Application of the Backward-Smoothing Extended Kalman Filter to Attitude Estimation and Prediction using Radar Observations

by

Kyle Clarke Volpe

B.S., Space Operations
United States Air Force Academy, 2007

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTERS OF SCIENCES IN AERONAUTICS AND ASTRONAUTICS
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2009

Signature of Author: ____
Department of Aeronautics and Astronautics
May 22, 2009

Certified by: ____
Dr. Paul J. Cefola
Consultant in Aerospace Systems, Spaceflight Mechanics, & Astrodynamics
Thesis Reader

Certified by: ____
Dr. David Chan
Assistant Group Leader, MIT Lincoln Laboratory
Thesis Supervisor

Certified by: ____
USAF Colonel (Ret) John E. Keesee
Senior Lecturer, Department of Aeronautics and Astronautics
Thesis Advisor

Accepted by: ____
Professor David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAY 2009**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Application of the Backward-Smoothing Extended Kalman Filter to Attitude Estimation and Prediction using Radar Observations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
The Department of the Air Force AFIT/ENEL WPAFB, OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
CI09-0051

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**322**	

[This page intentionally left blank.]

Application of the Backward Smoothing Extended Kalman Filter to Attitude Estimation and Prediction using Radar Observations

by

Kyle Clarke Volpe

Submitted to the Department of Aeronautics and Astronautics on May 8, 2009
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

ABSTRACT

The Lincoln Attitude Estimation System (LAES), a new tool being developed for the Space Situational Awareness Group (SSAG) at MIT Lincoln Laboratory, integrates several existing hardware and software systems, with a backward-smoothing extended Kalman filter (BSEKF). LAES is intended to determine the rotational motion of a freely tumbling spacecraft from a sequence of discrete-time radar images. The raw range-Doppler returns are collected using a ground-based sensor, which is owned and operated by the SSAG, and processed into a set of range/cross-range images. A three-dimensional geometric model is, through computer graphics procedures, displayed on top of the two-dimensional radar images, enabling an analyst to rotate (and scale in cross-range) the model in order to align it to the object's image. Therefore, the orthographic projection matrix that the computer graphics procedures computed to display the computer model, simultaneously describes the projection of the object onto the radar image plane. These measurements are essentially corrections to a nominal or baseline motion which had to be assumed in order to generate the images in the first place. Combining the reference motion, which describes the orientation of the image plane in inertial space, with the sequence of rotations describing the attitude of the spacecraft within the image plane, yields the final set of attitude measurements which are then passed to the BSEKF for processing. The existing free motion software currently in use within the Space Situational Awareness Group makes two critical assumptions: 1) that that the spacecraft is a symmetric rigid body and 2) that there are no disturbance torques acting on the spacecraft during the imaging period. The Lincoln Attitude Estimation System removes these simplifying assumptions in favor of a more flexible approach which is better suited for long-term studies of rigid body motion. Accordingly, several additions have been made to the backward-smoothing extended Kalman filter, including the addition of environmental torque models and an algorithm which generates an initial estimate for the inertia tensor using the same geometric model used in the image-model matching process. The BSEKF solves a nonlinear smoothing problem for the current and past sample intervals using iterative numerical techniques. This approach retains the nonlinearities of a fixed number of stages that precede the time of interest, and processes information from earlier stages in an approximate manner. The algorithm has been tested using simulated and actual data from a challenging spacecraft attitude estimation problem in which there is significant measurement noise, poor initial state estimates, and highly nonlinear system dynamics. The filter compensates for this uncertainty through concurrent estimation of the attitude and moment of inertia parameters. The filter has been demonstrated to accurately and reliably converge on a motion solution in both types of test cases.

Thesis Advisor:	USAF Colonel (Ret) John E. Keesee
Title:	Senior Lecturer, Department of Aeronautics and Astronautics

DISCLAIMERS

The views expressed in this thesis are those of the author and do not reflect the official policy or position of MIT, MIT Lincoln Laboratory, the United States Air Force, Department of Defense, or the U.S. Government.

This work is sponsored by the Air Force under Air Force Contract FA8721-05-C-002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Acknowledgements

I would like to take this opportunity to thank the people who have helped me over the past two years and made this thesis possible.

First and foremost, I would like to thank Zachary Folcik and Dr. Arthur Lue for the invaluable technical assistance that they have provided throughout the algorithm development process and writing of this thesis. I am lucky to have had the opportunity to work with such talented individuals, and am truly grateful for the patience and encouragement that they have shown me. Their generosity and my appreciation for all that they have done cannot be overstated.

I would also like to thank my thesis advisors, Dr. Paul Cefola, Dr. Dave Chan, and Colonel John Keesee for their guidance, support, and review of this work. It has been an honor working with such distinguished professionals.

At MIT Lincoln Laboratory, I had the pleasure of working with a number of very dedicated and capable individuals, who were kind enough to share their time and expertise in various field from radar imaging to software development. I would like to acknowledge the many contributions make by those in the Space Situational Awareness Group, especially: Dr. George Zogbi for providing me with direction at the outset of this project; Kenneth Fields for his assistance in helping me learn how to operate XELIAS and LLMotion; Vladimir Dizhoor for explaining the subtle details of working with two-dimensional radar images and spin-precession motion parameters; Anye Li for his help in creating the three-dimensional satellite models used in this thesis; Deena Dubin for her expertise and aid in working with the many software systems used by the Space Situational Awareness Group; Lin Stowe and Thomas Morgan for helping me understand the radar image generation process, ARIES, and HAX; Dr. Rod Cole and Dr. Gregory Ushomirsky for their insight into bistatic radar systems, and Charles Diorio for his assistance in reimaging passes and depth of operational knowledge. I would also like to recognize Bonnie Tuohy and Deborah Simmons for their gracious help with release review and arranging travel; and finally, Dr. Forrest Hunsberger for allowing me to work for the Space Situational Awareness Group.

Credit is also due to Dr. Mark Psiaki (Cornell University) for providing me with source code and helpful comments during the course of this work. I would also like to express my gratitude to Dr. Malcolm Shuster and Dr. Landis Markley whose work in the field of Attitude Determination and Estimation has laid the foundations for much of what is presented in this thesis. Over the years, a small but continuing stream of MIT Aeronautics and Astronautics graduate students has been interested in Astrodynamics, Artificial Satellite Theory, Orbit Determination, Space Surveillance, and related topics. These students and their colleagues and mentors at MIT Lincoln Laboratory and the Charles Stark Draper Laboratory, Inc. have made significant contributions to the research environment at MIT. The author would like to acknowledge these contributions.

To my family and friends, thank you all so much for your love and support throughout this long and arduous process.

[This page intentionally left blank.]

Table of Contents

[This page intentionally left blank.]

List of Figures

[This page intentionally left blank.]

List of Tables

[This page intentionally left blank.]

1. Introduction

1.1 Motivation

Attitude defines the orientation of a space vehicle relative to some fixed reference frame. Every spacecraft carries a complement of instruments, called a payload, that must be directed in some manner and whose operation is fundamentally dependent upon the spacecraft attitude. Accordingly, determining and controlling the orientation of a space vehicle are two critical processes which deliver value to the owner/operators of a given satellite system. When an anomaly results in the breakdown of command, control, and communication between the vehicle and operators on the ground, the various stakeholders of the system may turn to MIT Lincoln Laboratory for assistance in trying to determine the state of their spacecraft. With an extensive array of sensors, algorithms, techniques, and operational concepts at its disposal, the Space Situational Awareness Group (SSAG) is uniquely equipped to help the operators of errant spacecraft track and characterize the motion of their vehicle. With a set of externally-derived telemetry values, it may be possible to identify and correct the problem so that the spacecraft can be restored to normal operations. Whether Lincoln has been tasked to help recover anomalous spacecraft or to provide intelligence and decision support tools to various government agencies, the underlying need is ultimately for a system that can predict the future orientation of an uncontrolled spacecraft in an accurate and timely manner. With nearly 50 years of experience in the fields of orbit determination and advanced satellite imaging systems, Lincoln Laboratory's Space Situational Awareness Group is now looking to further extend its know-how into the realm of attitude estimation. Given that this is an internal research and development effort being done to provide the group with a new capability, the architecture is constrained by the hardware and software systems currently in use at Lincoln. Thus, there is an explicit need for the architecture to properly interface with legacy systems as an add-on algorithm that can be seamlessly incorporated into the greater suite of situational awareness tools.

Attitude estimation involves a two-part process: 1) approximation of a spacecraft's orientation from body measurements and known reference observations, such as line-of-sight measurements to known stars, and 2) filtering the inherently noisy measurements in order to arrive at a more refined estimate of the rotational motion of the spacecraft. Traditionally, attitude is sensed via an array of onboard measurement devices, such as Earth horizon sensors, magnetometers, or star sensors. The accuracy limit is usually determined by a combination of hardware and processing procedures. The measurements they produce can then be combined with models, in a number of different ways. One method uses a kinematics model propagated with three-axis rate-integrating gyros. Since, the rates measured by gyro drift over time, three more states are typically appended to the attitude state vector in order to determine this drift. Another way involves combining the kinematics model with a dynamics model for the angular rate. However, even a detailed dynamics model, such as Euler's rotational equations, will have inherent errors. For example, the moment of inertia matrix or initial angular velocity of the spacecraft may not be well known. This is typically compensated for in filter designs by using process noise, which in turn leads to challenges in "tuning" the filter. Throughout this thesis the terms *filter* and *estimator* are used synonymously, because noisy measurements are involved. Additionally, the term *smoother* is used to refer to a batch estimator algorithm, which is not executed in real-time. Attitude prediction is the process of forecasting the future orientation of the spacecraft by using dynamical models to extrapolate the attitude history. Here the limiting features are the knowledge of the environmental torques and the accuracy of both the initial estimate and mathematical model of the spacecraft dynamics.

In the non-cooperative attitude estimation problem being considered in this thesis, a much different approach is used to determine the orientation of a satellite. Two-dimensional linear radar images are generated by Doppler processing the radar returns in each of the range cells of a coherent wideband radar signal, the Doppler frequency of the center of mass being first removed. This process results in a range-Doppler image. Assuming an initial motion for the object relative to the radar line of sight enables the range-Doppler representation to be scaled to form a range/cross-range image. Measurements of the attitude are then made by aligning the projection of a three-dimensional computer graphics model (the wireframe) of the spacecraft with each discrete radar image in a given pass. The rotations need to orient the wireframe model in the radar image plane, in conjunction with the assumed nominal motion used to process the returns, describe the orientation of the satellite in inertial space. Due to the significant uncertainties and large measurement errors associated with a problem of this type, the backward-smoothing extended Kalman filter (BSEKF) is used to filter the data. The BSEKF improves on the traditional extended Kalman filter (EKF) by relinearizing a finite number of measurements in the past when a new observation is processed. The filter has been shown to have superior performance when the estimation problem contains severe nonlinearities that might significantly degrade the accuracy or convergence reliability of other filters. The systems architecture presented in this thesis will hereafter be referred to as the Lincoln Attitude Estimation System (LAES) when describing the combination of remote-sensing/measurement-making techniques and backward-smooth extended Kalman filter.

1.2 Overview of the Thesis

The overall purpose of this thesis is to present the unique challenges posed by using a series of radar images as the basis for making attitude measurements and to evaluate the application of the backward-smoothing extended Kalman filter to the non-cooperative attitude estimation problem. Since the measurements are made via independent, external observation (i.e., there is no assistance from the satellite operator), the mass properties and internal dynamics of the vehicle are completely unknown. Additionally, because the two-dimensional radar images only provide range and range-rate information, the attitude and angular velocity of the spacecraft can only be determined up to a rotation around the radar line of sight. While several simplifying assumptions are necessary to overcome these difficulties, the existing software systems used by the SSAG, goes so far as to presume that the spacecraft is a symmetric rigid body operating in a torque-free environment. This is done in order to allow for closed-form motion propagation; removing the assumption of spin-precession motion requires a system of ordinary differential equations expressing Euler's equations of motion. Use of the BSEKF, therefore, represents a significant departure from the approach currently used to determine the rotational motion of a freely tumbling spacecraft from a sequence of radar images. The Lincoln Attitude Estimation System eliminates these critical restrictions, through the incorporation of a detailed set of environmental torque models and simultaneous estimation of the moment of inertia parameters. Consequently, another major aim of this thesis is to investigate the ability of the BSEKF to be used in both short-term (single pass) and long-term (multi-pass) attitude prediction.

Chapter 2 provides background on rigid-body mechanics. The fact that nearly all globally continuous and nonsingular representations of rotations have at least one redundant component, has led to alternatives using an attitude parameterization which is either singular or redundant. Thus, depending on the situation, it may be easier to describe the attitude in terms of a 3×3 matrix, three rotation angles, or by a four-dimensional vector (a quaternion). While several other fundamentally different and often exotic choices also exist, Section 2.1 focuses on the three most commonly used methods for expressing the orientation of an object in three-dimensional space. The equations of motion of attitude dynamics can be divided into two sets: the kinematic equations of motion and the dynamic equations of motion. Kinematics is the study of motion irrespective of the forces that bring about that motion. The kinematic equations are a set of first-order differential equations specifying the time evolution of the attitude parameters. These equations, which contain the instantaneous angular velocity vector $\boldsymbol{\omega}$, are considered in Section 2.2. This section also presents the relationship between the rate of change of a vector in an inertial frame and its rate of change in a reference frame rotating with angular velocity $\boldsymbol{\omega}$. In Section 2.3, the angular momentum and moment of inertia tensor are precisely defined and the relations between them presented. This is done in order to set up the dynamic equations of motion, which express the time dependences of $\boldsymbol{\omega}$. These are needed for both dynamic simulations and attitude prediction whenever gyroscopic measurements of $\boldsymbol{\omega}$ are unavailable.

In Chapter 3, the non-cooperative attitude estimation problem is divided into its constituent parts. First, the measurement process is described by considering how the raw range-Doppler returns are collected via a ground-based sensor and processed into a series of discrete range/cross-range images. A technique known as image-model matching is then used to determine the attitude of the spacecraft within the radar image plane. Unfortunately, because of the limited amount of information contained in the radar observations, the radar system of equations is underdetermined. As will be discussed at the end of Section 3.1, the uncertainty in the orientation of the radar image plane in inertial space severely complicates the attitude estimation process. However, the details of how these challenges have been approached are the subject of Chapter 4. In section 3.2, the filtering process is described, by looking at the various attributes of the backward-smoothing extended Kalman filter. Because the BSEKF has properties similar to that of a sliding batch least-squares estimator, extended Kalman filter, and fixed interval smoother, these three types of algorithms are covered in greater detail in Sections 3.2.2, 3.2.3, and 3.2.4 respectively. The remainder of the chapter is devoted to the specifics of the BSEKF, with an emphasis on addressing the implementation and operational attributes of this fairly complex algorithm.

Chapter 4 couples the radar equations developed in Section 3.2 with the dynamics equations presented in Chapter 2. In addition to compiling a comprehensive set of governing equations, the various legacy systems used by the Space Situational Awareness Group for motion analysis are documented in Section 4.1. A description of the models and features which have been added to the BSEKF is given is Section 4.2, along with an explanation of how the algorithm has been integrated into the greater collection of motion analysis tools.

Chapter 5 presents the results of a truth-model simulation and set of real-life test cases which have been used to assess the performance of the BSEKF. Sections 5.1 and 5.2 describe the testing methodology for the two types of test cases, as well as initialization and filter tuning procedures. Section 5.3 analyses the filtering results for a simulated dataset generated using the algorithm's own dynamics equations and torque models. Measurements have also been made using an actual inactive satellite, imaged over several consecutive passes. The actual test cases are divided into short- and long-term results, depending on whether the attitude estimation and/or prediction was made using observations from a single pass or multiple passes (i.e., there is a substantial time gap between the datasets). The results show the BSEKF is able to 1) accurately and quickly converge on a motion solution for individual data passes, 2) overcome large initial errors, 3) filter over long time gaps separating sequential passes, and 4) better predict the attitude of spacecraft than existing motion analysis software.

Chapter 6 summarizes the conclusions of the thesis and outlines future work to improve the non-cooperative attitude estimation and prediction capabilities of MIT Lincoln Laboratory. Appendix A outlines all the conversion formulas needed to transform between the different attitude representations. Appendix B is a collection of miscellaneous equations which are used in various sections throughout the document. Appendix C includes data handling procedures and operating instructions for the Lincoln Attitude Estimation System. Appendix D provides Matlab source code for implementing the backward-smoothing extended Kalman filter, the environmental torque models, and the polyhedral mass properties algorithm.

[This page intentionally left blank.]

2 Mathematical Background

2.1 Attitude Representations

The attitude of a spacecraft is its orientation in space, expressed as a relation between two coordinate systems. The overall motion of a rigid spacecraft is specified by its position, velocity, attitude, and angular velocity. The first two quantities describe the translational motion of the center of mass of the spacecraft and are the subject of orbit determination. The latter two quantities describe the rotational motion of the body of the spacecraft about the center of mass and are the subject of this thesis on spacecraft attitude – namely how it is determined and how its future motion is predicted. While translational motion is fairly simple, rotational motion is much more complicated, since there are no solitary point masses in attitude problems, and the equations are both nonlinear and coupled. Although knowledge of the spacecraft orbit is frequently required to describe the rotational motion of a vehicle, orbital mechanics in general is outside the scope of this work. Accordingly, the dynamical coupling that exists between the two will often times be ignored; the noticeable exception being the discussion provided in Section 2.3.4.3 on environmental torque models. Even in the instances where interdependence of the two fields is most prevalent, it will be assumed that the time history of the spacecraft position is known and has been supplied by some process external to the attitude determination and estimation system.

Of the many ways to represent attitude, the most prolific and widely used include: 1) the axis and angle of rotation; 2) the rotation matrix; 3) Euler angles; and 4) the quaternion. Because attitude is more difficult to describe than position, it is often necessary to utilize multiple representations in order to take advantage of a particular property that simplifies a specific part of the problem. Accordingly, the ability to easily convert between each of these representations is of great importance and the equations for doing so are provided in Appendix A. Of all the representations discussed in this chapter, the axis and angle of rotation appeal most to our geometrical intuition of what constitutes a rotation and helps us to express mathematically what the other rotations mean geometrically; otherwise they are of little practical use. The rotation matrix is often constructed in order to transform vectors from one frame to another. Euler angles are convenient for treating spinning spacecraft and archiving attitude, since there are only three variables to record. Euler angles are also advantageous when trying to visualize rotations, but are otherwise not very useful. The quaternion is the most convenient representation to use in dynamical simulation of attitude, because it makes the best compromise between simplicity of the kinematics and dynamics equations of motion and the dimension of the system (1 pp. 412 - 420).

2.1.1. Right-Handed Orthonormal Coordinate Systems

In order to uniquely describe the attitude of a rigid body, three external coordinates are needed to specify the position of some reference point in the body (the origin) and three more are needed to indicate how the body is oriented with respect to the axes of the external space. As depicted in Figure 2.1, the configuration of a rigid body can be identified using two sets of Cartesian coordinates, one fixed in the body and another parallel to the external axes, but with the same origin as the body set of axes.

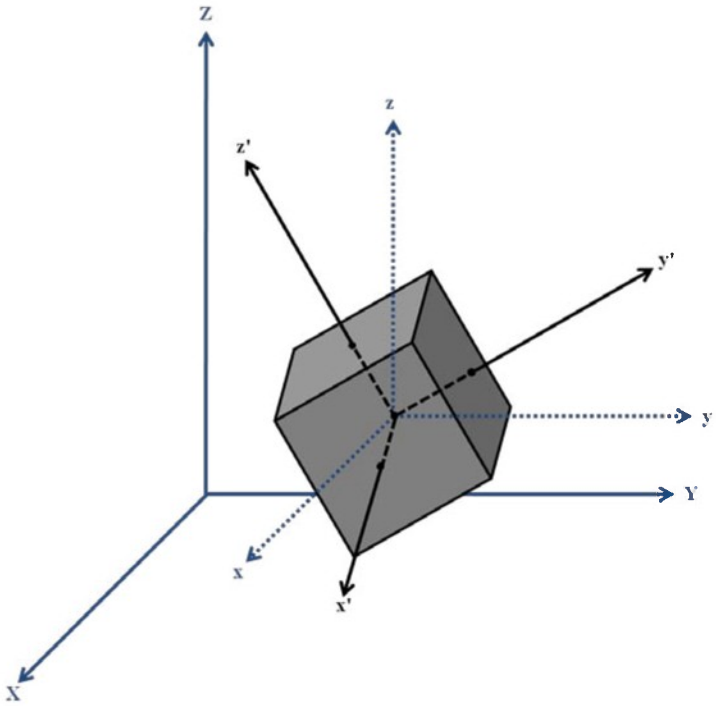


Figure 2.1 Orientation of the body-fixed coordinate system (black $x'y'z'$ axes) with respect to the inertial reference frame (blue xyz axes)

An orthonormal basis is defined a set of three unit vectors $\{\boldsymbol{i} \quad \boldsymbol{j} \quad \boldsymbol{k}\}$ along the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axis respectively, that are mutually orthogonal. Thus, the vectors of an orthonormal basis satisfy the following scalar-product relations (2 p. 239):

$$\begin{gathered} \boldsymbol{i}\cdot\boldsymbol{j}=\boldsymbol{i}\cdot\boldsymbol{k}=\boldsymbol{j}\cdot\boldsymbol{k}=0 \\ \boldsymbol{i}\cdot\boldsymbol{i}=\boldsymbol{j}\cdot\boldsymbol{j}=\boldsymbol{k}\cdot\boldsymbol{k}=1 \end{gathered} \tag{2.1}$$

Additionally, the bases that will be used are not only orthonormal but also right-handed; that is, their vector products satisfy (2 p. 239):

$$\begin{gathered} \boldsymbol{i}\times\boldsymbol{j}=-\boldsymbol{j}\times\boldsymbol{i}=\boldsymbol{k} \\ \boldsymbol{j}\times\boldsymbol{k}=-\boldsymbol{k}\times\boldsymbol{j}=\boldsymbol{i} \\ \boldsymbol{k}\times\boldsymbol{i}=-\boldsymbol{i}\times\boldsymbol{k}=\boldsymbol{j} \end{gathered} \tag{2.2}$$

Given a physical vector $\boldsymbol{r}$ in three-dimensional space and an orthonormal basis $\{\boldsymbol{i} \quad \boldsymbol{j} \quad \boldsymbol{k}\}$, coordinates $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ can be found such that

$$\boldsymbol{r}=\boldsymbol{x}\boldsymbol{i}+\boldsymbol{y}\boldsymbol{j}+\boldsymbol{z}\boldsymbol{k} \tag{2.3}$$

The components of $\boldsymbol{r}$ can be arranged into a column-vector of size 3 × 1 (matrix with n number of rows and m number of columns):

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.4}$$

If the physical vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are given in terms of an orthonormal basis as

$$\boldsymbol{u} = u_1\boldsymbol{i} + u_2\boldsymbol{j} + u_3\boldsymbol{k} \tag{2.5}$$

and

$$\boldsymbol{v} = v_1\boldsymbol{i} + v_2\boldsymbol{j} + v_3\boldsymbol{k} \tag{2.6}$$

then their column-vector equivalents are simply

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{ and } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{2.7}$$

The scalar or dot product of two column-vectors with respect to a common basis, may be expressed as (2 p. 240)

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^{\mathrm{T}}\mathbf{v} = u_1v_1 + u_2v_2 + u_3v_3 \tag{2.8}$$

where $\mathbf{u}^{\mathrm{T}}$ is the transpose of $\mathbf{u}$; a 1 × 3 row vector of the form:

$$\mathbf{u}^{\mathrm{T}} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \tag{2.9}$$

Similarly, the column-vector representation of $\boldsymbol{u} \times \boldsymbol{v}$, the vector or cross product, is simply (2 p. 240):

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix} = [\mathbf{u}\times]\mathbf{v} \tag{2.10}$$

where $[\mathbf{u}\times]$ is defined to be the 3 × 3 matrix:

$$[\mathbf{u}\times] = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ u_2 & u_1 & 0 \end{bmatrix} \tag{2.11}$$

It is important to note that unless the coordinate system (i.e. the basis) is indicated, a column-vector, typically denoted with a nonitalicized boldface character (**v**), is not strictly speaking, equivalent to a physical vector $\boldsymbol{v}$. Consequently, the following work primarily utilizes the column-vector representation – since vector operations can be replaced by matrix operations – and references the relevant coordinate system using a subscript, such as

$$\mathbf{v}_I = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_I \tag{2.12}$$

In the subsequent attitude estimation problem, the subscript I is used to represent the inertial coordinate system and B the body-fixed coordinate system. Unless otherwise indicated, the inertial reference frame used is the Geocentric Equatorial Coordinate System or Earth Centered Inertial (ECI), which is typically designated with the letters $\boldsymbol{IJK}$. As the name implies, this system originates at the center of the Earth; the I axis points towards the vernal equinox; the J axis is 90° to the east and lies in the equatorial or fundamental plane; and the K axis extends along the Earth's axis of rotation through the North Pole (3 p. 157). The body-fixed system is called the Satellite Coordinate System and is given the letters $\boldsymbol{XYZ}$. The origin of this frame is the satellite's center of mass; the X axis is perpendicular to the YZ plane; the Y axis runs parallel to some distinguishing feature on the spacecraft; and the Z axis is aligned with the longest dimension of the vehicle or the axis of symmetry, as depicted in Figure 2.2 (4 pp. 8.9 - 8.10).

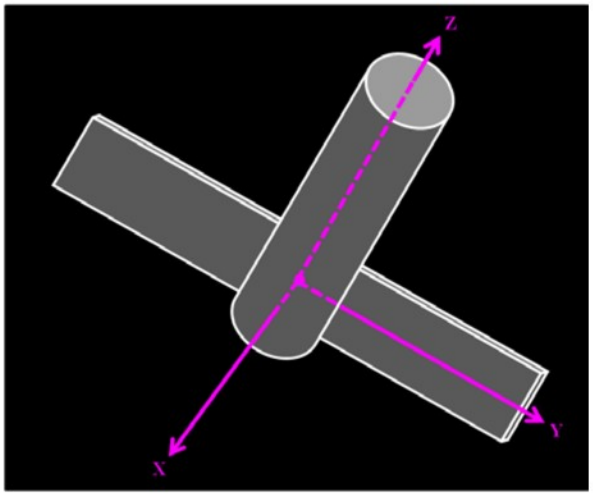


Figure 2.2 Convention used for the body-fixed reference frame (Satellite Coordinate System)

2.1.2. Rotation (Direction-Cosine) Matrix

A rotation matrix is a matrix whose multiplication with a vector rotates the vector while preserving its length. The special orthogonal group of all 3 × 3 rotation matrices is denoted by $SO(3)$. Thus if $\boldsymbol{R} \in SO(3)$, then it possesses the following two essential properties (5 pp. 52 - 53):

$$\det(\boldsymbol{R}) = \pm 1 \tag{2.13}$$

and

$$\boldsymbol{R}^{-1} = \boldsymbol{R}^{T} \tag{2.14}$$

Rotation matrices for which $\det \boldsymbol{R} = +1$ are called proper and those for which $\det \boldsymbol{R} = -1$ are called improper. While every orthogonal transformation preserves scalar products (maintains right-angles), only proper orthogonal transformations preserve vector products and therefore represent true rotations. Since every rotation leaves at least one axis unchanged (the axis of rotation), improper transformations, which change the direction of every vector, are impossible by means of a rotation (6 p. 5). Accordingly, improper matrices will not be discussed further, since they are not rigid-body transformations.

The elements of a rotation matrix are referenced as follows:

$$R = [\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{r_3}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.15}$$

Additionally, the convention that will be used defines the rotation matrix that encodes the attitude of a rigid-body to be the matrix that when pre-multiplied by a column vector expressed in inertial coordinates yields the same vector expressed in the body-fixed frame (i.e. the matrix which maps inertial coordinates into body-fixed coordinates). The following relations hold when transforming vectors from one coordinates system to another (6 p. 5):

$$\mathbf{v}_B = \boldsymbol{R}\mathbf{v}_I \Rightarrow \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_B = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_I \tag{2.16}$$

$$\mathbf{v}_I = \boldsymbol{R}^T\mathbf{v}_B \Rightarrow \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_I = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_B \tag{2.17}$$

A coordinate rotation is therefore, a rotation about a single coordinate axis, $\boldsymbol{x}$, $\boldsymbol{y}$, or $\boldsymbol{z}$, through an angle $\boldsymbol{\phi}$. The $\boldsymbol{x}$-, $\boldsymbol{y}$-, and $\boldsymbol{z}$,-axes rotations are often numbered 1, 2, and 3 respectively and are written as (6 p. 6)

$$\boldsymbol{R_x}(\boldsymbol{\phi}) = \boldsymbol{R_1}(\boldsymbol{\phi}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.18}$$

$$\boldsymbol{R_y}(\boldsymbol{\phi}) = \boldsymbol{R_2}(\boldsymbol{\phi}) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \tag{2.19}$$

$$\boldsymbol{R_z}(\boldsymbol{\phi}) = \boldsymbol{R_3}(\boldsymbol{\phi}) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

A sample rotation of this form is illustrated in Figure 2.3, which shows a rotation about the $\boldsymbol{z}$-axis by an angle $\boldsymbol{\phi}$.

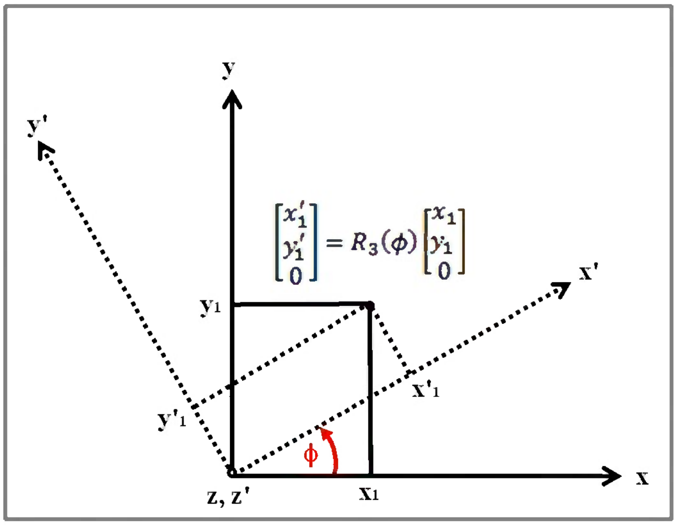

y
y'
$\begin{bmatrix} x'_1 \\ y'_1 \\ 0 \end{bmatrix} = R_3(\phi)\begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix}$
x'
y1
x'1
y'1
ϕ
x
z, z'
x1

Figure 2.3 Rotation about the $\boldsymbol{z}$-axis through an angle $\boldsymbol{\phi}$. Here $\boldsymbol{x}$ and $\boldsymbol{y}$ are the initial coordinate axes and $\boldsymbol{x'}$ and $\boldsymbol{y'}$ are the final coordinate axes (6 p. 6)

A rotation matrix is often referred to as a direction cosine matrix, because the elements of this matrix are the cosines of the unsigned angles between the body-fixed axes and the inertial axes. Denoting the inertial axes with $(x,\ y,\ z)$ and body-fixed axes with $(x',\ y',\ z')$, let $\theta_{x',y}$ be the unsigned angle between the x'-axis and the y-axis. In terms of these angles, the rotation matrix may be written as (6 p. 6)

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta_{x',x}) & \cos(\theta_{x',y}) & \cos(\theta_{x',z}) \\ \cos(\theta_{y',x}) & \cos(\theta_{y',y}) & \cos(\theta_{y',z}) \\ \cos(\theta_{z',x}) & \cos(\theta_{z',y}) & \cos(\theta_{z',z}) \end{bmatrix} \tag{2.21}$$

To illustrate this with a concrete example, consider the case shown in Figure 2.3. Here $\theta_{x',x} = \theta_{y',y} = \phi$, $\theta_{x',y} = \frac{\pi}{2} - \phi$, $\theta_{y',x} = \frac{\pi}{2} + \phi$, $\theta_{z',z} = 0$, and $\theta_{z',\{x,y\}} = \theta_{\{x,y\},z'} = \frac{\pi}{2}$. Therefore, equation (2.21) can be rewritten as (6 p. 6)

$$\boldsymbol{R} = \begin{bmatrix} \cos(\phi) & \cos((\pi/2) - \phi) & \cos(\pi/2) \\ \cos((\pi/2) + \phi) & \cos(\phi) & \cos(\pi/2) \\ \cos(\pi/2) & \cos(\pi/2) & \cos(0) \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

This is the same result that is presented in equation (2.20).

For an arbitrary rotation through an arbitrary angle ϕ about an axis $\mathbf{a}$, Euler's formula is given by (2 p. 246)

$$\boldsymbol{R}_{\mathbf{a}}(\phi) = \cos(\phi)\,\boldsymbol{I} + (1 - \cos(\phi))\mathbf{a}\mathbf{a}^{\mathrm{T}} - \sin(\phi)\,[\mathbf{a}\times] =$$

$$\begin{bmatrix} \cos(\phi) + a_1^2(1-\cos(\phi)) & a_1a_2(1-\cos(\phi)) + a_3\sin(\phi) & a_1a_3(1-\cos(\phi)) - a_2\sin(\phi) \\ a_2a_1(1-\cos(\phi)) - a_3\sin(\phi) & \cos(\phi) + a_2^2(1-\cos(\phi)) & a_2a_3(1-\cos(\phi)) + a_1\sin(\phi) \\ a_3a_1(1-\cos(\phi)) + a_2\sin(\phi) & a_3a_2(1-\cos(\phi)) - a_1\sin(\phi) & \cos(\phi) + a_3^2(1-\cos(\phi)) \end{bmatrix} \tag{2.23}$$

where $\mathbf{a} = [a_1 \quad a_2 \quad a_3]^T$ and $[\mathbf{a}\times]$ is given by equation (2.11). Since the determinant of equation (2.23) is always +1, rotation matrices are proper orthogonal and every rotation can therefore, be expressed as a rotation about a single axis. This result is known as Euler's Theorem. The axis of rotation has two free parameters, and the angle of rotation is a third parameter. Consequently, rotations are characterized by three independent parameters, which means that the nine elements of $\boldsymbol{R}$ must be subject to the six constraints, expressed in equation (2.14) (2 p. 246).

The successive application of two rotation matrices is equivalent to a third orthogonal transformation which can be considered as the product of the original two operators. For example (7 p. 144)

$$\mathbf{v}_B = \boldsymbol{R}_{\mathbf{a}_2}(\phi_2)\boldsymbol{R}_{\mathbf{a}_1}(\phi_1)\mathbf{v}_I \Rightarrow \mathbf{v}_B = \boldsymbol{R}_{\mathbf{a}_2\mathbf{a}_1}(\phi_2, \phi_1)\mathbf{v}_I \qquad (2.24)$$

where $\boldsymbol{R}_{\mathbf{a}_1}(\phi_1)$ is the first rotation matrix, $\boldsymbol{R}_{\mathbf{a}_2}(\phi_2)$ is the second rotation matrix, $\boldsymbol{R}_{\mathbf{a}_2\mathbf{a}_1}(\phi_2, \phi_1)$ is the combined linear transformation which takes a vector $\mathbf{v}$ from inertial coordinates to body-fixed coordinates. It is important to note that matrix multiplication is not commutative and therefore, (7 p. 144)

$$\boldsymbol{R}_{\mathbf{a}_2}(\phi_2)\boldsymbol{R}_{\mathbf{a}_1}(\phi_1) \neq \boldsymbol{R}_{\mathbf{a}_1}(\phi_1)\boldsymbol{R}_{\mathbf{a}_2}(\phi_2) \qquad (2.25)$$

Thus, the final coordinate system depends upon the order of application of the operators. However, matrix multiplication is associative; in a product of three or more matrices the order of the multiplications is unimportant (7 p. 145):

$$\left(\boldsymbol{R}_{\mathbf{a}_1}(\phi_1)\boldsymbol{R}_{\mathbf{a}_2}(\phi_2)\right)\boldsymbol{R}_{\mathbf{a}_3}(\phi_3) = \boldsymbol{R}_{\mathbf{a}_1}(\phi_1)\left(\boldsymbol{R}_{\mathbf{a}_2}(\phi_2)\boldsymbol{R}_{\mathbf{a}_3}(\phi_3)\right) \qquad (2.26)$$

The convention that will be used to annotate the proper order for each rotation sequence or combined rotation matrix is illustrated by equation (2.24) and requires matrices be multiplied from right to left.

2.1.3 Euler Angles

The Euler angle representation is defined by three successive rotations through angles ϕ_1, ϕ_2, and ϕ_3 about coordinate axes $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. Accordingly, an $\mathbf{a}_3$-$\mathbf{a}_2$-$\mathbf{a}_1$ Euler angle sequence would be one in which the first rotation is an angle ϕ_1 about the $\mathbf{a}_1$-axis, the second rotation is an angle ϕ_2 about the $\mathbf{a}_2$-axis, and the third rotation is an angle ψ about the $\mathbf{a}_3$-axis (8 p. 763). The particular sequence used is to some extent arbitrary since the initial rotation could be taken about any of the three Cartesian axes. In the subsequent two rotations, the only limitation is that no two successive rotations can be about the same axis. Consistent with the manner in which consecutive rotation matrices are written (applied from right to left), the combined rotation matrix is then given by (2 p. 246)

$$\boldsymbol{R}_{\mathrm{a}_3\mathrm{a}_2\mathrm{a}_1}(\phi_3,\phi_2,\phi_1) = \boldsymbol{R}_{\mathrm{a}_3}(\phi_3)\boldsymbol{R}_{\mathrm{a}_2}(\phi_2)\boldsymbol{R}_{\mathrm{a}_1}(\phi_1) \tag{2.27}$$

where the three rotation-axis column vectors, $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, must be chosen from the set consisting of the three unit column vectors aligned with one of the body axes

$$\mathbf{u}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \text{and } \mathbf{u}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{2.28}$$

While one could also write the transformation matrix in terms of four or more rotations, since rotations can be completely characterized by three parameters, three rotations about an appropriately chosen set of body axes are sufficient. This of course is the great advantage to using Euler angles – minimal dimensionality – the attitude can be efficiently stored and expressed using just three values, as opposed to the four parameters need for an axis and angle representation or the nine components required to construct the direction-cosine matrix.

For notational conciseness and to mitigate confusion, each of these angles is traditionally given a unique symbol $\phi\theta\psi$ and will typically be arranged in a three-dimensional vector called the Euler angle vector, defined by

$$\boldsymbol{E} = \begin{bmatrix}\phi\\\theta\\\psi\end{bmatrix} \tag{2.29}$$

There are 12 distinct conventions available for defining the Euler angles (in a right-handed coordinate system), which divide equally into two types: those whose rotations take place successively about each of the three coordinate axes and those in which the first and third rotations occur about the same axis, but the second is about one of the other two axes (8 p. 763). The two most frequently used of each type are the 1-2-3 and 3-1-3 Euler angle sequences. For illustration purposes, the 3-1-3 Euler angle rotation will be used extensively throughout this section, though both representations play a significant role in the attitude estimation problem presented in subsequent chapters. For this reason, the relevant conversion equations for both are provided in Appendix A for quick reference and comparison purposes.

Following the notation used in equation (2.24), the function which maps the 3-1-3 Euler angle vector to its corresponding rotation matrix is

$$\boldsymbol{R}_{313}(\psi,\theta,\phi) = \boldsymbol{R}_3(\psi)\boldsymbol{R}_1(\theta)\boldsymbol{R}_3(\phi) =$$

$$\begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\phi)\cos(\psi) - \sin(\phi)\cos(\theta)\sin(\psi) & \sin(\phi)\cos(\psi) + \cos(\phi)\cos(\theta)\sin(\psi) & \sin(\theta)\sin(\psi) \\ -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\theta)\cos(\psi) & \sin(\theta)\cos(\psi) \\ \sin(\phi)\sin(\theta) & -\cos(\phi)\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2.30}$$

The sequence presented above is started by rotating the initial system of axes, xyz, by an angle ϕ counterclockwise about the z-axis, and the resultant coordinate system is labeled $x'y'z'$, as shown in the first block of Figure 2.4. In the second stage, the intermediate axes, $x'y'z'$, are rotated about the x'-ais counterclockwise by an angle θ to produce yet another intermediate set, the $x''y''z''$ axes. Finally, the $x''y''z''$ axes are rotated counterclockwise by an angle ψ about the z''-axis to produce the desired $x'''y'''z'''$ system of axes.

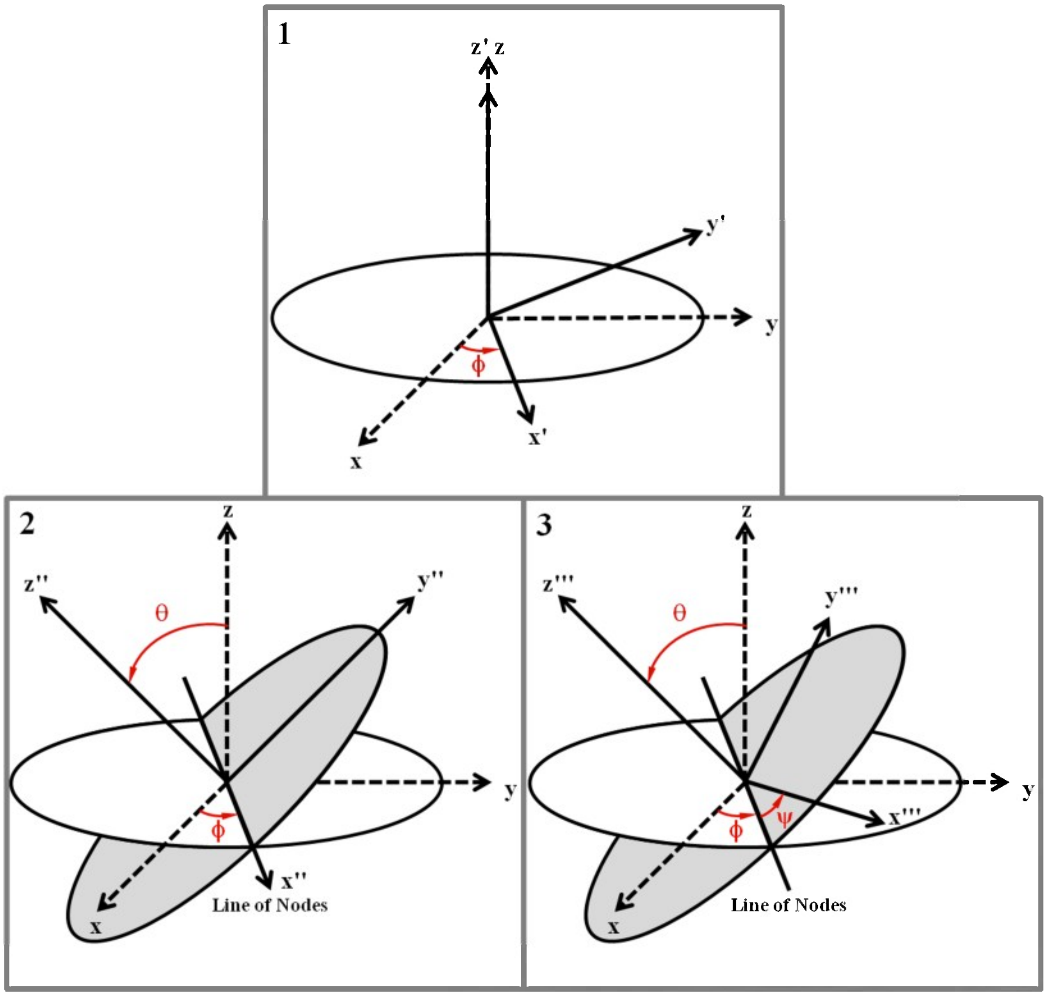


Figure 2.4 Rotations defining the 3-1-3 Euler angle rotation sequence

Unfortunately, the Euler angles are not a well-behaved representation of attitude. Like all the parameterizations of this type, the 3-1-3 series has singularities at nutation values of $\boldsymbol{\theta} = \boldsymbol{n\pi}$ for $\boldsymbol{n} = \pm$ integer value or 0. At these points, changes in spin and precession constitute the same motion. Intuitively, singularities arise, in this case, from the indistinguishability of changes in the first and third Euler angles when the second Euler angle is at the critical values just mentioned. For rotation types, such as the 1-2-3 Euler angles, that do not have a repeated axis of rotation, singularities occurs at $\boldsymbol{\theta} = \frac{\pi}{2} + \boldsymbol{n\pi}$ for $\boldsymbol{n} = \pm$ integer value or 0, because for these values of $\boldsymbol{\theta}$, the $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ angles have similar effects (5 p. 74). Thus, to avoid the singularity problem, one must resort to using two sets of Euler angles and occasionally switch from one set to the other; doing so, however, requires large numbers of trigonometric functions to be computed, negating the benefits derived from being able to represent attitude with a minimal number of parameters. In the case of spinning spacecraft, two of the Euler angles, θ and ψ, can be chosen to be the spherical angles of the spin axis (z-axis) and the remaining Euler angle ϕ the angle of rotation about the spin axis. In such situations, the spin axis is generally stable and confined to some small region about a nominal direction. Two of the Euler angles then have limited periodic variation, whereas the third tends to have an almost constant rate. Thus, in the case of spinning spacecraft, the singularity can be avoided and the angles do not experience complications.

2.1.4. The Quaternion

The deficiencies in the Euler angle representation have led many in the attitude field to use unit quaternions as a parameterization of the attitude of a rigid body. The relevant functions of unit quaternions have no singularities and the representation is well-suited to integrating the angular velocity of a body over time. However, using unit quaternions also have some disadvantages, namely that the four quaternion parameters do not have intuitive physical meaning, since they express rotations in four-dimensional space, and that a quaternion must have unity norm to be a pure rotation. The unit norm constraint, which is a quadratic in form, is particularly problematic if the attitude parameters are to be included in an optimization, as most standard optimization algorithms cannot encode such constraints (5 p. 169).

The quaternion is free of the analytical complexity that Euler angles typically encounter and only has one additional component. The quaternion can be defined as a 4 × 1 vector with the form (2 p. 250):

$$\boldsymbol{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.31}$$

where

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \sin(\phi/2)\hat{\mathbf{a}},\, q_4 = \cos\,(\phi/2) \tag{2.32}$$

The unit-column vector $\hat{\mathbf{a}}$ is the axis of rotation, and ϕ is the rotation angle. Additionally, $\mathbf{q}$ is the vector component of the quaternion and q_4 is the scalar component. Because a four-dimensional vector is used to describe three dimensions, the quaternion components are not independent from each other and must satisfy a single constraint given by (6 p. 14)

$$\boldsymbol{q}^T\boldsymbol{q} = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{2.33}$$

Other useful equations which will be referenced throughout this thesis include those for the adjoint, norm, and inverse of the quaternion, which can each be written as follows (6 p. 14):

$$\bar{\boldsymbol{q}} = \begin{bmatrix} -\mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix} \tag{2.34}$$

$$\|\boldsymbol{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \tag{2.35}$$

$$\boldsymbol{q}^{-1} = \frac{\bar{\boldsymbol{q}}}{\|\boldsymbol{q}\|} \tag{2.36}$$

The quaternion has several advantages over the rotation matrix as a representation of attitude. First, it has fewer elements (four instead of nine), so it requires less storage. Second, there are fewer constraints (one instead of six). Third, unlike the Euler angles, which cannot be merged easily when one combines rotations, the composition rule for the quaternion is very straightforward and requires fewer multiplications (16 instead of the 27 needed for rotation matrices) (2 p. 251). Fourth, if, because of accumulated numerical round-off error, the quaternion loses its' orthogonality, the constraint can be easily reinforced by simply replacing q with (2 p. 252)

$$\boldsymbol{q}_{\perp} = \frac{1}{\sqrt{\boldsymbol{q}^T\boldsymbol{q}}}\boldsymbol{q} \tag{2.37}$$

Finally, due to the redundant fourth term, the quaternion also has the advantage of being free of singularities. Apart from the overall sign, only one quaternion is needed to characterize a given attitude (2 p. 251).

The attitude matrix is related to the quaternion by (2 p. 250)

$$\boldsymbol{A}(\boldsymbol{q}) = (q_4^2 - |\mathbf{q}|^2)\mathbf{1}_{4\times4} + 2\mathbf{q}\mathbf{q}^T - 2q_4[\mathbf{q}\,\times] =$$

$$\begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_4q_3) & 2(q_1q_3 - q_4q_2) \\ 2(q_2q_1 - q_4q_3) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_4q_1) \\ 2(q_3q_1 + q_4q_2) & 2(q_3q_2 - q_4q_1) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{2.38}$$

where $\mathbf{1}_{4\times4}$ is a 4 × 4 identity matrix and $[\mathbf{q}\,\times]$ is the cross-product matrix defined by

$$[\mathbf{q}\,\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{2.39}$$

For small angles the vector part of the quaternion is approximately equal to half angles.

If $\boldsymbol{q}$ is the quaternion of the first rotation and $\boldsymbol{q}'$ is the quaternion of the second rotation, then the combined rotation is represented by $\boldsymbol{q}''$, where (2 p. 251)

$$\boldsymbol{q}'' = \boldsymbol{q}'\boldsymbol{q} = \begin{bmatrix} q_4'\mathbf{q} + q_4\mathbf{q}' - \mathbf{q}' \times \mathbf{q} \\ q_4'q_4 - \mathbf{q}' \cdot \mathbf{q} \end{bmatrix} \tag{2.40}$$

The upper right part of the expression gives the vector component $\mathbf{q}''$, while the lower part gives the scalar component q_4''. The composition rule for the quaternion is not unique, since the sign could have been changed on the rightmost element of equation (2.40). The sign convention used in the above equation is the one generally accepted and most convenient. The quaternion multiplication is not commutative and may be written more compactly as the second quaternion post-multiplied by a matrix-valued function of the first quaternion. That is, (6 p. 14)

$$\boldsymbol{q}'\boldsymbol{q} = \boldsymbol{Q}(q')\boldsymbol{q} = \bar{\boldsymbol{Q}}(q)\boldsymbol{q}' \tag{2.41}$$

$$\boldsymbol{q}\boldsymbol{q}' = \boldsymbol{Q}(q)\boldsymbol{q}' = \bar{\boldsymbol{Q}}(q')\boldsymbol{q} \tag{2.42}$$

where the quaternion matrix function $\boldsymbol{Q}(q)$ is defined by

$$\boldsymbol{Q}(q) = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \tag{2.43}$$

and the closely related conjugate quaternion matrix function $\bar{\boldsymbol{Q}}(q)$ is defined by

$$\bar{\boldsymbol{Q}}(q) = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \tag{2.44}$$

Substituting equation (2.34) into equations (2.43) and (2.44) yields

$$\boldsymbol{Q}(\bar{q}) = \boldsymbol{Q}(q)^T \tag{2.45}$$

$$\bar{\boldsymbol{Q}}(\bar{q}) = \bar{\boldsymbol{Q}}(q)^T \tag{2.46}$$

Therefore, the quaternion composition rule may be written as (9 p. 759)

$$\boldsymbol{q}'' = \boldsymbol{q}'\boldsymbol{q} = \boldsymbol{Q}(q')\boldsymbol{q} \Rightarrow \begin{bmatrix} q_1'' \\ q_2'' \\ q_3'' \\ q_4'' \end{bmatrix} = \begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.47}$$

and the quaternion difference rule can be expressed as follows:

$$\boldsymbol{q}'' = \boldsymbol{q}'\boldsymbol{q}^{-1} = \bar{\boldsymbol{Q}}(q)^T\boldsymbol{q}' \Rightarrow \begin{bmatrix} q_1'' \\ q_2'' \\ q_3'' \\ q_4'' \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}^T \begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix} \tag{2.48}$$

where $\boldsymbol{q}''$ is now the quaternion that is needed to transform between the first and second quaternion rotations, $\boldsymbol{q}$ and $\boldsymbol{q}'$ respectively, and can be thought of as the change or difference in attitude.

2.2 Rigid Body Kinematics

Attitude kinematics is the fundamental description of how a change in orientation with time is characterized and is inherently tied to a spacecraft's angular velocity (2 p. 252). Consider a rigid body rotating about some axis $\mathbf{n}$ and with a vector $\mathbf{v}$ that is fixed in the rotating body frame, as shown in Figure 2.5.

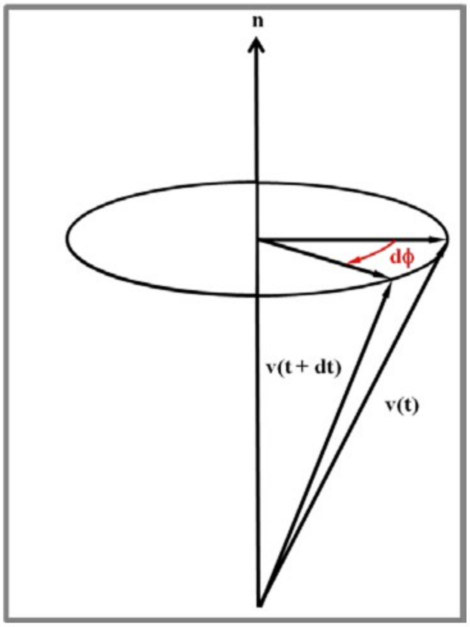


Figure 2.5 Change of a rotating vector over time (10 p. 516)

The rate of change of the vector $\mathbf{v}$ over time t is given by the derivative (10 p. 515)

$$\frac{d}{dt}\mathbf{v} = \frac{d\phi}{dt}(\mathbf{n} \times \mathbf{v}) \tag{2.49}$$

where $\boldsymbol{d\phi/dt}$ is the angular rate through which the body has rotated about the instantaneous axis $\mathbf{n}$. Thus,

$$\omega = \frac{d\phi}{dt}(\mathbf{n}) \tag{2.50}$$

which is the angular velocity of the body (10 p. 515). Since a right-handed orthonormal basis fixed in the inertial frame does not change with time, the derivative of the inertial components of $\mathbf{v}$ are simply the inertial components of the temporal derivative of $\mathbf{v}$, given by (10 p. 515)

$$\frac{d}{dt}\mathbf{v}_I = \boldsymbol{\omega}_I \times \mathbf{v}_I \tag{2.51}$$

Conversely, the components of $\mathbf{v}$ with respect to a basis fixed in the body reference frame are constant since the vector is rotating with the body itself. Hence, (10 p. 516)

$$\frac{d}{dt}\mathbf{v}_B = 0 \tag{2.52}$$

2.2.1 Kinematics Equation for the Rotation Matrix

At any instant, the orientation of a rigid-body can be specified by an orthogonal transformation, the elements of which may be expressed in terms of some suitable set of parameters. As time progresses, the orientation will change, and hence the matrix of transformation will be a function of time and may be written as $\boldsymbol{R}(t)$. If the body axes are chosen such that they align with the space axes at the time $t = 0$, then the transformation is initially simply the identity transformation:

$$\boldsymbol{R}(0) = \mathbf{1}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.53}$$

At any later time, $\boldsymbol{R}(t)$ will in general differ from the identity transformation, but since the physical motion must be continuous, $\boldsymbol{R}(t)$ must be a continuous function of time (7 p. 156). Differentiating both sides of the expression (and suppressing the explicit time dependence) (7 p. 172):

$$\mathbf{v}_B(t) = \boldsymbol{R}(t)\mathbf{v}_I(t) \tag{2.54}$$

leads to

$$\frac{d}{dt}\mathbf{v}_B = \left(\frac{d}{dt}\boldsymbol{R}\right)\mathbf{v}_I + \boldsymbol{R}\frac{d}{dt}\mathbf{v}_I \tag{2.55}$$

Substituting in equations (2.51) and (2.52) allows equation (2.55) to be rewritten as:

$$0 = \left(\frac{d}{dt}\boldsymbol{R}\right)\mathbf{v}_I + \boldsymbol{R}(\boldsymbol{\omega}_I \times \mathbf{v}_I) \tag{2.56}$$

Using the fact $\boldsymbol{R}$ is a proper orthogonal matrix, the definition of the cross-product matrix, and the associative property of matrix multiplication enable equation (2.56) to be further reduced until one finally arrives at:

$$\frac{d}{dt}\boldsymbol{R} = -[\boldsymbol{\omega}\times]\boldsymbol{R} \Rightarrow \begin{bmatrix} \dot{r}_{11} & \dot{r}_{12} & \dot{r}_{13} \\ \dot{r}_{21} & \dot{r}_{22} & \dot{r}_{23} \\ \dot{r}_{31} & \dot{r}_{32} & \dot{r}_{33} \end{bmatrix} = -\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ \omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.57}$$

which is the kinematics equation for the rotation matrix. This result enables the general expression relating the time derivatives of the representations of vectors in two frames in the following manner (2 p. 254):

$$\frac{d}{dt}\mathbf{v}_B = -\boldsymbol{\omega}_B \times \mathbf{v}_B + \boldsymbol{R}\frac{d}{dt}\mathbf{v}_I \tag{2.58}$$

2.2.2 Kinematics Equation for the Euler Angles

While it is often convenient and/or necessary to express the angular velocity vector in terms of the Euler angles and their time derivatives, the kinematic relationship for this parameterization is more complicated than that of the rotation matrix. While a finite rotation cannot be represented by a single vector, the same objection does not hold if only infinitesimal rotations are considered. An infinitesimal rotation is an orthogonal transformation of coordinate axes in which the components of a vector are almost the same in both sets of axes – i.e. the change is exceedingly small (7 p. 173). The general infinitesimal rotation associated with $\boldsymbol{\omega}$ can be considered as consisting of three successive minute rotations with angular velocities $\omega_\phi = \dot{\phi}$, $\omega_\theta = \dot{\theta}$, and $\omega_\psi = \dot{\psi}$. Consequently, the vector $\boldsymbol{\omega}$ can be obtained as the sum of the three separate angular velocity vectors with respect to the body-fixed coordinate frame, given by (10 p. 513)

$$\boldsymbol{\omega}_B = \omega_\phi + \omega_\theta + \omega_\psi = \dot{\phi}\mathbf{a}_1 + \dot{\theta}\mathbf{a}_2 + \dot{\psi}\mathbf{a}_3 \tag{2.59}$$

Unfortunately, the directions ω_ϕ, ω_θ, and ω_ψ are not symmetrically positioned with the same basis: ω_ϕ is along the inertial z-axis, ω_θ is along the line of nodes in the intermediate body frame, while ω_ψ alone is along the body z'''-axis. However, a set of rotation matrices can be used to transform the representations of the Euler axes with respect to the intermediate bases into representations with respect to the final body axes. Carrying out the necessary transformations leads to (2 p. 256)

$$\boldsymbol{\omega}_B = \dot{\psi}\mathbf{a}_3 + \dot{\theta}\boldsymbol{R}_{\mathbf{a}_3}(\psi)\mathbf{a}_2 + \dot{\phi}\boldsymbol{R}_{\mathbf{a}_3}(\psi)\boldsymbol{R}_{\mathbf{a}_2}(\theta)\mathbf{a}_1 \tag{2.60}$$

where the unit column vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ are the representations of the three Euler axes with respect to the intermediate bases and take on any of the three values given by equation (2.28), depending on which Euler angle sequence is being employed.

For the 3-1-3 Euler angle sequence, equation (2.60) becomes (7 p. 174)

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B = \begin{bmatrix} 0 & \cos(\psi) & \sin(\theta)\sin(\psi) \\ 0 & -\sin(\psi) & \sin(\theta)\cos(\psi) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{2.61}$$

The inverse of the combined rotation matrix above results in a formula for converting an angular velocity vector into a vector of Euler angle rates (6 pp. 10 - 11)

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \frac{1}{\sin(\theta)} \begin{bmatrix} -\cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) & \sin(\theta) \\ \sin(\theta)\cos(\psi) & -\sin(\theta)\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B \tag{2.62}$$

For the 3-1-3 series, the Euler angle rates become infinite when $\sin(\theta) = 0$ even though the angular velocity vector may be finite. This is yet another indication of that the Euler angles are singular. This problem can be corrected or avoided by simply using a different sequence of Euler angles, say, 1-2-3, in calculations when this condition is approached. A set of 3-1-3 Euler angles may be written as a function of a set of 1-2-3 Euler angles according to (6 pp. 13 - 14)

$$\boldsymbol{E}_{123}\big(\boldsymbol{R}_{313}(\psi,\theta,\phi)\big) = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{123}$$

$$= \begin{bmatrix}\operatorname{atan2}(\cos(\phi)\cos(\theta)\sin(\psi) + \sin(\phi)\cos(\psi), -\sin(\phi)\cos(\theta)\sin(\psi) + \cos(\phi)\cos(\psi))\\ -\operatorname{asin}(\sin(\theta)\sin(\psi))\\ \operatorname{atan2}(\sin(\theta)\cos(\psi), \cos(\theta))\end{bmatrix} \tag{2.63}$$

However, the 1-2-3 series is not trouble-free either. In this case, the singularity arises when $\cos(\theta) = 0$. Consequently, one must alternate between two sequences of Euler angles to represent the attitude as it changes. This problem is inherent in the Euler angles but is absent in the rotation matrix and the quaternion, which are both well-behaved at all attitudes.

2.2.3 Kinematics Equation for the Quaternion

The time derivative of the unit quaternion is the vector of quaternion rates, and like the rotation matrix, has a very simple form. The quaternion rates, denoted by $\dot{\boldsymbol{q}}$, are related to the angular velocity. The functions that map a unit quaternion and its temporal derivative to the angular velocity in inertial and body-fixed coordinates are $\boldsymbol{\omega}_B$, defined by (6 p. 16)

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}_B = 2\begin{bmatrix}q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} \tag{2.64}$$

The inverse mapping, from the angular velocity and the unit quaternion to the quaternion rates, are given by (6 p. 16)

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} = \frac{1}{2}\begin{bmatrix}q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\end{bmatrix}^T\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}_B \tag{2.65}$$

or equivalently as (10 p. 512)

$$\frac{d}{dt}\boldsymbol{q} = \frac{1}{2}\begin{bmatrix}[\boldsymbol{\omega}\times] & \boldsymbol{\omega}\\ -\boldsymbol{\omega} & 0\end{bmatrix}\boldsymbol{q} \Rightarrow \begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} = \frac{1}{2}\begin{bmatrix}0 & \omega_3 & -\omega_2 & \omega_1\\ -\omega_3 & 0 & \omega_1 & \omega_2\\ \omega_2 & -\omega_1 & 0 & \omega_3\\ -\omega_1 & -\omega_2 & -\omega_3 & 0\end{bmatrix}_B\begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix} \tag{2.66}$$

The kinematic equation for the quaternion is linear in the elements of both $\boldsymbol{\omega}_B$ and $\boldsymbol{q}$. This simplifies numerical integration. For this reason and the simple method of correcting normalization errors, the quaternion is the preferred representation for simulation studies.

2.3 Rigid Body Dynamics

Attitude dynamics is concerned with the motion of a body in the presence of applied torques and a complete description of the attitude motion of a spacecraft depends on the treatment of both the dynamical and kinematical aspects (10 p. 510). A particular reference point in the rigid body is usually sought, such that the problem can be split into two separate parts, one purely translational and the other purely rotational about the point. For bodies without a fixed point, the most useful reference point is almost always the center of mass. Unfortunately, rotational motion is much more complicated than translational motion. In translational motion, the force-free case leads to movement which has constant linear velocity. For a body with an axis of symmetry, the force-free attitude motion is nontrivial but can be expressed using trigonometric functions. If however, a totally asymmetric rigid body is considered, even torque-free rotational motion requires elliptical integrals for its description (7 p. 184).

2.3.1 Angular Momentum and the Properties of the Inertia Tensor

When a rigid body moves with one point stationary (typically taken to be the center of mass), the total angular momentum $(\boldsymbol{L})$ about that point is (10 p. 516)

$$\boldsymbol{L} = \sum_{i=1}^{n} m_i \boldsymbol{r}_i \times \boldsymbol{v}_i \tag{2.67}$$

where m_i, $i = 1, \dots n$, are the component masses comprising the rigid body, and $\boldsymbol{r}_i$ and $\boldsymbol{v}_i$ are the radius vector and velocity, respectively of the i^{th} particle relative to the given point. Since $\boldsymbol{r}_i$ is a fixed vector relative to the body, the velocity with respect to the space set of axes arises solely from the rotational motion of the rigid body about the fixed point according to the formula (10 p. 516)

$$\boldsymbol{v}_i = \omega \times \boldsymbol{r}_i \tag{2.68}$$

Hence, equation (2.67) can be written as (2 p. 275)

$$\boldsymbol{L} = \sum_{i=1}^{n} m_i[\boldsymbol{r}_i \times (\boldsymbol{\omega} \times \boldsymbol{r}_i)] \tag{2.69}$$

and then fully expanded to (7 p. 187)

$$\boldsymbol{L} = \sum_{i=1}^{n} m_i\left[\boldsymbol{\omega}\boldsymbol{r}_i^2 - \boldsymbol{r}_i(\boldsymbol{r}_i \cdot \boldsymbol{\omega})\right] = \begin{bmatrix} \omega_x m_i\left(\boldsymbol{r}_i^2 - x_i^2\right) - \omega_y m_i x_i y_i - \omega_z m_i x_i z_i \\ -\omega_x m_i y_i x_i + \omega_y m_i\left(\boldsymbol{r}_i^2 - y_i^2\right) - \omega_z m_i y_i z_i \\ -\omega_x m_i z_i x_i - \omega_y m_i z_i y_i + \omega_z m_i\left(\boldsymbol{r}_i^2 - z_i^2\right) \end{bmatrix} \tag{2.70}$$

The angular momentum vector is therefore related to the angular velocity by a linear transformation that can be summarized as (2 p. 276):

$$\boldsymbol{L} = \boldsymbol{I\omega} \tag{2.71}$$

In this equation ***I*** is the inertia tensor, a symmetric matrix with the form (7 p. 187) and (10 p. 518):

$$\boldsymbol{I}_B = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_i^2 - x_i^2 & -x_i y_i & -x_i z_i \\ -y_i x_i & \boldsymbol{r}_i^2 - y_i^2 & -y_i z_i \\ -z_i x_i & -z_i y_i & \boldsymbol{r}_i^2 - z_i^2 \end{bmatrix} m_i = \int \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -yx & x^2 + z^2 & -yz \\ -zx & -zy & x^2 + y^2 \end{bmatrix}_B dm \tag{2.72}$$

where the center expression appears in the form suitable if the rigid body were composed of discrete particles; and the right side, is the more appropriate form for continuous bodies, where the summation is replaced by a volume integration and the particle masses become a mass density. The diagonal elements of ***I*** are called the moments of inertia and the off-diagonal elements are called the products of inertia. Unlike the operator of rotation, ***I*** will have dimensions – mass times length squared ($\mathrm{kg \cdot m^2}$) – and is not restricted by any orthogonality conditions (7 p. 188). Since, the angular momentum of a rigid body about its center of mass depends on the mass distribution, only the inertia tensor is needed in order to properly describe its attitude dynamics. It should also be noted that for rigid bodies, the mass density in body coordinates is constant. Therefore, it is preferable to work in the body-fixed referenced frame, where the inertia tensor will be invariable over time (2 p. 276).

From the defining equation given by equation (2.72), it can be seen that the components of the tensor are symmetrical. This means that, while the inertia tensor will in general have nine components, only six of them will be independent – the three along the diagonal plus three of the off-diagonal elements (2 p. 276). The inertia coefficients depend both upon the location of the origin of the body set of axes and upon the orientation of these axes with respect to the body. This symmetry suggests that there exists a set of coordinates in which the tensor is diagonal with the three principal values I_1, I_2, and I_3. In this system, the components of $\boldsymbol{L}$ would involve only the corresponding components of $\boldsymbol{\omega}$, thus (10 pp. 519 - 520)

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{2.73}$$

Transformation of the inertia tensor from one right-handed orthonormal basis to another with the same origin can be done with the following simple equation:

$$\boldsymbol{I'} = \boldsymbol{RIR}^T \tag{2.74}$$

where I' is the inertia tensor in the new coordinate frame and $\boldsymbol{R}$ is the proper rotation matrix connecting the two bases. This rotation can be expressed in terms of the Euler angles ϕ, θ, and ψ as shown in equations $(\mathrm{A}.1)$ and $(\mathrm{A}.2)$. A proper choice of these angles will transform $\boldsymbol{I}$ into its diagonal form

$$\boldsymbol{I'} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \tag{2.75}$$

where I_1, I_2, and I_3 (which are the eigenvalues of $\boldsymbol{I}$) are referred to as the principal moments of inertia and the directions x', y', and z' defined by the rotation matrix in equation (2.74) are called the principal axes of the inertia tensor (7 p. 196). The three principal values of the moment of inertia tensor can be found by solving the following cubic equation for the values of $\boldsymbol{I}$ that arise from the determinant of (7 p. 197)

$$\begin{bmatrix} I_{xx} - I & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} - I & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} - I \end{bmatrix} = 0 \tag{2.76}$$

The principal moments of inertia cannot be negative, because as the diagonal elements in the principal axes system they have the sum of squares. For one of the principal moments to vanish requires that the body's axis of symmetry pass through the origin. Since the inertia tensor is positive definite, negative values on the diagonal indicate a mass distribution which is physically impossible. While this may seem inconsequential, the existing software system used to generate free-motion solutions at Lincoln Laboratory does not discard state estimates which may result in negative values on the diagonal of the inertia matrix. Such physically unrealistic motion solutions have been encountered during the course of this research effort and are a strong indicator that all is not well.

2.3.2 Euler's Equation

For the rotational motion about the center of mass, the direct Newtonian approach leads to a set of expression known as Euler's equations of motion. In the inertial coordinate frame the torque $\boldsymbol{N}$ acting on a rigid body is related to the angular moment through the formula (2 p. 277)

$$\boldsymbol{N}_I = \frac{d}{dt}\boldsymbol{L}_I \tag{2.77}$$

Conversely, the derivatives with respect to axes fixed in the body leads to (10 p. 521)

$$\frac{d}{dt}\boldsymbol{L}_I = \frac{d}{dt}\boldsymbol{L}_B + \boldsymbol{\omega}_B \times \boldsymbol{L}_B \Rightarrow \boldsymbol{R}_{\mathrm{a}}(\phi)\boldsymbol{N}_I = \frac{d}{dt}\boldsymbol{L}_B + \boldsymbol{\omega}_B \times \boldsymbol{L}_B \tag{2.78}$$

which, after applying the rotation matrix $\boldsymbol{R}_{\mathrm{a}}(\phi)$ to transform the torque from inertial to body-fixed coordinates, results in Euler's equation

$$\frac{d}{dt}\boldsymbol{L}_B = -\boldsymbol{\omega}_B \times \boldsymbol{L}_B + \boldsymbol{N}_B \tag{2.79}$$

Substituting equation (2.71) into the above equation, results in the following dynamics equation (10 p. 522):

$$\begin{aligned} N_1 &= I_1\dot{\omega}_1 - \omega_2\omega_3(I_2 - I_3) \\ N_2 &= I_2\dot{\omega}_2 - \omega_3\omega_1(I_3 - I_1) \\ N_3 &= I_3\dot{\omega}_3 - \omega_1\omega_2(I_1 - I_2) \end{aligned} \tag{2.80}$$

which can be rearranged into

$$\frac{d}{dt}\boldsymbol{\omega} = \boldsymbol{I}^{-1}[-\boldsymbol{\omega} \times (\boldsymbol{I\omega}) + \boldsymbol{N}] \tag{2.81}$$

However, equation (2.81) is not complete and must be combined with one of the kinematical relationships for the rotation matrix, Euler angles, quaternion, or some other attitude representation.

2.3.3 Torque-Free Motion of a Symmetric Rigid Body

Assuming that the spacecraft body axes are aligned with the principal axes of the inertia tensor and that the torque acting on the spacecraft is zero ($\boldsymbol{N} = 0$), enables Euler's equations to be rearranged/rewritten as (2 p. 278):

$$\begin{aligned} I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 = 0 \\ I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 = 0 \\ I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 = 0 \end{aligned} \tag{2.82}$$

As is described extensively in References (1), (6), (9), and (11), these equations are solvable only in terms of the Jacobian elliptic integrals and for the special case of a symmetric spacecraft ($I_1 = I_2 \neq I_3$), these become

$$\begin{aligned} I_1\dot{\omega}_1 &= (I_3 - I_1)\omega_2\omega_3 \\ I_2\dot{\omega}_2 &= (I_1 - I_3)\omega_3\omega_1 \\ I_3\dot{\omega}_3 &= 0 \end{aligned} \tag{2.83}$$

which are now solvable in terms of simple trigonometric functions. Since ω_3 is constant, it can therefore be treated as one of the initial conditions of the problem. The remaining two equations can now be written (7 p. 206)

$$\begin{aligned} \dot{\omega}_1 &= \Omega\omega_2 \\ \dot{\omega}_2 &= -\Omega\omega_1 \end{aligned} \tag{2.84}$$

where Ω is the angular frequency

$$\Omega = \left(\frac{I_1 - 3}{I_1}\right)\omega_3 \tag{2.85}$$

Elimination of ω_2 in equation (2.84) leads to a differential equation for simple harmonic motion

$$\ddot{\omega}_1 = \Omega^2\omega_1 \tag{2.86}$$

with the typical solution (10 p. 525)

$$\omega_1 = A\cos(\Omega)\,t \tag{2.87}$$

where t is the time and A is the amplitude of the precession given by

$$A = \sqrt{\omega_1^2 + \omega_2^2} \tag{2.88}$$

The corresponding solution for ω_2 can be found by substituting this expression for ω_1, back into the first part of equation (2.84) to produce:

$$\omega_2 = -A\sin(\Omega)\,t \tag{2.89}$$

The solutions for ω_1 and ω_2 show that the angular velocity vector has a constant magnitude and rotates uniformly about the z-axis of the body with the angular frequency Ω – which is ultimately equivalent to the spin rate $\dot{\psi}$ (10 p. 490). Consequently, in the body coordinate frame, the angular velocity and angular momentum vectors cone about the spacecraft symmetry axis with an angular precession rate $\dot{\phi}$ and half-cone angle θ, where (11 pp. 491 - 492)

$$\tan(\theta) = \frac{A}{\omega_3} \tag{2.90}$$

Having determined the angular velocity, one can begin the process of describing the attitude motion. In order to simplify the problem conceptually, the angular momentum axis is typically aligned with the inertial z-axis, using a rotation matrix generated with the 3-1-3 Euler angles sequence. The body reference angular momentum is (10 p. 536)

$$\boldsymbol{L_B} = |L|(\boldsymbol{R}_{313}(\psi,\theta,\phi)\mathbf{u}_3)$$

$$= |L| \begin{bmatrix} \cos(\phi)\cos(\psi) - \sin(\phi)\cos(\theta)\sin(\psi) & \sin(\phi)\cos(\psi) + \cos(\phi)\cos(\theta)\sin(\psi) & \sin(\theta)\sin(\psi) \\ -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\theta)\cos(\psi) & \sin(\theta)\cos(\psi) \\ \sin(\phi)\sin(\theta) & -\cos(\phi)\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} |L|\sin(\theta)\sin(\psi) \\ |L|\sin(\theta)\cos(\psi) \\ |L|\cos(\theta) \end{bmatrix} \tag{2.91}$$

where $|L|$ is the magnitude of the angular momentum vector in body coordinates. Thus, in body coordinates, two of the Euler angles become spherical angles of the angular momentum vector. Because ω_3 is a constant, it follows that the coning angle is also a constant, and the coning rate $\dot{\theta}$ is equal to zero. Accordingly, the kinematics equation for the Euler angles becomes (2 p. 280)

$$\boldsymbol{\omega_B} = \begin{bmatrix} 0 & \cos(\psi) & \sin(\theta)\sin(\psi) \\ 0 & -\sin(\psi) & \sin(\theta)\cos(\psi) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ 0 \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \dot{\phi}\sin(\theta)\sin(\psi) \\ \dot{\phi}\sin(\theta)\cos(\psi) \\ \dot{\phi}\cos(\theta) + \dot{\psi} \end{bmatrix} \tag{2.92}$$

Recalling that $I_1 = I_2$, enables $\boldsymbol{L_B} = \boldsymbol{I_B}\boldsymbol{\omega_B}$ to be written as

$$||L|| \begin{bmatrix} \sin(\theta)\sin(\psi) \\ \sin(\theta)\cos(\psi) \\ \cos(\theta) \end{bmatrix} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_1 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} \dot{\phi}\sin(\theta)\sin(\psi) \\ \dot{\phi}\sin(\theta)\cos(\psi) \\ \dot{\phi}\cos(\theta) + \dot{\psi} \end{bmatrix} = \begin{bmatrix} I_1(\dot{\phi}\sin(\theta)\sin(\psi)) \\ I_1(\dot{\phi}\sin(\theta)\cos(\psi)) \\ I_3(\dot{\phi}\cos(\theta) + \dot{\psi}) \end{bmatrix} \tag{2.93}$$

Solving the first line of equation (2.93) for the precession rate $\dot{\phi}$ yields

$$\dot{\phi} = \frac{|L|}{I_1} \tag{2.94}$$

which, along with the last row of equation (2.92), can be substituted into equation (2.93) in order to find the spin rate $\dot{\psi}$, according to (10 pp. 537 - 538)

$$\omega_3 = \frac{|L|\cos(\theta)}{I_1} + \dot{\psi} = \frac{I_3\omega_3 + I_1\dot{\psi}}{I_1} \Rightarrow \dot{\psi} = \left(\frac{I_1 - I_3}{I_1}\right)\omega_3 \qquad (2.95)$$

$$\boldsymbol{\dot{\psi} = \left(\frac{I_1 - I_3}{I_1}\right)\dot{\phi}\cos(\theta) = \left(\frac{I_1 - I_3}{I_1 I_3}\right)|L|\cos(\theta)}$$

This result is exactly the same as what was given in equation (2.85) for the angular frequency Ω. Thus, for the torque-free motion of a symmetric rigid body, the Euler angle rates are all constant, and the solution for the attitude in terms of the Euler angles is (2 p. 280)

$$\begin{aligned} \phi &= \dot{\phi}(\Delta t) + \phi_0 \\ \theta &= \theta_0 \\ \psi &= \dot{\psi}(\Delta t) + \psi_0 \end{aligned} \qquad (2.96)$$

where ϕ is the precession angle, ϕ_0 is the initial precession direction (x-initial), ψ is the spin angle, ψ_0 is the initial spin direction (z-initial), and Δt is the change in time. The most important point of this discussion is that, in the inertial reference frame, the direction of the angular momentum remains fixed, and the spin axis and angular velocity vector cone about it. Thus, a point on the spin axis describes a circle in inertial space. An observer fixed in the space axes would see ω move on the surface of a space cone and correspondingly, an observer fixed in the body would see the angular velocity vector move on the surface of a body cone, as depicted below (2 p. 281).

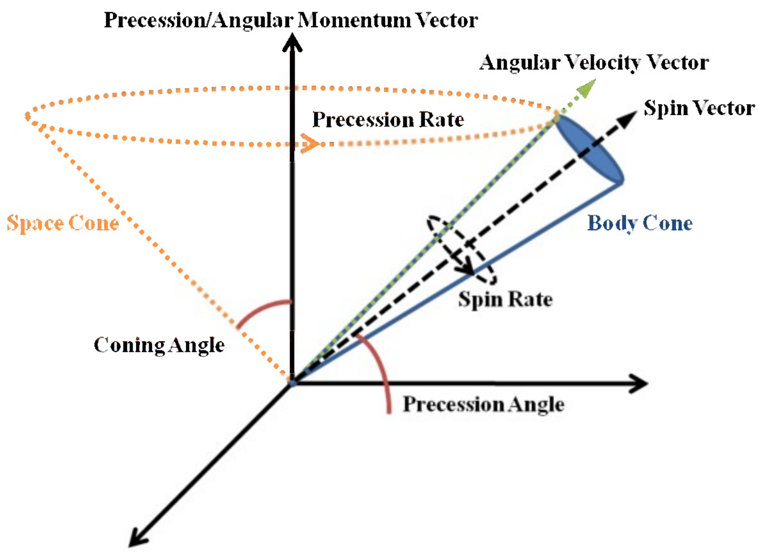


Figure 2.6 Torque-free motion of a symmetric rigid body (spin-precession motion) (11 p. 492)

If one regards the body and space cones as those swept out by the angular velocity vector with respect to the body's $\boldsymbol{z}$-axis and the angular momentum vector, respectively, then one cone must roll over (or within) the other, with the angular velocity vector representing the line of contact between the two. If the moment of inertia about the symmetry axis is less than that about the other two principal axes, then the body cone will be outside the space cone; and conversely, when the moment of inertia about the symmetry axis is the greater value, the body cone rolls around the inside of the space cone. In either case, the physical description of the motion is that the direction $\boldsymbol{\omega}$ of precesses in time about the axis of symmetry of the body (11 pp. 491 - 493).

2.3.4 Attitude Prediction and Simulation

Though the environmental torques that operate on the spacecraft are generally quite small, they cannot be ignored, since they act over a very long period of time. Similarly, while it is often the case that the difference between two principal moments of inertia is small compared with the difference of these from the third, the spacecraft is never exactly symmetric (even for the second-order moments which make up the inertia tensor). Accordingly, a major focus of the architecting process has been devoted to developing and integrating more refined models into the filtering algorithm in order to overcome these invalid underlying assumptions and perform multi-pass attitude prediction. Since the primary guess that initiates the measurement sub-process describes the motion of the spacecraft in terms of these body-symmetric, torque-free Euler angle parameters, it is necessary to switch to a more flexible representation. As has been mentioned before, the best choice is the unit quaternion, whose kinematic equation is linear and which satisfies only a single constraint that is easy to enforce.

Furthermore, accurate prediction of the time evolution of the attitude of a spacecraft requires three things: 1) a refined set of initial conditions (the output of the filtering sub-process discussed in the next chapter), 2) specifying the differential equations governing the rotational motion of the spacecraft (outlined in the preceding sections), and 3) a method of solution (12 p. 558). The dynamic and kinematic equations of motion are taken as a set of coupled differential equations and integrated using one the methods described in Section 2.3.4.1. The integration state vector **x**, consists of the attitude quaternion and three angular velocity body rates or angular momentum components. Given the nature of the problem, the method of solution is necessarily constrained to dynamic modeling, since the only other option is gyro modeling, which consists of using rate sensors or gyroscopes onboard the spacecraft to replace the dynamic model such that only the kinematic equations need be integrated. To properly integrate both the set of equations, requires detailed models be developed in order to estimate the physical characteristics of the spacecraft and external disturbance torques. Both the attitude estimation and prediction components of the overarching system architecture make use of different aspects of the models and algorithms described in the subsequent sections.

2.3.4.1 Numerical Integration Methods

Once the appropriate differential equations for attitude propagation have been established, it is necessary to choose a method for solving them. Because exact closed-form solutions of the complete equations to be integrated are almost never available, an approximation method is needed. The approach discussed in this section is direct integration using standard methods of numerical analysis. The equations of motion of attitude dynamics are a set of first-order coupled differential equations of the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}) \tag{2.97}$$

where **f** is a known vector function of the scalar t and the vector **x**. For simplicity, only the single differential equation $dx/dt = f(t,x)$ will be considered in this section, since the extension to coupled equations is fairly straightforward. Numerical algorithms will not give a continuous solution $x(t)$, but rather a discrete set of values x_n $(n = 1, 2, \dots)$ that are approximations to $x(t)$ a specific times $t_n = t_0 + ns$, where the parameter s is called the integration step size. Values of $x(t)$ for arbitrary times can be obtained by means of interpolation (Matlab function *interp*). For interpolation equations and procedures, please consult References (13) and (14). A minimum requirement on any algorithm is that it converge to the exact solution as the step size is decreased, that is,

$$\lim_{s \to 0} x_n = x(t_n) \tag{2.98}$$

where the number of steps n, is increased during the limiting procedure in such a manner that $ns = t_n - t_0$ remains constant.

Three important considerations in choosing an integration method are truncation error, round-off error, and stability. Truncation error, or discretization error, is the difference between the approximate and exact solutions $x_n - x(t_n)$, assuming that the calculations in the algorithm are performed exactly. According to Reference (15), “if the truncation error introduced in any step is of order s^{p+1}, the integration method is said to be of order p.” Round-off error is the additional error resulting from the finite accuracy of computer calculations due to “fixed word length”. An algorithm is unstable if errors introduced at some stage in the calculations (from truncation, round-off, or inexact initial conditions) propagate without bound as the integration proceeds. Truncation error is generally the limiting factor on the accuracy of numerical integration; it can be decreased by increasing the order of the method or by decreasing the step size. It is often useful to vary the step size during the integration, particularly if the “characteristic frequencies of the problem change significantly”; the ease with which this can be done depends on the integration method used (12 p. 560). The computation time required is usually proportional to the number of function evaluations, i.e., calculations of $f_n = f(t_n, x_n)$ that are required. Clearly, decreasing the step size increases the number of function evaluations for any fixed integration algorithm.

Two categories of integration techniques are commonly employed. In one-step methods, the evaluation of x_{n+1} requires knowledge of only x_n and f_n. Multistep methods, on the other hand, require information about previous values x_m and f_m for some number of values $m < n$ as well. One-step methods are widely used, due to the fact that they are relatively easy to apply – only x_0 and f_0 are needed as initial conditions – and the step size can be changed as necessary, without any additional computations. The most common one-step approach includes the classical R-stage Runge-Kutta method (15 p. 561)

$$x_{n+1} = x_n + s\varphi(t_n, x_n, s) \tag{2.99}$$

The increment function φ is a weighted average of R evaluations of $f(t, x)$ at different points in the integration interval and is given by (15 p. 561)

$$\varphi(t_n, x_n, s) = \sum_{r=1}^{R} c_r k_r \tag{2.100}$$

subject to the following constraint (15 p. 562)

$$\sum_{r=1}^{R} c_r = 1 \tag{2.100}$$

$$k_1 = f(t, x) \tag{2.101}$$

$$k_r = f\left(t + s a_r, x + s \sum_{s=1}^{r-1} b_{rs} k_s\right) \quad r = 2, 3, \dots, R \tag{2.102}$$

$$a_r = \sum_{s=1}^{r-1} b_{rs} \quad r = 2, 3, \dots, R \tag{2.103}$$

The different choices of the parameters c_r and b_{rs} define the different methods. Note that an R-stage method involves R function evaluations. The constants are always chosen to give the maximum order (and thus the maximum truncation error) for a given R; this order is R for $R = 1, 2, 3, 4$; $R - 1$ for $R = 5, 6, 7$; and $\leq R - 2$ for $R \geq 8$. For this reason, fourth-order four-stage Runge-Kutta methods are the most popular and take the following form (16 p. 603)

$$x_{n+1} = x_n + \frac{s}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.104}$$

$$k_1 = f(t_n, x_n) \tag{2.105}$$

$$k_2 = f(t_n + 0.5s, x_n + 0.5sk_1) \tag{2.106}$$

$$k_3 = f(t_n + 0.5s, x_n + 0.5sk_2) \tag{2.107}$$

$$k_4 = f(t_n + s, x_n + sk_3) \tag{2.108}$$

The algorithm summarized by equations (2.104) – (2.108) is utilized extensively in the attitude estimation problem described in Chapter 4, along with the Matlab function known as *ode45*, which uses the Dormand–Prince method for solving ordinary differential equations (17). The method is a member of the Runge–Kutta family of ODE solvers, which uses seven stages, but only six function evaluations per step to calculate fourth- and fifth-order accurate solutions. The difference between these solutions is then taken to be the error of the (fourth-order) solution. This error estimate is very convenient for adaptive step size integration algorithms.

A k-step multistep integration method has the form (16 p. 610)

$$x_{n+1} = s\sum_{j=0}^{k} \beta_j f_{n+1+j-k} \sum_{j=0}^{k-1} \alpha_j x_{n+1+j-k} \tag{2.109}$$

where different choices of the parameters α_j and β_j define alternative methods, and determine the number of back values of f_n and x_n. One drawback of these methods is that they are not self-starting; some other method, often a Runge-Kutta, must be used to calculate the first k values of f_n and x_n. Another disadvantage is that the step size changes are more difficult than for single step methods; additional back values must be available if the step size is increased, and intermediate back values must be calculated by interpolation if the step size is decreased. The most commonly used k-step algorithms utilize a procedure in which an explicit method $(\beta_j = 0)$, known as a predictor, calculates x_{n+1}; then f_{n+1} is evaluated and an implicit method $(\beta_j \neq 0)$, known as a corrector in this application, is used to obtain a refined value of x_{n+1}, followed by a second evaluation of f_{n+1} using the new x_{n+1}. The chief advantages of a predictor-corrector pair, such as the Adams-Bashforth-Moulton algorithm, is that only two function evaluations are needed per integration step and the difference between the predicted and corrected values of x_{n+1} give an estimate of the truncation error and can be used for step size control. This is in contrast to the Runge-Kutta methods, for which the step size changes are relatively easy, but estimates of truncation error are difficult to obtain. The 4th-order Adams-Bashforth-Moulton pair is given by (16 p. 627)

Predictor (explicit)

$$x_{n+1} = x_n + \frac{s}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \qquad (2.110)$$

Corrector (implicit)

$$x_{n+1} = x_n + \frac{s}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \qquad (2.111)$$

This specific algorithm is implemented in the Matlab function *ode113* (17). Higher order methods are also widely used and, unlike higher order Runge-Kutta methods, cost only additional storage space and not additional function evaluations.

In choosing an integration method, the factors of programming complexity, computer storage requirements, execution time, and computational accuracy must all be considered. For a specific application where the characteristic frequencies of the system are known to be nearly constant, a fixed-step method is appropriate. If the step size is limited by variations in the driving terms rather than by integration error (noisy input and/or low-accuracy requirements) or if function evaluations are relatively inexpensive, a Runge-Kutta method is preferred. If on the other hand, the integration step is set by integration error or function evaluations are expensive, a predictor-corrector method is better. Adams methods are favored in this class because they combine good stability properties with relatively low computer storage requirements and programming complexity. Because predictor-corrector algorithms provide an automatic estimate of local truncation error, they are the preferred variable-step methods. According to Reference (12), the best general-purpose integration methods currently available are programs with variable-step and variable-order Adams-Bashforth-Moulton integrators.

2.3.4.2 Geometric Satellite Model

The mass, center of mass, and inertia tensor for a solid, simple polyhedron of constant mass density require computing volume integrals VInt_i of the type (18 p. 536)

$$\text{VInt}_i = \int\limits_V p(x,y,z)dV \qquad (2.112)$$

where V is the volumetric region of integration and dV is an infinitesimal measure of volume. The function $p(x,y,z)$ is a polynomial selected from 1, x, y, z, x^2, y^2, z^2, xy, xz, and yz. We are interested in computing these integrals where V is the region bounded by a simple polyhedron. A volume integral may be reduced to a surface integral via the Divergence Theorem (18 p. 537)

$$\text{VInt}_i = \int\limits_V p(x,y,z)dV = \int\limits_V \nabla \cdot f(x,y,z)dV = \int\limits_A \mathbf{n} \cdot f(x,y,z)dA \qquad (2.113)$$

where A is the boundary of the polyhedron, dA is an infinitesimal measure of surface area, the function $f(x,y,z)$ is chosen so that $\nabla \cdot f(x,y,z) = p(x,y,z)$, and the vector $\mathbf{n}$ denotes outward-pointing, unit-length surface normals. The choices for f are given in the following table (19 pp. 2 - 5):

Table 2-1 Function values for volume integration

VInt	$p(x,y,z)$	$f(x,y,z)$	$q(x,y,z)$	c
0	1	$(x,0,0)$	$(x,0,0)$	1
1	x	$(x^2/2,0,0)$	$(x^2,0,0)$	1/2
2	y	$(0,y^2/2\ ,0)$	$(0,y^2\ ,0)$	1/2
3	z	$(0,0,z^2/2)$	$(0,0,z^2)$	1/2
4	x^2	$(x^3/3,0,0)$	$(x^3,0,0)$	1/3
5	y^2	$(0,y^3/3,0)$	$(0,y^3,0)$	1/3
6	z^2	$(0,0,z^3/3)$	$(0,0,z^3)$	1/3
7	xy	$(x^2y/2,0,0)$	$(x^2y,0,0)$	1/2
8	xz	$(0,z^2x/2,0)$	$(0,z^2x,0)$	1/2
9	yz	$(0,0,y^2z/2)$	$(0,0,y^2z)$	1/2

The wireframe model created in the software program known as X-Based Enhanced Lincoln Interactive Analysis System (XELIAS) and used primarily in the measurement making process – described extensively in Section 3.1.3 – can also be leveraged to generate an initial estimate for the inertia matrix and center of mass. The geometric model can be decomposed into a set of matrices consisting of vertex coordinates and the order in which they should be connected to create planar faces. The polyhedron surface is, thus, a union of thousands of triangular faces, so the surface integrals are effectively integrals in various planes. Projection of the faces onto coordinate planes is used to set up yet another reduction in dimension (19 p. 3). The figure below shows an example patch model for a simple cube shaped rigid body and the projection of a particular triangular facet into the $\boldsymbol{uv}$ plane.

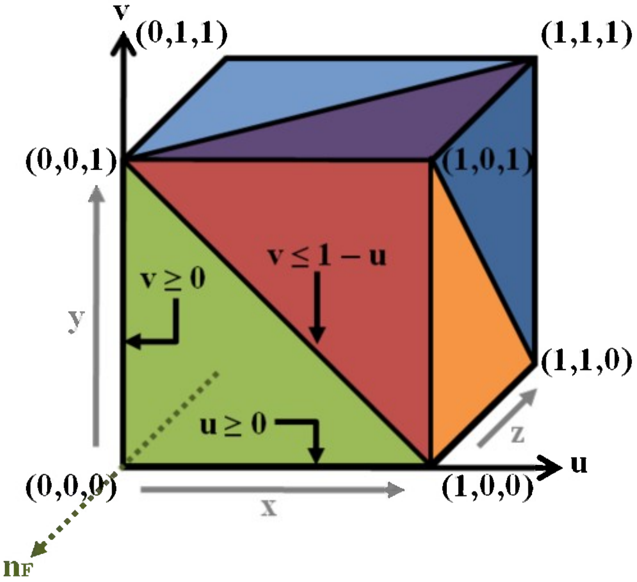


Figure 2.7 Patch model of a simple geometric shape

Green's Theorem, the two-dimensional analog of the Divergence Theorem, is employed to reduce the planar integrals to line integrals around the boundary of the projected faces. After the complicated integrals have been decomposed into their simplest form, they can be evaluated, combined, and propagated backward to evaluate the original ones. The integrals to be calculated have thus far been reduced to the form (19 p. 4)

$$\mathrm{VInt}_i = \int_V p(x,y,z)dV = \sum_{F\in A} (\mathbf{n}_F \cdot \boldsymbol{l}) \int_F f(x,y,z)dA = c\sum_{F\in A} (\mathbf{n}_F \cdot \boldsymbol{l}) \int_F q(x,y,z)dA \quad (2.114)$$

where $\mathbf{n}_F$ is the outward-pointing, unit-length normal to face F, $\boldsymbol{l}$ is a unit vector aligned with either the $\boldsymbol{i}$, $\boldsymbol{j}$,or $\boldsymbol{k}$ axis, $\boldsymbol{c}$ is the constant denominator values (1, $1/2$, or $1/3$) of the function f, and the function $q(x, y, z)$ are the corresponding numerator variables of f presented in Table 2-1. If the triangular facets are counterclockwise ordered and have vertices $p_i = (x_i, y_i, z_i)$, where $i = 1, 2$, or 3, then two of the vector edges $\mathbf{e}$ connecting the three points are given by

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}_1 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_2 - \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_1 \text{ and } \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}_2 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_3 - \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_1 \qquad (2.115)$$

A parameterization of the face projected into the uv plane is depicted in Figure 2.7 and can be expressed mathematically as (18 p. 541)

$$\mathbf{p}(u, v) = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_1 + u \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}_1 + v \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}_2 \qquad (2.116)$$

where $u \geq 0$ and $v \geq 0$, and $u + v \leq 1$, the infinitesimal measure of surface area is given by (18 p. 542)

$$dA = |\mathbf{e}_1 \times \mathbf{e}_2| dudv \qquad (2.117)$$

and the outward pointing unit-vector perpendicular to the face is

$$\mathbf{n}_F = \frac{\mathbf{e}_1 \times \mathbf{e}_2}{|\mathbf{e}_1 \times \mathbf{e}_2|} \qquad (2.118)$$

Therefore, the integrals in equation (2.114) can be reduce to (19 pp. 7 - 8)

$$\mathrm{VInt}_i = \int_V p(x, y, z) dV = c \sum_{F \in A} (\mathbf{e}_1 \times \mathbf{e}_2 \cdot \boldsymbol{l}) \int_0^1 \int_0^{1-v} q(x(u, v), y(u, v), z(u, v)) dudv \qquad (2.119)$$

where $x(u, v)$, $y(u, v)$, and $z(u, v)$ are the components of the parameterization in equation (2.116). Computing the integrals on the right-hand side of equation (2.119) has been done using the symbolic toolbox provided as part of the Matlab software program and yields the following set of equations (19 pp. 9 - 10):

$$\left.\begin{array}{c}
(\mathbf{n}_F \cdot \boldsymbol{i}) \int\limits_F x dA = \frac{n_1}{6} f_1(x) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{i}) \int\limits_F x^2 dA = \frac{n_1}{12} f_2(x) \quad (\mathbf{n}_F \cdot \boldsymbol{i}) \int\limits_F x^3 dA = \frac{n_1}{20} f_3(x) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{j}) \int\limits_F y^2 dA = \frac{n_2}{12} f_2(x) \quad (\mathbf{n}_F \cdot \boldsymbol{j}) \int\limits_F y^3 dA = \frac{n_2}{20} f_3(x) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{k}) \int\limits_F z^2 dA = \frac{n_3}{12} f_2(x) \quad (\mathbf{n}_F \cdot \boldsymbol{k}) \int\limits_F z^3 dA = \frac{n_3}{20} f_3(x) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{i}) \int\limits_F x^2 y dA = \frac{n_1}{60} \big(y_1 g_1(x) + y_2 g_2(x) + y_3 g_3(x)\big) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{j}) \int\limits_F y^2 z dA = \frac{n_2}{60} \big(z_1 g_1(y) + z_2 g_2(y) + z_3 g_3(y)\big) \\[2ex]
(\mathbf{n}_F \cdot \boldsymbol{k}) \int\limits_F z^2 x dA = \frac{n_3}{60} \big(x_1 g_1(z) + x_2 g_2(z) + x_3 g_3(z)\big)
\end{array}\right\} \qquad (2.120)$$

where the common subexpressions, f and g, required in the surface integrals may be obtained by means of some additional factoring and are listed below using a place holder variable w. (19 pp. 8 - 9)

$$a_n(w) = \sum_{i=0}^{n} w_0^{n-i} w_1^i \qquad (2.121)$$

$$f_0(w) = 1 \text{ and } f_n(w) = a_n(w) + w_2 f_{n-1}(w) \text{ for } n \geq 1 \qquad (2.122)$$

$$f_1(w) = w_0 + w_1 + w_2 = [w_0 + w_1] + w_2 \qquad (2.123)$$

$$f_2(w) = w_0^2 + w_0 w_1 + w_1^2 + w_2 f_1(w) = \Big[[w_0^2] + w_1\{w_0 + w_1\}\Big] + w_2\{f_1(w)\} \qquad (2.124)$$

$$f_3(w) = w_0^3 + w_0^2 w_1 + w_0 w_1^2 + w_1^3 + w_2 f_2(w) = w_0\{w_0^2\} + w_1\{w_0^2 + w_0 w_1 + w_1^2\} + w_2\{f_2(w)\} \qquad (2.125)$$

$$g_i(w) = f_2(w) + w_i f_1(w) + w_i^2 = \{f_2(w)\} + w_i(\{f_1(w)\} + w_i) \qquad (2.126)$$

The square brackets [ ] indicate that the subexpression is computed and saved in terms of temporary variables for later use, while the curly braces { } indicate the subexpression was computed earlier and can be accessed from the cache. Finally, the mass m, center of mass vector $\mathbf{c}_m$, and inertia matrix $\boldsymbol{I}$ can be assembled by first substituting the result of each component of equation block (2.120) back into equation (2.114) to solve for the volume integrals Int_i and then entering the outcome into (19 p. 2):

$$m = \text{VInt}_0 \tag{2.127}$$

$$\mathbf{c}_m = \left(\frac{1}{m}\right)\begin{bmatrix}\text{VInt}_1\\ \text{VInt}_2\\ \text{VInt}_3\end{bmatrix}_B \tag{2.128}$$

$$\boldsymbol{I}_B$$

$$= \begin{bmatrix}\text{VInt}_5 + \text{VInt}_6 - m\left(c_{my}^2 + c_{mz}^2\right) & -\text{VInt}_7 + mc_{mx}c_{my} & -\text{VInt}_9 + mc_{mz}c_{mx}\\ -\text{VInt}_7 + mc_{mx}c_{my} & \text{VInt}_4 + \text{VInt}_6 - m(c_{mz}^2 + c_{mx}^2) & -\text{VInt}_8 + mc_{my}c_{mz}\\ -\text{VInt}_9 + mc_{mz}c_{mx} & -\text{Int}_8 + mc_{my}c_{mz} & \text{VInt}_4 + \text{VInt}_5 - m\left(c_{mx}^2 + c_{my}^2\right)\end{bmatrix} \tag{2.129}$$

As will be describe in greater detail in Section 4.3.2 the initial guess for the inertia matrix parameters is then calculated through singular value decomposition and back solving for the parameters that correspond to the moment of inertia matrix generated using the process described above.

2.3.4.3 Environmental Torques

Environmental torques are naturally occurring body disturbances that impact the attitude of a spacecraft independent of any action it may take. To numerically integrate Euler's equations, the torque must be modeled as a function of time and the spacecraft's position and attitude. As was mention in the chapter introduction, in general, orbit and attitude are interdependent and nowhere is this union more evident than when analyzing the effects of environmental torques. For example, in a low altitude Earth orbit, the attitude will affect the atmospheric drag on the vehicle, which will impact the semi-major axis and eccentricity of the orbit; conversely, the orbit establishes the spacecraft position which determines both the atmospheric density and the magnetic field strength which will, in turn, affect the attitude. However, since only the impacts to attitude are of consequence to the subsequent problem, orbital information is always taken as input and the affects of attitude on orbit are overlooked. Even though this unilateral look at environmental torques enables the complex dynamics involved to be dramatically simplified, the non-cooperative nature of the problem implies that certain physical characteristics of the spacecraft, such as the coefficient of drag and surface reflectivity, will either need to be inferred through external observation or reasonably estimated. Even with such an approach, several other critical assumptions and approximations are required in order to reduce the modeling effort to a manageable level.

The objective of this section is to briefly outline the conventional models used to describe the dominant sources of attitude disturbance torques, which include the Earth's gravitational and magnetic fields, solar radiation pressure, and atmospheric drag. The relative importance of each of these torques to a given attitude prediction problem is a function of the vehicle's size, shape, mass, mass distribution, and altitude.

2.3.4.3.1 Gravity-Gradient Torque

All nonsymmetrical objects of finite dimension in orbit are subject to a gravitational torque because of the variation in the Earth's gravitational force over the object. Since there would be no gravitational torque in a uniform gravitational field, the magnitude of the force from the Earth is not constant but varies roughly as R^{-2}, where R is the distance from the geocenter (12 p. 566). The general expression for the gravity-gradient torque $\boldsymbol{N_{GG}}$ on a spacecraft of arbitrary shape and using a nonspherical Earth model can be expressed as (12 p. 567)

$$\boldsymbol{N_{GG}} = \frac{3\mu}{|R^3|}\mathbf{R} \times (\boldsymbol{I} \cdot \mathbf{R}) \tag{2.130}$$

where $\boldsymbol{I}$ is the inertia tensor, $\mathbf{R}$ is the position vector of the spacecraft' s center of mass with respect to the Earth's geometric center, and $\mu = GM_e = 398600.5\,\mathrm{km}^3/\mathrm{sec}^2$ is the Earth's gravitational constant. From equation (2.130), several general characteristics of the gravity-gradient torque may be readily inferred: 1) the torque is normal to the local vertical, 2) the torque is inversely proportional to the cube of the geocentric distance, and 3) the torque vanishes for a fully-symmetric spacecraft (i.e. the principal moments of inertia are all equal) (2 p. 283). Because the Earth is not perfectly spherical it becomes necessary to calculate general potential function for the Earth and the gravity-gradient tensor (20 pp. 128 - 129)

$$\mathbf{G}_F = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}_F \tag{2.131}$$

The elements of which are given by

$$g_{11} = \bar{\mathbf{r}}\left[1 - 3\left(\mathbf{R}_1{}^2\right)\right] + \bar{\bar{\mathbf{r}}}\left\{1 - 5\left[\mathbf{R}_1{}^2 + \mathbf{R}_3{}^2\right] + 35\left(\mathbf{R}_1{}^2\right)\left(\mathbf{R}_3{}^2\right)\right\} \tag{2.132}$$

$$g_{22} = \bar{\mathbf{r}}\left[1 - 3\left(\mathbf{R}_2{}^2\right)\right] + \bar{\bar{\mathbf{r}}}\left\{1 - 5\left[\mathbf{R}_2{}^2 + \mathbf{R}_3{}^2\right] + 35\left(\mathbf{R}_2{}^2\right)\left(\mathbf{R}_3{}^2\right)\right\} \tag{2.133}$$

$$g_{33} = \bar{\mathbf{r}}\left[1 - 3\left(\mathbf{R}_3{}^2\right)\right] + \bar{\bar{\mathbf{r}}}\left\{3 - 30\left(\mathbf{R}_3{}^2\right) + 35\left(\mathbf{R}_3{}^4\right)\right\} \tag{2.134}$$

$$g_{12} = g_{21} = -3(\bar{\mathbf{r}})(\mathbf{R}_1)(\mathbf{R}_2) + \bar{\bar{\mathbf{r}}}\{-5(\mathbf{R}_1)(\mathbf{R}_2) + 35(\mathbf{R}_1)(\mathbf{R}_2)(\mathbf{R}_3)^2\} \tag{2.135}$$

$$g_{13} = g_{31} = -3(\bar{\mathbf{r}})(\mathbf{R}_1)(\mathbf{R}_3) + \bar{\bar{\mathbf{r}}}\{-5(\mathbf{R}_1)(\mathbf{R}_3) + 35(\mathbf{R}_1)(\mathbf{R}_3)(\mathbf{R}_3)^2\} \tag{2.136}$$

$$g_{12} = g_{21} = -3(\bar{\mathbf{r}})(\mathbf{R}_2)(\mathbf{R}_3) + \bar{\bar{\mathbf{r}}}\{-5(\mathbf{R}_2)(\mathbf{R}_3) + 35(\mathbf{R}_2)(\mathbf{R}_3)(\mathbf{R}_3)^2\} \tag{2.137}$$

where $\mathbf{r}$ is the vector from the center of the Earth to the satellite the terms common to each component of the gravitational tensor, $\mathbf{R}$, $\bar{\mathbf{r}}$, and $\bar{\bar{\mathbf{r}}}$ are given by

$$\mathbf{R} = \frac{\mathbf{r}}{|r|} = \begin{bmatrix} \mathbf{r}_1/|r| \\ \mathbf{r}_2/|r| \\ \mathbf{r}_3/|r| \end{bmatrix} \tag{2.138}$$

$$\bar{\mathbf{r}} = \frac{\mu}{|r|^3} \tag{2.139}$$

$$\bar{\bar{\mathbf{r}}} = (1.5)J_2\bar{\mathbf{r}}\left(\frac{R_e}{|r|}\right)^2 \tag{2.140}$$

It is important to note that the gravitational tensor is referenced to the Earth Centered Earth Fixed (ECEF) reference frame $[\boldsymbol{GEN}]$, denoted by the subscript $\boldsymbol{F}$ rather than the Earth Centered Inertial (ECI) coordinate system which uses the subscript $\boldsymbol{I}$. As the name implies, the ECEF coordinate system is fixed to the rotating Earth, with an origin at the center of the Earth and fundamental plane that runs through the equator. The principal direction $\boldsymbol{G}$ axis is aligned with the Greenwich meridian ($0°$ longitude), the $\boldsymbol{N}$ axis runs through the North Pole, and the $\boldsymbol{E}$ axis points towards the East (3 p. 158). In order to transform between the ECEF and ECI coordinate frames, one need only rotate about the $\boldsymbol{z}$-axis by the Greenwich Mean Sidereal Time $\boldsymbol{\theta_g}$, which is the angle in degrees measured from the vernal equinox to the Greenwich meridian, as depicted in Figure 2.8 (3 p. 189).

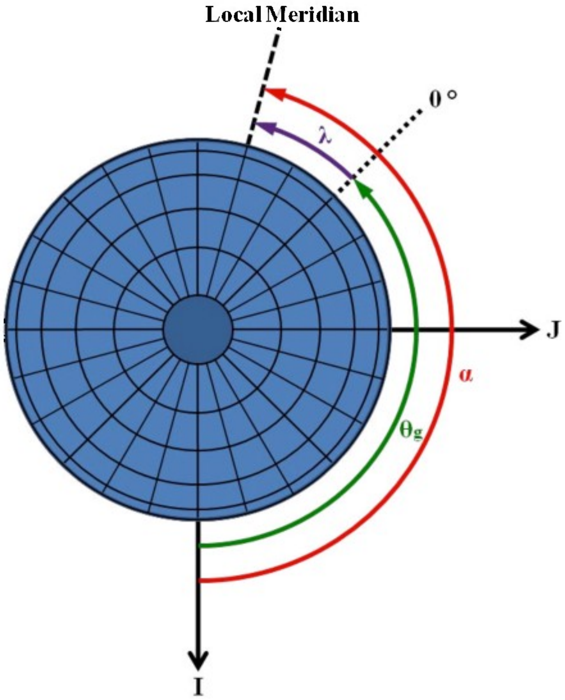


Figure 2.8 Spherical Angles: Greenwich mean sidereal time $\boldsymbol{\theta_g}$, longitude $\boldsymbol{\lambda}$, and right ascension $\boldsymbol{\alpha}$

The rotation angle can be calculated using the following set of equations (3 pp. 191 - 193):

$$\theta_g = \theta_{g0} + 1.002737909359795(360°)(D) \tag{2.141}$$

$$D = (Day\ Number - 1) + \left(\frac{hr}{24}\right) + \left(\frac{min}{1440}\right) + \left(\frac{sec}{86400}\right) \tag{2.142}$$

$$\theta_{g0} = mod360(100.4606184 + [36000.770053(T)] + [0.00038793(T^2)] - [2.6 \times 10^{-8}(T^3)])\ \text{deg} \tag{2.143}$$

$$T = \frac{(JD - 2451545.0)}{36525} \tag{2.144}$$

where D is the total elapsed time in solar days (please see Table B-1 in Appendix B for information on calculating day numbers) and θ_{g0} is the Greenwich Mean Sidereal Time, both from the epoch 1 January 00:00:00 of the year of interest. In equation (2.144), JD is the Julian date, which is the interval of time measured in days from the epoch 1 January 4713 B.C., 12:00:00 and can be calculated using equation (B.1) in Appendix B. Having converted solar time into the proper sidereal angle, the coordinate transformation for the gravitational tensor can then be found in the following manner (3 p. 173):

$$\mathbf{G}_I = \boldsymbol{R}_3(\theta_g)\mathbf{G}_F\boldsymbol{R}_3^T(\theta_g) \Rightarrow \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix}_I$$

$$= \begin{bmatrix} \cos(\theta_g) & -\sin(\theta_g) & 0 \\ \sin(\theta_g) & \cos(\theta_g) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix}_F \begin{bmatrix} \cos(\theta_g) & \sin(\theta_g) & 0 \\ -\sin(\theta_g) & \cos(\theta_g) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.145}$$

and the gravity gradient torque is then given by (21 p. 15)

$$\begin{bmatrix} N_{GG1} \\ N_{GG2} \\ N_{GG3} \end{bmatrix}_B = \begin{bmatrix} (I_{22} - I_{33})G_{23} + (G_{33} - G_{22})I_{23} + I_{12}G_{13} - I_{13}G_{12} \\ (I_{33} - I_{11})G_{13} + (G_{11} - G_{33})I_{13} + I_{23}G_{21} - I_{21}G_{23} \\ (I_{11} - I_{22})G_{12} + (G_{22} - G_{11})I_{12} + I_{31}G_{32} - I_{32}G_{31} \end{bmatrix}_B \tag{2.146}$$

2.3.4.3.2 Magnetic Torque

Magnetic torques results from the interaction between the spacecraft's residual magnetic field and the geomagnetic field. For near Earth spacecraft at altitude greater than 500 km, magnetic torques are often the principal disturbance affecting spacecraft attitude. The magnetic torque $\boldsymbol{N_M}$ is typically expressed as (12 p. 575)

$$\boldsymbol{N_M} = \mathbf{m} \times \mathbf{B} = \begin{bmatrix} m_2 B_3 - m_3 B_2 \\ m_3 B_1 - m_1 B_3 \\ m_1 B_2 - m_2 B_1 \end{bmatrix} \tag{2.147}$$

where $\boldsymbol{m}$ is the effective magnetic moment of the spacecraft in $\mathrm{A \cdot m^2}$ and $\boldsymbol{B}$ is the geocentric magnetic flux density with unit of $\mathrm{Wb/m^2}$. In order to minimize errors in evaluating the magnetic torques, the magnetic field of the Earth was computed from the spherical harmonic model and Gaussian coefficients g_n^m and h_n^m, obtained from the 10th generation international geomagnetic reference field (IGRF) database created by the National Oceanic and Atmospheric Administration (NOAA) (22). The Gaussian coefficients are determined empirically by a least-squares fit to measurements of the field and include terms through 13$^{\text{th}}$ order and degree. Since the atmosphere is essentially nonmagnetic, the equations for the geocentric field intensity components may be expressed in terms of the magnetic flux density $\mathbf{B}$, according to (20 pp. 117 - 118) and (23 p. 783), as

$$B_1 = -\sum_{n=1}^{n_{max}} \sum_{m=0}^{n} \left(\frac{R_m}{r}\right)^{n+2} \left(g_n^m \cos(m\lambda) + h_n^m \sin(m\lambda)\right) \frac{dP_n^m(\Theta)}{d\Theta} \tag{2.148}$$

$$B_2 = \sum_{n=1}^{n_{max}} \sum_{m=0}^{n} \frac{m}{\sin(\Theta)} \left(\frac{R_m}{r}\right)^{n+2} \left(g_n^m \sin(m\lambda) - h_n^m \cos(m\lambda)\right) P_n^m(\Theta) \tag{2.149}$$

$$B_3 = \sum_{n=1}^{n_{max}} \sum_{m=0}^{n} (n+1) \left(\frac{R_m}{r}\right)^{n+2} \left(g_n^m \cos(m\lambda) + h_n^m \sin(m\lambda)\right) P_n^m(\Theta) \tag{2.150}$$

where R_m is the mean radius of the Earth (6371.2 km), as opposed to the Earth's equatorial radius R_e which is given as 6378.137 km. Additionally, the Schmidt-normalized associated Legendre functions $P_n^m(\Theta)$ and their derivatives may be evaluated from the following recursive formulas (23 pp. 775 - 776)

$$\left.\begin{aligned} P_0^0(\Theta) &= 1 \\ \frac{dP_0^0(\Theta)}{d\Theta} &= 0 \end{aligned}\right\} n = m = 0 \tag{2.151}$$

$$\left.\begin{aligned} P_n^n(\Theta) &= \sqrt{\frac{2n-1}{2n}}\,P_{n-1}^{n-1}(\Theta)\sin(\Theta) \\ \frac{dP_n^n(\Theta)}{d\Theta} &= \sqrt{\frac{2n-1}{2n}}\left[\frac{dP_{n-1}^{n-1}(\Theta)}{d\Theta}\sin(\Theta) + P_{n-1}^{n-1}(\Theta)\cos(\Theta)\right] \end{aligned}\right\} n = m \geq 1 \qquad (2.152)$$

$$\left.\begin{aligned} P_n^m(\Theta) &= \frac{2n-1}{\sqrt{n^2-m^2}}\left[P_{n-1}^m(\Theta)\cos(\Theta) - \frac{\sqrt{(n-1)^2-m^2}}{2n-1}P_{n-2}^m(\Theta)\right] \\ \frac{dP_n^m(\Theta)}{d\Theta} &= \frac{2n-1}{\sqrt{n^2-m^2}}\left[\frac{dP_{n-1}^m(\Theta)}{d\Theta}\cos(\Theta) - P_{n-1}^m(\Theta)\sin(\Theta) - \frac{\sqrt{(n-1)^2-m^2}}{2n-1}\frac{dP_{n-2}^m(\Theta)}{d\Theta}\right] \end{aligned}\right\} n > m \geq 0$$

$$(2.153)$$

In this set of equations Θ, λ, and r are the colatitude ($\pi/2 -$ latitude), longitude, and range of the spacecraft respectively; values for which can be easily obtained by generating the corresponding report in the software program Satellite Tool Kit (STK). A detailed set of directions for creating the necessary text file is provided in Appendix C. The desired principal axis components of the geomagnetic field are expressed in terms of a basis in which the $\boldsymbol{R}$ axis always points from the Earth's center along the radius vector towards the satellite as it moves through the orbit. The $\boldsymbol{T}$ axis points in the direction of the velocity vector and is perpendicular to the radius vector and the $\boldsymbol{N}$ axis is normal to the orbital plane (3 pp. 162 - 163). In order to transform from the Satellite Orbit Coordinate System [$\boldsymbol{RTN}$] coordinate system (given the subscript S) to the ECI frame, requires that the longitude λ be converted into right ascension α (also known as Local Sidereal Time), using the values found in equations (2.141) – (2.144) and the simple formula (21 pp. 34 - 35)

$$\alpha = \lambda + \theta_g \qquad (2.154)$$

the combined rotation matrix can then be expressed as

$$\mathbf{B}_I = \boldsymbol{R}_1(\alpha)\boldsymbol{R}_2(\Theta)\mathbf{B}_S \Rightarrow \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}_I = \begin{bmatrix} \cos(\Theta) & 0 & -\sin(\Theta) \\ \sin(\Theta)\sin(\alpha) & \cos(\alpha) & \cos(\Theta)\sin(\alpha) \\ \sin(\Theta)\cos(\alpha) & -\sin(\alpha) & \cos(\Theta)\cos(\alpha) \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}_S \qquad (2.155)$$

2.3.4.3.3 Aerodynamic Torque

Aerodynamic torque is created by the impact of rarefied air particles with the satellite surface and depends primarily on the atmospheric density at a given altitude. At altitude below about 800 km the aerodynamic torque is generally the most important of environmental disturbance torques. The force due to the impact of atmospheric molecules on the spacecraft surface can be modeled as an elastic impact without reflection, in which the incident particle's energy is generally completely absorbed upon collision (12 pp. 573 - 574).

As was discussed in Section 2.3.4.2, the surface area of the satellite can be decomposed into simple geometric shapes, enabling the aerodynamic force acting on each individual shape making up the vehicle's body, to be considered independently and summed together in order to find the total torque. Because the resultant forces on any given panel of the spacecraft acts at the center of pressure of the exposed surface area, the same geometric model used in estimating the moment of inertia matrix, can also be used to locate the corresponding centroid of each triangular facet that comprises the satellite as depicted in Figure 2.9 (12 p. 574).

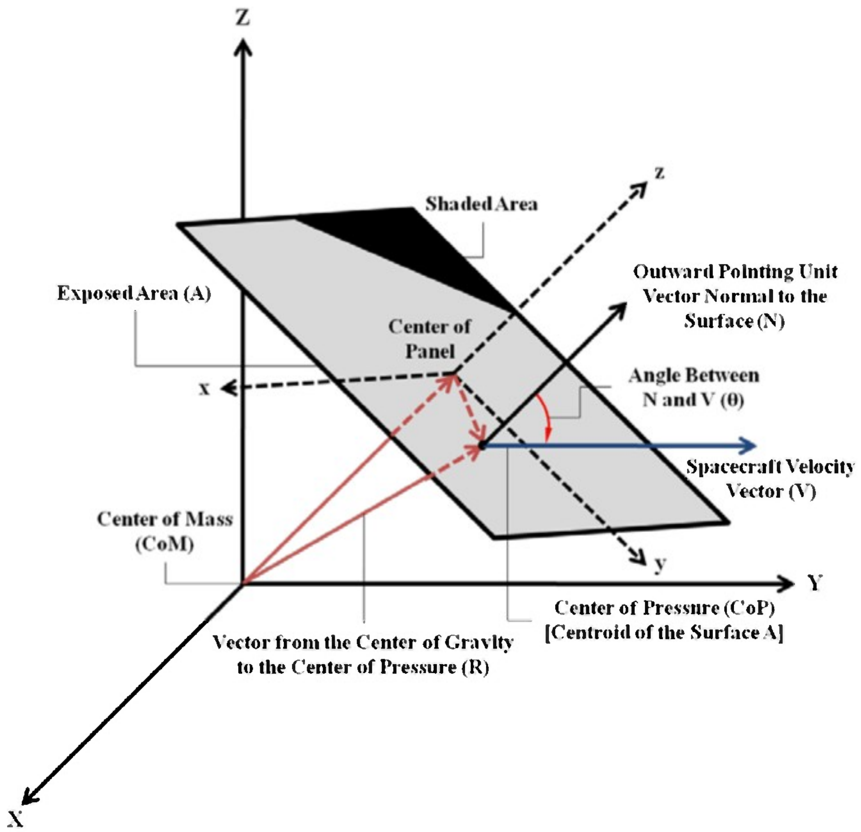

Z
z
Shaded Area
Exposed Area (A)
Outward Pointing Unit Vector Normal to the Surface (N)
Center of Panel
x
Angle Between N and V (θ)
Spacecraft Velocity Vector (V)
Center of Mass (CoM)
y
Y
Center of Pressure (CoP) [Centroid of the Surface A]
Vector from the Center of Gravity to the Center of Pressure (R)
X

Figure 2.9 Aerodynamic drag acting at the centroid of each surface element

This approach avoids time-consuming repetitive evaluation of surface integrals by taking advantage of the regular geometry of the panels and the fixed orientation of each surface element with respect to the center of mass/center of rotation. If spacecraft is characterized as being a finite collection of surfaces, then the aerodynamic torque $\boldsymbol{N_A}$ can be written as (2 pp. 285 - 286)

$$\boldsymbol{N_A} = -\frac{1}{2} C_D \rho \sum_{i=1}^{n} A_i (\mathbf{n}_i \cdot \mathbf{v})(\mathbf{r}_i \times \mathbf{v}) \qquad (2.156)$$

and the summation is over that part of the spacecraft for which $\mathbf{n}_i \cdot \mathbf{v} > 0$, that is, the portion of the spacecraft surface facing into the wind. In the above equation, C_D is the drag coefficient (which is assumed to be 2 since no measured value is available), ρ is the atmospheric density, $\mathbf{r}_i$ is the position vector to the center of pressure of the i^{th} surface, measured from the center of mass, $\mathbf{v}$ is the velocity vector in ECI coordinates obtained from STK, A_i is the area of the i^{th}

surface element, and $\mathbf{n}_i$ is the outward pointing unit-vector, normal to the surface element. For a flat surface, if there are no shadowing effects, the center of pressure is located at the geometric center (centroid) of the face.

Unfortunately, the calculation of aerodynamic torques is in general, not very accurate, due to large uncertainties in the atmospheric density, drag coefficient, and shadowing effects. Furthermore, at altitudes above 200 km, the atmospheric density is sensitive to solar activity, which may cause the lower atmosphere to expand, with sometime severe consequences (20 pp. 109 - 110). In order to mitigate errors to the greatest extent possible, values for the atmospheric density and the solar flux F10.7, are computed using the NRL-MSISE-2000 Atmospheric Model and Space Physics Interactive Data Resource (SPIDR), respectively. The NRL-MSISE-2000 model uses the daily F10.7 value, the orbital parameters of the satellite (namely the latitude, longitude, and altitude), and characteristics of certain atmospheric molecules, to compute the neutral temperature and density of the Earth's atmosphere over a fixed time interval. To obtain ρ values at specific observation times in a given pass requires the use of interpolation, which can be done using the Matlab function *interp1*. For more detailed information on the density model used in this thesis, consult Reference (24).

2.3.4.3.4 Solar Radiation Torque

Radiation incident on a spacecraft's surface produces a force which results in a torque about the satellite's center of mass. The surface is subject to radiation pressure or force per unit area equal to the vector difference between the incident and reflected momentum flux. Because the solar radiation varies as the inverse square of the distance from the Sun, the pressure is essentially altitude independent for spacecraft in Earth's orbit (12 p. 570). The major factors determining the radiation torque are the intensity and spectral distribution of the incident radiation, the geometry of the surface and its optical properties, and the orientation of the Sun vector relative to the spacecraft (20 pp. 129 - 130). Though electromagnetic radiation is also reflected by and emitted from the Earth and its atmosphere, these contributions to the overall torque are small and will, therefore, be ignored. The solar radiation torque, from a geometrical standpoint, is very similar to the aerodynamic torque, except that the incident particles are photons rather than air molecules. Radiation from the Sun may be either completely absorbed, specularly reflected, or diffusely reflected, as shown in Figure (2.10) (12 p. 571).

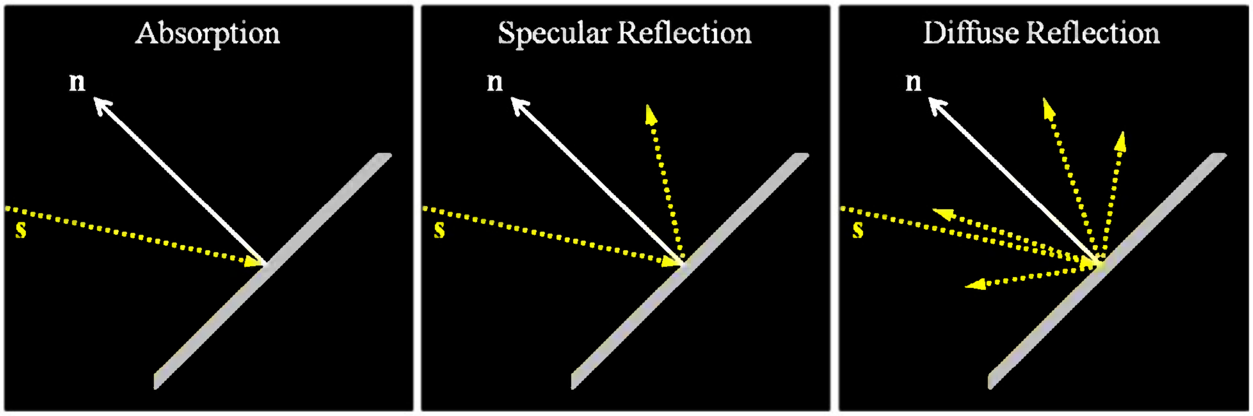


Figure 2.10 Absorption and Reflection of Incident Radiation

The probabilities of each of these occurrences are called the coefficients of absorption, specular reflection, and diffuse reflection, and satisfy

$$c_a + c_s + c_d = 1 \tag{2.157}$$

The total solar torque $\boldsymbol{N_{SR}}$ acting on a collection of planes about the center of mass of the spacecraft is the vector sum of the individual torques calculated by (12 p. 572)

$$\boldsymbol{N_{SR}} = s_i \sum_{i=1}^{n} \mathbf{r}_i \times \left[-\frac{F_e}{c} A_i (\mathbf{n}_i \cdot \mathbf{s}) \left[(1 - c_s)\mathbf{s} + 2\left(c_s (\mathbf{n}_i \cdot \mathbf{s}) + \frac{1}{3} c_d \right) \mathbf{n}_i \right] \right] \tag{2.158}$$

where F_e is the solar constant $(1400\ \mathrm{W/m^2})$, c is the speed of light $(3 \times 10^8\ \mathrm{m/sec})$, $\mathbf{r}_i$ is the position vector to the center of pressure of the i^{th} surface, measured from the center of mass, $\mathbf{s}$ is the unit vector pointing from the spacecraft to the Sun obtained from STK, along with the solar intensity s_i (a percentage value, typically 100% or 0%, that changes as the satellite passes in and out of eclipse), A_i is the area of the i^{th} surface element, and $\mathbf{n}_i$ is the outward pointing unit-vector, normal to the surface element. Again the summation is limited to that part of the surface for which $\mathbf{n}_i \cdot \mathbf{s} > 0$.

3 Attitude Determination and Estimation

For some applications, it is useful to determine the attitude of a spaceborne object over time, via non-cooperative means. For example, suppose a satellite's attitude control system has failed and the satellite is in an uncontrolled tumble. The satellite owners wish to operate on the satellite to retrieve data or restabilize the object if possible. In many cases, the telemetry reveals nothing pertinent to the unstable motion or the communication link between the satellite and ground station may be severed or degraded, rendering the traditional array of onboard sensors unusable. Fortunately, from inverse synthetic aperture radar (ISAR) images of the target, the Space Situational Awareness Group is able to obtain attitude measurements helpful in the motion analysis. However, these measurements are generally not obtained in a straightforward manner, as considerable processing of the raw radar data is needed to form usable images for the attitude measurement process (known as image-model matching). Given the unconventional manner in which the attitude is being sensed in this particular problem, considerable attention is given in this chapter to understanding how the images are generated and the nature of the observations that can be derived from them.

A filtering algorithm is needed to calculate the motion of the object from the inherently noisy radar measurements and establish the uncertainty in the obtained motion solution. Since the dynamics equations are nonlinear, it is highly unlikely that a closed-form solution to the system exists. Additionally, since the number of images yielding measurements exceeds the minimum number necessary to solve for the unknown quantities characterizing the motion, the problem is said to be overdetermined. It is therefore useful to employ some sort of nonlinear least-squares procedure. While a number of least-squares estimators exist, the backward-smoothing extended Kalman filter (BSEKF) has been selected, due to the unique challenges presented by the ISAR attitude measurements and nonlinear system dynamics. As will be discussed in greater detail in the subsequent sections, the BSEKF possesses the attributes of a number of different algorithms. Also, the filter has been shown to have significantly better convergence reliability and accuracy when compared against other leading algorithims in the field, namely the extended Kalman filter (EKF) and unscented Kalman filter (UKF), for estimation problems that start with large initial attitude or attitude rate errors. In the final phase of the entire process, output displays describe the computed motion to the analyst and make predictions about the future attitude of the spacecraft. With this critical external information, it may be possible for operators to then identify the source of the anomaly, develop resolution plans, and hopefully, reestablish communication with their valuable space asset.

Figure 3.1 summarizes the critical objects and processes involved in attitude determination and estimation. While by no means comprehensive, this high-level diagram forms the basis for much of what will be covered, not only in this chapter, but throughout the remainder of the thesis. Those algorithms and techniques of particular importance to the subsequent discussion have been highlighted and referenced by section.

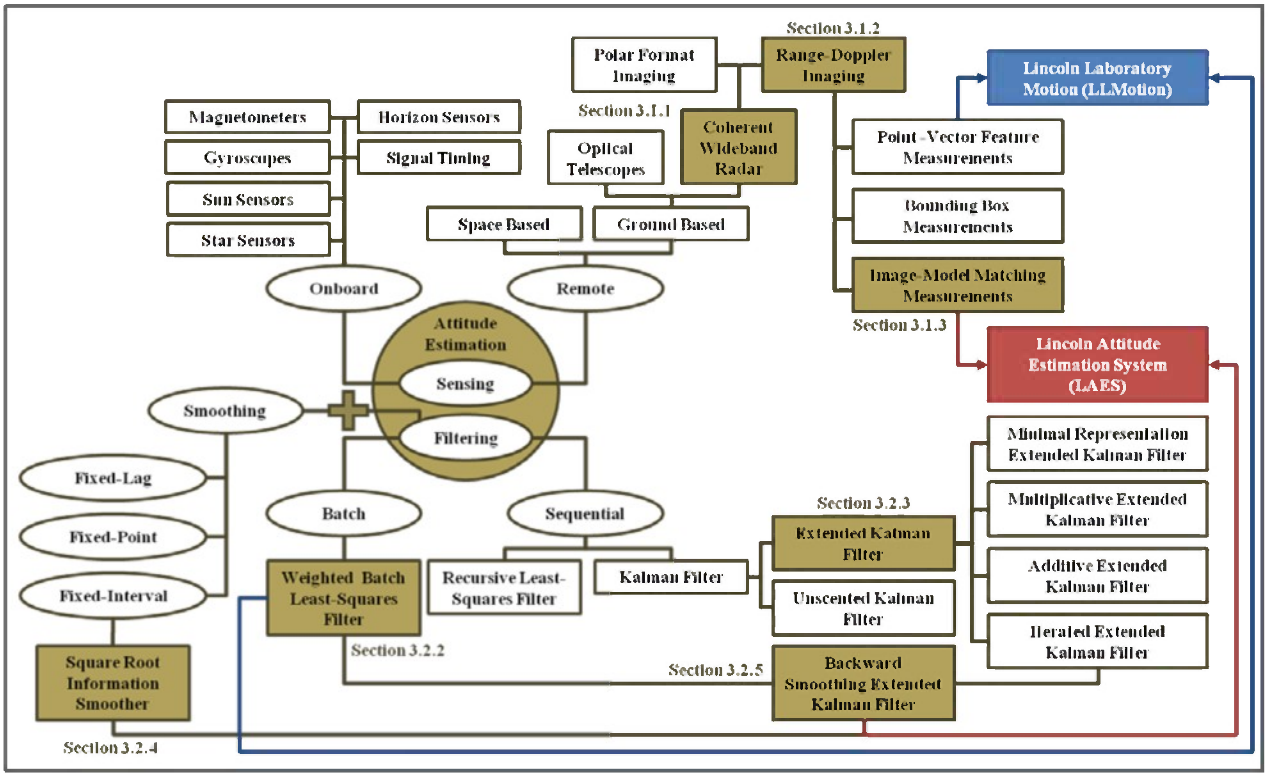

Section 3.1.2
Polar Format Imaging
Range-Doppler Imaging
Lincoln Laboratory Motion (LLMotion)
Magnetometers
Horizon Sensors
Section 3.1.1
Coherent Wideband Radar
Point-Vector Feature Measurements
Gyroscopes
Signal Timing
Optical Telescopes
Sun Sensors
Bounding Box Measurements
Star Sensors
Space Based
Ground Based
Image-Model Matching Measurements
Onboard
Remote
Section 3.1.3
Attitude Estimation
Lincoln Attitude Estimation System (LAES)
Sensing
Smoothing
Filtering
Minimal Representation Extended Kalman Filter
Fixed-Lag
Section 3.2.3
Multiplicative Extended Kalman Filter
Batch
Sequential
Fixed-Point
Extended Kalman Filter
Additive Extended Kalman Filter
Weighted Batch Least-Squares Filter
Recursive Least-Squares Filter
Kalman Filter
Fixed-Interval
Unscented Kalman Filter
Iterated Extended Kalman Filter
Section 3.2.2
Square Root Information Smoother
Section 3.2.5
Backward Smoothing Extended Kalman Filter
Section 3.2.4

Figure 3.1 Decomposition of the attitude estimation process into relevant sub-processes and different types of sensors/measurement techniques and data filters/estimators

3.1 The Measurement Process

Radar images are formed by coherently combining many observations of an object, called the target, over a range of frequencies and viewing geometries. The imaging process is coherent in that it uses the phase as well as the amplitude of the target echoes. Anything that degrades system coherence also degrades the quality of the final image. One such item is uncorrected radial target motion, which causes the radar echoes to be shifted in range. These shifts may be removed from the signal during the imaging process using a technique called motion compensation. A high level of precision in the measurement of the satellite's translational and rotational motion is needed to implement motion compensation with the required accuracy. In order to achieve the necessary precision, the radar data itself can be used to refine the orbital and attitude estimates in an algorithmic process called autofocus.

At a fundamental level, radar imaging techniques attempt to extract information about the spatial arrangement of a target's various scattering components from the radar echoes. That information is encoded in differences in the returns as a function of frequency and viewing geometry. The amount of information contained in the variations over frequency is limited by the frequency span, or bandwidth, of the transmitted waveform. Similarly, the amount of information contained in the variations over viewing geometry is limited by the span of the viewing angle. Imaging of satellites from the ground falls into a class of problems in which target motion accounts for most of the change in viewing geometry over the observation period. Such target-imaging situations are given the name inverse synthetic aperture radar (ISAR). The term inverse is used to emphasize the importance of target motion, in contrast to synthetic aperture radar (SAR), where platform motion accounts for most of the change in viewing geometry over the observation period. From the ISAR perspective, radar imaging constructs a map of the target's complex scattering amplitude from the collection of radar returns, each of which is a projection of that function at a specific rotation angle. The description of radar imaging in terms of projections, and the fundamental insight that description provides, forms the basis of the range-Doppler imaging algorithm.

The collection of algorithms used by the Space Situational Awareness Group to generate and process radar images is known as the Advanced Radar Imaging Environment Software (ARIES). For the purposes of this thesis project, only a basic understanding of how the raw radar returns are used to form images is needed. The intent is to provide a sufficient amount background in the image generation process in order to support more critical discussions concerning: 1) how the attitude measurements are made and 2) the challenges associated with using radar imagery as the basis for what's called image-model matching. To that end, the subsequent sections will focus on the FFT radar-Doppler imaging technique, general motion compensation methods, and the specifics of the measurement making process.

3.1.1 Inverse Synthetic Aperture Radar Basics

An inverse synthetic aperture radar system coherently combines signals obtained from a single ground-based aperture as it observes a rotating target. The rotational motion of the target provides the aspect change needed to approximate the result that could otherwise only be achieved by a larger antenna aperture (25). Consider the simplified ISAR system depicted in Figure 3.2.

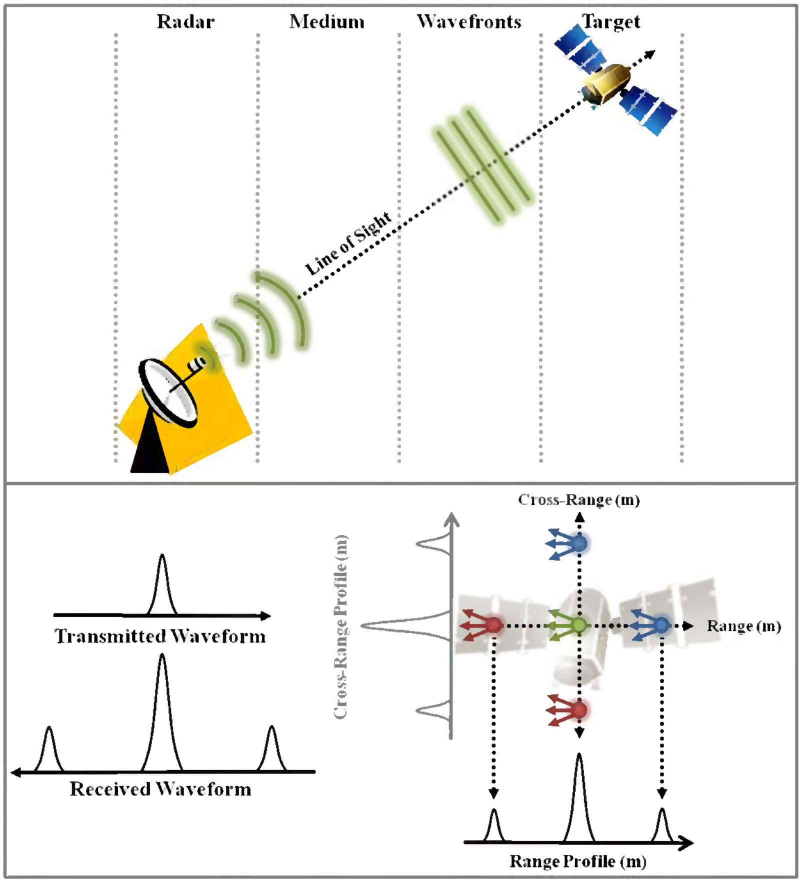


Figure 3.2 A simple ISAR system consisting of a coherent wideband radar sensor imaging a satellite (26 p. 15)

This simple two-dimensional system includes a radar transmitter/receiver, a medium through which the radar signal propagates, and an object of interest, called the target. For illustration purposes, the complex satellite being imaged has been reduced to five point scatterers. The vector from the sensor to the satellite is called the radar line of sight (RLOS) and defines the range direction. In this two-dimensional case, the orthogonal direction is defined as the cross-range direction. The radar emits a burst of energy, called a pulse, which propagates through the medium to the target, interacts with the satellite scattering centers, and is reflected back to the radar. It is assumed that the distance to the target – the range – is sufficiently large for the wavefronts to be considered planar (27 pp. 2 - 3). According to Reference (28), this is equivalent to "assuming that the target is much smaller than a Fresnel zone radius in cross-range extent" – an assumption that dramatically simplifies the imaging algorithm. For the monostatic configuration shown in Figure 3.2, the radar receiver observes only the reflected part of the total scattered signal. The ideal point scatterers composing the spacecraft each produce an echo that is an exact replica of the transmitted waveform. The observed signal, called the range profile, is the sum of the individual echoes. The range profile begins with a reflection from the point on the target nearest the radar and ends with the reflection from the furthest point. The three points on the vertical line through the origin are at the same range from the radar, so their echoes add to form a single component in the range profile having three times the amplitude of that due to the single point. This illustrates the idea that range profiles are projections, or integrals of the target's complex scattering amplitude. More precisely, in two-dimensions the range profile is the result of integrating the complex scattering amplitude along lines in the cross-range direction. It is a complex amplitude signal, in the sense that magnitude and phase are preserved for use in coherent imaging process (26 p. 16).

Note that the reflected energy from a single pulse yields a one-dimensional signal, called a radar cross section (RCS), which contains direct information about the locations of the target's components in the range direction but no information about their positions in cross-range. Also, the bandwidth of the transmitted pulse determines the accuracy with which the locations can be measured in the range direction. Now, if the target is rotated, and another pulse it transmitted, the range profile may look like that plotted in the figure below.

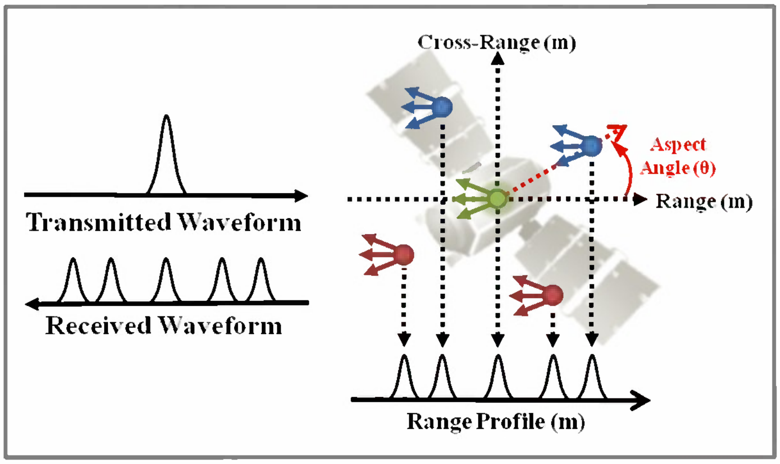


Figure 3.3 A simple ISAR system in which the target attitude has changed (26 p. 17).

As the satellite rotates, its radar cross section, or projection into the radar line of sight, changes. The difference between the range profiles, such as those illustrated in Figures 3.2 and 3.3, provides information about the locations of the target's scattering centers in the cross-range direction. If a number of range profiles are collected at various aspect angles, they can be processed to separate target components into appropriate range and cross-range locations (27 p. 8). Note that the range and cross-range directions, defined with respect to the individual radar pulses, rotate with respect to the target. However, in most of the descriptions to follow, these are defined with respect to the image (where they are constant) and by the corresponding directions of the radar pulses that occur at the center of the target observation period.

Returning to the Figure 3.3, the angle of the target coordinate frame relative to a stationary frame is called the aspect angle, and the total angular span over which data is collected is called the ISAR angular aperture. The aperture determines the cross-range resolution of the final image, and the angular spacing between adjacent range profiles determines the unambiguous cross-range extent of the image area (28 p. 371). Thus far, an idealized picture of the radar imaging process has been presented. For three-dimensional targets, the axis of rotation is typically neither perpendicular to the radar LOS nor stationary with respect to the radar. The geometry of the configuration and the variations in the aspect angle due to changes in attitude during data acquisition in an actual system are most easily visualized by considering the system geometry from the perspective of the target spacecraft. To assist in the visualization, a preliminary coordinate frame is fixed to the target with an arbitrary orientation. That frame is then rotated, with respect to the vehicle, to obtain the target-fixed frame appropriate for imaging (this should not be confused with the body-fixed frame discussed in the previous chapter). In choosing the orientation of the target-fixed (image-plane) coordinate system, it is convenient to imagine a unit sphere fixed to the spacecraft at the center of mass. Then consider the point at which the line of sight vector intersects that sphere each time a radar pulse is transmitted. In operational ISAR systems, those pierce points generally follow some curved trajectory on the surface of the unit-sphere, as illustrated in Figure 3.4a (27 p. 13). The orientation of the target-fixed frame is depicted in Figure 3.4b and is chosen such that the xy plane provides an optimal fit to the intersection points, say, by minimizing the root-mean-square (rms) distance from the points to the plane. The y-axis is selected as the direction of the RLOS at the center of the aperture, also known as the angular span. The location of the pierce points can now be specified by two angles: the azimuthal or in-plane angle α and the polar or out-of-plane angle β. The image is ultimately formed in the xy plane, where the x-axis defines the cross-range direction of the image, and the y-axis defines the range (27 p. 14).

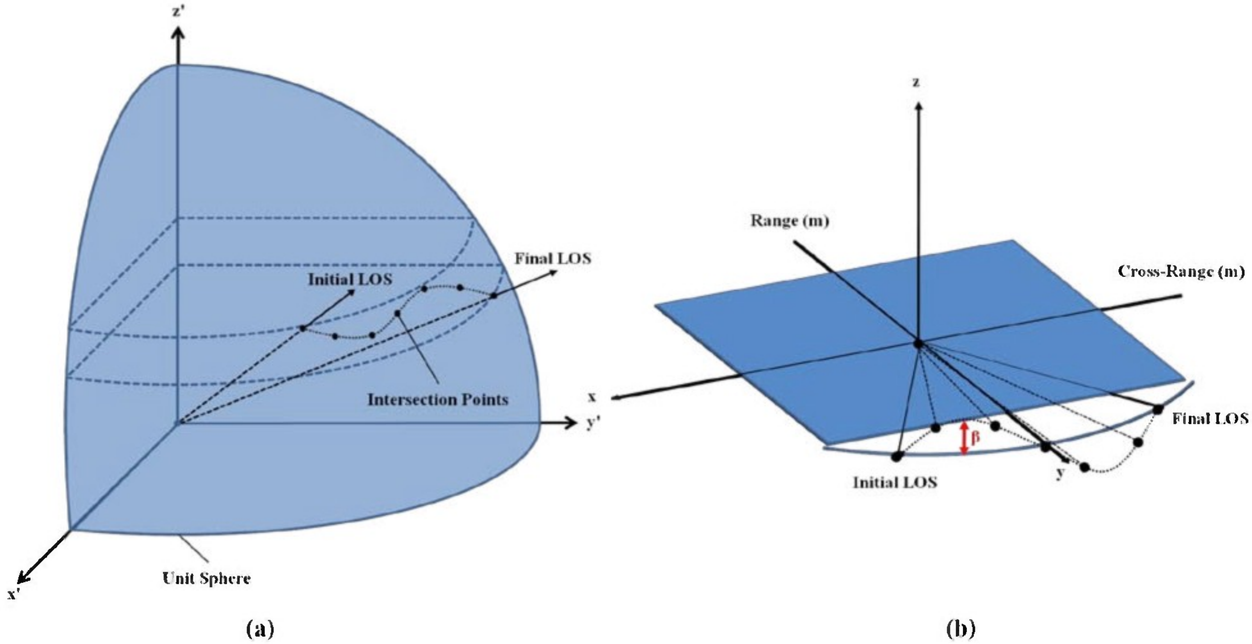


Figure 3.4 Image plane orientation (27 p. 13)

3.1.2 Radar Range-Doppler Imaging

The Fast Fourier Transform (FFT) Range-Doppler imaging algorithm produces two-dimensional images of a three-dimensional object. This is appropriate if the out-of-plane angle β remains sufficiently small, since the radar echoes then contain no information about the third orthogonal dimension. If the pierce points all lie in the xy plane $(\beta = 0)$, the distance from a point on the target to the radar is independent of the z-coordinate. When that happens, the radar is unable to resolve the target on the z-dimension. The resulting image can be thought of as a projection of the 3-D object into the image plane. The image value at each pixel is then the integral of the complex scattering amplitude and phase information of the target along the line through the pixel and perpendicular to the image plane (27 pp. 14 - 16).

To understand how range profiles observed at different aspect angles can be combined to form an image, it is important to remember that they constitute projections of the target onto a line parallel to the radar LOS. For a three-dimensional object, the points in the ideal range profile are integrals of the target's complex scattering amplitude and phase over planes perpendicular to the radar LOS. The observed range profile is then the convolution of the ideal range profile and the transmitted waveform. For two-dimensional imaging of a three-dimensional object, the target is resolved in the range and cross-range dimensions, but not the direction normal to the range/cross-range plane (28 p. 365). Again, the range direction is along the radar LOS at the center of the angular aperture. For an arbitrary target-rotation vector, cross range is normal to the plane defined by the LOS and the target-rotation vector. The unresolved third-dimension is therefore, normal to the LOS and in the plane defined by the line of sight and target rotation vector.

The idea of using Doppler frequency shift of the target echo for imaging purposes is credited to Carl Wiley (29 p. 14). Classical range-Doppler theory describes imaging in terms of the range and Doppler-frequency history of points in the image area over the target observation period. Though wideband radars give excellent information along the radar LOS, cross-range information is inherently poor. Fortunately, the motion visible through Doppler enables information in the cross-range direction to be extracted. In the range-Doppler technique, points on the target are sorted into appropriate (x, y) bins based on their observed trajectories during the imaging period. In the range direction, the sorting is based on the distance from the radar to the target scattering points. The resolution is limited by radar bandwidth, since it determines the accuracy with which the distance can be measured (30 pp. 79 - 80). In the cross-range direction, the sorting is based on the Doppler frequency history of the points over the imaging period. More precisely, the ideal Doppler frequency histories of points in the image plane are compared, often through cross-correlation, with observed amplitude histories, along appropriate trajectories, in the radar data (30 p. 84).

As the satellite rotates about its center of mass, each scatterer on the target has a slightly different velocity relative to the radar. On the scale of the wavelength, the different range-rates between adjacent scatter points give rise to a corresponding phase variation in the received signal, known as a Doppler frequency shift. This shift, in units of Hertz, is given by (29 p. 8)

$$f = \frac{\dot{\phi}}{2\pi} = \frac{2\dot{R}}{\lambda} \tag{3.1}$$

where $\dot{R}$ is the range rate, $\dot{\phi}$ is the rate of change in the phase of the received signal, and λ is the wavelength of the radar center frequency. It is important to note that in radar imaging, the sign for Doppler shift is opposite the usual convention, where a positive Doppler frequency corresponds to an approaching target and a negative Doppler frequency indicates a receding target (31 p. 551). The Doppler history is, therefore, generated as a target changes orientation with respect to the radar line of sight and is used to resolve targets into multiple Doppler bins. The received signal strength in each bin belongs at a particular distance (range) from the center of rotation of the object. The combination of received signal strengths in their resolved locations plus range resolution available from wideband waveforms gives a two-dimensional image of the target (29 pp. 8 - 9). Radar systems that employ continuous wave signals allow observation of the Doppler frequency shift directly. For pulse radar systems, however, the phase of the echo from the point target is a single sample of the phase function $\phi(t)$ at the time the pulse reaches the target. As expressed in equation (3.1), the Doppler frequency is then found from the time rate-of-change of the phase samples from the set of echoes collected during the imaging period. Since coherent pulsed radar systems measure phase directly, it is more convenient to describe the imaging procedure in terms of phase history rather than Doppler frequency history. The results, of course, can be written in terms of the Doppler frequency by differentiating the phase function (29 p. 10).

The phase history of a target can be described by considering a point in the image area, as depicted in Figure 3.5.

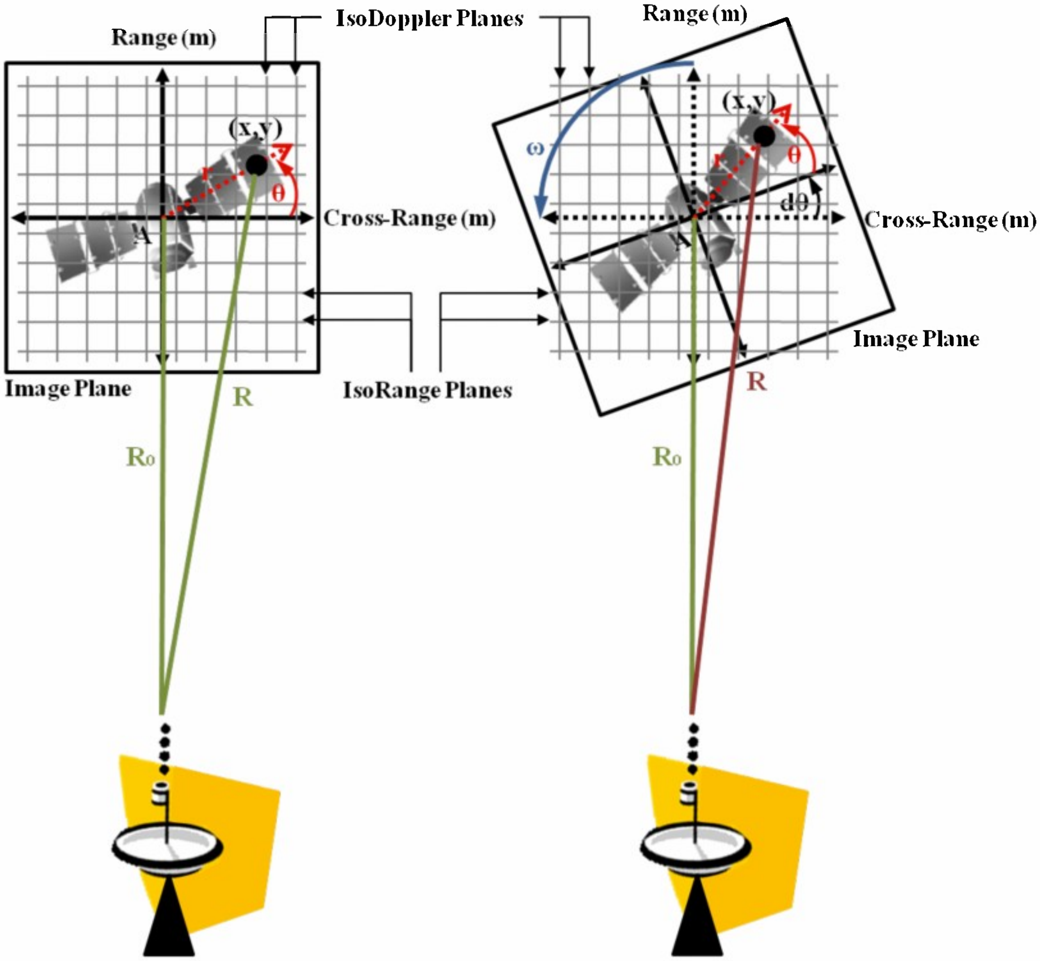


Figure 3.5 Distance to a scattering point in the image area. As the target rotates, the distance $\boldsymbol{R}$ from the radar to a point $(\boldsymbol{x}, \boldsymbol{y})$ on the target varies sinusoidally with $\boldsymbol{\theta}$. The amplitude and phase of that sinusoidal range variation encode the range and cross-range location of the point (28 p. 364).

The three-dimensional satellite has been projected into the $\boldsymbol{xy}$ plane, and is rotating with a uniform angular motion about the z-axis. A scatterer is offset from the origin A of a coordinate frame fixed to the target, by x in cross-range and y in range. The distance from the coherent radar to the point (x, y) is then (29 p. 10)

$$\mathbf{R}(t) = \mathbf{R}_0 + \mathbf{r}\sin(\theta) \qquad (3.2)$$

where $\mathbf{R}_0$ is the distance from the radar to the origin of the target-fixed coordinate frame, typically the center of mass, and $(\mathbf{r}, \theta)$ are the polar coordinates of the scattering point. As the target rotates about point A through the aspect angle $(\theta(t) = |\omega|t)$, the distance to the point becomes (29 p. 10)

$$\mathbf{R}(t) = \mathbf{R}_0 + \mathbf{r}\sin(\theta + d\theta) = \mathbf{R}_0 + \mathbf{r}_x \sin(|\omega|t) + \mathbf{r}_y \cos(|\omega|t) \tag{3.3}$$

where t is the observation time ($t = 0$ being the time at the center of the aperture), $\mathbf{r}_x$ and $\mathbf{r}_y$ are the cross-range and range relative to the center of rotation, and $|\omega|$ is the magnitude of the total angular velocity (composite of the orbit and rotation rate of the spacecraft) orthogonal to the RLOS. Equation (3.3) describes a unique trajectory for each point in the image area. Again, imaging involves matching point-target trajectories observed in the data to those described by equation (3.3). The resolution is determined by the minimum detectable difference in an observed trajectory. For coherent systems, the distance to the point target is determined from the radar signal phase, $\phi = 4\pi\mathbf{R}(t)/\lambda$. As a result, the accuracy of the trajectory measurement depends on the accuracy of the phase measurement (29 pp. 10 - 11).

Since the phase measurement is somewhat ambiguous for many ranges, the reference trajectory $\mathbf{R}_0$ is typically removed, enabling the relative Doppler frequency, range, and cross-range of the returned signal to be rewritten as (29 pp. 10 - 11)

$$f = \frac{2(\dot{\mathbf{R}} - \dot{\mathbf{R}}_0)}{\lambda} = \frac{2|\omega|\mathbf{r}_y}{\lambda}\cos(\omega t) - \frac{2|\omega|\mathbf{r}_x}{\lambda}\sin(|\omega|t) = \frac{2|\omega|\mathbf{r}\cos(|\omega|t)}{\lambda} \tag{3.4}$$

$$\mathbf{r}_y = \mathbf{R} - \mathbf{R}_0 = \mathbf{r}\sin(\omega t) \tag{3.5}$$

$$\mathbf{r}_x = \frac{f\lambda}{2\omega} = \mathbf{r}\cos(\omega t) \tag{3.6}$$

Accordingly, the amount of Doppler shift is a function of the scatterer's distance from the satellite's center of mass. By analyzing the return radar signal in terms of range delay and Doppler frequency, the range and cross-range components of the position of the point scatterers can be estimated. The surfaces of constant range are a set of planes parallel to the radar line of sight and one another, and the surfaces of constant Doppler are a set of planes parallel to the plane formed by the rotation axis and radar LOS (28 p. 370).

The presence of the target rotation rate ω in equation (3.6) implies that in order to obtain a properly scaled image of the object, the magnitude of ω must be known. If the target is assumed to rotate at a constant angular velocity, scatterers will approach or recede from the radar at a rate depending only on their cross-range position. The generation of cross-range dependent Doppler frequencies can be sorted via a Fourier transform. This operation is equivalent to the production of a large synthetic aperture phased array antenna formed by the coherent summation of the receiver outputs for varying target/antenna geometries (26 p. 32). For small angles, an ISAR image is therefore, the two-dimensional Fourier transform of the received signal as a function of Doppler frequency and target aspect angle. If however, the target rotates through large angles, the Doppler frequency history of a scatterer becomes nonlinear, following a sine-wave trajectory (described in Figure 3.4) that cannot be processed directly by a FFT without smearing the Doppler spectrum and, thus, degrading the cross-range resolution (28 p. 388). Removal of the defocusing effects due to rotational motion is accomplished by applying a phase correction to the synthetic aperture and increasing the pulse repetition frequency to obtain an unambiguous Doppler extent. Another implicit assumption, which further complicates the entire process, is that the reference distance $\mathbf{R}_0$ from the radar antenna to the center of the target is known and constant. In applications where $\mathbf{R}_0$ is a function of time, the effect of time-varying range must be removed from the received signal in the radar receiver and/or processor (28 p. 389).

3.1.2.1 Fast Fourier Transform Range-Doppler Imagery

Expanding upon the example from the previous section, suppose that the target being imaged is composed of two scattering centers connected by a rigid bar. As the target rotates, the observed coherent radar data, as illustrated by the middle left image of Figure 3.6b, is a complex combination of Doppler from multiple scatterers over time. These Doppler histories are arranged into an xy grid, such that: 1) along the y-axis are the wideband range profiles indicating the relative location and intensity of scattering centers along the radar line of sight and 2) along the x-axis are the Doppler profiles indicates the relative velocity of scattering centers toward and away from the radar (25 p. 20). Therefore, for each range cell in the resulting range-time-intensity (RTI) grid, the aligned profiles constitute a new complex series, containing amplitude A and phase ϕ information over time, according to Euler's equation (30 pp. 138 - 139)

$$e^{i\phi} = \cos(\phi) + i\sin(\phi) \tag{3.7}$$

$$Ae^{i\phi} \Rightarrow \begin{cases} I = A\cos(\phi) \\ Q = A\sin(\phi) \end{cases} \tag{3.8}$$

where I and Q are the in-phase and quadrature phasor of the signal, respectively, and are commonly referred to as the raw IQ-data.

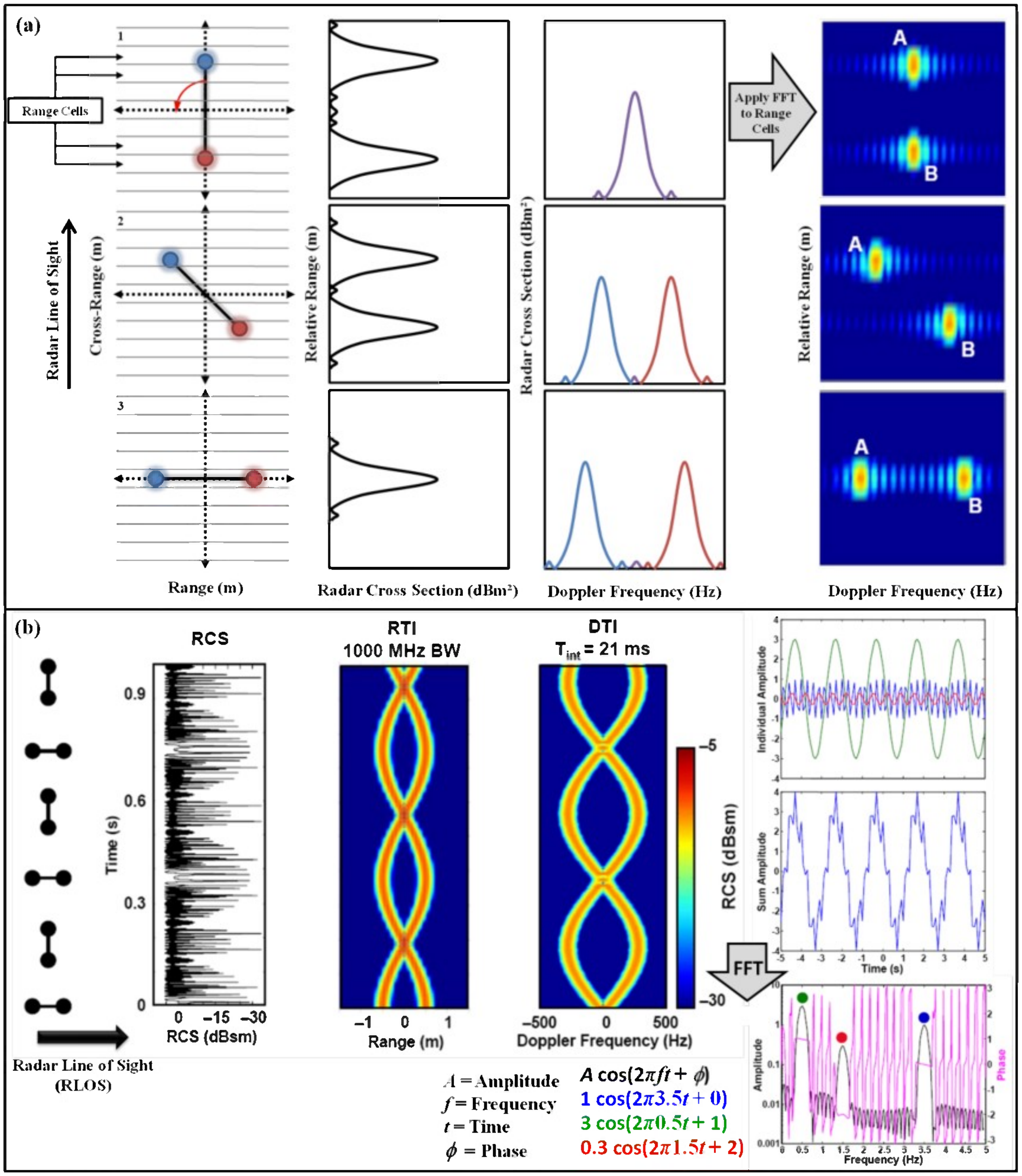


Figure 3.6 Range- and Doppler-Time-Intensity histories for a rotating dumbbell. The received radar cross sections are organized such that range information is along the y-axis, time is on the x-axis, and amplitude and complex phase information are contained in the z-direction.
(29 pp. 12 - 14)

The Doppler frequency shifts over time are then sorted by means of weighted fast Fourier transforms (FFT) which essentially act as collection of narrow bandpass filters. Since narrowband filters achieve their sensitivity by integrating the input signal over a period of time, the longer the integration time, the smaller the Doppler filter bandwidth and thus, greater the ability to resolve closely spaced frequencies in the cross-range direction (30 p. 127). This inverse relationship between the Fourier processing interval and Doppler resolution is the subject of the next section. The result of the FFT is depicted in the bottom right image of Figure 3.6b and clearly shows the separation of the received signal based on frequency. The weighting window that is applied before Fourier transformation, alters the resulting function by reducing sidelobes (secondary pulses of energy adjacent to the primary return), generally at the expense of widening the main lobe centered on the scatterer (30 p. 130). Combining the Doppler spectra for all range cells produces the desired two-dimensional range-Doppler images, as illustrated in the upper right-hand column of the figure above. Bright spots in the image correspond to energy scatterers on the satellite. Therefore, the radar image is essentially each scatterer's reflectivity mapped onto the range-Doppler plane, indicating the relative location along the radar LOS and the relative velocity of scattering centers toward and away from the ground-based sensor.

The final step in the image generation process utilizes equation (3.6), which provides the critical relationship needed to convert the range-Doppler information into a series of discrete-time range/cross-range images for a given pass. Knowledge of the motion of the object enables the transformation of Doppler (velocity) into cross-range (length). The presence of the angular velocity term in equation (3.6) also indicates that consistent scaling of the imagery is heavily dependent on a priori information about the rotational motion of the target relative to the radar line of sight. In addition to properly scaling the images in the cross-range direction, the magnitude of the angular velocity $|\omega|$ is critical to selecting a proper integration time and pulse repetition frequency (29 p. 21). This term, in effect, has the single greatest impact on the attitude estimation process, given the fact that an inaccurate cross-range extent can be easily misinterpreted as a change in satellite orientation when working with two-dimensional imagery. The mutual dependence of both the attitude measurements and image processing on the angular velocity estimate will become apparent in the subsequent sections.

3.1.2.2 Range and Cross-Range Resolution

Resolution in the range direction is achieved by conventional means using a series of short or long coded pulses which provide the size of a range resolution cell Δr_y, determined by the bandwidth B of the radar system according to the equation (28 p. 364)

$$\Delta r_y = \frac{c}{2B} \tag{3.9}$$

where c is the speed of light. Accordingly, wider bandwidths result in finer range resolution. From equation (3.6) it follows that cross-range resolution Δr_x is achieved by measuring Doppler frequencies with a resolution of (28 p. 364)

$$\Delta f = \frac{2\omega \Delta r_x}{\lambda} \tag{3.10}$$

Since a frequency resolution Δf requires a coherent processing time interval of approximately $\Delta T = 1/\Delta f$, the cross-range resolution is ultimately given by (28 p. 364)

$$\Delta r_x = \frac{\lambda}{2\omega \Delta T} = \frac{\lambda}{2\Delta\theta} \tag{3.11}$$

where $\Delta\theta = \omega\Delta T$ is the angle through which the object rotates during the coherent processing time. Fine cross-range resolution implies coherent processing over a large $\Delta\theta$; however, equations (3.3) and (3.4) indicate that both the range and Doppler frequency of a particular point scatterer can vary greatly over such a lengthy processing interval (31 p. 549). This means that during an imaging interval sufficiently long to give the desired cross-range resolution, points on the rotating target may move through several resolution cells. Hence, the usual range-delay measurement and Doppler frequency analysis implied by equations (3.5) and (3.6) will result in degraded imagery for the long integration time case (29 pp. 23 - 24). As shown in Figure 3.7, over or under estimating the Fourier integration time results in an apparent stretching or compressing effect in the resulting image.

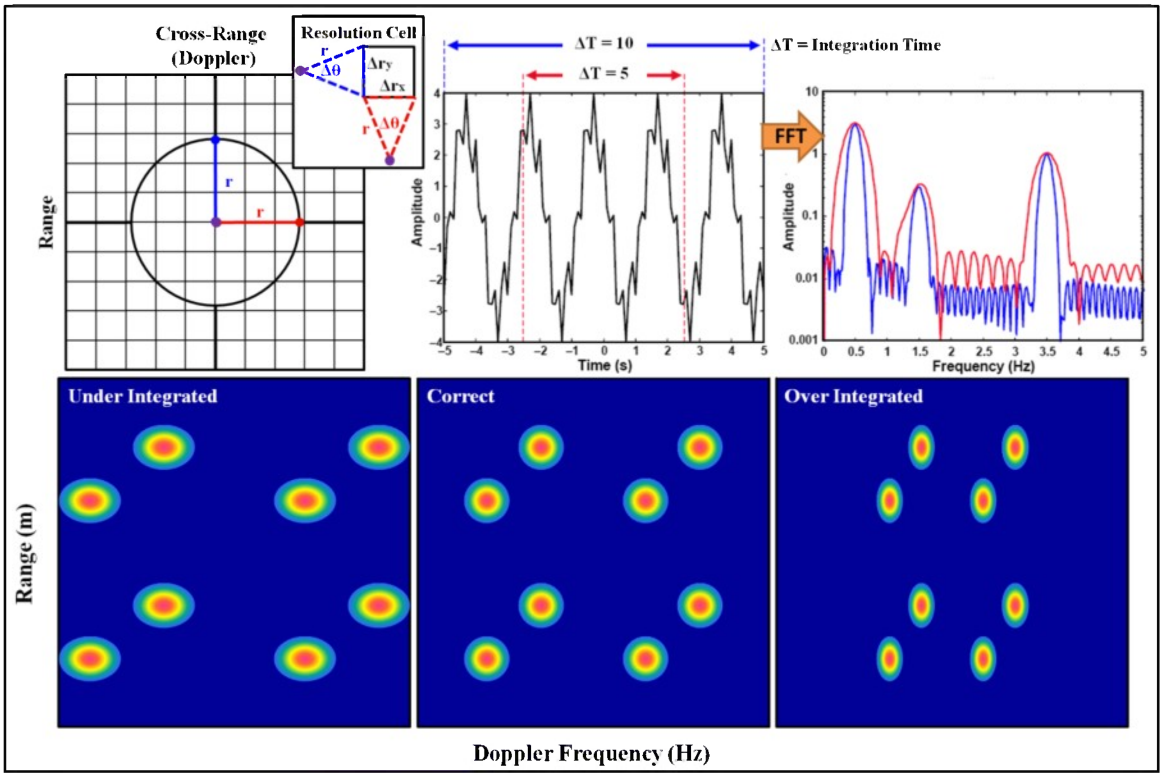


Figure 3.7 Impact of the imaging interval on Doppler resolution. Movement of the satellite during the FFT processing time span results in energy being distributed through multiple resolution cells, distorting the image (29 pp. 23 - 25)**.**

To avoid image degradation caused by motion through resolution cells while using the simple range-Doppler analysis described above, requires limiting the size of the coherent processing time ΔT. If one assumes a constant rotation rate and a radar LOS perpendicular to the axis of rotation, then there is no motion through a range resolution cell if $\Delta T < 2\Delta r_y/\omega d_x$. Similarly, a Doppler resolution cell will occur if $\Delta T < \left(\sqrt{\lambda/d_y}\right)/\omega$, where d_y and d_x in this set of equations, are the maximum range and cross-range dimensions of the object (27 p. 9). Consequently, one must limit the resolution of the imaging system such that (27 p. 10)

$$\Delta r_x^2 > \lambda d_y/4 \qquad (3.12)$$

$$\Delta r_x(\Delta r_y) > \frac{\lambda d_x}{4} \tag{3.13}$$

In general, the image scene dimensions are not the only parameters regulating the extent of the coherent processing interval and hence the cross-range resolution of conventional range-Doppler images. When the angular velocity is variable and/or the radar range directions are not coplanar in a coordinate system that rotates with the object, the constraint of no motion through a Doppler resolution cell may have to be modified to a more stringent one, leading to even smaller values of ΔT $(20\ \text{p.}\,365)$. Often, a finer cross-range resolution is desired, and hence points in the object move through range and/or Doppler resolution cells during the coherent integration time. In this case, simple frequency analysis will yield degraded imagery; the effect of motion through resolution cells must be compensated by analyzing the rotational motion the target. This of course is the aim of attitude estimation and hence the direction from which the problem will be addressed in the subsequent chapters.

3.1.2.3 Motion Compensation

The objective of radar processing is to estimate the target's reflective density function from received baseband signal samples, the so-called frequency signature. Ideally, the target's range is known precisely and the spacecraft’s orbital and angular motion are constant and known over the imaging pass. Under such ideal conditions, the extraneous phase term of the motion can be precisely removed, so that the reflective density function of the target can be obtained simply by taking the Fourier transform of the phase compensated frequency signatures. The process of estimating the target's range and removing the extraneous phase term is called focusing.

In the ISAR imaging formulation of Section 3.1.2, it is assumed that the target remains at a constant range, and that the only motion is rotation with respect to the radar. In reality, the range to the center of mass of the satellite is time dependant, changing as the target passes overhead. For satellite ISAR, the distance to the target may in fact change by several hundred kilometers during the data acquisition period. Errors in the estimate of the orbital flight track, or target trajectory, can significantly degrade cross-range resolution and thus the overall scaling of the radar image. Distance deviations due to over or underestimating the reference trajectory $\mathbf{R}_0$ that is removed from the phase measurements result in a variation in the range delay to each point target that is larger than the range sample spacing, causing range migration (26 p. 34). This smearing of the signal energy over several range bins must be corrected prior to imaging. The relative range to each image grid point must be known to within some fraction of a wavelength over the integration period being used in order to obtain fine cross-range resolution. The need for such high-precision estimates of the target trajectory is common to all operational ISAR systems and is accomplished using a combination of range alignment and phase correction, collectively known as motion compensation (26 pp. 34 - 35).

Removing the effects of translational motion from the radar signal is relatively straightforward. Initial estimates of the satellite trajectory can be obtained from published orbital information in the form of two line element sets, state parameters, and/or active tracking of the satellite. The state vector is then propagated forward in time using the numerical integration method discussed in Section 2.3.4.1. Furthermore, for each pulse return, we can extract measured range values based on a tracking algorithm. Fitting an orbit to the observed data is then done via a batch least-squares algorithm, the details of which are presented in Section 3.2.2. Subtracting the estimated trajectory from the measured trajectory produces a residual indicating the amount by which each range should be shifted in order to obtain the proper signal amplitude and phase data (25 pp. 15 - 17). The figure below summarizes this process.

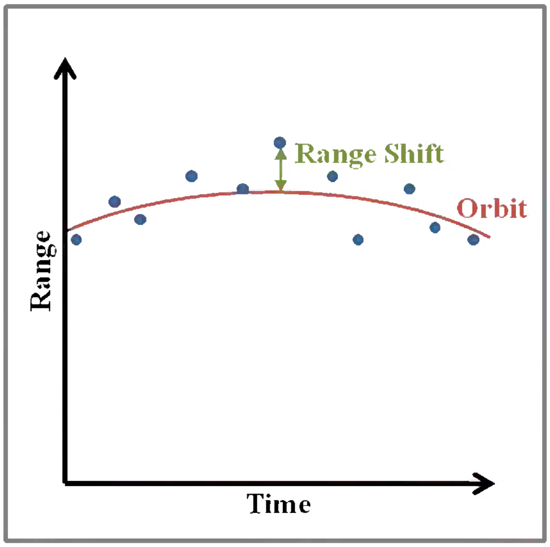


Figure 3.8 Range alignment and phase correction are performed by means of an orbit fitting process (25 p. 15)

If the radar echo is digitized directly, without down-conversion, motion compensation is accomplished by a simple time shift of the sampled waveform. In most radar systems, however, the signal is altered to minimize the required bandwidth of the recording system. Motion compensation then involves a time shift plus a phase shift to account for the phase change in the local oscillator (LO). The required phase shift $\Delta\phi$ is constant over individual pulses and is given by (25 p. 18)

$$\Delta\phi = 2\pi f \Delta t = \frac{4\pi f \Delta R}{c} \tag{3.14}$$

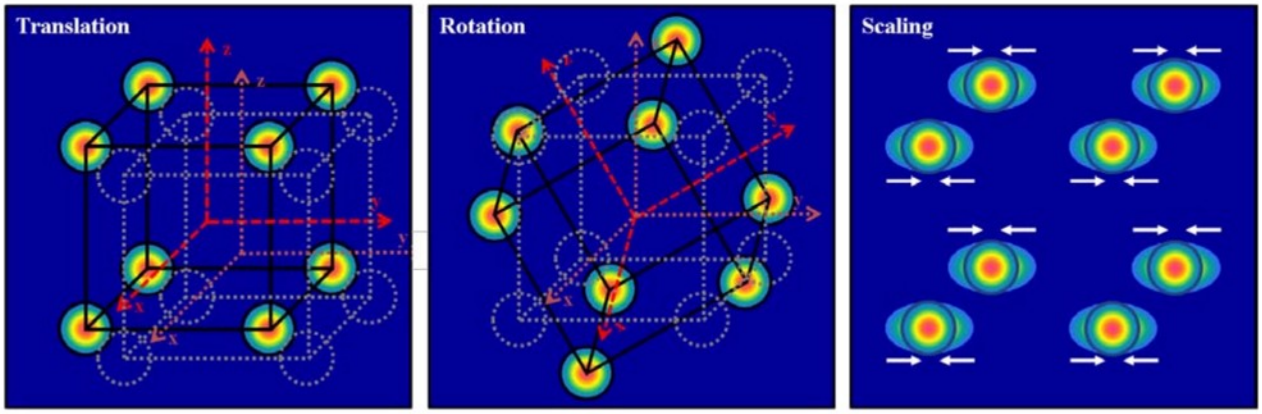
where ΔR is the range shift, $\Delta t = 2\Delta R/c$ is the corresponding time shift, and f is the LO frequency for homodyne receivers or the total frequency shift for heterodyne receivers. Removal of the bulk range variations – the variations in the distance from the phase center of the radar antenna to the origin of the target coordinate frame – does not change the imaging information content of the radar data. Since it is assumed that the wavefronts are planar, range variations do not affect the shape of the echo signal, only its strength and delay (25 pp. 18 - 19). The fact that the target is observed over a pass that is hundreds of kilometers long does not provide any direct information for imaging. Again, it is only the rotation of the target with respect to the radar that produces the differences between the individual radar echoes, and thus, the information needed for imaging. A value for the maximum tolerable error in the satellite range estimates can be obtained by observing that the imaging procedure relies on coherent combination of the individual echoes (correlation of range-derived phase information over some coherent aperture) (26 p. 27). For this reason, the pulse-to-pulse coherence starts to degrade when the error in the range to the target is some fraction of the shortest wavelength in the transmitted waveform. Any error in knowledge of relative position R_ϵ will give rise to a phase error given by (26 p. 28)

$$\phi_\epsilon = \frac{4\pi R_\epsilon}{\lambda} \tag{3.15}$$

Therefore, while it is not possible to set a universal threshold on motion determination accuracy, by placing limits upon image quality and allowable phase error, one can roughly determine the maximum tolerable error in the flight track estimate and thus, the precision that must be maintained over the entire angular aperture.

3.1.3 Image-Model Matching

A radar image shows the projection of a three-dimensional object onto a two-dimensional plane. In the common viewing format, the vertical axis of the image is range and the horizontal axis is either range rate (Doppler) or cross-range. One purpose of radar images is to allow the analyst to determine the motion of a target from measurements of the radar image. Often measurements take the form of parameters expressing the alignment between a computer-graphics model of the target satellite and the underlying radar imagery. In order to obtain measurements of the attitude, an analyst is able to manipulate the computer generated model with respect to an initial coordinate frame, in order to precisely match the two-dimensional projection of the model with the discrete-time radar images of the satellite. Thus, the image-model matching process attempts to align a wireframe model with fixed, known dimensions to the underling radar imagery through the use of three matching operations, as shown in Figure 3.9: 1) translation, produces horizontal and vertical displacements used to indicate a new image center; 2) rotation, orients the model in three-dimensional space over the image in order to obtain a proper alignment; and 3) scaling, alters the horizontal extent of the radar image (cross-range compression or stretching) (4 p. 280).


Translation

Rotation

Scaling

Figure 3.9 Alignment operations

Once a precise fit has been made for each discrete image in the pass, the set of Euler angles used to rotate the wireframe model from some initial coordinate frame to its perceived 'true' attitude can be recorded and passed to the filtering algorithm for processing. When viewed in sequence, the images provide information about the rotational motion of the satellite over time. This additional information is critical in ensuring that the attitude of the vehicle has been properly determined. For example, working solely with the computer generated wireframe models can easily result in an optical illusion, in which the analyst experiences a change in perceived attitude due to the presence of multiple interpretations in the two-dimensional projection of the wireframe model. This flip in perspective is illustrated in Figure 3.10, and is easily resolved by simply switching to a solid representation of the satellite from time to time in order to ensure that the alignment reflects the apparent orientation of the target. Additionally, while a lack of spacecraft symmetry complicates the overall dynamics of the problem, it does aid in the image model matching process, providing the distinguishing body features necessary to uniquely determine the attitude of the vehicle.

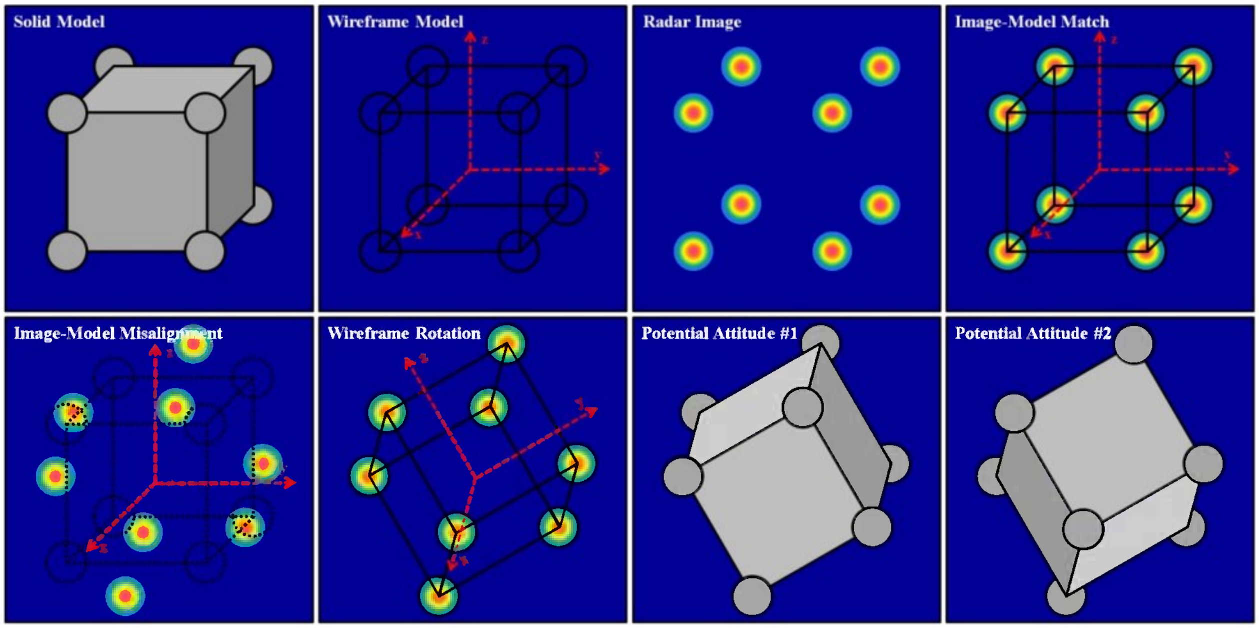


Figure 3.10 Image-model matching process for a set of point scatterers arranged in a simple three-dimensional shape.

3.1.3.1 Motion Ambiguity

In order to convert the range rate to length along the cross-range axis of the radar image, either one must perform certain measurements on the image or have knowledge of the satellite's total angular velocity. An incorrect estimate for $|\omega|$ causes both improper scaling (cross-range deformation) and poor resolution (cross-range defocusing) which may lead to misinterpretation during the alignment process (measurement noise). The mutual dependence of the range-Doppler imaging and image-model matching processes on a priori knowledge of the angular velocity, results in a circular problem in which both require the same input ($|\omega|$) from one another in order to produce their respective outputs. The attitude measurement process operates on the assumption that the images are consistently scaled, and thus provide an accurate representation of the spacecraft's orientation at a given discrete-time; conversely, in order to generate highly resolved, well-focused, and precisely scaled radar images requires detailed information about the translational and rotational motion of the target with respect to the radar line of sight. Consequently, incorrect scaling introduces a major source of error into the attitude measurements since a compression or stretching of the two-dimensional image in the cross-range direction is virtually indistinguishable from a rotation about the axis aligned with the radar line of sight.

The set of Euler angles that parameterize the image-model match can be easily converted into a rotation matrix $\mathbf{R}_{align}$ using the methods discussed in Section 2.1.3. Since the Doppler axis has units of speed, the analyst also measures a scalar s to rectify the axis and thereby produce cross-range coordinates. The quantity s simply scales the image parallel to the Doppler axis, as shown in Figure 3.9. A single radar image measures the range to and range rate of individual points on the imaged object. Distance and velocities orthogonal to the radar LOS are not measured and therefore are, without external information, unknown. A particular motion was assumed in making the images, and the time of the imaging interval T depends on the motion. Let t_{center} be the center of the imaging interval of duration T. The magnitude of the total angular velocity orthogonal to the radar LOS associated with the assumed, or nominal, motion at time t_{center} is $|\omega_{nom}^{x}| = |\hat{\mathbf{p}}_y \times \boldsymbol{\omega}_{total}|$ (24 p. 45). The radar image plane is the two-dimensional subspace on which the three-dimensional rigid body is projected. It is useful when forming equations to summarize the image plane as a matrix. Defining the vectors $\boldsymbol{\omega}_x = \hat{\mathbf{p}}_y \times \boldsymbol{\omega}_{total}$ and $\boldsymbol{\omega}_z = \boldsymbol{\omega}_y \times \boldsymbol{\omega}_x$ enables the image plane matrix to be defined by a basis of unit vectors in the cross-range, range, and the out-of- plane directions, such that: $\boldsymbol{P} = [\hat{\mathbf{p}}_x \quad \hat{\mathbf{p}}_y \quad \hat{\mathbf{p}}_z]$. It is worth emphasizing that $\boldsymbol{P}$ is a function of both the orbit and angular velocity of the rigid body relative to the sensor. Consequently, in the radar coordinate frame, the attitude of the satellite is $\boldsymbol{P}^T\boldsymbol{A}$ (32 p. 46). Because a single image provides a single component of the angular velocity we only know that $\boldsymbol{P}(:,2) = \widehat{\mathbf{p}}_y$ (Matlab notation for the second column of a matrix) and that $\boldsymbol{P}$ is orthonormal.

Often assuming an initial motion for an object is helpful in forming and understanding equations. As many objects may have a known nominal motion from which the target deviates occasionally, presuppose nominal attitude $\boldsymbol{A}_{nom}$ and $\boldsymbol{\omega}_{nom}$ is reasonable. Thus, the true image plane matrix is given by (32 p. 50):

$$\boldsymbol{P}_{true} = \boldsymbol{R}_y(\theta)\boldsymbol{P}_{nom} \tag{3.16}$$

the true image-plane matrix differs from the assumed one by a right-handed rotation of angle θ around the radar LOS.

Suppose the analyst has performed an alignment of the computer model to the image of the target, obtaining a matrix $\boldsymbol{R}_{align}$. The initial orientation of the model in the nominal image plane is $\boldsymbol{P}_{nom}^T\boldsymbol{A}_{nom}$; the analyst's measurements indicate that in the nominal image plane, the object has orientation $\boldsymbol{P}_{nom}^T\boldsymbol{A}_{nom}\boldsymbol{R}_{align}$. We do not have knowledge of $\boldsymbol{A}_{true}$, but we can write (32 p. 50)

$$\boldsymbol{P}_{true}^T\boldsymbol{A}_{true} = \boldsymbol{P}_{nom}^T\boldsymbol{A}_{nom}\boldsymbol{R}_{align} \tag{3.17}$$

Combining equations (3.16) and (3.17) yields the equation (32 p. 51)

$$\boldsymbol{A}_{true} = \boldsymbol{R}_y(\theta)\boldsymbol{A}_{nom}\boldsymbol{R}_{align} \tag{3.18}$$

Evidently, from a single image, one can determine the attitude of the satellite only up to a rotation around the radar line of sight. Intuitively, this fact is an immediate consequence of our ability to measure only range and range rate. A similar situation holds for $\boldsymbol{\omega}_{total} = \boldsymbol{\omega}_{orbit} + \boldsymbol{\omega}$, where the component of the total angular velocity parallel to RLOS is entirely unknown. The analyst, by scaling the image along the Doppler dimension, has measured $s = [|\omega_{nom}^z|/|\omega_{true}^z|]$, which rectifies the cross-range axis and measures the magnitude of the total angular velocity projected onto the plane orthonormal to the RLOS, $\boldsymbol{\omega_z}$, is known up to a rotation around the radar line of sight (32 p. 54):

$$\boldsymbol{\omega}_{true}^{z} = \frac{\boldsymbol{R}_y(\theta)\boldsymbol{\omega}_{nom}^{z}}{s} \qquad (3.19)$$

where θ has the same unknown value as before.

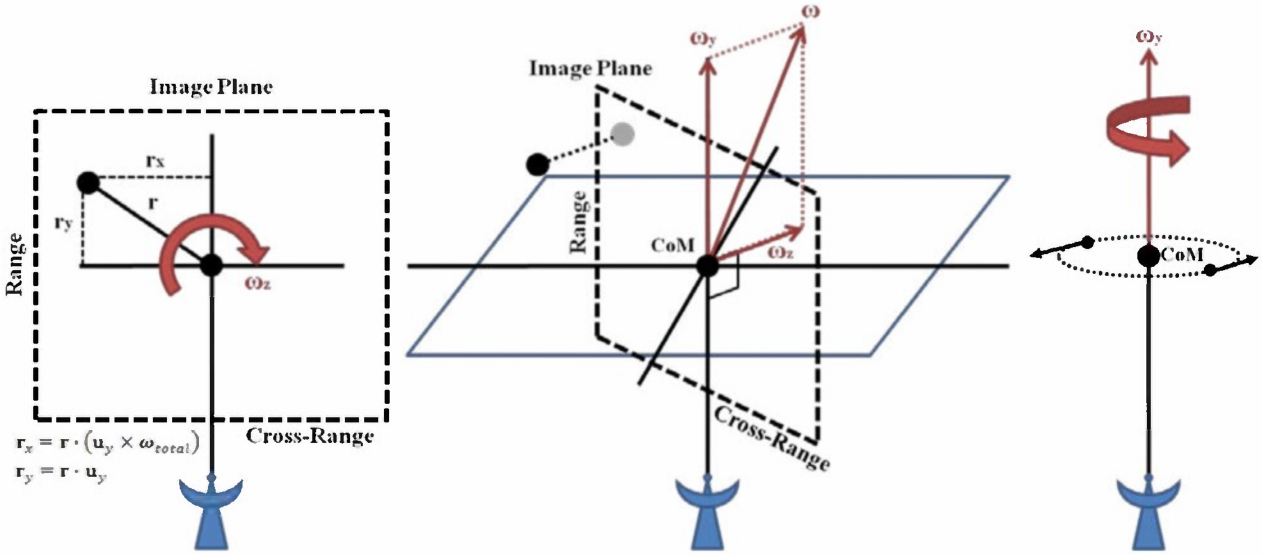


Figure 3.11 Observation of a target's orientation relative to the radar image plane. The magnitude of the component of the angular velocity perpendicular to range can be inferred, while, the range component remains unknown.

This ambiguity is inherent in all existing algorithms since no technique yet exists for making radar measurements that provide information beyond the RLOS ambiguity in each image. However, such measurements are not physically impossible. Returning to the conceptual framework developed in Section 3.1.1 for the geometry of the image plane, recall that if a unit sphere is fixed to the target's body coordinate system, as the object passes over the radar, the RLOS pierces the sphere at given latitude and longitude points. The set of observations over the duration of the pass can be summarized by a curve traced over the surface of the sphere, where each image represents a small piece of the curve and the image plane is an optimized fit to the

LOS vectors (26 pp. 18 - 19). What is of greatest importance here is that there is no reason for the image plane to be planar; more likely, it is gently curved. The geometry considered thus far ignores such curvature, and in effect rely on a set of equations which throw out independent measurements (i.e., the warping of the image plane). However, given enough bandwidth (range resolution), a high enough center frequency (Doppler frequency), a high enough pulse repetition frequency, and an accurate enough model, it is physically possible (though not necessarily practical) to measure the curve on the sphere (27 pp. 13 - 14). Knowledge of this nonlinearity in addition to initial conditions provides information about the true motion of the object, using measurements from a single image. Therefore, the ambiguity present in the current two-dimensional radar measurements is not essential and can be resolved in a more realistic manner by considering the motion revealed through making attitude measurements over a sequence images. Mathematically, meaningful assumptions yield additional independent equations to enable the calculation of unknown quantities that characterize the motion.

It is important to reiterate that this discussion concerns the limitation in the amount of information that can be derived from considering the motion of the target in a single discrete-time radar image. When looking at the pass as a whole, the change in aspect angle provides different perspectives of the satellite which can, if sampled densely enough by the radar signal, be coherently combined to form an unambiguous three-dimensional image. The potential also exists for using multiple synchronized radar systems (bistatic radar) to image a given target at the same time. If no particular baseline motion is assumed, full attitude determination by these methods is only possible, if the object is imaged simultaneously by two radars with a significant angle between the lines of sight. The benefits and challenges that arise from using two or more geographically separated radar systems in generating three-dimensional radar images is being investigated by the Aerospace Sensor Technology Group at Lincoln Laboratory. Additionally, research is currently being done in the Space Situational Awareness Group on automating the attitude measurement making process using a minimum bounding box technique (see References (33) and (34)). While both of these research efforts will likely have a considerable impact on the ISAR attitude sensing and measurement making process, they are for the time being beyond the scope of the problem being considered in this thesis. For more information on these advanced radar imaging and computer automation methods, readers are encouraged to consult References (28) and (35) respectively.

3.2 The Filtering Process

In deterministic attitude methods the same number observations as variables is used to obtain one or more discrete attitude solutions. In contrast, state estimation methods for attitude determination use the partial derivatives of the observables with respect to a collection of solved-for parameters (i.e. a state vector) to correct an a priori estimate of these same values. However, neither the number of solved-for attitude parameters nor the number of attitude observations used is important as far as the process itself is concerned. If the number of observations is less than the number of solved-for parameters, some combination of the unknowns will retain their a priori value, or, in some cases, an algebraic singularity will result. In general, the state vector and the various attitude estimates will be vectors in an n-dimensional phase space. The purpose of the data filter is to then calculate a state vector which is optimal by some statistical measure. Unfortunately, the optimal estimate is often difficult to achieve exactly due to the nonlinear characteristics of the attitude dynamics and measurement models involved. As result, nearly all the filters currently in use simply work around the problem by approximating the dynamics using linear equations [1]. Allowing for nonlinear equations would negate many of the basic statistical assumptions and techniques underpinning estimation theory, some of which include linear algebra and Gaussian normal distributions of errors (36 p. 437). A fundamental assumption allowing the use of linear techniques to solve estimation problems is that the linear least squared error (LLSE) estimator and the Bayesian least squared error (BLSE) estimator, also known as the minimum mean-square error (MMSE) estimator, are equivalent when process and measurement noise statistics are independent and identically distributed (iid) Gaussian distributions (37 pp. 190 - 191). As long as the dynamic and measurement equations transform the associated errors linearly, the LLSE and BLSE equivalence assumption holds.

A least-squares filter determines the state vector which minimizes the weighted square of the difference (error) between the observed data and the estimate computed from the observation model. The contribution of an individual observation in this process may be weighted according to the observation's expected accuracy and importance. Because least-squares filters provide the best estimate of the state parameters when the uncertainty is a result of Gaussian noise, they are by far the most common type of estimator and can be divided into two major classes: batch and sequential (36 p. 437). A batch estimator updates a state vector at a given reference time using a block of observations taken during a fixed time span. Conversely, in a sequential or recursive estimator, the state vector is updated after each observation is processed. Generally, sequential filters will be more sensitive to individual data points than will batch processors; that is, the sequential estimator may converge to a solution more quickly but be less stable than a batch algorithm (36 pp. 437 - 438).

[1] Reference (42) provides a survey of all the nonlinear attitude estimation methods currently in use. Among the many estimators covered, is the orthogonal attitude filter, which represents the first approach to a truly nonlinear filter.

Referring all the way back to Figure 3.1, the two major sequential estimators are the recursive least-squares estimator and various forms of the Kalman filter. Like a batch estimator, a recursive least-squares estimator corrects the state vector at a specified reference time. Since the confidence in the updated state at the reference time improves as more and more measurements are processed, the sensitivity of the filter to the observations diminishes over time. A recursive least-squares process allows the estimate to be updated at each measurement time, thereby avoiding the need to maintain an extensive backlog of previous measurements (2 p. 273). The basic Kalman filter goes a step further than the recursive least-squares algorithm by employing the statistical knowledge of the modeling process and the attitude measurements. A Kalman filter is a sequential estimator with a fading memory, meaning that it generally corrects the state vector at the time that each observation was made, rather than at a set epoch time. After the state is updated using one or more observations, it is propagated or extrapolated by a mathematical model to the time of the next observation to provide an initial estimate for the next update. The filter's confidence in its estimate of the state is allowed to degrade from one update to another, using models for the noise in the state vector. This causes the influence of earlier data on the current state to fade with time so that the filter does not lose sensitivity to current observations (2 pp. 274 - 275).

The major advantages of batch estimators is that they are simple to implement and are also generally less sensitive to bad data points than the more sophisticated algorithms which will be described. Another advantage of an algorithm of this type is that all observation residuals can be seen simultaneously, so that any obviously invalid observations, i.e., those with unusually large residuals, can be easily removed (36 p. 456). According to Reference (36), an observation is commonly removed if the absolute value of its residual is greater than three times the weighted root mean square residual. On the other hand, the computational burden imposed by a batch filter can be significant and depends on the number of state parameters, observations to be processed, and iterations required for convergence, as well as the overall complexity of the state and observation models employed by the filter (36 pp. 456 - 457). If a large number of iterations are required, a recursive estimator should be considered in order to reduce computer storage requirements and decrease execution time. Additionally, if the state undergoes minor unmodeled variation during the time spanned by the observations, the Kalman filter will generally track better than either the recursive or batch least-squares algorithms. On the other hand, such algorithms tend to be more sensitive to bad data, particularly at the beginning of a pass (2 p. 274). As will be discussed in Section 3.2.5, the backwards-smoothing extended Kalman filter is a very eclectic estimator, possessing attributes of both a batch least-squares algorithm and a Kalman filter. The consequence of such a union is an estimator that is able to incorporate statistical information about the modeling process and observations, while also being less reactive to invalid measurements which would otherwise cause the filter to misbehave.

The intent of the following sections is to provide the conceptual framework and relevant equations leading up to the central algorithm used in the Lincoln attitude estimation system (LAES), the backward-smoothing extended Kalman filter (BSEKF). Much can be gained by looking at the various attributes of its component parts as well as the emergent properties of the fully assembled algorithm. To that end, Section 3.2.1 provides the theoretical foundation for the algorithms presented in Sections 3.2.2 – 3.2.5. In addition to looking at the extended Kalman filter (EKF) and square-root information filter and smoother (SRIF&S), the batch least-squares filter is covered in considerable detail since it is used in both the BSEKF and the attitude estimation software, called LLMotion, currently being used at Lincoln Laboratory. The backward-smoothing extended Kalman filter is then thoroughly presented, with a focus on its specific attributes in relation to other estimators, operational characteristics, and specifics on how to implement the algorithm. In Chapter 4, the backward-smoothing EKF will be discussed in the context of the specific attitude estimation problem under investigation.

3.2.1 Foundational Concepts in Probability and Statistics

This section is meant to provide a brief description of the key equations and concepts in probability and statistics which form the foundation upon which the filters presented in latter sections have been developed. To that end, only those equations which are directly related to the development and understanding the maximum-likelihood (ML) and maximum a posteriori (MAP) estimation problems will be presented. For a more detailed discussion of the statistical properties and derivations of the algorithms presented in this chapter, see References (38) and (37).

3.2.1.1 Covariance Analysis

Covariance analysis is a general statistical procedure for studying the relationship between errors in the measurements and error in quantities derived from the measurements. The expected value or mean of a continuous random variable X is written as $E[X]$ or m_X, and is defined by (38 p. 219)

$$E[X] = \int x f_X(x) dx \tag{3.20}$$

where $f_X(x)$ is the probability density function, which represents the probability of X assuming a value somewhere in the interval $(x, x + dx)$; that is, (38 p. 220)

$$p(a \leq X \leq b) = \int_a^b f_X(x) dx = F_X(b) - F_X(a) \tag{3.21}$$

$F_X(x)$ is the cumulative distribution function (cdf), whose properties are treated in detail in Reference (38), along with other important probabilistic concepts and equations related to estimation. What is of greatest importance here, is that geometrically, m_X is one of a number of possible devices for locating the center or centroid of the probability distribution with respect to the origin. If one considers a second random variable Y, in which (38 p. 220)

$$Y = g(X) = (x - a)^n \tag{3.22}$$

then the expected value of the random variable can be expressed as (38 p. 220)

$$m_Y = E[g(X)] = \int g(x) f_X(x) dx = \int (x - a)^n f_X(x) dx \tag{3.23}$$

Notice that if $a = 0$ and $n = 1$ then the function g is the first moment of X about the origin is and is equivalent to equation (3.20). Similarly, the n^{th} moments about the mean are also very useful in describing X. The second central moment or variance σ_X^2 is defined by setting $a = m_X$ and $n = 2$, to produce (38 p. 221):

$$\sigma_X^2 = E[g(X)] = E[(X - m_x)^2] = E[X^2] - E[X]^2 = \int (x_i - m_x)^2 f_X(x) dx \tag{3.24}$$

The variance of the probability distribution is simply the average square deviation of X from the mean value and the positive root-mean-square (rms) deviation or standard deviation σ_X, defined by $\sigma_X = \sqrt{\sigma_X^2}$, is a measure of the dispersion of the distribution about m_X (1 p. 429). The magnitude of the variance of an estimate/measurement provides a sense of how much noise is in the estimate/measurement. In physical terms, if one interprets the density function as a mass distribution, the first moment about the origin becomes the center of mass and the variance is the moment of inertia about an axis through the mean (1 p. 430). For example, consider the following density functions:

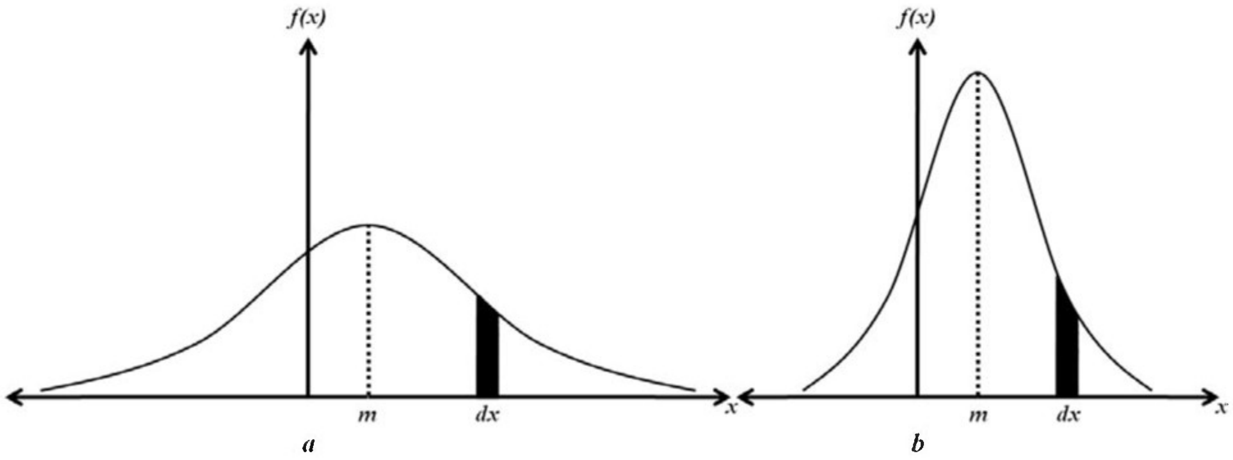


Figure 3.12 Two probability density functions with the same mean but different variances
(38 p. 238)

Both density functions $\boldsymbol{a}$ and $\boldsymbol{b}$ have their mean value indicated by $\boldsymbol{m}$. Clearly, the variance of $\boldsymbol{b}$ is much larger than the variance of $\boldsymbol{a}$, since $(\boldsymbol{x}-\boldsymbol{m})^2$ generally will be greater than that of $\boldsymbol{a}$ for an increment of area $\boldsymbol{dx}$ $(\mathbf{28\ p.237})$.

All the information that can be known about the random variable $\boldsymbol{X}$ is contained in the probability density function (pdf). In estimation applications, one is most concerned with the first two moments, namely the mean and variance. Since the pdf will change with time, the prediction of the future values for the state dynamic system can be obtained by propagating the joint density function $\boldsymbol{f_{X,Y}(x,y)}$ forward in time and using it to calculate the mean and variance (1 pp. 433 - 434). While, the equations for propagating these values forward are the subject of the subsequent sections the specific equations need for calculating joint, marginal, and conditional probabilities are given in Appendix B.

If the principal of expected value is extended to a continuous random vector composed of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, the initial and central moments are defined by (38 pp. 220 - 221)

$$m_{X,Y}=E[X,Y]=\int\int xyf_{X,Y}(x,y)dxdy=\Sigma_{X,Y}+m_X m_Y \qquad (3.25)$$

$$\Sigma_{X,Y} = E[(x - m_X)(y - m_Y)] = \int\int (x - m_X)(y - m_Y) f_{X,Y}(x, y)dxdy = \begin{bmatrix} \sigma_x^2 & \Sigma_{X,Y} \\ \Sigma_{X,Y} & \sigma_x^2 \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_x^2 & \rho_{X,Y}\sigma_X\sigma_Y \\ \rho_{X,Y}\sigma_X\sigma_Y & \sigma_x^2 \end{bmatrix} \tag{3.26}$$

where $m_{X,Y}$ is the mean of the vector and $\Sigma_{X,Y}$ is the covariance matrix – a measure of the interdependence or linear correlation between the two variables. The correlation coefficient $\rho_{X,Y}$ of X and Y is the normalized covariance (38 p. 222)

$$\rho_{X,Y} = E\left[\frac{(x - m_X)}{\sigma_X}\frac{(y - m_Y)}{\sigma_Y}\right] = \frac{\Sigma_{X,Y}}{\sigma_X\sigma_Y} \tag{3.27}$$

which satisfies the inequality $\rho_{X,Y} \leq 1$. For independent variables, $\Sigma_{X,Y} = \rho_{X,Y} = 0$, and for completely correlated variables $\rho_{X,Y} = 1$. Covariance analysis relates the presumably know variance and covariance in one set of variables (e.g., measurement errors) to a second set of variables (e.g., computed attitude errors) (38 p. 223). In what follows, it is assumed that the n computed quantities x_i are functions of the m measurements y_i with $m \geq n$. Thus, $x_i = x_i(y_1, y_2, \dots y_m)$ or in vector notation, $\mathbf{x} = \mathbf{x}(\mathbf{y})$. The link between the known variance and correlation in the measurements and these same metrics in the desired computed quantities is given by the linear transformation (1 pp. 432 - 433)

$$\mathbf{P} = \mathbf{H}\mathbf{R}\mathbf{H}^T \tag{3.28}$$

where $\mathbf{H}$ is the $n \times m$ matrix of partial derivatives with the elements $H_{ij} = \partial x_i/\partial y_j$, $\mathbf{R}$ is the $m \times m$ measurement error covariance matrix, and $\mathbf{P}$ is the symmetric $n \times n$ state error covariance matrix. Equation (3.28) relates the variance (diagonal terms) and covariance (off-diagonal terms) in the measurements and computed quantities without implying anything further about the distributions of the errors in either $\mathbf{x}$ or $\mathbf{y}$. However, two specific assumptions are often used in attitude analysis:

1. If the distribution of errors in **y** is Gaussian or normal, then the distribution of errors in **x** is also Gaussian. A continuous random process X that is normally distributed with mean m and variance σ^2, $[X \sim N(m_X, \sigma_X^2)]$, possesses a probability density function which is given by (38 p. 161)

   $$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2} \tag{3.29}$$

   for $-\infty < x < \infty$.

2. This special probability distribution is used extensively in modeling random systems since it can be shown that any random variable made up of a sum of m independent and identically distributed (iid) random components tends towards a normal distribution as $m \to \infty$. This property is formally known as the Central Limit Theorem and is a fundamental assumption in the development of statistical estimation algorithms. If there are a large number of uncorrelated measurements, then the Central Limit Theorem can be used to infer the distribution of errors in **x**, irrespective of the form of the distribution of the measurement errors. The theorem may also be used to compute the variance and distribution of errors in a measurement which is contaminated by many errors sources with presumably known variances (38 pp. 239 - 240).

3.2.1.2 Minimum Variance Estimation

As has already been mentioned, the weighted least-squares approach does not include any information about the statistical characteristics of the measurement errors or the a priori errors in the values of the parameters to be estimated. The minimum variance approach is one method for removing this limitation and is used widely in developing solutions to estimation problems because of the simplicity of its use (37 p. 188). It has the advantage that the complete statistical description of the random errors in the problem is not required; rather, only the first and second moments of the probability density function of the observation errors are needed. This information is expressed in the mean and covariance matrix associated with the random error. If it is assumed that the observation error $\boldsymbol{\epsilon}$ is random with zero mean and specified covariance, then the goal of the state estimation problem is to find the best linear, unbiased, minimum variance estimate $\hat{\mathbf{x}}_k$ of the true state $\mathbf{x}_k$ (38 pp. 501 - 502). The consequences of formulating the problem in this manner are outlined as follows:

1. Linear: The requirement of a linear estimate implies that the estimate is to be made up of a linear combination of the observations, that is (37 p. 184)

$$\hat{\mathbf{x}}_k = \mathbf{My} \tag{3.30}$$

The $n \times m$ matrix $\mathbf{M}$ is unspecified and is to be selected to obtain the best estimate. An in depth derivation of the value of $\mathbf{M}$ that satisfies the requirements listed in the problem statement can be found in Reference (37) – for the sake of brevity, the consequence of that effort yields the following equation $\mathbf{M} = \mathbf{PH}^T\mathbf{R}^{-1}$. This result, which when combined with equation (3.34) for the minimum variance (provided in step 3) and substituted back into equation (3.30), produces the following estimate of $\mathbf{x}_k$ (37 p. 186):

$$\hat{\mathbf{x}}_k = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} \tag{3.31}$$

where $\mathbf{H}$ is the Jacobian matrix of partial derivative of the state with respect to the observations and $\mathbf{R}$ is the measurement error covariance matrix. This solution will agree with the weighted batch least-squares solution provided in Section 3.2.2, if the weighting matrix $\mathbf{W}$, used in the least-squares approach is equal to the inverse of the observation noise covariance matrix; that is, if $\mathbf{W} = \mathbf{R}^{-1}$ (37 p. 190).

2. Unbiased: If the estimate is unbiased, then by definition (38 p. 503):

$$m_e = E[\mathbf{e}] = E[\hat{\mathbf{x}}_k] - E[\mathbf{x}_k] = 0 \tag{3.32}$$

where $\mathbf{e}$ is the prediction error or the difference between the true and estimated state $\mathbf{x}_k - \hat{\mathbf{x}}_k$.

3. Minimum Variance: If the estimate is unbiased, then the estimation error covariance matrix will be diagonal with values (1 p. 432)

$$\sigma_e^2 = E[(\mathbf{e} - m_e)^2] = E[\mathbf{e}\mathbf{e}^T] \tag{3.33}$$

Centering the mean value of the error on zero and then minimizing the variance-covariance matrix, ensures that the error probability is focused as tightly as possible around the mean value of zero. This process is depicted in Figure 3.2, where the dashed blue density function represents the best unbiased minimum variance estimate.

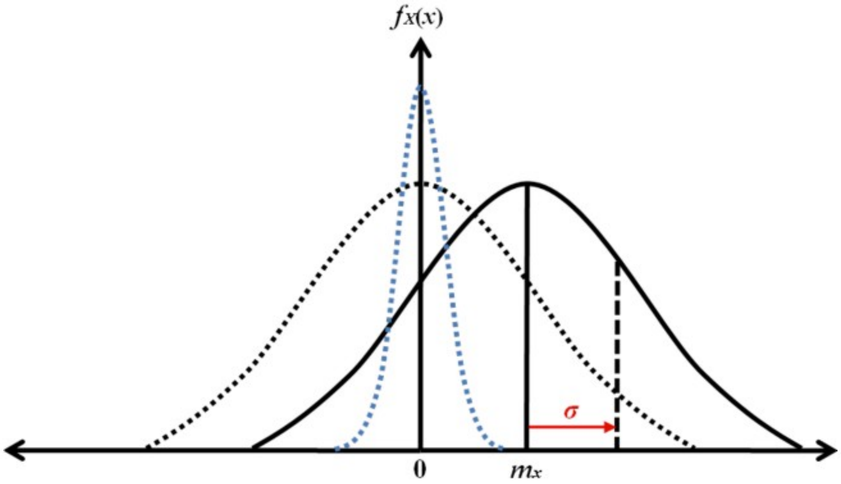


Figure 3.13 Normal or Gaussian error probability distribution function with mean m and standard deviation σ (28 p. 503)

The expression for the minimum state estimation covariance matrix is given by

$$\mathbf{P}_k = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} \tag{3.34}$$

and is the direct result of the fact that $E[\epsilon\epsilon^T] = \mathbf{R}$ and $\mathbf{P}_k = \mathbf{MRM}^T$. The complete mathematical explanation can again be found in Reference (37).

3.2.1.3 Maximum Likelihood and Bayesian Estimation

The method of maximum likelihood estimation (MLE) for determining the best estimate of a variable is accredited to Ronald Fisher and is a nonrandom approach which tries to estimate an unknown constant. The MLE of a parameter X – given observations $y_1, y_2, \dots, y_k$ and the joint probability density function (39 p. 1445)

$$f_{X,Y}(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k; \mathbf{x}_k) \tag{3.35}$$

is defined to be the value of X that maximizes the probability density function. However, if X is a random variable and one has knowledge of its pdf, the MLE of X is defined to be the value of X, which maximizes the probability density function of X conditioned on information provided by the observations $y_1, y_2, \dots, y_k$ (39 p. 1446):

$$f_X(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) \tag{3.36}$$

The Bayes estimate for X is defined to be the mean of the conditional density function given by the equation above. The estimator based on the Bayesian approach is known as the maximum a posteriori (MAP) estimator. The joint density function, equation (3.35), and the conditional density function are referred to as the likelihood function L. The logic behind maximizing L is that of all the possible values of X one should choose the value that maximizes the probability of obtaining the measurements that were actually observed (39 p. 1448). If X is a random variable, this corresponds to the mode, or peak, of the conditional density function. In the case of a symmetric, unimodal, density function such as a Gaussian function, this will be equivalent to the mean of the conditional density function. Hence, the MLE and the MAP estimate for a Gaussian density function are identical (39 pp. 1448 - 1449). In fact, the maximum likelihood approach can be thought of a degenerate case of the maximum a posteriori estimator, since as the variance of the normal a priori pdf approaches infinity, the density function will approach that of uniform distribution. In this case, the MAP estimate and ML estimate coincide, given that if a uniform prior distribution is assumed over the parameters, the maximum likelihood estimate coincides with the most probable values thereof (37 p. 191).

Posing the estimation problem in the MAP format is a useful starting point in the development of statistically based estimation criteria. If one defines X to be the state vector and Y to be the observation vector, then $f_X(x|Y = y)$ is the a posteriori density function, $f_Y(y|X = x)$ is the a priori density function, and Bayes theorem is given by (38 p. 81)

$$f_X(x|Y = y) = \frac{f_{X,Y}(x,y)}{f_Y(y)} = \frac{f_Y(y|X = x)f_X(x)}{f_Y(y)} \tag{3.37}$$

$$f_Y(y|X = x) = \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{f_X(x|Y = y)f_Y(y)}{f_X(x)} \tag{3.38}$$

From a Bayesian viewpoint, the goal is to develop a filter to propagate as a function of time the probability density functions of the desired quantities conditioned on knowledge of the actual data coming from the measurement devices (37 pp. 191 - 192). Once such a conditional density function is projected, the optimal estimate can be defined. Possible choices for the optimal estimate include the mean, median, and mode. By generating the pdf, some judgment can be made as to which criterion defines the most reasonable estimate for a given attitude problem. However, by assuming that the density function is normal, the mean, median, and mode will be identical. Hence, any criterion that chooses one of these values for $\hat{\mathbf{x}}_k$ will yield the same estimator (37 p. 194).

Another nonrandom estimator is the least squares estimator (LSE) which will be describe in more detail in the next section. If one is given scalar and nonlinear measurements (36 p. 449):

$$y_i = h_i(\mathbf{x}_i, t_i) + v_i \text{ for } j = 1, 2, \dots, k \tag{3.39}$$

then the least-squares estimate of the vector **x** is obtained by minimizing the argument (36 p. 448):

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{k} [y_i - h_i(\mathbf{x}_i, t_i)]^2 \tag{3.40}$$

Again, if the measurement errors are Gaussian, i.e., $v_i \sim N(0, \sigma^2)$, then the least squares estimator coincides with the maximum likelihood estimator (MLE) described previously. For random parameters, the counterpart of the LS estimator is the minimum mean square error (MMSE) estimator which finds the value of the state estimate that minimizes the expectation or the square of the error (minimum variance) in the estimate according to the equation (39 p. 1450):

$$\hat{\mathbf{x}}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^2 | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k] = E[(\mathbf{e})^2 | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k] \tag{3.41}$$

The solution to this estimator is the conditional mean of $\mathbf{x}$ (27 p. 194):

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k] = \int \mathbf{x}_k f_X(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) dx$$

$$= \left[\mathbf{P}_{\mathrm{k}}^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k\right]^{-1} \left[\mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{y}_{\mathrm{k}} - \mathbf{h}_k) + \mathbf{P}_0^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k)\right] \tag{3.42}$$

Because the mean and mode of a normal distribution are equal, the MMSE and MAP estimates are equal when given a Gaussian a posteriori pdf. In fact, reference (37) shows that due to the Gaussian statistics assumed for measurement and process noise, the ML, MAP and minimum variance estimators are all equivalent. Accordingly, the estimation techniques described in the subsequent sections all make use of the Gaussian independent and identically distributed error assumption, but also operate with the crucial flaw that the approximate linear dynamic and measurement models differ from the actual nonlinear physics occurring. The extended Kalman filter (EKF), for example, works well for systems with a small degree of nonlinearity. However, for systems with higher degrees of nonlinearity, smoothing and other methods are needed to improve the accuracy and convergence reliability of the state estimate.

3.2.2 Batch-Least Squares Filter

The Gauss-Newton least-squares procedure, formulated independently by Karl Gauss and Adrien Legendre, begins by considering the m-component state vector $\mathbf{x}$, which is allowed to vary with time according to the dynamics function (37 p. 176)

$$\mathbf{x}(t) = f(\mathbf{x}^0, t) \tag{3.43}$$

where $\mathbf{x}^0$ is the state vector at the reference time t_0 and $\hat{\mathbf{x}}^0$ is the estimate of this initial state vector, calculated using the batch least-squares algorithm. Since $\mathbf{x}$ contains parameters whose time variation is not insignificant, propagation of the attitude state via direct integration (the R-stage Runge-Kutta method is described in Section 2.3.4.1), is required. If the state undergoes a minor unmodeled variation during the time spanned by the observations, a batch estimator will calculate a weighted average value for $\hat{\mathbf{x}}^0$ $(37\ \text{p}.176)$. In such instances, a Kalman filter allows better tracking of the state variations than does a batch technique.

Consider a set of n observations,

$$\mathbf{y} = [y_1, \dots, y_n]^T \tag{3.44}$$

taken during the timespan of interest. In order to determine the state vector $\mathbf{x}$, it is assumed that $\mathbf{y}$ equals the observation model vector $\mathbf{h} = h(\mathbf{x}, t)$ based on the mathematical model of the observations plus additive random measurement noise $\mathbf{v}$. Thus, for each element of $\mathbf{y}$, (37 p. 177)

$$y_i = h_i(\mathbf{x}(t_i), t_i) + v_i \tag{3.45}$$

Equation (3.45) can be used to estimate $\mathbf{x}^0$, given an a priori estimate $\hat{\mathbf{x}}_0^0$, the observations $\mathbf{y}$, the functional forms of $f(\mathbf{x}^0, t)$ and $h(\mathbf{x}, t)$, and the statistical properties of $\mathbf{v}$. To accomplish this, the least-squares criterion is employed as a measure of good fit; the best value of $\mathbf{x}_0$ minimizes the weighted sum of the squares of the residuals between the elements of the observation and observation model vectors. This is done quantitatively by minimizing the loss function given by (37 p. 177)

$$J = \frac{1}{2}\Delta\mathbf{y}^{\,T}\mathbf{W}\Delta\mathbf{y} \tag{3.46}$$

where the observation residual vector $\Delta\mathbf{y}$ is defined by

$$\Delta\mathbf{y} = \mathbf{y} - \mathbf{h} \tag{3.47}$$

$\mathbf{W}$ is an $n \times n$ symmetric, nonnegative definite matrix chosen to weight the relative contributions of each observation, according to its expected accuracy or importance. In the simplest case, $\mathbf{W}$ is the identity matrix indicating that equal weight is given to all observations. The general form of $\mathbf{W}$ can be expressed as (37 p. 177)

$$\mathbf{W} = \mathrm{diag}([\sigma_1^2 \quad \cdots \quad \sigma_n^2]) = \begin{bmatrix} \sigma_{y1}^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_{y2}^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & \sigma_{yn}^{-2} \end{bmatrix} \tag{3.48}$$

where σ_{yi} for $i = 1, 2, \ldots, n$ is the uncertainty in the i^{th} observation $\mathbf{y}$.

An important variation of the loss function given by equation (3.46) penalizes any deviation from the a priori estimate in proportion to the inverse of the uncertainty in that estimate; that is (37 p. 178)

$$J = \frac{1}{2}\left(\Delta\mathbf{y}^{T}\mathbf{W}\Delta\mathbf{y} + [\mathbf{x}^0 - \hat{\mathbf{x}}_0^0]^T\mathbf{S}_0[\mathbf{x}^0 - \hat{\mathbf{x}}_0^0]\right) \tag{3.49}$$

where $\mathbf{P}_0$ is the $m \times m$ state weight matrix. If the elements of $\mathbf{P}_0$ are zero, no weight is assigned to the a priori state estimate, and equation (3.49) is equivalent to equation (3.46). Commonly, $\mathbf{P}_0$ has the form (37 p. 178)

$$\mathbf{S}_0 = \mathrm{diag}([\sigma_1^2 \quad \cdots \quad \sigma_n^2]) = \begin{bmatrix} \sigma_{x1}^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_{x2}^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & \sigma_{xn}^{-2} \end{bmatrix} \tag{3.50}$$

where σ_{xi} for $i = 1, 2, \ldots, n$ is the uncertainty in the a priori estimate $\hat{\mathbf{x}}_0^0$. The use of $\mathbf{S}_0$ is especially valuable when the lack of observability is a problem, which occurs when a change in one or more state parameters causes little changes in the observations (i.e., when the observations do not contain enough information to completely specify the state).

For J to be a minimum with respect to $\mathbf{x}_0$, $\partial J/\partial\mathbf{x}_0$ must be zero. Therefore, the value of $\mathbf{x}_0$ which minimizes J is a root of the equation (37 p. 198)

$$\frac{\partial J}{\partial \mathbf{x}_0} = -\Delta \mathbf{y}^T \mathbf{W} \mathbf{H} + [\mathbf{x}^0 - \hat{\mathbf{x}}_0^0]^T \mathbf{S}_0 = [\mathbf{0}]^T \tag{3.51}$$

where **0** is a vector of zeros and **H** is the $n \times m$ measurement transition matrix (37 p. 198)

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}^0} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1^0} & \cdots & \frac{\partial h_1}{\partial x_m^0} \\ \vdots & & \vdots \\ \frac{\partial h_n}{\partial x_1^0} & \cdots & \frac{\partial h_n}{\partial x_m^0} \end{bmatrix} \tag{3.52}$$

Values for $\partial h_i / \partial \mathbf{x}$ are normally computed analytically from the observation model. Values for $\partial h_i / \partial \mathbf{x}^0$ are then calculated from (40 p. 42)

$$\frac{\partial h_i}{\partial \mathbf{x}^0} = \frac{\partial h_i}{\partial \mathbf{x}}(t_i) \mathbf{\Phi}(t_i, t_0) \tag{3.53}$$

where $\mathbf{\Phi}(t_i, t_0)$ is the $m \times m$ state transition matrix consisting of the partial derivatives of the state at t_i with respect to the state at the reference time t_0; that is, (40 p. 42)

$$\mathbf{\Phi}(t_i, t_0) = \frac{\partial \mathbf{x}}{\partial \mathbf{x}^0}(t_i) = \begin{bmatrix} \frac{\partial x_1(t_i)}{\partial x_1^0} & \cdots & \frac{\partial x_1(t_i)}{\partial x_m^0} \\ \vdots & & \vdots \\ \frac{\partial x_m(t_i)}{\partial x_1^0} & \cdots & \frac{\partial x_m(t_i)}{\partial x_m^0} \end{bmatrix} \tag{3.54}$$

The values of $\mathbf{\Phi}$ may be calculated either numerically or analytically, depending on the functional form of $f(\mathbf{x}^0, t)$.

The most common method of solving equation (13.51) is to linearize the observation model vector $h(\mathbf{x}, t)$ about a reference vector $\tilde{\mathbf{x}}_0^0$ and expand each element of **h** in a Taylor Series of $\tilde{\mathbf{x}}_0^0$. It is important to note that $\tilde{\mathbf{x}}_0^0$ may be different from the a priori estimate $\hat{\mathbf{x}}_0^0$. Truncating the higher order terms yields (40 p. 43)

$$h_i = h_i(\tilde{\mathbf{x}}_0^0) + \frac{\partial h_i}{\partial \mathbf{x}^0}(\tilde{\mathbf{x}}_0^0)[\mathbf{x}^0 - \tilde{\mathbf{x}}_0^0] \tag{3.55}$$

for each element of **h.** Expressing the above equation in vector form gives (40 p. 43)

$$\mathbf{h} = \tilde{\mathbf{h}} + \widetilde{\mathbf{H}} \mathbf{x}^0 - \widetilde{\mathbf{H}} \tilde{\mathbf{x}}_0^0 \tag{3.56}$$

if the same reference vector is used for each element of $\mathbf{h}$. The ~ above each variable signifies evaluation at $\mathbf{x}^0 = \tilde{\mathbf{x}}_0^0$. Substituting equation (3.56) into equation (3.51) and solving for $\mathbf{x}^0$ (the result is designated using the notation for an estimate $\hat{\mathbf{x}}^0$) yields (37 p. 194)

$$\hat{\mathbf{x}}^0 = \tilde{\mathbf{x}}_0^0 + \left[\mathbf{S}_0 + \tilde{\mathbf{H}}^T\mathbf{W}\tilde{\mathbf{H}}\right]^{-1}\left[\tilde{\mathbf{H}}^T\mathbf{W}\left(\mathbf{y} - \tilde{\mathbf{h}}\right) + \mathbf{S}_0(\hat{\mathbf{x}}_0^0 - \tilde{\mathbf{x}}_0^0)\right] \qquad (3.57)$$

If $\tilde{\mathbf{x}}_0^0 = \hat{\mathbf{x}}_0^0$ and if $\mathbf{h}$ is a linear function in $\mathbf{x}^0$, then this equation will provide the best estimate for the true initial state vector $\mathbf{x}_0$. However, if $\mathbf{h}$ is nonlinear, $\mathbf{x}^0$ will not be corrected exactly by equation (3.57) unless the a priori estimate $\hat{\mathbf{x}}_0^0$ is already close to the optimum value. Consequently, if the correction determined using equation (3.57) is not small, then an iterative process is usually necessary (37 p. 195). In this case, $\mathbf{h}$ is linearized about the a priori estimate, which is then corrected to become $\hat{\mathbf{x}}_1^0$, as follows (37 p. 195):

$$\hat{\mathbf{x}}_1^0 = \hat{\mathbf{x}}_0^0 + [\mathbf{P}_0^{-1} + \mathbf{H}^T\mathbf{W}\mathbf{H}]^{-1}[\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{h})] \qquad (3.58)$$

The corrected value $\hat{\mathbf{x}}_1^0$, then replaces $\hat{\mathbf{x}}_0^0$ as the reference for the linearization of $h(\mathbf{x}, t)$ in the next iteration. The $(k + 1)$ estimate for $\mathbf{x}^0$ is derived using the equation (37 p. 195)

$$\hat{\mathbf{x}}_{k+1}^0 = \hat{\mathbf{x}}_k^0 + \left[\mathbf{S}_0 + \mathbf{H}_k^T\mathbf{W}\mathbf{H}_k\right]^{-1}\left[\mathbf{H}_k^T\mathbf{W}(\mathbf{y}_{\mathrm{k}} - \mathbf{h}_k) + \mathbf{S}_0\left(\hat{\mathbf{x}}_0^0 - \hat{\mathbf{x}}_k^0\right)\right] \qquad (3.59)$$

This iterative process continues until the differential correction (i.e., the difference between $\hat{\mathbf{x}}_{k+1}^0$ and $\hat{\mathbf{x}}_k^0$) approaches zero and/or until the loss function no longer decreases. At this point, $\hat{\mathbf{x}}_{k+1}^0$ has converged to its optimum value. Finally, for the converged solution the $m \times m$ error covariance matrix is given by (36 p. 452)

$$\mathbf{P}_k = [\mathbf{S}_0 + \mathbf{H}^T\mathbf{W}\mathbf{H}]^{-1} = \mathrm{E}[\mathbf{e}\mathbf{e}^T] \qquad (3.60)$$

Assuming that $E[\mathbf{e}] = 0$, where the estimation error vector $\mathbf{e} = \mathbf{x}^0 - \hat{\mathbf{x}}^0$, and E denotes the expected value.

Another useful quantity is the weighted root-mean-square (rms) residual, which can be found using (36 p. 453)

$$\Delta y_{rms} = \sqrt{\frac{\sum_{i=1}^{n} W_i (y_i - h_i)^2}{\sum_{i=1}^{n} W_i}} \tag{3.61}$$

where W_i is the i^{th} diagonal element of the measurement weight matrix and the units of Δy_{rms} are the same as for the y_i terms. Because Δy_{rms} is normalized according to the sum of the observation weights, it is frequently more useful than the loss function as a relative measure of the degree to which the solution fits the observed data. However, this parameter alone is insufficient for detecting the two major causes of a poor fit – unmodeled biases and a high level of noise in the observations. Insight into the contributions of these two sources of error can be gained by writing Δy_{rms} in the form (36 p. 453)

$$\Delta y_{rms}^2 = m_{\Delta y}^2 + \sigma_{\Delta y}^2 \tag{3.62}$$

where $m_{\Delta y}$ is the weighted mean of the residuals (36 p. 453)

$$m_{\Delta y} = \frac{\sum W_i \Delta y_i}{\sum W_i} \tag{3.63}$$

and $\sigma_{\Delta y}$ is the weighted rms deviation (standard deviation) of the residuals, (36 p. 453)

$$\sigma_{\Delta y} = \sqrt{\frac{\sum W_i \left(\Delta y_i - m_{\Delta y}\right)^2}{\sum W_i}} \tag{3.64}$$

The mean of the residuals should be near zero, because Δy_i can be either positive or negative. A large value for $m_{\Delta y}$, therefore, indicates the unmodeled biases are probably present in the observations and a large value for $\sigma_{\Delta y}$ signifies that the observation noise is large.

3.2.2.1 Guarded Gauss-Newton Method

Gauss-Newton differential correction procedure outlined in the previous section may be unsuitable for some nonlinear problems because convergence cannot be guaranteed unless the a priori estimate is close to a minimum in the loss function. Furthermore, its rate of convergence can be difficult to control (40 p. 48). An alternative approach to solving the batch least-squares problem which guarantees convergence is the gradient search method, in which the state parameters are adjusted so that the resultant direction of travel in the state space is along the negative gradient of the cost function J (i.e., in the direction of steepest decent). Although this method initially converges rapidly, it slows down when the solution approaches the vicinity of the minimum (40 p. 50). Thus, the guarded Gauss-Newton method was developed in order to overcome both the difficulties of the standard Gauss-Newton technique when an accurate initial estimate is not available and the slow convergence problems of the gradient search approach when the solution is close to the loss function minimum. The algorithm uses an expression of the form (41 p. 888)

$$\hat{\mathbf{x}}_{k+1}^{0} = \hat{\mathbf{x}}_{k}^{0} + \left[\mathbf{H}_{k}^{T}\mathbf{W}\mathbf{H}_{k} + \gamma\mathbf{1}_{m\times m}\right]^{-1}\left[\mathbf{H}_{k}^{T}\mathbf{W}(\mathbf{y} - \mathbf{h}_{k})\right] \tag{3.65}$$

where $\mathbf{1}_{m\times m}$ is an identity matrix and γ is a proportionality constant. If γ is small, equation (3.65) is equivalent to the Gauss-Newton procedure and if γ is large, $\mathbf{x}^0$ is corrected in the direction of the negative gradient of J, but with a magnitude which decreases as γ increases. To implement the guarded Gauss-Newton method for improved convergence requires the following simple procedure (41 p. 888):

1. Compute the loss function using the a priori state estimate $\hat{\mathbf{x}}_0^0$
2. Apply the first state correction to the state vector to form $\hat{\mathbf{x}}_1^0$ using equation (3.65) with $\gamma \gg \mathbf{H}^T\mathbf{W}\mathbf{H}$
3. Recalculate the cost function at $\hat{\mathbf{x}}^0 = \hat{\mathbf{x}}_1^0$. If $J(\hat{\mathbf{x}}_1^0) \geq J(\hat{\mathbf{x}}_0^0)$, then $\hat{\mathbf{x}}_1^0$ is discarded and γ is replaced by γc, where c is a fixed positive constant. The state estimate $\hat{\mathbf{x}}_1^0$ is then recomputed using the new value of γ in equation (3.65). If $J(\hat{\mathbf{x}}_1^0) \leq J(\hat{\mathbf{x}}_0^0)$, then $\hat{\mathbf{x}}_1^0$ is retained, but γ is replaced by γ/c.

4. After each subsequent iteration, compare $J(\hat{\mathbf{x}}_{k+1}^{0})$ and replace γ with either γc or γ / c as indicated in step 3. The state estimate $\hat{\mathbf{x}}_{k+1}^{0}$ is retained if the cost continues to decrease and discarded if J increases.

This procedure continues until the difference in the cost function J between two consecutive iterations is small, or until γ reaches a small predetermined cutoff value $(\gamma < \varepsilon)$. The details of how to implement the guarded Gauss-Newton method in the backward-smoothing Extended Kalman filter (BSEKF) will be provided in Section 3.2.5.1.

3.2.3 Extended Kalman Filter

To estimate the value of a state vector at an arbitrary time t_k, the state estimate at t_0, from a batch or recursive algorithm, must be propagated from t_0 to t_k using a model of the system dynamics. The continuous-discrete Kalman filter formulated by Rudolf Kalman in the 1960's, on the other hand, estimates the m-component state vector $\hat{\mathbf{x}}(t_k)$ directly based on all the observations up to and including $\mathbf{y}_k$ and the dynamics model evaluated between observations (40 p. 257). Although all filters require a dynamics model to propagate the state estimate between observations, the accuracy requirements for this model are normally less severe for the Kalman filter than for batch or recursive estimators because propagation is not preformed at one time over the entire block of data. In addition, the Kalman filter compensates for dynamics model inaccuracy by incorporating a noise term which gives the filter a fading memory – that is, each observation has a gradually diminishing effect on the future state estimates (36 p. 468).

By statistically combining attitude measurements with various noise characteristics, along with the model and process noise, the basic Kalman filter arrives at the statistically optimal estimate for a linear system. Unfortunately, the attitude problem being considered utilizes process and observation models which are both nonlinear. The extended Kalman filter (EKF) solves this problem by linearizing the estimation around the current mean and covariance, using the partial derivatives of the process and measurements functions to compute new state estimates even in the presence of nonlinear relationships (36 pp. 462 - 463). The reference attitude for the EKF is updated after each observation to reflect the best estimate of the true attitude. Consequently, the EKF generally converges on the best estimate more rapidly than the standard Kalman filter since errors introduced in the linearization process are reduced by relinearizing the differential equations for the reference attitude are after each observation is processed. (37 p. 209) Convergence however, is by no means guaranteed. Unlike its linear counterpart, the extended Kalman filter is not an optimal estimator. If the initial estimate of the state is wrong, or if the process is modeled incorrectly, the filter may quickly diverge, due to its linearization. Another problem with the EKF is that the estimated covariance matrix tends to underestimate the true covariance and therefore risks becoming inconsistent in the statistical sense without the addition of "stabilizing noise" (36 p. 463) With that said, the extended Kalman filter does perform reasonably well in most applications and is the most prolific and well established filter in the field.

Each time a set of observations $\mathbf{y}_k$ is obtained, the algorithm uses it to update the a priori state vector estimate at t_k, denoted by $\hat{\mathbf{x}}_{k-1}(t_k)$, to produce an a posteriori estimate $\hat{\mathbf{x}}_k(t_k)$. The EKF also, converts the a priori error covariance matrix estimate $\mathbf{P}_{k-1}(t_k)$ into the a posteriori estimate $\mathbf{P}_k(t_k)$. These a posteriori estimates are then propagated to t_{k+1} to become the a priori $\hat{\mathbf{x}}_k(t_{k+1})$ and $\mathbf{P}_k(t_{k+1})$ for the next observation set $\mathbf{y}_{k+1}(t_{k+1})$. The subscript k indicates the current estimate, which is based on all the observations up to and including the observations in $\mathbf{y}_k$ (40 p. 257).

Since the system dynamics is nonlinear, the state propagation is commonly performed by integrating an equation of the form (40 p. 257)

$$\frac{d}{dt}\hat{\mathbf{x}} = f(\mathbf{x}, t, \mathbf{w}) \Rightarrow \dot{\mathbf{x}} = f(\mathbf{x}, t) \tag{3.66}$$

with the initial condition $\mathbf{x}(t_k) = \hat{\mathbf{x}}(t_k)$, where f is a function which is nonlinear in $\mathbf{x}$, and $\mathbf{w}$ is the process noise vector which is assumed to be normally distributed with zero mean and covariance $\mathbf{Q} \ldots [p(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k)]$. Similarly, the measurement model relating $\mathbf{x}$ to $\mathbf{y}$, is given by the function $h(\mathbf{x}, t, \mathbf{v}) \Rightarrow h(\mathbf{x}, t)$, where $\mathbf{v}$ is the zero-mean measurement noise vector with covariance $\mathbf{R} \ldots [p(\mathbf{v}_k) \sim N(0, \mathbf{R}_k)]$ (40 p. 258).

The algorithm for computing the extended sequential estimate can be summarized as follows:

Time Update (Predict)

Given an initial state estimate $\hat{\mathbf{x}}(t_k)$, state covariance matrix $\mathbf{P}(t_k)$, and observation $\mathbf{y}_k(t_k)$

1. Propagate the state vector and state transition matrix from t_{k-1} to t_k via equation (3.66) with the noise term omitted. If the higher order terms are neglected, under the assumption that they are small compared to the first, the linearized dynamics function can be expressed as (37 pp. 209 - 210):

   $$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{A}(t)\hat{\mathbf{x}} \tag{3.67}$$

   where $\mathbf{A}$ is the $m \times m$ state Jacobian matrix composed of partial derivatives of f with respect to $\mathbf{x}$ $[\mathbf{A}(t) = \partial f / \partial \mathbf{x}]$. Since $\mathbf{A}$ is time varying, the state transition matrix $\mathbf{\Phi}$ is obtained by integrating the following matrix equation either analytically or numerically as discussed in Section 2.3.4.1 (37 p. 210):

   $$\dot{\mathbf{\Phi}} = \frac{d}{dt}\mathbf{\Phi}(t, t_k) = \mathbf{A}(t)\mathbf{\Phi}(t, t_k) \tag{3.68}$$

   with the initial condition $\mathbf{\Phi}(t_k, t_k) = \boldsymbol{I}_{m \times m}$. Thus, the solution to equation (3.68) can be written as (37 p. 210):

   $$\hat{\mathbf{x}}_k(t_{k+1}) = \mathbf{\Phi}(t_{k+1}, t_k)\hat{\mathbf{x}}_k(t_k) \tag{3.69}$$

2. Project the state error covariance forwards in time using

$$\mathbf{P}_k(t_{k+1}) = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{P}_k(t_k)\mathbf{\Phi}(t_{k+1}, t_k)^T + \mathbf{Q}(t_{k+1}, t_k) \tag{3.70}$$

Measurement Update (Correct)

3. Calculate the predicted observation vector $\mathbf{h}_k$, measurement residual vector $\Delta\mathbf{y} = \mathbf{y}_k - \mathbf{h}_k$, and measurement Jacobian matrix $\mathbf{H}_k$ which consists of the partial derivatives of h with respect to $\mathbf{x}$ [$\mathbf{H}_k = \partial h/\partial\mathbf{x}$].

4. Compute the $m \times l$ Kalman gain matrix (l being some subset of the total number of observations, denoted by n) (37 p. 210):

$$\mathbf{K}_k = \mathbf{P}_{k-1}(t_k)\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k-1}(t_k)\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{3.71}$$

5. Update the state estimate with the latest measurement $\mathbf{y}_k$ (37 p. 210)

$$\hat{\mathbf{x}}_k(t_k) = \hat{\mathbf{x}}_{k-1}(t_k) + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}_k) \tag{3.72}$$

In some cases it is necessary to iterate the estimate of $\hat{\mathbf{x}}_k(t_k)$ to reduce the effects of nonlinearities in the observation model. If this occurs, then $\mathbf{h}_k$ and $\mathbf{H}_k$ will be evaluated about a reference vector $\tilde{\mathbf{x}}_k(t_k)$, which may be different from $\hat{\mathbf{x}}_{k-1}(t_k)$. The above equation is then replaced by the more general form (36 p. 463)

$$\hat{\mathbf{x}}_k(t_k) = \tilde{\mathbf{x}}_k(t_k) + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}_k + \mathbf{H}_k[\tilde{\mathbf{x}}_k(t_k) - \hat{\mathbf{x}}_{k-1}(t_k)]) \tag{3.73}$$

Iteration may then be done using equation (3.73) with $\tilde{\mathbf{x}}_k(t_k) = \hat{\mathbf{x}}_{k-1}(t_k)$ to estimate $\hat{\mathbf{x}}_k(t_k)$. The operation is cyclically repeated using $\tilde{\mathbf{x}}_k(t_k) = \hat{\mathbf{x}}_k(t_k)$ and so on, until the change in the state estimate $\hat{\mathbf{x}}_k(t)$ is negligible. This modification forms the basis of the iterated extended Kalman filter (IEKF). In most cases the time between observations is small and local iteration is not needed.

6. Update the $m \times m$ state error covariance matrix (36 p. 463)

$$\mathbf{P}_k(t_k) = (\boldsymbol{I}_{m\times m} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k-1}(t_k) \tag{3.74}$$

- Note that in these equations, the l-vector $\mathbf{h}_k$ and the $l \times m$ matrix $\mathbf{H}_k$ are equivalent at $\hat{\mathbf{x}}_k(t_k)$.

7. Select the next observation at t_{k+1} and return to step 1.
8. The extended Kalman filter has achieved a steady state performance when the corrections to the state vector reach a consistent level and when the error covariance matrix is stable.

The EKF will yield a new state estimate at each observation, which is of value when a real-time solution is desired as the filter processes data. By adding the corrections to the state at each observation, the effects of the nonlinearities in the equations of motion are not as severe, since the attitude parameters are being updated with each measurement. Also, the partials of the system dynamics function are recomputed at each time step given the updated state. This allows for a more accurate state transition matrix (40 p. 560). Conversely, it is important to remember that a fundamental flaw of the EKF is that the probability densities functions of the various random variables are no longer normal after undergoing their respective nonlinear transformations. The EKF is simply an ad hoc state estimator that only approximates the optimality of the conditional probability equation of Bayes Theorem through linearization (37 pp. 209 - 210).

3.2.3.1 Kalman Filter Divergence

Divergence occurs when the estimated state moves away from the true state. This is the most common problem associated with Kalman filters. The most frequent causes of divergence are linearization errors, cumulative round-off and truncation errors, modeling noise, and unknown noise statistics (40 p. 467). Linearization problems can be reduced using local iteration or more frequent selection of observations. Round-foff and truncation errors may be partially solved by using: 1) a square-root filter, which substitutes the square root of the error covariance matrix for its full value in the filter gain equation and 2) regularly enforcing the unit norm constraint on the attitude quaternion (37 p. 331). Adjusting the state noise covariance matrix using the residuals between actual and computed observations is commonly used to reduce the effects of modeling errors. While problems associated with unknown noise statistics may be solved after extensive testing with both simulated and real data, proper filter response will only result when the appropriate balance between the state noise and measurement noise covariance matrices is found. This process is typically called tuning the filter (42 p. 13). A data rejection scheme which removes all observations whose uncertainties are not accurately known is also necessary to prevent divergence. If, for example, state noise has been underestimated with respect to observation noise, the state estimation procedure will become less and less sensitive to the observation residuals. Divergence could then result even though the filter may have reached steady state. Alternatively, if observation noise has been underestimated, the state estimation procedure may be incorrectly influence by the observation errors (42 p. 14).

3.2.4 Nonlinear Smoothing Filters

When dealing with systems that have highly nonlinear system dynamics and when observations can be post processed – i.e. real-time state estimates are not needed – smoothing is a way to compute more accurate state estimates than the Kalman Filter can alone. Unlike the other estimators discussed thus far, which only use past measurements to estimate **x**, smoothers incorporate information from both past and future measurements when estimating each state. References (37) and (43) classify smoothing problems into three categories: fixed-interval smoothing, fixed-point smoothing, and fixed-lag smoothing. Fixed-interval smoothing keeps the time interval of the measurements fixed and then searches for the optimal state estimate within the given timespan. Information from both past and future measurements is applied to compute optimal state estimates for these interior points. Fixed-point smoothing is used to find state estimates for a single point in time. The measurements occurring after the time of interest are subsequently used to improve the estimate at that point. An example of this would be the estimation of initial conditions based on later observations of a spacecraft's orientation. Fixed-lag smoothing is used to seek estimates of a state which is a fixed number of time points behind the current measurement time (43 pp. 1 - 3). Because the backward-smoothing extended Kalman filter described in Section 3.2.5 incorporates fixed-interval smoother in its algorithm, this type of smoothing will be the focus of this section.

It is often desirable to perform a smoothing operation when using a sequential filter. In this case, one is searching for the best estimate of the state at some time t_k based on all observations through time t_l where $l > k$. While the batch estimation approach has difficulty including the effects of process noise, smoothing algorithms have been developed to overcome this limitation, using a Bayesian approach of maximizing the density function of the state conditioned on knowledge of the observations through time t_l (33 p. 2). In this section the notation $\hat{\mathbf{x}}_k^l$ is used to indicate the best estimate of $\mathbf{x}$ at t_k based on observations through t_l. In accordance with the maximum likelihood approach, the aim is to find a recursive expression for $\hat{\mathbf{x}}_k^l$ in terms of $\hat{\mathbf{x}}_{k+1}^l$, which maximizes the conditional probability density function (37 p. 246)

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_1, \dots, \mathbf{y}_k, \dots, \mathbf{y}_l) = p(\mathbf{x}_{k+1} | \mathbf{x}_k)\, p(\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_k) \tag{3.75}$$

Accordingly, the problem can be solved by a least-squares technique that determines the vector sequences $\mathbf{x}_k$, $\mathbf{w}_k$, and $\mathbf{v}_k$ that satisfy the following linearized constraints (37 p. 246)

$$\mathbf{x}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{x}_k + \mathbf{\Gamma}(t_{k+1}, t_k)\mathbf{w}_k \tag{3.76}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{3.77}$$

and maximize the equation (37 p. 247)

$$\ln(L) = -\frac{1}{2}[\mathbf{x}_{k+1} - \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{x}_k]^T\, [\mathbf{\Gamma}(t_{k+1}, t_k)\mathbf{Q}_k\mathbf{\Gamma}(t_{k+1}, t_k)^T]^{-1}[\mathbf{x}_{k+1} - \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{x}_k]$$

$$-\frac{1}{2}[\mathbf{x}_k - \hat{\mathbf{x}}_k]^T\mathbf{P}_k^{-1}[\mathbf{x}_k - \hat{\mathbf{x}}_k] \tag{3.78}$$

where $\mathbf{x}_k$ is the state at stage k; $\mathbf{\Phi}_k$ and $\mathbf{\Gamma}_k$, are transition matrices which relate the state at a later time to one at the current time; $\mathbf{P}_k$ and $\mathbf{Q}_k$ are the corresponding error covariance matrices; $\mathbf{H}_k$ is the measurement transition matrix; and vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are the random process disturbance and measurement error, respectively, both of which are assumed to be Gaussian white-noise processes with zero mean and known covariance.

Fixed-interval smoothing consists of a forward recursive filter pass followed by a backward smoothing pass. The forward filter is identical to the extended Kalman filter (EKF) algorithm described in section 3.2.3. The backward pass requires that the a priori and a posteriori estimates $\hat{\mathbf{x}}_k$, and associated covariance matrices $\boldsymbol{P}_k$, be saved. The backward sweep starts with initial conditions which are the last state estimate and covariance computed using the forward filter sweep (44 p. 128). With each step of the backward pass, the old estimate from the forward filter is updated to yield an improved smoothed estimate. This improved estimate is based on all the measurement data. The recursive equations for the backward sweep are (37 p. 248):

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^k + \mathbf{S}_k\left(\hat{\mathbf{x}}_{k+1}^l - \mathbf{\Phi}(t_{k+1}, t_k)\hat{\mathbf{x}}_k^k\right) \tag{3.79}$$

Again, the notation $\hat{\mathbf{x}}_k^l$ means the estimate of $\mathbf{x}$ at time k, given measurements, $\mathbf{y}_1$ to $\mathbf{y}_l$. The smoothing gain $\mathbf{S}$ is given by (37 p. 248):

$$\mathbf{S}_k = \mathbf{P}_k^k \mathbf{\Phi}^T(t_{k+1}, t_k)\left[\mathbf{P}_{k+1}^k\right]^{-1} \tag{3.80}$$

The error covariance for the smoothed estimates is given by the recursive equation (37 p. 250):

$$\mathbf{P}_k^l = \mathbf{P}_k^k + \mathbf{S}_k\left(\mathbf{P}_{k+1}^l - \mathbf{P}_{k+1}^k\right)\mathbf{S}_k^T \tag{3.81}$$

It should be noted that the smoothed error covariance matrix is not required in order to compute the state estimates in the backward pass. This is, of course, different than for the forward filter in which the filtered error covariance is needed to compute the gain used in computing updated state estimates (43 p. 5).

The formulation described above is known as the Rauch, Tung, and Striebel (RTS) smoother and is different from the square-root information filter and smoother (SRIF&S) that is ultimately used in section 3.2.5 with the backward-smoothing extended Kalman filter. As was discussed in the previous section, the two fundamental reasons for filter divergence include: 1) inaccuracies in the mathematical model used to describe the dynamics process or the model used to relate the observations to the state and 2) errors that occur in the measurement update of the state error covariance matrix. In particular, the presence of round-off or truncation errors can cause the matrix to become nonpositive definite (37 p. 331). Square root covariance filters minimize the effect due to the loss of significant digits during the computational procedure, by replacing the state error covariance matrix with its square root such that (37 p. 360)

$$\mathbf{P}^* = [\mathbf{R}_{xx}]^{-1}[\mathbf{R}_{xx}]^{-T} \tag{3.82}$$

where $\mathbf{R}_{xx}$ is the square-root information matrix associated with $\mathbf{x}$.

Consequently, Mark Psiaki uses the SRIF introduced by Dyer and MeReynolds in the BSEKF in order to incorporate estimation of process noise vectors and to take advantage of the improved numerical stability of the SRIF&S form over the original Kalman filter/smoother formulation presented in this section. Square root formulations increase numerical computation accuracy by “guaranteeing positive-definiteness of the associated covariances and by decreasing the condition numbers of the manipulated matrices” (33 p. 1). Additionally, the computation time scales linearly with the number of stages N, which is much better than a batch least-squares algorithm, whose execution time scales according to $(N)^3$ (33 p. 5). Since the algorithm is included in the BSEKF, a summary of the SRIF&S will be provided in this section in preparation for the more detailed discussion to follow. Though the specific equations needed to implement the filter are saved for Section 3.2.5.1., conceptually, the SRIF&S can be outlined as follows:

1. Consider a fixed-length interval containing $N+1$ noisy measurements, indexed from $\mathbf{y}_0$ to $\mathbf{y}_N$. The filtering problem is to find the best estimate of the state at stage N conditioned on the measurements up to and including stage N: $\hat{\mathbf{x}}_N = E[\mathbf{x}_N | \mathbf{y}_0, \dots, \mathbf{y}_N]$. The fixed-interval smoothing problem is thus, to find the best estimate of the state time history for stages 0 to N conditioned on the measurements for the entire interval:

$$\mathbf{x}_k^* = E\left[\mathbf{x}_k | \mathbf{y}_0, \dots, \mathbf{y}_N\right] \text{ for } k = 0, 1, 2, \dots, N \tag{3.83}$$

2. This method finds $\mathbf{x}_k$, $\mathbf{w}_k$, $\mathbf{v}_k$, and $\mathbf{v}_{w(k)}$ sequences that satisfy the linearized restrictions given by equations (3.76), (3.77),

$$\mathbf{R}_{ww(l)}\mathbf{w}_l = \mathbf{z}_{w(l)} - \mathbf{v}_{w(l)} \tag{3.84}$$

and minimizes the cost function of the errors (45 p. 2)

$$J = \frac{1}{2}\left(\sum_{l=0}^{N} \mathbf{v}_k^T \mathbf{v}_k + \sum_{l=0}^{N-1} \mathbf{v}_{w(k)}^T \mathbf{v}_{w(k)}\right) \tag{3.85}$$

Equation (3.84) is used for calculating statistics of the process noise's measurement error where $\mathbf{R}_{ww(k)}$ is the square-root information matrix associated with $\mathbf{w}_k$, found by Cholesky factorizing the process noise covariance matrix $\mathbf{Q}$ (this can be easily accomplished using the Matlab function *chol*), $\mathbf{z}_{w(k)}$ is the process noise measurement vector, and $\mathbf{v}_{w(k)}$ is the process noise measurement noise vector.

3. The algorithm works in a recursive manner, starting at stage k with a priori state information. The filter first performs a measurement update to combine its a priori state data with the measurement equations for that stage, equation (3.77). The propagation phase of the generalized SRIF computes the a priori state information at stage $k+1$ using the a posteriori state information at stage k in conjunction with equations (3.76) and (3.84) for stage k. The dynamic model is used to eliminate some stage-k variables from

the information equations, and then they are QR-factorized to obtain an information equation for $\mathbf{x}_{k+1}$ that is independent of all remaining variables (45 pp. 2 - 3). Complete QR factorizations of matrices are needed in the generalized algorithm in order to deal with singularities and may be readily calculated using the Matlab function called *qr*.

4. The smoother then uses data from the filtering solution to execute a recursive backwards pass. Each iteration of this recursion starts with the smoothed state estimate at stage $k + 1$, $\mathbf{x}_{k+1}^{*}$. It uses an information equation and a part of the difference equation to determine the smoothed process disturbance estimate $\mathbf{w}_{k}^{*}$. This process noise, $\mathbf{x}_{k+1}^{*}$, and the difference equation determine a component of the smoothed state at stage k, $\mathbf{x}_{a(k)}^{*}$. $\mathbf{x}_{a(k)}^{*}$ is used in another information equation to determine another component of $\mathbf{x}_{k}^{*}$, $\mathbf{x}_{b(k)}^{*}$. Finally, the smoothed and deterministic components are assembled and transformed to compute $\mathbf{x}_{k}^{*}$, which completes the recursion (45 p. 4).
5. A backwards covariance recursion is developed by using the equations of the backwards state recursion, the definition of covariance, and the expectation operation. The recursion begins at stage $j + 1 = N$, where the smoothed state covariance is known because it is the same as the filtered state covariance $\mathbf{P}_{N}^{*} = \widehat{\mathbf{P}}_{N}$.

A more thorough description of the square-root information filter and smoother, which includes derivations and detailed equations, can be found in References (37), (45), and (43).

3.2.5 Backward-Smoothing Extended Kalman Filter

The backward-smoothing extended Kalman filter (BSEKF) can be classified as a special kind of causal batch filter which blends together many of the key attributes of the algorithms discussed in previous sections. In addition to using a moving-window batch filter with nonlinear Gauss-Newton updates, the current formulation works for a general discrete-time problem which simplifies the mathematics as follows: 1) it eliminates the need to designate deterministic and nondeterministic parts of the state vector and 2) it enable joint probability distributions to be developed which include the process noise vector. The BSEKF is also similar in spirit to an extended Kalman filter (EKF), an iterated extended Kalman filter (IEKF), and an unscented Kalman filter because each algorithm is an approximate method that uses a bounded number of computations. A better understanding of how the BSEKF makes fewer approximations than these other estimators begins by posing the problem as a MAP estimation problem (discussed in Section 3.2.1.3).

The k^{th} step in a nonlinear filtering problem can be expressed in the maximum a posteriori (MAP) form by writing the joint probability density function as $p_k = \exp(-J_k)$ with the cost function (42 p. 16)

$$J_k = \frac{1}{2}\sum_{i=0}^{k-1}\left(\mathbf{w}_i^T \mathbf{Q}_i^{-1}\mathbf{w}_i + [\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]^T \mathbf{R}_{i+1}^{-1}[\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]\right) + \frac{1}{2}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \mathbf{P}_0^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}_0) \tag{3.86}$$

The MAP estimate $\hat{\mathbf{x}}_k$ is the vector $\mathbf{x}_k$ that, along with $\mathbf{x}_i$ and process noise $\mathbf{w}_i$ for $i = 0, 1, \dots, k - 1$, minimizes J_k subject to the dynamics equation

$$\mathbf{x}_{i+1} = f_i(\mathbf{x}_i, \mathbf{w}_i) \text{ for } i = 0, 1, \dots, k-1 \tag{3.87}$$

The process noise covariance is $\mathbf{Q}_i$, the measurement noise covariance is $\mathbf{R}_i$, $\mathbf{y}_{i+1}$ is the measurement at time t_{i+1}, $h_{i+1}(\mathbf{x}_{i+1})$ is the nonlinear measurement model, and $\hat{\mathbf{x}}_0$ is the a priori estimate of the state with covariance $\mathbf{P}_0$. The joint a posteriori probability density of $\mathbf{x}_0$ and $\mathbf{w}_0, \dots, \mathbf{w}_{k-1}$ conditioned on $\hat{\mathbf{x}}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k$, is proportional to $\exp(-J_k)$ if $\mathbf{x}_1, \dots, \mathbf{x}_k$ are generated by iterating the dynamics function. Minimization of the cost J is equivalent to maximization of this probability density. As has already been mentioned, for linear Gaussian estimation problems, MAP estimation is equivalent to minimum mean-square error estimation, in which $\hat{\mathbf{x}}_k =$

$E[\mathbf{x}_k|\hat{\mathbf{x}}_0,\mathbf{y}_1,\ldots,\mathbf{y}_k]$, the conditional mean. For nonlinear problems, the MAP estimate of $\mathbf{x}_k$ will not necessarily equal the conditional mean, but is still a useful estimate (42 p. 17). Selection of the MAP framework for the BSEKF results in a nonlinear least-squares problem, for which there exist powerful solution techniques that have been exploited by its developer, Mark Psiaki, in implementing this algorithm.

One normally solves a sequence of filtering problems that is parameterized by an increasing sequence of terminal stage indices, $k = 1, 2, 3, \ldots$ Consequently, the size of this problem grows with k. The usual EKF avoids this growth by not explicitly recomputing the values of $\hat{\mathbf{x}}_i$ for $i < k$ when $\mathbf{x}_k$ is optimized in the k^{th} step. The iterated EKF improves upon the EKF by iterating the nonlinear measurements update equation for $\hat{\mathbf{x}}_k$, relinearizing about the updated state estimate at each iteration, but it does not explicitly recompute the values of $\hat{\mathbf{x}}_i$ for $i < k$ (33 p. 17). Any Kalman filter implicitly recalculates the past state estimates with each new measurement update; however, this point is often overlooked because estimates in the past are generally of no interest. For linear dynamics and measurements, these past estimates are most favorable; unfortunately, this is not the case when the dynamics or measurements are highly nonlinear. Thus the EKF linearizations are not about the optimal estimates.

The backward-smoothing extended Kalman filter improves on the iterated EKF by relinearizing a finite number of measurements in the past when a new measurement is processed. Therefore, the BSEKF combines some of the properties of an EKF, a smoother, and a sliding-batch estimator and has been shown to have superior performance when the estimation problem contains severe nonlinearities that might otherwise significantly degrade the accuracy or convergence reliability of other filters (41 p. 885). Unlike the EKF, the BSEKF relinearizes the current and past measurement and dynamics functions about the improved guesses of the current and past state and process noise vectors. Appropriate relinearization points are chosen by means of iterative nonlinear smoothing over an interval of time that ends at the current sample time. This approach retains the nonlinearities of a fixed number of stages that precede the stage of interest, and it processes information in earlier stages in an approximate manner (41 p. 887). In his paper, Dr. Mark Psiaki applied the BSEKF to a difficult spacecraft attitude estimation problem in which fewer than three axes were sensed and significant dynamic model uncertainties were present. The filter was able to compensate for this uncertainty via simultaneous estimation of moment of inertia parameters. Using the BSEKF, he was able to demonstrate improved performance and more rapid convergence over either the extended Kalman filter or unscented Kalman filter (UKF) for estimation problems with large initial attitude or attitude rate errors. The results of this performance comparison are provided in Reference (41) along with the original formulation for the BSEKF. A detailed description of the unscented Kalman filter can be found in References (46) and (47), while a more general overview of the algorithm is provided in Reference (42).

The algorithm incorporates a Gauss-Newton iteration to solve for the state vector $\mathbf{x}_k$ and process noise vector $\mathbf{w}_k$ for $i = k - m, \dots, k - 1$ to minimize the cost function (41 p. 887)

$$J = \frac{1}{2}\sum_{i=k-m}^{k-1}\left(\mathbf{w}_i^T\mathbf{Q}_i^{-1}\mathbf{w}_i + [\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]^T\mathbf{R}_{i+1}^{-1}[\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]\right)$$

$$+\frac{1}{2}(\mathbf{x}_{k-m} - \hat{\mathbf{x}}_{k-m}^*)^T(\mathbf{P}_{k-m}^*)^{-1}(\mathbf{x}_{k-m} - \hat{\mathbf{x}}_{k-m}^*) \tag{3.88}$$

where $\hat{\mathbf{x}}_{k-m}^*$ and P_{k-m}^* are used to represent old data and should not be confused with the filtered a posteriori state estimate and corresponding error covariance matrix. Equation (3.88) is also subject to the dynamics function expressed in equation (3.87) for $i \geq k - m$. The cost function is a form of negative feedback that penalizes the computed state vector if it drifts too far from the reference state over the smoothing interval (more precisely, J penalizes measurement error and state estimate error in the manner described in Section 3.2.1.2). Furthermore, equation (3.88) retains all of the nonlinearities of the most recent m stages, with all the nonlinear effects of the previous stages being represented by the quantity $\frac{1}{2}(\mathbf{x}_{k-m} - \hat{\mathbf{x}}_{k-m}^{*})^{T}(\mathbf{P}_{k-m}^{*})^{-1}(\mathbf{x}_{k-m} - \hat{\mathbf{x}}_{k-m}^{*})$. This quadratic second term is an approximation of the cost function, J_{k-m} for fixed $\mathbf{x}_{k-m}$ optimized over all the $\mathbf{x}_k$ and $\mathbf{w}_k$ for $i < k - m$ (41 p. 887). A value m_{target} is used to denote the number of stages, which will be retained and is chosen to balance accuracy and computational effort. When $k \leq m_{target}$, the BSEKF uses $m = k$ stages, and makes no approximations. However, when $k > m_{target}$ it uses $m = m_{target}$ stages, $\hat{\mathbf{x}}_{k-m}^{*}$, and $\mathbf{P}_{k-m}^{*}$ to approximate the optimal cost function $J_{\mathrm{opt}[k-m]}(\mathbf{x}_{k-m})$. Because the measurement equation, $\mathbf{y}_{i+1} = h_{i+1}(\mathbf{x}_{i+1}) + \mathbf{v}_{i+1}$ and the dynamic equation, $\mathbf{x}_{i+1} = f_i(\mathbf{x}_i, \mathbf{w}_i)$ are both included in the Gauss-Newton cost minimization for not one but m stages, the nonlinearities in both the measurement and dynamic equations are treated explicitly for those m stages (42 p. 17). This yields a more accurate representation of the cost minimization problem if enough of the past stages are treated explicitly by the nonlinear smoother. A large m, therefore, makes both the optimal state estimate $\hat{\mathbf{x}}_k$ insensitive to the accuracy of the approximation of $J_{\mathrm{opt}[k-m]}(\mathbf{x}_{k-m})$ and the smoothed estimate of $\hat{\mathbf{x}}_i$ for $i < k - m$ insensitive to the terminal time of the smoothing problem. Finally, as was mentioned above, the target number of explicitly optimized stages is a design choice. It may be practical to modify m_{target} dynamically. m_{target} could be set to a larger value during initialization to ensure convergence and greater accuracy, then lowered to mitigate computation time, once steady state performance is achieved. For the time being, however, m_{target} is held fixed in the interest of algorithm simplicity (41 pp. 886 - 887).

3.2.5.1 Detailed BSEKF Algorithm Description

The nonlinear least-squares smoothing problem presented in equations (3.87) and (3.88) is solved in the BSEKF algorithm using the guarded Gauss-Newton method described in Section 3.2.2.1. This iterative process takes an initial guess and computes a converging sequence of estimates; that is, each guess has a lower least-squares cost than the previous one. Convergence is guaranteed by a guarding procedure that scales down the computed increment to a given estimate, and if necessary, forces a decrease in the least-squares cost from one guess to the next. Additionally, the algorithm preserves a set batch size over which it filters and smooths. The forward filtering pass is performed using an extended Kalman filter, though the form employed in the BSEKF is slightly different than what is presented in Section 3.2.3. Similarly, the backward-smoothing pass is accomplished via the square-root information smoother discussed in Section 3.2.4. The number of stages retained by the BSEKF grows until it reaches the requisite interval size and then proceeds to "slide along" until all the observations in the pass have been processed. The estimator described below is therefore, a synthesis of all the algorithms discussed in the proceeding sections, utilizing aspects of a sliding batch least-squares estimator, a guarded Gauss-Newton algorithm, extended Kalman filter, and fixed-interval smoother.

The following algorithm is adapted from Mark Psiaki's backward-smoothing extended Kalman filter (Reference (41)), and employs the following notation: $\mathbf{a}_b^c$ refers to a vector $\mathbf{a}$ at time step b for iteration c. All time points are referenced from the current time, denoted using the subscript k. Typically, the times points are incremented from $k-m$ to $k-1$ meaning that m time points or stages are currently being utilized and operated on. Hence, the collection of m previous state vectors, observation vectors, covariance matrices, noise vectors, and so on, are all collectively referred to in the algorithm as the m-buffer or cache. Since the program has been implemented in Matlab, which does not allow indexing from zero, the initial guesses are written as $\mathbf{a}_1^c$ rather than in the more traditional form:$\mathbf{a}_0^c$. The backwards smoothing extended Kalman filter proceeds as follows (41 pp. 887 - 889):

1. Choose values for the target number of stages to be used by the fixed-interval smoother (m_{target}), the maximum number of Gauss-Newton iterations to be performed (j_{max}), and cutoff values for the cost minimization and guarded Gauss-Newton procedure (J_ε) and (γ_ε) respectively.
2. Set $m = 0, k = 1, j = 0$ and assign the initial guesses for the state and process noise vectors, i.e. $\hat{\mathbf{x}}_{k-m}^j$ and $\mathbf{w}_{k-m}^j, \mathbf{w}_{k-m+1}^j, \mathbf{w}_{k-m+2}^j, \dots, \mathbf{w}_{k-1}^j$. The initial state guess is the set of attitude parameters such that $\hat{\mathbf{x}}_1 = \{\mathbf{y}_0 \quad \boldsymbol{\omega}_0 \quad \boldsymbol{p}_{I(0)}\}$ [2]. The initial guess for each of the process noise vectors is typically zero. Set the initial state error covariance matrix $\mathbf{P}_1$, process noise covariance matrices $\mathbf{Q}_i$, and measurement noise covariance matrices $\mathbf{R}_i$. Factor each matrix using Cholesky decomposition [Matlab – *chol*]: $\mathbf{P}_1 = [\mathbf{R}_{xx}^{-1}][\mathbf{R}_{xx}^{-T}]$, $\mathbf{Q}_i = [\mathbf{R}_{ww}^{-1}][\mathbf{R}_{ww}^{-T}]$, and $\mathbf{R}_i = [\mathbf{R}_{vv}^{-1}][\mathbf{R}_{vv}^{-T}]$, to obtain $\mathbf{R}_{xx}$, $\mathbf{R}_{ww}$, and $\mathbf{R}_{vv}$, the square root information matrices associated with $\mathbf{x}$, $\mathbf{w}$ and $\mathbf{v}$, in that order. These matrices will be used later in the square root information filter and smoother (SRIF&S). Details on how the initial estimates for the state and covariance matrices are calculated are presented in Section 4.3.2, with specific examples given in Section 5.1.1.
3. Set the stage counter $m = k$, if $k < m_{target}$ or $m = m_{target}$, if $k \geq m_{target}$. Establish the total number of observations n to be processed after using the first measurement in the

[2] The initial state $\hat{\mathbf{x}}_1$ is composed of the vector components of the attitude quaternion, angular velocity vector, and moment of inertia parameters. Since the observation vector is made up of the full quaternion, only the first three elements of $\mathbf{y}_0$ are actually used as part of the initial estimate.

initial state estimate and begin the observation loop. The observation counter k, is used to identify each measurement from 1 to n and control operations in the outermost loop.

Time	**State**	**Observation**
t_1	$\hat{\mathbf{x}}_1$	← $\mathbf{y}_0$
t_2	$\mathbf{x}_2$	$\mathbf{y}_1$
t_3	$\mathbf{x}_3$	$\mathbf{y}_2$
$\vdots$	$\vdots$	$\vdots$
t_{n+1}	$\mathbf{x}_{n+1}$	$\mathbf{y}_n$

4. If m_{target} observations have been processed, i.e. $k \geq m_{target}$, then the fixed-interval cache (m-buffer) of saved variables needed for the smoothing process has been filled. Values will therefore, be replaced rather than appended. In order to set up the next smoothing problem the caches for the subsequent variables need to be shifted by making the following assignments:

$$\Delta\mathbf{z}_{x(k-m)} = \Delta\mathbf{z}_{x(k-m+1)} \tag{3.89}$$

$$\mathbf{R}_{xx(k-m)} = \mathbf{R}_{xx(k-m+1)} \tag{3.90}$$

$$\mathbf{w}_{k-m} = \mathbf{w}_{k-m+1} \tag{3.91}$$

$$\hat{\mathbf{x}}_{k-m} = \hat{\mathbf{x}}_{k-m+1} \tag{3.92}$$

where $\Delta\mathbf{z}_x$ is the state estimation error information vector. The vectors and matrices at sample $i = k - m + 1$ represent the values from the last Gauss-Newton iteration of the smoothing problem that ends at the current observation k. Having shifted these final smoothed estimates into the preceding position in the m-buffer, they may now be set to zero in preparation for the next smoothing problem. The quantities used to approximate the optimal cost function are then found by

$$\hat{\mathbf{x}}^*_{k-m} = \hat{\mathbf{x}}_{k-m} + \mathbf{R}^{-1}_{xx(k-m)}\Delta\mathbf{z}_{x(k-m)} \tag{3.93}$$

$$\mathbf{P}^*_{k-m} = \mathbf{R}^{-1}_{xx(k-m)}\mathbf{R}^{T}_{xx(k-m)} \tag{3.94}$$

5. If m_{target} observations have not yet been processed, i.e. $k < m_{target}$, then the m-buffer is still being filled and there is no need to perform the assignments in the proceeding step. Instead:

$$\hat{\mathbf{x}}^*_{k-m} = \hat{\mathbf{x}}_1 \text{ and } \mathbf{P}^*_{k-m} = \mathbf{P}_1 \tag{3.95}$$

At this point in the algorithm the cache will be growing from the size of the current stage counter value m to the target number of stages m_{target}. The range in cache size is summarized in the table below:

Table 3-1 Algorithm Cache Size

Cache Type	Dimension Range for $k \leq m_{target}$ $(rows \times columns \times sections)$ *	Fixed Dimension for $k > m_{target}$ $(rows \times columns \times sections)$*
Scalars	$(1 \times m) \rightarrow (1 \times m_{target})$	$(1 \times m_{target})$
Vectors	$(r \times m) \rightarrow (r \times m_{target})$	$(r \times m_{target})$
Matrices	$(r \times c \times m) \rightarrow (r \times c \times m_{target})$	$(r \times c \times m_{target})$

* Sections are an element of depth used to indicate the number of matrices in the cache (rows = 1st Dimension × columns = 2nd Dimension × sections = 3rd Dimension)

6. Compute the inverse state error covariance matrix $\mathbf{P}^{*-1}_{k-m}$. This is the covariance used in the approximation for the cost function for all stages before $k - m$.

7. Initialize the cost function $(J^0 = 0)$, reset the cost convergence variable $(J_{newapprox} = 0)$, set the Gauss-Newton iteration counter $j = 0$, and begin the guarded Gauss-Newton loop (the loop will continue until $j < j_{max}$).

8. Starting from the state estimate $\hat{\mathbf{x}}^j_{k-m}$, propagate the state forward in time from $i = k - m, \ldots, k-1$ using the process noise guesses $\mathbf{w}^j_{k-m}, \ldots, \mathbf{w}^j_{k-1}$ and dynamics function

$$\hat{\mathbf{x}}_{i+1} = f_i(\hat{\mathbf{x}}_i, \mathbf{w}_i) \text{ for } i = k - m, \ldots, k - 1 \tag{3.96}$$

The state guesses $\hat{\mathbf{x}}^j_{i+1}, \ldots, \hat{\mathbf{x}}^j_k$ are calculated using the 4th-order Runge-Kutta numerical integration method described in Section 2.3.4.1 [Matlab – *ode45*] (17).

9. Evaluate the state transition matrix $\boldsymbol{\Phi}_i = \partial \mathbf{f}_i / \partial \mathbf{x}_i$ and process noise transition matrix $\boldsymbol{\Gamma}_i = \partial \mathbf{f}_i / \partial \mathbf{w}_i$ at the points $(\mathbf{x}_i^j, \mathbf{w}_i^j)$ for $i = k - m, \ldots, k - 1$ [Matlab – symbolic toolbox + *Jacobian*] (17). If l is used to indicate the dimension of the state vector $(l \times 1)$, then both $\boldsymbol{\Phi}_i$ and $\boldsymbol{\Gamma}_i$ are $l \times l$ matrices. The linearized dynamics matrices can propagated to each observation time by either: 1) numerically integrating the analytic equations for the partial derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$ and $\mathbf{w}$ or 2) performing finite differencing using the following set of equations to find the central difference:

$$\boldsymbol{\Phi}_i = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} = \frac{f_i(\mathbf{x}_i + \mathbf{s}) - f_i(\mathbf{x}_i - \mathbf{s})}{2\boldsymbol{s}} \tag{3.97}$$

$$\boldsymbol{\Gamma}_i = \frac{\partial \mathbf{f}_i}{\partial \mathbf{w}_i} = \frac{f_i(\mathbf{w}_i + \mathbf{s}) - f_i(\mathbf{w}_i - \mathbf{s})}{2s} \tag{3.98}$$

where $\mathbf{s}$ is the component step size vector, that is, the finite amount by which each element of the state and/or process noise vector is incremented up and down before numerically integrating the dynamics function. The vector is of length l and is composed of all zeros except at the position of the state value being incremented in that particular iteration. It is important to note that because the state must be propagated to each observation time twice (i.e., for both the forward and backward finite difference) for every component in the state vector (l-times), this approach is computationally expensive and should only be considered as a method of last resort. The alternative to numerical partial derivatives is to find the analytic expressions for the partial derivatives. This eliminates the major inaccuracy of the numerical computations resulting either from no solution or from an erroneous one. The use of analytic partial derivatives also eliminates the problem of undefined or incorrect solutions choice at the cost of potentially very complex algebra. The principal advantage of the numerical procedure is that it is both simple and direct.

10. Collect the propagated state estimate $\hat{\mathbf{x}}_{i+1}^{j}$ and observation $\mathbf{y}_i$ at the corresponding time in order to evaluate the measurement error vector

$$\Delta\mathbf{y}_{i+1} = \mathbf{y}_i - h_{i+1}\left(\hat{\mathbf{x}}_{i+1}^{j}\right) \tag{3.99}$$

and measurement transition matrix $\mathbf{H}_{i+1} = \partial\mathbf{h}_{i+1}/\partial\mathbf{x}_{i+1}$. In these equations, $\mathbf{h}_{i+1} = h_{i+1}\left(\hat{\mathbf{x}}_{i+1}^{j}\right)$ is a vector function mapping components form the propagated state to a form which is comparable to that of the observation vector. Additionally, $\Delta\mathbf{y}_{i+1}$ is the residual used to quantify the additive measurement noise vector $\mathbf{v}_{i+1}$ in the nonlinear measurement model given by

$$\mathbf{y}_{i+1} = h_{i+1}(\mathbf{x}_{i+1}) + \mathbf{v}_{i+1} \tag{3.100}$$

11. Calculate the cost function at iteration j if the cost has not been computed for the current observation k:

$$J^j = \frac{1}{2}\sum_{i=k-m}^{k-1}\left(\mathbf{w}_i^T\mathbf{Q}_i^{-1}\mathbf{w}_i + [\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]^T\mathbf{R}_{i+1}^{-1}[\mathbf{y}_{i+1} - h_{i+1}(\mathbf{x}_{i+1})]\right)$$

$$+\frac{1}{2}(\hat{\mathbf{x}}_{k-m} - \hat{\mathbf{x}}_{k-m}^{*})^T(\mathbf{P}_{k-m}^{*})^{-1}(\hat{\mathbf{x}}_{k-m} - \hat{\mathbf{x}}_{k-m}^{*}) \tag{3.101}$$

12. Set the sample index $i = k - m$ and begin the forward square-root information filtering pass, by assigning

$$\Delta\mathbf{z}_{\mathbf{x}(i)} = \mathbf{R}_{xx(k-m)}\left(\hat{\mathbf{x}}_{k-m}^{*} - \mathbf{x}_{k-m}^{j}\right) \tag{3.102}$$

13. Form the block matrix on the right hand side of the following equation and orthonormal/upper-triangular (QR) factorize that matrix to determine the matrices on the left left-hand side [Matlab – *qr*]:

$$\mathbf{T}_i \begin{bmatrix} \overline{\mathbf{R}}_{ww(i)} & \overline{\mathbf{R}}_{wx(i)} \\ 0 & \mathbf{R}_{xx(i+1)} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ww(i)} & 0 \\ -\mathbf{R}_{xx(i)}\boldsymbol{\Phi}_i^{-1}\boldsymbol{\Gamma}_i & \mathbf{R}_{xx(i)}\boldsymbol{\Phi}_i^{-1} \\ 0 & \mathbf{R}_{vv(i+1)}\mathbf{H}_{i+1} \end{bmatrix} \tag{3.103}$$

The block matrix has dimensions $(2(l) + \ell) \times 2(l)$, where l and ℓ are the length of the state vector and measurement vector respectively; thus, the resulting orthonormal matrix T and upper-triangular matrix are both square with dimensions: $(2(l) + \ell) \times (2(l) + \ell)$.

14. Compute the information vectors associated with the state estimation error, process noise, and measurement noise ($\Delta\mathbf{z}_x, \Delta\bar{\mathbf{z}}_w$, and $\Delta\boldsymbol{z}_r$) by performing the following matrix multiplication:

$$\begin{bmatrix} \Delta\bar{\mathbf{z}}_{w(i)} \\ \Delta\mathbf{z}_{x(i+1)} \\ \Delta\mathbf{z}_{r(i)} \end{bmatrix} = \mathbf{T}_i^T \begin{bmatrix} -\mathbf{R}_{ww(i)}\mathbf{w}_i^j \\ \Delta\mathbf{z}_{x(i)} \\ \mathbf{R}_{vv(i+1)}\Delta\mathbf{y}_{i+1} \end{bmatrix} \tag{3.104}$$

Steps 13 and 14 are essentially combining the state propagation and measurement update into a single QR factorization step.

15. If $i = k - 1$, then terminate the iteration and proceed to step 16; otherwise, set $i = i + 1$ and go back to step 13.

16. Calculate the incremental change in the state estimate

$$\Delta\mathbf{x}_k = \mathbf{R}_{xx(k)}^{-1}\Delta\mathbf{z}_{x(k)} \tag{3.105}$$

17. Set $i = k - 1$ and perform the backward square-root information smoothing pass, which consists of the following set of equations:

$$\Delta\mathbf{w}_i = \mathbf{R}_{ww(i)}^{-1}\left[\Delta\bar{\mathbf{z}}_{w(i)} - \overline{\mathbf{R}}_{wx(i)}\Delta\mathbf{x}_{i+1}\right] \tag{3.106}$$

$$\Delta\mathbf{x}_i = \boldsymbol{\Phi}_i^{-1}[\Delta\mathbf{x}_{i+1} - \boldsymbol{\Gamma}_i\Delta\mathbf{w}_i] \tag{3.107}$$

18. If $i = k - m$ terminate the iteration and go to step 18; if not, set $i = i - 1$ and return to step 17.

19. Having completed the square-root information filter and smoother (SRIF&S) portion of the algorithm, the guarded Gauss-Newton method can now be used to search for the minimum arguments of the cost function given by equation (3.101) in step 11. Set the initial trial search step size $\gamma = 1$.

20. Calculate the candidate next guess of the smoothed solution by computing the state and process noise vectors with the addition of the corrections obtained from the SRIF&S given in step 12 – 18:

$$\mathbf{x}_{k-m}^{j+1} = \mathbf{x}_{k-m}^{j} + \gamma \Delta \mathbf{x}_{k-m} \tag{3.108}$$

$$\mathbf{w}_i^{j+1} = \mathbf{w}_i^{j} + \gamma \Delta \mathbf{w}_i \text{ for } i = k-m, \ldots, k-1 \tag{3.109}$$

$$\hat{\mathbf{x}}_{i+1}^{j+1} = f_i\left(\hat{\mathbf{x}}_i^{j+1}, \mathbf{w}_i^{j+1}\right) \text{ for } i = k-m, \ldots, k-1 \tag{3.110}$$

Equations (3.108) – (3.110) are in effect a repetition of steps 8 and 10 using the refined estimate for the state. Therefore, in addition to propagating the state to each observation time in the m-buffer, the corresponding observation vectors needs to be retrieved so that the residuals can be determined for the costing function.

21. Recalculate the cost J^{j+1} by evaluating equation (3.101).

22. If $J^{j+1} \geq J^{j}$, then activate the guarding procedure in order to ensure a reduction in cost, by setting $\gamma = 0.5\gamma$ and going back to step 20. Otherwise, proceed to step 23.

23. Compute the linearized prediction of the cost and determine whether convergence has been reached:

$$J_{newapprox} = \frac{1}{2} \sum_{i=k-m}^{k-1} \Delta \mathbf{z}_{r(i)}^{T} \Delta \mathbf{z}_{r(i)} \tag{3.111}$$

If $J_{newapprox}^{j+1} - J_{newapprox}^{j} \leq J_{\varepsilon}$ is sufficiently small, then the estimate has converged to the local cost minimum. Alternatively, if the actual new cost based on nonlinear dynamics and measurement functions is nearly the same the linearized prediction, then the algorithm may be terminated in very few steps because the linearized model is very accurate. If either of these conditions are true, if $j > j_{max}$, or if $\gamma = 0$, then assign the state estimate for the current observation time k: $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{j+1}$, obtained in step 20. Return to step 3 to begin processing the next observation and set $k = k + 1$. If the algorithm has not yet converged and j is not yet too large ($j < j_{max}$), set $j = j + 1$ and go back to step 8.

24. Note that if trial search step size gets to be very close to zero, that is if $\gamma < \gamma_{\varepsilon}$, then algorithm is either searching for the next state in the wrong direction or has already found the estimate that minimizes the cost; in either case print a warning message, set $\gamma = 0$, and return to step 8 for one last iteration. If multiple warning messages appear in a row then the state estimates are not being refined and there is a strong chance that the BSEKF will diverge. If this occurs then further 'tuning' is in order and different initial values for the state, measurement, and/or process noise covariance matrices (**P**, **R**, and **Q**) should be tried.

3.2.5.2 Specifics on Iterations and Indexing

Since much of what makes the BSEKF challenging to implement lies in the intricate looping structure and offset indexing scheme, the following simple example has been provided to help clarify the dynamic behavior of the algorithm itself while in operation.

Table 3-2 Iterations and Indexing Operations

$m_{target} = 10$	k	m	$i = k - m: k - 1$	$ic = i - ((k - m) - 1)$
$k \leq m_{target}$	1	1	0:0	1:1
	2	2	0:1	1:2
	3	3	0:2	1:3
	⋮	⋮	⋮	⋮
	10	10	0:9	1:10
$k > m_{target}$	11	10	1:10	1:10
	12	10	2:11	1:10
	13	10	3:12	1:10
	⋮	⋮	⋮	⋮
	n	10	n-10:n-1	1:10

The BSEKF utilizes the subsequent list of critical variables: k is the observation counter, n is the total number of observations to be processed, m is the current size of the smoothing interval, m_{target} is the target size of the interval, i is the iteration counter, and ic is an indexing value. The variable j is then used to keep track of the number of Gauss-Newton iterations.

In the example provided above, m_{target} is set to 10; this represents the maximum number of observations over which the algorithm will propagate the state vector and try to minimize the residuals between the estimated and observed attitude. When the algorithm begins to run, only a single observation - designated with k - is present in the cache, called the m-buffer, the current size of which is denoted using m. In accordance with the MAP problem, the algorithm will attempt to keep the cost/penalty function as small as possible over the range $i = k - m: k - 1$. At this point, the BSEKF operates in a manner which is indistinguishable from the traditional EKF, since the extended Kalman filter can be interpreted as a BSEKF that uses $m_{target} = 1$ and only one Gauss-Newton step. As more observations are added on subsequent iterations, the size of the cache that the algorithm is using to approximate and refine the state begins to grow. When the current observation value exceeds the target stage size, the algorithm stops adding observations to the m-buffer and begins to shift, attaching the most recent estimates to the end of the cache and deleting the oldest. From this point forward, the number of stages that will be retained is fixed at 10 and the algorithm begins to "slide along" until all n observations have been processed. The iteration counter i regulates the size of the m-buffer over time and the indexing value ic ensures that all the vectors and arrays used throughout the filtering sub-process are properly referenced and ordered within the cache.

To further illustrate the algorithm dynamics, the following diagram (see Figure 3.14) depicts the growth and then sliding effect that occurs within the observation cache (y) as k progresses from 1 to n = 8, with a target number of retained stages equal to 5. Note that this example is different from the one provided in Table 3-2, which uses an unspecified total number of observations and an m_{target} of 10.

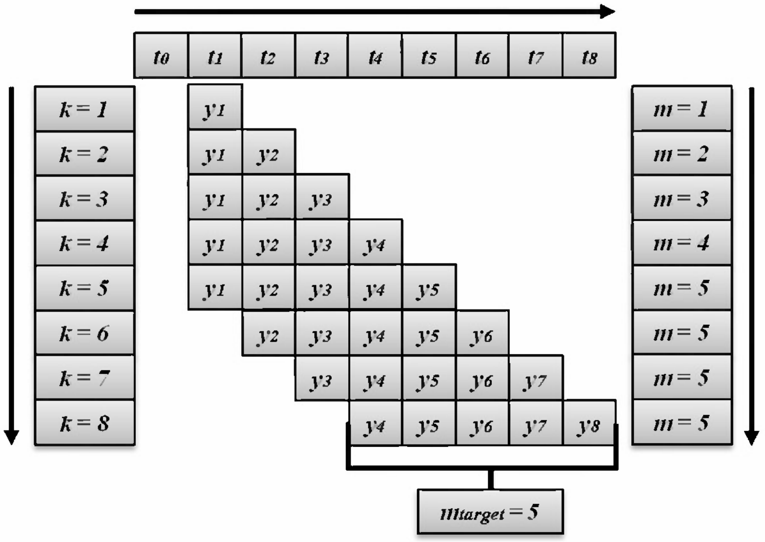


Figure 3.14 Depiction of the growth and sliding effect that occurs in the observation cache over time. The target number of stages in this example is 5 and the total number of measurements to be processed is 8.

3.2.5.3 Relevance to this Specific Research Problem

The strength of the BSEKF lies in its ability to rapidly recover from poor initial estimates of the state vector and covariance matrices, and still achieve greater accuracy than more widely used algorithms. It accomplishes this by maintaining a backlog of observations over which the algorithm is able to filter and smooth. However, building up and preserving information comes at a cost, namely significant computational time. The filter is ideal for solving attitude estimation problems using radar observations, due to the large number of uncertainties involved with remotely measuring the attitude. In addition to having greater measurement noise than more conventional sensors, using externally derived observations, further complicates the modeling process, since the internal dynamics and mass distribution of the satellite are completely unknown. Since the internal composition is not readily observable, the problem is necessarily constrained to spacecraft, which utilize either passive attitude control (i.e. reliance on a vehicle's natural response to the environment to maintain a nominal orientation) or have experienced an anomaly that has resulted in the loss of active command/control of the vehicle's attitude (i.e. use of computed actuator torques to achieve and maintain the desired pointing requirements is no longer possible). Restricting the use context in this manner ensures that the only forces impacting the rotational motion of the vehicle are due to environmental torques. Approximating the inertia tensor, using the rough physical dimensions of the vehicle (geometric modeling), is used as an initial first guess to get the filter moving in the right direction; but ultimately, the BSEKF's ability to simultaneously estimate and refine these values over time, using indirect measurements (only attitude observations), is invaluable. Consequently, the computational burden is accepted in order to compensate/overcome the greater initial uncertainties and measurement error associated with a problem of this type.

4 Lincoln Attitude Estimation System

The Lincoln Attitude Estimation System (LAES) is a new approach being developed for the Space Situational Awareness Group at MIT Lincoln Laboratory, which attempts to determine the motion of a freely tumbling rigid body from a series of radar images. LAES integrates several legacy hardware and software systems with a backward-smoothing extended Kalman filter and a detailed set of environmental torque models.

Section 2.3.1 and 2.3.2 gave a comprehensive account of the rotational motion of rigid bodies, while Section 2.3.4.3, provided a useful introduction to including sources of torques in the Euler equations. This chapter does not repeat these expositions but rather focuses on the coupling of the rotational motion equations with the radar equations developed in Section 3.1.3. The inherent ambiguity that arises when trying to determine the attitude of a spacecraft from a single radar image adds an interesting challenge to developing a motion analysis algorithm. The approach currently used by the Space Situational Awareness Group, to resolve this uncertainty, is to utilize measurements from a sequence of images and make several strategic assumptions about the satellites' motion capabilities and physical characteristics. Specifically, the existing motion analysis tools suppose torque-free motion ($\boldsymbol{N} = [\boldsymbol{0}]$) of an axial symmetric ($I_1 = I_2$) rigid body, the details of which are presented in Section 2.3.3. The restrictions that these simplifying assumptions impose on the system of equations go both ways, enabling closed-form motion propagation at the expense of inaccurate system dynamics and a loss of operational flexibility. Conversely, the Lincoln Attitude Estimation System accepts the greater complexity associated with incorporating environmental torques and dealing with asymmetrical moments of inertia ($I_1 \neq I_2 \neq I_3$), in order to better model the rotational motion of a given satellite.

4.1 Legacy Hardware and Software Systems

The Space Situational Awareness Group at Lincoln Laboratory (Group 93) operates the Lincoln Space Surveillance Complex (LSSC), comprising the Millstone deep-space satellite tracking radar and the Haystack and Haystack Auxiliary (HAX) wideband satellite imaging radars. These radars are remotely controlled from the Lexington Space Situational Awareness Center (LSSAC), which serves as a data processing and fusion node for the LSSC and other ground- and space-based space surveillance sensors. Two-dimensional radar images are generated using the Advanced Radar Imaging Environment Software (ARIES) by Doppler processing the radar returns collected with HAX in each of the range cells of a coherent wideband radar signal – the Doppler frequency of the center of mass being first removed. Consistent with the methods described in Section 3.1, this process is summarized in Figure 4.1 and results in a sequence of range/range-rate images. For a more in depth explanation of the functions performed by HAX and ARIES, please refer back to Section 3.1 and/or consult References (48) and (49).

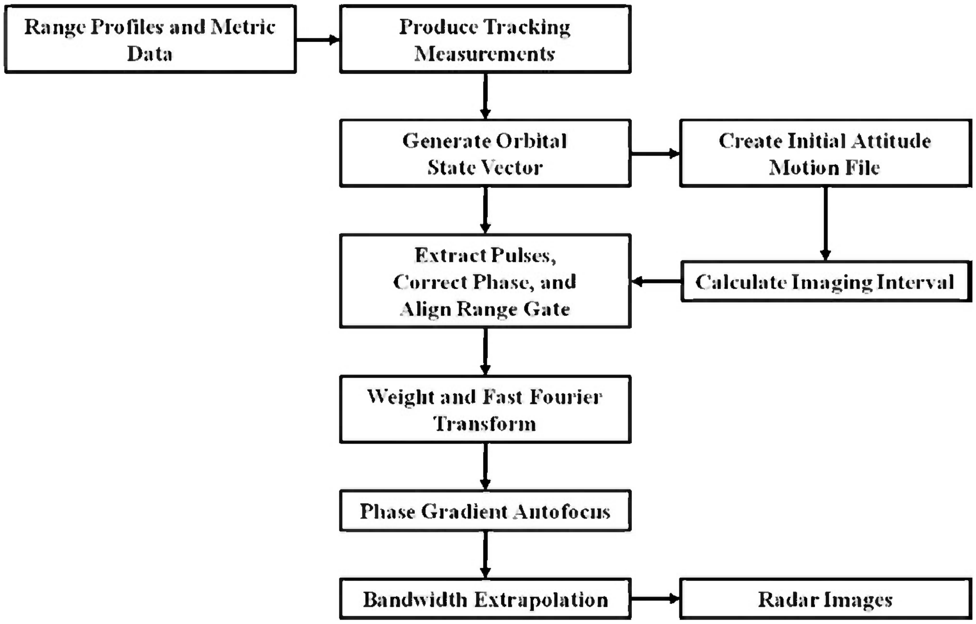

Range Profiles and Metric Data
Produce Tracking Measurements
Generate Orbital State Vector
Create Initial Attitude Motion File
Extract Pulses, Correct Phase, and Align Range Gate
Calculate Imaging Interval
Weight and Fast Fourier Transform
Phase Gradient Autofocus
Bandwidth Extrapolation
Radar Images

Figure 4.1 ARIES is used to read raw data from sensors and perform the image formation processing. Steps include data extraction, aligning the pulses according to the center of mass (given by the orbital state vector), image interval calculation, weighting and Fourier transformation, and phase gradient autofocus to correct phase errors and thereby improve image focus.

The formation of highly resolved, well-focused, and precisely-scaled radar images requires the accurate estimation of the target's translational and rotational motion during the imaging interval. However, in order to accurately estimate the attitude of a spacecraft, one needs a set of consistently-scaled and focused set of images, which begs the question: how does this whole process get started? The answer is the radar collection and processing manager within the LSSAC makes an initial first guess, which will be referred to as the nominal or baseline motion. Using this preliminary approximation of the object's motion relative to the radar line of sight allows the range/range-rate dataset to be roughly scaled in the $\boldsymbol{x}$ direction and centered to form a series of discrete-time range/cross-range images for a given pass (4 pp. 1 - 2). Again, the image plane coordinate system $\boldsymbol{P}$ is defined by the plane which includes the radar line of sight and is perpendicular to the instantaneous angular velocity vector $\boldsymbol{\omega}$ of the spacecraft. These conditions are in line with the derivation of a radar image formed using pulse returns (range) and Doppler shift (cross-range).

The attitude of the spacecraft is then determined relative to the image plane using the image-model matching process, discussed in Section 3.1.3. A three-dimensional computer model is, through computer graphics procedures, displayed on top of the two-dimensional image. The analyst rotates the wireframe model and manually scales the cross-range of the image such that the projection of the wireframe onto the image plane coincides with the figure of the target. Therefore, the orthographic projection matrix that the computer graphics procedures computed to display the model, simultaneously describes the orientation of the object within the radar image plane. As a result, the attitude of the plane itself needs to be known in order to correctly determine the attitude of the satellite. However, given that the raw radar measurements only provide range and range-rate information in two-dimensions, it becomes readily apparent that from a single image, one can determine the attitude and angular velocity of a target only up to a rotation around the radar line of sight. Specifically, each image provides four measurements when six $(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$ are needed in order to uniquely determine the motion of the spacecraft. With a solitary radar system, the problem is therefore, underdetermined and additional external information or simplifying assumptions are required (at least initially).

Given the ambiguous nature of the ISAR imagery, an iterative process is necessary to properly/uniquely determine both the attitude of the spacecraft and the image plane into which the target is being projected. Figure 4.2 shows the basic interdependencies that exist between the hardware and software systems involved in the non-cooperative attitude estimation process.

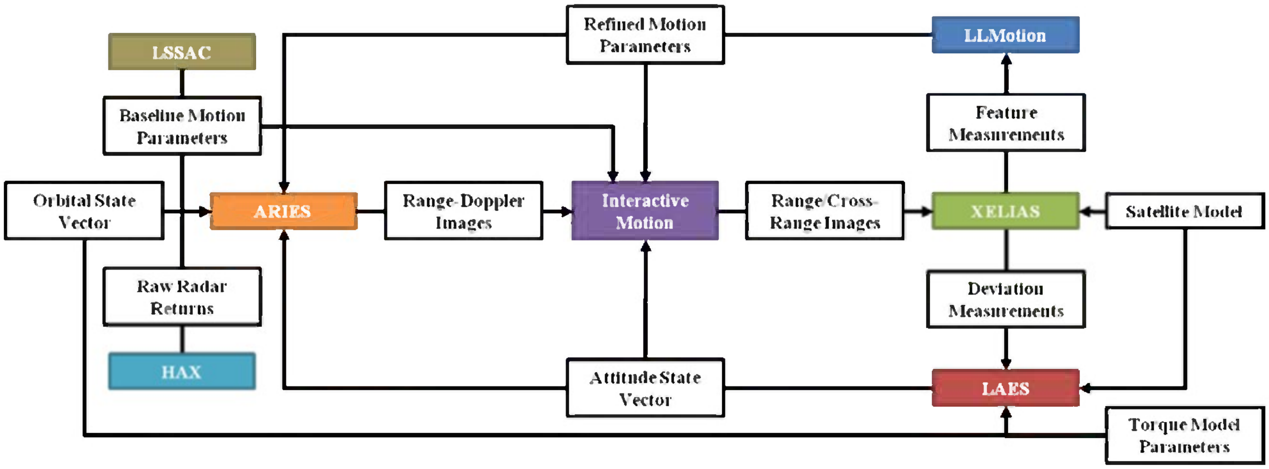


Figure 4.2 High-level system inputs and outputs

It should be noted that while all information appears to flow through Interactive Motion, in reality, data handling functions are performed by the X-Based Enhanced Lincoln Interactive Analysis System (XELIAS). Interactive motion is simply a subprogram in XELIAS, which enables an analyst to directly manipulate the baseline motion parameters which are applied to the pass. Manually altering these global parameters has a direct impact on the cross-range scaling and initial orientation of the wireframe model used in the measurement making process. Accordingly, Interactive Motion is a critical interface, providing operators with immediate feedback and an easy way to qualitatively assess a given motion solution.

LLMotion is the existing force-free motion solution software system, accessible through XELIAS. The program utilizes a batch least-squares filter, in conjunction with a search algorithm, to determine the attitude of a spacecraft by locating the minimum of an objective function calculated over a sample grid of motion parameters. Since only so many degrees of freedom are observable in a single pass with reasonably small error, LLMotion makes several assumptions in order to simplify the motion determination task. While the software has proven quite useful in short term analyses, for more sophisticated endeavors, such as understanding tumbling motion over several hours or multiple passes, a different approach is needed. The

Lincoln Attitude Estimation System (LAES), discussed in Section 4.3, is an attempt to satisfy this need and operates on the premise that the existing Group 93 software can be leveraged to obtain a refined set of radar images and initial state estimate for use in a second iteration.

Therefore, it is important recognize the critical role that LLMotion plays in preprocessing the baseline motion solution used to generate the radar images. The output from the system is a list of all the local minima found in the state space, sorted in ascending order based on their objective function values. This essentially provides the analysts with a means to quantitatively assess of the potential motion solutions within the search space. When dealing with nonlinear systems of equations, producing robust motion analysis software becomes somewhat delicate as there is rarely a guarantee of finding the global extreme that presumably corresponds to the true rotational motion of the spacecraft. Each minimum shows the parameter values associated with that specific grid solution. The analyst is then able to apply and visually examine the prospective motion solutions in order to qualitatively determine which set of parameters gives rise to the "best" alignment. This improved estimate of the satellite's attitude over time becomes the new baseline motion used in the image-model matching process and backward-smoothing extended Kalman filter. Having reduced some of the uncertainty in the measurements enables several of the more limiting assumptions, i.e. those which tend to breakdown over longer time spans, to be removed. The modified BSEKF is subsequently able to further refine the attitude estimate for the first pass as well as propagate the estimate with greater accuracy over the time/observation gaps which separate the imaging passes.

4.2 XELIAS/LLMotion: Single Pass Attitude Determination and Estimation

LLMotion solves for a satellite's rotational motion using relative range and range-rate returns from body scatterers, measured on coherent radar images. Given the measurements from a set of range-Doppler images, solving for the body dimensions and motion of the spacecraft by analytical means would, in general, be very difficult if not impossible. This is due to the fact that the dynamic motion of the spacecraft not only determines the body's orientation as a function of time, but also affects the attitude of the radar image plane and the relationship between Doppler and cross-range position. However, the reverse process, in which the body model and motion parameters are given and the projection into the image plane is derived, is straightforward and forms the basis of LLMotion. Hence, it should be possible to find a motion solution using the reverse process in an iterative loop that performs a least-squares fit to the measurements. This is the technique used in both LLMotion and LAES.

The first critical assumption in the existing Group 93 software is that the torque exerted on the satellite is zero. The important sources of environmental perturbations that are thereby ignored include the gravity gradient, atmospheric drag, solar radiation pressure, and geomagnetic moment. As was discussed in Section 2.3.4.3, while these external forces are typically quite small, they are persistent and therefore, significant in multi-pass attitude estimation problems. Disregarding the variation in the angular velocity of the spacecraft due to environmental torques can result in large motion residuals and phase errors which produce a smearing in the cross-range direction that is proportional to the distance from the center of the image. Second, the satellite is assumed to be a rigid body. While many satellites are not entirely rigid, their articulating components are considerably less massive than their main bodies; the assumption therefore, seems valid. Third, the principal axes of the inertia tensor are aligned with the model axes, i.e., the principal axes are the columns of the attitude matrix $\boldsymbol{A}$. Manmade structures often do follow this principle. Fourth, only spin-precession tumbling motion is considered, constraining the first two moments of inertia to be the same, that is, $I_1 = I_2 \neq I_3$. As was discussed in Section 2.3.3, assuming the spacecraft is a symmetric rigid body in a torque-free environment enables closed-form motion propagation. Removing this constraint requires numerical integration of elliptic integrals or a system of ordinary differential equations expressing Euler's equations of motion. In this thesis, the latter option has been selected and will be covered in greater detail in Section 4.3.2. Finally, as ground-based observations of an orbiting object to do not reveal the total mass, the moments of inertia are arbitrarily normalized. Thus, of the original six free parameters in the inertia tensor, Group 93's software leaves one. In an inertial coordinate system, the inertia tensor is therefore, given by

$$\boldsymbol{I} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T \tag{4.1}$$

where $\boldsymbol{D} = \mathrm{diag}(1, 1, I_z)$.

When expressing the angular velocity of a satellite in terms of the radar image plane, the fundamental ambiguity is, as discussed in Section 3.1.3, a rotation around the radar line of sight of angle θ, $\boldsymbol{R}_2(\theta)$. The analyst has scaled the image by s to estimate cross-range from Doppler. If $\hat{\mathbf{p}}_z$ is, as before, the nominal unit-vector normal to the image plane matrix $\boldsymbol{P}$, then the component of the total angular velocity orthogonal to the RLOS is (27 p. 13)

$$\frac{\boldsymbol{R}_y(\theta)\boldsymbol{\omega}_{total}^z}{s} \tag{4.2}$$

To this must be added the component parallel to the RLOS, which is $\hat{\mathbf{p}}_y|\omega_{total}^y|$, for unknown $\boldsymbol{\omega}_y$. Finally, to obtain the angular velocity of the satellite from the total angular velocity, one must remove the contributions from the orbit, producing (27 p. 16)

$$\boldsymbol{\omega}_{sat} = \boldsymbol{\omega}_{total}^y + \boldsymbol{\omega}_{total}^z - \boldsymbol{\omega}_{orbit} = \hat{\mathbf{p}}_y|\omega_{total}^y| + \left(\frac{\boldsymbol{R}_y(\theta)\boldsymbol{\omega}_{total}^z}{s} - \boldsymbol{\omega}_{orbit}\right) \tag{4.3}$$

Recall from Section 3.1.3.1 that

$$\boldsymbol{A} = \boldsymbol{A}_{nom}\boldsymbol{R}_{align} \tag{4.4}$$

and the true attitude is $\boldsymbol{AR}_y(\theta)$. Combining the preceding arguments and substituting into Euler's equation, relating angular velocity to angular momentum ($\boldsymbol{L} = \boldsymbol{I\omega}$), yields (27 p. 16)

$$\boldsymbol{L} = \boldsymbol{R}_y(\theta)[\boldsymbol{ADA}^T]\boldsymbol{R}_y(-\theta)\left[\widehat{\mathbf{p}}_y\left|\omega_{total}^{y}\right| + \left(\frac{\boldsymbol{R}_y(\theta)\boldsymbol{\omega}_{total}^{z}}{s} - \boldsymbol{\omega}_{orbit}\right)\right] \tag{4.5}$$

The free parameters, after the analyst has obtained the measurements $\boldsymbol{R}_{align}$ and s for the image, are θ, I_z, and $\left|\omega_{sat}^{y}\right|$. Zero torque implies that the angular momentum vector is invariant over the pass. Euler's equation with zero torque and the inertia tensor used in LLMotion and XELIAS immediately implies that eight independent parameters are sufficient to describe rotational motion. Most directly these are attitude, angular velocity, the single free inertia tensor parameter, and the time at which the first six parameters are valid. Holding to the notation used in Section 2.3.3, the eight symmetric parameters for the torque-free motion of a symmetric rigid body are:

$$\mathbf{x}_{param} = \begin{bmatrix} t_0 \\ \alpha \\ \delta \\ \theta \\ \phi_0 \\ \psi_0 \\ \dot{\psi}^{-1} \\ \dot{\phi}^{-1} \end{bmatrix} \tag{4.6}$$

where t_0 is the time to which all the parameters are referenced; α and δ are the right ascension and declination used to specify the precession direction, which is aligned with the angular momentum vector $\boldsymbol{L}$; θ is the coning angle, indicating the angle between the angular momentum vector and the z-axis of the spacecraft (note that $\boldsymbol{D} = \mathrm{diag}(1, 1, I_z)$ is consistent with this choice of axis); the coning angle is the angle between the angular momentum axis and body cone of the spacecraft; x-initial ϕ_0 and z-initial ψ_0 are the preliminary precession and spin angles respectively; and the precession period $\dot{\phi}^{-1}$ and spin period $\dot{\psi}^{-1}$ are the times need by the spacecraft to complete one revolution about the angular momentum axis and z-axis (aligned with the vehicle's longest dimension) respectively (please refer to Figure 2.6 for more detail). The initial first guess (baseline motion solution), made by the individual in the LSSAC, is comprised of these parameters, which roughly describe the rotational motion of the vehicle over time.

A sequence of range/cross-range images are generated using this arbitrary set of motion parameters. These images do not have to be correctly scaled in cross-range, but they must not be so distorted that less than two or three scatterers can be consistently identified. Identifiable returns are selected in pairs, each pair constituting a body feature which is assigned an identifying number. The range and cross-range coordinates of one end of each feature is measured with respect to the other on a number of images. Along with each measurement, a weighting factor can be assigned. The angular velocity used in generating the individual image is also noted.

Each image contributes an equation of the form given in equation (4.5). LLMotion determines the motion of a satellite by searching for the minimum of a cost function calculated over a sampled grid of motion parameters. A numerical batch least-squares minimization, in conjunction with what is called dynamic hill climbing (DHC) or grid search, reduces the error between the measurements from the image and calculated values from the motion parameters. There are two different objective functions used in LLMotion, one for feature measurements and another for angular velocity (32). The meter measure case solves for all the motion parameters, five angles and two rates along with the feature lengths if requested. The angular velocity case can only solve for the right ascension and declination of the precession vector and the precession period. The angular velocity option is, thus, primarily used when the imagery data is badly smeared and feature measurements are difficult to identify. In most other instances, the meter measure option is used.

In the meter measure case the measurements are feature lengths in the image plane, as depicted in the following figure:

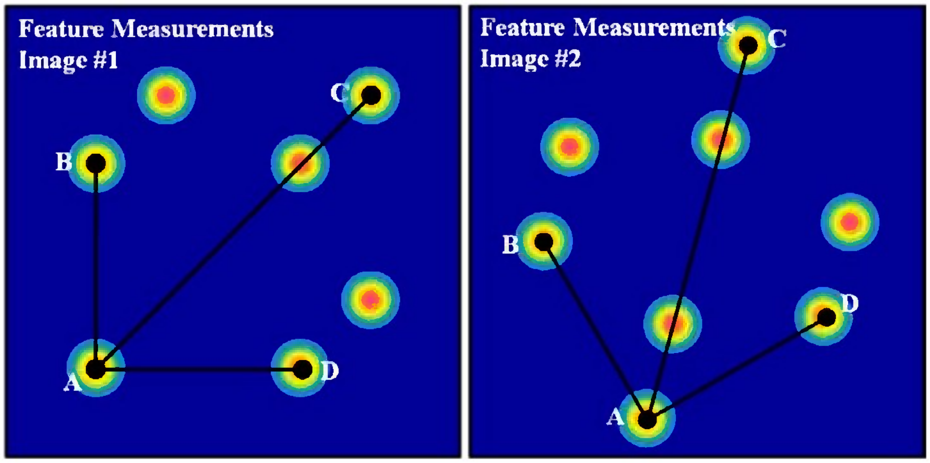


Figure 4.3 Specifying measurement points and features vectors in XELIAS

Meter measure is performed in XELIAS and consists of a set of measurement points placed at distinct locations within the sequence of images. Connecting the point in a specific order, consistently over the entire pass, provides LLMotion with feature measurements (i.e., vector lengths) of the target in terms of the image plane coordinates. These observations are typically made with the aid of the wireframe model of the satellite, which provides both a useful reference for scaling the imagery and acts as a guide for point placement when the satellite features being measured are obscured or difficult to distinguish in the imagery alone.

The residual feature vectors (measured – calculated) are divided by the cross range resolution Δr_x. A single value for Δr_x is used for both the x and y components of all the measured features vectors on all the images, as shown in the cost function equation below (32 pp. A-1):

$$J_r = \frac{1}{N\Delta r_x^2}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\left(\frac{\omega_{m_i}^z}{\omega_{c_i}^z}\mathbf{m}_{ij}^x - \mathbf{c}_{ij}^x\right)^2 + \left(\mathbf{m}_{ij}^y - \mathbf{c}_{ij}^y\right)^2 \qquad (4.7)$$

In equation (4.7), $\mathbf{m}_{ij}$ and $\mathbf{c}_{ij}$ are the measured and calculated j^{th} feature vector of the i^{th} image; $\omega_{m_i}^z$ and $\omega_{c_i}^z$ are the z components of the angular velocity values for the i^{th} image reference to the measurements and calculated for the new motion, respectively; n is the total number of images; m_i the total number of features for image i; and $N = \sum_{i=1}^{n} m_i$. Calculated feature vectors are determined by varying the motion parameters and combining them with the state vector to obtain a 3×3 transformation matrix $\boldsymbol{T}$ for each image which transforms a feature vector $\mathbf{r}$ from body coordinates to radar image plane coordinates $\left(\mathbf{c}_{ij} = \boldsymbol{T}_i \mathbf{r}_j\right)$. In LLMotion, the transformation matrix is calculated only for a dynamic RLOS, corresponding to varying azimuth and elevation angle during the integration interval. The radar tracking of a moving target is the dynamic RLOS case. The rather lengthy sequence of rotations needed to calculate $\boldsymbol{T}$ are given in Appendix B.

Conversely, in the angular velocity case the measurements are the angular velocity values used to rescale each image. Accordingly, $\omega_{m_i}^{z}$ can be computed by taking the angular velocity used in generating the images nearest the reference time and modifying it according the estimated cross-range scaling error in the image $\left(\omega_{m_i}^{z} = \omega_{base}^{z}/s\right)$. The scaling measurements are taken using the interactive scaling option in XELIAS and give a different instantaneous angular velocity for each image. These measurements are used in LLMotion to calculate the corresponding objective function defined by (32 pp. A-3)

$$J_\omega = \sum_{i=1}^{n} \left(\frac{\omega_{m_i}^{z} - \omega_{c_i}^{z}}{\omega_{m_i}^{z}} \right)^2 \tag{4.8}$$

In this case, all the motion parameters are held constant except α and δ for the precession vector and period. The vector $\boldsymbol{\omega}$, of which $\omega_{c_i}^{z}$ is the third component, is calculated from the motion parameters and the state vector data. The calculations are explained in Appendix B. Finally, for the sake of completeness, the root mean square (rms) equations for both cases can be summarized as (32 pp. A-2 - A-3)

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\left(\frac{\omega_{m_i}^{z}}{\omega_{c_i}^{z}}\mathbf{m}_{ij}^{x} - \mathbf{c}_{ij}^{x}\right)^2 + \left(\mathbf{m}_{ij}^{y} - \mathbf{c}_{ij}^{y}\right)^2} \tag{4.9}$$

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\omega_{m_i}^{z} - \omega_{c_i}^{z}\right)^2} \tag{4.10}$$

Minimization of the cost functions can be found in one of two ways – by means of a calculated grid search or a dynamic hill climbing global/local search. In either case the user can vary any of the seven primary motion parameters and also specify the range for each motion parameter. Although, in reality the angles are bound, LLMotion does not impose these constraints on the user. Instead after a minimum is found, if it is out of normal range, it is converted back to the correct quadrant. The calculate grid function allows the analyst to investigate the entire solution space by calculating the cost function over a user defined motion parameter grid where the user specifies the range and step size for each motion parameter. The minimum within this grid are the possible solutions. After the entire objective function space is calculated, a simple search is done to find all the minima within the space. The search tests the nearest and next nearest neighbors of a given point to identify all local minima. These minima are then improved and the angles are converted to the correct quadrant if necessary.

According to Reference (32), the implementation of a search algorithm in LLMotion was motivated by shortcomings in the gradient descent method. As discussed in Section 3.2.2.1 the guarded Gauss-Newton technique was also developed in response to the deficiencies intrinsic to gradient descent, namely, problems with convergence to local minima and ill-conditioning of the matrix inversion. Dynamic hill climbing is similar in nature to the guarded Gauss-Newton process used in the BSEKF. As described in References (32) and (50), the full DHC algorithm consists of local and global search components. The local search (inner loop) consists of a series of tests with a step vector which is doubled if the cost function decreases and is halved otherwise. This feature allows the algorithm to "climb" out of poor locations and rapidly converge to a unimodal minima. The direction of the step vector is determined by the sum of the two previous successful moves, a feature with encourages movement towards a successful solution. The inner loop has converged if the step vector size is reduced below a user-defined threshold. The step size and direction adaptation in DHC are designed to mimic the features of gradient descent without the problems of local minima and matrix conditioning. Dynamic hill climbing is "designed to efficiently incorporate the random search capabilities of simulated annealing, optimum step size in gradient descent, and global search capability of genetic algorithms into a single algorithm." (50 p. 10) Similarly, the DHC global search (outer loop) procedure consists of gridding the search space into 2^n points or initial guesses, where n is the number of search parameters. The process then uses each of the grid points as an initial guess to perform a coarse search, which uses fewer iterations and greater step size threshold than the fine grid analysis performed during a DHC local search. Essentially, the outer loop identifies the areas which will most likely contain favorable outcomes. Since several minima are usually found, the results are filtered to eliminate duplicate solutions before being used as the initial guesses for the fine search processing. The search is done repeatedly until there is no improvement in the solution. The output of the DHC local search is filtered in the same manner as the DHC global results to produce the final set of potential motion solutions (32 pp. H-1 - H-2).

An evaluation function in LLMotion allows the analyst to compare the different minima and select the best set of parameters to fit the given imagery data. The analyst can activate one of the minima and display both the imagery and model with that minimum defined as the active motion solution. Although in general the minimum with the smallest objective function value is the best solution this is not always the case. In fact, for most passes, there are two possible solutions that give rise to equally small residuals. During a pass, the RLOS sweeps out a surface that is almost planar. For an object that is rotating about an axis fixed in space, there is a false model and rotation axis pair that are mirror images of the true model and axis in the plane of the RLOS. This false solution gives outline projections that are nearly identical to those given by the true solution and cannot be identified reliably on the basis of the residuals. Because the false model is a mirror image of the true model, any known asymmetry in the object can be used to differentiate between the two solutions. If there is no known asymmetry, the true solution might be selected on the basis of data from multiple passes, or from other information pertaining to the satellite motion or orientation.

4.3 XELIAS/LAES: Multi-Pass Attitude Estimation and Prediction

The Lincoln Attitude Estimation System serves two primary functions: 1) further refine the attitude estimate output from LLMotion and 2) propagate the spacecraft's attitude over time. The goal is to produce high precision attitude estimates by using information about the mass distribution of a satellite and by accurately modeling the external disturbance torques acting on the vehicle over time. One of the reasons for doing this is to assist in the analysis of motion solutions calculated from multiple passes of radar imagery data. Figure 4.4 is a simple conceptual plot showing how one might pick a motion solution for a particular pass using this tool. Generally speaking an LLMotion solution will only be valid over the duration of the pass; thus, LAES attempts to determine the rotation motion of the spacecraft using observations from several passes.

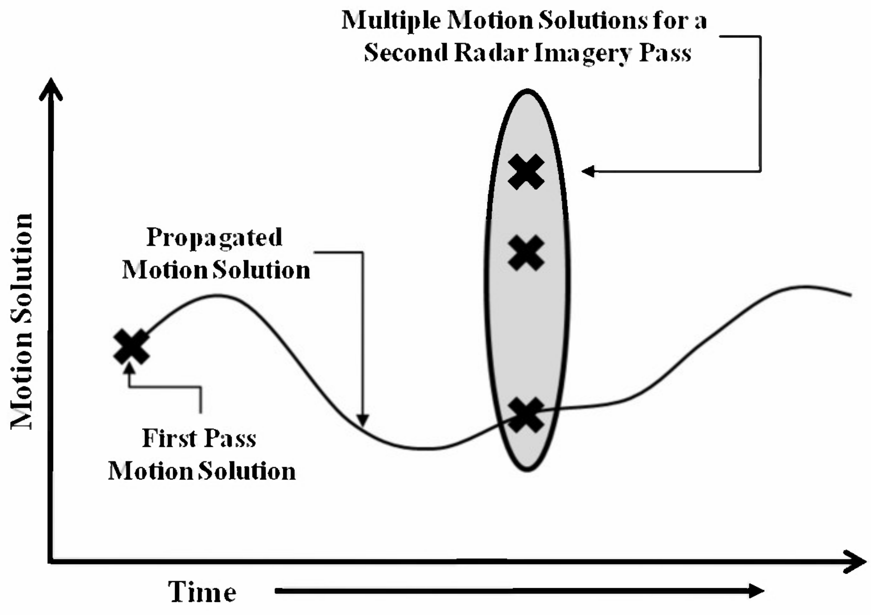

Multiple Motion Solutions for a Second Radar Imagery Pass
Propagated Motion Solution
First Pass Motion Solution
Motion Solution
Time

Figure 4.4 Correlating motion solutions from multiple passes of radar imagery data

An important attribute of the new software system is that it allows the user to view propagated motion solutions over time. An overlay feature is used for viewing two or more propagated motion solutions right on top of one another. For example, turning off the effects of the gravitational torque on the motion of the satellite corresponds to the manner in which LLMotion would propagate the state parameters. With no gravitational torque the right ascension and declination of the angular momentum axis will remain constant.

4.3.1 Measurement Sub-process

The Lincoln Attitude Estimation System is intended to be an add-on program which can be easily integrated into the greater collection motion analysis tools currently in use within the Space Situational Awareness Group. In order to properly interface with legacy systems, LAES must be able to take in the eight symmetric motion parameters used in XELIAS/LLMotion and output a state vector of the same form. However, because the assumptions of zero disturbance torques and spacecraft symmetry are removed in the new program, the attitude of the spacecraft can no longer be characterized in terms of simple spin-precession motion. The filtered state estimate generated by the backward-smoothing extended Kalman filter is, therefore, converted into a unique set of parameters for each image in the pass:

$$\mathbf{x}'_{param(i)} = \begin{bmatrix} t \\ \alpha \\ \delta \\ \theta \\ \phi \\ \psi \\ \dot{\psi}^{-1} \\ \dot{\phi}^{-1} \end{bmatrix}_i \tag{4.11}$$

Because the parameterization is only valid for each discrete observation time t_i, x-initial ϕ_0 and z-initial ψ_0 are replaced by the instantaneous precession angle ϕ_i and spin angle ψ_i, respectively.

An understanding how $\mathbf{x}'_{param(i)}$ is computed, begins with an assessment of the output from LLMotion. The refined set of motion parameters that the program produced can be passed to ARIES for use in reimaging the pass. In addition to focusing and scaling the images in the cross-range direction, the improved estimate of the rotational motion enables the imaging interval to be recalculated (based on the change in aspect angle) and thus, more images to be produced. However, the motion solution may still contain a considerable amount of noise. As a result, additional measurements of the apparent "true" attitude of the spacecraft need to be added to the new baseline motion before the filtering sub-process in LAES can begin. To assist in processing the collected data, the Space Situational Awareness Group has developed a comprehensive set of modeling and advanced radar image analysis techniques which have again been implemented in the software program known as the X-Based Enhanced Lincoln Interactive Analysis System. In addition to image manipulation features, XELIAS provides a complete computer-aided tool for creating or modifying an existing geometric model of a satellite of interest (4). If a "wireframe" model of the target is available, then it can be displayed on top of the collected imagery for analysis. Measurements of a spacecraft's attitude, relative to the initial guess, i.e., the baseline motion used to regenerate the images, are made through the manual alignment of the wireframe model with each of the underlying radar images. This image-model matching process is, as described in Section 3.1.3, performed in XELIAS through the translation, rotation, and scaling functions provided as part of the software (4). The figure below depicts how each discrete measurement is made by aligning a wireframe model (depicted on the right) with the underlying radar image (depicted on the left). The radar observations are composed of range and cross-range returns at different intensity levels, which are denoted using an artificial color band in which stronger signals appear as warmer colors. It should also be noted the images below depict a fictitious satellite and are caricatures of the true wireframe models and radar observations used in the measurement process (created in Microsoft PowerPoint).

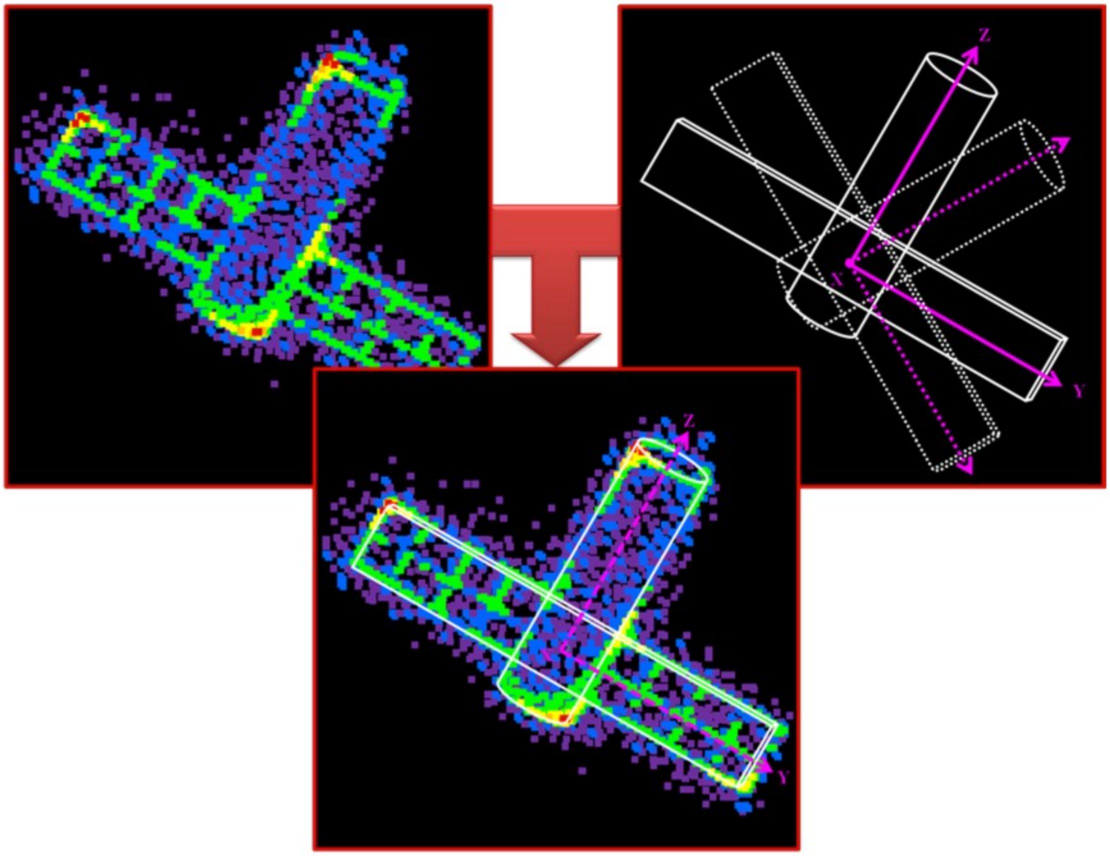


Figure 4.5 Alignment of a wireframe model (right) with an underlying radar image (left). The set of rotations that parameterize the attitude deviations between the baseline motion solution and perceived true attitude are recorded by XELIAS as a set of Euler angles.

The rotations (in degrees) that are needed to transform the wireframe model from the baseline set of body-fixed coordinates, denoted with a lowercase b subscript, into the corrected set of body-fixed coordinates, designated with a capital B subscript, are recorded by XELIAS as a set of Euler angles that can be easily recovered and transferred to Matlab for filtering (saved in the paramcdf file). This set of Euler angles represent the deviations in attitude from the baseline motion solution, and can be expressed as a series of rotations about the z-, y-, and x-axis and applied in that order. The rotations can be written as (6 p. 9):

$$\mathbf{u}_B = \boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)\mathbf{u}_b \Rightarrow \mathbf{u}_B = \boldsymbol{R}_{123}\mathbf{u}_b \tag{4.12}$$

where ϕ, θ, and ψ are the angles of rotation about the x-, y-, and z-axis respectively, $\mathbf{u}$ is an arbitrary column vector, and $\boldsymbol{R}_{123} = \boldsymbol{R}_{align}$ is the combined set of rotation matrices needed in the image-model matching process. Similarly, the sequence of rotations needed to transform a vector $\mathbf{v}$ in baseline body-fixed coordinates b to the inertial coordinate frame I is given by the equation (51 p. 1):

$$\mathbf{v}_I = \boldsymbol{R}_0\boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta - \pi/2)\boldsymbol{R}_z(\psi)\mathbf{v}_b \Rightarrow \mathbf{v}_I = \boldsymbol{R}_{0123}\mathbf{v}_b \tag{4.13}$$

where ϕ is the precession angle, θ is the coning angle, ψ is the spin angle, given by the following set of equations (51 p. 1):

$$\phi = \dot{\phi}(\Delta t) + \phi_0 \tag{4.14}$$

$$\theta = \theta_0 \tag{4.15}$$

$$\psi = \dot{\psi}(\Delta t) + \psi_0 \tag{4.16}$$

$$\Delta t = t_i - t_0 \tag{4.17}$$

In the relationships expressed above, the spin rate $\dot{\psi}$ and precession rate $\dot{\phi}$ can found by simply inverting the two periods given in the baseline parameterization; $\boldsymbol{R}_{0123} = \boldsymbol{A}_{base}$ and is the combined set of rotation matrices that makeup the nominal attitude; and $\boldsymbol{R}_0$, can be written as (51 p. 2):

$$\boldsymbol{R}_0 = \begin{bmatrix} \cos(\alpha)\cos(\delta) & -\cos(\alpha)\sin(\delta) & \sin(\alpha) \\ \sin(\alpha)\cos(\delta) & -\sin(\alpha)\sin(\delta) & -\cos(\alpha) \\ \sin(\delta) & \cos(\delta) & 0 \end{bmatrix} \tag{4.18}$$

where α is the right ascension and δ is the declination of the precession vector. The normalized angular momentum vector for the reference motion solution can be obtained directly from $\boldsymbol{R}_0$:

$$\hat{\boldsymbol{L}}_{base} = \boldsymbol{R}_0(:,1) \tag{4.19}$$

and used in combination with the last column of the baseline rotation matrix to find the angular velocity (27 p. 18):

$$\boldsymbol{\omega}_{base} = \dot{\phi}\boldsymbol{L}_{base} + \dot{\psi}\boldsymbol{A}_{base}(:,3) \tag{4.20}$$

Other useful relations involving the spin and precession rates ($\dot{\psi}$ and $\dot{\phi}$), the moment of inertia d_i, coning angle θ, and angular momentum magnitude $|L|$, include (27 p. 18):

$$\dot{\psi} = \frac{I_x - I_z}{I_x I_z}|L|\cos(\theta) \tag{4.21}$$

$$\dot{\phi} = \frac{|L|}{I_x} \tag{4.22}$$

Expressing the single moment of inertia parameter in terms of the spin-precession motion parameters can be accomplished by means of the following equation (27 p. 18):

$$I_z = \frac{\dot{\phi}\cos(\theta)}{\dot{\phi}\cos(\theta) + \dot{\psi}} \tag{4.23}$$

The total set of rotations $\boldsymbol{A}$ and corresponding quaternion, which ultimately comprises the measurement vector $\mathbf{y}$, can be calculated using:

$$[\mathbf{w}]_B = \boldsymbol{R}_{align}\boldsymbol{A}_{base}^T[\mathbf{w}]_I \Rightarrow [\mathbf{w}]_B = \boldsymbol{A}[\mathbf{w}]_I \tag{4.24}$$

Equation (4.24) expresses the transformation of a vector $\mathbf{w}$ from the inertial frame directly to the corrected body-fixed coordinate frame. Though the formulas needed to convert from a rotation matrix $\boldsymbol{A}$ to an attitude quaternion $\boldsymbol{q}$ are relatively straightforward, they are rather lengthy and are, therefore, saved for Appendix A. The result of applying equations $(\mathrm{A}.10)$ through $(\mathrm{A}.13)$ to the combined rotation matrix $\boldsymbol{A}$, is the measurement vector used in the filtering sub-process, which has the form:

$$\mathbf{y}=\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}=\begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \tag{4.25}$$

4.3.2 Filtering Sub-Process

The state vector used in the Lincoln Attitude Estimation System is of dimension 12×1 ($\mathbf{x}=[q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, p_{I1}, \dots, p_{I6}]^T$) and is comprised of the vector components of the attitude quaternion (the fourth term - the redundant scalar component - has been removed due to the singularity that results in the state transition and covariance matrices), the attitude rate vector (angular velocity), and the six moment of inertia matrix parameters as presented in Mark Pasaki's formulation for the BSEKF. The moment of inertia parameters can be used to construct the moment of inertia tensor $\boldsymbol{I}_B$, given by (41 p. 890):

$$\boldsymbol{I}_B=[\boldsymbol{A}_1(p_{I4})\boldsymbol{A}_2(p_{I5})\boldsymbol{A}_3(p_{I6})]\begin{bmatrix} \frac{p_{I2}^2+p_{I3}^2}{12} & 0 & 0 \\ 0 & \frac{p_{I1}^2+p_{I3}^2}{12} & 0 \\ 0 & 0 & \frac{p_{I1}^2+p_{I2}^2}{12} \end{bmatrix}[\boldsymbol{A}_3^T(p_{I6})\boldsymbol{A}_2^T(p_{I5})\boldsymbol{A}_1^T(p_{I4})] \tag{4.26}$$

In this formula, $\boldsymbol{A}_n(p)$ is the 3×3 direction cosine matrix for a rotation of p radians about the n^{th} axis; p_{I1}, p_{I2}, and p_{I3} are the lengths of the sides of an equivalent uniform rectangular box multiplied by the square root of its mass, and p_{I4}, p_{I5}, and p_{I6} are the three Euler angles that parameterize the rotation of this box with respect to spacecraft coordinates.

The quaternion parameterization has been chosen for several practical reasons: 1) the prediction equations are treated linearly, 2) the representation is free from singularities, and 3) the attitude matrix is algebraic in the quaternion components (thus eliminating the need for trigonometric functions). However, the use of the quaternion as the attitude state presents some difficulty in the application of the filter equations. This difficulty is due to the lack of independence of the four components, which are related by the constraint that the quaternion have unit norm. This restriction results in the singularity of the covariance matrix $\mathbf{P}$ of the quaternion state and is difficult to maintain numerically due to the accumulation of round-off error. The most straightforward way to circumvent this complication is to represent $\mathbf{P}$ by a matrix of smaller dimension (12×12 instead of the 13×13); the most obvious approach being to simply delete one of the quaternion components. The fourth element has been selected, although in principal any component could be removed. Since the quaternion is not unique the correct sign of the scalar element must be recorded and stored in the m-buffer for use in reconstructing the full 4×1 vector for certain critical algorithm operations (namely, propagating the state to the next observation time and computing the residual). The redundant fourth component can then be easily recovered using the relationship: $q_4 = (\pm 1)\sqrt{1 - q_1^2 - q_2^2 - q_3^2}$, which simultaneously satisfies the constraint on the quaternion. However, it should be noted that the accumulation of numerical round-off error and the linearization approximation inherent in the update equations can distort the calculation of q_4. Though this need not cause concern in the propagation of the quaternion, due to the fact that this operation is linear, care must be taken when calculating the attitude matrix or when an update is to be performed. As a result the constraint on the quaternion should be regularly enforced (typically after numerous operations using the full quaternion or before switching back to the minimal attitude representation) using equation (2.37) from Section 2.1.4.

The filter also utilizes the Euler angle representation as a means of visualizing the attitude and residuals, and removing discontinuities which arise in the measurements and state estimates. The discontinuities manifest themselves as sign changes in the attitude quaternion, as depicted in Figure 4.6, and correspond to a set of Euler angles passing through $\pm\pi$. Because the components of the quaternion regularly pass through zero, it is more difficult to determine the proper sign that each should have, than it is to establish the appropriate Euler angle. If uncorrected, these artificial jumps in the data can lead to wildly inaccurate residuals, which in turn disrupt the cost function.

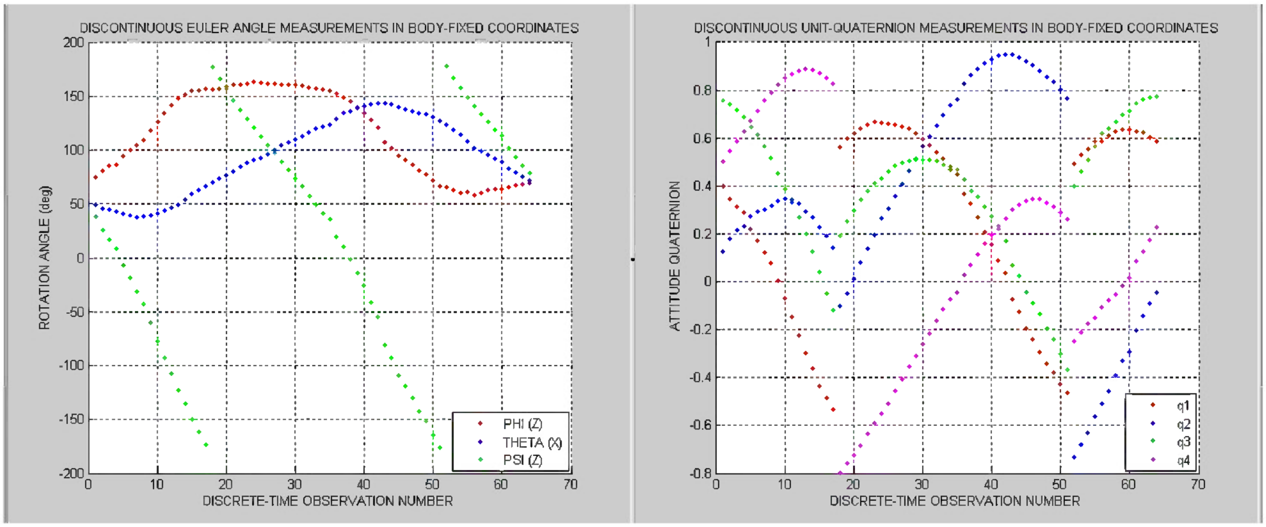


Figure 4.6 Discontinuities in the quaternion and equivalent Euler angle measurements

The procedure for removing discontinuities in the data is as follows:

1. Convert the attitude quaternion to a set of 3-1-3 Euler angles using the equation $(A.7)$
2. Compare and modify the current Euler angle estimate based on the previous value

$$\phi_k = \phi_k + \left(round\left(\frac{\phi_{k-1} - \phi_k}{2\pi}\right) * 2\pi\right) \tag{4.27}$$

$$\theta_k = \theta_k + \left(round\left(\frac{\theta_{k-1} - \theta_k}{2\pi}\right) * 2\pi\right) \tag{4.28}$$

$$\psi_k = \psi_k + \left(round\left(\frac{\psi_{k-1} - \psi_k}{2\pi}\right) * 2\pi\right) \tag{4.29}$$

3. Convert the Euler angles back into an attitude quaternion using equation $(\mathrm{A}.4)$

Reasonable initial values for each of the state components can then be obtained in the following manner. As was mentioned in Section 3.2.5.1, the preliminary estimate for attitude quaternion is simply the first three elements of the initial measurement vector $(\mathbf{y}_0)$. The angular velocity vector, leverages the work done to remove the discontinuities, by plotting the Euler angles about each axis over time and performing a linear least squares fit to the first n observations. To determine the angular rates, the transformation into a body-referenced angular velocity is done using the equation for a 3-1-3 Euler angle sequence (2 p. 256):

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}_B = \begin{bmatrix}0 & \cos(\psi) & \sin(\theta)\sin(\psi)\\ 0 & -\sin(\psi) & \sin(\theta)\cos(\psi)\\ 1 & 0 & \cos(\theta)\end{bmatrix}\begin{bmatrix}\dot{\psi}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} \tag{4.30}$$

An alternative method would be to use the scale factor s, measured in the image-model matching process, and saved by XELIAS in the paramcdf file, to correct the baseline angular velocity terms given by equation (4.20). The process and equation needed to do so are given in Appendix B.The wireframe model used in making the attitude measurements is also used in computing an initial estimate for the inertia tensor of the satellite, by first decomposing the model into a set of matrices consisting of the vertex coordinates and the order in which they should be connected to create planar faces. This information can be obtained from the *iges* file used by XELIAS, which contains a description of the three-dimensional structure of the satellite for computer graphics rendering and software manipulation. The polyhedral mass properties algorithm described in Section 2.3.4.2 is then used to compute the mass properties for a solid, simple polyhedron of constant mass density. Finally, the initial guesses for the moment of inertia parameters are calculated through singular value decomposition and back solving for the components which correspond to the inertia matrix generated using the wireframe model.

It is important to note that the uncertainties inherent in using a purely geometric model of the spacecraft are compensated for by augmenting the state to include the six parameters that determine the moment of inertia matrix. Estimating the inertia matrix parameters is done to refine the Euler dynamics model and thus improve the estimates of the attitude and rates. The overall scaling of the inertia matrix is unobservable, but it does not need to be estimated because the scaling does not affect the system's rotational motion (41 p. 890). What matters is the relative ratio between each of the elements of the moment of inertia. This is accounted for by assigning a very small priory variance to one the inertia matrix magnitude parameters. This acts as a soft constraint on the scaling.

This implementation of the BSEKF also incorporates the effects due to the gravity-gradient, magnetic field, aerodynamic, and solar radiation torques. The equations and definition of terms that were uses are described in Section 2.3.4.3. The input values needed to execute these models were obtained from reports generated using Satellite Tool Kit (STK), the MSISE-2000 atmospheric model, the International Geomagnetic Reference Field (IGRF), and the online ephemeris computation service provided by the Jet Propulsion Laboratory's HORIZONS Database (23) (52) (53) (54) (55). Since the relative importance of each environmental torque is a function of the spacecraft's size, geometry, mass, mass distribution, and altitude, each model occupies a separate module that can be turned on or off depending on the unique characteristics of each spacecraft being analyzed. Additionally, the modular design enables the effects of each torque on the overall motion solution to be systematically tested. The figure below depicts the list of input files used by each torque model and their source.

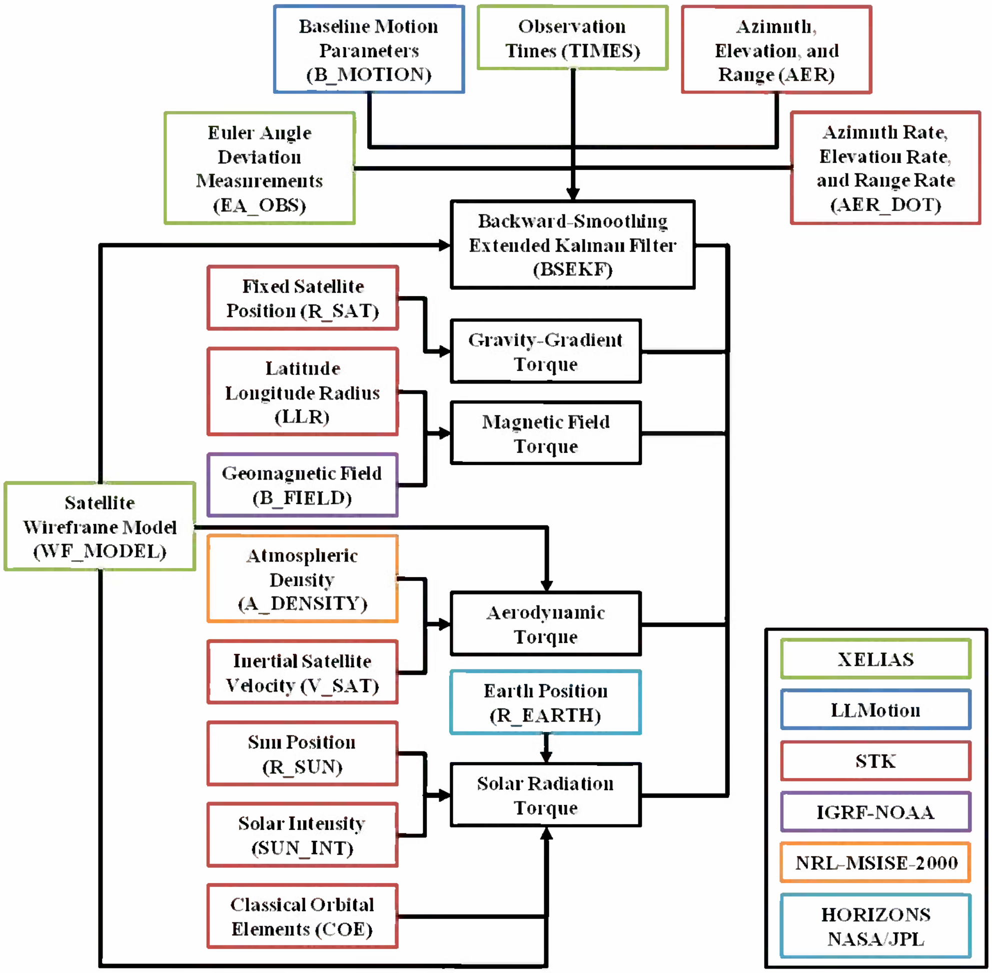

Baseline Motion Parameters (B_MOTION)
Observation Times (TIMES)
Azimuth, Elevation, and Range (AER)
Euler Angle Deviation Measurements (EA_OBS)
Azimuth Rate, Elevation Rate, and Range Rate (AER_DOT)
Backward-Smoothing Extended Kalman Filter (BSEKF)
Fixed Satellite Position (R_SAT)
Gravity-Gradient Torque
Latitude Longitude Radius (LLR)
Magnetic Field Torque
Geomagnetic Field (B_FIELD)
Satellite Wireframe Model (WF_MODEL)
Atmospheric Density (A_DENSITY)
Aerodynamic Torque
Inertial Satellite Velocity (V_SAT)
Earth Position (R_EARTH)
Sun Position (R_SUN)
Solar Radiation Torque
Solar Intensity (SUN_INT)
Classical Orbital Elements (COE)
XELIAS
LLMotion
STK
IGRF-NOAA
NRL-MSISE-2000
HORIZONS NASA/JPL

Figure 4.7 Input files needed in the backward-smoothing extended Kalman filter, inertia tensor model, and various torque models

The dynamics function $f_i(\mathbf{x}_i, \mathbf{w}_i)$ for this system uses the kinematics equation for the attitude quaternion (41 p. 890)

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} q \tag{4.31}$$

together with Euler's equation for the attitude dynamics (41 p. 890)

$$\dot{\omega} = I_B^{-1}[-\omega_B \times (I_B\omega_B) + N_B + A(q)\mathbf{w}_i] \tag{4.32}$$

to propagate the state from sample time t_i to sample time t_{i+1}. The term N_B on the right-hand side of equation (4.32) is the combined environmental torque vector in spacecraft coordinates.

The expression involving $\mathbf{w}_i$ implies that the process noise vector is a 3×1 disturbance torque that is measured relative to inertial coordinates, and is constant during the sample interval. Numerically integrating the analytic equations of motion and the partial derivatives for the state and process noise vectors produces the attitude quaternion (elements 1 – 4 of $\mathbf{f}_i$) and angular velocity (elements 5 - 7 of $\mathbf{f}_i$) vectors at time t_{i+1}, and linearized dynamics matrices, $\mathbf{\Phi}_i = \partial \mathbf{f}_i / \partial \mathbf{x}_i$ and $\mathbf{\Gamma}_i = \partial \mathbf{f}_i / \partial \mathbf{w}_i$. The remaining six elements of $\mathbf{f}_i$ characterize the dynamics of the inertia matrix parameters. Since these parameters are modeled as being constants, $(f_i)_l = (x_i)_l$ for $l = 7, \ldots, 12$. Because the state vector $\mathbf{x}_{i+1}$ must be of dimension 12×1, the fourth term of the discrete-time dynamics function output vector $\mathbf{f}_i$ is removed after noting the sign. Similarly, the fourth row and column of the state and process noise transition matrices $\mathbf{\Phi}_\mathrm{i}$ and $\mathbf{\Gamma}_\mathrm{i}$ are deleted so that they are both 12×12. The measurement function $h_i(\mathbf{x}_i)$ represents a conversion from the reduced attitude representation to the full quaternion and takes the following simple form:

$$h_i(\mathbf{x}_i) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ (\pm 1)\sqrt{1 - x_1^2 - x_2^2 - x_3^2} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4.33}$$

Taking the partial derivatives of the measurement function with respect to the quaternion components of the state vector results in the 4×12 linearized measurement matrix

$$\mathbf{H}_i = \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_i} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times9} \end{bmatrix} \tag{4.34}$$

Because the constraint of the quaternion must be preserved in order to be a true representation of the attitude of the spacecraft, care must be taken when operating on state vector, specifically, adding or subtracting values from the quaternion components. Accordingly, the following equations in the backward-smoothing extended Kalman filter have been amended to ensure that changes being applied to the quaternion components are themselves incremental rotations

- From BSEKF step 4:

$$\hat{\mathbf{x}}^{*}_{k-m} = \hat{\mathbf{x}}_{k-m} + \mathbf{R}^{-1}_{xx(k-m)}\Delta\mathbf{z}_{x(k-m)} \tag{4.35}$$

- From BSEKF step 12:

$$\Delta\mathbf{z}_{\mathrm{x}(i)} = \mathbf{R}_{xx(k-m)}\left(\hat{\mathbf{x}}^{*}_{k-m} - \mathbf{x}^{j}_{k-m}\right) \tag{4.36}$$

- From BSEKF step 20:

$$\mathbf{x}^{j+1}_{k-m} = \mathbf{x}^{j}_{k-m} + \gamma\Delta\mathbf{x}_{k-m} \tag{4.37}$$

In equations (4.35) – (4.37), + and – operators are, correspondingly, replaced with the composition rule, given by equation (2.47), and difference rule, given by equation (2.48), from Section 2.1.4. It should be noted that the full 4×1 vector $\boldsymbol{q}$ must be reconstructed in order to implement the procedure above, and that these changes only apply to the first three components of the state vector, i.e., those related to the attitude quaternion.

Similarly, the quaternion error in this representation can also be expressed not as the arithmetic difference between the measured and the estimated quaternion, but as the quaternion which must be composed with the estimated quaternion in order to obtain the true attitude. Since this incremental quaternion corresponds almost certainly to a small rotation, the fourth component will be close to unity and, hence, all the attitude information of interest is contained in the three vector components. Therefore, the twelve part state vector also provides a non-redundant representation of the state error. The error quaternion is defined as (56 p. 424)

$$\delta\boldsymbol{q}_{i+1} = \boldsymbol{q}_{meas}\boldsymbol{q}^{-1}_{calc} = \mathbf{y}_{i+1}\left[h_{i+1}\left(\mathbf{x}^{j}_{i+1}\right)^{-1}\right] \text{ for } i = k-m, \dots, k-1 \tag{4.38}$$

and can also be expressed in matrix form using equation (2.48). Consequently, equation (4.38) can be used in place of the measurement error vector $\Delta\mathbf{y}_{i+1}$ and within the cost function, given by equation (3.101) from BSEKF step 11.

In addition to the modifications and added functions detailed in the section thus far, the following post processing operations are needed to convert the final state estimates into a set of motion parameters $\mathbf{x}'_{param(i)}$ that are consistent with what is used by the existing Group 93 software systems. Picking up where Section 3.2.5.1 left off, insert the subsequent steps:

25. Once the backward-smoothing extended Kalman filter has processed the last observation in the pass, that is, $k = n$ and all the termination criteria have been satisfied, propagate the final state estimate $\hat{\mathbf{x}}_k$ to all the times of interest. Again, propagation of the state vector is accomplished by numerical integrating equations (4.31) and (4.32), as described in Section 2.3.4.1. In practice this means using the Matlab function *ode45* to backwards propagate $\hat{\mathbf{x}}_k$ to each discrete-time observation in the pass.
26. Calculate and plot the residuals between the measured attitude $\mathbf{y}$ and final, i.e., filtered, state estimates $h(\mathbf{x})$, using equations (4.38) and (3.99). The state error covariance over time should also be plotted for the quaternion, angular velocity, and inertia components. As will be discussed in the results section, these are the primary quantitative measures of accuracy used in assessing filter performance and a given motion solution.
27. Compute the attitude rotation matrix $\boldsymbol{A}_i$ and inertia tensor $\boldsymbol{I}_B$ using equations (A.5) and (4.26) respectively. Determine the angular momentum vector in spacecraft coordinates at each observation time in the pass using Euler's formula: $(\boldsymbol{L}_i)_B = \boldsymbol{I}_B(\boldsymbol{\omega}_i)_B$. Using the attitude matrices, transform the angular momentum and angular velocity vectors into the ECI coordinate frame by means of the following simple formulas:

$$(\boldsymbol{\omega}_i)_I = \boldsymbol{A}_i^T(\boldsymbol{\omega}_i)_B \text{ and } (\boldsymbol{L}_i)_I = \boldsymbol{A}_i^T(\boldsymbol{L}_i)_B \tag{4.39}$$

28. Calculate the right ascension, declination, and coning angle using the angular momentum vector $\boldsymbol{L}$ and spin axis of the spacecraft $\mathbf{z}_B = [0 \quad 0 \quad 1]^T$.

$$\alpha_i = \text{atan2}\left(\frac{(L_2)_i}{(L_1)_i}\right)_I \tag{4.40}$$

$$\delta_i = \text{atan2}\left(\frac{(L_3)_i}{\sqrt{(L_1)_i^2 + (L_2)_i^2}}\right)_I \tag{4.41}$$

$$\theta_i = \text{atan2}\left(\frac{(\boldsymbol{L}_i)_B \times \mathbf{z}_B}{(\boldsymbol{L}_i)_B \cdot \mathbf{z}_B}\right) \tag{4.42}$$

29. The transformation from inertial coordinates to the spin/precession coordinate frame is computed using the right ascension and declination angles:

$$\boldsymbol{R}_{23} = \boldsymbol{R}_y(\pi/2 - \delta_i)\boldsymbol{R}_z(\alpha_i) = \begin{bmatrix} \sin(\delta_i) & 0 & -\cos(\delta_i) \\ 0 & 1 & 0 \\ \cos(\delta_i) & 0 & \sin(\delta_i) \end{bmatrix} \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & 0 \\ -\sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sin(\delta_i)\cos(\alpha_i) & \sin(\delta_i)\sin(\alpha_i) & -\cos(\delta_i) \\ -\sin(\alpha_i) & \cos(\alpha_i) & 0 \\ \cos(\delta_i)\cos(\alpha_i) & \cos(\delta_i)\sin(\alpha_i) & \sin(\delta_i) \end{bmatrix} \tag{4.43}$$

The rotation matrix which takes a vector from body coordinates directly into the spin/precession system, denoted by the subscript SP, is given by

$$[\mathbf{v}]_{SP} = \boldsymbol{R}_{23}\boldsymbol{A}_i^T[\mathbf{v}]_B \Rightarrow [\mathbf{v}]_{SP} = \boldsymbol{R}'_{base}[\mathbf{v}]_B \tag{4.44}$$

30. Once the new baseline rotation matrices $\boldsymbol{R}'_{base}$ have been converted into n set of 3-1-3 Euler angles using equation (A. 9), extract the spin and precession angles (ψ and ϕ). If the angles are greater than zero, then:

$$\psi_i = \psi_i - \pi \tag{4.45}$$

and

$$\phi_i = \phi_i - \frac{\pi}{2} \tag{4.46}$$

However, if the angles are less than zero, then 2π needs to be added to equations (4.45) and (4.46). One should also ensure that the angle θ obtained from the conversion to Euler angles is equivalent to the coning angle found in equation (4.42).

31. Computation of the angular rates is accomplished using equation (A.19), after first transforming the angular velocity vector from body coordinates to the spin/precession coordinate frame my means of the following equation:

$$(\boldsymbol{\omega}_i)_{SP} = \boldsymbol{R}'_{base}(\boldsymbol{\omega}_i)_B \tag{4.47}$$

    Inverting the precession rates $\dot{\phi}_i$ and spin rates $\dot{\psi}_i$ yields the corresponding precession and spin period parameters.

32. Once the new motion parameters have all been converted into the proper units (degrees and seconds) $\mathbf{x}'_{param(i)}$ is ready to be transferred to Interactive Motion for qualitative analysis.

33. If the estimates are deemed acceptable (i.e., quantitative and qualitative measures of accuracy suggest that the true rotational motion of the vehicle has been properly determined) then the final state vector can be propagated forwards in time to the first observation in next pass for further analysis.

For more specific information on how the filter has been implemented, please refer to the Matlab source code provided in Appendix D.

[This page intentionally left blank.]

5 Results

The aim of this chapter is to presents the results of a truth-model simulation and set of real-life test cases which have been used to assess the performance of the backward-smoothing extended Kalman filter. Given a lack of real-world truth data for the non-cooperative attitude estimation problem, it has become necessary to generate attitude measurements using the algorithm's own dynamics equations and torque models. In addition to being a vital tool for properly tuning the filter, the simulated test cases are intended to demonstrate the basic functionality and overall operability of the algorithm. The actual test cases are then meant to investigate the BSEKF's accuracy and convergence reliability under a wide variety of very challenging circumstances. The results presented in Section 5.3 are divided into short- and long-term test cases, depending on whether the attitude estimation and/or prediction was made using observations from a single pass or multiple discrete passes separated by an extended period of time ($t > 90$ minutes). The results show that the BSEKF is able to 1) accurately and quickly converge on a motion solution for individual data passes, 2) overcome large initial errors, 3) filter over long time gaps separating sequential passes, and 4) better predict the attitude of spacecraft than existing motion analysis software.

Throughout the chapter, observations are plotted with respect to a sequential numbering system rather than the specific time at which some event occurred or the elapsed time from some reference event. This was done for the sake of consistency, simplicity, and space. Figure 5.1 shows an example set of attitude measurements plotted against time on the right and by discrete observation number on the left. The false discontinuities seen in the left-hand graphs are a direct result of equally spacing the data points and become more pronounced as the time between consecutive values becomes larger. However, when working with dataset from multiple passes plotting the observations over time results in an enormous blank space at the center of the graph. This gap in the data corresponds to the long time span in between the two passes and effectively pushes all the relevant information towards the edges of the graph, making it very difficult to read.

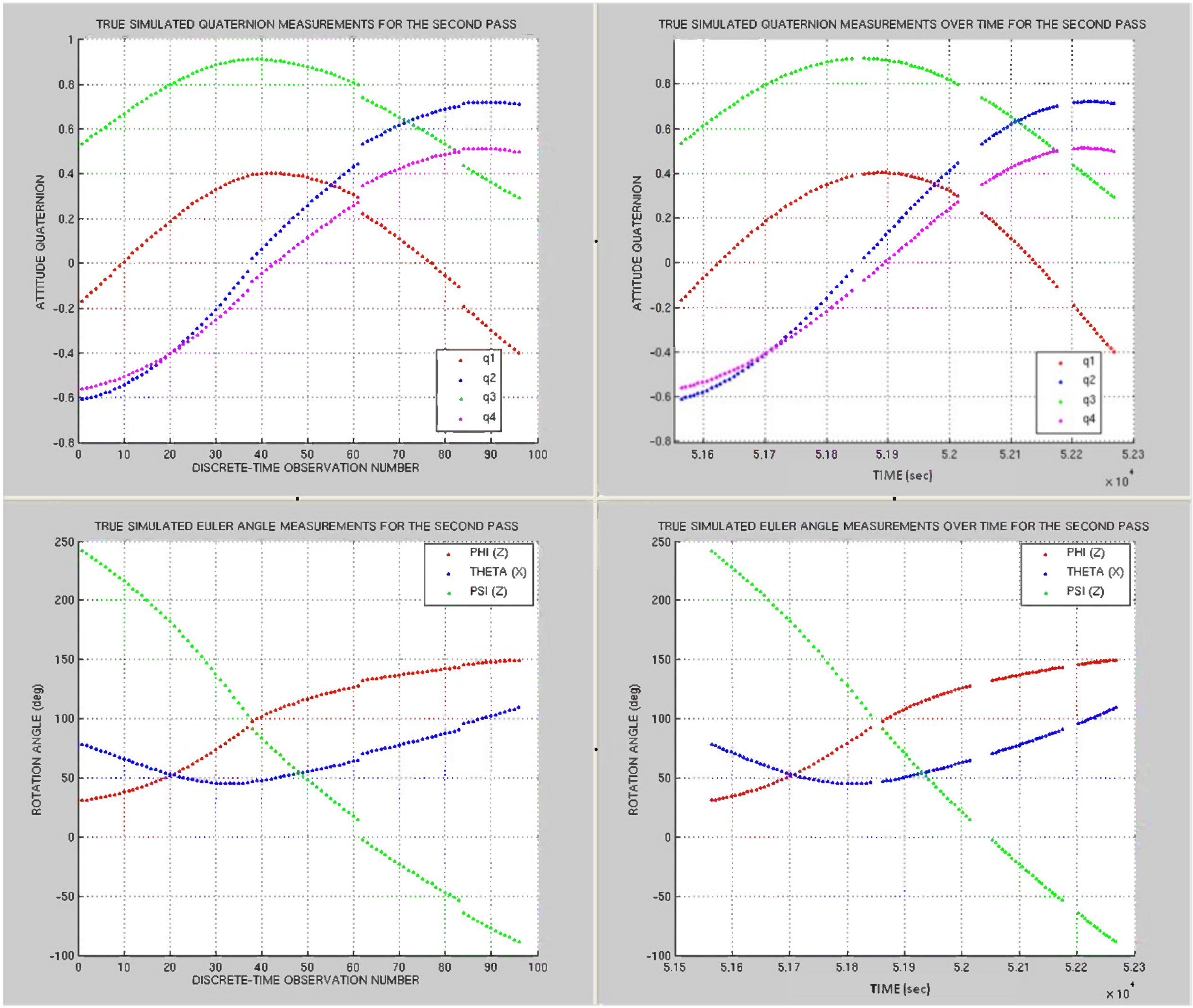


Figure 5.1 Example plot of the quaternion (top) and Euler angle (bottom) measurements referenced by observation number (left) and time (right)

It should also be noted that even though the unit-quaternion is used extensively throughout the BSEKF, for the purposes of visualizing and evaluating results, the Euler angle representation it the better choice, since it is generally more intuitive and conceptually meaningful than the quaternion. While both types of plots are provided for side-by-side comparison, the analysis in Section 5.3 typically focuses on and makes reference to the Euler angle version a given set of figures. Ultimately, it is much easier to speak in terms of degrees, as opposed to radians or the unitless values of the unit-quaternion. Additionally, while Section 4.3.2 stressed the importance of the error quaternion in determining the accuracy of an attitude estimate, plots of this particular performance measure are only presented in the real-life tests cases. Accordingly, the straight residuals between measured and calculated values provide the primary means by which to depict and assess the filter's functionality. Finally, in the context of this estimation problem, a measure of total attitude error is obtained by calculating the magnitude of a vector containing all the component errors (residual values) in a given attitude estimate.

5.1 Spacecraft Attitude Estimation Problem – Actual Test Case

The performance of the Lincoln Attitude Estimation System has been assessed using several real-world test cases, which utilize observations of an actual inactive satellite. The raw radar returns were collected using the Haystack Auxiliary (HAX) wideband imaging radar on 24 September 2003 and archived/stored in a location accessible from the data handling menu of XELIAS. The spacecraft of interest is at an altitude of 1,650 km and is in a slightly elliptical orbit, with an eccentricity of 0.0195 and inclination of 63.45 deg. The satellite and datasets selected for analysis were chosen for a number of reasons, to include:

1. The spacecraft is sufficiently large to give good detail to the signature.
2. The spacecraft has a number of distinguishing features which produce strong radar returns and aid in uniquely determining the orientation of the vehicle during the image-model matching process. In general, solar panels, booms, trusses, antennas, and other structures which tend to protrude from the main body, and break the symmetry of the

spacecraft, make the measurement process much easier. However, these same appendages and irregular features tend to complicate the dynamics of the system, especially when it comes to estimating the mass distribution of the spacecraft and trying to account for the effects of environmental torques acting on the body.

3. A detailed wireframe model was available for the satellite. Creating new computer graphics models in XELIAS can be a rather challenging and time consuming task, depending on the complexity of the target being analyzed.
4. According to the XELIAS data archive, the spacecraft became unstable in August of 2003.
5. Three consecutive passes of imaging data were available in the archive for the same day, with a two hour gap between the first and second pass and a four hour gap between the second and third pass. Since the orbital period is roughly two hours, the three datasets are not from consecutive revolutions, though data was collected every time the satellite was in view of the ground sensor.
6. The observation periods for the three datasets are all reasonably long. The first and third passes are both approximately 22 minutes in length, while the second is roughly half that, at about 12 minutes.

The datasets used in this thesis include 64 discrete-time images for the first pass and 96 images for the second pass. While this may seem illogical/counter intuitive given that the first pass is almost twice as long as the second, it is important to remember that the number of images generated by ARIES is more a function of the change in aspect angle of the satellite relative to the radar line of sight and amount of overlap between independent images. It should also be noted that, due to time constraints, the third pass has not been included in the current round of testing, and will, therefore, be presented as future work item in the next chapter.

As in the previous chapter, the baseline solution used to produce the images assumes torque-free motion of a symmetric rigid body and describes rotational motion in terms of the eight parameters listed in Table 5-1. It is important to note that in the absence of perturbations the values for the second pass are the same as the first, with the exception of the angles ϕ_0 and ψ_0, which correspond to the center time of the first image in the dataset.

Table 5-1 XELIAS/LLMotion Spin/Precession Motion Parameters

Variable Description	Symbol	Value for Pass #1	Value for Pass #2
Reference Time	t_0	43450	51564
Right Ascension	α	23.503	23.503
Declination	δ	-2.842	-2.842
Coning Angle	θ	49.572	49.572
X-Initial	ϕ_0	-76.242	32.5635
Z-Initial	ψ_0	49.556	36.5508
Spin Period	$\dot{\psi}^{-1}$	1501.2959	1501.2959
Precession Period	$\dot{\phi}^{-1}$	1396.78	1396.78

These parameters can be used to calculate and plot the angular momentum, angular velocity, and spin axis at each discrete observation time. Doing so yields the following plots (Figure 5.2) for the nominal motion of the spacecraft in both the inertial frame and body-fixed coordinate frames, for the first pass.

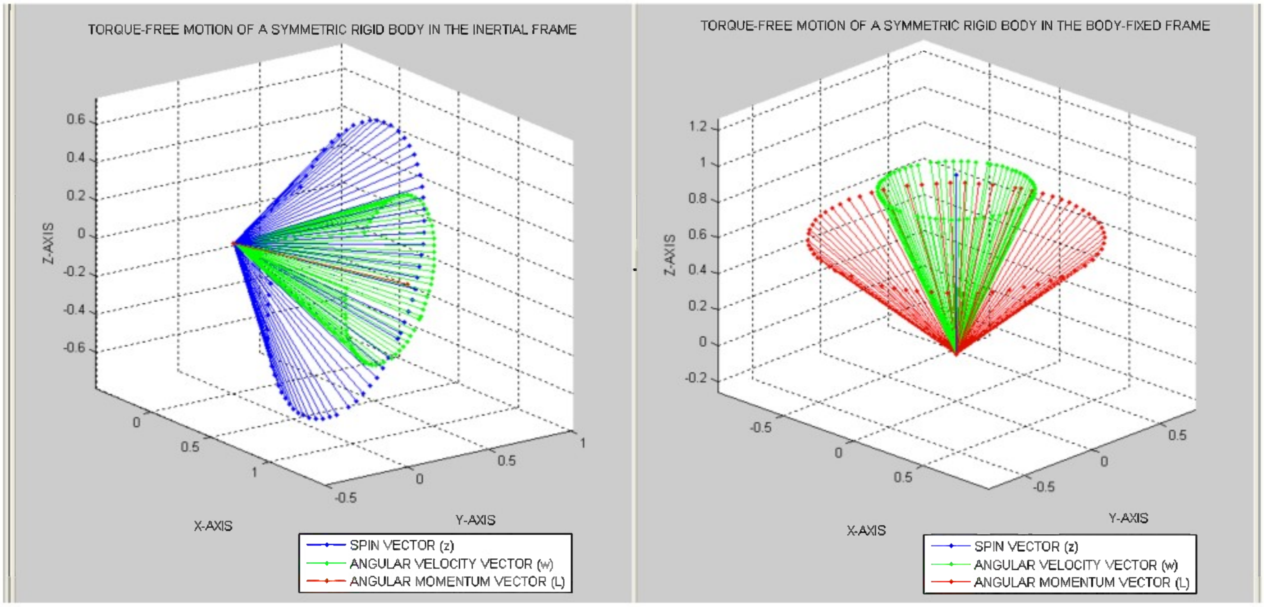


Figure 5.2 The angular momentum vector (red), angular velocity vector (green) and spin vector (blue) over time for the baseline motion solution which includes 64 discrete-time observations made over a 22.3 minute pass.

This result is consistent with the geometrical construction for the axial symmetric case, provided in Section 2.3.3, which pictures the rotational motion of a rigid body in terms of space and body cones rolling on top of one another. If the angular momentum vector $\boldsymbol{L}$, in red, and spin vector $\mathbf{z}$, in blue, represent the central axes of the space and body cones respectively, then the angular velocity vector $\boldsymbol{\omega}$, depicted in green, is the line of contact between the two cones.

The alignment of the wireframe model with each radar image produces a set of deviations or correction measurements, shown in Figure 5.3, which can be converted from sets of 1-2-3 Euler angles into rotation matrices via equation $(\mathrm{A}.1)$.

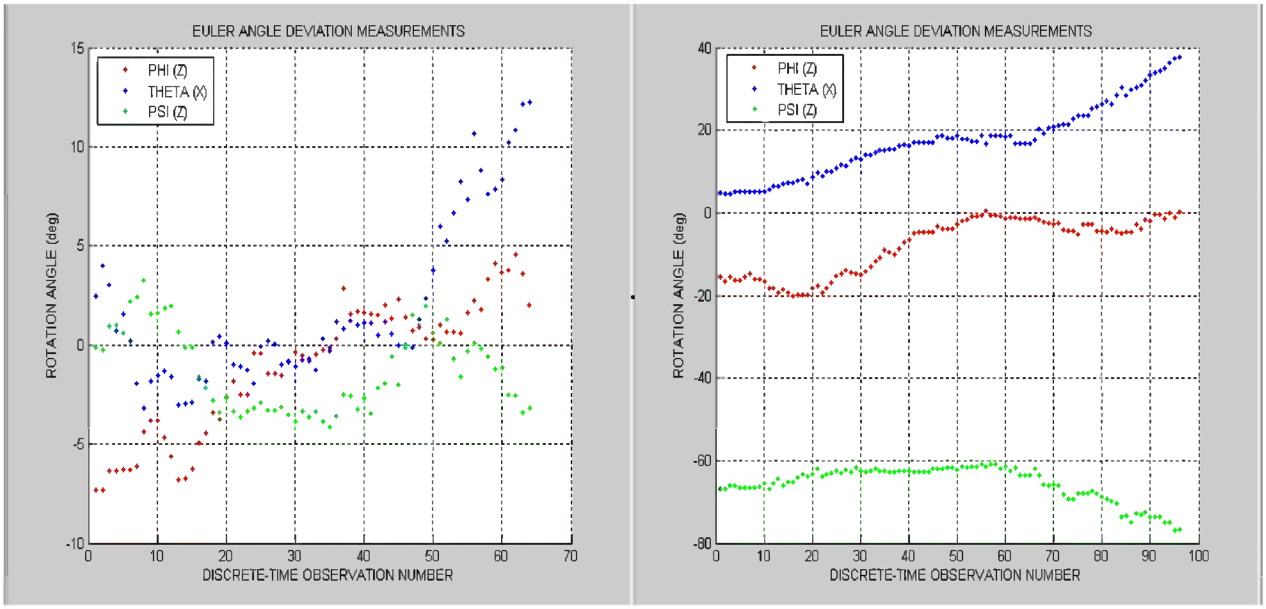


Figure 5.3 Euler angle deviations for the first (left) and second (right) pass recorded in XELIAS from the image-model matching process

As discussed in Section 4.3.1, the resulting alignment matrices can then be used to refine the baseline motion solution and generate the final set of quaternion observations. Unlike the first pass, the second has not been preprocessed using LLMotion. Therefore, the right-hand plot of Figure 5.3, is indicative of the large systematic errors which result when propagating an LLMotion solution over multiple passes. The final set of quaternion measurements and their Euler angle equivalents parameterize the attitude of the spacecraft relative to the Earth Centered Inertial (ECI) coordinate frame (i.e., the rotations needed to transform a vector from inertial to body coordinates), and are depicted in Figure 5.4 (for the first pass) and Figure 5.5 (for the second pass).

5.1.1 Filter Tuning and Initialization

The initial estimate for the quaternion components of the state vector are pulled directly from the vector components of first measurement $\mathbf{y}_0$:

1st Pass:
$$\left(\mathbf{x}_q\right)_1 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 0.37659353627381 \\ 0.11532246529129 \\ 0.77140682768356 \end{bmatrix} \qquad (5.1)$$

2nd Pass: $$(\mathbf{x}_q)_1 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} -0.14238997655619 \\ 0.92632927580853 \\ 0.25508611913828 \end{bmatrix} \qquad (5.2)$$

After the transfer has been made, the initial measurement vector $\mathbf{y}_0$ is removed from the observation cache and the total number of observations to be processed by the filter is reset to 63 for the first pass and 95 for the second.

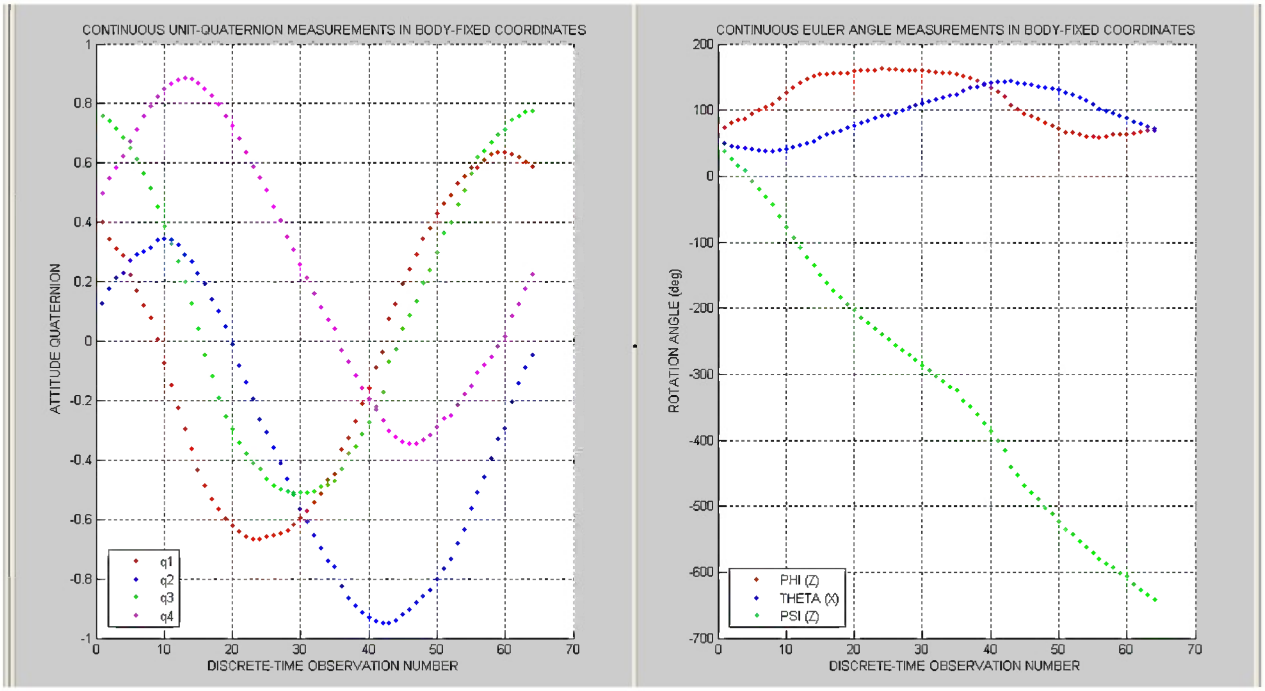


Figure 5.4 Baseline motion solution plus correction measurements for the first pass expressed in terms of the unit-quaternion (left) and as set of Euler angles (right)

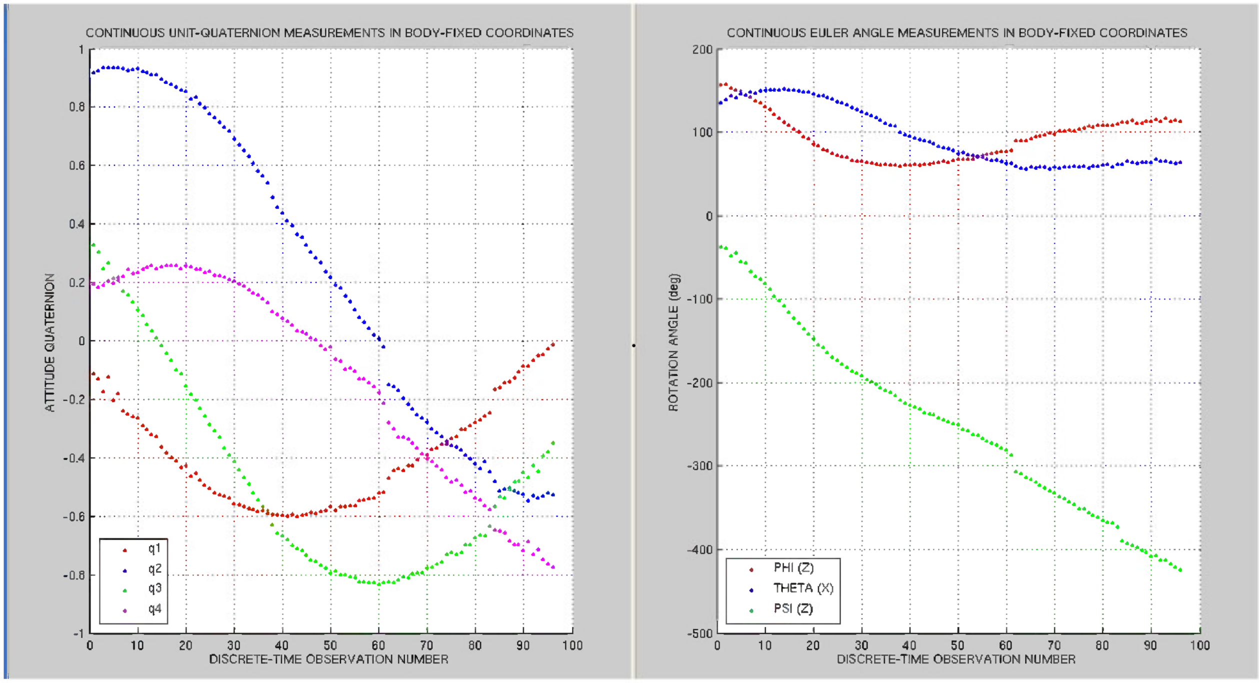


Figure 5.5 Baseline motion solution plus correction measurements for the second pass expressed in terms of the unit-quaternion (left) and as set of Euler angles (right)

Initial guesses for the angular velocity components of the state vector $\mathbf{x}_1$, the first six components of the state error covariance matrix $\mathbf{P}_1$, and measurement noise covariance matrix $\mathbf{R}$ are calculated using the built-in Matlab functions *polyfit* and *polyval* (17). *Polyfit* finds the coefficients of a polynomial $p(x)$ of degree n that fits a specified number of data points in a least squares sense. The result is a row vector of length $n + 1$ containing the polynomial coefficients in descending powers (17)

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n-1} \tag{5.3}$$

Conversely, the *polyval* function returns the value of a polynomial of degree n evaluated at each element x. Fitting a line ($n = 1$) to the first ten observations of ϕ, θ, and ψ, produces an initial estimate for the rate of change of each Euler angle (the slopes correspond to the angular rates: $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$), which can then be converted into an angular velocity vector using equation $(\mathrm{A}.17)$. The initial estimates for the two datasets are:

Pass #1: $$(\mathbf{x}_\omega)_1 = \begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}0.00144322242047\\ 0.00293828535360\\ -0.00702823296347\end{bmatrix} \tag{5.4}$$

Pass #2: $$(\mathbf{x}_\omega)_1 = \begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}0.00442350045872\\ 0.00021493637615\\ -0.00670942286442\end{bmatrix} \tag{5.5}$$

A similar procedure is also used to calculate values for the error covariance matrices. In the case of $\mathbf{P}_1$, 1st degree polynomials must also be computed for the first 10 observations of the vector components of the attitude quaternion. The estimates are then found by calculating the variance between the points on the line and the actual values at each discrete-time of interest. The component with the largest variance value is then rounded to the nearest power of 10 and used for all the elements on the diagonal. The state error covariance matrix for the actual test case, is given by:

$$\begin{aligned}\left(\mathbf{P}_{q\omega}\right)_1 &= \mathrm{diag}(0.0232, 0.0151, 0.0348, 1.38\times10^{-3}, 2.45\times10^{-3}, 2.96\times10^{-3})\\ &= \mathrm{diag}(0.035, 0.035, 0.035, 0.003, 0.003, 0.003)\end{aligned} \tag{5.6}$$

Computation of the measurement noise is automated in a similar manner, the only difference being, that a 10th degree polynomial ($n = 10$) is fitted to all four components of the quaternion, for all 64 measurements. The variance in the residuals between the actual values and those calculated with the polynomial equation result in the following measurement noise covariance matrix:

$$\mathbf{R}_i = \mathrm{diag}(0.0096, 0.0074, 0.0147, 0.0122) = \mathbf{1}_{4\times4} * (1.5\times10^{-2}) \tag{5.7}$$

Again the final matrix used in the filter sets all the diagonal components to that of the largest variance value rounded to the nearest power of 10. The filter is further tuned by selecting a process noise covariance matrix, which for all the test cases presented in this section is given by:

$$\mathbf{Q}_i = \mathbf{1}_{3\times3} * \left(\frac{1\times10^{-10}}{\Delta t_i}\right) \tag{5.8}$$

where $\Delta t_i = t_{i+1} - t_i$ and $\mathbf{1}_{3\times 3}$ is a 3×3 identity matrix. This process noise intensity is equivalent to what was used in Reference (41) and is approximately "3 to 10 times larger than the low-frequency limit of the power spectral density of the combined drag and solar/albedo-radiation pressure disturbance torques."

The computer graphics model in XELIAS is then transferred to Matlab for use in the in generating an approximation for the mass distribution of the target satellite. The inertia tensor for the satellite of interest, assuming uniform density within the bounds of the geometric model, is given by:

$$\boldsymbol{I}_B = \begin{bmatrix} 6.0075 & 0.4543 & -0.2232 \\ 0.4543 & 6.0257 & -0.0464 \\ -0.2232 & -0.0464 & 2.0552 \end{bmatrix} \mathrm{kg} \cdot \mathrm{m}^2 \tag{5.9}$$

Again, since the mass of the spacecraft is unobservable, the overall scaling of the inertia tensor is arbitrary. Performing singular value decomposition on equation (5.9) (this operation can be accomplished in Matlab by means of the *svd* function) and solving for the moment of inertia parameters $(p_{I1}, \ldots, p_{I6})$ in equation (4.26) yields:

$$\left(\mathbf{x}_{p_I}\right)_1 = \begin{bmatrix} p_{I1} \\ p_{I2} \\ p_{I3} \\ p_{I4} \\ p_{I5} \\ p_{I6} \end{bmatrix} = \begin{bmatrix} 2.6040 \\ 4.2106 \\ 7.7466 \\ 0.0053 \\ -0.0557 \\ 2.3518 \end{bmatrix} \tag{5.10}$$

where the first three values have units of $\mathrm{kg}^{0.5} \cdot \mathrm{m}$ and the last three are in rad. The block of $\mathbf{P}_1$ that models the initial uncertainty in p_{I1}, p_{I2}, and p_{I3} is $\left(\mathbf{P}_{p_I}\right)_1 = \mathrm{diag}(0.6, 0.6, 0.00015)\ \mathrm{kg} \cdot \mathrm{m}^2$. These values indicate a 10% - 30% [3] one-sigma uncertainty in the initial estimates of p_{I1} and p_{I2} and a 0.5% initial uncertainty in p_{I3}. The latter uncertainty enforces a soft constraint on the unobservable scaling of the spacecraft inertias. The block of $\mathbf{P}_1$ that corresponds to p_{I4}, p_{I5}, and

[3] The equation used for calculating the percentage of one-sigma uncertainty in the initial moments of inertia is given by: $\% = \sqrt{P_{p_I}}/p_I$. The 30% value corresponds to $p_{I1}$and the 10% uncertainty corresponds to p_{I2}.

p_{I6} is set to $I_{3\times3} * (8 \times 10^{-2})$ rad, which reflects the initial uncertainty in the spacecraft's principal axis directions.

The variables contained with the subsequent table are required to implement the backward-smoothing extended Kalman filter as detailed in Section 3.2.5.1 and have a significant impact both the accuracy and execution time of the algorithm. In the case of the BSEKF, improved filter accuracy can be achieved by: 1) increasing m_{target} and/or j_{max} and 2) decreasing the values associated with s_{max}, γ_{ε}, and/or J_{ε}. Greater precision obviously comes at the expense of increased computational burden, in terms of both longer run times and greater memory usage.

Table 5-2 Critical Backward-Smoothing Extended Kalman Filter Variables

Variable Description	Symbol	Value for Pass #1	Value for Pass #2
Total Number of Observations to be Processed by the BSEKF	n	63	95
Target Number of Stages Used in the Forward Filter and Backward Smoother	m_{target}	40	50
Maximum Number of Gauss-Newton Iterations	j_{max}	15	15
Maximum Numerical Integration Step Size	s_{max}	0.5	0.5
Threshold Value for the Guarding Procedure in Gauss-Newton Minimization	γ_{ε}	1×10^{-10}	1×10^{-10}
Cutoff Value for the Cost Function Minimization	J_{ε}	1×10^{-10}	1×10^{-10}

It is important to note that the BSEKF has been shown to perform better than other filters (specifically the UKF and EKF) only if the number of explicitly optimized sample intervals m_{target} is sufficiently large. Per the recommendations made in Reference (41), the m_{target} value was chosen to allow the filter to converge nearly to steady-state when $k = m_{target}$.

5.2 Spacecraft Attitude Estimation Problem – Simulated Test Case

A simplified test case, consisting of a new set of quaternion measurements over the same time span as the actual observations, has also been produced using the filter's own dynamics equations and propagator. This truth-model simulation is used to create an artificial situation in which the true orientation, rates, and mass distribution of the spacecraft are known precisely, in order to assess the filter's convergence accuracy and reliability. The simulated dataset was generated using the same initial estimates for the attitude, angular velocity, and moment of inertia parameters as the first pass of the actual test case. The initial state vector, given in the second column of Table 5-3, was propagated to the same observation times using the kinematics equation (4.31) and dynamics equation (4.32) in conjunction with the Matlab function called *ode45*, which solves initial value problems for their ordinary differential equations. A random number generator was then used to add error to the new set of observations. This "measurement noise" is normally distributed with zero mean and standard deviation of three $\sim N(0,3)$ for the first and last 10 observations in the pass, and $N(0,2)$ for the other 44 observations. The increased noise added to the ends of the pass is meant to reflect the challenges associated with aligning the wireframe model to images whose resolution is somewhat degraded the further it is from the center of the pass. The sources of this error are discussed in greater detail in the context of the actual test case described in Section 5.3.2. The 64 simulated observations with the addition of artificial measurement error are shown in Figures 5.6 and 5.7.

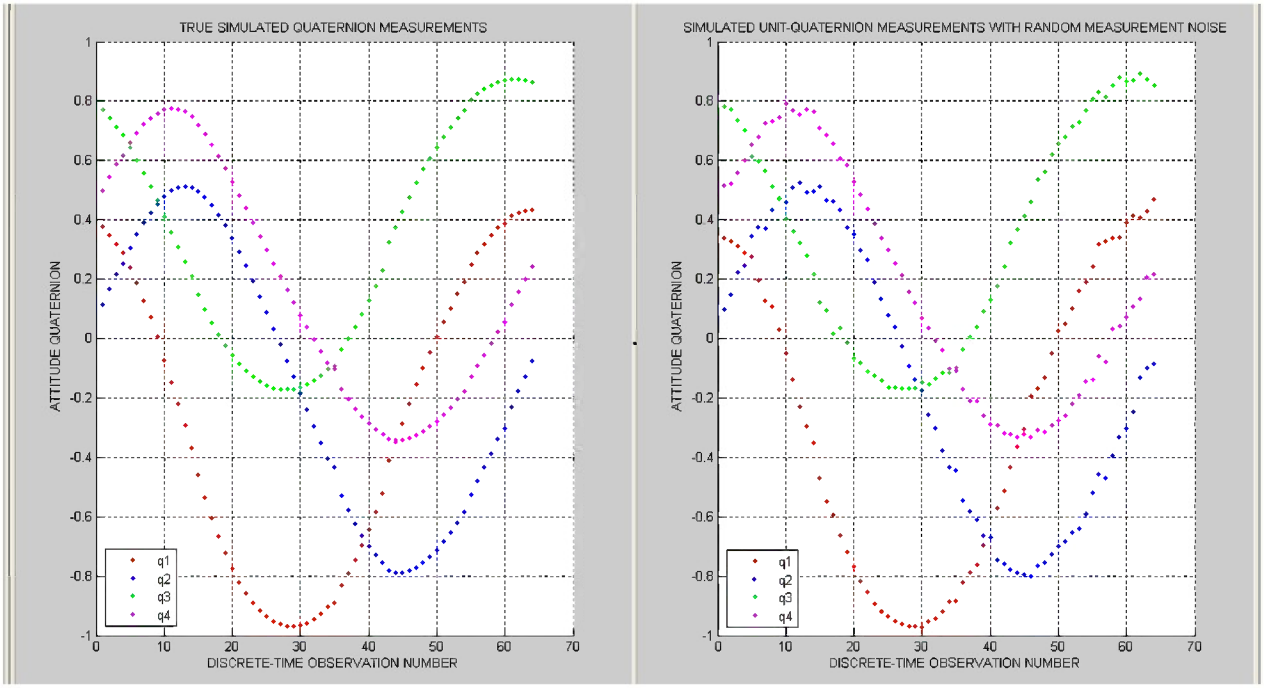


Figure 5.6 True simulated attitude quaternion measurements (left) and simulated quaternion with the addition of random noise (right)

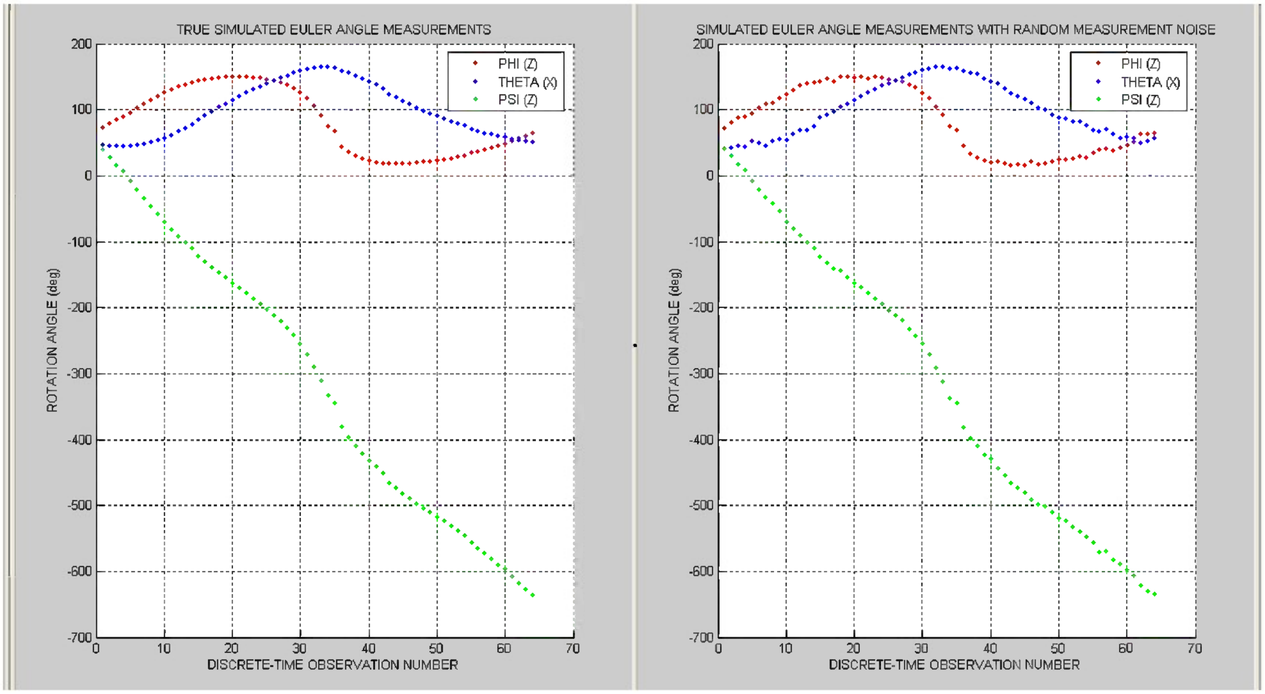


Figure 5.7 True simulated Euler angle measurements (left) and simulated Euler angles with the addition of random noise (right)

5.2.1 Filter Tuning and Initialization

In order to fully test the ability of the backward-smoothing extended Kalman filter to rapidly recover from poor initial estimates, the algorithm was given an extremely inaccurate preliminary state vector. Though the particular values used for the first six components of $\mathbf{x}_1$ were chosen arbitrarily, they do represent a physically real attitude quaternion and angular velocity vector. Using equation $(\mathrm{A}.7)$ one is able to determine that the initial attitude, in terms of 3-1-3 Euler angles, is wrong by approximately 35 degrees about the first axis of rotation (ϕ) and 95 degrees about the second and third (θ and ψ). The error in the angular velocity terms varies widely, ranging from as little as 4% (which is the case for ω_3) to as much as 88% (for ω_1). Finally, the inertia matrix has also been changed to a fully-symmetric model in which all the principal moments are equal:

$$\boldsymbol{I}_B = [\mathbf{1}_{3\times3} * (10)]\ \mathrm{kg}\cdot\mathrm{m}^2 \tag{5.11}$$

Consequently, the filter starts out with an estimate for the moment of inertia matrix in which the principal inertia ratios are wrong by as much as 65% and the principal axes directions are incorrect by between 35 and 45 degrees. The error in any given element is no more than 34% of the maximum principal inertia. This level of inaccuracy is meant to test the robustness of the filter and reflects a level of uncertainty which is far greater than what is likely to be encountered in the real non-cooperative attitude estimation problem. The subsequent table (Table 5-3) and set of figures (Figures 5.8 and 5.9) present the poor initial estimates, employed in the simulation, compared against the true state vector used to generate the observations.

Table 5-3 Initial State Vectors for the Simulated Test Case

State Vector Component	True State Vector Value	Initial Estimate Value
q_1	0.37659353627381	0.140134360644793
q_2	0.11532246529129	0.941434088194620
q_3	0.77140682768356	0.145834358774792
ω_1	0.00144322242047	0.002705006174102
ω_2	0.00293828535360	−0.002116872057220
ω_3	−0.00702823296347	−0.007055041225524
p_{I1}	2.60400465586434	7.7597
p_{I2}	4.21063662937094	7.7597
p_{I3}	7.74726396398345	7.7597
p_{I4}	0.00529770721622	0
p_{I5}	−0.05562767690357	0
p_{I6}	2.35274994257264	0

To get a better sense of just how bad the initial state vector is, the attitude was propagated forwards to the same discrete observation times of interest. The predicted attitude is shown in Figures 5.8 and 5.9, plotted against the true simulated measurements.

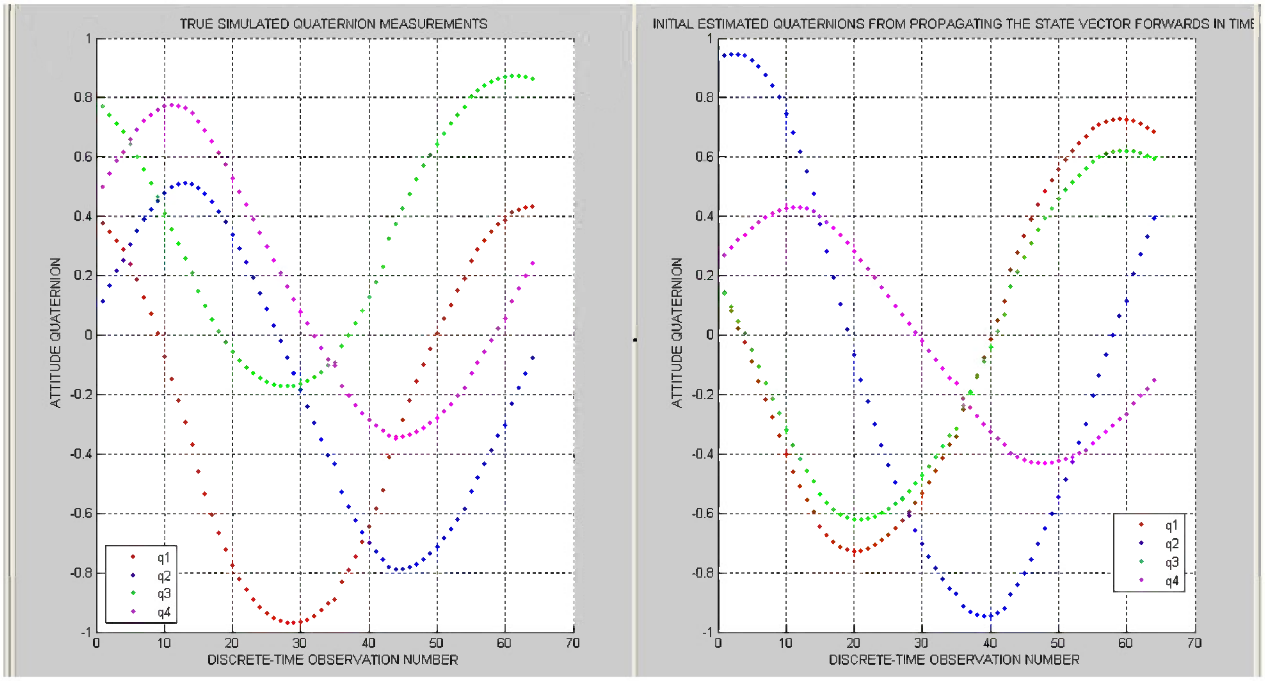


Figure 5.8 Comparison between the true simulated quaternion measurements (left) and the poor initial attitude quaternion propagated to the same discrete times (right)

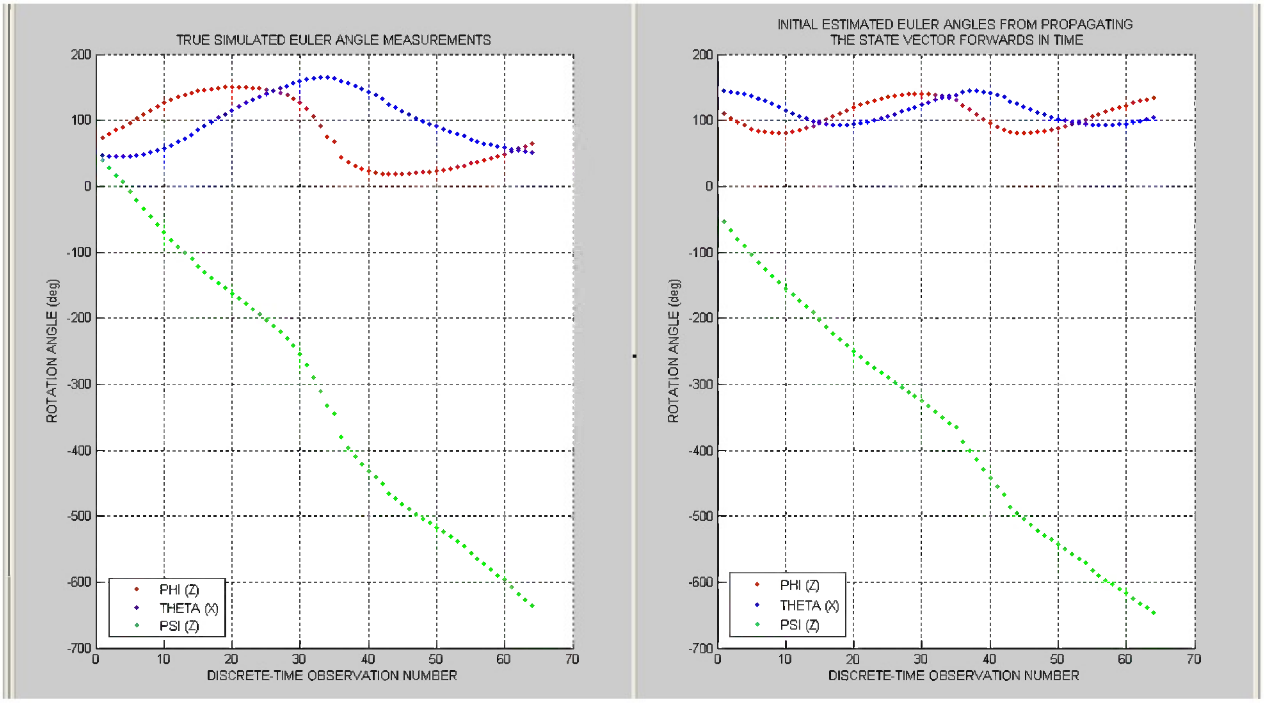


Figure 5.9 Comparison between the true simulated Euler angle measurements (left) and the poor initial attitude propagated to the same times of interest (right)

The state and measurement noise covariance matrices were revised to reflect the increased amount of uncertainty in the initial attitude and angular velocity vectors. The measurement noise covariance was recomputed using the same procedure as described in the previous section, while the state error covariance matrix for the simulated dataset simply scales the values given in equation (5.4) by a factor of 10. The recalculated matrices are:

$$\left(\mathbf{P}_{q\omega}\right)_1 = \mathrm{diag}(0.35, 0.35, 0.35, 0.03, 0.03, 0.03) \tag{5.12}$$

and

$$\mathbf{R}_i = \mathrm{diag}(0.0966, 0.0740, 0.0839, 0.0222) = \mathbf{1}_{4\times4} * (9.7 \times 10^{-2}) \tag{5.13}$$

Similarly, the error covariance associated with p_{I1}, p_{I2}, and p_{I3} was recalculate based on reflect a 40% one-sigma uncertainty in the initial estimates of p_{I1} and p_{I2} and a 0.5% initial uncertainty in p_{I3}. The initial uncertainty in the spacecraft's principal axis directions. The state error covariance for the moment of inertia parameters is given by:

$$\left(\mathbf{P}_{p_I}\right)_1 = \mathrm{diag}(10, 10, 0.00015, 0.068, 0.068, 0.068) \tag{5.14}$$

Finally, in addition to using the same process noise covariance $\mathbf{Q}_i$ as the actual datasets, the total number of explicit nonlinear smoothing samples has been set to 40, which is equivalent to the m_{target} used in the first pass.

5.3 Filtering Results

The residuals that result from subtracting the estimated quaternion from the measured or simulated quaternion are the primary quantitative measures used in assessing filter performance. Though the residual is only one indicator of convergence, it is particularly sensitive to the accuracy of the attitude state over time. The filter has reached a steady state when the residuals over time no longer appear to fluctuate significantly and when the error covariance matrix is stable. As discussed in Section 3.2.3.1, divergence occurs when the residuals appear to get consistently larger over time, as the attitude estimate moves steadily away from the true state. The filter will consistently converge on a motion solution only after the appropriate balance between the state noise and measurement noise covariance matrices has been found. If the filter has been properly tuned, the residual plots can also be used to readily identify individual observations which have a particularly large error. Such results indicate a problem with the cross-range scaling or a misalignment of the computer graphics model with the underlying image. The analyst can either remeasure that particular image or omit that attitude observation altogether in future filtering operations.

The causes of attitude uncertainty may be separated into the two categories of random and systematic errors. A random error is an indefiniteness of the result due to the finite precision of the test, or a measure of the fluctuations in the result after repeated experimentation. A systematic error on the other hand, is a reproducible inaccuracy introduced by faulty calibration or technique. Systematic errors are biases in the measurements which lead to the situation where the mean of many separate observations differs significantly from the actual value of the measured attribute. All measurements are prone to systematic errors, often of several different types. Sources of systematic error may be imperfect calibration of measurement instruments, changes in the environment which interfere with the measurement process and sometimes imperfect methods of observation can be either zero error or percentage error. For example, the range values measured by radar will be systematically overestimated if the slight slowing down of the waves in air is not accounted for.

5.3.1 Simulated Test Case Results

For the simulated test case, the residuals provide a fairly clear cut picture of the filter's performance. The BSEKF is able to reduce the total attitude error in the estimate to less than 5 degrees in the first 10 minutes, and settles in to a steady-state performance in less than 30 minutes. Its steady-state peak per-axis attitude error is around 4 degrees and its peak total attitude error is approximately 10 degrees (see Figures 5.10 and 5.11).

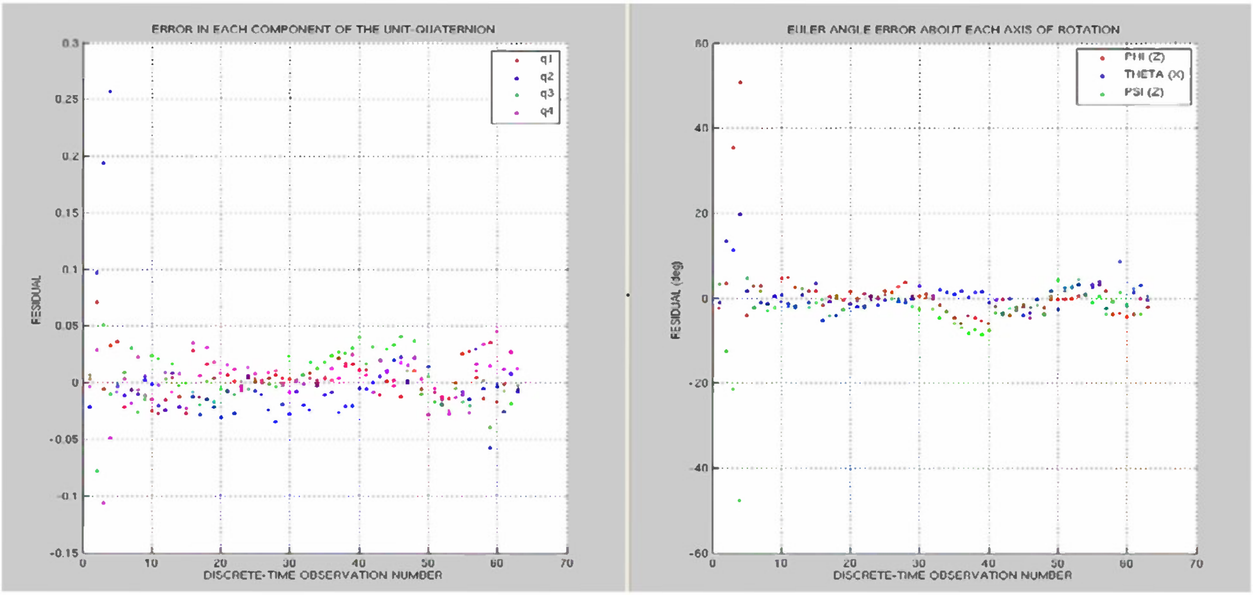


Figure 5.10 Error in each component of the quaternion (left) and Euler angle (right) for the noisy simulated test case as each new observation is processed

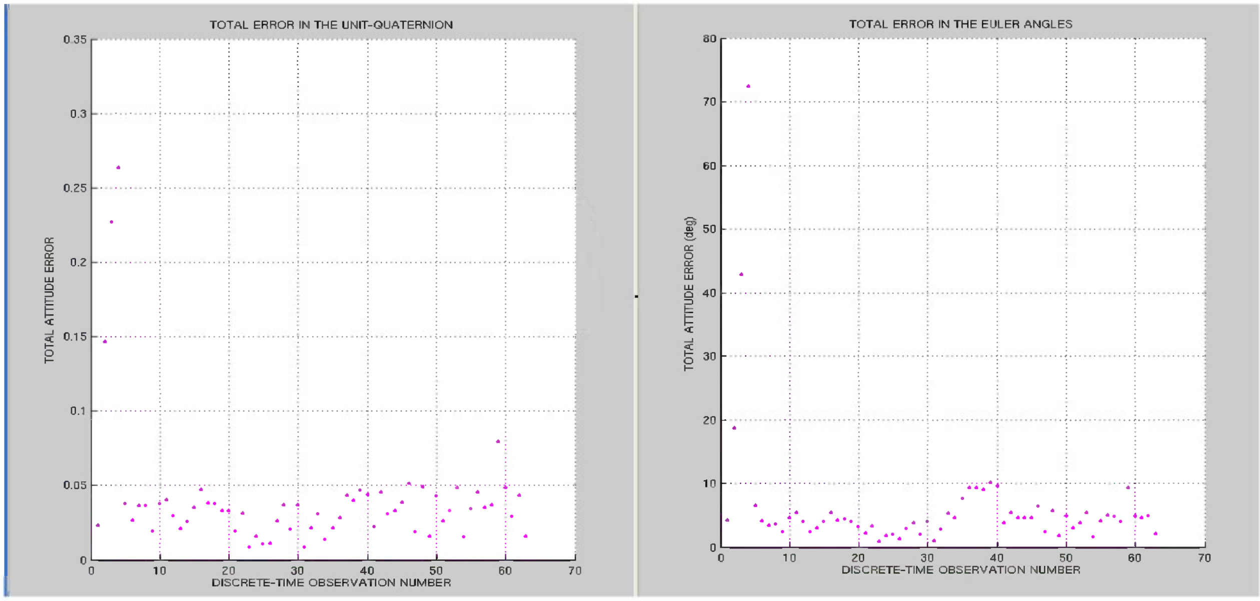


Figure 5.11 Total attitude error in terms of the unit-quaternion (left) and Euler angle (right) representations for the simulated test case

The diagonal terms of the state error covariance matrix, obtained after processing each observation in the pass, are plotted in Figures 5.12 and 5.13. The variance in the attitude and angular velocity error are both rapidly reduced from their initial values and quickly stabilize around 0.002 (radians)^2 and 1.5×10^{-6} (radians/second)^2, respectively. The variance in the moment of inertia parameters, on the other hand, is much more erratic and never appears to settle into a definitive steady state performance. Not surprisingly, this result indicates that, given a poor initial estimate of the inertia tensor, there are not enough attitude observations in a single pass to completely determine the mass distribution of the spacecraft (i.e., information saturation has not occurred). From Figure 5.13 one can also see that error covariance for p_{I6} does not begin to consistently reduce until after observation 35, which is about when the variance in the attitude error settles into a consistent performance. This suggests the filter has difficulty distinguishing between the first and second moments of inertia for rapidly spinning spacecraft; especially when there is only a 0.3% difference between the two values. Since the spacecraft is nearly symmetric, the mass is fairly evenly distributed along the x- and y-axes. Once the uncertainty in the other state component has been sufficiently reduced, it may become easier for the filter to detect these subtle differences.

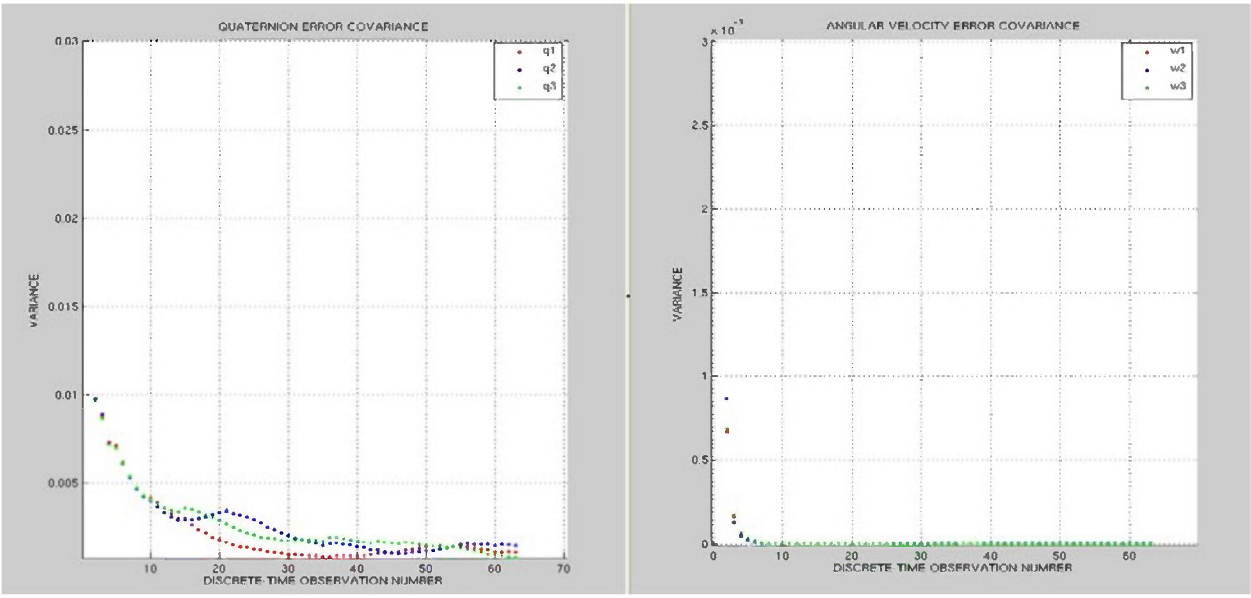


Figure 5.12 Covariance for the vector components of the attitude quaternion (left) and angular velocity (right) after processing each observation

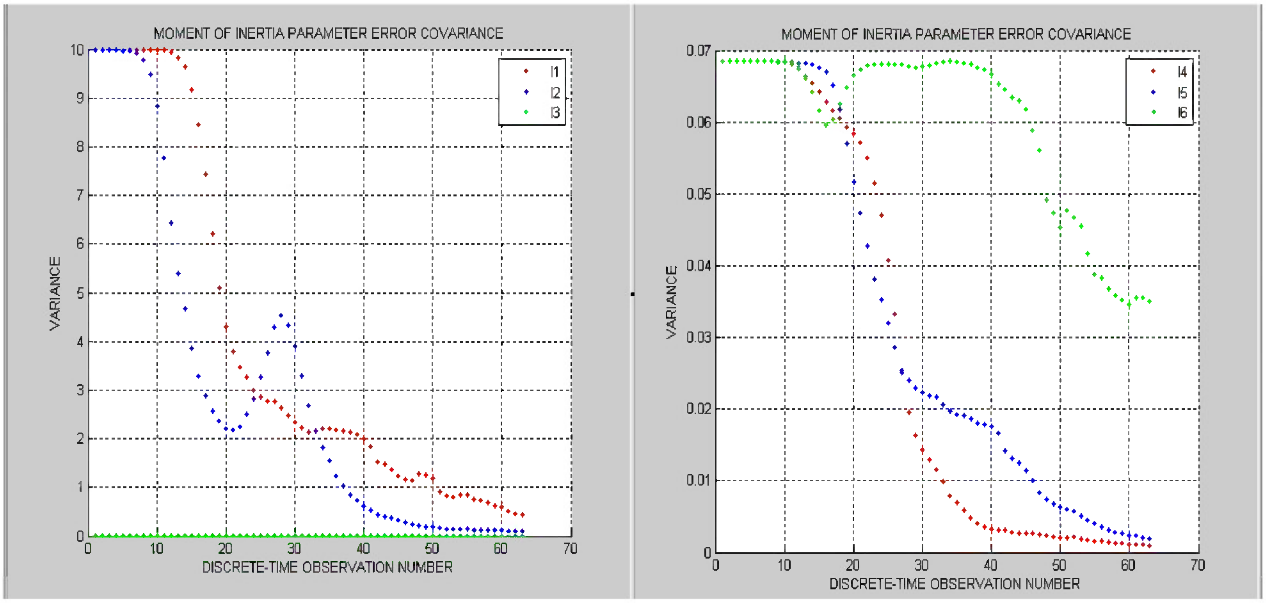


Figure 5.13 Covariance for the moment of inertia parameters after processing each discrete-time observation. The uncertainty in the parameters associated with the lengths of the sides of a uniform rectangular box are given in the leftmost graph and those related to the 1-2-3 Euler angles are shown in the graph on the right.

The total attitude error after processing the final observation is about 2 degrees and the error covariance settles around 0.001, midway through the pass. Another measure of the accuracy and effectiveness of the algorithm can be obtained by propagating the final processed state vector backwards in time and comparing the resulting attitude estimate against both the noisy measurements and true simulated observations.

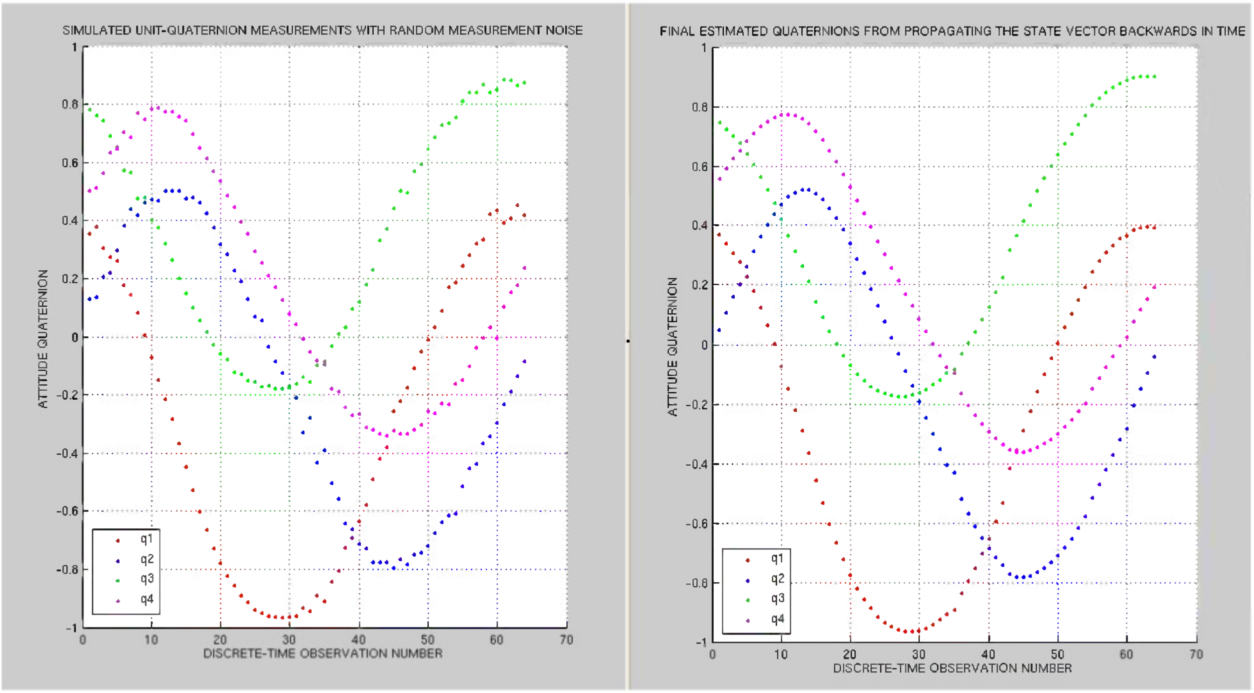


Figure 5.14 Visual comparison of the final predicted attitude quaternion against the noisy measurements for the simulated test case

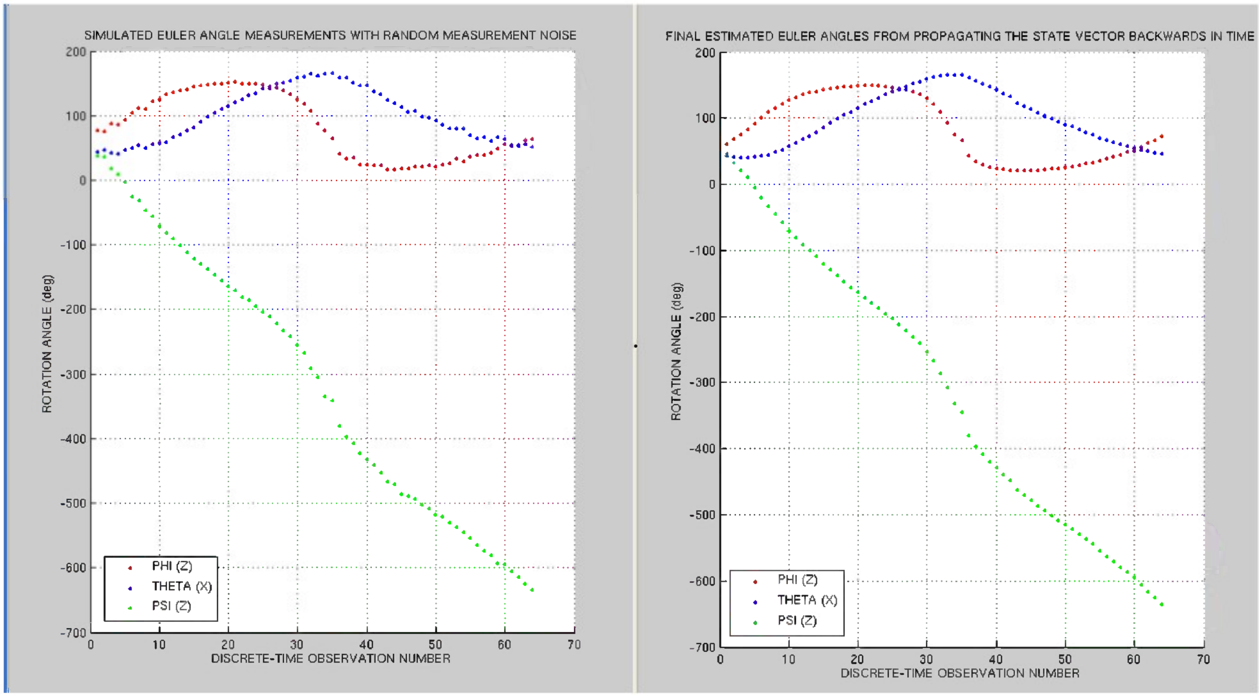


Figure 5.15 Visual comparison of the final predicted attitude, in terms of the Euler angle representation, against the noisy measurements for the simulated test case

From this qualitative evaluation, the backwards propagated state estimate seems to fit the data set very well. Of particular interest is level of precision which can be achieved for those estimates at observation times near the beginning of the pass. In order get a better feel for the accuracy of the final filtered state consider the residuals that result from subtracting each estimated attitude quaternion from the corresponding noisy measurement. From Figures 5.16 and 5.17, it appears that the BSEKF has been able to remove the random errors which were added to each of the simulated measurements; this is reflected in the fact that residuals do not follow any regular pattern, are centered on zero, and fall primarily within $\pm$ 4 degrees of the mean. This however, is not the case for the residuals corresponding to the first and last 10 observation times. The total error at the beginning and end of the pass is as much 7 degrees, which is more than twice the amount of error present at any other point in the pass. These results therefore, accurately reflect the additional noise $(N(0,3))$ which was added to these boundary regions in order to more closely simulate the conditions which would be experienced in the actual datasets. As will be explained in greater detail the next section, it is much more difficult to make accurate measurements of the attitude towards the ends of a given pass.

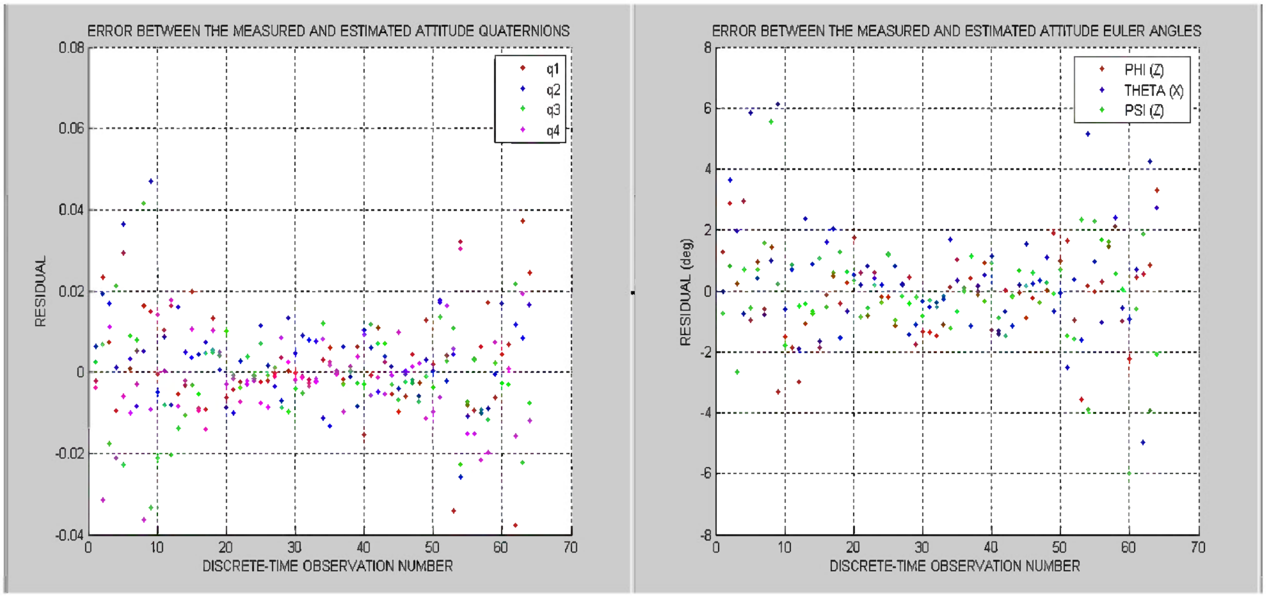


Figure 5.16 Error between the final predicted attitude and noisy measurements expressed in terms of the unit-quaternion (left) and Euler angles (right)

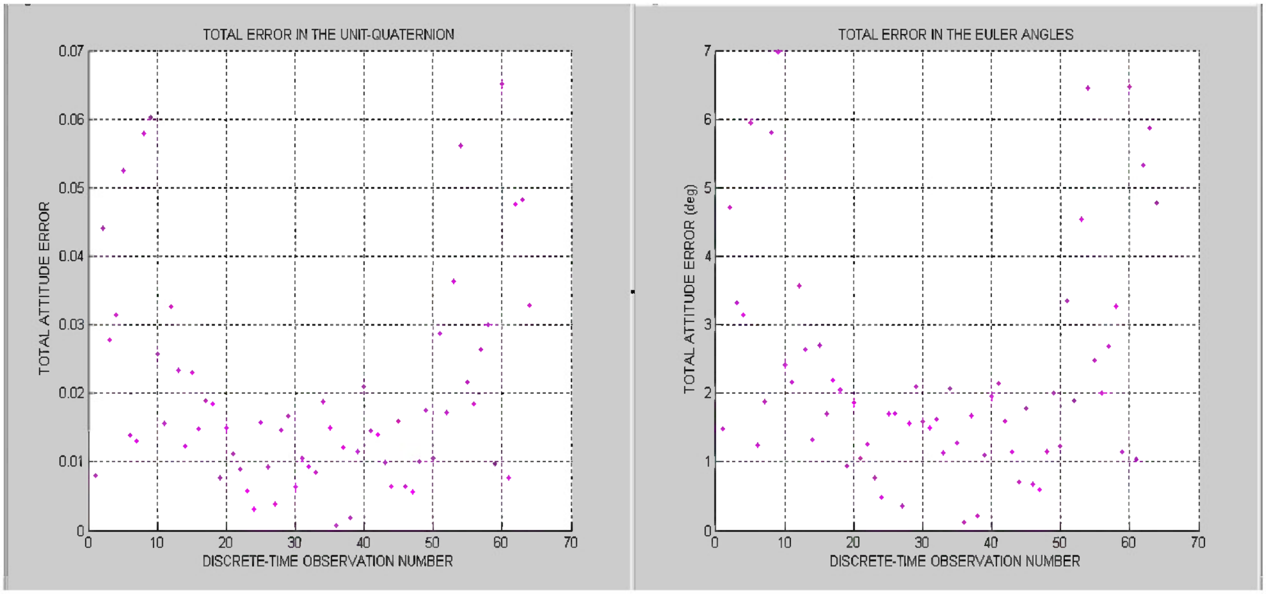


Figure 5.17 Total error in the final predicted attitude in terms of the unit-quaternion (left) and Euler angles (right)

In the actual test cases, the true motion of the spacecraft over time is unknown. The quality of an attitude prediction can only be assessed with respect to the noisy observations. For the simulation however, the state parameters at any given instant are known precisely, enabling the propagated quaternion to be compared against the true attitude. A side by side comparison of the true and predicted attitude is provided in Figures 5.18 and 5.19; neither plot reveals any noticeable discrepancies.

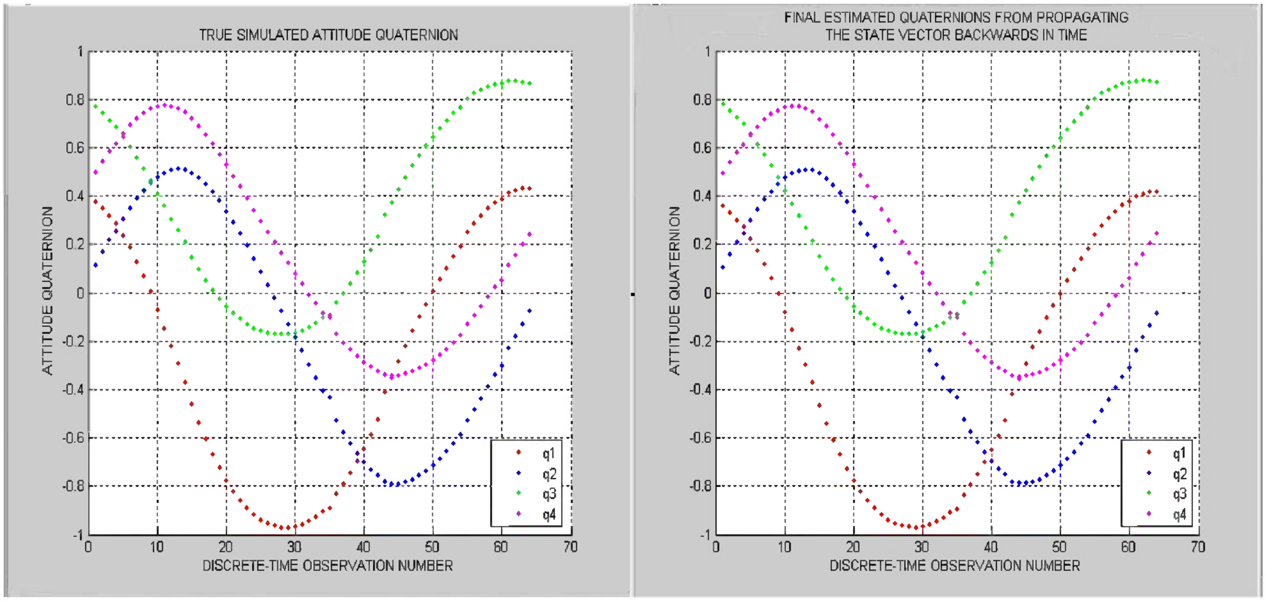


Figure 5.18 The true simulated attitude (left) versus predicted attitude quaternion (right) after processing all the observations in the pass

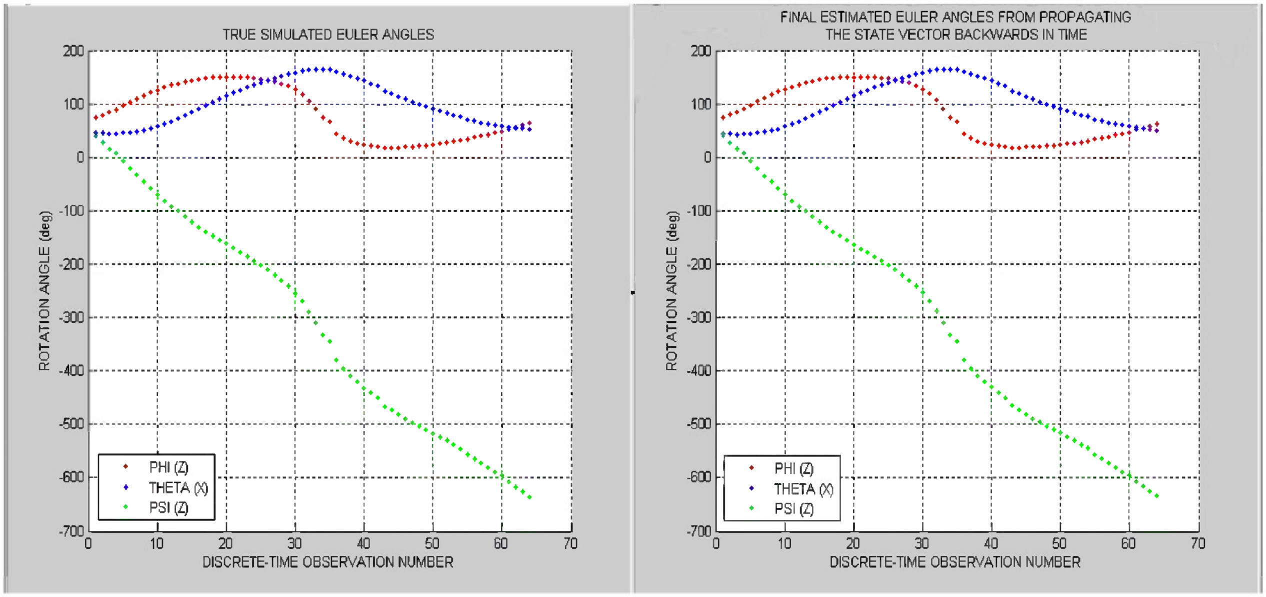


Figure 5.19 The true simulated attitude (left) versus predicted Euler angles (right) after processing all the observations in the pass

Returning to more quantitative performance measures confirms that the BSEKF has been able to accurately determine the motion of this fictitious satellite. Even with the spikes in boundary observations, the total error in any given attitude estimate within the pass does not exceed 2.25 degrees. However, the set of residuals depicted in Figure 5.20 reveal that what appeared to be mostly random noise in Figure 5.16 is, in reality, also comprised of very small systematic errors. This suggests that further tuning may be needed, since under ideal conditions the noise should be Gaussian. It should also be noted that in the case of the quaternion residuals, the second greatest departure from the true attitude occurs in the last 5 observations (see Figure 5.21). Because the measurements were made with the filter's own dynamics equations and torque models, the divergence of the attitude state towards the beginning of the pass, can be directly attributed to the remaining errors in the final state estimate.

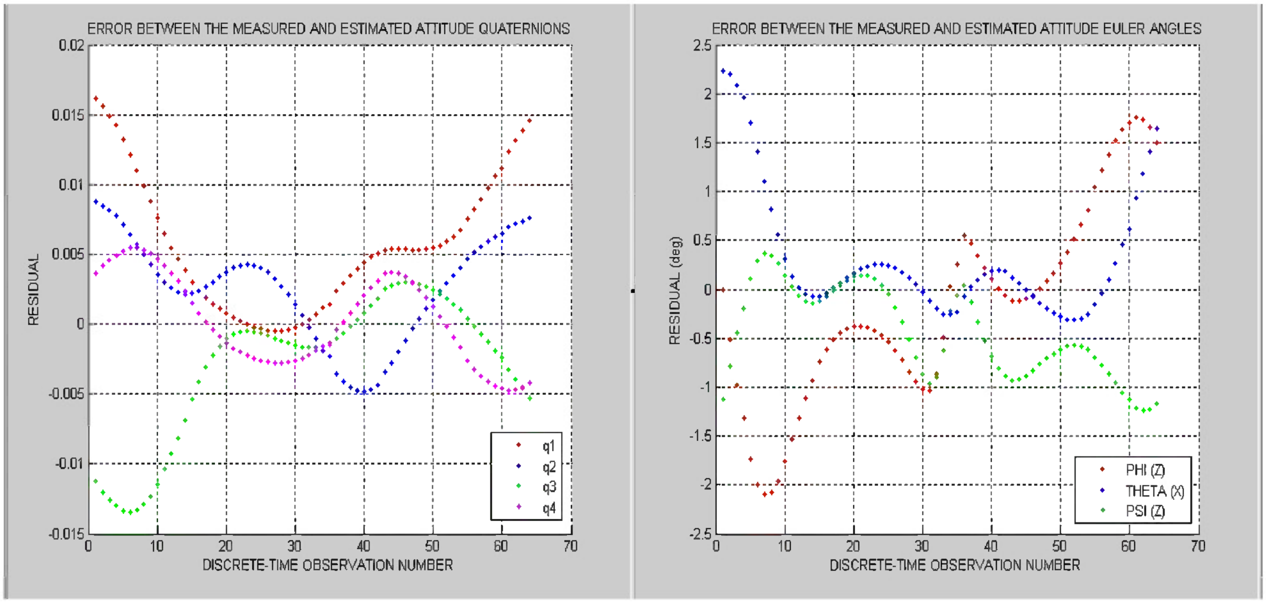


Figure 5.20 Residuals between the final predicted state estimate and the true simulated observations

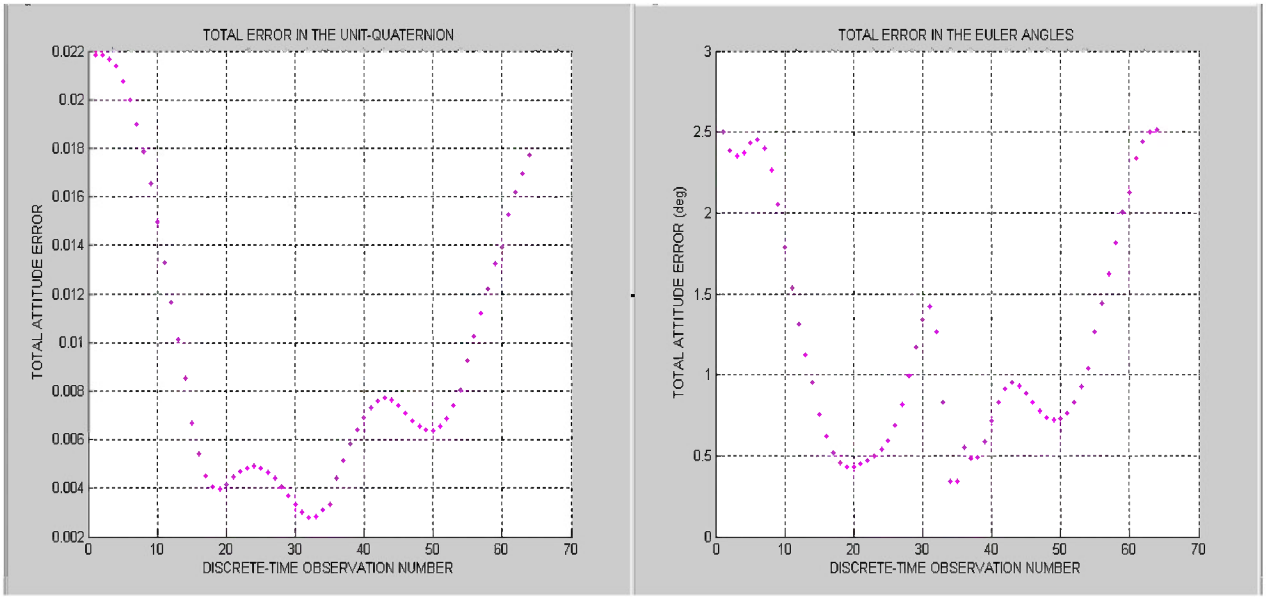


Figure 5.21 Total attitude error between the final predicted state estimates and the true simulated observations

It is important to remember that the cost is only minimizes over those estimates contained within the m-buffer (cache), which in this case is 40. Consequently, as the smoothing window begins to "shift along", those observations which have already been processed have a diminishing effect on the current state estimate $\hat{\mathbf{x}}_k$. In order to remain sensitive to new observations, the BSEKF is allowing the influence of earlier data to fade over time. Because the filter does maintain a backlog of observations, similar to that of a batch algorithm, the estimator's memory does not fade as quickly as other types of Kalman filters. In the preceding figures, it is therefore, not surprising that as the state estimate begins to breakdown when propagated to observation times outside what is contained in the cache. Unfortunately, this sort of outcome does not bode well for being able to do long term attitude prediction. While the filter is able to rapidly recover from large initial errors and converges on the correct motion solution, the final estimate is still not known with enough accuracy to precisely fit all the observations in a short 22 minute pass. Unless the filter is able determine the motion parameters of the spacecraft more precisely, it is highly unlikely that the attitude can be predicted with very much accuracy over multiple passes, spanning several hours, as any errors in the initial state will cause the attitude to diverge.

The fractional error norm of the estimated inertia is given by the quantity

$$\frac{\left\|\hat{\boldsymbol{I}}_B * \mathrm{tr}(\boldsymbol{I}_B)/\mathrm{tr}\left(\hat{\boldsymbol{I}}_B\right) - \boldsymbol{I}_B\right\|}{\|\boldsymbol{I}_B\|} \tag{5.15}$$

and provides a good metric of the accuracy with which the filter's moment of inertia matrix estimate $\hat{\boldsymbol{I}}_B$ approximates the true inertia matrix $\boldsymbol{I}_B$. The trace ratio in the expression is included to remove the unobservable overall scaling. The 41% initial moment of inertia estimation error converges to 1.6% by the end of the filtering process. This represents a 96.1% decrease in the inertia matrix modeling error. The ability of the filter to simultaneously recover from poor initial estimates and cut through measurement noise in the simulated dataset lends credibility to results obtained in the actual test case. Since the true attitude, angular velocity, and mass distribution of the target satellite are unknown in the real-life test case, the simulation is very useful in tuning the filter and evaluating its ability to refine estimates.

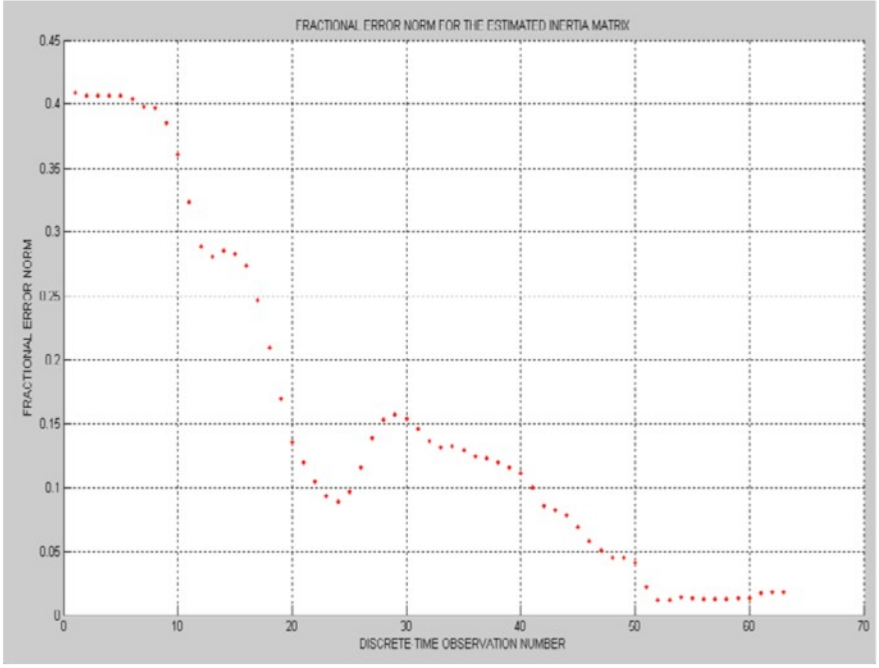


Figure 5.22 Fractional error norm for the estimated inertia matrix as each new observation is processed.

5.3.1.1 Simulated Test Case – Long-Term Attitude Prediction

In order to evaluate the ability of the final BSEKF state estimate to be used in predicting the attitude over the time gap separating two consecutive passes, a new set of truth measurements was generated by propagating the simulated state vector to the same observation times used in the second pass of the actual test case. A comparison of the true and estimated state at the last observation time for the first pass reveals that there are less than 2 degrees of total error in the attitude and 1.6% error in the moment of inertia estimate. The least accurate component is the sixth moment of inertia parameter corresponding to the z-axis rotation needed to orient a uniform rectangular box multiplied by the square root of its mass with respect to spacecraft coordinates. The amount of inaccuracy in p_{I6} is in agreement with the error covariance depicted in right hand graph of Figure 5.13, which indicates that this particular parameter is fairly difficult to determine from the attitude measurements.

Table 5-4 Final State Vectors for the Simulated Test Case

State Value	True State Vector	Estimated State Vector	Difference
q_1	0.433359215122371	0.436244890157324	−0.00288567503495
q_2	−0.074790520515558	−0.073988166134436	−0.00080235438112
q_3	0.864399308793746	0.863566032869862	0.00083327592388
ω_1	0.002888628273754	0.002862109016238	0.00002651925752
ω_2	0.001400626259306	0.001300661040758	0.00009996521855
ω_3	−0.007039370441506	−0.007071061229068	0.00003169078756
p_{I1}	2.604004655864339	2.497433299943323	0.10657135592102
p_{I2}	4.210636629370945	4.250976971438172	−0.04034034206723
p_{I3}	7.747263963983449	7.746025950913061	0.00123801307039
p_{I4}	0.005297707216223	0.016551166355644	−0.01125345913942
p_{I5}	−0.055627676903572	−0.057455881580290	0.00182820467672
p_{I6}	2.352749942572640	−0.861133876734452	3.21388381930709

Even though the state estimate has minimal errors, the long propagation time, over a gap of 1.88 hours, amplifies even slight differences in certain attitude components. Figures 5.23 and 5.24 show the predicted-versus-true attitude at each of the 95 discrete observation times in the second pass.

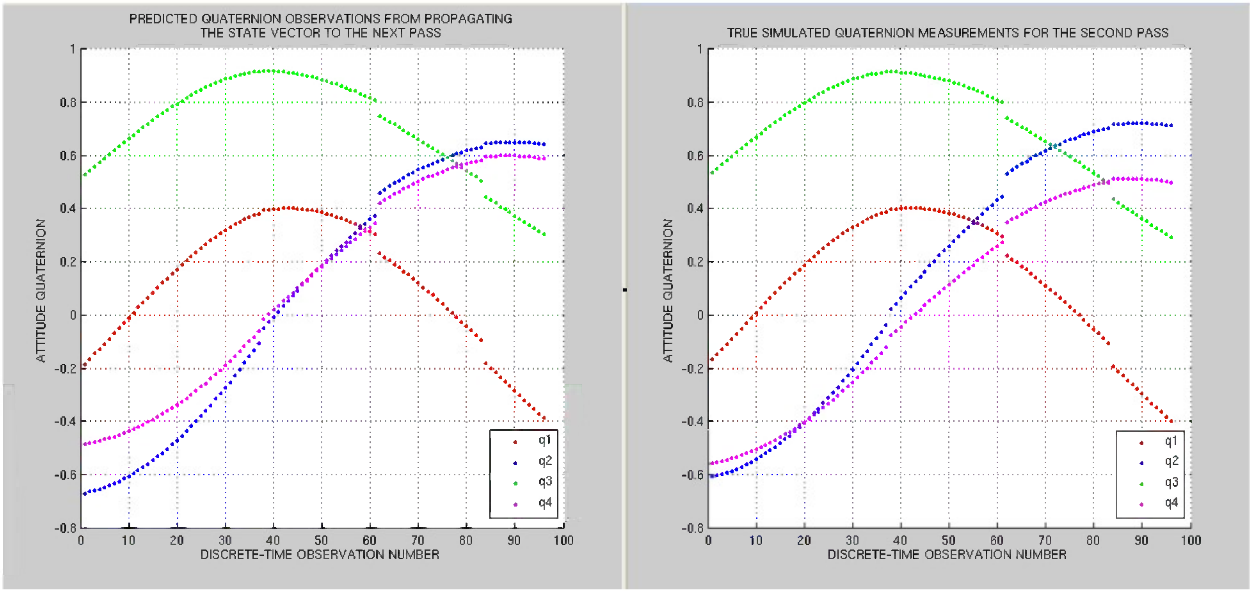


Figure 5.23 Side-by-side comparison of the predicted quaternion (left) and true attitude measurements (right) for the second pass

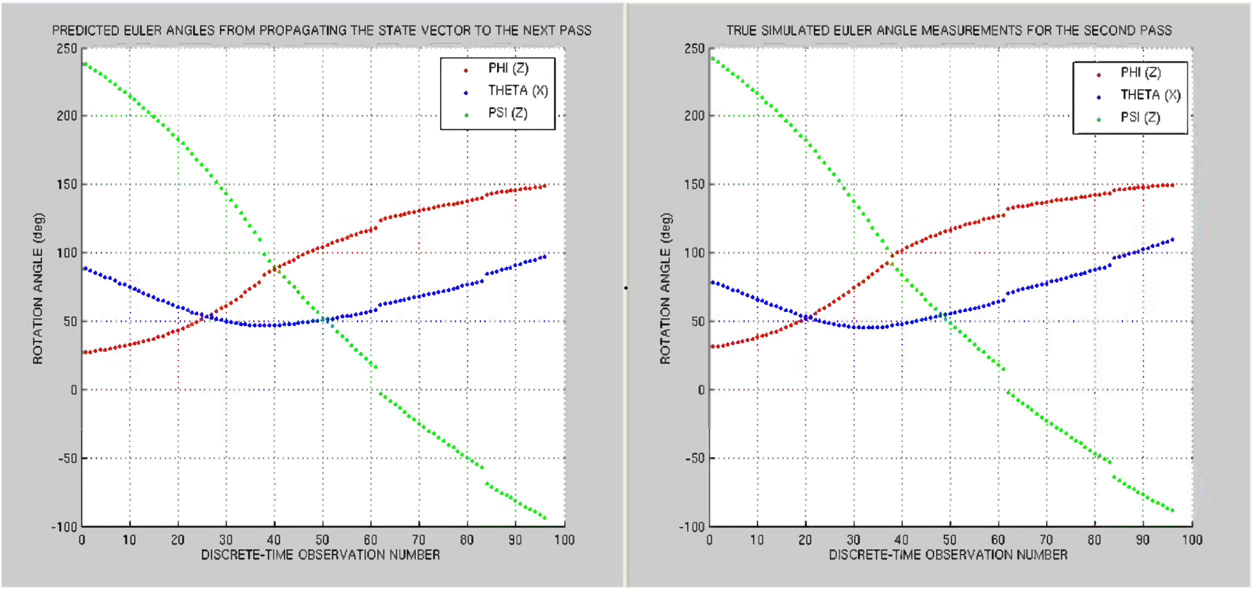


Figure 5.24 Side-by-side comparison of the predicted Euler angles (left) and true attitude measurements (right) for the second pass

From Figure 5.23, one is able to quickly discern that q_1 (red data points) and q_3 (green data points) are very close to truth, while, qualitatively speaking, the sinusoidal curve created by plotting q_2 over time appears to be the correct shape but offset by approximately 0.065. The nonlinear relationships that exist between different state components results in a very complex state space. It is therefore, rather interesting that the error in the predicted attitude would predominantly affect a single quaternion component, q_2 (blue datapoints), and manifest itself as a fairly consistent underestimation of the true value. Since the fourth component of the quaternion is not independent, the deviation in q_2 in turn produces an equal and opposite error in the values of q_4 (i.e., the magenta data points appear to be shifted upwards by about – 0.065). Plotting the residuals reveals that the error in each of the quaternion components is in fact fairly constant, fluctuating by no more than 0.03. The Euler angles, on the other hand, vary by as much as 22 degrees in the case of θ and have a total error which ranges from as little as 10.6 degrees, to as much as 15.7 degrees.

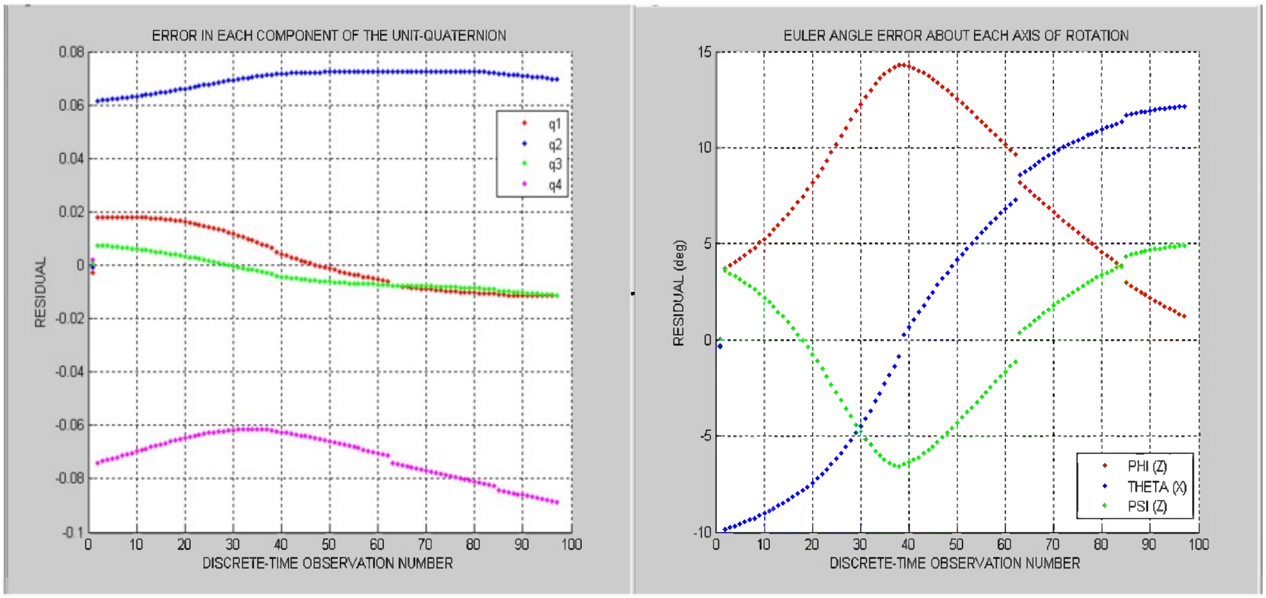


Figure 5.25 Residuals for the attitude quaternion (left) and Euler angles (right)

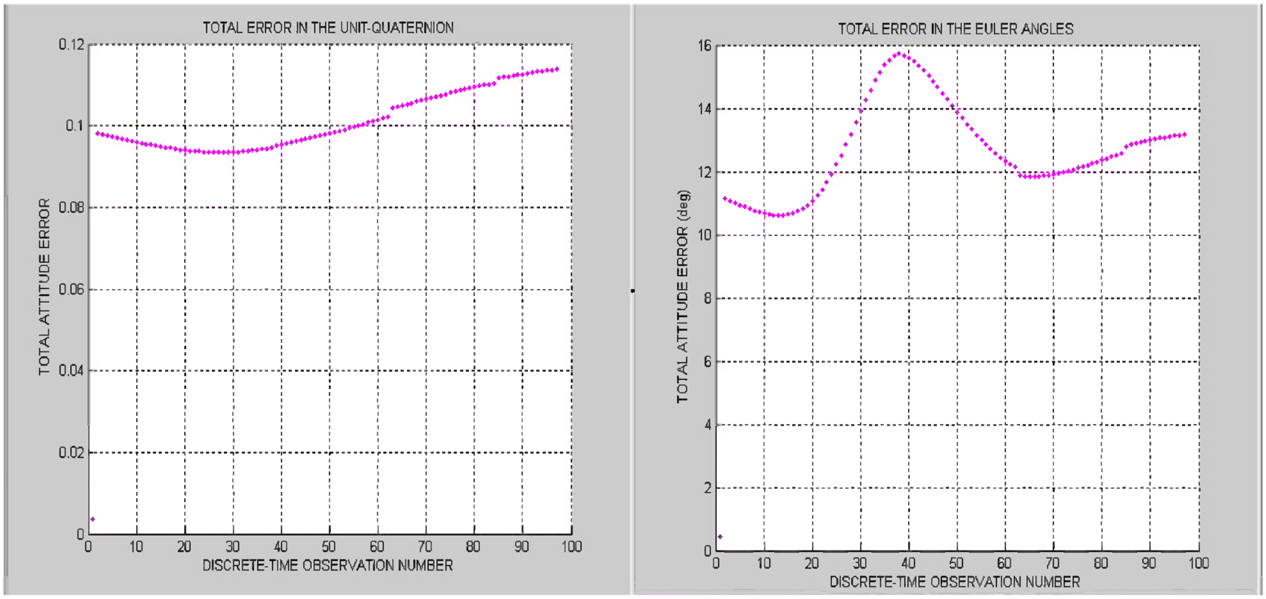


Figure 5.26 Total error for the second pass attitude prediction

These results indicate the even minor errors in the final estimated state vector can produce large differences in the attitude quaternion when propagated over extended periods of time. While this outcome is not overly surprising, it sets an important benchmark against which actual long-term attitude prediction results can be evaluated. It is unlikely that the error in the final state estimate of the actual test case will be less than what has been achieved by the filter in the truth-model simulation. Consequently, one would expect to see even greater residuals for real-life attitude predictions.

5.3.1.2 Simulated Test Case – Multi-Pass Attitude Estimation

In this test case, the BSEKF has been given two consecutive pass of simulated measurements, in order evaluate the ability of the algorithm to filter over long time gaps in the data. The amount of time separating the two datasets is approximately 1.88 hours. The artificial measurement noise used in this scenario is assumed to be Gaussian, unbiased, and greater towards the ends of the pass. Thus, the random error is $N(0,2)$ for the first and last 10 observations in the combined pass, $N(0,1.5)$ for observations 11 to 20 and 140 to 149, and $N(0,1)$ for the other 120 measurements in between. The 160 simulated observations with and without the addition of artificial measurement error are shown in Figure 5.27.

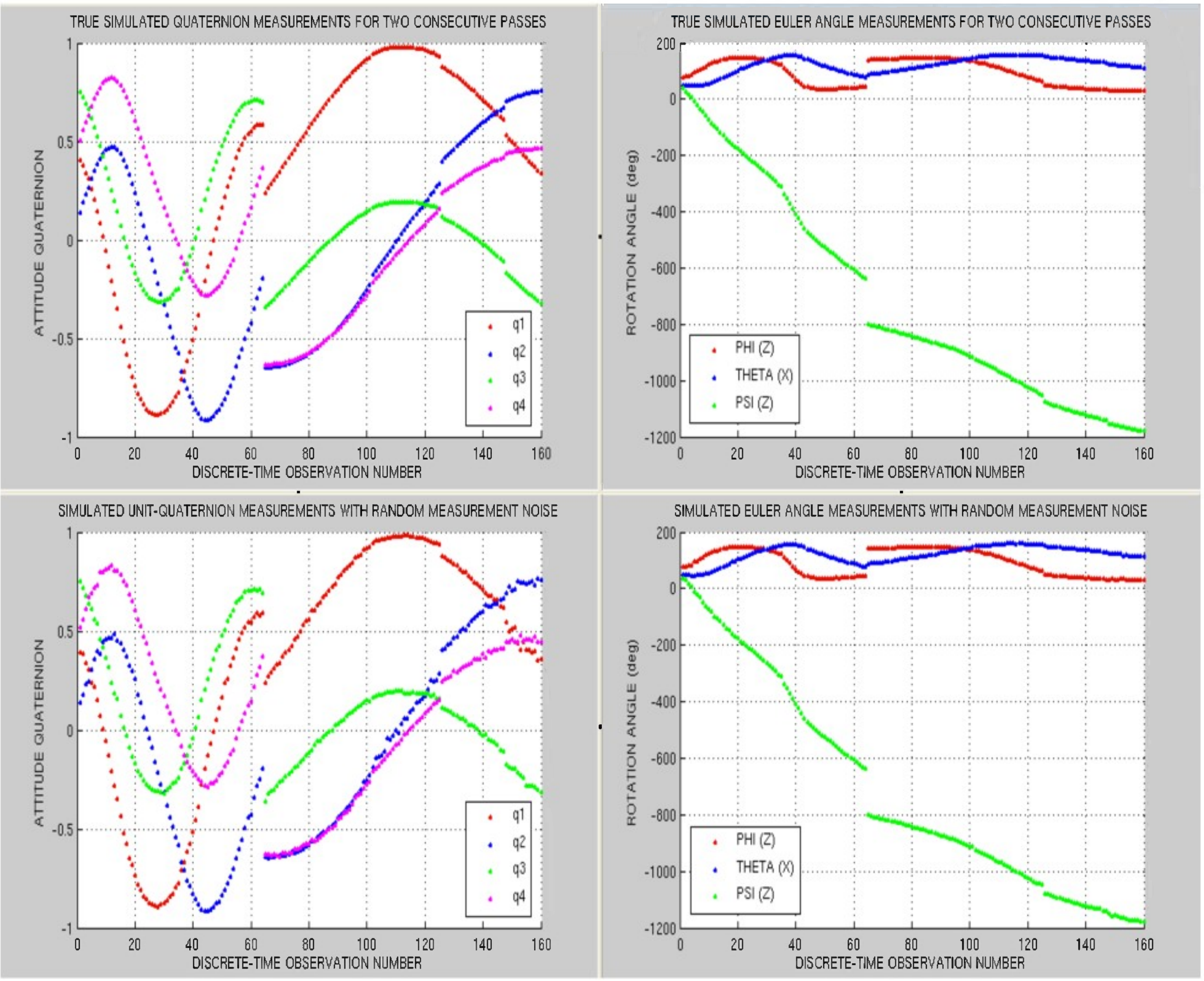


Figure 5.27 True and noisy quaternion and Euler angle measurements for the simulated multi-pass test case

It is important to note that referencing each attitude based on its sequential observation number, rather than a specific time, results in a distorted view of the spacecraft's motion. Because the second dataset contains 32 more observations, the first pass is significantly compressed and, thus, looks to be much shorter, when, in reality it is approximately twice as long as the second. Additionally, longer time gaps between observations result in what appear to be discontinuities in the attitude measurements. However, displaying the attitude versus time is not without problems. Specifically, the long time span separating the two datasets results in a plot which is mostly empty space and, therefore, difficult to read.

From Figure 5.28, one can see that the filter is able to rapidly reduce the total amount of error in the initial attitude estimate, settling into a clear steady state performance around 2 degrees within the first hour of filtering. The larger residuals observed at the end of the pass are a direct result of the increased amount of measurement noise which has been added to the last 20 observations. The nonlinearities in the measurements are responsible for the brief spike in the Euler angle residuals preceding observation 40.

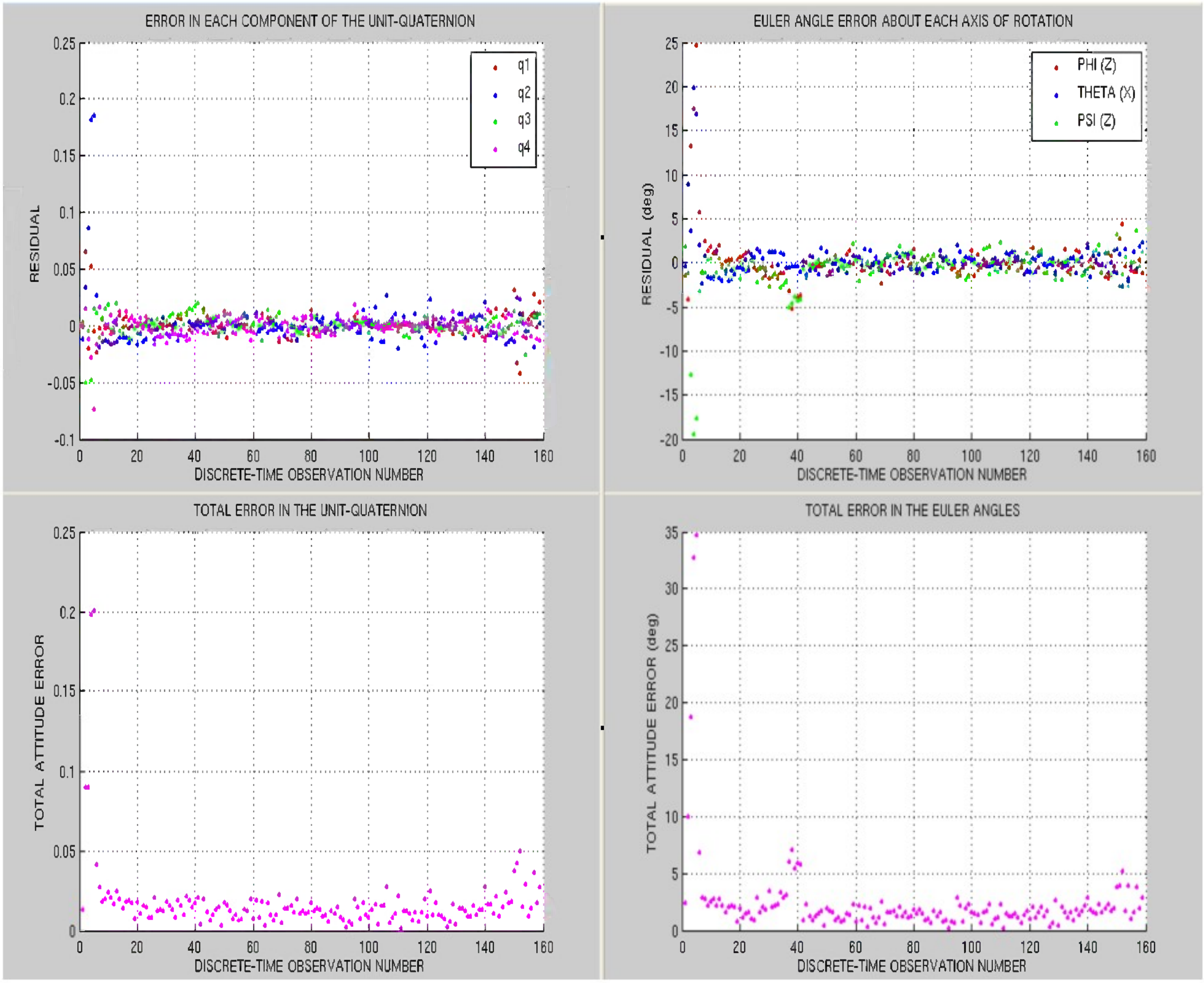


Figure 5.28 Residuals and total error after processing each discrete observation in the simulated multi-pass test case

After all the observations in the first pass have been processed, the state error covariance matrix is increased to half its initial value in order to reflect the fact there is a substantial amount uncertainty in the state estimate beginning the new pass. Thus, the covariance matrix is reset to:

$$\mathbf{P}_{64} = \mathrm{diag}(0.025, 0.025, 0.025, 2.5 \times 10^{-3}, 2.5 \times 10^{-3}, 2.5 \times 10^{-3}, 5, 5, 7.5 \times 10^{-4}, 0.4, 0.4, 0.4) \tag{5.16}$$

According to graphs in Figure 5.29, these values appear to have been rather excessive, given that within the first observation of the second pass the error covariance associated with each state parameter is significantly reduced from what is provided in equation (5.16).

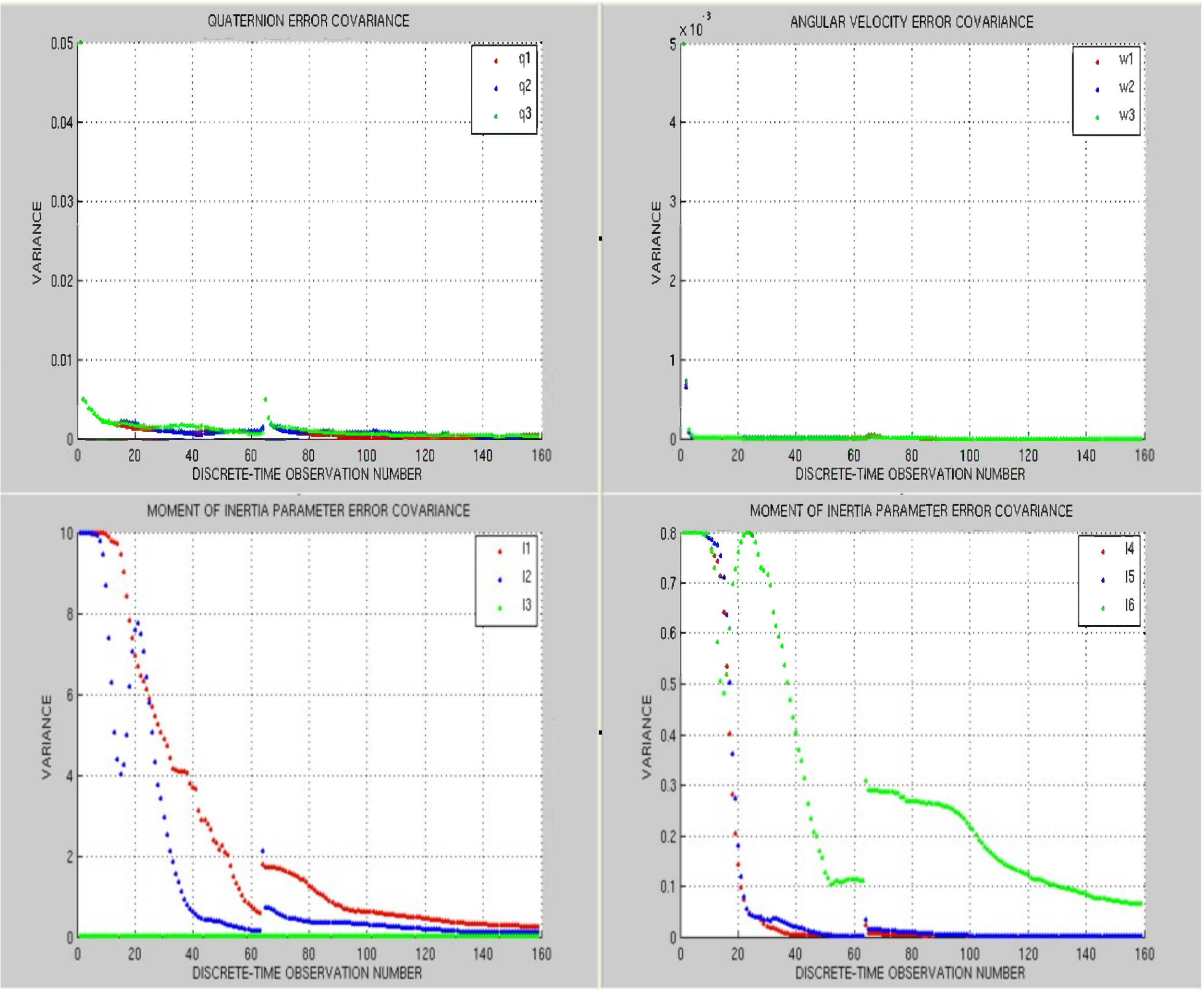


Figure 5.29 State error covariance for the simulated multi-pass test case

The right side of Figure 5.30 shows the fractional error norm for the estimated inertia tensor after processing each observation in the combined dataset. In this test case, the 41% initial moment of inertia estimation error converges to 0.8% by the end of the first pass and remains at about this level for the remainder of the filtering process. This represents a 98.05% decrease in the inertia matrix modeling error.

The left side of Figure 5.30 depicts the amount of time needed to process each observation in the pass. The computation time increases fairly linearly until the target number smoothing stages is reached $(m_{target} = 40)$, at which point the algorithm spends approximately 2.5 minutes per observation. As the algorithm begins to filter over the time gap separating the observations from the two passes, the processing time increases dramatically, averaging roughly 12 minutes per measurement, and becomes highly variable, fluctuating from 6 to 28 minutes per measurement. During this period, the computational burden is most directly influenced by the numerical integration step size s, the target number of smoothing stages m_{target}, and the number of Gauss-Newton iterations j preformed. Once the fixed interval (m-buffer) no longer spans the two passes, the filter again settles into a steady state performance around 2.5 minutes per observation.

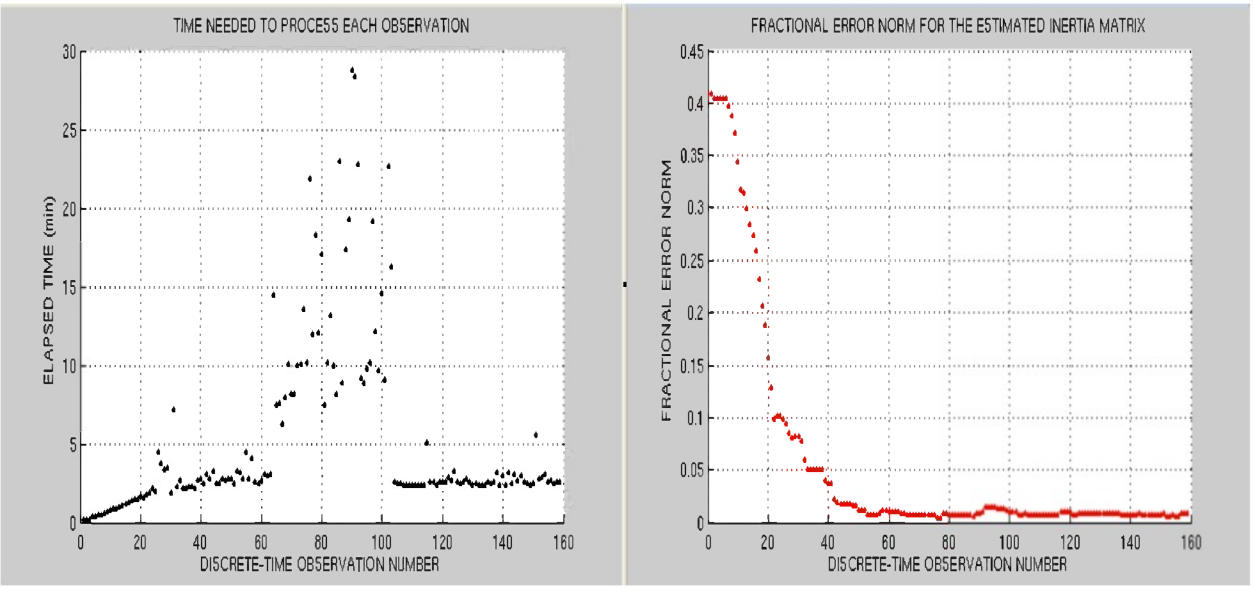


Figure 5.30 Algorithm processing time (left) and fractional error norm for the estimated inertia tensor (right)

Despite having datasets separated by almost two hours, the BSEKF is able to propagate the attitude to the next pass and begin updating the state estimate using the new observations. This capability is invaluable, given the limited amount of time in which the satellite is in view of the ground sensor on any given pass. Finite pass length, along with imaging geometry, dictates the number of independent images which can be generated and the diversity of attitude information contained within the sequence of images. In the non-cooperative attitude estimation problem, the ability to accurately determine a satellite's motion from a single set of two-dimensional images is dependent on: 1) the extent to which the aspect angle changes relative to the RLOS and 2) the amount of angular separation in the RLOS over the course of the imaging period. While a single pass may not be able to meet these criteria, linking together several discrete datasets should provide the filter with a sufficient number of measurements to refine the state estimate. Furthermore, difference in the imaging geometry should help resolve the ambiguity in the attitude of the radar image plane by increasing the parallax (i.e., the apparent displacement or difference of orientation of an object viewed along two different radar lines of sight). Propagating the final state estimate $\hat{\mathbf{x}}_{160}$ backwards over both passes and plotting the resulting data points against the true attitude measurements yields the set of graphs in Figure 5.31. The predicted values are depicted as colored dots while the true attitude components are shown as black plus symbols. The fact that there are only minor discrepancies in the attitude prediction over the two passes is a testament to the correctness of the final state estimate.

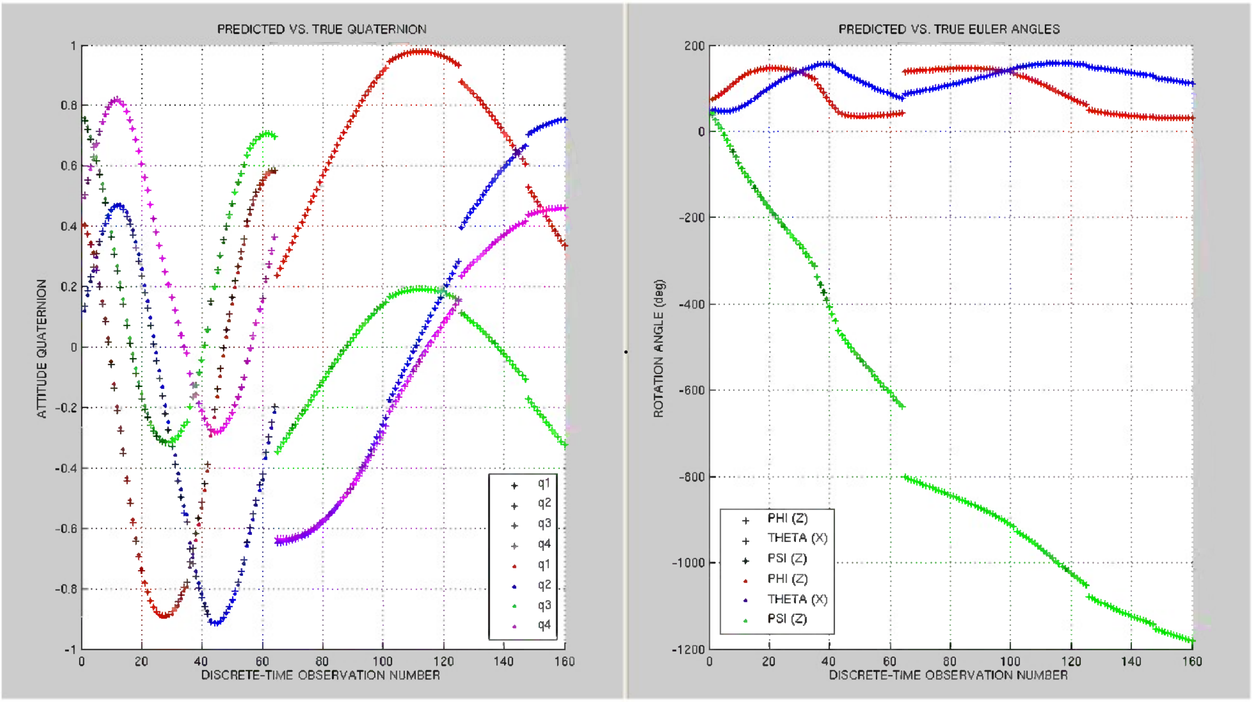


Figure 5.31 The predicted versus true attitude. The predicted values are depicted as colored dots and are obtained from propagating the final state estimate backwards in time. The true values for each quaternion component (left) and Euler angle (right) are designated using the + symbol.

In the subsequent figures the predicted attitude is compared against the noisy and true measurements in order obtain a more objective measure of the accuracy. Over the second pass, the error about each axis of rotation is roughly between ± 2 degrees up until the last 20 observations; at which point, the residuals increase to about ± 4 degrees. This result is in agreement with the measurement error standard deviation which was applied to the end of the pass. A similar result is obtained for the first pass, except that there appear to be much larger systematic errors due to the long propagation time. It is interesting that while the quaternion residuals appear to be more or less centered on zero, ψ is somewhat biased, with a mean value of approximately 3 degrees. Removing the random noise reveals that the systematic errors do not exceed 1.2 degrees for the second pass and 3.5 degrees for the first pass. The total errors for the two passes are less than 3.5 and 1.5 degrees, respectively, confirming that the backward smoothing extended Kalman filter is able to perform multi-pass attitude estimation.

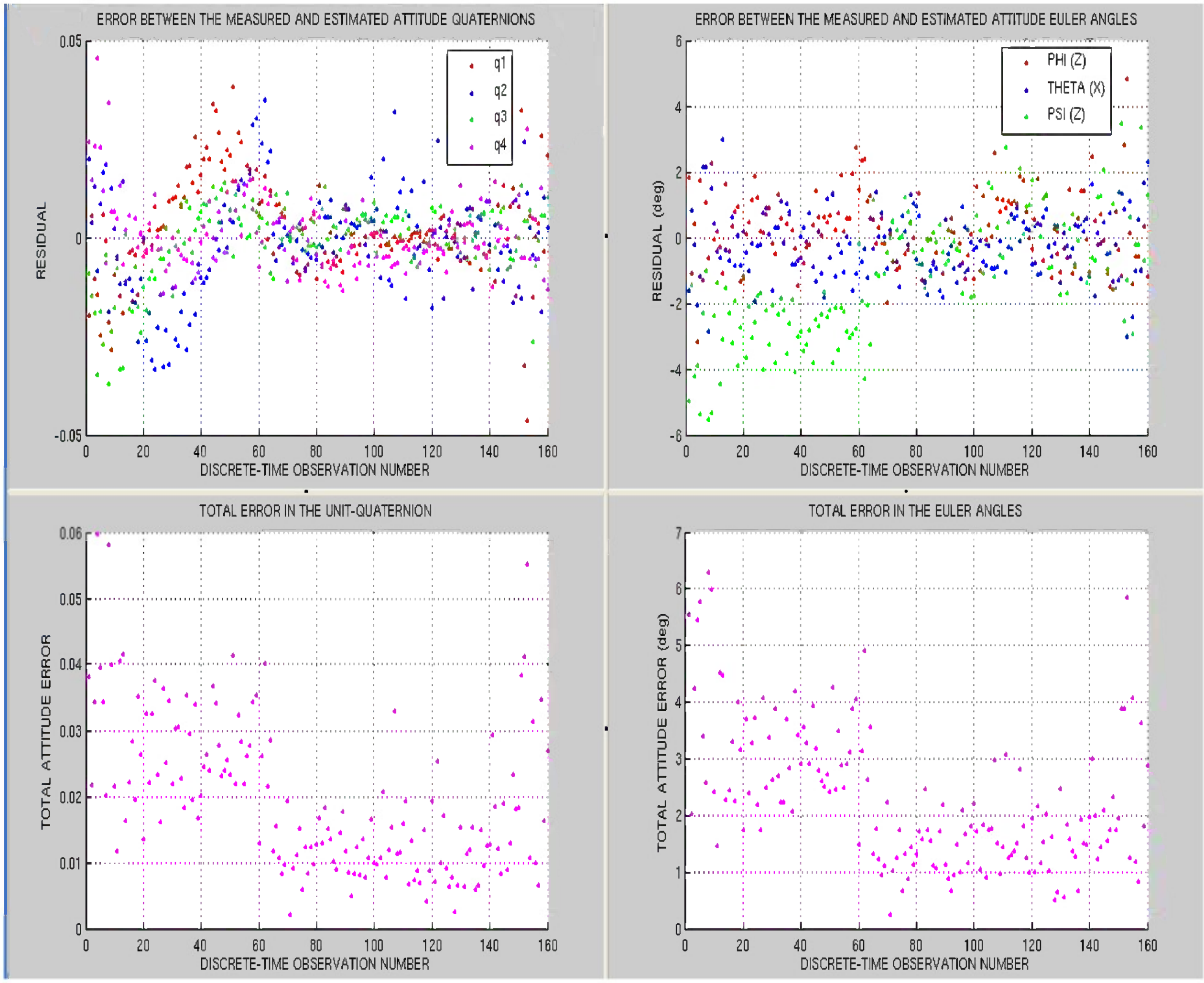


Figure 5.32 Total error (bottom) and residuals (top) for the predicted attitude versus noisy measurements

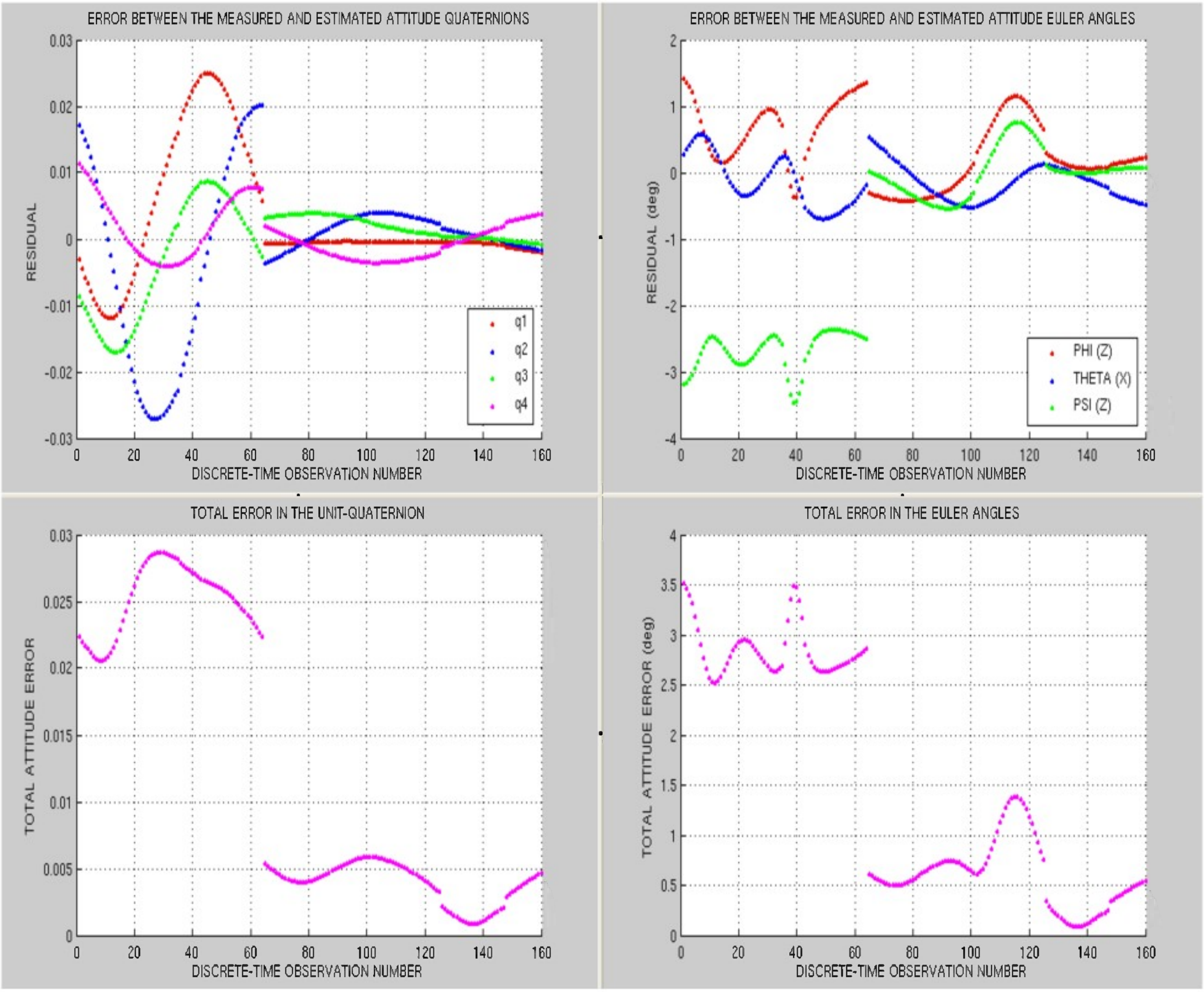


Figure 5.33 Total error (bottom) and residuals (top) for the predicted attitude versus true measurements

5.3.2 Actual Test Case Results

For the first pass of the real-life test case, the filter converges from a relatively large initial error within approximately 15 minutes. However, the BSEKF does not appear to settle into a clear steady-state performance, given that at around observation 35 the residuals being to exhibit larger systematic errors, growing from less than 2 degrees of error to as much as 8 degrees by the end of the pass. After an extensive tuning effort, it is felt that these results do not necessarily indicate a problem finding the right combination of state, measurement, and process noise covariance matrices per se, but rather reflect the inherent challenges in making attitude measurements from radar observations.

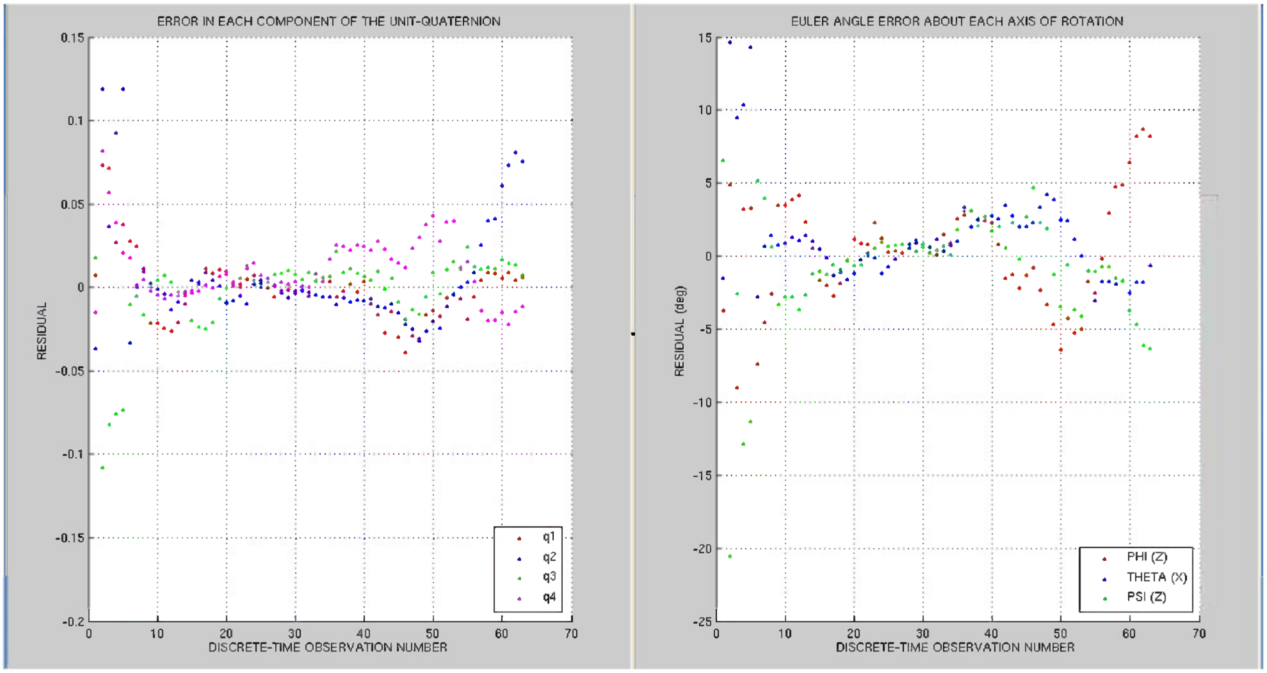


Figure 5.34 Quaternion and Euler angle residuals after processing each observation in the pass

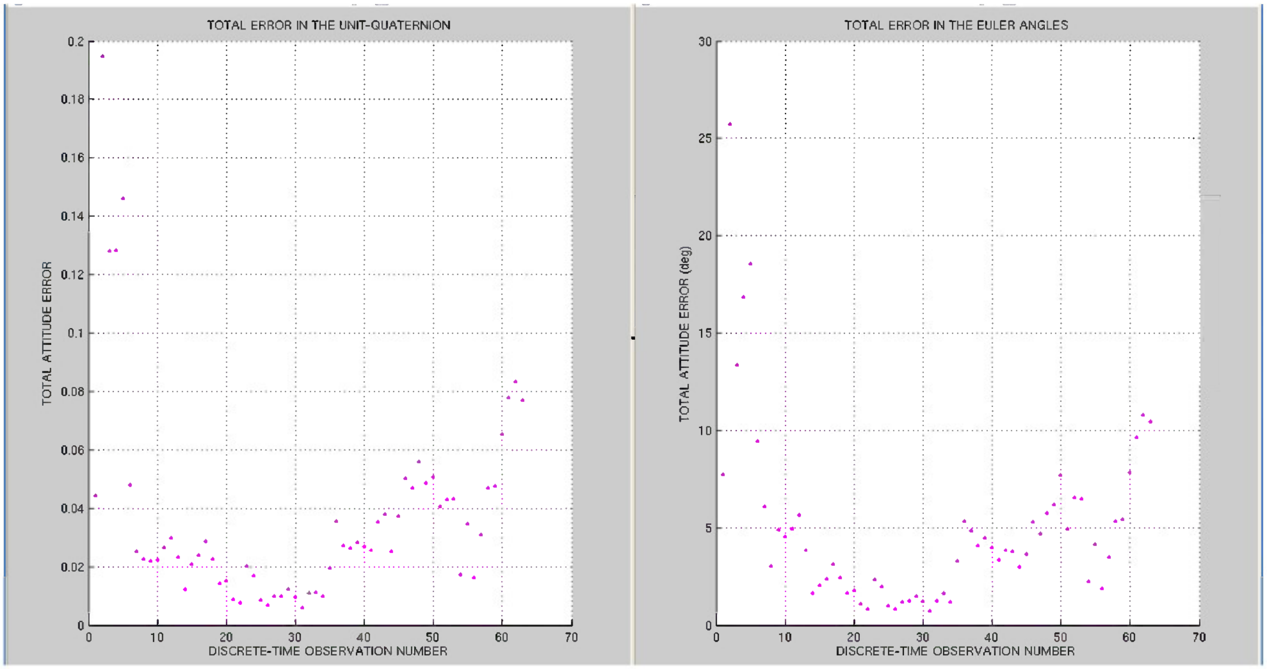


Figure 5.35 Total quaternion and Euler angle error after processing each observation in the pass

From Figures 5.34 and 5.35, it is clear that the greatest amount of measurement noise is present in the observations towards the beginning and end of the pass. However, this result is to be expected considering the manner in which the attitude measurements are made. The factors which contribute most to the increased error in these data points likely include: 1) signal attenuation and/or distortion caused by the fact that the radar pulses are passing through more of the atmosphere when the satellite is at lower elevation angles and greater distances from the ground sensor, and 2) the amount of aspect angle change relative to the radar line of sight tends to be less when the spacecraft is near the horizon, requiring longer integration times to detect the subtle changes in the Doppler frequency. As has already been mentioned in Section 3.1.2.2, longer imaging intervals are more susceptible to errors in the orbital state vector and baseline motion parameters. Unexpected movement of the satellite during the FFT processing time span (imaging interval) results in energy being distributed through multiple resolution cells, distorting the images and further complicating the image-model matching process. Furthermore, when curve fitting via a least-squares procedure, estimates tend to overshoot the actual values in the boundary regions of a given dataset; this effect is typically referred to as the Gibbs Phenomenon when working with Fourier series and other sinusoidal basis functions (a similar oscillation, called the Runge Phenomenon, also arises when fitting higher order polynomials).

Rather than using the straight arithmetic difference, the error can also be expressed as an incremental quaternion, which must be composed with the estimated quaternion in order to obtain the true attitude. Using this method for representing the measurement noise in the system yields the following plot:

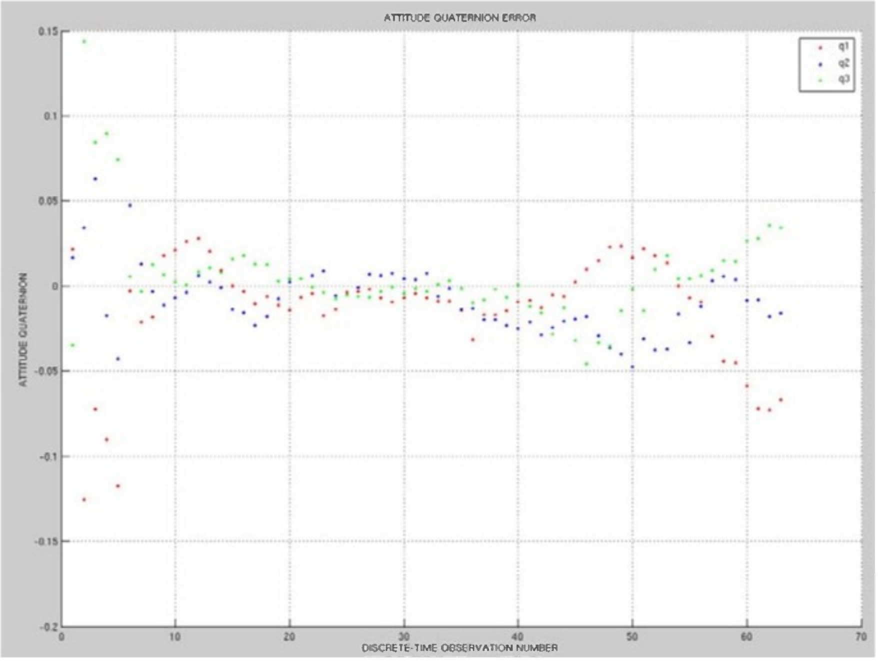


Figure 5.36 Vector components of a quaternion expressing the set of small rotations needed transform the estimated attitude quaternion into the measured value

This is likely the most accurate manner in which to convey the amount of error in the attitude estimate over time, given the emphasis put on maintaining a proper unit-quaternion throughout the filtering process. It is interesting to compare the results of Figure 5.36 with those of Figure 5.34. Expressing the error in this manner provides a slightly different perspective of the filters performance, since the components of the error quaternion begin to diverge much later than the residuals presented in Figure 5.33. Additionally, even after observation 45 – which is about the point where the attitude deviations start to grow – the errors do not appear to undergo as dramatic an increase.

Another valuable measure of filter performance is obtained from plotting the diagonal components of the state error covariance matrix over time. The subsequent set of figures show the amount of uncertainty present in the various state parameters, after processing each discrete observation k in the pass. The results depicted in right most graph of Figure 5.37 indicate that there is a high degree observeability in the angular velocity components, despite the lack of direct measurements. The variances in the angular rates are dramatically reduced from their initial values within the first few minutes of running the filter and appear to reach a steady state performance at 1×10^{-6} (radians/second)^2 by about observation 10. Intuitively, it makes sense that motion of the spacecraft would be very sensitive to even small changes in the angular velocity terms. Similarly, the covariance of the attitude quaternion is decreased by approximately 86% within the first 10 observations and seems to settle around 2.5×10^{-3} midway through the pass. The increase in the covariance of the second quaternion component (indicated by the blue dots on the left side of Figure 5.37) at observation 45 corresponds to the divergence observed in the residual plots. Unlike the other state elements, the uncertainty in the moment of inertia parameters does not reach a steady-state by the end of the pass. This indicates that there is not enough information contained in the attitude observations of a single 20 minute pass to completely determine the inertia tensor of the satellite. The results of Figure 5.38 suggest that p_{I6}, which parameterizes the rotation of a uniform rectangular box about the z-axis, is relatively insensitive to changes in the attitude measurements over time. Conceptually, if the spacecraft is spinning rapidly then, in effect, the mass distributed along the x and y axes is being averaged, making it more difficult to distinguish between p_{I1} and p_{I2}. Again, these particular parameters are the lengths of the sides of a box multiplied by the square root of its mass. Accordingly, it takes a greater number of observations to reduce the ambiguity in the relative ratio of I_1 to I_2.

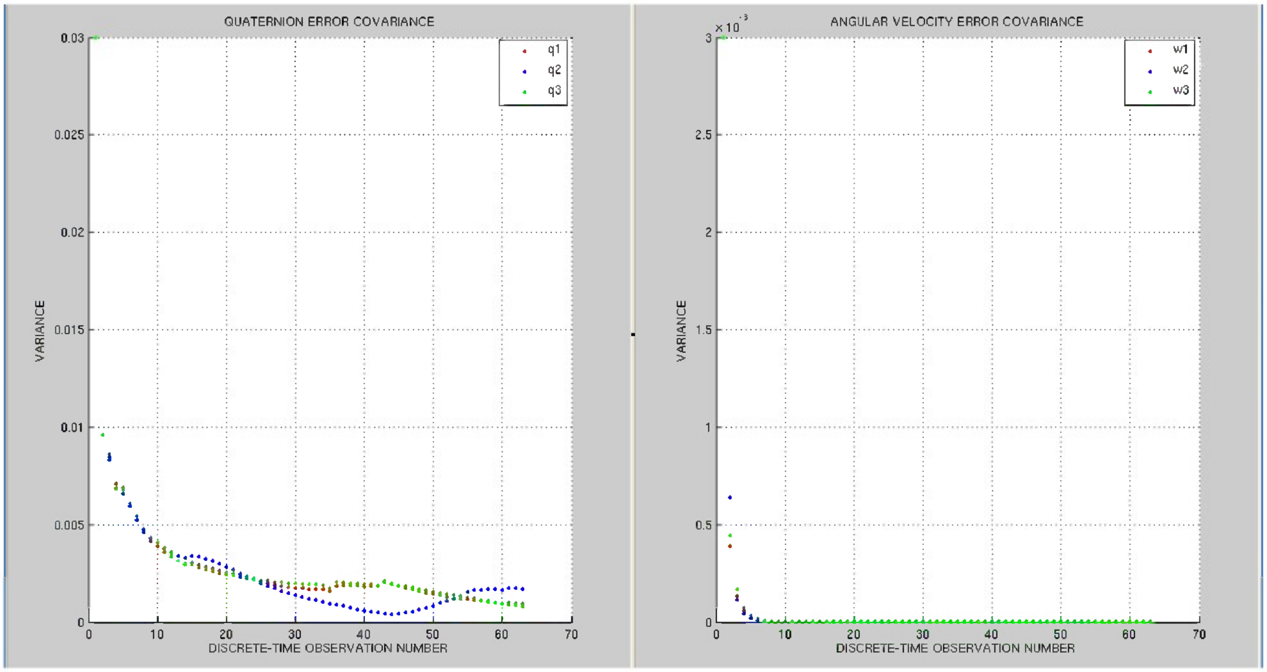


Figure 5.37 Covariance in the attitude quaternion and angular velocity components of the state vector

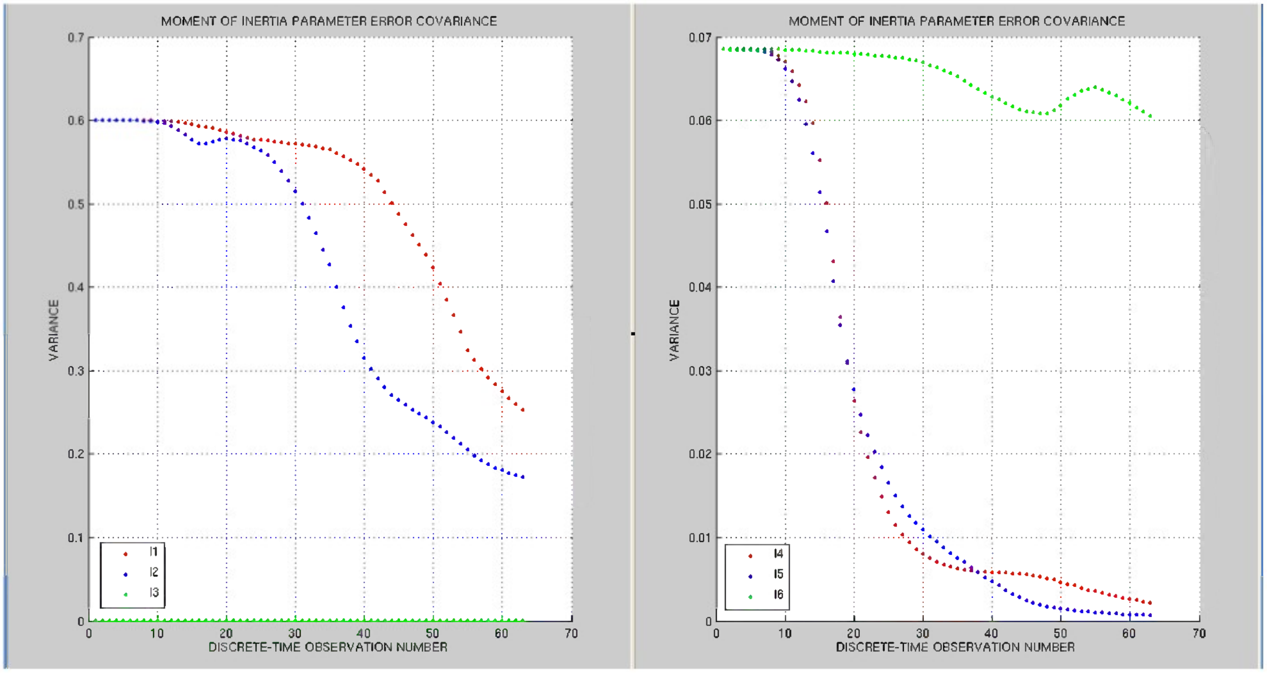


Figure 5.38 Covariance in the moment of inertia parameters

Though the BSEKF only computes first derivatives of its nonlinear dynamics and measurement functions, the algorithm is able to capture curvature effects through the Gauss-

Newton iterations that it uses to solve its nonlinear smoothing problem. It is in these iterations, specifically the forward filter and backward smoothing operations, that information about changes in the attitude work their way into the other state variables. Consequently, if the filter only performs a single Gauss-Newton minimization, the moment of inertia parameters, which are modeled as being constants in the dynamics function, will not be altered/refined from one observation time to the next. If this occurs for multiple observations fairly early on in the pass, the filter "thinks" it has found the minimum and, thus, begins to reduce the covariance. Unless the estimates for the moment of inertia parameters are in fact very accurate, the filter will almost certainly begin to diverge, even if it appears to have reached a steady state performance. In Reference (41), Psiaki concludes that the larger number of Gauss-Newton iterations typically required towards the beginning of a pass, "indicate the smoothing problems are harder to solve when the inertia matrix uncertainties are large...this makes sense because these introduce strong nonlinearities in the estimation problem." In the process of tuning the filter this sequence of events was encountered several times and is a major indicator that divergence is imminent. As has already been mentioned in Section 3.2.3.1, this is likely the case when the state noise has been underestimated with respect to observation noise, as the state estimation procedure will become less and less sensitive to the observation residuals. Resolving this sort of problem is a simple matter of increasing the error covariance associated with the inertia parameters, to reflect the fact that the distribution of mass within the satellite is almost completely unknown.

In addition to filter divergence, other signs that all is not well include abrupt increases in the residuals and imaginary numbers in the state vector. In both cases the source, invariably, has been the presence of discontinuities (denoted by a full sign inversion) in the attitude quaternion, which in turn breaks the constraint that the representation have unity norm to be considered a pure rotation. In the process of implementing the BSEKF, this has been the single greatest reoccurring problem encountered; the reason for this is twofold: 1) the quaternion is not unique, that is, the sign of all the components may be switched and still describe the same attitude, and 2) the fourth quaternion component is not utilized throughout the algorithm in order to avoid singularities in the covariance and transition matrices. In particular, problems seemed to arise in equation (3.108), when applying the state correction $\Delta\mathbf{x}$, used to calculate the candidate next guess of the smoothed solution. During the forward filter and backward smoothing portion of the algorithm the direction and amount by which attitude should be altered is determined; however, because q_4 is not included in this operation it becomes unclear which sign to apply to this specific element when it is close to zero (it is uncertain whether the value is going to pass through, or simply approach, zero). Coming up with a set of rules and tests to ensure that the correct sign gets consistently applied to the quaternion has proven to be a nontrivial endeavor. While a solution appears to have finally been found (see the Matlab source code provided in Appendix D for specific information), the indicators of a discontinuity problem are important to keep in mind, in the event that some new/unforeseen failure mode arises.

Propagating the final state estimate $\hat{\mathbf{x}}_n$ backwards in time to each observation in the pass enables the accuracy of the motion solution to be assessed in two ways: 1) quantitatively, in terms of the residuals, and 2) qualitatively, via a visual comparison of the quaternion components and Euler angles over time. It should also be mentioned that the exactness of the attitude prediction can be evaluated by entering the instantaneous motion parameters, described in Section 4.3.1, into Interactive Motion. In this case, the quality of the image-model alignment that results from applying the unique set of parameters indicates the accuracy of the motion solution. No manipulation of the images or wireframe model should be required if the rotational motion of the satellite has been adequately determined. Because LAES removes the assumption of spin-precession motion, a unique state values has been computed for each discrete observation. However, Interactive Motion applies the motion parameters to the entire pass. Unfortunately, this means that the analyst must manually enter the new parameters for each discrete-time image. This can be a rather time consuming process, given the large number of images which typically comprise each pass. For this reason, in addition to the fact that radar images and wireframe models are not releasable, only the attitude components and residual are presented in the subsequent figures.

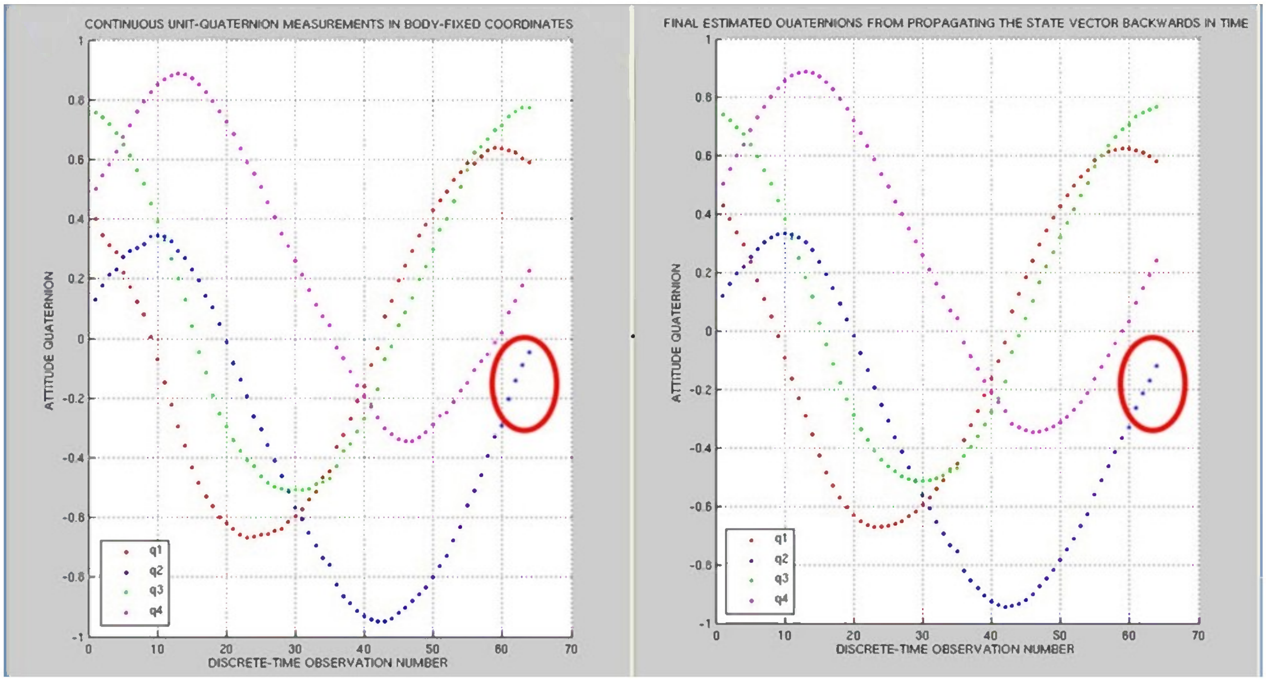


Figure 5.39 The noisy attitude measurement (left) versus the predicted quaternion (right) obtained from propagating the filtered state components backwards over the pass

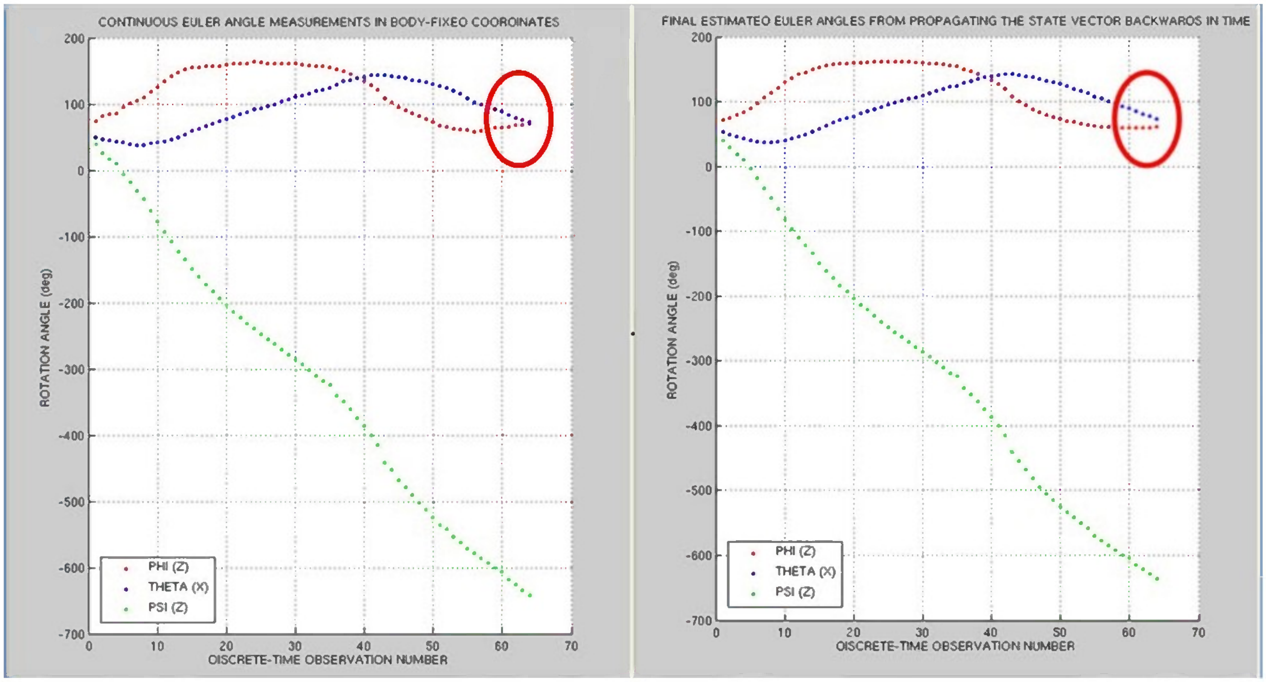


Figure 5.40 The noisy attitude measurements (left) versus the predicted Euler angles (right) obtained from propagating the filtered state components backwards over the pass

From closely examining the end of the pass, one can readily see that the measured and predicted values begin to diverge at around observation 60, with the most noticeable deviations occurring in q_2 (blue points in Figure 5.39) and ϕ (red points in Figure 5.40). Otherwise, the fit appears to be fairly good. The residual between the measured quaternion and estimated quaternion are again plotted for each time of interest, and confirm that by far the greatest difference is in the first and last 5 observations, where the total error reaches 10.4 degrees. Throughout the rest of the pass, the residuals do not exceed ± 4.2 degrees when analyzed in terms of the Euler angles. Unlike the simulated test case, the systematic errors which exist in the data are more likely the result of problems in the measurement process rather than improper filter tuning. Since the resolution and overall clarity of the last 10 images appears to be significantly degraded when compared with the rest of the pass, it is highly possible that the motion predicted by the BSEKF is in fact more correct than what the following residual plots depict. Regrettably, because the true attitude is unknown these suspicions cannot be confirmed; however, the calibration test case suggested in the future works section offers a potential solution to this problem.

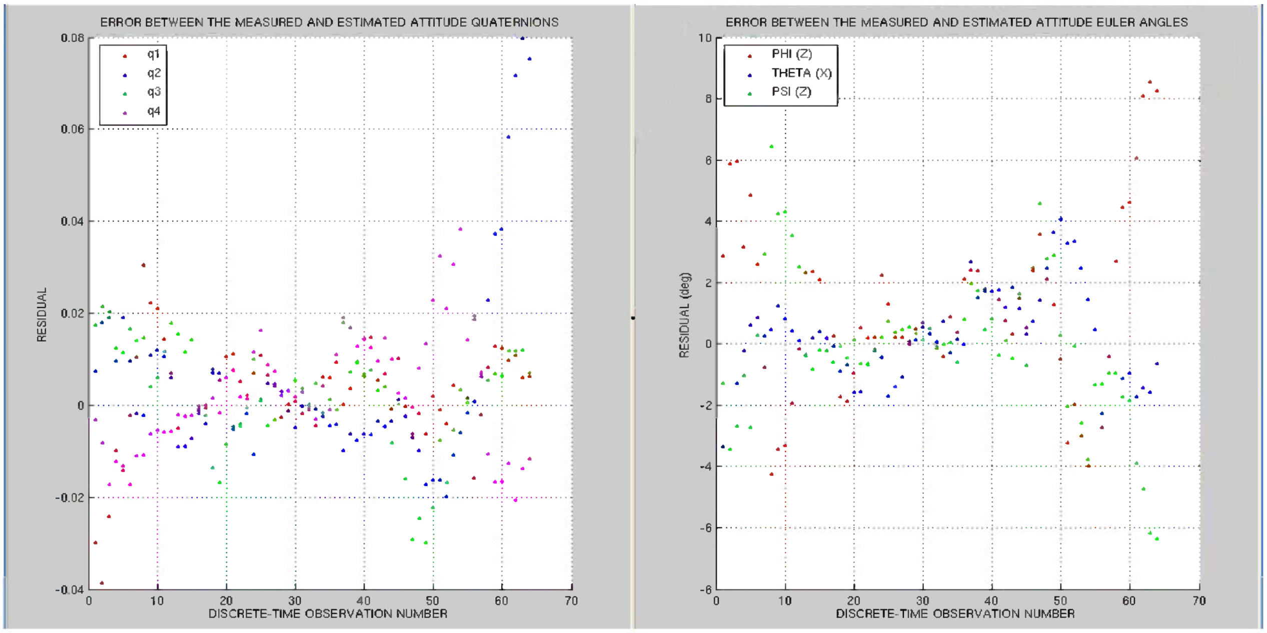


Figure 5.41 Residuals arising from the difference between the measured and estimated attitude quaternion (left) and Euler angles (right) for the first pass

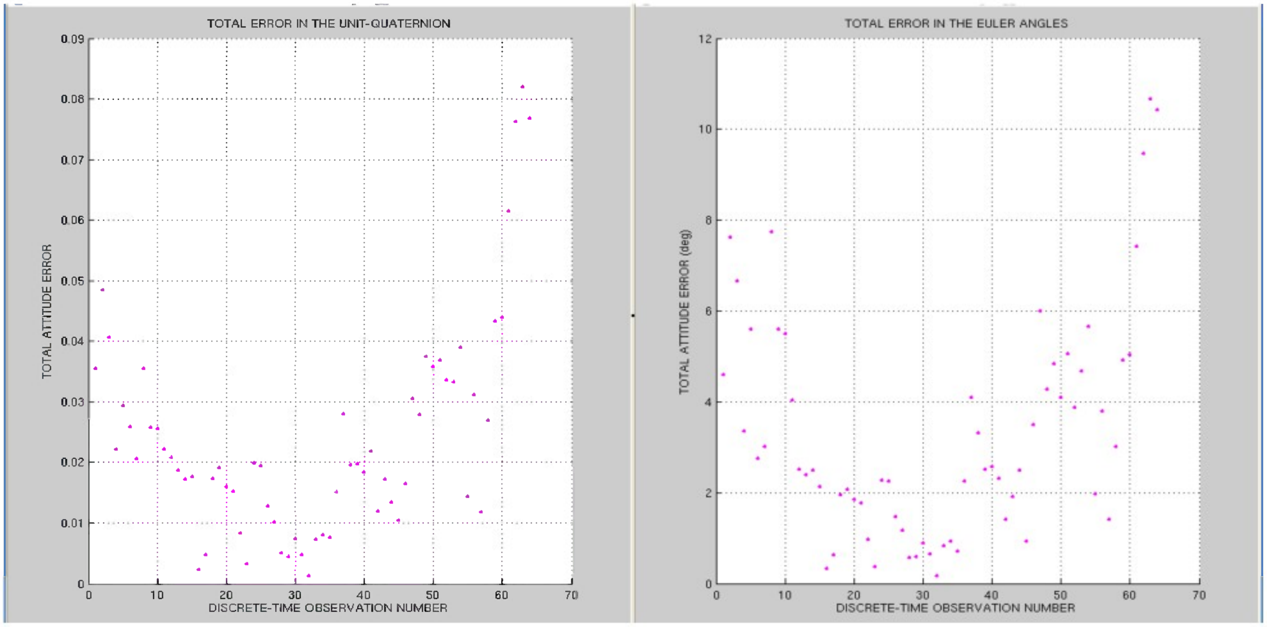


Figure 5.42 Total error in the unit-quaternion (left) and Euler angles (right) for the first pass

In order to compare the results of the BSEKF with those of LLMotion, the following procedure was used:

1. Since the image-model matching process has already been preformed, the wireframe model axes can be used as the basis for assigning measurement points to the radar images. A tool within XELIAS has automated this task. Placing and connecting these points in a consistent manner from image to image results in a set feature vectors, such those illustrated in Figure 5.43.
2. The feature measurements and scaling factors are input into LLMotion and a local search is preformed via the Dynamic Hill Climbing function to obtain a new motion solution.
3. Converting the new baseline motion parameters into a series of attitude quaternions can be done using equations (4.13) – (4.18) and equations $(A.10)$ – $(A.13)$. Subtracting the calculated attitude from the original measurements provided in Figure 5.4, yields the residuals found in Figure 5.44.

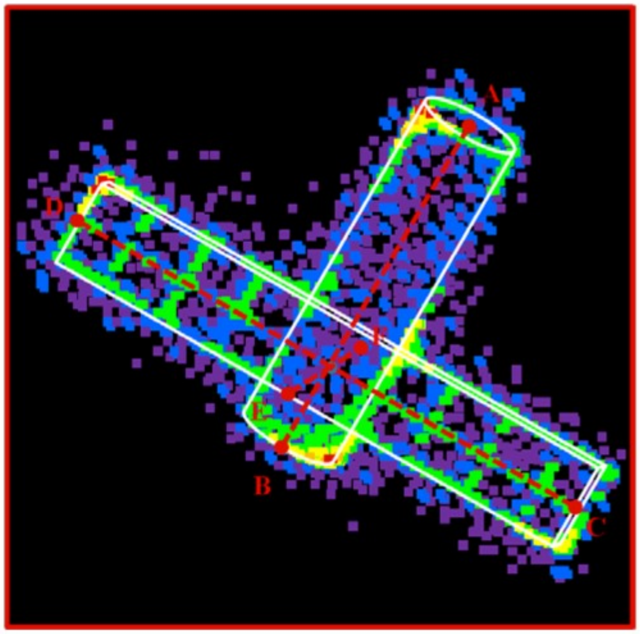


Figure 5.43 Alignment of feature measurements with model axes

Based on the residual plots provided in Figures 5.41 and 5.44, the performance of the BSEKF appears to be slightly better than that of batch least-squares algorithm used by LLMotion. Overall the LLMotion residuals are more dispersed than those of the BSEKF, and have greater peak values ($\geq |10|$deg) towards the ends of the pass. For example, in both methods, the greatest amount of error occurs in observation 63, where the Euler angle residuals for the BSEKF and batch least-squares filter are 8.6 degrees and 12.2 degrees, respectively.

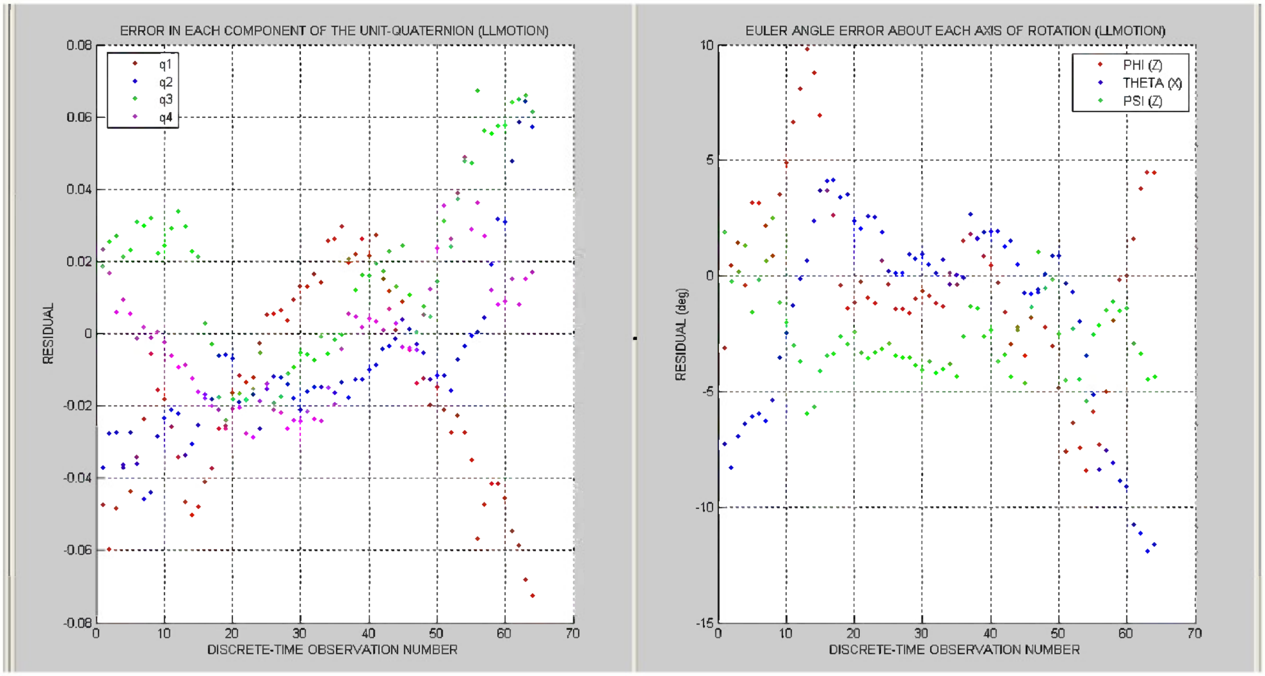


Figure 5.44 LLMotion residuals arising from the difference between the measured and estimated attitude for the first pass

The BSEKF's performance improvements come at a large computational cost. When $m_{target} = 40$, the BSEKF requires between 40 and 60 times as much computation as an EKF for each iteration of its Gauss-Newton nonlinear least-squares solver, and multiple iterations are usually required to solve the problem. The BSEKF has been run with a maximum limit of 15 Gauss-Newton iterations per problem, and the average number of iterations for all the test cases was 8.4 per problem. During the first 30 minutes of filtering, the average number of iterations per problem was higher, between 10.2 and 10.8, with many problems terminating prematurely at the maximum of 15 iterations. During the remaining time, the average number of iterations per problem was between 6.6 and 7.5, with a majority terminating after just two iterations.

The total time needed to process all 64 observations was approximately 1.8 hours. It should be noted that the BSEKF used in the Lincoln Attitude Estimation System has not been optimized for speed. The execution time is highly variable and depends on the following factors:

1. The total number of observations being processed
2. The threshold values used for cost function and guarded Gauss-Newton procedure
3. The maximum number of Gauss-Newton iterations performed
4. The target number of smoothing stages (fixed interval size)
5. The maximum numerical integration step size
6. The number and combination of torque models used in the dynamics function
7. The amount of interpolation required to obtain input values for the torque models
8. The attributes of the computer system running the Matlab files (processing power and memory)
9. Converting the Matlab files into C code can be done using the Matlab *Compiler* and appears to cut the amount of processing time in half [4]

Figure 5.45 shows the amount of time needed to process each observation in the pass. The computation time per observation increases linearly as the size of the m-buffer (smoothing interval) grows. When $k = m_{target}$ the number observations retained in the smoothing interval m has reached the cut-off value of 40 and becomes fixed. At this point, the computation time settles into a steady state performance around 1.75 minutes per observation. The spike in processing

[4] Note: Matlab files were not compiled into C code for the test cases presented in this chapter

time between observations 25 and 35, is the result of an increased number of Gauss-Newton iteration which were preformed as the algorithm began to encounter strong nonlinearities in the measurements.

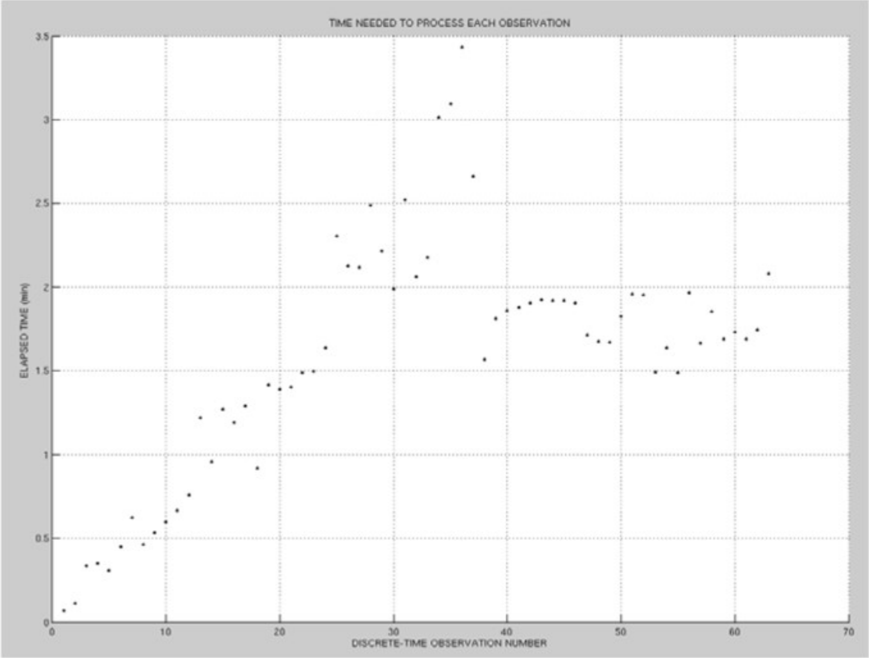


Figure 5.45 Algorithm computation time

5.3.2.1 Actual Test Case – Refined First Pass Measurements

The attitude determination process is more complex whenever data selection based on an a priori attitude is not sufficiently accurate. This occurs, for example, in the presence of smoothly varying systematic anomalies in which some of the data are clearly invalid but presumably valid and invalid data run smoothly together. Attitude determination in the presence of such errors requires iterative processing to obtain successive attitude estimates. The general procedure for this is as follows:

1. Discard "obviously" bad data and use the remaining data to estimate the attitude as accurately as possible.

2. Use the new attitude estimate to reject additional data (or recover previously rejected data) as appropriate.
3. Iterate until a self-consistent solution has been obtained, i.e., when step 2 makes no change in the set of selected data.

This procedure does not establish that the final attitude estimate is correct, or that the data selection has been correct. It is possible that the iterative process will not converge – it may reject all the data or oscillate between two distinct data sets. This method can at best obtain an attitude solution which is consistent with the data selection process. Therefore, whenever problems of this type are encountered, it is important to attempt to find the physical cause or a mathematical model of the data anomaly to provide an independent test of whether the data selection is correct.

The central problem of the above iteration procedure is the data rejection in step 2. Operator judgment is the main criterion used, both because general mathematical tests are unavailable and because the anomaly is unanticipated. Tables of data are of little or no use for analyst identification of systematic anomalies; therefore, data plots are normally required. Four types of data plots have been used throughout this thesis for this purpose: 1) plots of raw data, 2) plots comparing directly the observed data and computed data based on the most recent attitude estimate, 3) plots of deterministic attitude solutions obtained from individual pairs of points within the data, and 4) plots of residuals between the observed data and predictions from a least-squares procedure or similar processing method based on the entire collection of data.

In practice, the author has found that the most likely culprits of erroneous motion solutions are things like the mirror image anomalies (discussed in Section 4.2) and the application of nonphysical motion solutions when preprocessing the data using LLMotion.

LLMotion displays nonphysical minima if there are no true minima or if the non-real motion solution has a smaller objective function value than the smallest true minimum. This failure mode was encountered in the course of this research effort and prompted the inclusion a simple check within the algorithm to ensure that the baseline motion parameters do not produce negative values on the diagonal of the inertia tensor. While negative moments of inertia indicate the presence of a physical impossibility, the mirror image errors represent a physically equivalent motion set due to the physics of range-Doppler imaging. This is demonstrated in Figure 5.46, which shows a scatterer rotating about the angular velocity axis and the resulting radar image plane attached to the center of the satellite.

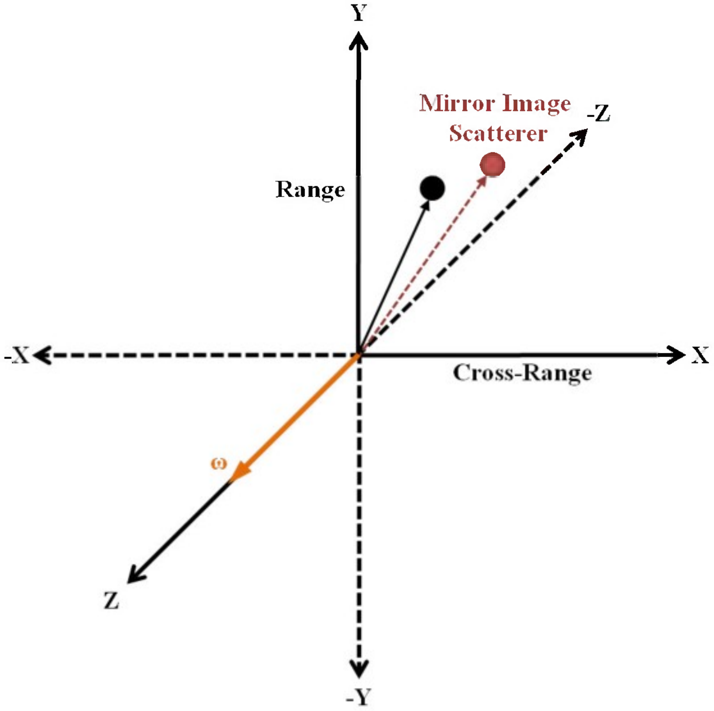


Figure 5.46 Mirror image mapping in the radar image plane coordinate system.

The range-Doppler image projects the scatterer into the (cross-range, range) plane which includes the radar line of sight and is perpendicular to the plane defined by the ROLS and the angular velocity vector $\boldsymbol{\omega}$. An equivalent scatterer, shown in red, also projects to the same point in the plane. This suggests that the view of the image plane from the $-z$ direction will yield an equivalent motion solution. Although the three-dimensional wireframe model for a single image is fundamentally ambiguous under the reverse mapping, the sequence of images reflects a different satellite motion relative to the radar. For example, the counterclockwise rotation of the scatterer is transformed into a clockwise rotation from the opposite direction. Also structures on the satellite, which are not symmetric to a mirror reflection through the image plane, will yield a different 3-D satellite model under the transformation. For this reason, the derived alternative motion solution, obtained from a reverse mapping should be checked by wireframe overlay and multiple pass analysis.

The value of the predicted-versus-observed data plots, as part of the data validation procedure, can be seen in Figures 5.39 and 5.40, which illustrates areas of noticeable divergence in the pass. In order to obtain an accurate attitude solution, the measurements in the highlighted regions must be corrected. As mentioned in the previous section, the larger residuals towards the ends of the first pass are primarily due to a combination of human error and noise in the radar images. The image-model matching process is somewhat subjective, since it is based on the analyst's ability to perceive the spacecraft's orientation within the radar image plane and to finely manipulate the wireframe model. Anything that degrades the analyst's capacity to interpret the attitude of the target from the sequence of images, such as, poor image resolution, the correlation between cross-range scaling and certain rotations, and ambiguity in the geometric projection of scattering centers, all contribute to the measurement noise. Adjusting the alignment of the wireframe in the first and last 15 images resulted in a new set of measurements, which are presented in Figure 5.47.

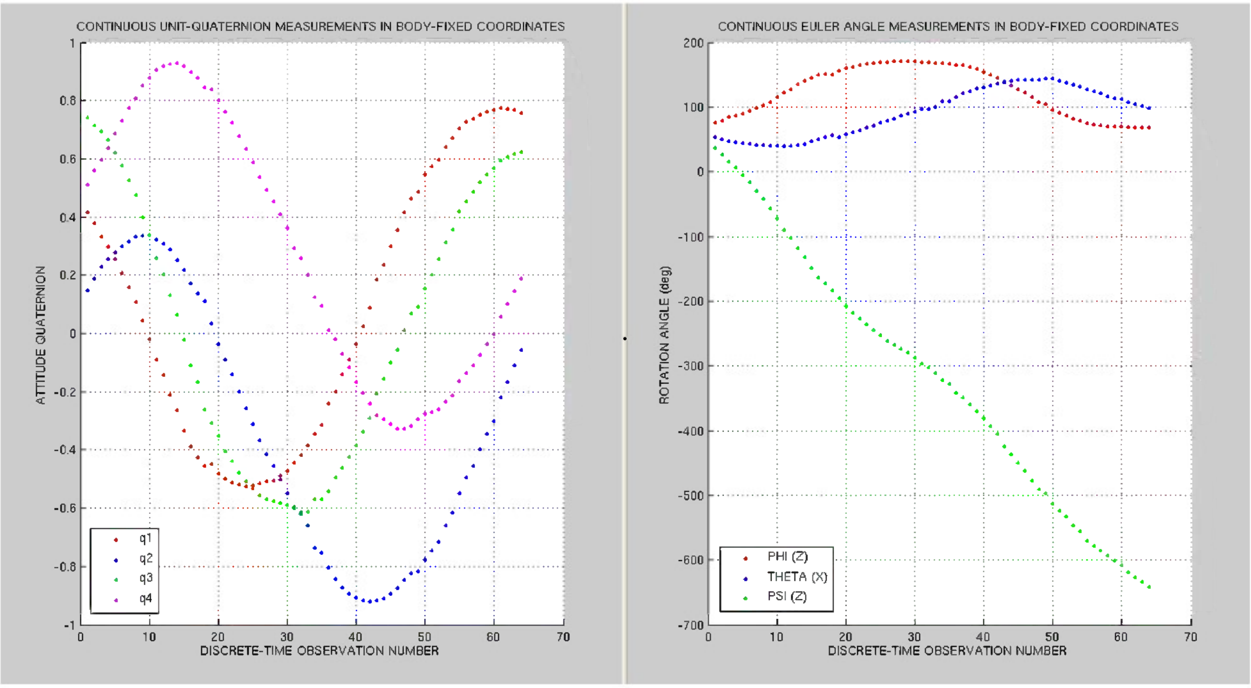


Figure 5.47 Adjusted quaternion (left) and Euler angle (right) measurements for the first pass

The predicted state vector from the first round of filtering was used to calculate the initial state estimate $\hat{\mathbf{x}}_1$ for this test case. The initial state error covariance matrix $\mathbf{P}_1$ that was utilized in this example, is provided in equation (5.12). The performance of the BSEKF, using these refined measurements, is shown in Figures 5.48 and 5.49. When given a fairly accurate initial guess, the filter does not go through a transient phase in which it appears to converge to a steady state performance. Consequently, the residuals predominantly oscillate between ± 3 degrees, with a few values reaching ± 6 degrees, around observations 35 and 45. In the preliminary filtering run, the total error at the end of the pass was approximately 11 degrees. Thought the maximum total error in this follow-on test case is still somewhat high, at 9.8 degrees, the noise in the last 10 observations has been reduced to less than 4 degrees.

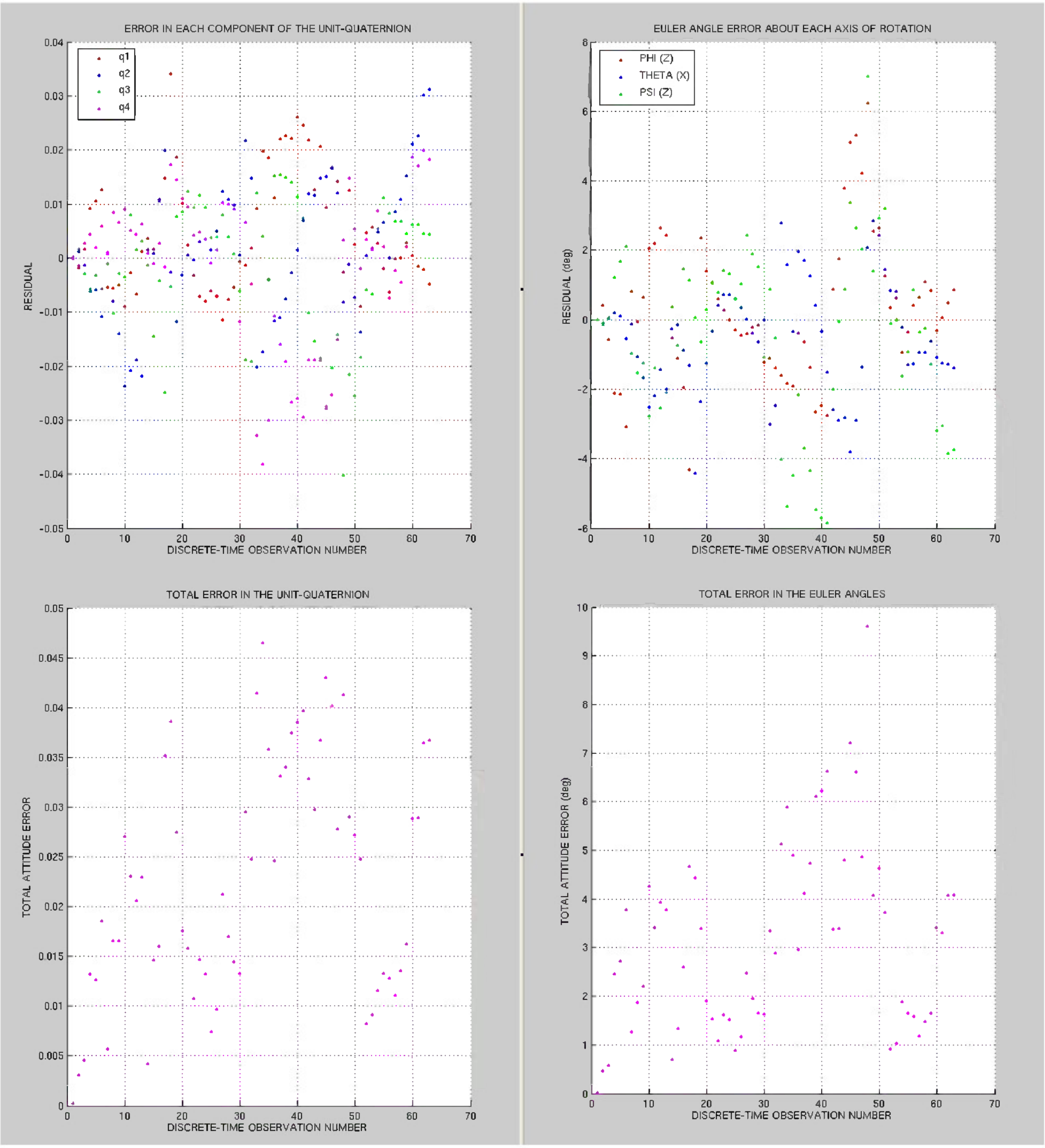


Figure 5.48 Residuals (top) and total attitude error (bottom) for the altered first pass

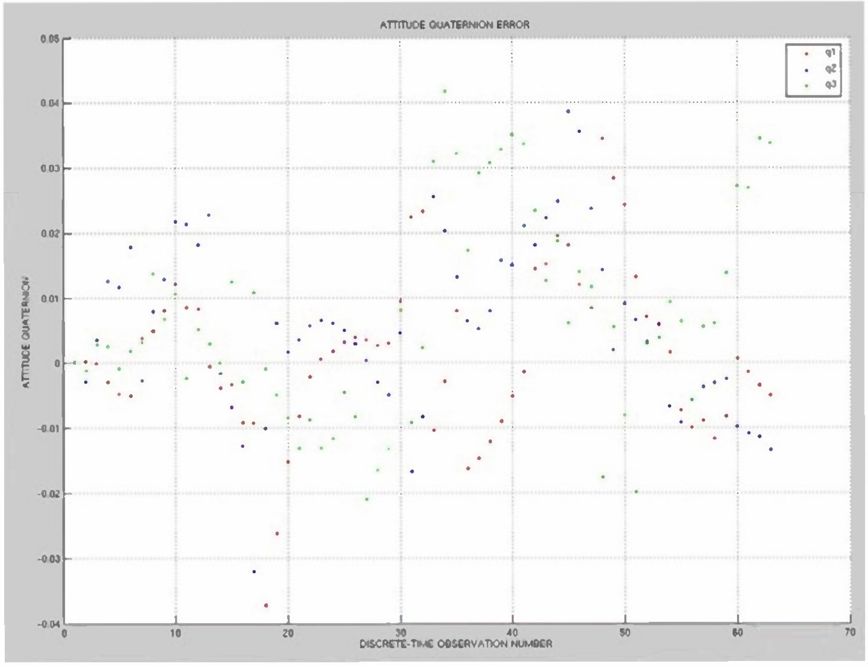


Figure 5.49 Vector components of a quaternion expressing the set of small rotations needed transform the estimated attitude quaternion into the measured value

In Figure 5.47, one can clearly see that the transition between the first and second round of measurements is a bit rough. Plotting the predicted attitude (depicted as colored dots) against the noisy measurements (denoted using black + symbols), confirms that the largest adjustments to the attitude do in fact occur around observations 15 and 49, since these are the boundary points between the original set of Euler angle deviation measurements and the amended set. The peak per-axis attitude errors at these points are around 6.9 and 5.1 degrees, respectively. The total error in the pass does not exceed 8.1 degrees, which is an improvement when compared with the 11 degrees of error seen in the first test run. Additionally, the error in the first and last 10 observations has been reduced to less than 5 degrees, indicating that the adjustments made during the image-model matching have resulted in an attitude estimate which appears to be closer to truth. Indeed, when visualizing the motion solution using Interactive Motion, there is a noticeable improvement in overall alignment of the wireframe model to each image.

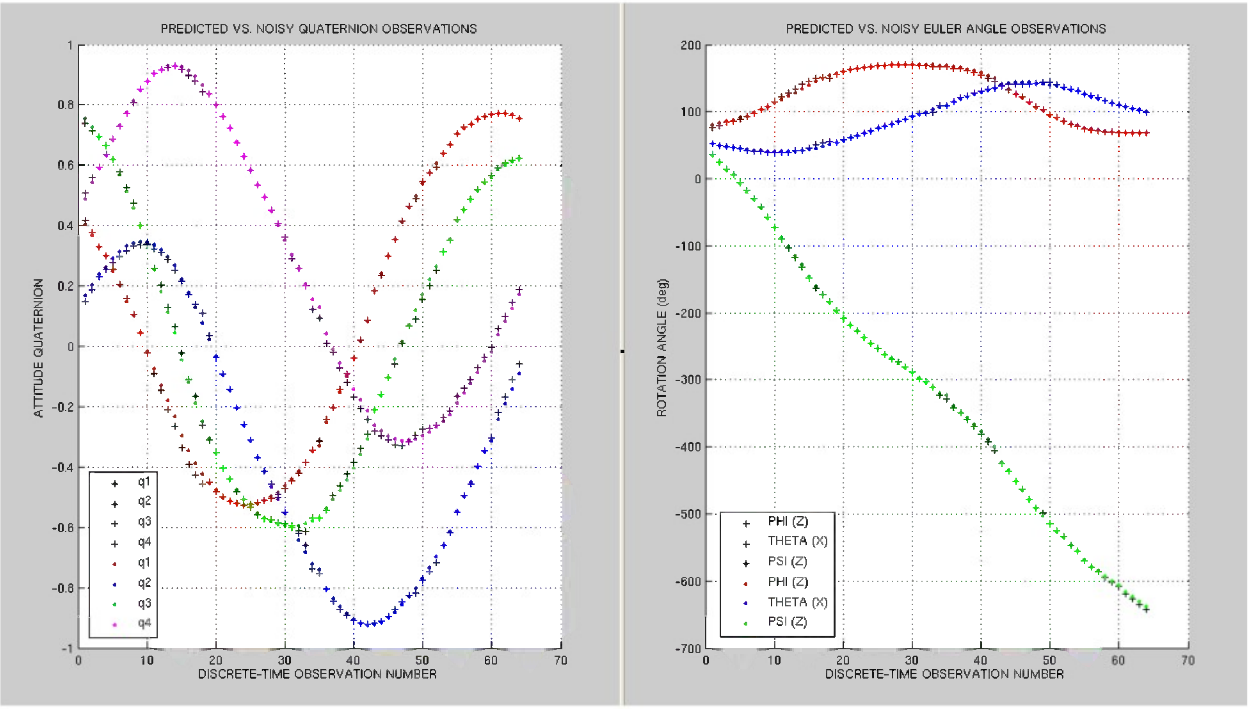


Figure 5.50 The predicted attitude versus noisy quaternion (left) and Euler angle (right) measurements

Table 5-5 Predicted State Vectors for the Refined Actual Test Case

State Value	* Initial State Vector ($\hat{\mathbf{x}}_1$)	Final State Vector ($\hat{\mathbf{x}}_{64}$)
q_1	0.402738275810855	0.761008695080058
q_2	0.167084329810110	−0.089615661178056
q_3	0.755574691127142	0.619412688605402
q_4	0.488867665178345	0.170770959120085
ω_1	−0.000419570565546	−0.000570979038240
ω_2	0.002455347785409	0.002487806650175
ω_3	−0.007230447943850	−0.007282034022134
p_{I1}	3.140482859217543	3.140482859217543
p_{I2}	3.871535464684956	3.871535464684956
p_{I3}	7.746023003273300	7.746023003273300
p_{I4}	0.033085770810414	0.033085770810414
p_{I5}	0.070640011195889	0.070640011195889
p_{I6}	0.644519559782514	0.644519559782514

* Propagating the final state vector (column 3) to the initial observation time results in the state values provided in column 2.

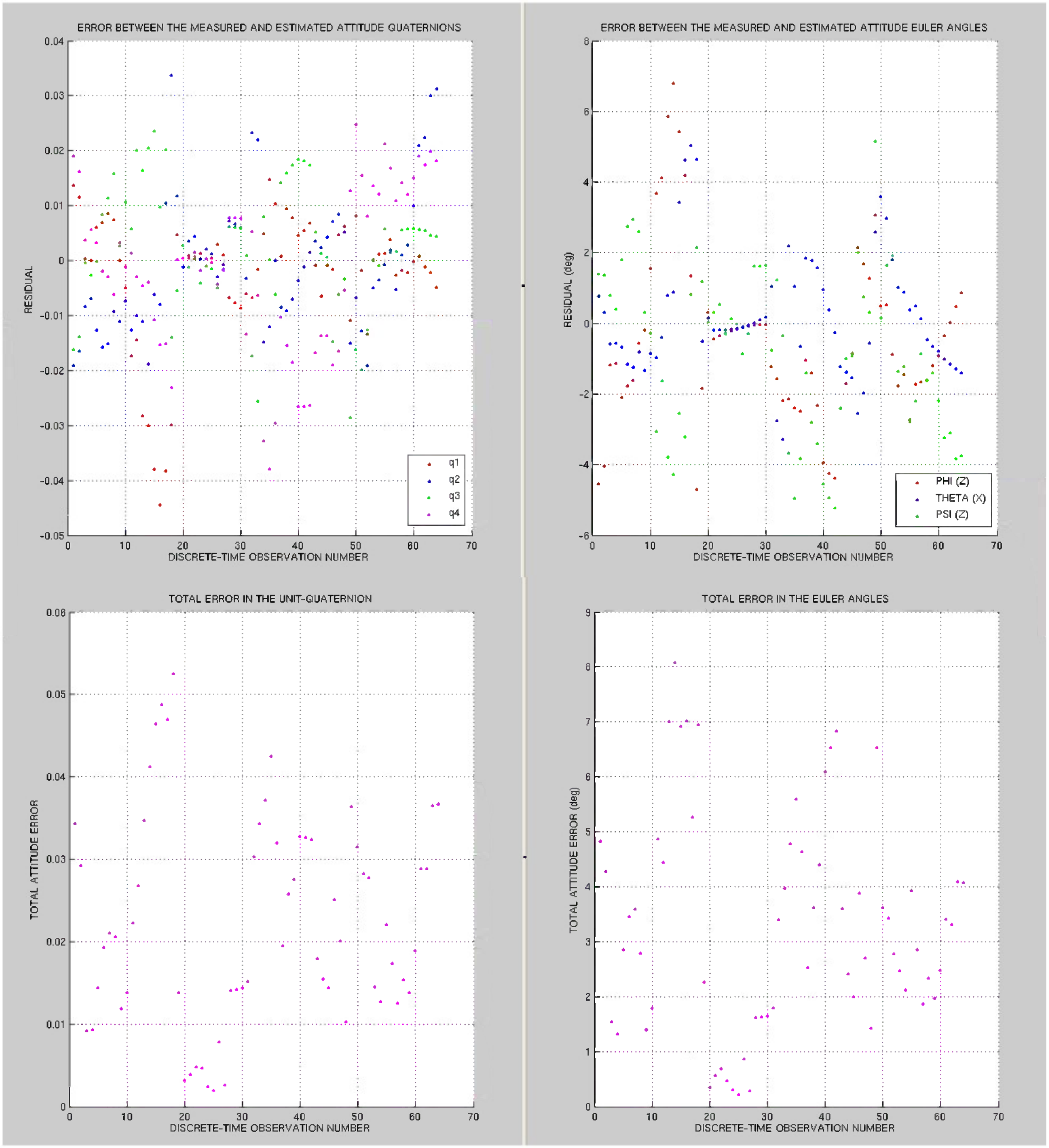


Figure 5.51 Residuals (top) and total attitude error (bottom) in the predicted attitude for the altered first pass

5.3.2.2 Actual Test Case – Long-Term Attitude Prediction

The baseline motion solution obtained from processing the first pass measurements with LLMotion is provided in Table 5-1 of Section 5.1. This set of parameters, was selected from among several other potential minima as the best estimate for the first pass, based on three criteria: 1) the cost function value (FCN), 2) the root mean square value (RMS), and 3) a subjective visual assessment of how well the wireframe aligns with the sequence of radar images after applying the motion parameters to the entire pass. In Figure 5.52, the LLMotion solution has been propagated without external torques to each of the discrete observation times of interest in the two passes.

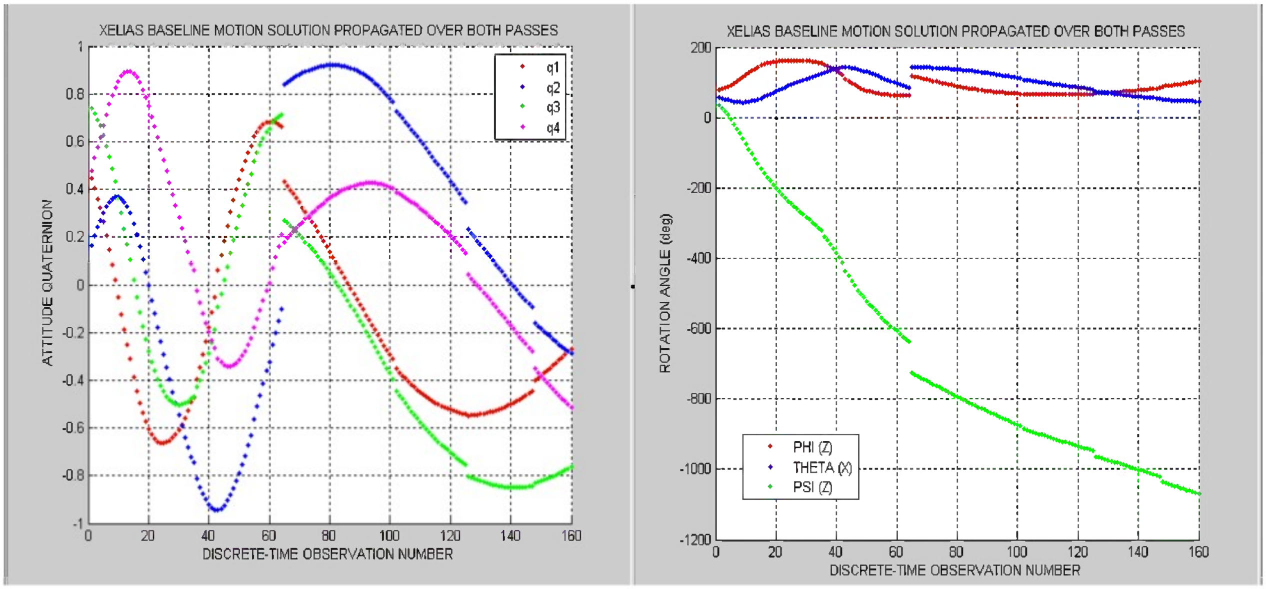


Figure 5.52 Baseline motion solution from XELIAS propagate over the first and second pass

Applying the corrective Euler angle measurements, shown in Figure 5.3, to the baseline motion yields the final set of observations used in the BSEKF (illustrated in Figures 5.4 and 5.5). The results from processing the first pass (presented in Section 5.3.2) indicate that state estimate calculated using the backward smoothing extended Kalman filter is more accurate than the local minimum found using LLMotion; that is, a local DHC search and batch least-squares algorithm. Propagating the final estimated state vector over the 1.88 hour time gap separating the first and second pass results in the following plots of the predicted-versus-measured attitude:

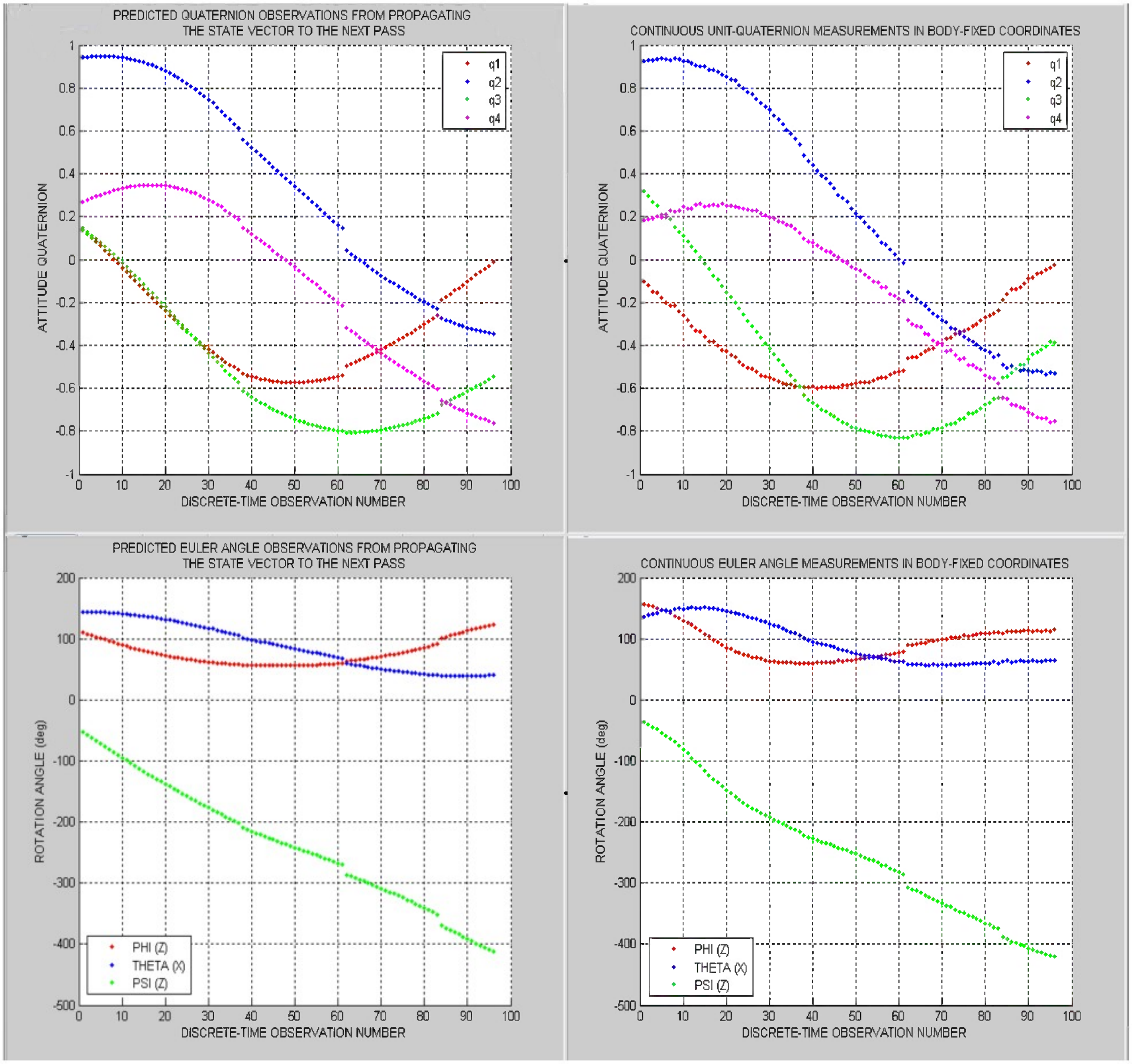


Figure 5.53 Long-term attitude prediction for the actual test case. The noisy second pass measurements (right) versus the predicted attitude observations (left) when propagating the state estimate from the end of the first pass.

While the predicted values, presented in the left-hand plots, follow the same general trend as the actual measurements for the second pass, they also appear to be out of phase and somewhat misshapen. This outcome is yet another measure of the accuracy of motion solution obtained from filtering the first pass and indicates that the true motion of the spacecraft has not been found by the BSEKF. As was seen in the simulated test case presented in Section 5.3.1.1, even small errors in the state parameters produce wildly different results when propagated over long periods of time. It should also be noted that when projecting the state vector to the next pass, only the gravity-gradient torque model has been used. Accordingly, the results presented in Figures 5.53, do not account for the disturbances caused by the Earth's magnetic field, solar radiation pressure, or atmospheric drag. While, the gravity-gradient should be the dominant torque impacting the motion of this particular satellite, the effects of the environmental torques will certainly affect the accuracy of the prediction over such extended time spans. Systematic testing using all the torque models developed for this algorithm is left as a future work item. However, it is felt that the errors in the final state estimate are the principal factor limiting the accuracy of the long term prediction, given the sensitivity of the attitude to even minor changes in certain state parameters. The solution approach used in the multi-pass simulated test case was to provide the BSEKF with both sets of measurements and to allow the algorithm to filter over the time gap separating the two datasets. This method worked very well, yielding an attitude prediction with less than 3.5 degrees of total error over the combined set of passes.

Though the BSEKF is physically able to process the two passes together it is unsuccessful in finding a state estimate which accurately describes the motion of the spacecraft over both. The result is a final state vector that when propagated backwards in time, fits the second pass observations but not the first. Though the presence of environmental torques cause gradual nonlinear changes in the motion parameters over time, such disturbances are not large enough to account for the differences seen in the state estimates from one pass to the next. Though the baseline motion parameters are the same for both passes, the corrections made during the image-model matching process have resulted in attitude measurements which are fundamentally different from one another and, ultimately, lead to two separate local minima. The basic disparity between the two passes is most easily observed in the estimated inertia tensor over time. Though the two matrices should be essentially the same, the side-by-side comparison in Table 5-6 reveals that after processing the second pass, the ratios between each of the moments of inertia were 1.325 to 1.483 times larger than what they were at the end of the first pass.

Table 5-6 Estimated Moment of Inertia Ratios

Moment of Inertia Ratio	First Pass Estimate	Second Pass Estimate
I_3/I_1	0.393	0.583
I_3/I_2	0.412	0.546

1st Pass:
$$\hat{\boldsymbol{I}}_B = \begin{bmatrix} 6.3755 & 0.2647 & 0.0803 \\ 0.2647 & 6.0899 & 0.2682 \\ 0.0803 & 0.2682 & 2.5069 \end{bmatrix} \tag{5.17}$$

2nd Pass:
$$\hat{\boldsymbol{I}}_B = \begin{bmatrix} 5.7710 & -0.1674 & 1.4187 \\ -0.1674 & 6.1640 & 0.4232 \\ 1.4187 & 0.4232 & 3.3673 \end{bmatrix} \tag{5.18}$$

An explanation for why the two sets of measurements do not mesh with one another requires a brief summary of the iterative process utilized in LAES. Essentially, the approach taken in this thesis has been to use the DHC algorithm in LLMotion to divide up the state space and systematically vary each parameter to find the minima which presumably describe the motion of the spacecraft. Based on the corrective measurements made during the alignment process, the BSEKF refines the motion parameters and propagates the estimate to the next pass. Because the attitude measurements have been decoupled from the image plane coordinate system and expressed as corrections to the baseline motion parameters, the inherent ambiguity in the orientation of the image plane is no longer being taken into account. Essentially, this approach erroneously assumes that the attitude of the radar image plane in inertial space is more or less known (i.e., that it was solved for during the preprocessing phase with LLMotion). Thus, when the differences between datasets become larger than the natural fluctuations caused by external torques, problems arise in trying to fit the two passes together. Correcting the BSEKF's measurement model is fairly straightforward and will be discussed in greater detail in Section 6.4.

Performing a global grid search using the Dynamic Hill Climbing (DHC) algorithm in LLMotion, produced the four potential motion solutions given in Table 5-7.

Table 5-7 LLMotion Solutions for the Second Pass

Parameter	Motion Solution #1	Motion Solution #2	Motion Solution #3	Motion Solution #4
FCN	0.0293	0.0586	1.34×10^{-7}	0.0196
RMS	0.0251	0.0354	5.36×10^{-5}	0.0205
MoI Ratio	0.398	0.499	0.45	0.406
$\boldsymbol{\alpha}$	60.9	134.2	101.7	126.8
$\boldsymbol{\delta}$	-8.5	15.0	6.2	9.0
$\boldsymbol{\theta}$	47.0	21.2	34.8	24.7
$\boldsymbol{\phi_0}$	157.2	140.0	157.2	155.7
$\boldsymbol{\psi_0}$	167.9	-163.8	-180.0	-179.2
$\dot{\boldsymbol{\psi}}^{-1}$	1740.4	2235.5	2112.1	2259.4
$\dot{\boldsymbol{\phi}}^{-1}$	1684.0	2389.9	2104.2	2315.0

Plotting the four motion solutions without any corrective measurements reveals the extent to which the attitude can vary and the image-model alignments still agree (see Figure 5.54). While the third motion solution results in the smallest cost and root mean squared error, in terms of the quality of fit when visualizing the motion solutions in XELIAS, the fourth set of parameters seems to be the most correct. All four sets of parameters and corresponding attitude plots appear to be inherently different from that used in the first pass. One can safely conclude that either: 1) the iterative attitude estimation process has not converged to the global minima in this particular test case or 2) physical events onboard the spacecraft, such as fuel slosh, have invalidated the rigid dynamics assumption (i.e., the only torques acting on the vehicle are due to external perturbations).

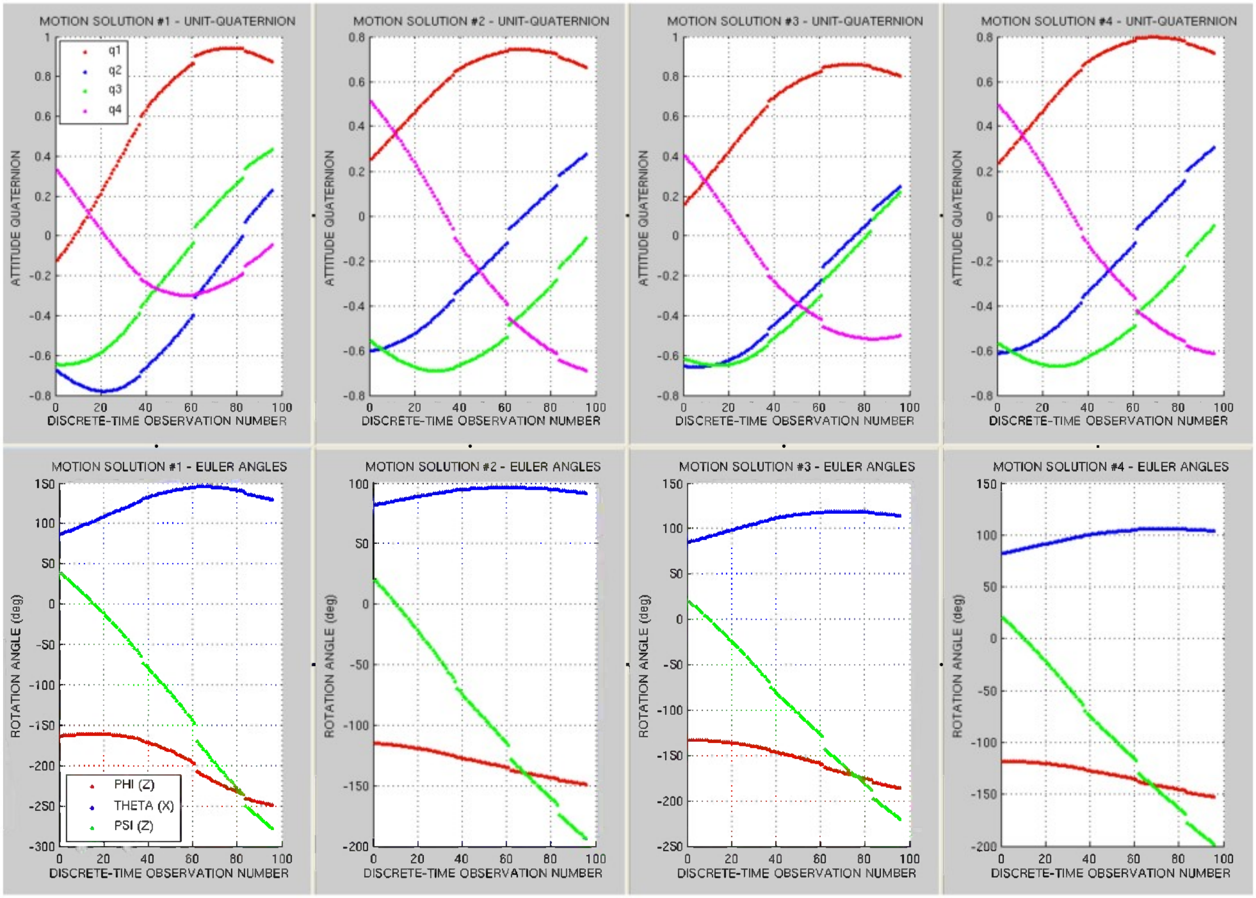


Figure 5.54 LLMotion solutions for the second pass

To be fair, each of the minima found in the global grid search for the first pass would need to be refined, propagated (with all the torque models turned on), and compared with the list of minima found for the second pass. Selection of a baseline motion solution would then be based on the set of parameters which most closely matched those generated for the subsequent pass. However, such a procedure would be incredibly time consuming given the fact that the measurement process is not automated. Because even small errors in the dynamics model and/or state estimate can have a dramatic impact on the accuracy of an attitude prediction, it is unclear to what extent each perspective motion solution would need to be refined in order to achieve a sufficient correspondence with those of the next pass. Though further research on this subject is needed, it is the opinion of the author that the state space is simply too large and nonlinear for this to be a truly viable approach. Automation of the measurement making process is underway, but is a significant effort. A better alternative would be to use a bistatic or multistatic radar system to obtain a unique set of unambiguous measurements describing the attitude of the spacecraft over the duration of the pass. With such measurements, the BSEKF would be able to filter over any number of passes in order to determine the motion of the target.

6 Conclusion and Future Work

This thesis was motivated by the need for improved attitude estimation and prediction capabilities when using radar images as the basis for estimating a spacecraft's attitude. Specifically, this study addressed the issue of incorporating disturbance torques and spacecraft asymmetry into the dynamic equations of motion. In order to accomplish these tasks considerable care has been taken in the selection of a suitable attitude representation and data filter.

6.1 Problem Summary

Since only a small fraction of an orbit is visible from ground-based sensors, limited attitude information can be collected and used in estimating a target's rotational motion. The situation is made worse by the fact that the number of independent images which can be produced is restricted by the amount of aspect angle change in the spacecraft relative to the radar line of sight. Additionally, given the nature of the radar measurements themselves, the orientation and angular velocity of the spacecraft can only be determined up to a rotation around the RLOS. This uncertainty, in conjunction with the error introduced from trying to align the projection of a three-dimensional model with a noisy two-dimensional image, makes the non-cooperative attitude estimation problem incredibly challenging. On top off all this, information about the mass distribution of the vehicle is completely unknown and the measurements and dynamics of the system can be highly nonlinear. Ultimately, the entire method is reliant on human powers of perception and judgment not only in terms of the measurement procedure (image-model matching), but also in the data validation and motion evaluation processes.

Given the large amount of ambiguity and noise, the solution approach used in this thesis has been threefold. First, an iterative estimation process is needed to converge on the motion of the satellite of interest and gradually refine the imagery upon which the attitude determination is based. Second, the existing software program known as LLMotion is used to narrow the solution space and preprocess the motion estimate by removing large initial scaling errors and reducing deviations between the geometric model and underlying imagery. LLMotion operates under the assumption that there are no disturbance torques acting on the vehicle and that the spacecraft is a symmetric rigid body with no articulating or flexible surfaces. The software system utilizes the Euler angle attitude parameterization, as it is well suited to a spin-precession motion description and closed-form motion propagation. LLMotion identifies potential motion solutions (i.e., local minima) via a batch least-squares procedure, in combination with a grid search algorithm which systematically analyzes an extensive state space. Third, a backward-smoothing extended Kalman filter with additional torque models is used to filter the remaining measurement noise and overcome nonlinearities in the measurement and dynamics equations. This method relinearizes the current and past measurement and dynamics functions about improved guesses of the current and past state and process noise vectors. Appropriate relinearization points are chosen via iterative numerical smoothing over an interval of time that ends at the current sample time. This process results in a state estimation algorithm that treats all of the nonlinearities over a number of stages without any approximation. Additionally, by simultaneously estimating the moment of inertia parameters, the BSEKF is able to compensate for the uncertainty which results from sensing fewer than three axes and having little to no information about the mass distribution of the vehicle. It is also important to reiterate that the BSEKF was selected over a batch least-squares algorithm so that measurements from follow-on passes could be filtered as they became available. Thus, greater operational flexibility could be achieved by allowing an analyst to easily join observations made over several passes and obtain a new state estimate after each observation is processed. Other critical design decisions include the numerical integration of a system of ordinary differential equations expressing Euler's equations of motion (as opposed to the use of elliptical integrals) and the use of the attitude quaternion whose kinematic equation is linear and which satisfies only a single, easily enforceable constraint.

6.2 Modeling Enhancements

The Lincoln Attitude Estimation System is a new tool which has been developed for the Space Situational Awareness Group at MIT Lincoln Laboratory, in order to accurately estimate the rotational motion of an uncontrolled satellite over time. The system has modified a backward-smoothing extended Kalman filter to produce an algorithm that can be integrated with the existing systems currently in use at Lincoln Laboratory. The major features which have been added to the BSEKF, as presented in Reference (41), include:

1. The addition of environmental torque models for the Earth's magnetic field, gravity-gradient, atmospheric drag, and solar radiation pressure.
2. A geometrically derived initial estimate for the inertia tensor that removes the assumption of symmetry in the principal moments of inertia and includes estimates for the off-diagonal products of inertia.
3. A complete quaternion parameterization that maintains the unity norm requirement while preventing discontinuities in the measurements and attitude estimates, as well as singularities in the state covariance and transition matrices.
4. Conversion functions which transform between spin-precession motion parameters and the 12 × 1 BSEKF state vector given at the beginning of Section 4.3.2, in order to properly interface with the existing software systems used by the SSAG.

5. Automated generation of initial estimates for the state vector, state error covariance matrix, and measurement noise covariance matrix by means of polynomial fitting algorithms.

6.3 Data Analysis

Test results for a simulated and actual non-cooperative attitude estimation problem confirm that the BSEKF can be used to filter the noisy measurements produced from fitting geometric (wireframe) models to discrete-time radar images. By retaining the nonlinearities of a fixed number of sample intervals before the time of interest and approximating the effects of earlier sample times' dynamics and measurement nonlinearities, the algorithm is able to achieve considerable accuracy and convergence reliability despite the large uncertainties that arise when measuring spacecraft attitude using ground-based sensors. For the truth-model simulation, the 41% initial moment of inertia estimation error converges to 1.6% by the end of the pass and the total attitude error was reduced to 1.8 degrees after processing all the noisy observations. For the actual dataset, the filter converged to a steady-state performance at around 5 degrees of attitude error but finished the pass with more than 10 degrees of total attitude error. This, however, is not a problem with the filter, but rather with the measurements. Adjusting the Euler angle deviation measurements at the beginning and end of the pass enables the total error to be reduced to around 4 degrees.

Both test cases demonstrate that the BSEKF is ideally suited for use on non-cooperative attitude estimation problems. However, the results obtained from this research effort also suggest that a 10 or 20 minute pass from a single sensor provides an insufficient number of observations to determine the motion of the spacecraft to the level of accuracy needed to perform precise long-term attitude prediction. The error in the measurements is several orders of magnitude greater than that introduced by ignoring environmental disturbance torques. Nevertheless, the BSEKF has demonstrated the ability to filter over multiple passes (i.e. long time gaps). Thus, the potential exists for refining the state vector using observations strung together from multiple short duration imaging opportunities. In this way, an estimate of the inertia tensor can be built-up over time, enabling the orientation of a vehicle to be predicted with greater and greater accuracy as more datasets are processed. In the multi-pass simulated test case, the total attitude error which resulted from propagating the final state estimate backwards over the combined passes was less than 3.5 degrees for the first dataset and under 1.5 degrees for the second. On the other hand, for the actual test case, significant differences in the attitude measurements from pass to pass prevent the BSEKF from successfully filtering the combined dataset. Essentially, the BSEKF begins to converge on a particular motion solution during the first pass, goes through a second transient phase as it propagates over the time gap, and then settles into an entirely different local minimum by the end of the second pass. The resulting set of parameters does not consistently describe the motion of the spacecraft over both passes. This indicates that that there is a problem with the measurement model which needs to be resolved before further testing can begin.

6.4 Unresolved Issues

A major focus of this thesis has been to address some of the many difficulties posed by the non-cooperative attitude estimation problem. The major issues which have yet to be fully resolved involve the measurement sub-process and are a direct consequence of the complicated imaging geometry (dynamics of the satellite relative to the radar station) and limited number of measurements which can be obtained from each two-dimensional radar image. In the course of writing this thesis it has become apparent that the measurements cannot be decoupled from the image plane coordinate system for several very important reasons. First, the measurement errors may no longer be Gaussian due to the fact that a series of nonlinear transformations are needed to go from the body-fixed reference frame to the radar image plane coordinate system. Because the orientation of the image plane in inertial space is not entirely known, the measurements cannot be separated from the image plane without making some assumptions about the motion of the spacecraft, which may or may not be true. The current approach used in LAES does just that, measuring the attitude of the spacecraft with respect to an image plane whose orientation is inherently unknown. Second, the measurement errors may not be independent, since corrections to the baseline motion parameters are made through a combination of cross-range scaling and model rotations. There is a very complex nonlinear relationship between the measurements made during the image-model matching process (i.e., the cross-range scaling factor and 1-2-3 Euler angle rotations), and the nominal motion of the spacecraft. For example, a change in the scaling term impacts both the perceived true attitude of the target within the radar image plane and the direction of the target's angular velocity vector. Consequently, it is unclear how much of the error can be attributed to scaling issues and how much is tied to deviations in the attitude.

In order to solve the non-cooperative attitude estimation problem more measurements are needed and can be obtained in one of two ways: 1) stereoscopic radar images from a bistatic or multistatic radar system or 2) use measurements from multiple passes. Modifying the BSEKF observation model so that it is no longer tied to a set of baseline motion parameters, would enable LAES to completely replace LLMotion. In the procedure described below, the measurements are made with respect to the radar image plane coordinate system. In order solve for the attitude of the spacecraft, the orientation of the image must be estimated simultaneously.

The difference in imaging geometry from one pass to the next should help to resolve the uncertainty in the attitude of the image plane by providing a difference in the RLOS direction.

The following modifications need to be made to the BSEKF measurement model in order to ensure that the measurements are independent and normally distributed:

1. During the image generation process, that number of discrete-time radar image output by ARIES should be entirely based on the amount of aspect angle change over the duration of the pass and not the amount overlap between images.
2. After aligning the wireframe to all the images in the pass, use the model's axes as the basis for selecting and connecting a set of six points in each radar image. Because these are the same feature measurements used by LLMotion, they are automatically saved in the *paramcdf* file. Each image i in the pass contributes j measurement vectors of the form:

$$\mathbf{y}_{ij} = \begin{bmatrix} ms\_xcoord \\ ms\_ycoord \end{bmatrix} = \begin{bmatrix} y_x \\ y_y \end{bmatrix} \tag{6.1}$$

   where y_x is the x coordinate of the measurement point in Hertz and y_y is the y coordinate of the point in meters. The scaling factors needed to convert the x coordinates from Doppler to meters are contained in the *imagecdf* file under the variable name x_scale. Both files can be obtained from XELIAS by following the instructions provided in Appendix C.
3. The observation model $h(\mathbf{x})$ transforms elements of the state vector into a form which is equivalent to the measurement vector. In this instance feature vectors in the spacecraft body are rotated and projected into the image plane via the attitude quaternion and angular velocity vector. The quaternion expressing the transformation between the body-fixed coordinate system and the inertial frame is currently used throughout the BSEKF

and can be easily converted into a rotation matrix $\boldsymbol{A}$ using equation $(\mathrm{A.5})$. A series of rotations, given by equations $(\mathrm{B.17})$ – $(\mathrm{B.20})$, are then required to transform from the inertial frame (I) to the image plane. Combining the attitude matrix $\boldsymbol{A}$ with this sequence of rotations results in the following expression:

$$[\mathbf{y}_{ij}]_{IP} = \boldsymbol{T}_i[\mathbf{y}_j]_B \tag{6.2}$$

where $\boldsymbol{T}$ is the rotation matrix which relates feature vectors in the body-fixed coordinate frame (B) to those in the radar image plane coordinate system (IP). Note that since this is the same matrix used by LLMotion, an initial estimate for the state vector can be calculated using equations $(\mathrm{B.11})$ – $(\mathrm{B.20})$. The values needed to calculate the preliminary rotation matrix $\boldsymbol{T}_1$, given by equation $(\mathrm{B.21})$, are contained in the *imagecdf* and *motioncdf* files (see Table C-1). It is important to recognize that as the imaging geometry changes, only the sequence of rotation need to go from $I \rightarrow IP$ are impacted. Thus, the change in RLOS from image-to-image and pass-to-pass is now properly being accounted for.

4. The measurement transition matrix $\mathbf{H}_i$ is then given by equations $(\mathrm{A.30})$ – $(\mathrm{A.33})$, which are the derivatives of a rotation matrix with respect to the components of an attitude quaternion. Note that since the final rotation about the RLOS is found using the angular velocity vector, the partial derivatives of the rotation matrix with respect to the components ω_x and ω_z will also be needed.

6.5 Future Work Items

This section discusses several areas of future research which are related to this study. Generally speaking, the future work items fall into two categories: those related to resolving problems with the measurement process and those concerned with further improving and validating the filtering process.

6.5.1 Systematic Torque Model Testing

A major aim of this thesis has been the inclusion of the environmental torque models in Euler's equations of motion. However, only the gravity-gradient model has been consistently included in all the test cases which have been presented in the results section of Chapter 5. The reasons for this include: 1) given an orbital altitude of over 1000 km, the gravity-gradient was assumed to be the predominant disturbance torque acting on this particular satellite, 2) the accuracy of the model was able to be validated by comparing the torque generated using the model presented in Section 2.3.4.3.1 against that produced using source code provided by Dr. Mark Psiaki [5], 3) a direct relationship exists between the attitude of the spacecraft and magnitude and direction of the torque (i.e., in addition to the state vector, only a single orbital parameter is needed to compute the perturbation), 4) the accuracy of the other models could not be verified due to a lack of truth data against which to compare the results, 5) the aerodynamic and solar radiation models can impose a significant computational burden on the system depending on the level of detail included in the geometric model of the spacecraft, and 6) the magnetic, aerodynamic, and solar radiation torques all require a number of variables (such as the drag coefficient, magnetic dipole moment, and surface reflectivity) which cannot be observed through ground-based radar measurements and, therefore, must be assumed. Consequently, there is much work that could be done to systematically turn on the other three torque models which have been developed in order to evaluate their impact on the propagated attitude estimate. The challenge

[5] The Matlab source code, graciously provided by Dr. Mark Psiaki of Cornell University, was used to validate and refine the gravity-gradient torque model, state transition matrices, and numerical integration technique used in the BSEKF.

will be in the acquisition of accurate and reliable truth data against which model outputs can be evaluated.

6.5.2 Cooperative/Non-cooperative Calibration Test Case

The potential exists for coordinating with the owner/operator of a given satellite system, to obtain onboard attitude telemetry over the same time span as the remotely sensed attitude observations. The goal would be to use a satellite with known attitude time history and moment of inertia properties time history to calibrate the radar-based attitude determination process. Such a test case would provide invaluable truth data against which the results of the Lincoln Attitude Estimation System could be compared in order to determine the source of the error limiting the accuracy of the algorithm: improper filter tuning, faulty attitude measurements, imprecise torque models, or some combination of all the above. The calibration test case would be based on a spacecraft for which the following conditions could be met:

1. Obtain an owner-operator attitude determination based on onboard sensors, making measurements over a predetermined time span. The inertia tensor, surface material properties, and dimensions of the spacecraft would also be of great value in developing an accurate computer model for the target.
2. Obtain the non-cooperative attitude estimate based on the Lincoln Laboratory ground-based sensor measurements, image-model matching process, and the capabilities of the backward-smoothing extended Kalman filter over the same time span as the reference attitude provided by the satellite's operators.

While the organizational challenges in coordinating the measurement process would be nontrivial, it is the opinion of the author that such a test case would be well worth the effort. Much work has been done on the part of Dr. Paul Cefola to find a suitable satellite, whose operators are willing and able to provide the requisite attitude information. Several satellites have been proposed the most promising of which include the Hubble Space Telescope (HST), RADARSAT, and ENVISAT. For an overview of each of the perspective satellite systems see References (57), (58), and (59), respectively. Ideally, the satellite selected for this study would have all the following attributes:

Table 6-1 Satellite Attributes for the Calibration Test Case

Selection Criteria	Ideal Attributes
Satellite Geometry	A large, asymmetric rigid body with several distinguishing features (booms and non-body-mounted solar arrays are preferable) and widely dispersed scattering centers would be helpful during the image-model matching process. Ultimately, the satellite should be large enough to show sufficient structure to be able to assign point scatterers in the image to corners of the wireframe model and the shape should be suitable for a unique assignment
Satellite Mass Properties	Two principal moments of inertia should be approximately equal to test the algorithm under the assumptions of a symmetric gyro on the one hand, but should be sufficiently different in order to challenge the algorithm with a satellite contradicting the assumption.
Attitude Control	Satellite should be inactive, uncontrolled, or otherwise not maneuvering during the imaging pass (passive attitude control systems are acceptable). The ideal target would also have minimal onboard thruster propellant in order to mitigate impact of fuel sloshing and would not use active internal control mechanisms such as reaction wheels.
Operational Considerations	The imaging opportunities should be selected on the basis of maximizing the amount of time in which the satellite is in view of the ground sensor and optimizing the ISAR geometry (e.g., passes with higher elevation angles will tend to have less atmospheric distortion and be easier to track). If possible, measurements should be collected for several consecutive passes.
Orbital Properties	A low altitude $(1000 \geq h \geq 500)$, high inclination $(i \approx 42 \text{ deg})$ orbit would minimize the time between imaging passes (short orbital period), maximize the time in view of the radar, and provide the best imaging geometry. A sun-synchronous orbit would be ideal since the satellite passes over any given point on the Earth's surface at the same local solar time. Additionally, a LEO satellite will enable the aerodynamic torque model to be more thoroughly tested.
Onboard Attitude Sensors	The array of onboard sensors that a particular satellite uses to determine its attitude should also be considered, since they will be gathering the "truth data" against which the non-cooperative attitude estimate will be judged.

6.5.3 Variation-of-Parameters Formulation

This approach extends the closed-form motion propagation, used in LLMotion, to account for motion in the presence of disturbance torques. In addition to improving LLMotion, this method could also be easily integrated into a backward-smoothing extended Kalman filter, since it is the dynamic equations which are being altered and not the estimation algorithm. A variation-of-parameters formulation represents an intermediate technique between what is done in the Lincoln Attitude Estimation System and LLMotion.

The spin-precession motion parameters specify the orientation of the spacecraft body principal axes relative to an inertial frame in which the angular momentum vector is along the inertial z-axis. It is frequently more convenient to specify the orientation of the spacecraft relative to some other inertial frame. This is especially important if one wishes to use the resulting closed-form solution as the starting point for a variation-of-parameters analysis of the motion in the presence of torques, because the angular momentum vector is not fixed in inertial space when the torque does not vanish. Changing this reference system is straightforward if there is a convenient rule for the parameters representing the product of two successive orthogonal transformations. Since the quaternion composition rule is well suited to the task, the closed-form solution for this kinematic representation is also needed. The solution, in the axial symmetry case, is (60 pp. 40 - 42)

$$\boldsymbol{q}(t) = \begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix} \boldsymbol{q}_0 \tag{6.1}$$

where

$$q_1' = u_1 \cos(\alpha) \sin(\beta) + u_2 \sin(\alpha) \sin(\beta) \tag{6.2}$$

$$q_2' = u_2 \cos(\alpha) \sin(\beta) - u_1 \sin(\alpha) \sin(\beta) \tag{6.3}$$

$$q_3' = u_3 \cos(\alpha)\sin(\beta) + \sin(\alpha)\cos(\beta) \tag{6.4}$$

$$q_4' = \cos(\alpha)\cos(\beta) - u_3 \sin(\alpha)\sin(\beta) \tag{6.5}$$

$$\alpha = \frac{1}{2}(\dot{\psi})t \tag{6.6}$$

$$\beta = \frac{1}{2}(\dot{\phi})t \tag{6.7}$$

$$\mathbf{u} = \frac{\boldsymbol{L}_0}{|L_0|} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{6.8}$$

In the equations above, all the constants of the motion have been reexpressed in terms of initial values of the quaternion components and $\boldsymbol{L}_0$, the angular momentum vector in body coordinates. These initial values are arbitrary (except that the sum of squares of the quaternion must be unity) because the inertial reference frame can be chosen subjectively. The variation-of-parameters formulation of attitude dynamics is a method for exploiting the torque-free solutions in the presence of torques. The parameters to be varied are the initial values of both the quaternion and the components of the angular momentum vector along the body principal axes. The forward solutions are equation (6.1) and

$$\boldsymbol{L}(t) = \begin{bmatrix} (L_1)_0 \cos(2\alpha) + (L_2)_0 \sin(2\alpha) \\ (L_2)_0 \cos(2\alpha) + (L_1)_0 \sin(2\alpha) \\ (L_3)_0 \end{bmatrix} \tag{6.9}$$

where α is given by equation (6.6). These are obtained by multiplying the following equations

$$\boldsymbol{\omega} = \begin{bmatrix} (\omega_1)_0 \cos(\dot{\psi}t) + (\omega_2)_0 \sin(\dot{\psi}t) \\ (\omega_2)_0 \cos(\dot{\psi}t) - (\omega_1)_0 \sin(\dot{\psi}t) \\ (\omega_3)_0 \end{bmatrix}$$

by the principal moments of inertia along the three axes. The backward solutions are obtained from equations (6.1) and (6.9) by interchanging $\boldsymbol{L}$ with $\boldsymbol{L}_0$ and $\boldsymbol{q}$ with $\boldsymbol{q}_0$, and changing the sign of t (and thus of α and β). Differentiating the backward solutions and substituting the forward equations of motion on the right-hand sides yields the variation-of-parameters equations of motion for the axial symmetric case (60 p. 44):

$$\dot{\boldsymbol{q}}_0 = \frac{1}{2}\begin{bmatrix} 0 & \widehat{\omega}_3 & -\widehat{\omega}_2 & \widehat{\omega}_1 \\ -\widehat{\omega}_3 & 0 & \widehat{\omega}_1 & \widehat{\omega}_2 \\ \widehat{\omega}_2 & -\widehat{\omega}_1 & 0 & \widehat{\omega}_3 \\ -\widehat{\omega}_1 & -\widehat{\omega}_2 & -\widehat{\omega}_3 & 0 \end{bmatrix}\boldsymbol{q}_0 \tag{6.11}$$

$$\dot{\boldsymbol{L}}_0(t) = \begin{bmatrix} N_a + 2\left(1 - \frac{I_1}{I_3}\right)u_2 N_3 \beta \\ N_b - 2\left(1 - \frac{I_1}{I_3}\right)u_1 N_3 \beta \\ N_3 \end{bmatrix} \tag{6.12}$$

where α, β, and **u** are given by equations (6.6) – (6.8); and N_a, N_b, and $\boldsymbol{\widehat{\omega}}$ are defined by (60 pp. 45 - 48)

$$N_a = N_1\cos(2\alpha) - N_2\sin(2\alpha) \tag{6.13}$$

$$N_b = N_2\cos(2\alpha) + N_1\sin(2\alpha) \tag{6.14}$$

$$\boldsymbol{\widehat{\omega}} = \begin{bmatrix} [(u_3N_b - u_2N_3)(1-\cos(2\beta)) - u_1G - N_a\sin(2\beta)]/\boldsymbol{L}_0 \\ [(u_1N_3 - u_3N_a)(1-\cos(2\beta)) - u_2G - N_b\sin(2\beta)]/\boldsymbol{L}_0 \\ [(u_2N_a - u_1N_b)(1-\cos(2\beta)) - u_3G - N_3(\sin(2\beta) - (1 - I_1/I_3)2\beta)]/\boldsymbol{L}_0 \end{bmatrix} \tag{6.15}$$

with

$$G = (u_1N_a + u_2N_b + u_3N_3)(2\beta - \sin(2\beta)) \tag{6.16}$$

Note that the angle β must be expressed in radians in these equations. The variation-of-parameters method will be most useful if $|N|t \ll |L|$. In this case, equations (6.11) and (6.12) can be integrated with a large time step to find $\boldsymbol{q}_0(t)$ and $\boldsymbol{L}_0(t)$. The quaternion $\boldsymbol{q}(t)$ representing the attitude at time t, can be obtained from equations (6.1) – (6.8); the angular momentum vector $\boldsymbol{L}(t)$ can be found from equation (6.9) if desired. Equation (6.11) has the same structure as the kinematic equation for the instantaneous quaternion, equation (4.31), so any techniques used for solving the latter equation can also be used for the former.

The variation-of-parameters equations for the asymmetric case, $I_1 \neq I_2 \neq I_3$, are significantly more complicated than the axial symmetry case, largely because of the dependence of a particular parameter of the Jacobian elliptic functions on the initial values of $\boldsymbol{L}$ and $\boldsymbol{q}$. The extension of this approach to asymmetric spacecraft is covered extensively by the authors of Reference (60).

6.5.4 Formal XELIAS/LAES Interface

A formal interface between LAES and XELIAS is needed in order to properly visualize, evaluate, and correct the state estimates output by the BSEKF. As has already been mentioned, the current approach is to convert the BSEKF state vector for each discrete observation in the pass into an equivalent set of spin-precession motion parameters. Each set of parameters is then manually input into Interactive Motion in order to view the resulting alignment between the wireframe model and underlying radar image. Because, the parameters are applied to the entire pass, the motion solution can only be viewed for a single image at a time. Not only is this process incredibly inefficient, tedious, and time consuming, with no way to hold the values fixed over a series of images, it is very difficult to make adjustments to the measurements. A formal interface should, therefore, possess the following features and capabilities:

1. Automatically pass the wireframe model, attitude measurements, and other critical filter inputs, contained in the *iges*, *imagecdf*, and *paramcdf* files, to the BSEKF. Improving the level of integration between the BSEKF and Satellite Tool Kit (STK), would also greatly enhance the ability of the analyst to quickly acquire the orbital information needed to run the various environmental torque models. Data handling functions can and should be entirely automated.

2. Discontinue the use of the eight symmetric motion parameters as the primary attitude representation (or allow the analyst to select the appropriate parameterization) and enable the XELIAS to work directly with the BSEKF state vectors. With the changes detailed in Section 6.4, this should be a relatively straightforward process, since the attitude quaternion can be easily converted into a rotation matrix using equation (A.5) and applied directly to the computer generated model.
3. A robust visualization function would allow the analyst to display a motion solution over one or more passes. Rather than using a single set of parameters over the entire pass, the BSEKF should provide XELIAS with a *motioncdf* file containing the attitude matrix, angular velocity, moment of inertia, and reference time at which this information is valid. In addition to displaying the wireframe model overlaid onto radar images, it would be useful to see the attitude depicted in terms of the quaternion or Euler angles over time as well. As the orientation of the model is adjusted, the attitude representation could be updated to reflect the correction made to the motion. Note that these plots would provide the analysis with an objective measure of an estimate's accuracy (similar to the predicted-versus-measured plots given in Chapter 5), but should not be taken as the new attitude measurements; to do so would be a return to the previous method of adjusting the motion parameters.

6.5.5 Bistatic Radar Measurements

The single greatest limiting factor in the non-cooperative attitude estimation problem is the number of measurements which can be acquired from a monostatic radar over the course a single pass. A bistatic or multistatic radar system consists of separately located (by a considerable distance) transmitting and receiving sites. Therefore, a monostatic Doppler radar can be easily upgraded to a multistatic system by including additional receivers or (by use of the same frequency) two monostatic radars can work together like a bistatic system. Because a bistatic radar makes use of the forward scattering of the transmitted energy, a signal can also be received when the geometry of the reflecting object returns very little energy in the direction of a single sensor. Essentially, the larger distance between the transmitting unit and the receiving unit(s) results in a greater parallax. This difference in viewing geometry obtained from two concurrent images eliminates the ambiguity in the orientation of the radar image plane. Table 6-2 summarizes some of the primary advantages and challenges to using multiple ground sensors.

Table 6-2 Principal Advantages and Disadvantages of Bistatic/Multistatic Radar Systems

Advantages	Disadvantages
Compared to the transmitting unit, the procurement, operations, and maintenance costs for receivers are minimal	Additional cost required to provide communication between sites
Receivers are relatively simple and the advent of GPS solves many of the synchronization and timing problems that have previously limited the performance of bistatic systems. Additionally, increases in signal processing power mean that many of the signal digitization and processing operations are now feasible in real time.	Increased system complexity with multiple receivers
Receivers are mobile and can, therefore, be repositioned in order to maximize the viewing geometry based on the satellite's orbit	Harder to deploy/implement
Possible enhanced radar cross section of the target due to geometrical effects	Reduced low-level coverage due to the need for line of sight from several locations
Greater apparent difference in the orientation of a satellite viewed along two different lines of sight	A limiting case of the bistatic geometry occurs when the target lies on the transmitter-receiver baseline
Multiple radar images at each discrete time result in an overdetermined system (more equations than unknowns), reducing uncertainty in the measurements	The technology is still relatively immature and, thus, requires a larger research and development effort

The current approach attempts to solve for six degrees of freedom using a single two-dimensional radar image, which provides only four measurements. In the stereoscopic approach, however, two radar images with substantially different radar lines of sight could be used to obtain eight independent measurements for each discrete observation time. The general measurement procedure begins by displaying each set of radar images next to one another and projecting the wireframe model simultaneously into the two radar image planes. During the image-model matching process, the analyst would then rotate and translate the model in order to align it with both images. Manipulating the wireframe projected onto one image plane results in a corresponding change in the model overlaid onto the other image. Similarly, a change in the cross-range scaling would impact both images equally. In this manner, the attitude and angular velocity of the spacecraft could be uniquely determined for each discrete time in the pass (i.e., there is no longer an uncertainty in these values about the RLOS). For details on bistatic radar systems and a list of relevant equations see Reference (61).

6.5.6 Computer Aided Image-Model Matching

References (33) and (34), describe how to constrain the rotation of a faceted three-dimensional object about a fixed axis to fit a desired extent in a particular direction. The algorithm described in these two articles, could be used as the starting point for partially automating the image-model matching process. The intent in developing and implementing such a program would be to: 1) reduce the amount of time needed to obtain accurate attitude measurements and 2) help resolve sources of error related to limitations in human perception and motor control. The computer aided measurement procedure currently envisioned would work in the following manner:

1. A bounding box is sized by the analyst expressing the maximum range and cross-range extents of the three-dimensional satellite projected into the two-dimensional radar image plane (see Figure 6.1).

2. Obtain a "coarse match" by having the analyst quickly translate and rotate the wireframe model to roughly align it with the image. The most time consuming part of the process is the series of fine manipulations needed to achieve a precise match between the model and underlying image; it is this activity which will be performed by the computer. One of the most challenging aspects of this problem will be to balance computation time and accuracy against what can be already be done by a well trained operator (the program is of little value if it does not reduce the time and effort required to make attitude measurements). A well designed graphical user interface (GUI) will be needed in order to make this both an efficient and effective tool.
3. Set the parameter step size and search range. In a manner similar to what is currently being done by the Dynamic Hill Climbing algorithm in LLMotion, the search space could be partitioned into smaller subspaces based on the initial range of values input by the analyst. The computer would then take over and use a brute force method to systematically vary the orientation of the wireframe model to find the projection which best fits within the bounding box. Parallel processing using the Lincoln Laboratory Grid (LLGrid) should be considered for this task. Once the search space has been gridded, each piece can be passed to a separate computer which would then increment and compare all the combinations of Euler angles within its particular subspace. Once all the perspective solutions have been aggregated, a final cost comparison could be preformed and the proper attitude selected/displayed.
4. Essentially, the algorithm examines ranges of angles resulting in distinct pairs of extreme vertices v^+ and v^-. The key is to observe that one only needs to consider the vertices on the boundary of the model's convex hull. The cost function penalizes attitude estimates

(i.e., combinations of Euler angles) which result in a model projection whose extreme vertices (i.e., the $\max(\pm x)$ and $\max(\pm y)$ coordinate values) are further from the measured maximum extent. In the end, the goal is to minimize the distance of the extreme vertices from the constraining perimeter.

5. As a final note, the algorithm should be designed so that a change in the cross-range scaling automatically resizes the bounding box, to prevent the analyst from having to manually redraw the perimeter after each scaling adjustment.

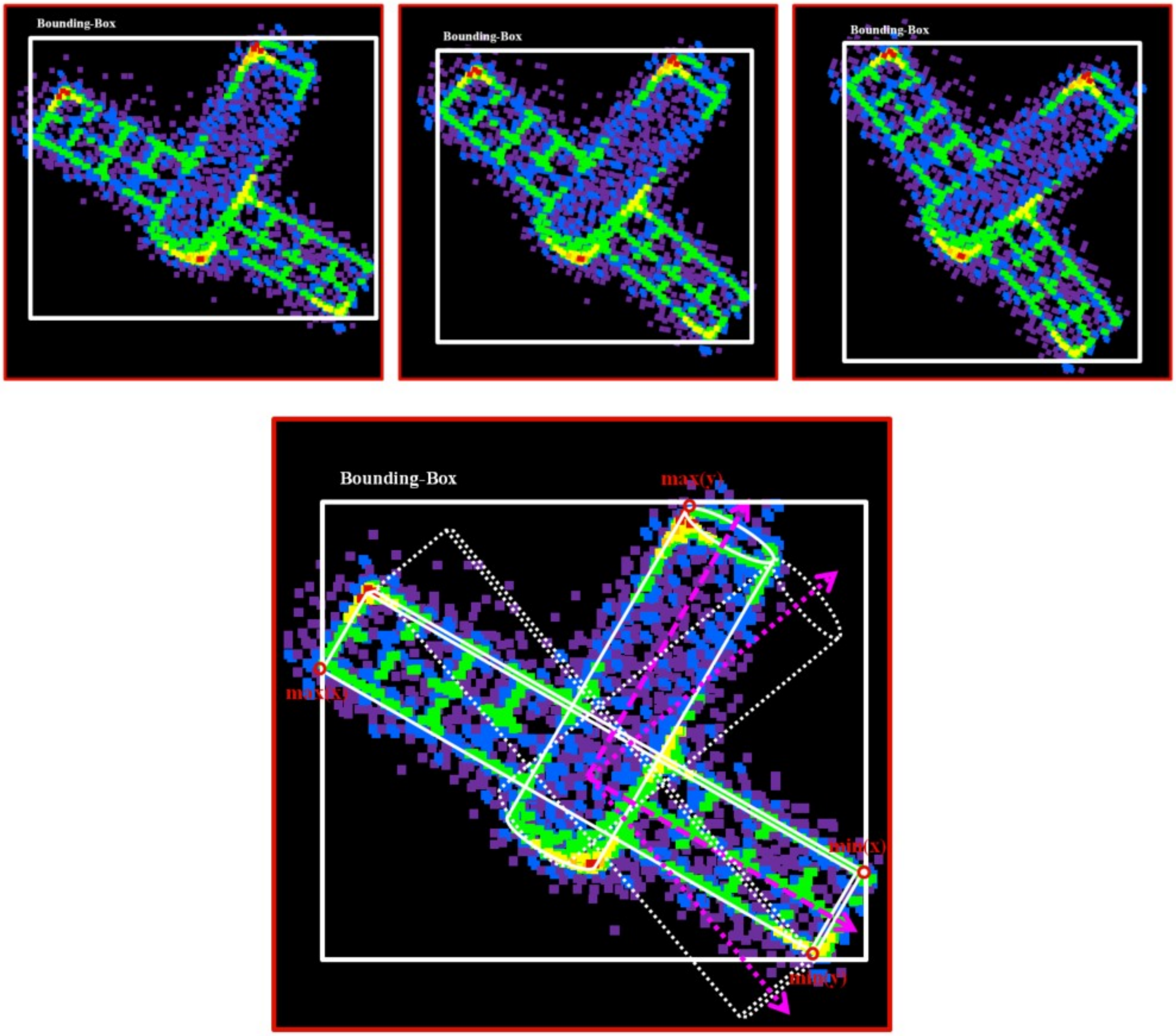


Figure 6.1 Partial Automation of the Measurement Process Using Bounding Boxes

6.5.7 Tomasi-Kanade Factorization Method

Inferring scene geometry and satellite motion from a stream of radar images is possible, but is an ill-conditioned problem (sensitive to noise) when the targets are distant with respect to their size. In Reference (62), the authors develop a factorization method that can overcome this difficulty by recovering shape and motion under orthography without computing depth as an intermediate step. In the Tomasi-Kanade Factorization Method, an image sequence is represented as a $2F \times P$ measurement matrix $\boldsymbol{Y}$, which is made up of the horizontal and vertical coordinates of P points tracked through F frames. If image coordinates are measured with respect to their centroid, it can be shown that under orthographic projection, the measurement matrix is of rank 3. This is known as the rank theorem and enables the matrix $\boldsymbol{Y}$ to be factored into the product of two matrices $\boldsymbol{R}$ and $\boldsymbol{S}$ (62 p. 137). Here, $\boldsymbol{R}$ is a $2F \times 3$ matrix that represents the target's rotation, and $\boldsymbol{S}$ is a $3 \times P$ matrix that represents the shape in a coordinate system attached to the to the object centroid. The two components of the target's translation in the radar image plane are computed as averages of the rows of $\boldsymbol{Y}$. When features appear and disappear in the image sequence because of occlusions or tracking failures, the resulting measurement matrix $\boldsymbol{Y}$ is only partially filled in. The factorization method can handle this situation by growing a partial solution obtained from an initial full submatrix into a complete solution with an iterative procedure (62 p. 137). As a side note: when the measurement points are selected using the axes of the wireframe model, even if a particular feature is obscured/absent in a particular image (or series of images), there should be no holes in the matrix.

The rank theorem captures precisely the nature of the redundancy that exists in an image sequence, and permits a large number of points and frames to be processed in a conceptually simple and computationally efficient way to reduce the effects of noise. The resulting algorithm is based on singular-value decomposition [Matlab function *svd*], which is numerically well behaved and stable. The robustness of the recovery algorithm in turn enables one to use an image sequence with a very short interval between frames (an image stream), which makes feature tracking relatively simple and the assumption of orthography easier to approximate (62 p. 138). For the further information on the method and the specific equations needed to implement the algorithm, see References (62), (63), and (64).

[This page intentionally left blank.]

7 Appendix A

Convert from Euler Angles to a Rotation Matrix (6 p. 24 & 32)

$$\boldsymbol{R}_{123}(\psi,\theta,\phi) = \boldsymbol{R}_1(\psi)\boldsymbol{R}_2(\theta)\boldsymbol{R}_3(\phi)$$

$$= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\phi)\cos(\theta) & \sin(\phi)\cos(\theta) & -\sin(\theta) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \cos(\theta)\sin(\psi) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \cos(\theta)\cos(\psi) \end{bmatrix} \tag{A.1}$$

$$\boldsymbol{R}_{313}(\psi,\theta,\phi) = \boldsymbol{R}_3(\psi)\boldsymbol{R}_1(\theta)\boldsymbol{R}_3(\phi)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\phi)\cos(\psi) - \sin(\phi)\cos(\theta)\sin(\psi) & \sin(\phi)\cos(\psi) + \cos(\phi)\cos(\theta)\sin(\psi) & \sin(\theta)\sin(\psi) \\ -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\theta)\cos(\psi) & \sin(\theta)\cos(\psi) \\ \sin(\phi)\sin(\theta) & -\cos(\phi)\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{A.2}$$

Convert from Euler Angles to a Quaternion (6 p. 24 & 32)

$$\boldsymbol{q}_{123}(\psi,\theta,\phi) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_{123} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) - \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \end{bmatrix} \tag{A.3}$$

$$\boldsymbol{q}_{313}(\psi,\theta,\phi) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_{313} = \begin{bmatrix} \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) + \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) - \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \end{bmatrix} \tag{A.4}$$

Convert from a Quaternion to a Rotation Matrix (6 p. 14)

$$\boldsymbol{R}_q(\boldsymbol{q}) = (q_4^2 - |\mathbf{q}|^2)\mathbf{1}_{4\times4} + 2\mathbf{q}\mathbf{q}^T - 2q_4[\mathbf{q}\,\times]$$

$$= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{A.5}$$

Convert from a Quaternion to Euler Angles (6 p. 24 & 32)

$$\boldsymbol{E}_{123}(\boldsymbol{q}) = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{123} = \begin{bmatrix}\operatorname{atan2}(2(q_1q_2 + q_3q_4), q_1^2 - q_2^2 - q_3^2 + q_4^2)\\ -\operatorname{asin}\big(2(q_1q_3 - q_2q_4)\big)\\ \operatorname{atan2}(2(q_2q_3 + q_1q_4), -q_1^2 - q_2^2 + q_3^2 + q_4^2)\end{bmatrix} \tag{A.6}$$

$$\boldsymbol{E}_{313}(\boldsymbol{q}) = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{313} = \begin{bmatrix}\operatorname{atan2}\big(2(q_1q_3 + q_2q_4), 2(-q_2q_3 + q_1q_4)\big)\\ \operatorname{acos}(-q_1^2 - q_2^2 + q_3^2 + q_4^2)\\ \operatorname{atan2}\big(2(q_1q_3 - q_2q_4), 2(q_2q_3 + q_1q_4)\big)\end{bmatrix} \tag{A.7}$$

Convert from a Rotation Matrix to Euler Angles (6 p. 24 & 32)

$$\boldsymbol{E}_{123}\big(\boldsymbol{R}_{123}(\psi, \theta, \phi)\big) = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{123} = \begin{bmatrix}\operatorname{atan2}(r_{12}, r_{11})\\ -\operatorname{asin}(r_{13})\\ \operatorname{atan2}(r_{23}, r_{33})\end{bmatrix} \tag{A.8}$$

$$\boldsymbol{E}_{313}\big(\boldsymbol{R}_{313}(\psi, \theta, \phi)\big) = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{313} = \begin{bmatrix}\operatorname{atan2}(r_{31}, -r_{32})\\ \operatorname{acos}(r_{33})\\ \operatorname{atan2}(r_{13}, r_{23})\end{bmatrix} \tag{A.9}$$

Convert from a Rotation Matrix to a Quaternion (6 p. 15)

If $r_{22} < -r_{33}$ and $r_{11} > r_{22}$ and $r_{11} > r_{33}$

$$\boldsymbol{q} = \begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix} = \begin{bmatrix}\frac{1}{2}\sqrt{1 + r_{11} - r_{22} - r_{33}}\\ \frac{1}{4q_1}(r_{12} + r_{21})\\ \frac{1}{4q_1}(r_{31} + r_{13})\\ \frac{1}{4q_1}(r_{23} - r_{32})\end{bmatrix} \tag{A.10}$$

If $r_{22} > r_{33}$ and $r_{11} < r_{22}$ and $r_{11} < -r_{33}$

$$\boldsymbol{q} = \begin{bmatrix}q_2\\ q_1\\ q_3\\ q_4\end{bmatrix} = \begin{bmatrix}\frac{1}{2}\sqrt{1 - r_{11} + r_{22} - r_{33}}\\ \frac{1}{4q_2}(r_{12} + r_{21})\\ \frac{1}{4q_2}(r_{23} + r_{32})\\ \frac{1}{4q_2}(r_{31} - r_{13})\end{bmatrix} \tag{A.11}$$

If $r_{22} < r_{33}$ and $r_{11} < -r_{22}$ and $r_{11} < r_{33}$

$$\boldsymbol{q} = \begin{bmatrix} q_3 \\ q_1 \\ q_2 \\ q_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\sqrt{1 - r_{11} - r_{22} + r_{33}} \\ \frac{1}{4q_3}(r_{31} + r_{13}) \\ \frac{1}{4q_3}(r_{23} + r_{32}) \\ \frac{1}{4q_3}(r_{12} - r_{21}) \end{bmatrix} \tag{A.12}$$

If $r_{22} > -r_{33}$ and $r_{11} > -r_{22}$ and $r_{11} > -r_{33}$

$$\boldsymbol{q} = \begin{bmatrix} q_4 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \\ \frac{1}{4q_4}(r_{23} - r_{32}) \\ \frac{1}{4q_4}(r_{31} - r_{13}) \\ \frac{1}{4q_4}(r_{12} - r_{21}) \end{bmatrix} \tag{A.13}$$

Intra-Euler Angle Conversion (6 pp. 13 - 14)

$$\boldsymbol{E}_{123}\big(\boldsymbol{R}_{313}(\psi, \theta, \phi)\big) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{123}$$

$$= \begin{bmatrix} \operatorname{atan2}(\cos(\phi)\cos(\theta)\sin(\psi) + \sin(\phi)\cos(\psi), -\sin(\phi)\cos(\theta)\sin(\psi) + \cos(\phi)\cos(\psi)) \\ -\operatorname{asin}(\sin(\theta)\sin(\psi)) \\ \operatorname{atan2}(\sin(\theta)\cos(\psi), \cos(\theta)) \end{bmatrix} \tag{A.14}$$

$$\boldsymbol{E}_{313}\big(\boldsymbol{R}_{123}(\psi, \theta, \phi)\big) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{313}$$

$$= \begin{bmatrix} \operatorname{atan2}(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi), -\sin(\phi)\sin(\theta)\cos(\psi) + \cos(\phi)\sin(\psi)) \\ \operatorname{acos}(\cos(\theta)\cos(\psi)) \\ \operatorname{atan2}(-\sin(\theta), \cos(\theta)\sin(\psi)) \end{bmatrix} \tag{A.15}$$

Convert from Euler Angle Rates to an Angular Velocity (6 p. 24 & 32)

$$\boldsymbol{\omega}_B(\boldsymbol{E}_{123}) = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\psi) & \cos(\theta)\sin(\psi) \\ 0 & -\sin(\psi) & \cos(\theta)\cos(\psi) \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}_{123} \tag{A.16}$$

$$\boldsymbol{\omega}_B(\boldsymbol{E}_{313}) = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B = \begin{bmatrix} 0 & \cos(\psi) & \sin(\theta)\sin(\psi) \\ 0 & -\sin(\psi) & \sin(\theta)\cos(\psi) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}_{313} \tag{A.17}$$

Convert from an Angular Velocity to Euler Angle Rates (6 p. 24 & 32)

$$\dot{\boldsymbol{E}}_{123}(\boldsymbol{\omega}_B) = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}_{123} = \frac{1}{\cos(\theta)} \begin{bmatrix} \cos(\theta) & \sin(\theta)\sin(\psi) & \sin(\theta)\cos(\psi) \\ 0 & \cos(\theta)\cos(\psi) & -\cos(\theta)\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B \qquad (\mathrm{A.18})$$

$$\dot{\boldsymbol{E}}_{313}(\boldsymbol{\omega}_B) = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}_{313} = \frac{1}{\sin(\theta)} \begin{bmatrix} -\cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) & \sin(\theta) \\ \sin(\theta)\cos(\psi) & -\sin(\theta)\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B \qquad (\mathrm{A.19})$$

Convert from Quaternion Rates to an Angular Velocity (6 p. 16)

$$\boldsymbol{\omega}_I(\dot{\boldsymbol{q}}) = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_I = 2 \begin{bmatrix} q_4 & -q_3 & q_2 & -q_1 \\ q_3 & q_4 & -q_1 & -q_2 \\ -q_2 & q_1 & q_4 & -q_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \qquad (\mathrm{A.20})$$

$$\boldsymbol{\omega}_B(\dot{\boldsymbol{q}}) = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \qquad (\mathrm{A.21})$$

Convert from Angular Velocity and the unit quaternion to the quaternion rates (6 p. 16)

$$\dot{\boldsymbol{q}}(\boldsymbol{\omega}_I) = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & -q_1 \\ q_3 & q_4 & -q_1 & -q_2 \\ -q_2 & q_1 & q_4 & -q_3 \end{bmatrix}^T \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_I \qquad (\mathrm{A.22})$$

$$\dot{\boldsymbol{q}}(\boldsymbol{\omega}_B) = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix}^T \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B \qquad (\mathrm{A.23})$$

The Derivatives of the Rotation Matrix with Respect to the Euler Angles (6 p. 24 & 32)

$$\frac{\partial \boldsymbol{R}_{123}}{\partial \phi}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)\cos(\theta) \\ -\sin(\phi)\sin(\theta)\cos(\psi) + \cos(\phi)\sin(\psi) & -\sin(\phi)\sin(\theta)\sin(\psi) - \cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\theta) \end{bmatrix}$$

$$(\mathrm{A.24})$$

$$\frac{\partial \boldsymbol{R}_{123}}{\partial \theta} = \begin{bmatrix} -\sin(\theta)\cos(\psi) & -\sin(\theta)\sin(\psi) & -\cos(\theta) \\ \sin(\phi)\cos(\theta)\cos(\psi) & \sin(\phi)\cos(\theta)\sin(\psi) & -\sin(\phi)\sin(\theta) \\ \cos(\phi)\cos(\theta)\cos(\psi) & \cos(\phi)\cos(\theta)\sin(\psi) & -\cos(\phi)\sin(\theta) \end{bmatrix} \qquad (\mathrm{A.25})$$

$$\frac{\partial \boldsymbol{R}_{123}}{\partial \psi} = \begin{bmatrix} -\cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) & 0 \\ -\sin(\phi)\sin(\theta)\sin(\psi) - \cos(\phi)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & 0 \\ -\cos(\phi)\sin(\theta)\sin(\psi) + \sin(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & 0 \end{bmatrix} \tag{A.26}$$

$$\frac{\partial \boldsymbol{R}_{313}}{\partial \phi} = \begin{bmatrix} -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\theta)\cos(\psi) & \sin(\psi)\cos(\theta) \\ -\cos(\phi)\cos(\psi) + \sin(\phi)\cos(\theta)\sin(\psi) & -\sin(\phi)\cos(\psi) - \cos(\phi)\cos(\theta)\sin(\psi) & -\sin(\psi)\sin(\theta) \\ 0 & 0 & 0 \end{bmatrix} \tag{A.27}$$

$$\frac{\partial \boldsymbol{R}_{313}}{\partial \theta} = \begin{bmatrix} \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\phi)\sin(\theta)\sin(\psi) & \cos(\theta)\sin(\psi) \\ \sin(\phi)\sin(\theta)\cos(\psi) & -\cos(\phi)\sin(\theta)\cos(\psi) & \cos(\theta)\cos(\psi) \\ \sin(\phi)\cos(\theta) & -\cos(\phi)\cos(\theta) & -\sin(\theta) \end{bmatrix} \tag{A.28}$$

$$\frac{\partial \boldsymbol{R}_{313}}{\partial \psi} = \begin{bmatrix} -\sin(\phi)\cos(\psi) - \cos(\phi)\cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi) - \sin(\phi)\cos(\theta)\sin(\psi) & 0 \\ \sin(\phi)\sin(\psi) - \cos(\phi)\cos(\theta)\cos(\psi) & -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & 0 \\ \cos(\phi)\sin(\theta) & \sin(\phi)\sin(\theta) & 0 \end{bmatrix} \tag{A.29}$$

The Derivatives of the Rotation Matrix with Respect to the Elements of the Quaternion (6 p. 15)

$$\frac{\partial \boldsymbol{R}_q}{\partial q_1} = 2\begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & q_4 \\ q_3 & -q_4 & -q_1 \end{bmatrix} \tag{A.30}$$

$$\frac{\partial \boldsymbol{R}_q}{\partial q_2} = 2\begin{bmatrix} -q_2 & q_1 & -q_4 \\ q_1 & q_2 & q_3 \\ q_4 & q_3 & -q_2 \end{bmatrix} \tag{A.31}$$

$$\frac{\partial \boldsymbol{R}_q}{\partial q_3} = 2\begin{bmatrix} -q_3 & q_4 & q_1 \\ -\ q_4 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{bmatrix} \tag{A.32}$$

$$\frac{\partial \boldsymbol{R}_q}{\partial q_4} = 2\begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix} \tag{A.33}$$

The Derivatives of the Euler Angles with Respect to the Elements of the Quaternion (6)

$$\frac{\partial\phi}{\partial q} = \begin{bmatrix} \frac{-q_2}{q_1^2 + q_2^2} & \frac{q_1}{q_1^2 + q_2^2} & \frac{q_4}{q_3^2 + q_4^2} & \frac{-q_3}{q_3^2 + q_4^2} \end{bmatrix} \tag{A.34}$$

$$\frac{\partial\theta}{\partial q} = \frac{2}{\sqrt{1 - 2q_1^2(q_2^2 - q_3^2 - q_4^2) + 2q_2^2(q_3^2 + q_4^2) - 2q_3^2(q_4^2) - (q_1^4 + q_2^4 + q_3^4 + q_4^4)}} [q_1 \quad q_2 \quad -q_3 \quad -q_4] \tag{A.35}$$

$$\frac{\partial\psi}{\partial q} = \begin{bmatrix} \frac{q_2}{q_1^2 + q_2^2} & \frac{-q_1}{q_1^2 + q_2^2} & \frac{q_4}{q_3^2 + q_4^2} & \frac{-q_3}{q_3^2 + q_4^2} \end{bmatrix} \tag{A.36}$$

8 Appendix B

Day of the Year: (3 p. 191)

Table B-1 Day Numbers

Month	Days of the Year (Leap-Year Values)	
January	1 – 31	Day
February	32 – 59 (32 – 60)	Day + 31
March	60 – 90 (61 – 91)	Day + 59 (60)
April	91 – 120 (92 – 121)	Day + 90 (91)
May	121 – 151 (122 – 152)	Day + 120 (121)
June	152 – 181 (153 – 182)	Day + 151 (152)
July	182 – 212 (183 – 213)	Day + 181 (182)
August	213 – 243 (214 – 244)	Day + 212 (213)
September	244 – 273 (245 – 274)	Day + 243 (244)
October	274 – 304 (275 – 305)	Day + 273 (274)
November	305 – 334 (306 – 335)	Day + 304 (305)
December	335 – 365 (336 – 366)	Day + 334 (335)

Julian Date: (3 p. 189)

$$JD = 367(year) - \mathbf{int}\left[\frac{7\left(year + \mathbf{int}\left(\frac{month + 9}{12}\right)\right)}{4}\right] + \mathbf{int}\left(\frac{275(month)}{9}\right) + day + 1721013.5$$

$$+ \frac{\left(\frac{(second/60) + minute}{60}\right) + hour}{24} \qquad (B.1)$$

The **int** relation denotes real truncation (remove decimal values or round towards zero)

Joint Probability:

The joint probability density function f, cumulative distribution function F, and joint probability element for a continuous random vector composed of random elements X and Y, are given by the set of equations:

$$f_{X,Y}(x,y) = \frac{\partial F_{X,Y}^{2}(x,y)}{\partial x \partial y} \qquad (B.2)$$

$$F_{X,Y}(x,y) = p(X \le x, Y \le y) = \int\int f_{X,Y}(x,y)dxdy \qquad (B.3)$$

$$p(x \leq X \leq x + dx, y \leq Y \leq y + dy) = f_{X,Y}(x,y)dxdy \tag{B.4}$$

Marginal Probability:

Determining the probability behavior of one random variable, given the joint probability behavior of two, is then done by means of the following:

$$f_X(x) = \int f_{X,Y}(x,y)dy \tag{B.5}$$

$$F_X(x) = F_X(x,\infty) = p(X \leq x) = \int f_X(x)dy \tag{B.6}$$

$$f_Y(y) = \int f_{X,Y}(x,y)dx \tag{B.7}$$

$$F_Y(y) = F_Y(\infty,y) = p(Y \leq y) = \int f_Y(y)dx \tag{B.8}$$

Hence, the marginal density function of a random variable is obtained from the joint density function by integrating over the unwanted variable.

Conditional Probability:

Bayes' Formula offers a way to specify the probability density of the random variable *X* given random variable *Y* and is given as

$$f_{(X|Y=y)}(x) = \frac{f_{X,Y}(x,y)}{f_Y(y)} = \frac{f_{(Y|X=x)}(y)f_X(x)}{f_Y(y)} \tag{B.9}$$

$$f_{(Y|X=x)}(y) = \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{f_{(X|Y=y)}(x)f_Y(y)}{f_X(x)} \tag{B.10}$$

LLMotion Transformations: (32 pp. B-1 - B-13)

The first three rotation matrices convert the body axes to precession axes

$$\boldsymbol{R}_x(\phi)\boldsymbol{R}_y\left(\theta - \frac{\pi}{2}\right)\boldsymbol{R}_z(\psi) \tag{B.11}$$

where ϕ, θ, and ψ are the precession, coning, and spin angles, respectively, and are a function of the satellite precession rate $(\dot{\phi})$, spin rate $(\dot{\psi})$, x-initial (ϕ_0), z-initial (ψ_0), observation time (t_i), and reference time (t_0) and are given by:

$$\phi = \dot{\phi}(\Delta t) + \phi_0 \tag{B.12}$$

$$\theta = \theta_0 \tag{B.13}$$

$$\psi = \dot{\psi}(\Delta t) + \psi_0 \tag{B.14}$$

$$\Delta t = t_i - t_0 \tag{B.15}$$

The conversion from the precession system to an inertial frame is performed by two more rotations, involving the right ascension (α) and declination (δ) angles:

$$\boldsymbol{R}_{\mathrm{y}}(\alpha)\boldsymbol{R}_z(\delta) \tag{B.16}$$

For a dynamic radar line of sight, transformation to a radar oriented system is accomplished using the following four matrices:

$$\boldsymbol{R}_z\big(El(t_i)\big)\boldsymbol{R}_x\big(Az(t_i)\big)\boldsymbol{R}_z(-\varphi)\boldsymbol{R}_y\big(-\lambda(t_i)\big) \tag{B.17}$$

Where Az and El are the azimuth and elevation angles of the RLOS, φ is the latitude of the sensor and λ is the sum of the radar longitude, the right ascension of Greenwich at t_0, and the rotation of the Earth since the reference time $(t_i - t_0)$. Finally, in order to rotate the body vector into the radar image system requires two additional matrices given by:

$$\boldsymbol{R}_y\big(\varepsilon(t_i)\big)\boldsymbol{R}_y\big(\eta(t_i)\big) \tag{B.18}$$

where

$$\eta(t_i) = -\operatorname{atan}\left(\frac{\dot{A}z\cos(El)}{\dot{E}l}\right) \Rightarrow \boldsymbol{R}_y\big(\eta(t_i)\big) = \frac{1}{\sqrt{\big(\dot{A}z\cos(El)\big)^2 + \big(\dot{E}l\big)^2}}\begin{bmatrix} \dot{E}l & 0 & -\dot{A}z\cos(El) \\ 0 & 1 & 0 \\ \dot{A}z\cos(El) & 0 & \dot{E}l \end{bmatrix} \tag{B.19}$$

and

$$\varepsilon(t_i) = -\operatorname{atan}\left(\frac{\omega_x}{\omega_z}\right) \tag{B.20}$$

The total transformation matrix $\boldsymbol{T}$ and the instantaneous angular velocity vector $\boldsymbol{\omega}_{calc}$ corresponding to the series of transformations and satellite spin/precession vector are:

$$\boldsymbol{T} = \boldsymbol{R}_y\big(\varepsilon(t)\big)\boldsymbol{R}_y\big(\eta(t)\big)\boldsymbol{R}_z\big(El(t)\big)\boldsymbol{R}_x\big(Az(t)\big)\boldsymbol{R}_z(-\varphi)\boldsymbol{R}_y(-\lambda)\boldsymbol{R}_{\mathrm{y}}(\alpha)\boldsymbol{R}_z(\delta)\boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta - \pi/2)\boldsymbol{R}_z(\psi)$$

(B.21)

$$
\begin{aligned}
\boldsymbol{\omega}_{calc} = \boldsymbol{R}_y\big(\varepsilon(t)\big)\boldsymbol{R}_y\big(\eta(t)\big)\Bigg[&\boldsymbol{R}_z\big(El(t)\big)\boldsymbol{R}_x\big(Az(t)\big)\boldsymbol{R}_z(-\varphi)\boldsymbol{R}_y(-\lambda)\boldsymbol{R}_y(\alpha)\boldsymbol{R}_z(\delta)\boldsymbol{R}_x(\phi)\boldsymbol{R}_y\left(\theta-\frac{\pi}{2}\right)\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}\\
&+\boldsymbol{R}_z\big(El(t)\big)\boldsymbol{R}_x\big(Az(t)\big)\boldsymbol{R}_z(-\varphi)\boldsymbol{R}_y(-\lambda)\boldsymbol{R}_y(\alpha)\boldsymbol{R}_z(\delta)\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}\\
&+\boldsymbol{R}_z\big(El(t)\big)\boldsymbol{R}_x\big(Az(t)\big)\boldsymbol{R}_z(-\varphi)\begin{bmatrix}0\\-\dot{E}\\0\end{bmatrix}+\boldsymbol{R}_z\big(El(t)\big)\begin{bmatrix}\dot{Az}\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\\dot{El}\end{bmatrix}\Bigg]
\end{aligned}
\tag{B.22}
$$

where $\dot{E}$ is the Earth's rotation rate, $\dot{Az}$ is the azimuth rate, and $\dot{El}$ is the elevation rate. The total angular velocity for general spin/precession motion in the ECI coordinate system I and satellite body coordinate system B are given by

$$
\boldsymbol{\omega}_I = \boldsymbol{R}_y(\alpha)\boldsymbol{R}_z(\delta)\left(\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}+\boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta-\pi/2)\boldsymbol{R}_z(\psi)\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}\right) \tag{B.23}
$$

and

$$
\boldsymbol{\omega}_B = \boldsymbol{R}_z(\psi)^T\boldsymbol{R}_y(\theta-\pi/2)^T\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} \tag{B.24}
$$

respectively.

Angular Velocity Measurement in the Lincoln Attitude Estimation System

Recall that the image scale factor s, measured by the analyst, is a correction to the magnitude of the z component of the instantaneous angular velocity of the spacecraft. Because the direction of the vector is unchanged, the orientation of the image plane and spacecraft are unaffected; rather, it is the rate components which are being adjusted. The scaling values obtained from image-matching process can be recovered from the *paramcdf* file generated by XELIAS and used to correct the baseline motion. The new angular velocity value could then be used as an auxiliary or alternate measurement model in the backward-smoothing extended Kalman filter.

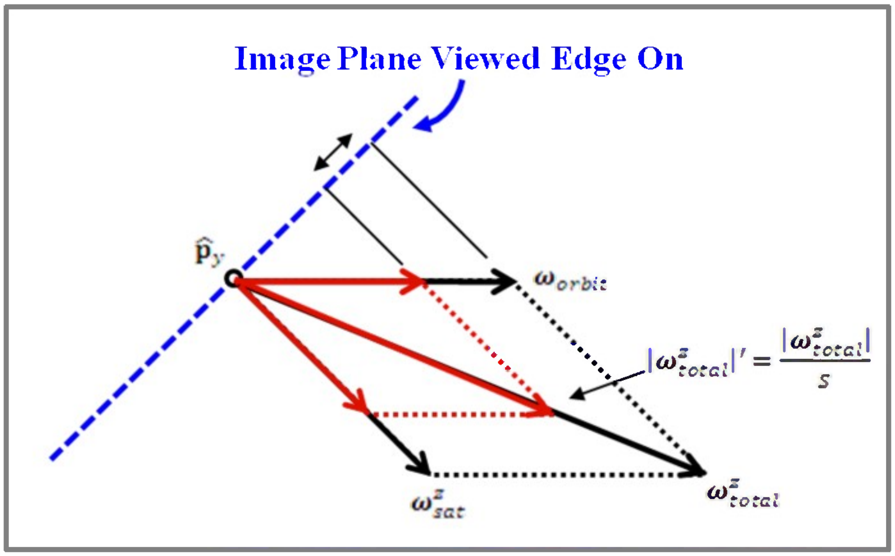


Image Plane Viewed Edge On
$\hat{p}_y$
ω_{orbit}
$|\omega^z_{total}|' = \frac{|\omega^z_{total}|}{s}$
ω^z_{sat}
ω^z_{total}

Figure B.1 Angular velocity correction geometry

Recall from the image plane geometry presented in Section 2.1, that the total angular velocity of the system is composed of components related to the rotation of the vehicle $\boldsymbol{\omega_{sat}}$ and values connected to the orbital motion of the spacecraft relative to the ground sensor $\boldsymbol{\omega_{orbit}}$. The corrected angular velocity of the satellite is denoted by $\boldsymbol{\omega'_{sat}}$ and can be decomposed into in-plane and out-of-plane components according to the equation:

$$\boldsymbol{\omega'_{sat}} = \boldsymbol{\omega^{y'}_{sat}} + \boldsymbol{\omega^{z'}_{sat}} \tag{B.25}$$

Combining equation $(\mathrm{B.25})$ with the definition of the total angular velocity yields the following set of relationships:

$$\boldsymbol{\omega^{y}_{total}} = \boldsymbol{\omega^{y}_{sat}} = \boldsymbol{\omega^{y'}_{sat}} \tag{B.26}$$

$$\boldsymbol{\omega^{z}_{total}} = \boldsymbol{\omega^{z}_{sat}} + \boldsymbol{\omega_{orbit}} \tag{B.27}$$

$$\boldsymbol{\omega^{z'}_{sat}} = \frac{\boldsymbol{\omega^{z}_{total}}}{s} - \boldsymbol{\omega_{orbit}} \tag{B.28}$$

Since the baseline motion solution is what is being corrected in this procedure, $\boldsymbol{\omega_{base}}$ is equivalent to the initial angular velocity of the satellite $\boldsymbol{\omega_{sat}}$. Inserting equation $(\mathrm{B.27})$ into $(\mathrm{B.28})$ and substituting the result, along with equation $(\mathrm{B.26})$, into equation $(\mathrm{B.25})$ produces:

$$\boldsymbol{\omega}'_{sat} = \boldsymbol{\omega}^{y}_{base} + \left(\frac{\boldsymbol{\omega}^{z}_{base} + \boldsymbol{\omega}_{orbit}}{s}\right) - \boldsymbol{\omega}_{orbit} = \boldsymbol{\omega}^{y}_{base} + \frac{1}{s}(\boldsymbol{\omega}^{z}_{base}) + \left(\frac{1-s}{s}\right)\boldsymbol{\omega}_{orbit} \quad (\mathrm{B.29})$$

where $\boldsymbol{\omega}_{orbit}$ is the constant angular velocity due to the trajectory of the satellite and is perpendicular to the RLOS. Additionally, $\boldsymbol{\omega}^{z}_{base}$ and $\boldsymbol{\omega}^{y}_{base}$ are the components of the baseline angular velocity vector $\boldsymbol{\omega}_{base}$ that are perpendicular and parallel to the radar line of sight, respectively. These values can also be written as:

$$\boldsymbol{\omega}^{y}_{base} = (\boldsymbol{\omega}_{base} \cdot \hat{\mathbf{p}}_y)\hat{\mathbf{p}}_y \quad (\mathrm{B.30})$$

$$\boldsymbol{\omega}^{z}_{base} = \boldsymbol{\omega}_{base} - \boldsymbol{\omega}^{y}_{base} \quad (\mathrm{B.31})$$

$$\boldsymbol{\omega}_{orbit} = \frac{(\mathbf{v}_{sat} - \mathbf{v}_{sensor}) \times \hat{\mathbf{p}}_y}{\left[\frac{(\mathbf{r}_{sat} - \mathbf{r}_{sensor})}{\hat{\mathbf{p}}_y}\right]} \quad (\mathrm{B.32})$$

$$\boldsymbol{\omega}_{base} = \dot{\phi}\boldsymbol{L}_{base} + \dot{\psi}\boldsymbol{A}_{base}(:,3) \quad (\mathrm{B.33})$$

Substituting equations $(\mathrm{B.30})$ – $(\mathrm{B.33})$ in to equation $(\mathrm{B.29})$ yields the final corrected angular velocity vector in body-fixed coordinates:

$$\begin{aligned}\boldsymbol{\omega}'_{sat} = {} & \left(\left[\dot{\phi}\boldsymbol{L}_{base} + \dot{\psi}\boldsymbol{A}_{base}(:,3)\right] \cdot \hat{\mathbf{p}}_y\right)\hat{\mathbf{p}}_y \\ & + \frac{1}{s}\left[\left[\dot{\phi}\boldsymbol{L}_{base} + \dot{\psi}\boldsymbol{A}_{base}(:,3)\right] - \left(\left[\dot{\phi}\boldsymbol{L}_{base} + \dot{\psi}\boldsymbol{A}_{base}(:,3)\right] \cdot \hat{\mathbf{p}}_y\right)\hat{\mathbf{p}}_y\right] \\ & + \left(\frac{1-s}{s}\right)\left[\frac{(\mathbf{v}_{sat} - \mathbf{v}_{sensor}) \times \hat{\mathbf{p}}_y}{\frac{(\mathbf{r}_{sat} - \mathbf{r}_{sensor})}{\hat{\mathbf{p}}_y}}\right]\end{aligned} \quad (\mathrm{B.34})$$

Measured:$[\mathbf{v}]_i = \boldsymbol{STR}^{T}_{align}[\mathbf{v}]_B$

$$\omega^{z}_{meas} = \omega^{z}_{base}/s$$

Calculated: $[\mathbf{v}]_i = \boldsymbol{S}'\boldsymbol{T}'[\mathbf{v}]_B$

The matrix $\boldsymbol{S}$ applies a scaling factor to the cross-range component to allow for error in the nominal angular velocity used in generating the image. $\boldsymbol{S}$ and $\boldsymbol{S}'$ are therefore a function of the motion parameters and the nominal angular velocity $|\omega^{z}_{base}|$

$$\boldsymbol{S} = \begin{bmatrix} s & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{S}' = \begin{bmatrix} s' & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where s is the scale factor measured by the analyst during the image-model matching process (recorded by XELIAS in the *paramcdf* file) and $s' = |\omega^z_{calc}|/|\omega^z_{base}|$

[This page intentionally left blank.]

9 Appendix C – Algorithm Setup Procedures

File Types:

The *paramcdf* files contain the analyst measurements for each image in the *imagecdf* file; the *motioncdf* file, stores the motion profiles for the satellite; and the *iges* file, contains a description of the 3D structure of the satellite for computer graphics rendering and software manipulation.

Table C-1 XELIAS File Variables

Name	Symbol	Location	Description
motion_ref_time	t_0	*motioncdf*	Reference time of motion (seconds since 01 Jan 1970)
right_ascension	α	*motioncdf*	Right ascension of the angular momentum axis (degrees)
declination	δ	*motioncdf*	Declination of the angular momentum axis (degrees)
zinit	ψ_0	*motioncdf*	Initial precession angle of spin axis about angular momentum axis (degrees)
coning_angle	θ	*motioncdf*	Coning angle (degrees)
precession_rate	$\dot{\phi}$	*motioncdf*	Precession rate (degrees/sec)
xinit	ϕ_0	*motioncdf*	Initial angle about the spin axis (degrees)
spin_rate	$\dot{\psi}$	*motioncdf*	Spin rate (degrees/sec)
longitude	λ	*imagecdf*	Longitude of the radar sensor (degrees)
latitude	φ	*imagecdf*	Latitude of the radar sensor (degrees)
height	h	*imagecdf*	Altitude of the radar sensor (meters)
pass_date	d_0	*imagecdf*	Date of pass (seconds since 01 Jan 1970
start_time	t_i	*imagecdf*	Start time of image (seconds)
stop_time	t_f	*imagecdf*	Stop time of image (seconds)
x_scale	x_s	*imagecdf*	Doppler to meter conversion
range	ρ	*imagecdf*	Range (meters)
azimuth	Az	*imagecdf*	Azimuth (degrees)
elevation	El	*imagecdf*	Elevation (degrees)
range_rate	$\dot{\rho}$	*imagecdf*	Range rate (meters/sec)
azimuth_rate	$\dot{Az}$	*imagecdf*	Azimuth rate (degrees/sec)
elevation_rate	$\dot{El}$	*imagecdf*	Elevation rate (degrees/sec)
im_xscale	s	*paramcdf*	Cross-range scale factor
ms_number	y_n	*paramcdf*	Number of measurement points in image
ms_xcoord	y_x	*paramcdf*	x coordinate of measurement point (Hertz)
ms_ycoord	y_y	*paramcdf*	y coordinate of measurement point (meters)
wf_xrotate	ϕ	*paramcdf*	x rotation of wireframe model (degrees)
wf_yrotate	θ	*paramcdf*	y rotation of wireframe model (degrees)
wf_zrotate	ψ	*paramcdf*	z rotation of wireframe model (degrees)

vertices	*v*	*iges*	Vertex coordinates of the wireframe model
faces	*f*	*iges*	Order in which the vertices should be connected to create planar triangular facets

Saving and Formatting Measurement Times

1. Select Images from Database
    a. Enter the Satellite's Object Number and Check the Box that Filters the Images Based on this Criterion
    b. Select a Pass Number
2. Fit Wireframe Model to Images
3. Jump to Data Handling
4. Save Image Subset as *DDMMYY_images*.imagecdf
5. ncdump *DDMMYY_images*.imagecdf | head –n 2000 | parse_times.pl > ***TIMES.txt***

Saving Radar Images, Wireframe Models, and Motion Parameters

1. Select Images from Database
    a. Enter the Satellite's Object Number and Check the Box that Filters the Images Based on this Criterion
    b. Select a Pass Number
2. Fit Wireframe Model to Images
3. Jump to Data Handling
4. Under the Save Heading, Select Data Set Files
5. Select the "File" tab at the top of the "Data Handling" menu
6. Check the "Select for Save" boxes on the right hand side of the "Data Handling" menu for "Model…"
7. Enter in a save path: /h/*account name/DDMMYY_BSEKF/…*
8. Save the wireframe model as: */model name*.iges
9. Push the "Save" button at the bottom of the "Data Handling" menu
10. Under the "Save" heading, select "Image Subset"
11. Select the "File" tab at the top of the "Data Handling" menu
12. Check the "Select for Save" boxes on the right hand side of the "Data Handling" menu for "Image…" and "Param…"
13. Enter in a save path: /h/*account name/directory name/…*
14. Save the images as: /*DDMMMYY_ images*.imagecdf
15. Save the parameters as: /*DDMMYY_param*.paramcdf
16. Push the "Save Subset" button at the bottom of the "Data Handling" menu

Recovering and Parsing Files

1. Open unix shell
2. cd *h/account name/DDMMMYY_BSEKF*
3. ncdump *DDMMMYY_param*.paramcdf | parse_measurements.pl > ***EA_OBS.txt***

Transferring and Accessing Information Over the Network

Windows Machine

1. Start
2. Run
3. \\sweepea
4. Enter
5. …\homes\account name\DDMMMYY_BSEKF
6. Place STK files into the folder

SGI Machine

7. Open unix shell
8. cd *h/account name/DDMMMYY_BSEKF*
9. parse_stk_dates.pl < *DDMMMYY_coe.txt* > ***COE.txt***
10. parse_stk_dates.pl < *DDMMMYY_sun_int.txt* > ***SUN_INT.txt***
11. parse_stk_dates.pl < *DDMMMYY_llr.txt* > ***LLR.txt***
12. parse_stk_dates.pl < *DDMMMYY_sun_r.txt* > ***SUN_R.txt***

[This page intentionally left blank.]

10 Appendix D – Matlab Source Code

This Appendix provides the Matlab source code for the main backward-smoothing extended Kalman filter (BSEKF) function. In the interest of space, the numerous sub-functions which comprise the dynamics and measurement models are omitted from the text, but included in the CD attached to the back cover of this thesis.

```
%% LINCOLN ATTITUDE ESTIMATION SYSTEM (LAES) - MATLAB CODE %%
%%%2LT KYLE VOLPE
%%%MIT LINCOLN LABORATORY
%%%MIT DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
%%%UNITED STATES AIR FORCE

%% INPUT FILES:
%TIMES.txt - Measurement Times File [XELIAS]
%EA_OBS.txt - Deviation Measurements File [XELIAS]
%M_PARAM.txt - Baseline Motion Parameters File [XELIAS/Interactive Motion]
%COE.txt - Classical Orbital Elements File [STK]
%LLR.txt - Latitude/Longitude/Range File [STK]
%SAT_RV.txt - Satellite Position and Velocity File [STK]
%SUN_R.txt - Sun Position File [STK]
%SUN_INT.txt - Solar Intensity File [STK]
%B_FIELD.txt - Magnetic Field File [NOAA]
%A_DENSITY.txt - Atmospheric Density File [NRLMSISE-00]
%EARTH_R.txt - Earth Position File [NASA/JPL - HORIZONS]

%% OUTPUT FILES:
%run_time.txt
%state.txt
%covarience.txt
%q_error.txt
%ea_error.txt
%parameters.txt

%% INTERNAL FUNCTIONS:
%m_times
%ob_times
%xdot
%ydot
%qdot
%wdot
%idot
%ndot
%adot
%a_dot
%n_dot
%t_dot
%xpg_dot
%xpg_int
%xpg_pro
```

```
%ypro
%xpro
%qpro
%q_enforce
%n_param
%m_param
%f_model
%h_model
%p_model
%torque
%i_cache
%j_cost
%b_motion
%initial_xp
%eaobs

%% LIST OF VARIABLES:
%JCOST = Current Cost Function
%jcost = Previous Cost Function
%J_pn = Process Noise Subcost
%J_mn = Measurement Noise Subcost
%J_x = State Noise Subcost
%JAPPROX = Current Linearized Prediction of the Cost
%japprox = Previous Linearized Prediction of the Cost
%J_eps = Cost Function Epsilon Value
%q = Attitude Quaternion Vector
%w = Angular Velocity Vector
%I = Moment of Inertia Parameter Vector
%A = A(q) = Quaternion Rotation Matrix
%W = Angular Velocity Matrix
%MoI = Inertia Tensor
%N = Environmental Torques Vector (Body Referenced Accelerations)
%EA = Euler Angle Matrix
%t = Observation Time Vector [CACHE]
%x = f(x,w) = Predicted State Vector at the Current Gauss-Newton Iteration
(j)[CACHE]
%xj = Predicted State Vector at the Next Gauss-Newton Iteration (j+1)[CACHE]
%xhat = Modified State Vector Used in the Ordinary Differential Equation 45
%y = Measurement Vector [CACHE]
%yhat = h(x) = Predicted Measurement Vector [CACHE]
%w_pn = Process Noise Vector at the Current Gauss-Newton Iteration (j)[CACHE]
%wj_pn = Process Noise Vector at the Next Gauss-Newton Iteration (j+1)[CACHE]
%v_mn = dy = delta y = Measurement Noise Vector [CACHE]
%xhatstar = Updated State Estimate for the Cost Function
%Pstar = Updated State Estimation Error Covariance Matrix for the Cost
Function
%xdot = f(x,t) = Dynamics Function/State Propagator
%qdot = Kinematics Equation [Time Dependent Quaternion d/dt(q)]
%wdot = Dynamics Equation [Time Dependent Angular Velocity d/dt(w)]
%TIME = Predicted Time Vector [T_ic to T_ic+1]
%X = Predicted State Matrix at the Current Gauss-Newton Iteration (j) [X_ic
to X_ic+1]
%XJ = Predicted State Matrix at the Next Gauss-Newton Iteration (j+1) [XJ_ic
to XJ_ic+1]
%PHI = df/dx = Linearized Dynamics Matrix [State Jacobian]
%GAMMA = df/dw_pn = Linearized Dynamics Matrix [Process Noise Jacobian]
%H = dh/dx = Linearized Measurement Matrix [Measurement Jacobian]
```

```
%phi = df/dx = State Transition Matrix [CACHE]
%gamma = df/dw_pn = Process Noise Transition Matrix [CACHE]
%h = dh/dx = Measurement Transition Matrix [CACHE]
%R = Measurement Noise Covariance Matrix
%Q = Process Noise Covariance Matrix
%P = State Error Covariance Matrix
%Rvv = Square Root Information Matrix Associated with the Measurement Noise
(v_mn)
%Rww = Square Root Information Matrix Associated with the Process Noise
(w_pn)
%Rxx = Square Root Information Matrix Associated with the Initial State Cost
Function
%Rbar_wx = Square Root Information Matrix Associated with the State and
Process Noise After Being QR Factored
%Rbar_ww = Square Root Information Matrix Associated with the Process Noise
After Being QR Factored
%dz_r = Change in the Weighted Cost Related to the Measurement Noise
%dzbar_w = Change in the Weighted Cost Related to the Process Noise
%dz_x = Change in the Weighted Cost Related to the State
%dx = Change in the State Vector
%dw_pn = Change in the Process Noise Vector
%k = Observation Counter
%n = Total Number of Observations
%m = m_buffer = Collection of Previous States, Observations, Process Noise
Vectors, and Covariance Sqrt to be Filtered Forward and Smoothed Backward
%m_target = "Window/Buffer Size"
%j = Gauss-Newton Iteration Counter
%j_max = Maximum Gauss-Newton Iteration Count
%term_crit = Termination Criteria
%g_count = Trial Search Step Size Counter
%G = gamma = Current Trial Search Step Size
%g = Previous Trial Search Step Size
%G_eps = Trial Search Step Size Epsilon Value
%Y = Measurement Vectors [Y_1 to Y_k]
%T = Observation Times [T_1 to T_k]
%B = Block Matrix to be QR Factored
%Z = Block Matrix Containing the Vectors dzbar_w, dz_x, and dz_r
%T_qr = Q Matrix That Results From the Factorization of B
%R_qr = R Matrix That Results From the Factorization of B

%% BACKWARD SMOOTHING EXTENDED KALMAN FILTER %%

%START THE ALGORITHM TIMER
START_TIME = clock;

%SELECT ALGORITHM RUN LOCATION
%%%HOME COMPUTER = 1
%%%WORK COMPUTER = 2
%%%NOMAD2 COMPUTER = 3
LOCATION = 1;

%TURN PARALLELISM ON OR OFF
PARALLEL = 0; %%%OK TO CHANGE%%%

%TURN ENVIRONMENTAL TORQUES ON OR OFF
%%%+ GRAVITY-GRADIENT = 0
```

```
%%%+ MAGNETIC FIELD = 1
%%%+ AERODYNAMIC DRAG = 2
%%%+ SOLAR RADIATION PRESSURE = 3
TORQUE = 0; %%%OK TO CHANGE%%%

%SELECT THE TYPE OF TEST CASE TO RUN:
%%% REAL TEST CASE = 1
%%% SYMMULATED TEST CASE = 2
TEST = 1; %%%OK TO CHANGE%%%

%SELECT THE PROPAGATION METHOD TO USE:
%%% ANALYITIC EQUATIONS = 1
%%% FINITE DIFFERENCING = 2
PROPAGATOR = 1; %%%OK TO CHANGE%%%

%SELECT THE MOMENT OF INERTIA MODEL TO USE:
%%% GEOMETRIC MODEL = 1
%%% SYMMETRIC MODEL = 2
%%% XELIAS MODEL = 3
INERTIA = 1; %%%OK TO CHANGE%%%

%SET THE NUMBER OF PASSES TO BE PROCESSED:
PASSES = 2; %%%OK TO CHANGE%%%

%TURN GRAPHS ON OR OFF
GRAPH = 1; %%%OK TO CHANGE%%%

%TURN PROFILE ON OR OFF
%profile on %%%OK TO CHANGE%%%

if LOCATION == 1
    PATH = ('C:\Documents and Settings\Kyle Volpe\Desktop\LAES_BSEKF\');
elseif LOCATION == 2
    PATH = ('C:\Documents and
    Settings\ky20900\Desktop\THESIS_LAES\LAES_BSEKF\');
elseif LOCATION == 3
    PATH = ('/home/ky20900/LAES_BSEKF/');
end

%SPECIFY THE FOLDER(S) IN WHICH TO SAVE THE FOLLOWING OUTPUT FILES:
sta = fopen([PATH,'state.txt'],'wt');
res1 = fopen([PATH,'q_residuals.txt'],'wt');
res2 = fopen([PATH,'ea_residuals.txt'],'wt');
res3 = fopen([PATH,'attitude_error.txt'],'wt');
ela = fopen([PATH,'elapsed_t.txt'],'wt');
cov = fopen([PATH,'covariance.txt'],'wt');

if PARALLEL == 1
    matlabpool open
    %pmode start local 4
end

%SET THE TARGET SIZE OF FILTER WINDOW [M_BUFFER] (m_target)
m_target=40; %%%OK TO CHANGE%%%
```

```
%SET THE MAXIMUM NUMBER OF GAUSS-NEWTON ITERATIONS (j_max)
j_max=15; %%%OK TO CHANGE%%%

%SET THE MAXIMUM NUMERICAL INTEGRATION STEP SIZE (s_max)
s_max=0.5; %%%OK TO CHANGE%%%

%SET THE FOLLOWING COUNTERS:
m=1;

%SET THE FOLLOWING EPSILON VALUES:
G_eps=1e-10; %%%OK TO CHANGE%%%
J_eps=1e-10; %%%OK TO CHANGE%%%

%% BEGIN PROGRAM

%INITIALIZE THE FOLLOWING VECTORS AND MATRICES
T=[];
TIME=[];
Y_ea=[];
w=[];
n=0;

%GENERATE THE ATTITUDE MEASUREMENTS USING THE BASELINE MOTION AND CORRECTION
MEASUREMENTS
for PASS = 1:PASSES
    %SET THE BASELINE MOTION PARAMETERS AND REFERENCE DATE FOR EACH PASS
    [DATE,RA,D,CA,ZI,XI,PP,SP] = b_motion(PASS);

    %EDIT AND COLLECT THE EULER ANGLE (DEVIATION) MEASUREMENTS (EA)
    [EA,N] = eaobs(PASS);

    %EDIT THE OBSERVATION TIMES
    [time,t] = m_times(PASS);
    DN = ob_times(DATE);

    %ADD THE XELIAS MOTION PARAMETERS TO THE EULER ANGLE MEASUREMENTS TO GET
    THE TRUE ATTITUDE MEASUREMENTS (Y)
    [Y,MoI,W] = a_obs(t,RA,D,CA,ZI,XI,PP,SP,EA);

    %COMBINE THE OBSERVATIONS
    T=[T;t];
    TIME=[TIME;time];
    Y_ea=[Y_ea,Y];
    w=[w,W];
    n=n+N;
end

%INITIALIZE ALL THE CACHES THAT WILL BE USED BY THE FILTER
[y_ea,y_q,t,x,w_pn,xj,wj_pn,Q_pn,Q,yhat_ea,yhat_q,v_mn,V_mn,phi,gamma,h,Rvv,R
ww,R_ww,Rxx,Rbar_wx,Rbar_ww,dz_r,dzbar_w,dz_x,dx,dw_pn,q4_s,qj4_s,y_mn] =
i_cache(n,m_target);

%CALCULATE THE INITIAL ATTITUDE QUATERNION, ANGULAR VELOCITY, AND MOMENT OF
INERTIA PARAMETERS VECTORS (q, w, and I):
```

```
%CALCULATE THE MOMENT OF INERTIA MATRIX USING EITHER THE GEOMETRIC MODEL OR
THE SYMMETRIC MODEL
if INERTIA == 1
    %GEOMETRIC MODEL (ASTMMETRIC MODEL)
    [M,CoM,MoI] = p_model();
    disp('GEOMETRIC MODEL')
    disp(MoI)
elseif INERTIA == 2
    %FULLY SYMMETRIC MODEL
    MoI=10*eye(3); %%%OK TO CHANGE%%%
    disp('FULLY SYMMETRIC MODEL')
    disp(MoI)
elseif INERTIA == 3
    %XELIAS MODEL (PARTIALLY SYMMETRIC)
    disp('XELIAS MODEL')
    disp(MoI)
end

%ASSEMBLE THE INITIAL STATE VECTOR (x); MEASUREMENT NOISE COVARIANCE MATRIX
(R_mn); AND INITIAL STATE COVARIANCE MATRIX (P)
[Y_ea,Y_q,X,q4_s,p,R_mn] = initial_xp(Y_ea,MoI,T);
x(:,1)=X;

%COMPARE THE ANGULAR VELOCITIES DERIVED USING THE TWO DIFFERENT METHODS
disp('SYMMETRIC MOTION PARAMETER ANGULAR VELOCITY VS CORRECTED LINEAR FIT
ANGULAR VELOCITY')
disp([w(:,1),x(4:6,1)])

%SAVE THE FULL 13 x 1 INITIAL STATE VECTOR TO A TEXT FILE
fprintf(sta,'%f %f %f %f %f %f %f %f %f %f %f %f
%f\n',x(1,1),x(2,1),x(3,1),q4_s(1)*sqrt(1-x(1,1)^2-x(2,1)^2-
x(3,1)^2),x(4,1),x(5,1),x(6,1),x(7,1),x(8,1),x(9,1),x(10,1),x(11,1),x(12,1));

%DELETE THE FIRST COLUMN OF THE MEASUREMENT MATRIX (Y)
Ybar_ea=Y_ea;
Ybar_q=Y_q;
Y_ea(:,1)=[];
Y_q(:,1)=[];

%CALCULATE THE FOLLOWING TORQUE MODEL INPUT PARAMETERS
[V_SAT,R_SUN,SOLAR_I,R_SAT,B_FIELD,CoP,U_NORM,S_AREA,A_DENSITY] =
n_param(DN,T,TORQUE);

%RESET THE TOTAL NUMBER OF OBSERVATIONS (n)
s=size(Y_q);
n=s(1,2);

%SET THE INITIAL PROCESS NOISE VECTORS (w_pn) [3 x 1 Disturbance Torque]
w_pn(:,1)=zeros(12,1);

%SET THE INITIAL STATE COVARIANCE MATRIX (P)
p=[5e-2;5e-2;5e-2;5e-3;5e-3;5e-3;10;10;0.0001;8e-1;8e-1;8e-1];
P(:,:,1)=diag(p);
```

```
%CALCULATE THE SQRT INFORMATION MATRIX ASSOCIATED WITH THE STATE (Rxx)
%Factor P_0 into [Rxx^-1][Rxx^-T]using Cholesky decomposition
Rxx(:,:,1)=chol(P(:,:,1)^-1);

%CALCULATE THE PROCESS NOISE COVARIANCE MATRICES (Q_pn) [TUNE FILTER]
%Q=I_3x3*Process Error Variance
Q_pn(:,:,1)=eye(3)*1e-9; %%%OK TO CHANGE%%%
for k=1:n
    Q_pn(:,:,k+1)=eye(3)*(((1e-9))/(T(k+1)-T(k))); %%%OK TO CHANGE%%%
end

%CALCULATE THE LINEAR APPROXIMATION TO THE MEASUREMENT NOISE COVARIANCE
MATRIX (R_mn)
%R_mn=I_3x3*Measurement Error Variance Per Axis in Radians
R_mn=eye(4)*0.5e-2; %%%OK TO CHANGE%%%

%DECIDE IF THE PROPAGATED STATE VECTOR SHOULD BE SAVED AS A NEW SET OF
OBSERVATIONS
SAVE = 1;

%PROJECT THE INITIAL STATE FORWARD OVER THE TIME SPAN OF INTEREST
[Yhat_ea,Yhat_q,ybar_ea,ybar_q] =
ydot(DN,T,Y_ea,x,q4_s,s_max,CoP,U_NORM,V_SAT,R_SUN,SOLAR_I,A_DENSITY,B_FIELD,
S_AREA,R_SAT,SAVE,PATH,TEST);

if TEST == 2
    %SAVE THE FULL SET OF NOISY MEASUREMENTS
    Ybar_q=ybar_q;
    Ybar_ea=ybar_ea;

    %RESET THE OBSERVATION CACHE
    Y_q=ybar_q;
    Y_ea=ybar_ea;

    %SET THE NEW INITIAL STATE VECTOR, COVARIANCE MATRIX, AND OBSERVATIONS
    x(:,1)=[0.140134360644793;0.941434088194620;0.145834358774792;
    0.002705006174102;-0.002116872057220;-0.007055041225524;
    7.745966692414834;7.745966692414834;7.745966692414834;0;0;0];
    Y_q(:,1)=[];
    Y_ea(:,1)=[];
end

%BEGIN THE OBSERVATION LOOP
for k = 1:n

    %START THE OBSERVATION TIMER
    tic

    %DISPLAY THE CURRENT OBSERVATION NUMBER
    disp('CURRENT OBSERVATION')
    disp(k)

    %PERFORM THE FOLLOWING ASSIGNMENTS AND CALCULATE xhatstar and Pstar
    if k > m_target
        %%% THE M_BUFFER HAS BEEN FILLED AND VALUES WILL BE REPLACED RATHER
```

```
THAN APPENDED %%%

%MAKE THE FOLLOWING ASSIGNMENTS:

%SHIFT THE STATE CACHE
x=circshift(x,[0,m_target]);
x(:,end)=0;

%SHIFT THE 4TH QUATERNION COMPONENT SIGN CACHE
q4_s=circshift(q4_s,[0,m_target]);
q4_s(end)=0;

%SHIFT THE PROCESS NOISE CACHE AND SET THE LAST COLUMN EQUAL TO ZERO
w_pn=circshift(w_pn,[0,m_target]);
w_pn(:,end)=0;

%SHIFT THE CHANGE IN WEIGHT COST RELATED TO THE STATE VECTOR
dz_x(:,1)=dz_x(:,2);

%SHIFT THE STATE SQUARE ROOT INFORMATION MATRIX
Rxx(:,:,1)=Rxx(:,:,2);

if k == 64
    %CALCULATE THE SQRT INFORMATION MATRIX ASSOCIATED WITH THE STATE
    (Rxx)

    %Factor P_0 into [Rxx^-1][Rxx^-T]using Cholesky decomposition
    p=[2.5e-2;2.5e-2;2.5e-2;2.5e-3;2.5e-3;2.5e-3;5;5;0.00001;4e-1;
    4e-1;4e-1];
    P(:,:,1)=diag(p);
    Rxx(:,:,1)=chol(P(:,:,1)^-1);
end

%CHECK THE DIAGONAL OF Rxx FOR NEGATIVE VALUES AND NEGATE THE MATRIX
IF THERE ARE
if sum(diag(Rxx(:,:,1))) <= 0
    Rxx(:,:,1)=-Rxx(:,:,1);
end

%CALCULATE THE UPDATED STATE ESTIMATE FOR THE COST (xhatstar)
%xhat=(Rxx(:,:,1)^-1)*dz_x(:,1);
xhatstar=x(:,1)+(Rxx(:,:,1)^-1)*dz_x(:,1);

%ENFORCE THE CONSTRAINT ON THE QUATERNION
[cq,qstar4_s] = q_enforce(x(1:3,1),q4_s(1),((Rxx(1:3,1:3,1)^-1)
*dz_x(1:3,1)),qz4_s(1),1);

%REVISE THE STATE ESTIMATE FOR THE COST (xhatstar)
xhatstar(1:3,1)=cq(1:3,1);

%CALCULATE THE STATE ESTIMATION ERROR COVARIANCE FOR THE COST (Pstar)
AND ITS INVERSE (P_inv)
Pstar=(Rxx(:,:,1)^-1)*((Rxx(:,:,1)')^-1);
P_inv=Rxx(:,:,1)'*Rxx(:,:,1);
```

```
else %k <= m_target
    %%% THE M_BUFFER IS STILL BEING FILLED %%%

    %ASSIGN THE UPDATED STATE ESTIMATE FOR THE COST (xhatstar)
    xhatstar=x(:,1);

    %RECORD THE SIGN OF THE 4TH QUATERNION COMPONENT (qstar4_s)
    qstar4_s=q4_s(1);

    %ASSIGN THE STATE ESTIMATION ERROR COVARIANCE FOR THE COST (Pstar)
    AND ITS INVERSE (P_inv)
    Pstar=P(:,:,1);
    P_inv=Rxx(:,:,1)'*Rxx(:,:,1);
end

%SET m
if k <= m_target
    m=k;
    shift_obs = false;
else %k > m_target
    m=m_target;
    shift_obs = true;
end

%CREATE OR MODIFY THE CACHE
if k > m_target && shift_obs == true
    %%% ONCE THE CACHE IS FULL SHIFT THE ARRAY AND REPLACE THE LAST
    COLUMN WITH THE NEWEST VECTOR %%%

    %SHIFT THE OBSERVATION CACHE
    y_ea=circshift(y_ea,[0,m-1]);
    y_ea(:,end)=Y_ea(:,k);

    y_q=circshift(y_q,[0,m-1]);
    y_q(:,end)=Y_q(:,k);

    %SHIFT THE PREDICTED MEASUREMENT CACHE
    yhat_ea=circshift(yhat_ea,[0,m_target]);
    yhat_ea(:,end)=0;

    yhat_q=circshift(yhat_q,[0,m_target]);
    yhat_q(:,end)=0;

    %SHIFT THE TIME CACHE
    t=circshift(t,[m,0]);
    t(end)=T(k+1);

    %SHIFT THE PROCESS NOISE COVARIANCE CACHE
    ic=1;
    for i=k-(m-1):k+1
        Q(:,:,ic)=Q_pn(:,:,i);
        ic=ic+1;
    end
    %REASSIGN THE MEASUREMENT NOISE COVARIANCE CACHE
    R=R_mn;

else %k <= m_target & shift_obs == false
    %%% GROW THE CACHES TO THE SIZE OF THE M_BUFFER %%%

    %CREATE THE OBSERVATION CACHE
    y_ea=Y_ea(:,1:k);
    y_q=Y_q(:,1:k);

    %CREATE THE TIME CACHE
    t=T(1:k+1);

    %CREATE THE PROCESS NOISE COVARIANCE CACHE
    for i=1:k+1
        Q(:,:,i)=Q_pn(:,:,i);
    end

    %CREATE THE MEASUREMENT NOISE COVARIANCE CACHE
    R=R_mn;
end

%INITIALIZE THE COST FUNCTION Jcost
JCOST=0;
jcost=JCOST;

%SET THE TERMINATION CRITERIA FOR THE GAUSS-NEWTON ITERATION
term_crit=0;

%RESET THE Japprox CONVERGENCE VARIABLE
JAPPROX=0;
japprox=JAPPROX;

%SET THE GAUSS-NEWTON ITERATION COUNTER j TO ZERO
j=0;

%BEGIN THE GAUSS-NEWTON LOOP
while j < j_max;

    for i = k-m:k-1

        %SET THE INDEX COUNTER
        ic=i-((k-m)-1);

        %EXECUTE THE DYNAMICS FUNCTION/STATE PROPAGATOR (PREDICT NEXT
        STATE):

        %SET THE STATE PROPAGATION DIRECTION
        DIRECT = 1; %%%FORWARD = 1 | BACKWARD = -1%%%

        %DETERMINE IF THE PARTIAL DERIVATIVES NEED TO BE CALCULATED
        DERIV = 1; %%%NO = 0 | YES = 1%%%

        %CALCULATE THE NEXT STATE VECTOR (X) AND THE STATE AND PROCESS
```

```
      NOISE TRANSITION MATRICES (PHI AND GAMMA)
      [X,PHI,GAMMA] = xpg_pro(DN,t(ic:ic+1),x(:,ic),q4_s(ic),s_max,
      w_pn(4:6,ic),CoP,U_NORM,V_SAT,R_SUN,SOLAR_I,A_DENSITY,B_FIELD,
      S_AREA,R_SAT,TORQUE,DERIV);

      %DETERMINE IF THE PARTIAL DERIVATIVES NEED TO BE CALCULATED
      DERIV = 1; %%%NO = 0 | YES = 1%%%

      %USE THE OBSERVATION MODEL TO CALCULATE THE MEASUREMENT NOISE
      (v_mn) AND MEASUREMENT TRANSITION MATRIX [H]
      [x,q4_s,yhat_q,yhat_ea,v_mn,V_mn,y_mn,H] = hdot(X,x,q4_s,y_q,
      y_ea,yhat_q,yhat_ea,v_mn,V_mn,y_mn,ic,k,m_target,DERIV);

      %ASSIGN THE OUTPUT JACOBIAN MATRICES PHI [df/dx], GAMMA
      [df/dw_pn], AND H [dh/dx] TO THEIR RESPECTIVE CACHES (phi, gamma,
      AND h)
      phi(:,:,ic)=PHI;
      gamma(:,:,ic)=GAMMA;
      h(:,:,ic+1)=H;

      %FACTOR THE PROCESS NOISE MATRIX Q INTO [Rww^-1][Rww^-T]
      Rww(:,:,ic)=chol(Q(:,:,ic)^-1);
      R_ww(4:6,4:6,ic)=Rww(:,:,ic);

      %FACTOR THE MEASUREMENT NOISE MATRIX R INTO [R_vv^-1][R_vv^-T]
      Rvv(:,:,ic+1)=chol(R^-1);
  end

  %CALCULATE THE COST FUNCTION Jcost AT ITERATION j IF THE COST HAS
NEVER BEEN COMPUTED BEFORE FOR THIS OBSERVATION k
  if JCOST == 0
      JCOST = j_cost(k,m,x(:,1),w_pn,v_mn,Rww,Rvv,xhatstar,Pstar,P_inv,
      q4_s(1),qstar4_s(1));
  end

  %COMPUTE THE INITIAL CHANGE IN WEIGHT COST RELATED TO THE STATE
  VECTOR (dz_x)
  dz_x=Rxx(:,:,1)*(xhatstar-x(:,1));

  %ENFORCE THE CONSTRAINT ON THE QUATERNION
  [dq,qz4_s] = q_enforce(x(1:3,1),q4_s(1),xhatstar(1:3,1),qstar4_s(1),
  -1);

  %REVISE THE CHANGE IN WEIGHT COST RELATED TO THE STATE VECTOR (dz_x)
  dz_x(1:3,1)=Rxx(1:3,1:3,1)*dq(1:3,1);

  %ITERATE FOR THE FORWARD FILTER
  ic=1;
  for i = k-m:k-1

      %QR FACTORIZATION
      B=double([R_ww(:,:,ic),zeros(12,12);-Rxx(:,:,ic)*phi(:,:,ic)^-1
      *gamma(:,:,ic),Rxx(:,:,ic)*phi(:,:,ic)^-1;zeros(4,12),
      Rvv(:,:,ic+1)*h(:,:,ic+1)]);
```

```
    [T_qr,R_qr]=qr(B);
    Rbar_ww(:,:,ic)=R_qr(1:12,1:12);
    Rbar_wx(:,:,ic)=R_qr(1:12,13:24);
    Rxx(:,:,ic+1)=R_qr(13:24,13:24);

    %Z BLOCK VECTORS
    Z=T_qr'*[-R_ww(:,:,ic)*w_pn(:,ic);dz_x(:,ic);Rvv(:,:,ic+1)
    *v_mn(:,ic+1)];
    dzbar_w(:,ic)=Z(1:12,:);
    dz_x(:,ic+1)=Z(13:24,:);
    dz_r(:,ic)=Z(25:28,:);

    %INCREMENT THE ITERATION COUNTER
    ic=ic+1;
end

%CALCULATE dx(k)
dx(:,ic)=Rxx(:,:,ic)^-1*dz_x(:,ic);
%i=k-1 should be true
%Set ic=ic+1 so that the kth stage is processed

%BACKWARD SMOOTHING
ic=ic-1;
for i = k-1:-1:k-m

    %CALCULATE dw_pn
    Rbar_ww_inv=zeros(12);
    Rbar_ww_inv(4:6,4:6)=Rbar_ww(4:6,4:6,ic)^-1;
    dw_pn(:,ic)=Rbar_ww_inv*(dzbar_w(:,ic)-Rbar_wx(:,:,ic)
    *dx(:,ic+1));

    %CALCULATE dx(i)
    dx(:,ic)=phi(:,:,ic)^-1*(dx(:,ic+1)-gamma(:,:,ic)*dw_pn(:,ic));

    %INCREMENT THE ITERATION COUNTER IN REVERSE
    ic=ic-1;
end

%END OF SMOOTHING PROBLEM SOLUTION; RESET THE GAMMA VALUE (G) TO ONE
G=1;
g_count=0;
g=G;

%SAVE THE PREVIOUS COSTS TO jcost AND japprox
jcost=JCOST;
japprox=JAPPROX;

while (1)

    if q4_s(1) == 0
        STOP=1;
    end
```

```
%CALCULATE THE UPDATED STATE VECTOR (xj)
xj(:,1)=x(:,1)+G*dx(:,1);

if G ~= 0
    %RECORD THE SIGN OF THE QUATERNION COMPONENTS (q_s)
    qj_s=sign(xj(1:3,1));

    %ENFORCE THE CONSTRAINT ON THE QUATERNION
    [cq,qj4_s] = q_enforce(x(1:3,1),q4_s(1),(G*dx(1:3,1)),q4_s(1)
    ,1);

    %CHECK THE SIGN OF THE UPDATED QUATERNION
    cq_s=sign(cq(1:3,1));

    if sum(qj_s+cq_s) == 0
        %CHANGE THE SIGN ON THE UNKNOWN FOURTH COMPONENT OF THE
        STATE CHANGE VECTOR AND RE-ENFORCE THE CONSTRAINT ON THE
        QUATERNION
        [cq,qj4_s] = q_enforce(x(1:3,1),q4_s(1),(G*dx(1:3,1))
        ,-q4_s(1),1);
    end

    %REVISE THE UPDATED STATE VECTOR (xj)
    xj(1:3,1)=cq(1:3,1);
end

for i = k-m:k-1

    %SET THE INDEX COUNTER
    ic=i-((k-m)-1);

    %CALCULATE wj_pn
    wj_pn(:,ic)=w_pn(:,ic)+G*dw_pn(:,ic);

    %EXECUTE THE DYNAMICS FUNCTION/STATE PROPAGATOR (PREDICT NEXT
    STATE):

    %SET THE STATE PROPAGATION DIRECTION
    DIRECT = 1; %%%FORWARD = 1 | BACKWARD = -1%%%

    %DETERMINE IF THE PARTIAL DERIVATIVES NEED TO BE CALCULATED
    DERIV = 0; %%%NO = 0 | YES = 1%%%

    %CALCULATE THE NEXT STATE VECTOR (X) AND THE STATE AND
    PROCESS NOISE TRANSITION MATRICES (PHI AND GAMMA)
    [XJ,PHI,GAMMA] = fdot(DN,t(ic:ic+1),xj(:,ic),qj4_s(ic),
    wj_pn(4:6,ic),CoP,U_NORM,V_SAT,R_SUN,SOLAR_I,A_DENSITY,
    B_FIELD,S_AREA,R_SAT,TORQUE,DERIV,DIRECT);

    %DETERMINE IF THE PARTIAL DERIVATIVES NEED TO BE CALCULATED
    DERIV = 0; %%%NO = 0 | YES = 1%%%

    %USE THE FOLLOWING OBSERVATION MODEL TO CALCULATE THE
    MEASUREMENT NOISE (v_mn) AND MEASUREMENT TRANSITION MATRIX
    [H]
    [xj,qj4_s,yhat_q,yhat_ea,v_mn,V_mn,y_mn,H] = hdot(XJ,xj,
    qj4_s,y_q,y_ea,yhat_q,yhat_ea,v_mn,V_mn,y_mn,ic,k,
    m_target,DERIV);

end

%CALCULATE THE NEW COST FUNCTION (JCOST)
JCOST = j_cost(k,m,xj(:,1),wj_pn,v_mn,Rww,Rvv,xhatstar,Pstar,
P_inv,qj4_s(1),qstar4_s(1));

%CALCULATE THE APPROXIMATE COST FUNCTION (JAPPROX)
dz_r_sum=0;

ic=1;
for i = k-m:k-1
    dz_r_sum=dz_r_sum+(dz_r(:,ic)'*dz_r(:,ic));
    ic=ic+1;
end

%SET THE CONVERGENCE VARIABLE JAPPROX
JAPPROX=0.5*dz_r_sum;

%CHECK WHETHER THE COST HAS INCREASED
if JCOST >= jcost;
    g=G;
    G=0.5*G;
    g_count=g_count+1;
else
    break
end

%IF THE CURRENT TRIAL SEARCH STEP SIZE (g) WAS SET TO ZERO IN THE
PREVIOUS ITERATION, THEN EXIT
if G == 0
    break
end

%IF THE CURRENT TRIAL SEARCH STEP SIZE (g) GETS TO BE VERY CLOSE
TO ZERO, THEN PRINT A WARNING AND EXIT
if G <= G_eps
    disp('WARNING: TRIAL SEARCH STEP SIZE NEAR ZERO')
    disp('GAMMA COUNT (g_count)')
    disp(g_count)
    disp('PREVIOUS COST VALUE (jcost)')
    disp(jcost)
    disp('CURRENT COST VALUE (JCOST)')
    disp(JCOST)
    disp('PREVIOUS LINEARIZED PREDICTION OF THE COST VALUE
    (japprox)')
    disp(japprox)
    disp('CURRENT LINEARIZED PREDICTION OF THE COST VALUE
    (JAPPROX)')
    disp(JAPPROX)
    G=0;
    end
end

%TEST THE TERMINATION CRITERION
if abs(JAPPROX-japprox) <= J_eps
    term_crit=1;
end

%SET THE GAUSS-NEWTON STATE CACHE (xj) TO THE STATE CACHE (x) AND THE
GAUSS-NEWTON PROCESS NOISE CACHE (wj_pn) TO THE PROCESS NOISE CACHE
(w_pn)
ic=1;
for i = k-m:k
    x(:,ic)=xj(:,ic);
    q4_s(ic)=qj4_s(ic);
    w_pn(:,ic)=wj_pn(:,ic);
    ic=ic+1;
end

%ASSESS THE TERMINATION CRITERIA FOR THE GAUSS-NEWTON ITERATION
if term_crit == 1 || j+1 == j_max
    %DISPLAY THE NUMBER OF ITERATIONS NEEDED
    disp('NUMBER OF GAUSS-NEWTON ITERATIONS (j)')
    disp(j)

    %DISPLAY THE CURRENT SIZE OF THE M-BUFFER
    disp('CURRENT SIZE OF THE M-BUFFER (m)')
    disp(m)

    %DISPLAY THE COST VALUES
    disp('PREVIOUS COST VALUE (jcost)')
    disp(jcost)
    disp('CURRENT COST VALUE (JCOST)')
    disp(JCOST)
    disp('PREVIOUS LINEARIZED PREDICTION OF THE COST VALUE
    (japprox)')
    disp(japprox)
    disp('CURRENT LINEARIZED PREDICTION OF THE COST VALUE (JAPPROX)')
    disp(JAPPROX)

    %DISPLAY THE STATE VECTOR
    disp('STATE VECTOR (x)')
    disp(x(:,m+1))

    %SAVE THE FULL 13 x 1 ESTIMATED STATE VECTOR TO A TEXT FILE
    fprintf(sta,'%f %f %f %f %f %f %f %f %f %f %f %f %f\n',
    x(1,m+1),x(2,m+1),x(3,m+1),q4_s(m+1)*sqrt(1-x(1,m+1)^2
    -x(2,m+1)^2-x(3,m+1)^2),x(4,m+1),x(5,m+1),x(6,m+1),x(7,m+1),
    x(8,m+1),x(9,m+1),x(10,m+1),x(11,m+1),x(12,m+1));

    if TEST == 2
        %CALCULATE THE CURRENT ESTIMATED MOMENT OF INERTIA MATRIX
        (MoI_est)
        I=x(7:12,m+1);
        C=[cos(I(5))*cos(I(6)),cos(I(5))*sin(I(6)),-sin(I(5));
```

```
            sin(I(4))*sin(I(5))*cos(I(6))-cos(I(4))*sin(I(6)),sin(I(4))
            *sin(I(5))*sin(I(6))+cos(I(4))*cos(I(6)),sin(I(4))*cos(I(5));
            cos(I(4))*sin(I(5))*cos(I(6))+sin(I(4))*sin(I(6)),cos(I(4))
            *sin(I(5))*sin(I(6))-sin(I(4))*cos(I(6)),cos(I(4))
            *cos(I(5))];
            MoI_est=C*diag([(I(2)^2+I(3)^2)/12,(I(1)^2+I(3)^2)/12,
            (I(1)^2+I(2)^2)/12])*C';

            %CALCULATE THE MoI_est FRACTION ERROR NORM (MoI_error) [ERROR
            IN ESTIMATED MOMENT OF INERTIA MATRIX (MoI_est) RELATIVE TO
            THE TRUE INERTIA MATRIX (MoI)]
            MoI_error(k)=norm(MoI_est*trace(MoI)/trace(MoI_est)-
            MoI)/norm(MoI);

            %DISPLAY THE FRACTION ERROR NORM
            disp('FRACTION ERROR NORM FOR MoI_est')
            disp(MoI_error)
        end

        %DISPLAY AND SAVE THE FOLLOWING RESIDUALS:
        disp('QUATERNION RESIDUAL/MEASUREMENT ERROR (v_mn)')
        dy=v_mn(:,m+1);
        disp(dy)
        fprintf(res1,'%f %f %f %f\n',dy(1),dy(2),dy(3),dy(4));

        disp('EULER ANGLE RESIDUAL/MEASUREMENT ERROR (V_mn)')
        dy=V_mn(:,m+1);
        disp(dy)
        fprintf(res2,'%f %f %f \n',dy(1),dy(2),dy(3));

        disp('QUATERNION ATTITUDE ERROR (y_mn)')
        dy=y_mn(:,m+1);
        disp(dy)
        fprintf(res3,'%f %f %f %f\n',dy(1),dy(2),dy(3),dy(4));

        %DISPLAY AND SAVE THE STATE ERROR COVARIENCE MATRIX
        disp('STATE ERROR COVARIENCE (P)')
        Phat=(Rxx(:,:,m)^-1)*((Rxx(:,:,m)')^-1);
        phat=diag(Phat);
        disp(phat)
        fprintf(cov,'%f %f %f %f %f %f %f %f %f %f %f
        %f\n',phat(1),phat(2),phat(3),phat(4),phat(5),phat(6),phat(7),pha
        t(8),phat(9),phat(10),phat(11),phat(12));

        %DISPLAY THE AMOUNT OF TIME NEEDED TO PROCESS EACH OBSERVATION
        disp('TIME TO PROCESS THE CURRENT OBSERVATION')
        toc %sec
        e_time=toc/60; %min
        fprintf(ela,'%f %f\n',k,e_time);

        %SET THE GAUSS-NEWTON ITERATION COUNTER (j) EQUAL TO THE MAXIMUM
        VALUE (j_max)
        j=j_max;
    else
        j=j+1;
```

```
            %%% DISPLAY THE CURRENT GAUSS-NEWTON ITERATION %%%
            %disp('GAUSS-NEWTON ITERATION')
            %disp(j)
        end
    end
end

%CLOSE THE OUTPUT TEXT FILES
fclose(sta);
fclose(res1);
fclose(res2);
fclose(res3);
fclose(ela);
fclose(cov);

%DISPLAY THE FINAL PREDICTED AND MEASURED ATTITUDE ALONG WITH THE ERROR ABOUT
EACH AXIS, AND THE TOTAL ERROR
disp('FINAL PREDICTED AND MEASURED ATTITUDE [X Y ERROR (deg)]')
compare=[yhat_q(:,end),y_q(:,end),v_mn(:,end)];
disp(compare)
disp('TOTAL ERROR (deg)')
disp(norm(v_mn(:,end)))

%VIEW THE PROFILE REPORT
%profile viewer

%GRAPH BSEKF RESULTS
if GRAPH == 1
    %PLOT THE EULER ANGLE ERROR ABOUT EACH AXIS
    load ea_residuals.txt
    figure
    hold on
    grid on
    plot(ea_residuals(1:end,1),'r.')
    plot(ea_residuals(1:end,2),'b.')
    plot(ea_residuals(1:end,3),'g.')
    title('EULER ANGLE ERROR ABOUT EACH AXIS OF ROTATION')
    xlabel('DISCRETE-TIME OBSERVATION NUMBER')
    ylabel('RESIDUAL (deg)')
    legend('PHI (Z)','THETA (X)','PSI (Z)')

    %PLOT THE QUATERNION ERROR ABOUT EACH AXIS
    load q_residuals.txt
    figure
    hold on
    grid on
    plot(q_residuals(1:end,1),'r.')
    plot(q_residuals(1:end,2),'b.')
    plot(q_residuals(1:end,3),'g.')
    plot(q_residuals(1:end,4),'m.')
    title('ERROR IN EACH COMPONENT OF THE UNIT-QUATERNION')
    xlabel('DISCRETE-TIME OBSERVATION NUMBER')
    ylabel('RESIDUAL')
    legend('q1','q2','q3','q4')
```

```
%PLOT THE ATTITUDE QUATERNION ERROR
load attitude_error.txt
figure
hold on
grid on
plot(attitude_error(1:end,1),'r.')
plot(attitude_error(1:end,2),'b.')
plot(attitude_error(1:end,3),'g.')
%plot(attitude_error(1:end,4),'m.')
title('ATTITUDE QUATERNION ERROR')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('ATTITUDE QUATERNION')
legend('q1','q2','q3')

%CALCULATE AND PLOT THE TOTAL EULER ANGLE ERROR
n=length(ea_residuals);
ea_total_error=zeros(n,1);
for k=1:n
    ea_total_error(k,1)=norm(ea_residuals(k,:));
end
figure
hold on
grid on
plot(ea_total_error,'m.')
title('TOTAL ERROR IN THE EULER ANGLES')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('TOTAL ATTITUDE ERROR (deg)')

%CALCULATE AND PLOT THE TOTAL QUATERNION ERROR
n=length(q_residuals);
q_total_error=zeros(n,1);
for k=1:n
    q_total_error(k,1)=norm(q_residuals(k,:));
end
figure
hold on
grid on
plot(q_total_error,'m.')
title('TOTAL ERROR IN THE UNIT-QUATERNION')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('TOTAL ATTITUDE ERROR')

%PLOT THE TIME NEEDED TO PROCESS EACH OBSERVATION
load elapsed_t.txt
figure
hold on
grid on
plot(elapsed_t(1:end,1),elapsed_t(1:end,2),'k.')
title('TIME NEEDED TO PROCESS EACH OBSERVATION')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('ELAPSED TIME (min)')

%PLOT THE STATE ERROR COVARIANCE MATRIX ELEMENTS
load covariance.txt
```

```
%PLOT THE QUATERNION ERROR COVARIANCE
figure
hold on
grid on
plot(covariance(1:end,1),'r.')
plot(covariance(1:end,2),'b.')
plot(covariance(1:end,3),'g.')
title('QUATERNION ERROR COVARIANCE')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('VARIANCE')
legend('q1','q2','q3')

%PLOT THE ANGULAR VELOCITY ERROR COVARIANCE
figure
hold on
grid on
plot(covariance(1:end,4),'r.')
plot(covariance(1:end,5),'b.')
plot(covariance(1:end,6),'g.')
title('ANGULAR VELOCITY ERROR COVARIANCE')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('VARIANCE')
legend('w1','w2','w3')

%PLOT THE MOMENT OF INERTIA PARAMETER ERROR COVARIANCE
figure
hold on
grid on
plot(covariance(1:end,7),'r.')
plot(covariance(1:end,8),'b.')
plot(covariance(1:end,9),'g.')
title('MOMENT OF INERTIA PARAMETER ERROR COVARIANCE')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('VARIANCE')
legend('I1','I2','I3')

%PLOT THE MOMENT OF INERTIA PARAMETER ERROR COVARIANCE
figure
hold on
grid on
plot(covariance(1:end,10),'r.')
plot(covariance(1:end,11),'b.')
plot(covariance(1:end,12),'g.')
title('MOMENT OF INERTIA PARAMETER ERROR COVARIANCE')
xlabel('DISCRETE-TIME OBSERVATION NUMBER')
ylabel('VARIANCE')
legend('I4','I5','I6')

%PLOT THE FRACTION ERROR NORM FOR THE ESTIMATED MOMENT OF INERTIA MATRIX
if TEST == 1
    %CALCULATE AND DISPLAY THE FINAL ESTIMATED MOMENT OF INERTIA
    I=x(7:12,m+1);
    C=[cos(I(5))*cos(I(6)),cos(I(5))*sin(I(6)),-sin(I(5));sin(I(4))
    *sin(I(5))*cos(I(6))-cos(I(4))*sin(I(6)),sin(I(4))*sin(I(5))
    *sin(I(6))+cos(I(4))*cos(I(6)),sin(I(4))*cos(I(5));cos(I(4))
    *sin(I(5))*cos(I(6))+sin(I(4))*sin(I(6)),cos(I(4))*sin(I(5))
```

```
        *sin(I(6))-sin(I(4))*cos(I(6)),cos(I(4))*cos(I(5))];
        MoI_est=C*diag([(I(2)^2+I(3)^2)/12,(I(1)^2+I(3)^2)/12,
        (I(1)^2+I(2)^2)/12])*C';
        disp('FINAL ESTIMATED MOMENT OF INERTIA')
        disp(MoI_est)
    elseif TEST == 2
        figure
        hold on
        grid on
        plot(MoI_error,'r.')
        title('FRACTIONAL ERROR NORM FOR THE ESTIMATED INERTIA MATRIX')
        xlabel('DISCRETE-TIME OBSERVATION NUMBER')
        ylabel('FRACTIONAL ERROR NORM')
    end
end

%CALCULATE THE REFINED SET OF MOTION PARAMETERS AND PROJECT THE FINAL STATE
BACKWARDS OVER THE TIME SPAN OF INTEREST
[qhat,ehat,vhat_ea,vhat_q,xhat,RT,RA,D,CA,ZI,XI,PP,SP,PA,SA,PR,SR] =
m_param(Ybar_ea,Ybar_q,T,x(:,end),q4_s(end),w_pn,DN,CoP,U_NORM,V_SAT,R_SUN,SO
LAR_I,A_DENSITY,B_FIELD,S_AREA,R_SAT,PATH);

%DISPLAY THE REFERENCE TIME (sec), RIGHT ASCENSION (deg), DECLINATION (deg),
CONING ANGLE (deg), SPIN ANGLE (deg) PRECESSION ANGLE (deg), PRECESSION
PERIOD (sec/rev), AND SPIN PERIOD (sec/rev)
disp('[REFERENCE TIME | RIGHT ASCENSION | DECLINATION | CONING ANGLE | SPIN
ANGLE | PRECESSION ANGLE | PRECESSION PERIOD | SPIN PERIOD]')
disp([RT,RA,D,CA,SA,PA,PP,SP])

%DISPLAY THE Z INITIAL (deg), X INITIAL (deg), PRECESSION RATE (deg/sec), AND
SPIN RATE (deg/sec)
disp('[Z-INITIAL | X-INITIAL | PRECESSION RATE | SPIN RATE]')
disp([ZI,XI,PR,SR])

%STOP THE ALGORITHM TIMER
STOP_TIME = clock;
TOTAL_TIME = etime(STOP_TIME,START_TIME);

%DISPLAY THE TOTAL ALGORITHM RUN TIME
disp('TOTAL RUN TIME (hrs)')
disp(TOTAL_TIME/3600)
disp('DONE')
```

11 References

1. **Markley, F.L.** Three-Axis Attitude Determination Methods. [book auth.] James Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

2. **Shuster, Malcolm D. and Dellinger, Wayne F.** Spacecraft Attitude Determination and Control. [book auth.] Vincent L. Pisacane. *Fundamentals of Space Systems Second Edition.* New York, NY : Oxford University Press, 2005, pp. 236-325.

3. **Vallado, David.** *Fundamentals of Astrodynamics and Applications, Third Edition.* Hawthorne, CA : Microcosm Press, 2007.

4. **Imaging and Assessment Technology Group.** *XELIAS Software User's Manual Version 2.0.* Lexington, MA : MIT Lincoln Laboratory, 1 May 2000.

5. **Kuipers, Jack B.** *Quaternions and Rotation Sequences.* Princeton, NJ : Princeton University Press, 1999.

6. **Diebel, James.** *Representing Attitude: Euler Agles, Unit Quaternions, and Rotation Vectors.* Stanford, CA : Stanford University, 20 October 2006.

7. **Goldstein, Herbert, Poole, Charles and and Safko, John.** *Classical Mechanics, Third Edition.* San Francisco, CA : Addison Wesley, 2002. pg. 134 - 208.

8. **Tandon, Gyanendra K.** Coordinate Transformation . [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

9. **Fallon, Lawrence.** Quaternions. [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Bostion, MA : D. Reidel Publishing Company, 1985.

10. **Markley, F.L.** Attitude Dynamics. [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

11. **Wertz, James R.** Introduction to Attitude Dynamics and Control. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

12. **Spence, C.B and Markley, F.L.** Attitude Prediction. [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

13. **Carnahan, Brice, Luther, H. and Wilkes, James.** *Applied Numerical Methods.* New York, NY : John Wiley & Sons, Inc, 1969.

14. **Hildebrand, F.B.** *Introduction to Numerical Analysis.* New York, NY : McGraw-Hill, Inc, 1956.

15. **Lambert, J.D.** *Computational Methods in Ordinary Differential Equations.* New York, NY : John Wiley & Sons, Inc., 1973.

16. *Comparing Numerical Methods for Ordinary Differential Equations.* **Hull, T., Enright, W. and Fellen, B. and Sedgwick, A.** s.l. : SIAM Journal of Numerical Analysis, 1972, Vol. 9.

17. *MATLAB 7.7.0 (R2008b).* [technical computing software] Natick, MA : The MathWorks, Inc., 2008. http://www.mathworks.com/.

18. **Strang, Gilbert.** *Calculus.* Wellesley, MA : Wellesley-Cambridge Press, 1991.

19. *Fast and Accurate Computation of Polyhedral Mass Properties.* **Mirtich, Brian.** 2, Wellesley, MA : Journal of Graphics Tools, 1996, Vol. 1.

20. **Aiello, John, et al.** Modeling the Earth and Space Environment. [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

21. **Hodge, Ward F.** *Effect of Environmental Torques on Short-Term Attitude Prediction for a Spacecraft in a Sun-synchronous Orbit.* Hampton, VA : NASA Langley Research Center, January 1972.

22. **The National Oceanic and Atmospheric Administration.** *International Geomagnetic Reference Field.* [10th Generation] s.l. : The International Association of Geomagnetism and Aeronomy, 2005. http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html.

23. **Aiello, John and Plett, Michael.** Spherical Harmonics and Magnitic Field Models. [book auth.] James R. Wertz. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

24. **Picone, Michael, Hedin, Alan and Drob, Doug.** *NRLMSIS-00 Empirical Atmosphere Model.* [Source Code] Washington D.C. : Naval Research Laboratory, 2001. http://uap-www.nrl.navy.mil/models_web/msis/msis_home.htm.

25. **Stowe, Lin.** Radar Imaging. Lexington, MA : MIT Lincoln Laboratory, Jan 2009.

26. *Range-Doppler Imaging of Rotating Object.* **Walker, Jack L.** 1, Lexington, MA : IEEE Trans. on Aerospace and Electronic Systems, Jan 1980, Vols. AES-16.

27. **Bradley, Andrew M.** *Radar Imaging and Motion Analysis.* Lexington, MA : MIT Lincoln Laboratory, July 2005.

28. *Developments in Radar Imaging.* **Ausherman, Dale, et al.** 4, s.l. : IEEE Transactions on Aerospace and Electronic Systems, July 1984, Vols. AES-20.

29. **Lennon, Chris, et al.** Introduction to Radar Systems. Lexington, MA : MIT Lincoln Laboratory, Jan 2007.

30. **Hannen, Paul J. and Toolmay, J.C.** *Radar Principles for the Non-secialist Third Edition.* Raleigh, NC : SciTech Publishing Inc, 2004.

31. **Magura, K. and Wynne-Jones, I.** *Object Angular Velocity Estimation from ISAR Data.* Wachtberg, Germany : EUSAR'04. Ulm, May 2004.

32. **Space Situational Awareness Group: Carey, M.; Hiett, M.; and Levine, R.** *LLMotion User's Guide.* Lexington, MA : MIT Lincoln Laboratory, 1 May 2000.

33. **Li, Anye.** *Constraining Rotation Rate to Cross-Range Extent.* Lexington, MA : MIT Lincoln Laboratory, 26 June 2008.

34. *Constraining Rotation to Range Extent.* Lexington, MA : MIT Lincoln Laboratory , 30 April 2009.

35. **Bhat, Kiran, Seitz, Steven and Popovic, Jovan, and Khosla, Pradeep.** *Computing the Physical Parameters of Rigid-Body Motion from Video* . Berlin, Germany : IEEE International Conference on Multimedia and Expo, 2002.

36. **Wertz, James R.** State Estimation Attitude Determination Methods. *Spacecraft Attitude Determination and Control.* Boston, MA : D. Reidel Publishing Company, 1985.

37. **Born, George H., Schutz, Bob E. and and Tapley, Byron D.** *Statisitical Orbit Determination.* Burlington, MA : Elsevier Academic Press, 2004.

38. **Devore, Jay L.** *Probability and Statistics for Engineering and the Sciences Sixth Edition.* Belmont, CA : Brooks/Cole - Thompson Learning, Inc., 2004.

39. *Maximum Likelihood Estimates of Linear Dynamic Systems.* **Rauch, H. E, Tung, F.and Striebel, C. T.** 8, s.l. : AIAA Journal, 1963, Vol. 3.

40. **Musoff, Howard and Zarchan, Paul.** *Fundamentals of Kalman Filtering: A Practical Approach Second Edition.* Reston, VA : AIAA Publications, 2005.

41. *Backward-Smoothing Extended Kalman Filter.* **Psiaki, Mark L.** September - October 2005, Journal of Guidance, Control, and Dynamics, Vol. 28, No. 5, pp. 885-894.

42. *Survey of Nonlinear Attitude Estimation Methods.* **Crassidis, John L., Markley, Landis F and and Cheng, Yang.** 1, s.l. : Journal of Guidance, Control, and Dynamics , Jan - Feb 2007, Vol. 30.

43. *The Treatment of Bias in the Square-Root Information Filter/Smoother.* **Bierman, Gerald J.** 1, s.l. : Journal of Optimization Theory and Applications, 1975, Vol. 16.

44. **Folcik, Zachary J.** *Orbit Determination Using Modern Filters/Smoothers and Continuous Thrust Modeling.* Cambridge, MA : Massachusetts Institute of Technology, M.S. Thesis, June 2008.

45. **Psiaki, Mark L.** *Square-Root Information Filtering and Fixed-Interval Smoothing with Singularities.* Philadelphia, PA : American Control Conference Paper No. FA05-5, 1998.

46. **Julier, S. J., Uhlmann, J. K.** *A New Extension of the Kalman Filter to Nonlinear Systems.* s.l. : International Symposium Aerospace/Defense Sensing, Simulation and Controls, 1997.

47. *The Unscented Kalman Filter for Nonlinear Estimation.* **Wan, E., van der Merwe, R.** s.l. : IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium AS-SPCC, 2000.

48. *Radars for the Detection and Tracking of Missiles, Satellites, and Planets.* **Stone, Melvin and Banner, Gerald.** No. 2, Lexington, MA : Lincoln Laboratory Journal, 2000, Vol. 12.

49. **Bramley, M, Moor, T and Morgan, T and Seeley, T.** *ARIES User's Guide Verson 2.0.* Lexington, MA : MIT Lincoln Laboratory, Dec 1995.

50. **Yuret, D.** *From Genetic Algorithms to Efficient Optimization* . Cambridge, MA : MIT Department of Electrical Engineering and Computer Science, M.S. Thesis, 1994.

51. **Dizhoor, Vladimir.** Spin-Precession Motion Parameters. Lexington, MA : MIT Lincoln Laboratory, Nov 2007.

52. *Space Physics Interactive Data Resource.* [F10.7 Dataset] Boulder, CO : National Geophysical Data Center, NOAA. http://spidr.ngdc.noaa.gov/spidr/home.do.

53. *Satellite Tool Kit Version 8.* [Software System] Exton, PA : Analytical Graphics, Inc., 2006.

54. **California Institute of Technology.** *Horizons Database.* [Web-Interface] Pasadena, CA : Jet Propulsion Laboratory, NASA, 2006. http://jpl.nasa.gov/horizons.cgi.

55. *NRLMSISE-00 Empirical Model of the Atmosphere: Statistical Comparison and Scientific Issues.* **Picone, J., Hedin, A. and Drob, D. and Aikin, A.** s.l. : Journal of Geophysical Research, December 2001. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20020038771_2002061046.pdf.

56. *Kalman Filtering for Spacecraft Attitude Estimation.* **Leffers, E.J. and Markley, F.L. and Shuster, M.D.** No.5, s.l. : Jounal of Guidance, Sept - Oct 1982, Vol. 5. AIAA 82-0070R.

57. *Space Telescope Pointing Control System.* **Dougherty, H., Tompetrinit, K., Levinthal, J. and Nurre, G.** No. 4, Danvers, MA : AIAA Guidance and Control Conference, July-August 1982, Vol. 5. AIAA 80-1784R.

58. **Baeggli, Hans, Brûlé, Luc and Smyth, Jill.** *RADARSAT-2 Program Update.* s.l. : IEEE, 2004.

59. *Synthetic Aperture Radar in Europe: ERS, Envisat,and Beyond.* **Attema, Evert, Desnos, Yves-Louis and Duchossois, Guy.** No. 1, s.l. : Johns Hopkins APL Technical Digest, 2000, Vol. 21.

60. *Perturbation Formulations for Satellite Attitude Dynamics.* **Kraige, L.G. and Junkins, J.L.** s.l. : Celestial Mechanics, Feb. 1976, Vol. 13.

61. **Willis, Nicholas J.** *Bistatic Radar Second Edition.* Raleigh, NC : SciTech Publishing,Inc, 2005.

62. *Shape and Motion from Image Streams under Orthography: a Factorization Method.* **Tomasi, Carlo and Kanade, Takeo.** No. 2, s.l. : International Journal of Computer Vision, Nov 1992, Vol. 9.

63. *A Sequential Factorization Method for Recovering Shape and Motion from Image Streams.* **Morita, T. and Kanade, T.** s.l. : ARPA Image Understanding Workshops, November 1994, Vol. 2.

64. *An Iterative Improvement of the Tomasi-Kanade Factorization.* **Hajder, Levente.** Budapest : Third Hungarian Conference on Computer Graphics and Geometry, 2005.

65. **Strang, Gilbert.** *Computational Science and Engineering.* Wellesley, MA : Wellesley-Cambridge Press, 2007.

66. **Eterno, John S.** Attitude Determination and Control. [book auth.] Wiley J Larson and James R. Wertz. *Space Mission Analysis and Design Third Edition.* El Segundo, CA : Microcosm Press, 1999, pp. 354-380.

67. **Eberly, David.** *Polyhedral Mass Properties.* s.l. : Geometric Tools, LLC, 31 December 2002.

68. **Sellers, Jerry j.** *Understanding Space: An Introduction to Astronautics, Revised Second Edition.* Boston : McGraw-Hill Companies, Inc, 2004.

69. **Bierman, Gerald J.** *Factorization Methods for Discrete Sequential Estimation.* New York, NY : Academic Press, 1977.

70. **Meditch, James S.** *Stochastic Optimal Linear Estimation and Control.* New York, NY : McGraw-Hill, 1969.

71. **De La Maza, D. and Yuret, D.** *Dynamic Hill Climbing.* s.l. : AI Expert, March 1994.

72. *The Unscented Kalman Filter for Nonlinear Estimation.* **Wan, E., van der Merwe, R.** s.l. : IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium AS-SPCC, 2000.

73. *A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators.* **Julier, S., Uhlmann, J., Durrant-Whyte, H. F.** 3, pp. 477-482, s.l. : IEEE Transactions on Automatic Control, 2000, Vols. AC-45.

74. *Derivation and Simulation Testing of a Sigma-Points Smoother.* **Psiaki, M.** 1, s.l. : Journal of Guidance, Control and Dynamics, Jan-Feb. 2007, Vol. 30.

75. *A New Extension of the Kalman Filter to Nonlinear Systems.* **Julier, S. J., Uhlmann, J. K.** s.l. : International Symposium Aerospace/Defense Sensing, Simulation and Controls, 1997.