polynomials, and Numerical Integration Methods.

2.1 The Taylor Series Expansion

The Taylor series provides an infinite polynomial series expansion of a smooth function, $f(x)$ as a function of perturbation $\delta x$ around a point $x^*$. In the context of this thesis $\delta x$ is characterized as the deviation due to uncertainty in the random variable $x$. For a function to have a valid Taylor series expansion over an open interval, the function must be infinitely differentiable and analytic over that interval. Assume for the time being, that $f(x)$ is a scalar function. The taylor series for $f(x)$ can be written as:

$$f(x) = f(x^*) + \frac{f'(x^*)}{1!}(x - x^*) + \frac{f''(x^*)}{2!}(x - x^*)^2 + \frac{f'''(x^*)}{3!}(x - x^*)^3 + ... \quad (2.1)$$

This series can also be written in compact form given by Eq.(2.2) (see Ref. [8]).

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x^*)}{i!}(x - x^*)^i \quad (2.2)$$

Note that if $x^* = 0$, the series becomes a special case known as the Maclaurin series. The derivative terms $f^{(i)}(x^*)$ are constant because they are evaluated at the point
at which the expansion is about, so the Taylor series can be written as an infinite
polynomial series known as a Taylor polynomial.

\[ f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \ldots \] (2.3)

Since Taylor polynomials are infinite, for practical purposes the polynomial is
truncated to a finite number of terms and is referred to as the \( d^{th} \) order Taylor
polynomial approximation.

\[ f(x) \approx p^{(d)}(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_d x^d \] (2.4)

The validity of a \( d^{th} \) order Taylor series approximation is dictated by size of
the perturbation \( \delta x \), and the smoothness of the function \( f(x) \). Taylor series
approximations can be used to approximate many transcendental functions. One
such example is the sin function, for which the \( 7^{th} \) order approximation is shown
below:

\[ \sin^{(7)}(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \] (2.5)

Figure 2.1 shows the Taylor series approximation of \( \sin(x) \) about \( x^* = 0 \) for varying
approximation orders \( d \) on the interval \([-\pi, \pi]\). Notice that as the order increases,
the approximation becomes more accurate over a larger portion of the interval.
This is the fundamental motivation behind the polynomial approximation method.

In the general case, the polynomial described by Eq. (2.4) can be rearranged
into any linear combination of \( c_i x^i \) to express the polynomial \( p^{(d)}(x) \) in terms of
new coefficients and arbitrary polynomial basis functions \( \phi_i(x) \) of maximum power
\( i \) for \( i = 0, 1, \ldots, d \).

\[ p^{(d)}(x) = c^T \phi(x) = c_i \phi_i(x) \] (2.6)

Notice that for 1D systems the total number of basis functions is \( M = d + 1 \),
however, this linear growth breaks down in higher dimensions. Consider a general
multi-dimensional system with \((n \times 1)\) input vector \( \mathbf{x} \), and \((m \times 1)\) output vector
\( \mathbf{y} = f(\mathbf{x}) \).

\[ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \] (2.7)
Consider the Taylor polynomial approximation for this multi-dimensional system. It is necessary that the polynomial approximation contains all permutations of the input vector for each power, so the total number of basis functions $M$ follows the factorial relationship given by Eq.(2.8) [9].

\[
M = (n+d) \binom{n+d}{d} = \frac{(n+d)!}{d!n!} \tag{2.8}
\]

It follows naturally then, that the number of basis functions contained in just the $i^{th}$ approximation order follows the relationship given by:

\[
b_i = (n+i) \binom{n+i}{i} - (n+i-1) \binom{n+i-1}{i-1} = \frac{(n+i)! - i(n+i-1)!}{i!n!} \tag{2.9}
\]

The $d^{th}$ order polynomial approximation for $y$ must now contain all permutations of $x$ up to a maximum order $d$. Thus, the general case for the polynomial approximation is:

\[
y_j \approx c_j^{(0)} + c_j^{(1)T} \phi^{(1)} + c_j^{(2)T} \phi^{(2)} + \ldots c_j^{(d)T} \phi^{(d)} \tag{2.10}
\]

Where $y_j$ is the $j^{th}$ output for $j = 1, 2 \ldots m$, $\phi^{(i)}$ is a $(b_i \times 1)$ vector containing
all $i^{th}$ order permutations of $\mathbf{x}$, and $\mathbf{c}^{(i)}_j$ is the vector of corresponding coefficients. In this thesis $\mathbf{c}^{(i)}_j$ are refered to interchangably as approximation coefficients and sensitivity matrices. The $\mathbf{c}^{(0)}_j$ terms represent the mean of $y_j$ and the $\mathbf{c}^{(1)}_j$ terms represent the first order sensitivity terms (similar in a sense to the state transition matrix). Consider the example where $d = 2$ and $\phi^{(i)}$ are $i^{th}$ order monomials.

\[
\mathbf{c}^{(1)}_j = \begin{cases}
  \mathbf{c}^{(1)}_{j,1} \\
  \mathbf{c}^{(1)}_{j,2} \\
  \vdots \\
  \mathbf{c}^{(1)}_{j,b_1}
\end{cases}, \quad \phi^{(1)} = \begin{cases}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{cases}, \quad \mathbf{c}^{(2)}_j = \begin{cases}
  \mathbf{c}^{(2)}_{j,1} \\
  \mathbf{c}^{(2)}_{j,2} \\
  \vdots \\
  \mathbf{c}^{(2)}_{j,b_2}
\end{cases}, \quad \phi^{(2)} = \begin{cases}
  x_1^2 \\
  x_1x_2 \\
  \vdots \\
  x_2x_3 \\
  \vdots \\
  x_2x_n \\
  \vdots \\
  x_n^2
\end{cases}
\] (2.11)

Similar to the single dimensional case, $\mathbf{c}^{(i)}_j$ and $\phi^{(i)}$ can be rewritten into any arbitrary polynomial basis functions of maximum power $i$ and new coefficients. A polynomial approximation for each output element in $\mathbf{y}$ can now be written as

\[
y_j \approx \mathbf{c}_{j,i}\phi_i \quad \text{for} \quad \begin{cases}
  i = 1, 2, \ldots M \\
  j = 1, 2, \ldots m
\end{cases} \quad (2.12)
\]

\[
\mathbf{y} \approx \mathbf{C}\phi(\mathbf{x}) \quad (2.13)
\]

where $\mathbf{C}$ is an $(m \times M)$ matrix with elements $c_{j,i}$ in the $j^{th}$ row and $i^{th}$ column, and $\phi(\mathbf{x})$ is an $(M \times 1)$ vector of all arbitrary polynomial basis functions. Equation (2.12) is the most general expression for a Taylor polynomial approximation of a multi-dimensional system.

One additional thing to note on the use of polynomials in general is scaling. In general a $d^{th}$ order polynomial will become numerically ill conditioned if $d$ is large and if $x << -1$ or $1 << x$. An ill conditioned polynomial will cause numerical errors to accumulate due to finite precision computing capabilities. To circumvent this issue, it is common to express the variable $x$ in terms of the normalized
parameter $\zeta$ such that:

$$x = x^* + S\zeta$$  \hspace{1cm} (2.14)

Where $S$ is a scaling Matrix that is dependent on the type of weighting function being considered. The two most common distributions are the uniform and the Gaussian distribution. For a uniform distribution $\zeta$ is sampled on the interval $[-1, 1]$ and $S$ is a diagonal matrix with terms $(b_i - a_i)$ where $x_i$ is uniform on the interval $[a_i, b_i]$. For a Gaussian distribution, $\zeta$ is is sampled from a unit variance, zero mean normal distribution, and $S = \sqrt{\Sigma}$ where $\Sigma$ is the covariance matrix of $x$. By scaling $x$ in this manner, we can greatly reduce numerical errors in our polynomial approximation. To this end, the coefficients will be calculated with respect to the normalized variable $\zeta$ such that

$$y \approx C\phi(\zeta)$$  \hspace{1cm} (2.15)

Direct computation of Taylor series is cumbersome because coefficients $c_{j,i}$ requires the computation of higher order derivatives. Even when $f(x)$ is known, computing $c_{j,i}$ can be a tedious process. Furthermore, the Taylor series is valid only in a small neighborhood of $x^*$. In the next section we present a least squares approach to compute these coefficients without explicitly computing derivatives. In addition to eliminating the need to take higher order derivatives, the least squares process also guarantees that error is minimized over the desired domain.

### 2.2 The Least Squares Method

Least squares is a statistical method for solving an overconstrained system which provides the globally minimum error between a polynomial fit and an unknown function $f(x)$. The method was first published in 1805 by Adrien Marie Legendre, however the discovery is often attributed to Carl Friedrich Gauss for having developed the method as early as 1795 [10]. In general a best fit polynomial can be found for any non linear function, and the accuracy of this polynomial approximation will increase as the order of the polynomial increases. We begin by defining the error to be

$$e_j = y_j - c_{j,i}\phi_i$$  \hspace{1cm} (2.16)
In the general case, the cost function $J$ can be defined as the inner product of error squared

$$J = \frac{1}{2} \langle e_j, e_j \rangle_{\rho}$$

$$= \frac{1}{2} \langle y_j - c_{j,i} \phi_i, y_j - c_{j,i} \phi_i \rangle_{\rho}$$

(2.17)

Where $\langle \cdot, \cdot \rangle_{\rho}$ denotes the inner product with respect to the probability distribution function (pdf) $\rho(x)$ associated with $x$. Note that this formulation can be used for the inner product of either continuous or discrete variables. For the discrete case, the inner product of two $(N \times 1)$ vectors $v(x)$ and $u(x)$ is defined to be

$$\langle v(x), u(x) \rangle_{\rho} = \sum_{i=1}^{N} \rho(x_i)v(x_i)u(x_i)$$

(2.18)

Geometrically, the discrete inner product is a measure of the projection of the components of vector $v$ onto the components of vector $u$. A good discussion on vector projections and their relationships to orthogonality can be found in [11]. In higher dimensional space, it is difficult to represent vectors graphically, so the notion of an inner product is easier to represent algebraically. As a vector increases in length ($N \to \infty$), the discrete vectors become continuous functions. In the continuous case, the inner product becomes an integral over the domain of the pdf

$$\langle v, u \rangle_{\rho} = \int_{\Omega} v(x)u(x)\rho(x)dx$$

(2.19)

Returning to our cost function, we attempt to find the coefficients which minimize $J$. The 1st order necessary condition is:

$$\min_{c_{j,k}} J = \frac{\partial J}{\partial c_{j,k}} = 0, \text{ for } k = 1, 2, \ldots M$$

(2.20)

Taking the partial derivative of Eq. (2.17) with respect to the coefficient vector gives the following set of $M$ equations known as Normal equations:

$$\frac{\partial J}{\partial c_{j,k}} = 0 = \langle e_j, \frac{\partial e_j}{\partial c_{j,k}} \rangle_{\rho}$$

$$0 = \langle (y_j - c_{j,i} \phi_i), -\phi_k \rangle_{\rho}$$

(2.21)
Because the inner product is a linear operator, we can regroup Eq. (2.21) as:

\[ \langle y_j, \phi_k \rangle_\rho = c_{j,k} \langle \phi_i, \phi_k \rangle_\rho \]  

(2.22)

This can also be written in Matrix form where \( A \) is a \((m \times M)\) matrix, and \( B \) is a \(M \times M\) matrix.

\[ A = CB \]

(2.23)

\[ A_{j,k} = \langle y_j, \phi_k \rangle_\rho, \quad B_{i,k} = \langle \phi_i, \phi_k \rangle_\rho \]

Eq. (2.23) is known as the normal equation. It is important to note that the normal equation expressed using inner products is a generalization, which can be evaluated for either continuous or discrete cases. Up until this point \( \phi_i \) were arbitrary polynomial basis functions, however, if we select them specifically to be orthogonal with respect to \( \rho(x) \), \( B \) becomes a diagonal matrix. See Section 2.3 for details.

\[ B_{i,k} = \begin{cases} 
\langle \phi_i, \phi_i \rangle_\rho & \text{for } i = k \\
0 & \text{for } i \neq k 
\end{cases} \]  

(2.24)

A diagonal \( B \) matrix allows for easy computation of the inverse, and causes the coefficients to become linearly independent. The solution for the best fit coefficients can now be written as:

\[ C = AB^{-1} \]

(2.25)

\[ c_{j,i} = \frac{\langle y_j, \phi_i \rangle_\rho}{\langle \phi_i, \phi_i \rangle_\rho} \]

The computation of terms in \( B \) can be done offline for any pdf through application of the Gram-Schmidt Process. The major challenge in this problem lies in computing the multi-dimensional continuous inner products of the generalized non-linear function \( y_j \) found in the \( A \) matrix. Detailed discussions of both Gram-Schmidt orthogonalization and numerical integration methods can be found in the following sections.
2.3 Orthogonal Polynomials

The property of orthogonality stems from the definition of the inner product and is defined as:

\[
\langle v(x), u(x) \rangle_\rho = 0
\]  
(2.26)

Where \(v(x), u(x)\) are orthogonal functions of the variable \(x\). Gram-Schmidt orthogonalization is a method for computing polynomials that satisfy this property from a non-orthogonal basis set. A discussion of the Gram Schmidt Process can be found in [11,12]. For a single variable (n=1), take the set of non orthogonal polynomials \(\psi\) of maximum power \(d\) to be:

\[
\psi^{(d)} = \begin{cases} 
1 \\
x \\
x^2 \\
\vdots \\
x^d 
\end{cases}
\]  
(2.27)

We seek to find the orthogonal set of polynomials \(\phi\), from non orthogonal basis set \(\psi\). Since we are not restricted on what the first orthogonal polynomial \(\phi_0\) can be, we select it to be \(\phi_0 = 1\) for simplicity. To find the next orthogonal polynomial we subtract the first orthogonal polynomial multiplied by some scaling constant \(c\) from the second non-orthogonal polynomial.

\[
\phi_2(x) = \psi_2(x) - c\phi_1(x)
\]  
(2.28)

This guarantees to eliminate all components of \(\phi_1\) which have a non-zero projection on \(\phi_2\). Taking the inner product of both sides of this equation with \(\phi_1\) and applying the definition of orthogonality, we can guarantee that \(\phi_1\) and \(\phi_2\) are orthogonal.

\[
\langle \phi_1(x), \phi_2(x) \rangle_\rho = \langle \phi_1(x), (\psi_2(x) - c\phi_1(x)) \rangle_\rho = 0
\]  
(2.29)

The constant coefficient can be solved for as shown in Eq. (2.30)

\[
c = \frac{\langle \psi_2(x), \phi_1(x) \rangle_\rho}{\langle \phi_1(x), \phi_1(x) \rangle_\rho}
\]  
(2.30)
Figure 2.2 shows graphically how eliminating the projection of $\psi_2$ on $\phi_1$ can guarantee the orthogonality with Eq. (2.28). Continuing in this manner, we can always determine the next orthogonal polynomial $\phi_k(x)$ by eliminating projections of the previous orthogonal polynomials on the next non-orthogonal polynomial. This leads to the recursive relationship given Eq. (2.33) by known as the Gram-Schmidt formula [11].

$$\phi_k = \psi_k - \sum_{i=1}^{k-1} \frac{\langle \psi_k, \phi_i \rangle}{\langle \phi_i, \phi_i \rangle} \phi_i$$ for $k = 1, 1 \ldots d$  \hspace{1cm} (2.31)$$

It is important to note that these orthogonal polynomials are non-unique, and can be linearly scaled by any constant value $s_k$ and remain orthogonal.

$$\langle \phi_1, \phi_2 \rangle_\rho = \langle s_1 \phi_1, s_2 \phi_2 \rangle_\rho = 0$$  \hspace{1cm} (2.32)$$

Scaling the polynomial is also known as normalization and is given by Eq.(2.33).

$$\phi'_k = s_k \phi_k$$  \hspace{1cm} (2.33)$$

Where $\phi'_k$ is the normalized polynomial. Various normalizations are used for different orthogonal polynomials, but two commonly used normalizations are orthonormalization and monic normalization [8]. Orthonormalization scales the polynomial such that the inner product of the polynomial with itself is 1, and monic normalization scales the polynomial such that the leading coefficient is 1. Orthonormalization is
given by:

\[ s_k = \frac{1}{\|\phi_k\|}, \quad \langle \phi'_k, \phi_k \rangle_\rho = 1 \]  

(2.34)

Where \( \|\phi_k\| \) is defined as the 2-norm of the polynomial:

\[ \|\phi_k\|^2 = \int_\Omega \phi_k(x)^2 \rho(x) dx \]  

(2.35)

and monic normalization is given by:

\[ \phi_k(x) = c_k x^k + c_{k-1} x^{k-1} + \ldots c_0, \quad s_k = \frac{1}{c_k} \]  

(2.36)

Assume that we would like to determine the set of orthogonal polynomials with respect to the uniform pdf \( \rho(x) = \frac{1}{2} \). The first few orthogonal polynomials are calculated in the following manner:

\[ \phi_0 = \psi_0 = 1 \]

\[ \phi_1 = \psi_1 - \frac{\langle \psi_1, \phi_0 \rangle_\rho}{\langle \phi_0, \phi_0 \rangle_\rho} \phi_0 = x \]

\[ \phi_2 = \psi_2 - \frac{\langle \psi_2, \phi_0 \rangle_\rho}{\langle \phi_0, \phi_0 \rangle_\rho} \phi_0 - \frac{\langle \psi_2, \phi_1 \rangle_\rho}{\langle \phi_1, \phi_1 \rangle_\rho} \phi_1 = x^2 - \frac{1}{3} \]  

(2.37)

\[ \phi_3 = \psi_3 - \frac{\langle \psi_3, \phi_0 \rangle_\rho}{\langle \phi_0, \phi_0 \rangle_\rho} \phi_0 - \frac{\langle \psi_3, \phi_1 \rangle_\rho}{\langle \phi_1, \phi_1 \rangle_\rho} \phi_1 - \frac{\langle \psi_3, \phi_2 \rangle_\rho}{\langle \phi_2, \phi_2 \rangle_\rho} \phi_2 = x^3 - \frac{3}{5} x \]

Since we can use any normalization factor for the normalization in equation (2.33) and the polynomials will remain orthogonal, If we choose to normalize these polynomials such that \( s_k \phi_k(1) = 1 \) they become the well known Legendre Polynomials:

\[ \phi_0 = 1 \]

\[ \phi_1 = x \]

\[ \phi_2 = \frac{1}{2} (3x^2 - 1) \]  

(2.38)

\[ \phi_3 = \frac{1}{2} (5x^3 - 3x) \]

Another important property of orthogonal polynomials is that they can be computed recursively without performing the Gram-Schmidt process. This type of relation is known as a three-term recurrence relation. In general, any orthogonal basis set
can be found using a three-term recursive relationship of the form:

$$\phi_{k+1}(x) = (a_k x + b_k) \phi_k(x) - c_k \phi_{k-1}(x) \quad (2.39)$$

There are many different types of orthogonal polynomials for which this relationship is valid, but the most commonly used are Legendre Polynomials for uniform distributions and Hermite Polynomials for Gaussian distributions. Table 2.3 is adapted from Ref. [8] and summarizes a select few orthogonal polynomials, their weighting functions, and their recurrence relation coefficients. See Chapter 18 in Ref. [8] for details on more complex orthogonal polynomials. The next section combines many of the concepts discussed in this chapter, and demonstrates how orthogonal polynomials relate to numerical integration and computation of the higher order sensitivity matrices.

### 2.4 Numerical Integration Methods

This section will present various numerical integration methods and discuss the strengths and weaknesses of each. In general, the integral of a function $f(x)$ with respect to the pdf $\rho(x)$ can be expressed as:

$$F = \int_a^b f(x) \rho(x) dx \quad (2.40)$$

Note that for a standard unweighted integral the pdf is a uniform density function $\rho(x) = 1$. All numerical integration techniques are variations of the same basic method, which seek to approximate the integral of a function as the summation of
a finite number of weights \( w_i \), multiplied by the function evaluated at \( N \) points \( x_i \).

\[
F \approx \sum_{i=1}^{N} w_i f(x_i)
\]  

(2.41)

Note that a constraint on the weights is that the sum of \( w_i \) must equal 1.

\[
\sum_{i=1}^{N} w_i = 1
\]  

(2.42)

The method by which the points and weights are selected is what separates numerical integration techniques from one another.

### 2.4.1 The Monte Carlo Method

The Monte Carlo Method is a very commonly used method for performing numerical integration. The method involves randomly sampling points \( x_i \) from the pdf of \( x \) and assigning equal weight to all points.

\[
F \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)
\]  

(2.43)

To illustrate this method, an example covariance matrix for a Gaussian pdf will be computed. Determining the covariance matrix requires the computation of the following multi-dimensional integral:

\[
\Sigma = E[(x - \mu)(x - \mu)^T] = \int_{-\infty}^{\infty} (x - \mu)(x - \mu)^T \rho(x) \, dx
\]  

(2.44)

Where \( \rho(x) = e^{-\frac{1}{2}x^T x} \) is a standard unit variance zero mean Gaussian distribution. Assume that \( x \) is a \((3 \times 1)\) random vector pdf \( \rho(x) \) such that the true covariance matrix of \( x \) is given by equation (2.45).

\[
\Sigma_{\text{true}} = \begin{bmatrix}
\sigma_x^2 & 0 & 0 \\
0 & \sigma_y^2 & 0 \\
0 & 0 & \sigma_z^2 \\
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]  

(2.45)
The covariance can be computed numerically using the Monte Carlo method by randomly sampling $x_i$ from the standard Gaussian pdf $\rho(x)$, and computing the integral with equation (2.46).

$$\Sigma_{MC} \approx \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T$$

(2.46)

This method was used to compute the covariance matrix for values of $N$ ranging from $10$ to $100,000,000$. The resulting covariance matrix for $N = 100,000,000$ is shown below.

$$\Sigma_{MC} \approx \begin{bmatrix}
1.0001 & 1.6357 \times 10^{-4} & 3.8732 \times 10^{-5} \\
1.6357 \times 10^{-4} & 0.9998 & -7.7672 \times 10^{-5} \\
3.8732 \times 10^{-5} & 7.7672 \times 10^{-5} & 0.9999
\end{bmatrix}$$

The error $e$ was also computed for each value of $N$ as the matrix norm of true covariance minus calculated covariance, and plotted in Figure 2.3.

$$e = |\Sigma_{true} - \Sigma_{MC}|$$

(2.47)

From Figure 2.3, it is clear that a reasonable approximation of the covariance
integral can be found numerically with a very large number of Monte Carlo samples. However, there is significant burden associated with evaluating 100,000,000 sample points. Additionally, observing Figure 2.3 it can be seen that the error is reduced by approximately 1 order of magnitude for every 2 orders of magnitude of additional MC points. This trend gives an exponentially diminishing return on reducing error by adding more MC points. Clearly, we would like to avoid the computational burden associated with Monte Carlo numerical integration if possible.

### 2.4.2 Gaussian Quadrature

In General, deterministic alternatives to the random sampling numerical integration scheme are known as quadrature rules. A fundamental assumption of the quadrature method is that the function being integrated can be reasonably approximated by a polynomial. There are many different quadrature rules that are formulated to carefully leverage symmetries in a probability distribution function such that using a specific set of weights and points allows us to exactly recreate the integrals of polynomials. Perhaps the most well known quadrature is the Gaussian Quadrature Rule (GQR). Let us consider the integration of a polynomial function \( f(x) \) with respect to the probability distribution function \( \rho(x) \).

\[
\int_{a}^{b} f(x) \rho(x) dx = \sum_{i=1}^{N} w_i f(x_i) \quad (2.48)
\]

Substituting the \( d^{th} \) order Taylor series expansion for \( f(x) \) given by (2.4) into (2.48):

\[
\int_{a}^{b} \rho(x)(c_0 + c_1 x + c_2 x^2 + \ldots + c_d x^d) dx = \sum_{i=1}^{N} w_i (c_0 + c_1 x_i + c_2 x_i^2 + \ldots + c_d x_i^d) \quad (2.49)
\]

Recall that the coefficients \( c_i \) represent the partial derivatives of \( x \) evaluated at \( x^* \). Equating the coefficients gives a set of \( d + 1 \) equations known as the Moment
Constraint Equations (MCE’s) (2.50):

\[ \int_{a}^{b} \rho(x)dx = \sum_{i=1}^{N} w_{i} \]
\[ \int_{a}^{b} x\rho(x)dx = \sum_{i=1}^{N} w_{i}x_{i} \]
\[ \int_{a}^{b} x^{2}\rho(x)dx = \sum_{i=1}^{N} w_{i}x_{i}^{2} \]
\[ \vdots \]
\[ \int_{a}^{b} x^{d}\rho(x)dx = \sum_{i=1}^{N} w_{i}x_{i}^{d} \]

The left hand side of the MCE’s can be solved analytically for the given pdf, and the right hand side can be solved for the points \( x_{i} \) and weights \( w_{i} \). From (2.50), it is clear that there are \( 2N \) variables \( w_{i}, x_{i} \) available to solve the \( d+1 \) MCE’s. This implies that \( N \) points are required to exactly solve a \( d = 2N - 1 \) order polynomial. The fundamental theorem for Gaussian quadrature, and a discussion of the methods and algorithms needed to compute \( x, w_{i} \) can be found in Ref. [13].

The points \( x_{i} \) and weights \( w_{i} \) are closely related to orthogonal polynomials with respect to weighting function \( \rho(x) \). Take the polynomial \( \phi_{k}(x) \) to be an orthogonal polynomial with respect to \( \rho(x) \) with coefficients \( c_{k} \) and real roots \( a < t_{1} < t_{2} < \ldots < t_{k} < b \) defined as

\[ \phi_{k}(x) = c_{k} \prod_{i=1}^{k} (x - t_{i}) \]  

The quadrature points are simply the roots of this polynomial \( x_{i} = t_{i} \), and the weights can be calculated using Eq. (2.52) [8].

\[ w_{i} = \int_{\Omega} \frac{\phi_{k}(x)}{(x - x_{i})\phi_{k}'(x_{i})} \rho(x)dx \]  

To illustrate how this method works we will provide an example. Let us consider the integration of the function \( f(x) \) over the interval \([ -1, 1] \), and that that the function \( f(x) \) can be reasonably approximated by a third-order polynomial \( p(x) \).
Take the pdf to be uniform $\rho(x) = \frac{1}{2}$.

$$\frac{1}{2} \int_{-1}^{1} f(x)dx \approx \frac{1}{2} \int_{-1}^{1} p(x)dx \quad (2.53)$$

Take the 3rd order polynomial $p(x)$ and it’s derivative $p'(x)$ to be:

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$
$$p'(x) = a_1 + 2xa_2 + 3x^2a_3 \quad (2.54)$$

In this case there are 4 unknown coefficients $a_i$, so we need 4 equations to solve. Take 2 points $x_1$ and $x_2$, and plug them into $p(x)$ and $p'(x)$

$$p(x_1) = a_3x_1^3 + a_2x_1^2 + a_1x_1 + a_0$$
$$p(x_2) = a_3x_2^3 + a_2x_2^2 + a_1x_2 + a_0$$
$$p'(x_1) = 3a_3x_1^2 + 2a_2x_1 + a_1$$
$$p'(x_2) = 3a_3x_2^2 + 2a_2x_2 + a_1 \quad (2.55)$$

Solving for the coefficients in terms of $p(x)$, and it’s derivatives we find:

$$a_0 = \frac{p(x_2)x_1^2(x_1 - 3x_2) + x_2(p(x_1)(3x_1 - x_2)x_2 - x_1(x_1 - x_2)(x_2p'(x_1) + x_1p'(x_2)))}{(x_1 - x_2)^3}$$
$$a_1 = \frac{6x_1x_2(p(x_2) - p(x_1)) + (x_1 - x_2)(x_2^2p'(x_1) + x_1^2p'(x_2) + 2x_1x_2(p'(x_1) + p'(x_2)))}{(x_1 - x_2)^3}$$
$$a_2 = \frac{3p(x_1)(x_1 + x_2) - 3p(x_2)(x_1 + x_2)}{(x_1 - x_2)^3}$$
$$- \frac{(x_1 - x_2)(x_2(2p'(x_1) + p'(x_2) + x_1(p'(x_1) + 2p'(x_2))) = \frac{(x_1 - x_2)^3}{(x_1 - x_2)^3}$$
$$a_3 = \frac{-2p(x_1) + 2p(x_2) + (x_1 - x_2)(p'(x_1) + p'(x_2))}{(x_1 - x_2)^3} \quad (2.56)$$
The integral of the polynomial is known to be given by equation (2.57)

\[
\frac{1}{2} \int_{-1}^{1} p(x) dx = \frac{1}{2} \int_{-1}^{1} (a_3 x^3 + a_2 x^2 + a_1 x + a_0) dx \\
= \frac{1}{2} \left[ \frac{a_3 x^4}{4} + \frac{a_2 x^3}{3} + \frac{a_1 x^2}{2} + a_0 x \right]_{-1}^{1} \\
= a_0 + \frac{a_2}{3}
\]  

(2.57)

Substituting our expressions for \(a_0\) and \(a_2\) into (2.57) and regrouping similar terms of \(p(x)\) and \(p'(x)\) gives us equation (2.58)

\[
\frac{1}{2} \int_{-1}^{1} p(x) dx = \frac{1}{2} \left[ \frac{a_3 x^4}{4} + \frac{a_2 x^3}{3} + \frac{a_1 x^2}{2} + a_0 x \right]_{-1}^{1} \\
= \frac{1}{2} \left( a_0 x^2 + \frac{a_2}{3} \right) \\
\]

(2.58)

To find an expression for the integral of \(p(x)\) in the form of (??), we must set the coefficients of the \(p'(x)\) terms equal to zero and set the coefficients of the \(p(x)\) terms equal to \(w_i\).

\[
0 = -\frac{2x_2 + x_1 (1 + 3x_2^2)}{3(x_1 - x_2)^2}, \quad 0 = -\frac{2x_1 + x_2 (1 + 3x_1^2)}{3(x_1 - x_2)^2} \\
w_1 = \frac{(x_2 - x_1 (1 + 3x_2^2))}{(x_1 - x_2)^3}, \quad w_2 = \frac{(-x_1 + x_2 (1 + 3x_1^2))}{(x_1 - x_2)^3}
\]

(2.59)

Solving the equations equal to zero in the first row of (2.4.2) provides the points \(x_1\) and \(x_2\) which can then be substituted into the second row of equations and solved for \(w_1\) and \(w_2\). In the case of the 3\(^{rd}\) order polynomial, the GQR points and weights are:

\[
x_1 = \frac{1}{\sqrt{3}}, \quad x_2 = -\frac{1}{\sqrt{3}} \\
w_1 = \frac{1}{2}, \quad w_2 = \frac{1}{2}
\]

(2.60)

Notably, these points correspond to the roots of the 2\(^{nd}\) order Legendre Polynomial. To further emphasize this point, calculate the weights using Eq. (2.52) for \(\rho(x) = \frac{1}{2}\) and a second order Legendre Polynomial. The roots of the 2nd Legendre polynomial \(\phi_2(x)\) are easily found to be:

\[
\phi_2(x_i) = 0 = \frac{1}{2} (3x^2 - 1), \quad x_i = \left[ \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right]
\]

(2.61)
Plugging these values into Eq. (2.52)

\[
w_1 = \int_{-1}^{1} \frac{\phi_2(x)}{(x - x_1)\phi_2'(x_1)} \rho(x)dx = \frac{1}{2} \int_{-1}^{1} \frac{1}{2\sqrt{3}} (3x + \sqrt{3})dx
\]

\[
= \frac{1}{4\sqrt{3}} \left[ \frac{3x^2}{2} + \sqrt{3}x \right]_{-1}^{1} = \frac{1}{2}
\]

(2.62)

The result is the same whether the points and weights are solved for long-hand or using the orthogonal polynomial approach. In general, the points \(x_i\) can be calculated for many different pdf’s by taking the roots of orthogonal polynomials (see Table 2.3), and the weights \(w_i\) can be calculated using Eq (2.52). Next an example is presented which will summarize many of the concepts presented in this chapter and show how they are utilized in the polynomial approximation method.

### 2.5 Example Higher Order Sensitivity Problem

In this case, we wish to determine the least squares approximation for the function \(f(x)\) over the interval \([-1, 1]\).

\[f(x) = \sin(5x)e^{-2x^2}\]

(2.63)

In this problem, the scaling factor \(S\) in Eq. (2.14) is not necessary because \(x\) is chosen such that \(x = \zeta\). We compute the inner products assuming the function to
have a uniform distribution $\rho(x) = \frac{1}{2}$ and the orthogonal basis functions $\phi(x)$ to be the Legendre Polynomials. Starting with a discrete case, we take 50 uniformly spaced samples of $x$ and the function $f(x)$, and compute the discrete inner products (Eq. (2.18)) required for the $A$ and $B$ matrices. The normal equation (2.23), can then be solved for the least squares coefficients. This process is repeated for varying the approximation order between $d = 1$ and $d = 7$. After the coefficients are calculated the best fit polynomials are plotted in Figure 2.4(a), and the Root Mean Square Error (RMSE) for each of the approximations is plotted in Figure 2.4(b) using 1000 points uniformly spaced points. RMSE error is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{1000} [f(x_i) - c^T \phi(x_i)]^2}{1000}}$$

(2.64)

From inspection of Figure 2.4(a), it is readily apparent that as the approximation order is increased the approximation becomes closer to the true function $f(x)$. The same trend is quantitatively shown in Figure 2.4(b), which plots the RMSE between the polynomial approximation and $f(x)$ vs the order of polynomial approximation.

Now we hold the polynomial order to be fixed at $d = 7$, and vary the number of discrete points $N$ used to calculate the approximation. Figure 2.5(a) shows plots of polynomial approximations for varying $N$, and Figure 2.5(b) shows a log-log plot of RMSE vs $N$. Notice that as $N \to \infty$ the discrete 7th order approximation asymptotically approaches the continuous polynomial approximation. The con-
tinuous least square approximation is found by computing the inner products in Eq. (2.25) using the continuous definition in Eq. (2.19). A 10th order Gaussian quadrature rule is used to compute these integrals numerically. The least squares approximation coefficients found using continuous inner products represent the most accurate possible \( d \)th order polynomial approximation of \( f(x) \). The motivation behind the higher order sensitivity approximation method is to calculate the Taylor Polynomial coefficients \( c_i \) numerically using the continuous least squares method. This allows us to calculate the Taylor polynomial coefficients of a function without explicitly taking Jacobians of \( f(x) \).

### 2.5.1 Alternatives to Gaussian Quadrature

Gaussian Quadrature is a powerful tool, that provides a minimal number of points needed to compute polynomial integrals in 1D. Unfortunately, integrating polynomials in higher dimensions becomes significantly more difficult. To integrate a polynomial of the same degree in higher dimensions using Gaussian quadrature, one must construct a grid of quadrature points by taking the tensor product of \( x_i \). Unfortunately, this leads to exponential growth in the necessary number of points with increasing dimension. For a Gaussian quadrature rule that is able to integrate an \( n \) dimensional, \( d = 2N - 1 \) degree polynomial, one needs \( N^n \) Gaussian quadrature points. This means that even a moderately high polynomial degree and system dimensionality, say, \( n = 6 \) and \( N = 5 \), 15,625 points are needed to compute the integral. Due to the high computational load this creates, especially when the evaluation of a single point is computationally expensive, higher dimensions can quickly create restrictions on the degree of the polynomial approximation and by extension, on the accuracy of the solution.

Fortunately, there are other quadrature methods that exist which are able to reduce the number of points needed in higher dimensions. Sparse grid quadratures are constructed by taking specific tensor product combinations of one dimensional quadrature rules to reduce the effective dimension of the system [14]. Unfortunately, this process introduces negative weights which can introduce significant error to the integrated function. In particular, the Unscented Transformation (UT) which can integrate up to 2nd order polynomials in \( n \) dimensions [15], and its extension Conjugate Unscented Transformation (CUT) [16–20] will be utilized in
Figure 2.6. A Schematic of CUT Axes

the implementation of the polynomial approximation method.

The CUT approach can be considered an extension of the conventional UT method that satisfies additional higher order moment constraints. Rather than using tensor products to determine quadrature points and weights as in Gauss quadrature, the CUT approach judiciously selects special structures to extract symmetric quadrature points constrained to lie on specially defined axes as shown in Figure 2.6. For each cubature point, two unknown variables, a weight $w_i$ and a scaling parameter $r_i$ are assigned. The moment constraints equations for the desired order are derived in terms of unknown variables $r_i$ and $w_i$. Because of

Figure 2.7. Comparison of Quadrature Schemes for Same Order of Accuracy
the symmetries of cubature points, the odd-order moment constraint equations are automatically satisfied, so that \( w_i \) and \( r_i \) are found by solving just the even order equations. The order of these moment constraint equations dictates the set of cubature points, often referred to as CUT4, CUT6 and CUT8 points. For this analysis, the CUT8 points will be used because these points satisfy the most moment constraints, which provides the highest accuracy. These new sets of so-called sigma points are guaranteed to exactly evaluate expectation integrals involving polynomial functions with significantly fewer points. Tabulated values for the CUT8 scaling parameters \( (r_i) \) and weights \( (w_i) \) can be found in [20].

Figure 2.7 shows a comparison of the number of points required for CUT and Gauss-Legendre quadratures for similar accuracy, clearly illustrating the reduced growth exhibited by the CUT method. The CUT method has already been implemented successfully in cross-disciplinary fields of study, including: the study of volcanic ash plumes, conjunction analysis, control of glucose in diabetic patients, soil characterization for civil engineering purposes, and analysis of energy harvesters [21–25]. More details about the CUT methodology and its comparison with conventional quadrature rules can be found in Ref. [16–20,26]. With the application of the CUT approach, we can generate higher order sensitivity matrices in a computationally efficient manner.
Chapter 3  
The Uncertain Lambert Problem

3.1 Problem Background

The classical Lambert Problem is a Two Point Boundary Value Problem (2PBVP), which connects two position vectors at known times to provide an initial velocity vector. This problem and its multiple variants have been well studied in literature [27, 28]. In this chapter, we are particularly interested in characterizing the uncertainty associated with the Lambert Problem solution as a function of position vector uncertainties at initial and final times. Due to limited measurement accuracy in sensor data, the characterization of uncertainty in the Lambert’s solution is an important operational concern for many Space Situational Awareness (SSA) problems including initial orbit determination, conjunction analysis, data association, and maneuver detection. The Uncertain Lambert Problem is introduced and discussed in detail in Schumacher et al. [29], where a state transition matrix formulation is used to characterize uncertainty. A similar analytic approach to characterize the uncertainty associated with the Lambert solution as a function of the initial and final state uncertainties involves linearizing the lambert solution using first order partial derivatives about the nominal orbit [30,31]. The linear variational approximation to the lambert solution is a computationally efficient process, however these linear analyses are only valid if initial and final state uncertainties are relatively small and may not provide insight into the exact distribution of error associated with the Lambert solution.
There is a large base of existing literature dedicated to the characterization and propagation of uncertainty in nonlinear dynamical systems [32-34], however the application of these methods to the solution of the Lambert problem is an unexamined problem. Luo et al. [35] reviews a wide range of methods and algorithms for the characterization and propagation of uncertainty, noting that typically higher order statistical moments are neglected due to exponentially increasing computational load for increasing state dimensionality.

The motivation of this work is to include higher order terms (HOT’s) of the Lambert solution to increase solution accuracy and the domain of position uncertainties under which the solution remains valid. Computing these higher order sensitivity terms requires the evaluation of multi-dimensional expectation integrals which traditionally becomes quickly computationally expensive at higher dimensions. To circumvent this issue, the current analysis utilizes the non-product quadrature method known as **Conjugate Unscented Transformation (CUT)** to compute the higher order sensitivity terms of the Lambert solution with respect to uncertain initial and final position vector in a computationally tractable manner.

### 3.2 Simulation Description

This section will discuss the uncertain Lambert problem formulation, and specific assumptions and parameters for two test case simulations. The two test cases are for a Low Earth Orbit (LEO) and a Geosynchronous Transfer Orbit (GTO) and each will be analyzed using varying degrees of higher order sensitivity matrices. Both test cases are assumed to have uncertainty in the initial and final position and governed by deterministic two-body dynamics, however this method can be extended to apply to orbit dynamical models that include perturbing effects.

Assume that the nonlinear function $f(x^*)$ expresses the deterministic Lambert solution in terms of mean $(n \times 1)$ input vector $x^*$ and mean $(m \times 1)$ output vector $y^*$.

$$y^* = f(x^*)$$

In this analysis $f(x^*)$ represents the deterministic lambert solver, and is based on the universal variable formulation of the Battin-Vaughn algorithm [27,28]. Details on this algorithm can be found in Ref. [36]. We wish to formulate a stochastic
solution to this problem by substituting $\mathbf{x}$ into $f(\mathbf{x}^*)$ where $\mathbf{x}$ is a $(n \times 1)$ continuous random vector with the joint pdf $\rho(\mathbf{x})$.

$$y = f(\mathbf{x})$$  \hspace{1cm} (3.2)

In the uncertain Lambert Problem, $f(\mathbf{x})$ denotes the statistical mapping between the uncertain position vectors the solution for the initial velocity:

$$\mathbf{x} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad y = v_1$$  \hspace{1cm} (3.3)

The relationship between $\mathbf{x}$ and $\mathbf{x}^*$ is facilitated by the normalized $(n \times 1)$ continuous random vector $\boldsymbol{\zeta}$, where each element $\zeta_j$ is associated with a zero mean unit variance Gaussian pdf:

$$\mathbf{x} = \mathbf{x}^* + \sqrt{\Sigma} \boldsymbol{\zeta} = \mathbf{x}^* + \delta \mathbf{x}$$

$$\rho(\zeta_j) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\zeta_j^2}{2}}$$  \hspace{1cm} (3.4)

Where $\mathbf{x}^*$ is the mean vector of $\mathbf{x}$, and $\Sigma$ is the covariance matrix associated with $\mathbf{x}$. $\delta \mathbf{x}$ represents the deviation in input due to uncertainty. Note that because $\boldsymbol{\zeta}$ is a normally distributed vector, the orthogonal basis functions $\phi(\boldsymbol{\zeta})$ used for the polynomial approximation will be the Hermite Polynomials (see Table 2.3). Figure 3.1 illustrates the setup for the Uncertain Lambert Problem.

To solve the uncertain Lambert problem, we use the least squares solution developed in Chapter 2 to compute higher order sensitivity matrices of the deterministic Lambert Problem $f(\mathbf{x}^*)$ which approximates the solution to the stochastic mapping function $f(\mathbf{x})$. For each test case, the uncertain Lambert solution will be approximated as a $d^{th}$ order polynomial in terms of the standardized random variable $\zeta$

$$y_j \approx c_{j,0} + c_{j}^{(1)T} \phi(\boldsymbol{\zeta})^{(1)} + \ldots + c_{j}^{(d)T} \phi(\boldsymbol{\zeta})^{(d)}$$  \hspace{1cm} (3.5)

To compute the least squares coefficients, the CUT8 points and weights for $n = 6$
Recall that because the basis functions (Hermite Polynomials) are orthogonal
\( \langle \phi_i, \phi_j \rangle = 0 \) for \( i \neq j \), so the coefficients can be evaluated as:

\[
    c_{j,i} = \frac{\langle y_j(x), \phi(\zeta)_i \rangle_{\rho}}{\langle \phi(\zeta)_i, \phi(\zeta)_i \rangle_{\rho}}
\]

Figure 3.2 illustrates the overall structure of the polynomial approximation method,
and the process involved in solving for the approximation coefficients. For more
details on computing the approximation coefficients refer to Chapter 2.

Test case 1 is a near-circular LEO orbit with a radius of \( |r_1| \approx 7000 \text{km} \) and
inclination \( i = 2^\circ \). The nominal initial and final state vectors for case 1 correspond
to a time of flight \( dt = 1200 \text{s} \) and are shown in Table 3.1. This test case is replicated
from Ref. [29] for comparison purposes. The pdf prescribed to both the initial and
final position vectors are Gaussian distributions with equal $\sigma^2 = 0.01 \text{km}^2$ variances in the x, y, and z-directions. This corresponds to a position uncertainty standard deviation of 100m in each dimension.

$$
\Sigma = \begin{bmatrix}
0.01 & 0 & 0 \\
0 & 0.01 & 0 \\
0 & 0 & 0.01 \\
\end{bmatrix} \text{km}^2
$$

The second test case simulation is for a sample Geostationary Transfer Orbit (GTO) to demonstrate the utility of this method in orbits other than LEO. The assumed GTO orbit is an equatorial transfer ($i = 0^\circ$) between the LEO altitude in test case 1 (610.6 km) and GEO altitude (35786 km). The final position vector is at a true anomaly of $5^\circ$ before the apoapsis ($\theta = 175^\circ$). The time of flight for test case 2 is $dt = 16936.7$s. The nominal state vectors for test case 2 are summarized in Table 3.2. The pdf prescribed to both the initial and final position vectors is the same as in test case 1; Gaussian with 100m standard deviations ($\Sigma = 0.01 \times I_{6 \times 6}$).

### 3.2.1 Simulation Results

This section will present the results of each simulation and discuss their significance. First, we will consider the LEO case, and look into the results of using Monte Carlo Methods to calculate the covariance matrix $\Sigma$ and the mean vector $\mu$ for the joint
distribution \( \rho(\zeta) \), and the extent to which this methods can help us validate our solution. The covariance matrix, statistical moments, and mean state vector for the LEO case will then be calculated using the CUT method and directly compared to the results for this test case in Ref. [29]. For each test case, the higher order sensitivity matrices for orders 1-3 will be determined and used to calculate the error \( \epsilon \) at randomly sampled points in the distribution \( \rho(\zeta) \).

Monte Carlo sampling is often used as a method to approximate the solution to non-linear dynamical problems. In Ref. [29], the authors use Monte Carlo methods to validate their solution to test case 1 using both a differential correction and a linear variational analysis, noting that the convergence of the Monte Carlo method is restricted due to sampling limitations. For the current LEO test case, we will demonstrate the difficulties associated with Monte Carlo sampling to reproduce expectation values of the initial position and velocity vectors. To determine these expectation values numerically, points in the initial and final position space are sampled and a Lambert Problem is solved for \( \mathbf{v}_1 \) at each point.

We seek to compare both the mean \( \mu \) and covariance \( \Sigma \) of the input space calculated using a Monte Carlo method and using the CUT8 method. Consider the mean Eq. (3.9a) and variance Eq. (3.9b) approximated using \( N \) Monte Carlo Points, \( \mathbf{x}_i \), with \( \mathbf{r}_1 \) sampled from the true distribution \( \rho(\zeta) \), and \( \mathbf{v}_1 \) calculated by solving the corresponding Lambert Problem:

\[
\mu = \int_{-\infty}^{\infty} \mathbf{x} \rho(\zeta) d\zeta \approx \mu_{MC} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \quad (3.9a)
\]
Figures 3.3(a) and 3.3(b) show the RMS error in mean $\epsilon_\mu = RMS(\mu - \mu_{MC})$ and the matrix norm of the error in covariance: $\epsilon_\Sigma = \|\Sigma - \Sigma_{MC}\|$ . The RMS error in mean and covariance is plotted for varying values of $N$, as well as the error for each computed using CUT8 method. Observing Figures 3.3(a) and 3.3(b) we see an approximate trend of one decimal place of error reduction for every additional factor of 100 sample points. This steep computational cost for improved accuracy illustrates the limitations of the Monte Carlo method and how the need for high accuracy solutions can quickly conflict with the practical limitations of on-board computational capabilities. Contrasting the error in Monte Carlo approximation with the error computed using the CUT8 method (shown as a large red dot in Figures 3.3(a) and 3.3(b)) , there is a very clear advantage to CUT8 which gives both greater accuracy and drastically reduced computational load, requiring only 745 points. Due to the computationally expensive nature of solving a lambert problem to get a single point for the solution space and the relative inaccuracy of the Monte Carlo method, only the CUT8 method will be considered from this point on.

In addition to the mean and covariance, we will also calculate higher order statistical central moments of the initial position and initial velocity space. We will only concern ourselves with the diagonal terms of these central moments, which
can be calculated with Eq. 3.10:

\[ E[(x_j - E[x_j])^q] = \int_{\Omega} (x_j - x_j^*)^q \rho(\zeta) d\zeta = \sum_{i=1}^{N} w_i (x_{i,j} - x_j^*)^q \]  \hspace{1cm} (3.10)

Where \( E[(x_j - E[x_j])^q] \) is the \( q^{th} \) central moment. The results of the 1st raw moment \( x^* \) calculated using Eq. 3.9a, and the 2\textsuperscript{nd} – 4\textsuperscript{th} central moments calculated using Eq 3.10 are shown in Table 3.3. Looking at the first three rows of Table 3.3 we can see that the mean and variance exactly replicate the mean and variance prescribed to the initial position vector \( \mathbf{r}_1 \). We can also see that the skewness of both \( \mathbf{r}_1 \) and \( \mathbf{v}_1 \) are numerically very close to zero, indicating that both variables are symmetrically distributed about the mean vector. The variance of the \( \mathbf{v}_1 \) vector is relatively small for the given initial and final position distribution (10\textsuperscript{−8} km/s), however it is larger than the 3\textsuperscript{rd} and 4\textsuperscript{th} central moments by at least 7 orders of magnitude. The fact that the first and second moments of \( \mathbf{v}_1 \) are the only central moments with significant magnitude demonstrates that the solution for initial velocity is Gaussian in nature. To directly compare the CUT8 results with the results in Ref. [29], we must calculate the covariance matrix for \( \mathbf{r}_1 \) and \( \mathbf{v}_1 \). This covariance matrix can be calculated by evaluating the deterministic Lambert problem at the CUT points, and using Eq. (3.11) for \( \mathbf{x} = [\mathbf{r}_1, \mathbf{v}_1]^T \).

\[ \Sigma = \int (\mathbf{x} - \mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*)^T \rho(\zeta) d\zeta = \sum_{i=1}^{N} w_i (\mathbf{x}_i - \mathbf{x}^*)(\mathbf{x}_i - \mathbf{x}^*)^T \]  \hspace{1cm} (3.11)

As expected, the covariance matrix given in Ref. [29] Table 3 matches the Covariance calculated using the CUT8 method in Table 3.4 almost exactly to the precision given. Notably, in Ref. [29] the off-diagonal terms are assumed to be uncorrelated.
Table 3.4. Test Case 1: Covariance $\Sigma$ Using CUT8 Method

<table>
<thead>
<tr>
<th>$x_1$ (km)</th>
<th>$y_1$ (km)</th>
<th>$z_1$ (km)</th>
<th>$\dot{x}_1$ (km/s)</th>
<th>$\dot{y}_1$ (km/s)</th>
<th>$\dot{z}_1$ (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000E-02</td>
<td>-1.084E-19</td>
<td>-6.776E-21</td>
<td>-7.615E-06</td>
<td>5.365E-06</td>
<td>1.866E-07</td>
</tr>
<tr>
<td>-6.776E-21</td>
<td>1.000E-02</td>
<td>6.098E-20</td>
<td>5.365E-06</td>
<td>-1.182E-05</td>
<td>-3.063E-07</td>
</tr>
<tr>
<td>-6.776E-21</td>
<td>-4.065E-20</td>
<td>1.000E-02</td>
<td>1.865E-07</td>
<td>-3.062E-07</td>
<td>-3.037E-06</td>
</tr>
<tr>
<td>-7.615E-06</td>
<td>5.365E-06</td>
<td>1.865E-07</td>
<td>1.444E-08</td>
<td>-6.507E-09</td>
<td>-2.271E-10</td>
</tr>
<tr>
<td>5.365E-06</td>
<td>-1.182E-05</td>
<td>-3.062E-07</td>
<td>-6.507E-09</td>
<td>2.497E-08</td>
<td>3.987E-10</td>
</tr>
<tr>
<td>1.865E-07</td>
<td>-3.062E-07</td>
<td>-3.037E-06</td>
<td>-2.271E-10</td>
<td>3.987E-10</td>
<td>1.353E-08</td>
</tr>
</tbody>
</table>

Table 3.5. Test Case 1: 1st Order Approximation Coefficients

<table>
<thead>
<tr>
<th>$x_1$ (km)</th>
<th>$y_1$ (km)</th>
<th>$z_1$ (km)</th>
<th>$\dot{x}_1$ (km/s)</th>
<th>$\dot{y}_1$ (km/s)</th>
<th>$\dot{z}_1$ (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.499E+00</td>
<td>-1.468E+00</td>
<td>-5.104E-02</td>
<td>1.603E+03</td>
<td>-5.019E+02</td>
<td>-1.737E+01</td>
</tr>
<tr>
<td>-9.527E-01</td>
<td>1.676E+00</td>
<td>4.894E-02</td>
<td>-3.735E+02</td>
<td>1.250E+03</td>
<td>1.278E+01</td>
</tr>
<tr>
<td>-3.307E-02</td>
<td>4.888E-02</td>
<td>2.708E-01</td>
<td>-1.294E+01</td>
<td>1.275E+01</td>
<td>8.835E+02</td>
</tr>
<tr>
<td>1.899E-03</td>
<td>-3.013E-03</td>
<td>-1.045E-04</td>
<td>2.598E+00</td>
<td>-1.126E+00</td>
<td>-3.888E-02</td>
</tr>
<tr>
<td>-3.205E-05</td>
<td>3.846E-05</td>
<td>-1.028E-03</td>
<td>-1.516E-02</td>
<td>1.377E-02</td>
<td>2.343E-01</td>
</tr>
</tbody>
</table>

and set to zero, however, in evaluating the covariance matrix numerically we make no assumptions about correlation. In light of this, it is reassuring to see that the off-diagonal terms of the covariance matrix are indeed very small, further assuring us that the CUT8 method works properly.

Now that the CUT8 method has been shown to be effective in reproducing the results of both Monte Carlo and linear variational methods given in Ref. [29], the CUT method can be used to compute the inner products required to determine the least squares coefficients. The coefficients for the first order sensitivity coefficients are in many ways analogous to the State Transition Matrix used in linear variational methods. The State Transition Matrix given in Ref. [29] Table 2 matches the 1st order least squares coefficients given in Table 3.5 to a precision of roughly three decimal places. This demonstrates an approximate equivalency of the two methods. The results for the polynomial approximation to the Lambert Problem using higher order sensitivity matrices will now be presented for both test cases 1 and 2.

3.2.2 Test Case 1: LEO

Coefficients for the initial velocity approximation are calculated for polynomial orders 1-3, and the magnitudes of the coefficient for polynomial order approximation
are summarized in Figure 3.4. $C_{V_x}, C_{V_y},$ and $C_{V_z}$ are the RMS values for all elements in the 1st, 2nd, and 3rd row of coefficients in each $c_j^{(d)}$ matrix (see Eq. (2.10)). As the order of the polynomial approximation increases the order of magnitude of the coefficients decreases, indicating that the higher orders are contributing less to the approximation of $y$.

10,000 random points $x_i$ were sampled from the normalized distribution $\rho(\zeta)$, and used to compute the polynomial basis vector $\phi(\zeta)$. The approximation for $v_1$ can then be calculated at each sample point, and the error can be found using Eq. (2.16). The same 10,000 points are used for each polynomial approximation order shown. Figure 3.5 shows 3D scatter plots of the initial velocity with ECI position coordinates on the x,y, and z axes. Figure 3.5(a) shows the true velocity magnitude on the color axis, and Figures 3.5(b)-3.5(d) show initial velocity error on the color axis.

The same data is presented in figure 3.6 as a 2D contour plot where the x and y axes are the Mahalanobis Distance of initial and final position vectors respectively, and the colorbar is the same for that in Figure 3.5(a). The Mahalanobis Distance $D_i$ is a the multi-dimensional analog to the standard deviation, and describes how far a point is from the mean vector $x^*$. $D_1$ and $D_2$ are measurements of how far a sample point is from $r_i^*$ and $r_2^*$ respectively, and are calculated as follows:

$$D_i = \sqrt{(r_i - r_i^*)^T \Sigma^{-1} (r_i - r_i^*)}$$  (3.12)
Observing Figures 3.5 and 3.6, there is an obvious trend of decreasing error as polynomial approximation order increases, perhaps most drastically between 1\textsuperscript{st} and 2\textsuperscript{nd} order. In 3.6(b) it is notable that the error contours tend to monotonically increase as mahalanobis distance increases. Intuitively it makes sense that the linear 1\textsuperscript{st} order approximation drops in accuracy as the point moves farther away from the mean. Contrasting this trend with higher order approximations, the error decreases and then increases again as mahalanobis distance increases (see Figure 3.6(d)). The pattern shown in both 2nd and 3rd order approximations can be interpreted as the solution for \( v_1 \) having a near Gaussian distribution, because it is very accurately approximated by a second order approximation. This makes intuitive sense, because the input was Gaussian and due to a relatively short propagation time, there was not ample time for non-linearities to be introduced.
The RMS value for initial velocity error at the random samples for each polynomial approximation order are summarized in Table 3.6. Inclusion of HOT’s in the 2nd and 3rd order approximations shows a clear advantage over the 1st order approximation by lowering the RMS error by more than 3 orders of magnitude, and smoothing the error distribution contours (see Figures 3.6(c) and 3.6(d)).

**Table 3.6. Test Case 1: Initial Velocity Error**

<table>
<thead>
<tr>
<th>Polynomial Order</th>
<th>RMS($\epsilon_{V_1}$) (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Order</td>
<td>6.4256E-9</td>
</tr>
<tr>
<td>2nd Order</td>
<td>3.1885E-12</td>
</tr>
<tr>
<td>3rd Order</td>
<td>3.1881E-12</td>
</tr>
</tbody>
</table>
3.2.3 Test Case 2: GTO

The results of test case 2 will now be presented. The least squares coefficients for orders 1-3 are computed in the same manner as in test case 1. Figure 3.7 depicts the magnitude of coefficients $C_{V_x}$, $C_{V_y}$, and $C_{V_z}$ for approximation orders 1-3. Notice that the decreasing trend in coefficient magnitude as approximation order increases is similar in this test case as in test case 1.

A different 10,000 random samples are taken from $\rho(\zeta)$ and used to compute the approximated $v_1$ as well as error in $v_1$. The results are plotted in Figures 3.8 as a 3D scatter plot and in Figure 3.9 as a 2D Mahalanobis Distance contour plot. Additionally, the RMS error values for polynomial approximation orders 1-3 are given in Table 3.7.

Comparing Figure 3.8(b) with it’s counterpart in test case 1, Figure 3.5(b), it is noticeable that there is larger error in the 1st order approximation for test case 2 than for test case 1. Despite the larger error for the 1st order approximation relative to the LEO case, adding higher order terms still significantly decreases the approximation error to the point where the 3rd order approximation errors for both cases are the same order of magnitude. Table 3.7 shows the RMS error for various polynomial approximations of test case 2. This result shows that regardless of what the error is in the first order approximation, the error can always be reduced by including HOT’s. Another interesting insight of this result is that when
the time of flight for the Lambert Problem is increased the non-linear dynamics have a larger influence on the solution. For example, in test case one the time of flight is $dt = 1200s$ and in test case two the time of flight is $dt = 16936.7s$. After including the 2nd order approximation term in test case 1, the solution does not really improve by much because the non-linearities are not as influential as in test case two, where the solution needs to include the third order term to achieve the same level of accuracy.
Figure 3.9. Test Case 2: $V_1$ Mahalanobis Distance Representation
Chapter 4  |  The Reachability Set Problem

4.1 Problem Background

Uncertainty in all aspects of satellite operations including state measurements, dynamic modeling, physical parameters, and maneuver capabilities is a persistent problem that raises issues in many areas of Astrodynamics and Space Situational Awareness. The concept of reachability is well studied in mathematics, dynamics, optimal control theory, and game theory [37–43] and involves determination of all possible states of a dynamic system at some future time based on a set of initial conditions, constraints and uncertainties. Despite the rich history of reachability theory in these fields, there are few investigations into the application of reachability sets to SSA problems.

Reachability sets have important implications on SSA, and can provide us insight on creating efficient and accurate methods for sensor tasking, collision assessment, and data association. Imagine, for instance, the scenario of an orbit insertion maneuver. There are uncertainties associated not only with the initial position and velocity, but also with the attitude and magnitude of the maneuver itself. If the uncertainty is high enough that we cannot track the satellite post-maneuver, then knowledge of the reachability set and probability distribution of the future positions of the satellite would be very useful for finding an appropriate search space.

The use of reachability sets for SSA applications was first introduced by Holzinger and Scheeres in 2009 [44] where reachability theory was applied proximity operations situations using the linearized Clohessy-Wiltshire relative equations of motion to
provide insight into proximity operations. More recently, reachability sets for
maneuvering spacecraft [45–47] as well as trajectory design for non-keplerian
systems [48] have been investigated.

Due to the computational burden associated with propagating non-linear orbit
dynamics and the sheer number of propagation points required to accurately
capture higher order statistical moments, computing accurate reachability sets
has traditionally been a practical impossibility. The CUT approach to computing
higher order sensitivity matrices proposed in this thesis provides the tool necessary
to efficiently compute the matrices, enabling previously unavailable speed and
accuracy.

4.2 Simulation Description

The objective of the reachability set problem is to find the set of all possible outputs
$Y$ that can be reached at the final time $t_f$ given uncertain initial state $x$ and control
$u$ in the admissible control set $U$. Thus, the reachability set can be defined as:

$$Y(x_0, U, t_f) = \{ \forall x \in \mathbb{R}^n | u \in U, s.t. \ x(t_f) = x \}$$  \hspace{1cm} (4.1)

The underlying mathematical framework of this problem is the same as that in the
previous chapter, with the main difference being the inputs $x$, outputs $y$ and the
nonlinear function $f(x)$. In the Reachability Set Problem, $x$ is considered to be
the initial position and velocity of the satellite, $y$ is the final position and velocity,
and $f(x)$ is the dynamic model.

$$x = \left\{ \begin{array}{l} r_1 \\ v_1 \end{array} \right\}, \quad y = \left\{ \begin{array}{l} r_2 \\ v_2 \end{array} \right\}, \quad u = \left\{ \begin{array}{l} \Delta v \\ \theta \\ \psi \end{array} \right\}$$  \hspace{1cm} (4.2)

Each simulation discussed is comprised of periods of coasting ($u = 0$) separated by
impulsive $\Delta v$ maneuvers (where $\theta$ and $\psi$ are the attitude angles). For the examples
in this chapter, the function $f(x^*, u^*)$ represents the deterministic propagation of
two body dynamics in between impulsive thrust maneuvers.

\[ \dot{\mathbf{r}} = -\frac{\mu \mathbf{r}}{|\mathbf{r}|^3}, \quad f(\mathbf{x}, \mathbf{u}) = \int_{t_1}^{t_2} \dot{\mathbf{x}} + g(\mathbf{u}_k, t_k) dt \]  

(4.3)

Where \( \mu \) represents the gravitational parameter of the central body, and \( g(\mathbf{u}_k, t_k) \) represents the impulsive maneuvers. To compute \( f(\mathbf{x}^*, \mathbf{u}^*) \) for deterministic variables \( \mathbf{x}^*, \mathbf{u}^* \), the Runge-Kutta45 method is used to numerically integrate \( \dot{\mathbf{x}}^* \) via Matlab’s in-built ode45 command. We seek to compute the statistical mapping for the uncertain state vector \( \mathbf{x} \) which gives the reachability set.

\[ \mathbf{y} = f(\mathbf{x}, \mathbf{u}) \]  

(4.4)

To solve the reachability set problem, we use the least squares solution developed in Chapter 2 to compute higher order sensitivity matrices of each simulation with respect to the uncertain variables. For each test case, the reachability set will be approximated as a \( d^{th} \) order polynomial in terms of the standardized random variable \( \zeta \).

\[ y_j \approx c_{j,0} + c_{j}^{(1)} T \phi(\zeta)^{(1)} + \ldots + c_{j}^{(d)} T \phi(\zeta)^{(d)} \]  

(4.5)

To compute the least squares coefficients for the \( n \) dimensional uncertain inputs, the CUT8 points and weights are used to numerically evaluate the normal equations

\[ \mathbf{A} = \mathbf{CB} \]  

(4.6)

\[ A_{j,k} = \langle \mathbf{y}_j(\mathbf{x}), \phi_k \rangle_{\rho}, \quad B_{i,k} = \langle \phi_i, \phi_k \rangle_{\rho} \]

Recall that because the basis functions (Hermite Polynomials) are orthogonal \( \langle \phi_i, \phi_j \rangle = 0 \) for \( i \neq j \), so the coefficients can be evaluated as:

\[ c_{j,i} = \frac{\langle \mathbf{y}_j(\mathbf{x}), \phi_i \rangle_{\rho}}{\langle \phi_i, \phi_i \rangle_{\rho}} \]  

(4.7)

Figure 3.1 illustrates the problem setup for the Satellite Reachability Problem.

Two test cases taken from literature will be considered in the chapter. The first test case is a two burn maneuver based on the trajectory in Ref. [49] which takes a satellite from a LEO parking orbit into a specified orbit. The second test case is based on the Lunar Orbit Insertion (LOI) trajectory of the Korea Pathfinder Lunar
Orbiter mission described in Song et al [50]. Both test cases assume impulsive uncertain maneuvers.

### 4.2.1 Test Case 1: Two Burn Maneuver

This test case is based on Betts [49] which determines the optimal three burn transfer orbit from a set LEO parking orbit to a final operations orbit of a specific inclination. Betts’ paper considers three burn transfers to several target orbits with varying inclinations, however this test case will only look at the first example with target $i = 63.4^\circ$. Additionally test case 1, only considers the first two burns for the nominal optimal transfer orbit given [49], and the end of the simulation is taken to be the position where the third burn would occur. The parking orbit is a circular LEO with $i = 37.4^\circ$, and the first burn is applied at an argument of latitude $u = 255^\circ$. The RAAN is not important for this analysis and is set to zero for simplicity. The nominal orbital elements for the coasting transfer orbits are given in Table 4.1.

The nominal maneuvers are modelled as impulsive $\Delta v^*$ burns applied at the nominal yaw and pitch angles $\psi^*$ and $\theta^*$ respectively. To compute the post burn velocity, $\Delta v$ is computed with respect to the velocity, cross-track, orbit-normal (VCN) frame and transformed to the inertial frame. A diagram of pitch and yaw
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parking Orbit</th>
<th>1st Transfer</th>
<th>2nd Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^*(km)$</td>
<td>6667.32</td>
<td>22,835.4</td>
<td>29,243.5</td>
</tr>
<tr>
<td>$e^*$</td>
<td>0</td>
<td>0.7080</td>
<td>0.6785</td>
</tr>
<tr>
<td>$i^*(^\circ)$</td>
<td>37.40</td>
<td>35.78</td>
<td>57.88</td>
</tr>
<tr>
<td>$\omega^*(^\circ)$</td>
<td>N/A</td>
<td>254.0</td>
<td>259.6</td>
</tr>
</tbody>
</table>

Table 4.1. Test Case 1: Nominal Transfer Orbit Parameters

angles in the VCN frame is shown in Figure 4.2. The $\Delta v$ in the VCN frame can be

\[
\Delta v^{vcn} = |\Delta v| \begin{bmatrix} \cos \theta \cos \psi \\ \cos \theta \sin \psi \\ -\sin \theta \end{bmatrix}
\]  

(4.8)

The post maneuver velocity in the ECI frame $v_1$ is then calculated to be

\[
v_1 = T \begin{bmatrix} |v_0| \\ 0 \\ 0 \end{bmatrix} + \Delta v^{vcn}
\]

(4.9)

Where $T$ is the rotation matrix which maps the VCN frame to the ECI frame. The parameters of the first and second burn as well as the location that each burn occurs is summarized in Table 4.2.

It is assumed that the initial position and velocity are deterministic and known based on the given parking orbital elements and argument of latitude at the 1st burn. The uncertain variables are the maneuver magnitudes and attitude angles. It is assumed that the pitch and yaw angles as well as the $\Delta V$ magnitude are normally
Table 4.2. Test Case 1: Nominal Burn Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>1st Burn</th>
<th>2nd Burn</th>
<th>Final Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>Δv^+</td>
<td>(m/s)$</td>
<td>2,383.5</td>
</tr>
<tr>
<td>$θ^∗(°)$</td>
<td>-1.36</td>
<td>7.21</td>
<td>N/A</td>
</tr>
<tr>
<td>$ψ^∗(°)$</td>
<td>5.99</td>
<td>-74.7</td>
<td>N/A</td>
</tr>
<tr>
<td>$u^∗(°)$</td>
<td>255</td>
<td>37.7</td>
<td>121.7</td>
</tr>
<tr>
<td>$t(s)$</td>
<td>0</td>
<td>6,078.2</td>
<td>35,609.0</td>
</tr>
</tbody>
</table>

Figure 4.3. Test Case 1: Nominal Orbit

distributed random variables with mean given by the nominal parameters given in Table 4.2, and standard deviations given by Table 4.3.

Table 4.3. Nominal Maneuver Uncertainties

<table>
<thead>
<tr>
<th>Variable</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$σ_{Δv_1}$</td>
<td>5 (m/s)</td>
</tr>
<tr>
<td>$σ_{θ_1}$</td>
<td>$2^°$</td>
</tr>
<tr>
<td>$σ_{ψ_1}$</td>
<td>$2^°$</td>
</tr>
<tr>
<td>$σ_{Δv_2}$</td>
<td>5 (m/s)</td>
</tr>
<tr>
<td>$σ_{θ_2}$</td>
<td>$2^°$</td>
</tr>
<tr>
<td>$σ_{ψ_2}$</td>
<td>$2^°$</td>
</tr>
</tbody>
</table>
4.2.2 Test Case 2: Lunar Orbit Insertion

This section will provide details on the various parameters and assumptions made for the 2nd reachability test case. The current test case simulates a Lunar Orbit Insertion (LOI) maneuver based on 1st orbit insertion maneuver of the Korea Pathfinder Lunar Orbiter mission described in Song et al [50]. A similar example can be found in Houghton et al [51]. It is assumed that the orbiter approaches the moon on a hyperbolic arrival trajectory with \( i = 90^\circ \) such that the satellite can enter into a polar orbit. The nominal LOI maneuver is planned to be executed at the periapsis of the arrival orbit such that the target orbit has a period of 12 hours. A diagram showing the geometry of the arrival trajectory and insertion maneuver is shown in Figure 4.4. The nominal maneuver is planned to occur in the direction opposite the velocity vector at periapse and follows the relationship:

\[
\begin{align*}
\mathbf{v}_1^* &= \mathbf{v}_{hyp}^* + \mathbf{\Delta v}^* \\
\mathbf{r}_1^* &= \mathbf{r}_{hyp}^*
\end{align*}
\]  

Figure 4.4. Test Case 2: Diagram of Arrival Geometry
Table 4.4. Test Case 2: Nominal Orbit Elements

<table>
<thead>
<tr>
<th>Orbital Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-Major Axis ($a_0$)</td>
<td>6143.4 (km)</td>
</tr>
<tr>
<td>Eccentricity ($e_0$)</td>
<td>0.6847</td>
</tr>
<tr>
<td>Inclination ($i_0$)</td>
<td>90°</td>
</tr>
<tr>
<td>Argument of Periapse ($\omega_0$)</td>
<td>0°</td>
</tr>
<tr>
<td>RAAN ($\Omega_0$)</td>
<td>0°</td>
</tr>
<tr>
<td>True Anomaly ($\nu_0$)</td>
<td>0°</td>
</tr>
<tr>
<td>Period ($P$)</td>
<td>12 hr</td>
</tr>
</tbody>
</table>

The semi-major axis ($a_0$) and eccentricity ($e_0$) of the target orbit are defined by the relationships:

$$a_0 = \left( \frac{(P/2\pi)^2}{\mu} \right)$$  \hspace{1cm} (4.11)

$$e_0 = 1 - \frac{R_m + h_p}{a_0}$$  \hspace{1cm} (4.12)

Where $P$ is the orbital period, $\mu$ is the standard gravitational parameter of the Moon, $R_m$ is the radius of the Moon, and $h_p$ is the altitude above the lunar surface at periapse. It is assumed that the parameters $i_0$, $\omega_0$, $\Omega_0$, $\nu_0$, $h_p$ and hyperbolic arrival velocity at periapsis $v_{hyp}$ are design parameters that can be specified based on mission objectives. The current analysis will use the same target orbit as in Ref. [50], with $h_p = 200$ km, $|v_{hyp}^*| = 2.4$ km/s, time of maneuver $t_i = 0$ s and all other targeted orbit parameters shown in Table 4.4. The values given represent the nominal orbit of the spacecraft immediately after the burn. For test case 1, uncertainty will be accounted for in the (3×1) Cartesian pre-maneuver position and velocity vectors $r_{hyp}$ and $v_{hyp}$, as well as in the (2×1) attitude of the maneuver $\theta$, $\psi$ and the scalar magnitude of the maneuver vector $|\Delta v|$. Figure 4.2 illustrates the geometry of the orbit insertion maneuver. Note that all input variables are assumed to be normally distributed random variables and are represented in the VCN reference frame. This allows us to specify more realistic uncertainties (greater in the velocity direction and less in the orbit normal and cross track directions). The input variables are listed in Table 4.5, and the nominal orbit is shown in Figure 4.5.

To reduce the number of variables, we combine the arrival velocity and maneuver using Eq. (4.10). We must now determine the post maneuver velocity $\mathbf{v}_1^{*\text{vcn}}$ mean
Figure 4.5. Test Case 2: Nominal LOI Maneuver

Table 4.5. Test Case 2: Input Uncertainty

<table>
<thead>
<tr>
<th>Variable</th>
<th>Direction</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>( \hat{e}_v )</td>
<td>( \sigma_v^{(pos)} )</td>
</tr>
<tr>
<td>Position</td>
<td>( \hat{e}_c )</td>
<td>( \sigma_c^{(pos)} )</td>
</tr>
<tr>
<td>Position</td>
<td>( \hat{e}_n )</td>
<td>( \sigma_n^{(pos)} )</td>
</tr>
<tr>
<td>Velocity</td>
<td>( \hat{e}_v )</td>
<td>( \sigma_v^{(vel)} )</td>
</tr>
<tr>
<td>Velocity</td>
<td>( \hat{e}_c )</td>
<td>( \sigma_c^{(vel)} )</td>
</tr>
<tr>
<td>Velocity</td>
<td>( \hat{e}_n )</td>
<td>( \sigma_n^{(vel)} )</td>
</tr>
<tr>
<td>Attitude</td>
<td>N/A</td>
<td>( \sigma_\theta )</td>
</tr>
<tr>
<td>Attitude</td>
<td>N/A</td>
<td>( \sigma_\psi )</td>
</tr>
<tr>
<td>Burn Magnitude</td>
<td>N/A</td>
<td>( \sigma_{</td>
</tr>
</tbody>
</table>

and covariance \((\Sigma)\) in the VCN frame.

\[
\begin{align*}
\mathbf{v}_{1}^{\text{vcn}} &= E[\mathbf{v}_{1}^{\text{vcn}}] = E[\mathbf{v}_{\text{hyp}}^{\text{vcn}} + \Delta \mathbf{v}^{\text{vcn}}] = \mathbf{v}_{\text{hyp}}^{\text{vcn}} + \Delta \mathbf{v}^{\text{vcn}} \\
\Sigma_{\mathbf{v}_{1}} &= E[(\mathbf{v}_{1}^{\text{vcn}} - \mathbf{v}_{1}^{\text{vcn}})(\mathbf{v}_{1}^{\text{vcn}} - \mathbf{v}_{1}^{\text{vcn}})^T] = \Sigma_{\mathbf{v}_{\text{hyp}}} + \Sigma_{\Delta \mathbf{v}}
\end{align*}
\]  \hspace{1cm} (4.13)

The mean hyperbolic arrival velocity \( \mathbf{v}_{\text{hyp}}^{\text{vcn}} = 2.4 km/s \), and the mean maneuver vector \( \Delta \mathbf{v}^{\text{vcn}} \) can be calculated using the orbital elements for the target orbit. The covariance matrix for the hyperbolic arrival velocity \( \Sigma_{\mathbf{v}_{\text{hyp}}} \) is specified, however the covariance matrix for the maneuver in VCN frame \( \Sigma_{\Delta \mathbf{v}} \) is not directly specified.
from the burn magnitude and attitude uncertainties so we must calculate it. The attitude model for this test case is the same as the one used for test case 1 Eq. (4.8) and (4.9). Using this relationship we can use the CUT8 numerical integration scheme to calculate the covariance matrix in the VCN frame.

\[
\Sigma_{\Delta v} = \int (\Delta v^{vcn} - \Delta v^{*vcn})(\Delta v^{vcn} - \Delta v^{*vcn})^T \rho(\zeta) d\zeta
\]

\[
= w_i(\Delta v_i^{vcn} - \Delta v^{*vcn})(\Delta v_i^{vcn} - \Delta v^{*vcn})^T
\]

Where \( w_i \) are the CUT8 weights and \( \Delta v_i^{vcn} \) isThe post maneuver velocity covariance matrix \( \Sigma_i^{(vcn)} \) can now be calculated using Eq. (4.13). Assume that matrix \( T \) is the rotation matrix which maps the VCN frame to the Moon Centered Inertial (MCI) frame. To determine the mean velocity as well as position and velocity covariances in the MCI frame, the following relations are used:

\[
v^{*mci} = Tv^{*vcn}
\]

\[
\Sigma_i^{(mci)} = T\Sigma_i^{(vcn)}T^T
\]

\[
\Sigma_{r_1}^{(mci)} = T\Sigma_{r_1}^{(vcn)}T^T
\]

Now we have complete representations for the mean and covariance of the state vector in MCI frame:

\[
x = \left\{ \begin{array}{c} \mathbf{r}_1 \\ \mathbf{v}_1 \end{array} \right\}, \quad \Sigma_x = \begin{bmatrix} \Sigma_{r_1}^{(mci)} & 0_{3\times3} \\ 0_{3\times3} & \Sigma_{v_1}^{(mci)} \end{bmatrix}, \quad x^* = \left\{ \begin{array}{c} \mathbf{r}_1^{*mci} \\ \mathbf{v}_1^{*mci} \end{array} \right\}
\]

Using this framework we will examine 2 different scenarios: 1) Low Non-Linearity (LNL) Scenario, and 2) a High Non-Linearity (HNL) Scenario. For LNL, the final time of the simulation will be 30 minutes after the maneuver is performed, and for HNL, the final time will be 6 hours after the maneuver. The uncertainty conditions for low and high non-linearity cases are shown below in Table 4.6.

### 4.3 Results

The following section will discuss the results of both the two burn transfer orbit, and the LOI maneuver test cases. Both test cases use the CUT8 method to calculate the coefficients for polynomial approximation orders 1 through 4, then use the
coefficients to calculate a reachability set of 100,000 randomly sampled points. After computing the approximation coefficients, 5000 Monte Carlo points are randomly sampled from the distribution and propagated fully throughout the simulation. The error between the polynomial approximation and the fully propagated MC samples is computed to determine the accuracy of the approximation.

4.3.1 Test Case 1: Two Burn Maneuver

The following plots depict the results of the for test case 1. Figure 4.6 shows the evolution of the reachability set over time using the most accurate (4th order) sensitivity matrix. Each color in Figure 4.6 represents all possible positions of the satellite at a given time. It is apparent that over time and with the addition of uncertainty in the 2nd burn, the reachability set expands considerably in size. Figures 4.7 and 4.8 depicts the error at final time for varying polynomial approximation orders for 5000 random samples. Figure 4.7 is a scatter plot with the final satellite position \( r_2 \) plotted on the X,Y,Z axes and the final position error \( \epsilon \) is shown on a log scale in the color bar. Figure 4.8 plots the Mahalanobis Distance of the initial samples for \( r_1 \) on the x axis, \( v_1 \) on the y axis, and shows the final position error. Table 4.7 shows the RMSE for position and velocity at varying polynomial approximation orders. Even at a 4th order approximation, the RMSE in the reachability set is quite large \( \approx 14 \text{ km} \). Although this seems like a large error, considering the error relative to the 1st order error \( \approx 63 \text{ km} \) and the spatial volume of the reachability set (on the order of \( 10,000 \text{km}^3 \)), this error is

<table>
<thead>
<tr>
<th>Variable</th>
<th>LNL</th>
<th>HNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_{\text{pos}}^{v} ) (km)</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>( \sigma_{\text{pos}}^{c} ) (km)</td>
<td>0.1</td>
<td>2</td>
</tr>
<tr>
<td>( \sigma_{\text{pos}}^{n} ) (km)</td>
<td>0.1</td>
<td>2</td>
</tr>
<tr>
<td>( \sigma_{\text{vel}}^{v} ) (m/s)</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>( \sigma_{\text{vel}}^{c} ) (m/s)</td>
<td>0.5</td>
<td>7.5</td>
</tr>
<tr>
<td>( \sigma_{\text{vel}}^{n} ) (m/s)</td>
<td>0.5</td>
<td>7.5</td>
</tr>
<tr>
<td>( \sigma_{\theta} ) (deg)</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td>( \sigma_{\psi} ) (deg)</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td>( \sigma_{</td>
<td>\Delta v</td>
<td>} ) (m/s)</td>
</tr>
</tbody>
</table>
Table 4.7. Test Case 1: Error Summary

<table>
<thead>
<tr>
<th>Approximation Order</th>
<th>Position RMSE (m)</th>
<th>Velocity RMSE (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.294E5</td>
<td>6.294E1</td>
</tr>
<tr>
<td>2</td>
<td>5.251E4</td>
<td>9.432E0</td>
</tr>
<tr>
<td>3</td>
<td>1.619E4</td>
<td>2.682E0</td>
</tr>
<tr>
<td>4</td>
<td>1.454E4</td>
<td>9.810E-1</td>
</tr>
</tbody>
</table>

pretty reasonable. The Mahalanobis distance plots show a similar trend where increasing the approximation order decreases the error, even if the improvement is only marginal. The uncertainty in this simulation is relatively large compared to some of the other cases that will be presented in this chapter, and the roughly 10 hour propagation time allows non-linearities to accumulate in the solution, so the approximation error is correspondingly large.

4.3.2 Test Case 2: Lunar Orbit Insertion Maneuver

This section will present and discuss the results of the LOI maneuver reachability set computation for both HNL and LNL scenarios. The parameters for each can be found in Table 4.6. First the low uncertainty case will be given and then the corresponding results for the high uncertainty case. The same metrics will be used to assess the polynomial approximation as in test case 1.
Figures 4.10 and 4.11 depict the error in position between 5000 fully propagated MC points and sensitivity matrix approximation using scatter plot and Mahalanobis distance representations respectively. Table 4.11 summarizes the position and velocity RMSE at all MC points. Figure 4.9 shows the evolution of the LNL case 4th order reachability set over time. From inspection, we can tell that the accuracy of the polynomial approximation improves as polynomial approximation order increases. For the low uncertainty test case, even the 1st order approximation provides reasonable accuracy. From Table 4.8 we can see that a first order approximation gives us an RMSE of $15m$ in position and $3.7cm/s$ is velocity. This is fairly good accuracy considering the relatively large initial position uncertainties of $1km$. Depending on the accuracy requirements of the reachability set, a 1st order...
approximation may be sufficient. However if greater accuracy is required, increasing
the approximation to 4th order gives us a near exact solution with error of about 10
micrometers. Observing Figure 4.11 it can be seen that as polynomial approximation
order is increased, the variation of error with respect to Mahalanobis distance is
decreased. That is to say, the distribution of error becomes smoothed such that
the error becomes uniform over the entire domain of the approximation. This
phenomenon is similar to that which was observed for the uncertain Lambert
problem. It is also interesting to note that from Figure 4.11 it is clear that in
general the gradient of the error is more dependant of the velocity Mahalanobis
distance than on the position. Intuitively this makes sense, because a variation
in velocity has larger effects on the resulting orbit than a variation in position.
Propagated over time, a small change in velocity can greatly expand the possible
reachable positions.

Next, the results for the high uncertainty test case are presented and discussed.
Figure 4.9. Test Case 2 LNL: 4th Order Reachability Set Over Time

<table>
<thead>
<tr>
<th>Approximation Order</th>
<th>Position RMSE (m)</th>
<th>Velocity RMSE (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.566E1</td>
<td>3.764E-2</td>
</tr>
<tr>
<td>2</td>
<td>8.555E-2</td>
<td>2.997E-4</td>
</tr>
<tr>
<td>3</td>
<td>7.735E-4</td>
<td>3.468E-6</td>
</tr>
<tr>
<td>4</td>
<td>1.044E-5</td>
<td>5.564E-8</td>
</tr>
</tbody>
</table>

Table 4.8. Test Case 2 LNL: Error Summary

The evolution of the 4th order reachability set over time for the high uncertainty test case is given by Figure 4.12. Figure 4.13 plots the spatial distribution of final position error for varying polynomial approximation, and Figure 4.14 plots these errors vs initial position and velocity Mahalanobis distances. The RMSE for final position and velocity are summarized in Table 4.9. Unsurprisingly, the error in the HNL case are much higher than in the LNL case. In this case, adding higher order sensitivity matrices up to 4th order gains almost 2 orders of magnitude in position RMSE when compared with 1st order. Although this is a significant improvement, the 4th order approximation still gives a position RMSE of around 3.8 km. Depending on the application, this might still be too much error. Unfortunately, with the current theory of the conjugate unscented transform method and a 6 dimensional system, we are limited to 4th order approximations. If the desired accuracy is still not met using the 4th order approximation, then either the uncertainty or propagation time must be reduced, or another quadrature method is required to compute the 5th
order and higher sensitivity matrices. However, computing sensitivity matrices of orders greater than 4 with methods other than CUT can be computationally infeasible.

Another interesting trend to notice in the higher order sensitivity method is that the amount that accuracy improves by adding higher order sensitivity matrices, is largely dependant on the magnitude of the error to begin with. For example, test case 1 had the lowest accuracy out of any of the test cases analyzed, with a position RMSE of approximately 630 km for the 1st order approximation. Adding higher order matrices up to 4th order, did decrease the error but only by about 1.5 orders of magnitude. Contrasting this fact with the LNL case of test case 2, we see that the 1st order approximation gives a position RMSE of 1.6 m whereas the 4th order approximation has a position RMSE of $10^{-5}$. This means that adding...
higher order sensitivity matrices to the LNL case of test case 2 decreased the error by roughly 6 orders of magnitude. This is a somewhat undesirable trend which essentially means that the approximations with the highest error gain the least by adding higher order sensitivity matrices.
Figure 4.12. Test Case 2 HNL: 4th Order Reachability Set Over Time

Figure 4.13. Test Case 1 HNL: Spatial Position Error Plot
Figure 4.14. Test Case 1 HNL: Mahalanobis Distance Position Error Plot
Chapter 5  |  Conclusions

5.1 Summary of Current Work

The goal of this thesis was to provide a probabilistic framework for solving the uncertain Lambert problem and the reachability set problem. This objective was accomplished by computing sensitivity matrices, which represent the higher order Taylor series expansion terms of the solution to each problem. Necessarily, the Taylor series must be truncated to a finite order, which provides an approximate solution. In both the uncertain Lambert problem and the reachability set problem, it has been demonstrated that increasing the order of sensitivity matrices decreases the approximation error and expands the domain over which the approximation is valid. The main innovation is derived from the use of non-product quadrature method *Conjugate Unscented Transform (CUT)* to compute higher order sensitivity matrices in a non-intrusive and computationally efficient manner. The CUT method helps in curtailing the *curse of dimensionality* associated with computation of these sensitivity matrices.

Although the developed approach is valid for a generic nonlinear system, the accuracy of the developed approach is dependent upon the size of uncertain domain as well as the inherent nonlinearity of the problem. One will need to judiciously select the order of polynomial approximation depending upon the initial uncertainty and nonlinearity of the problem. The use of orthogonal polynomial is beneficial as calculation of higher order coefficients is independent of calculations of lower order coefficients and hence, these coefficients can be computed recursively to achieve the desired accuracy. Off course, the computational burden associated with
computing higher order sensitivity matrices will increase with increase in the order of polynomial approximation. The CUT method which provides computational efficiency provides $8^{th}$-order quadrature points only up to six dimensions. One may need to generalize the CUT method for higher order approximations or resort to other methods with increased computational burden to sensitivity matrices for polynomial higher than fourth order.

5.2 Furture Research Direction

The mathematical framework laid out in this thesis is a valid approach to approximate any stochastic non-linear function; however, the focus was placed on SSA applications due to the inherent uncertainty present in space operations. The solution to these problems is not meant to be a final product in SSA, but as a tool used in solving other problems. There are two sensible directions in which future research can go: 1) "Upward", in which this method is implemented into more macroscopic application, and 2) "Downward" towards refining the building blocks of the method itself.

First, consider the upward direction. The higher order sensitivity method has far-reaching applications in many fields, however, there are several clear applications in aerospace engineering, and in particular, in SSA. One possibility is to implement this method for collision assessment of two bodies that are predicted to have a close encounter. The reachability problem can be used to provide highly accurate probability distribution functions of each object during the encounter which would enable far more accurate numerical calculation of the probability that these objects will collide. Other possible applications of these problems include implementing this framework in sensor tasking, maneuver detection, and data association algorithms. All of these applications must account for uncertainty, and could greatly benefit from an efficient approach to computing the stochastic solution to the lambert problem and the reachability problem.

The second direction is to go inward. The CUT method is a very powerful tool, but like any numerical integration method, it has limitations. One possible research direction is to expand the CUT method to satisfy even more moment constraints or to determine CUT weights and points for higher dimensional spaces. This would enable even better accuracy in highly non-linear problems. There is
also potential to improve accuracy by looking into the possibility of combining several localized approximations to approximate the entire domain. Using multiple local approximations could have the benefit of increasing accuracy over the entire domain using relatively low order local approximations. Another possibility is to extend the reachability set problem to include continuous control variables. In the analyses provided in this thesis all control inputs were assumed to be impulsive; however, developing the framework to compute reachability sets for finite burns or continuous thrust would greatly expand the applicability of the method.
References


   URL https://www.nap.edu/read/13456/chapter/1


