

# **IN-SPACE AUTONOMOUS SPACECRAFT**

## **ALIGNMENT CALIBRATION**

by

**Kok-Lam Lai**

**September 1, 2003**

A thesis submitted to the  
Faculty of the Graduate School of  
State University of New York at Buffalo  
in partial fulfillment of the  
requirements for the degree of

**Master of Science**

Department of Mechanical & Aerospace Engineering  
State University of New York at Buffalo  
Buffalo, New York 14260

*To my parents*

# Acknowledgement

First of all, I would like to express my gratitude to my advisor, Prof. John L. Crassidis. His keen interest in the field with extensive knowledge and experience helped me greatly in accomplishing this research. I deeply appreciate his patience, guidance, support and being a good friend.

Thank you Prof. D. Joseph Mook for introducing me into this interesting field with your excellent teaching and thanks for being a committee member of my thesis. Prof. Tarunraj Singh, thank you for your interesting classes and enthusiasms in the field and also being my thesis committee member.

I would like to thank Mr. Richard Harman at National Aeronautics and Space Administration (NASA) - Goddard Space Flight Center for the research grant and attitude telemetry data.

None would be possible without the love and understanding of my parents Tak Yau and Lee Ah. Thank you for being my all time best friends.

My dears family members at 1017 Furnas especially Jong-Woo, Jongrae, Chii-Chuan, Randy, Andy and Matthias, thanks for providing me a second home, you guys have been great brothers. Jong-Woo, your advices will never be forgotten, especially the *backups* theory.

And finally, my friends and colleagues and also my high school friends, who I am indebted for the tremendous impact they had on my life.

# Contents

<b>Acknowledgement</b>	iii
<b>List of Figures</b>	vii
<b>List of Tables</b>	x
<b>Abstract</b>	xi
<b>1 Introduction</b>	1
1.1 Background . . . . .	1
1.2 Motivation . . . . .	2
1.3 Organization of This Thesis . . . . .	3
<b>2 Attitude Representation and Kinematics</b>	4
2.1 Introduction . . . . .	4
2.2 Background . . . . .	4
2.3 Euler Angles . . . . .	7
2.4 Eigenaxis Rotation . . . . .	10
2.5 Quaternion . . . . .	11
2.6 Generalized Rodrigues Parameters . . . . .	13

2.7	Attitude Kinematics with Quaternions . . . . .	14
2.8	Attitude Error Representation . . . . .	15
2.9	Summary . . . . .	16
<b>3</b>	<b>State Estimation</b>	<b>17</b>
3.1	Introduction . . . . .	17
3.2	Kalman Filter . . . . .	18
3.2.1	Discrete Kalman Filter . . . . .	18
3.3	Extended Kalman Filter . . . . .	23
3.3.1	Continuous-Discrete Extended Kalman Filter . . . . .	23
3.4	Unscented Filter . . . . .	27
3.5	Summary . . . . .	34
<b>4</b>	<b>Attitude Sensor Models with Errors</b>	<b>36</b>
4.1	Introduction . . . . .	36
4.2	Vector Attitude Sensor . . . . .	36
4.3	Gyro . . . . .	38
4.4	Summary . . . . .	43
<b>5</b>	<b>Attitude and Alignment Filter</b>	<b>44</b>
5.1	Introduction . . . . .	44
5.2	Attitude Estimation . . . . .	44
5.3	Wahba's Problem . . . . .	46
5.4	Calibration . . . . .	46
5.4.1	Unscented Alignment Calibration . . . . .	47
5.4.2	Extended Kalman Filter Alignment Calibration . . . . .	54
5.5	Performance Evaluation . . . . .	54

5.6	Summary	55
<b>6</b>	<b>Results</b>	<b>56</b>
6.1	Introduction	56
6.2	Simulated Calibration Results with Extended Kalman Filter and Unscented Filter	56
6.3	Calibration on Actual Attitude Telemetry Data Using Unscented Filter	69
6.4	Summary	78
<b>7</b>	<b>Conclusion and Future Work</b>	<b>80</b>
7.1	Conclusions	80
7.2	Future Work	80
<b>Bibliography</b>		<b>83</b>
<b>A</b>	<b>Matrix Inversion Lemma</b>	<b>88</b>
<b>B</b>	<b>Matlab Code</b>	<b>89</b>

# List of Figures

2.1	Attitude Transformation . . . . .	6
2.2	First Rotation about the “local” x-axis . . . . .	8
2.3	Second Rotation about the “local” y-axis . . . . .	8
2.4	Third Rotation about the “local” z-axis . . . . .	9
6.1	Spacecraft Rotational Maneuver for all Simulation Runs . . . . .	58
6.2	UF, $\Delta t = 0.2$ , Attitude Errors with $3\sigma$ Bounds . . . . .	59
6.3	UF, $\Delta t = 0.2$ , Gyros Bias Errors with $3\sigma$ Bounds . . . . .	59
6.4	UF, $\Delta t = 0.2$ , Gyros Nonorthogonal Misalignment Errors with $3\sigma$ Bounds .	60
6.5	UF, $\Delta t = 0.2$ , Gyros Symmetric Scale Factor Errors with $3\sigma$ Bounds . . .	60
6.6	UF, $\Delta t = 0.2$ , Star Sensor Misalignment Errors with $3\sigma$ Bounds . . . . .	61
6.7	UF, $\Delta t = 0.2$ , Payload Sensor Misalignment Errors with $3\sigma$ Bounds . . . .	61
6.8	EKF, $\Delta t = 2.0$ , Attitude Errors with $3\sigma$ Bounds . . . . .	62
6.9	EKF, $\Delta t = 2.0$ , Gyros Bias Errors with $3\sigma$ Bounds . . . . .	62
6.10	EKF, $\Delta t = 2.0$ , Gyros Nonorthogonal Misalignment Errors with $3\sigma$ Bounds	63
6.11	EKF, $\Delta t = 2.0$ , Gyros Symmetric Scale Factor Errors with $3\sigma$ Bounds . .	63
6.12	EKF, $\Delta t = 2.0$ , Star Sensor Misalignment Errors with $3\sigma$ Bounds . . . . .	64
6.13	UF, $\Delta t = 2.0$ , Attitude Errors with $3\sigma$ Bounds . . . . .	64

6.14 UF, $\Delta t = 2.0$ , Gyros Bias Errors with $3\sigma$ Bounds . . . . .	65
6.15 UF, $\Delta t = 2.0$ , Gyros Misalignment Errors with $3\sigma$ Bounds . . . . .	65
6.16 UF, $\Delta t = 2.0$ , Gyros Symmetric Scale Factor Errors with $3\sigma$ Bounds . . . . .	66
6.17 UF, $\Delta t = 2.0$ , Star Sensor Misalignment Error with $3\sigma$ Bounds . . . . .	66
6.18 UF, $\Delta t = 2.0$ , Payload Sensor Misalignment Error with $3\sigma$ Bounds . . . . .	67
6.19 CPU Time of EKF Simulation Runs . . . . .	69
6.20 Artist Rendering of WMAP Leaving the Earth/Moon Toward the L2 Halo Orbit . . . . .	70
6.21 WMAP Gyros Rotational Rate Measurements . . . . .	71
6.22 WMAP Estimated Attitude in Quaternion . . . . .	71
6.23 WMAP Estimated Gyros Bias . . . . .	72
6.24 WMAP Estimated Gyros Symmetric Scale Factor Error . . . . .	72
6.25 WMAP Estimated Gyros Nonorthogonal Misalignments . . . . .	73
6.26 WMAP Estimated Star Sensor 1 Misalignments . . . . .	73
6.27 WMAP Estimated Star Sensor 2 Misalignments . . . . .	74
6.28 WMAP Attitude Estimation $3\sigma$ Bounds . . . . .	74
6.29 WMAP Gyros Biases Estimation $3\sigma$ Bounds . . . . .	75
6.30 WMAP Gyros Nonorthogonal Misalignments Estimation Covariance $3\sigma$ Bounds	75
6.31 WMAP Gyros Symmetric Scale Factor Errors Estimation $3\sigma$ Bounds . . . . .	76
6.32 WMAP Star Sensor 1 Misalignments Estimation $3\sigma$ Bounds . . . . .	76
6.33 WMAP Star Sensor 2 Misalignments Estimation $3\sigma$ Bounds . . . . .	77
6.34 WMAP Star Sensor 1 Norm Residual Between Estimated Observation and Sensor Measurement . . . . .	77
6.35 WMAP Star Sensor 2 Norm Residual Between Estimated Observation and Sensor Measurement . . . . .	78

7.1 Norm Residual Between Body Estimates and Reference Vectors . . . . .	81
--	----

# List of Tables

3.1	Discrete Kalman Filter . . . . .	23
3.2	Continuous-Discrete Extended Kalman Filter . . . . .	28
3.3	Discrete Extended Kalman Filter . . . . .	29
3.4	Unscented Filter . . . . .	33
6.1	Filter Interval and Completion Time . . . . .	68

# **Abstract**

The main objective of this work is to develop an in-space algorithm for autonomous sensor calibration. A new spacecraft sensor alignment estimation approach based on the Unscented filter is derived. Basic six-state attitude estimation has been widely implemented in actual spacecraft missions. However, more stringent pointing accuracy of modern spacecraft requires sensor alignment calibration to be performed post-launch to accommodate in-space disturbances and launch shock vibration. Sequential filtering is preferred since the calibration parameters could drift over time. It also minimizes ground crew intervention and mission disruption. The Unscented filter gives a higher-order linearization than the extended Kalman filter and offers a more statistically correct treatment of the mean and covariance. This leads to better estimation accuracy. Simulated spacecraft data results show that the Unscented filter is more robust and more suitable for on-board implementation than the traditional extended Kalman filter. Also, experimental results from the Wilkinson Microwave Anisotropy Probe spacecraft using the Unscented filter are shown to test the performance of the algorithm using real attitude telemetry data.

# 1 Introduction

## 1.1 Background

Precise attitude estimation is crucial to most spacecraft missions today. The traditional six-state extended Kalman filter (EKF) estimates the current attitude and gyro biases simultaneously [1]. This filter assumes unvarying alignment of the sensors involved in attitude estimation. However, sensor misalignment is inevitable and would contribute to unreliable attitude estimates. More stringent attitude pointing accuracy requires misalignments to be estimated and implemented into the attitude estimator. It has been noted in some recent papers [2]–[4] of the importance of proper calibration for use in fault detection or rate derivation. Without in-space sequential calibration, attitude sensors have to be recalibrated by ground crew from time to time to improve residual characteristics [3]. Misalignment calibration is usually performed prior to launch. However, launch shock often makes this pre-launch calibration somewhat irrelevant and requires post-launch alignment calibration before normal mission mode. Thus in-flight alignment calibration is needed to accommodate these unanticipated changes in alignment. The ill effects of attitude estimation using misaligned sensors are shown in Ref. [5].

Onboard real-time or sequential filtering is preferred since misalignment parameters could drift, either due to in-space disturbances (for example solar wind, aerodynamics drag, persistence thermal shock) or unattended changes in the attitude sensor’s relative orientation,

especially those with moving mechanisms. A sequential filter constantly outputs the best estimate of the calibration parameters and can tolerate parameter drifts in real-time with minimal ground crew intervention or mission disruption. In this work, we propose a new real-time misalignment estimator using the Unscented filter (UF) [6]. This thesis addresses alignment estimation with respect to the gyros, thus achieving absolute alignment calibration [7].

This in-space calibration scheme would enhance the degree of autonomy of the Attitude Determination and Control System (ADCS) to reduce ground crew intervention for an extended period of time. Specifically, autonomy refers to a mode of ground system operation in which manual human actions are not required to accomplish desired functions. Autonomy refers to self-acting, self-regulating systems on the spacecraft wherein functions are delegated to the spacecraft systems. Because of operational necessity, budget constraints, and technology push a number of automation initiatives are underway at NASA-Goddard Space Flight Center (GSFC) [8].

## 1.2 Motivation

During the initial period of Tropical Rainfall Measurement Mission (TRMM) spacecraft mission, it was found that the Digital Sun Sensor (DSS) heads were misaligned by as much as 0.3 degrees. This caused biased DSS residuals, which ultimately leads to inaccurate attitude “solutions”. The significance of Inertial Reference Unit (IRU) calibration was felt too as the effect together with the misaligned DSS showed up in the biased estimates [9].

The reaction wheel assembly (RWA) calibration improved slewing performance of the Extreme Ultra-violet Explorer (EUVE) mission. It is noted in Ref. [10] that radiation, extreme thermal environment as well as electronics age typical in space flight could cause the RWA scale factors to change as much as 10% over the life of a typical mission. Also

from this reference the RWA scale factor is a source of slew error and most subject to drift after initial post launch calibration. Thus, a real-time sequential calibration scheme is best suited for RWA.

The gyro scale factor calibration has been traditionally and successfully implemented in various missions. However, during the Wilkinson Microwave Anisotropy Probe (WMAP) mission, due to high thermal variations, it is necessary to model the scale factors as temperature dependent. Reference [11] details this calibration effort and showed encouraging results.

The examples above show the importance that accurate calibration is paramount to higher instrument precision. In this thesis, we investigate the calibration of vector attitude sensors and rate gyros. A UF-based calibration algorithm will be developed and compared with the EKF results.

### **1.3 Organization of This Thesis**

The thesis organization is as follows. First, the attitude kinematics is briefly reviewed, and attitude sensor models with errors are derived. In the sequel, estimation with Kalman filter (KF) and UF are presented. The next chapter applies these filters to the calibration of attitude sensors. Next, a performance evaluation of the UF alignment calibration will be carried out via simulation with comparison to the EKF alignment calibration. Testing using real gyro measurements from the WMAP spacecraft will be demonstrated too. Lastly conclusions will be drawn and recommendations of future work will be given.

# 2 Attitude Representation and Kinematics

## 2.1 Introduction

This chapter reviews rotational attitude kinematics specifically suited for spacecraft applications. First, some background on the field is presented. Then some basic/essential material on attitude parametrizations is shown. Euler angles are shown to be one of the most intuitive attitude representation. Single axis rotation to represent an attitude transformation follows as a prequel to the quaternion representation that is favorable for on board implementation for its singular-free kinematics. A brief review on the generalized Rodrigues parameters (GRPs) will be presented since these parameters are will be used interchangeably with the quaternion in attitude error representation in filter to be developed in Chapter 5. Lastly, the quaternion kinematics equation will be given.

## 2.2 Background

The attitude of an object is defined as its orientation with respect to a usually fixed, non-moving *reference frame*. A *coordinate frame* is a set of three mutually perpendicular unit (normalized) vectors. The reference frame is also referred as the inertial frame. The coordinate system that's fixed on an object is usually called the *body frame*. Obviously, there is no non-moving object in the universe as everything is either rotating with respect to another celestial body (i.e. a star, a planet or mass center of a galaxy) or translating

farther from one another due to the expansion of the universe. In our context, an inertial frame could be a frame that has insignificant attitude changes with respect to the object of interest, say, a space vehicle.

In our applications, as with most dynamical applications, these coordinate systems have orthogonal unit vectors that follow the *right-hand rule*. The transformation matrix from an inertial frame to a body frame is called the *attitude matrix* (conventionally designated as the “ $A$ ” matrix). It is also referred as the *direction cosine matrix* (DCM) since a common way of constructing the attitude matrix is to take the cosine of the angles of the  $x$ ,  $y$  or  $z$  components of two frames. The attitude matrix is mathematically given by

$$\mathbf{b} = A\mathbf{r} \quad (2.1)$$

where  $\mathbf{b} = [b_x \ b_y \ b_z]^T$  is the body frame vector and  $\mathbf{r} = [r_x \ r_y \ r_z]^T$  is the reference/inertial frame vector. Note that the subscript  $x$ ,  $y$  and  $z$  are of the orthogonal axes of their respective frames. In a less confusing or more elaborate way, we would use

$$\mathbf{b} = A_{br}\mathbf{r} \quad (2.2)$$

Here we placed the appropriate letter closest to the attitude matrix. The first subscript corresponds to the frame whose base vector components are in the rows of  $A$  and the second subscript corresponds to the frame whose base vector components are in the column of  $A$ , i.e.

$$A_{br} = \begin{bmatrix} \mathbf{b}_{1r}^T \\ \mathbf{b}_{2r}^T \\ \mathbf{b}_{3r}^T \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{1b} & \mathbf{r}_{2b} & \mathbf{r}_{3b} \end{bmatrix} \quad (2.3)$$

This also corresponds to the dot product of the frames [12], i.e.,  $A_{br} = \mathbf{b} \cdot \mathbf{r}^T$ .

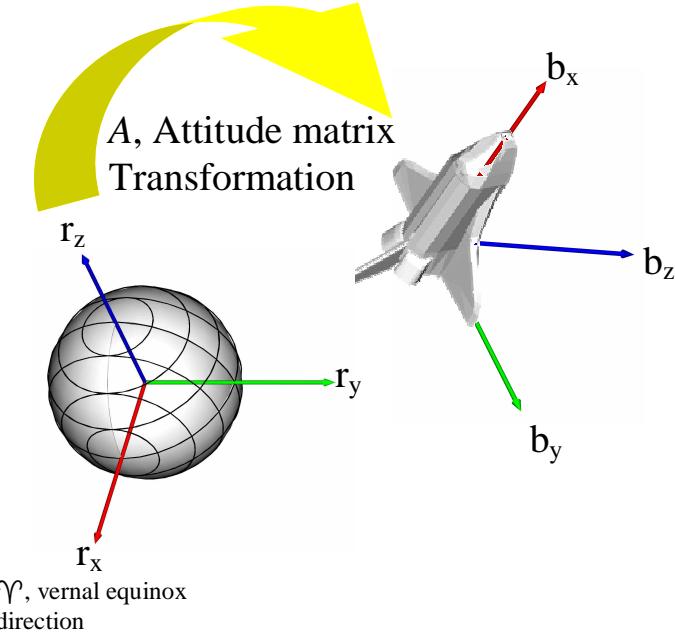


Figure 2.1: Attitude Transformation

Reference [13] showed that if the attitude matrix follows right hand rule (or *dextral reference frame*), its determinant is always +1. Then this matrix is also considered a *proper matrix*. Since the inverse of an attitude matrix is given by its transpose, it is also an *orthogonal matrix*.

The cumulative effect of two linear vectors is simply the addition of both, but linear addition of two attitude matrices does not yield the expected effect. The cumulative effect of two rotation matrices is given by the subsequent matrix multiplication. Let's say  $A_{\beta\alpha}$  rotates from the  $\alpha$ -frame to  $\beta$ -frame and  $A_{\gamma\beta}$  rotates from the  $\beta$ -frame to  $\gamma$ -frame. The direct transformation from  $\alpha$ - to  $\gamma$ -frame is simply  $A_{\gamma\alpha} = A_{\gamma\beta}A_{\beta\alpha}$ . Also notice that the sequence of the multiplication is reversed that the earlier rotation multiplies at the right hand side of the latter rotation.

There are many attitude parametrizations available, for example, Euler angles, quater-

nion, Cayley-Klein parameters, Gibb's vector, modified Rodrigues parameters (MRPs), etc. Particular attention will be stressed on Euler angles, quaternion and GRPs. Euler angles provide excellent physical intuition, however, it is the quaternion and the Rodrigues parameters that are more suited for our filter implementation.

## 2.3 Euler Angles

The physically intuitive Euler angles are especially appealing in representing attitude. Traditionally, roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) are used to denote the first, second and third rotational angles with respects to orthonormal axes from the reference frame. Note that the rotational sequences need not always start from  $x$ ,  $y$  then  $z$ -axis, i.e., there are numerous sequences possible. Let's derive the attitude matrix for the 1-2-3 Euler sequence which is common in spacecraft applications. The rotations are shown in Figures 2.2-2.4. First, we perform rotation about the first axis ( $x$ ) of reference frame  $\mathbf{r}$  by  $\phi$

$\mathbf{r}$

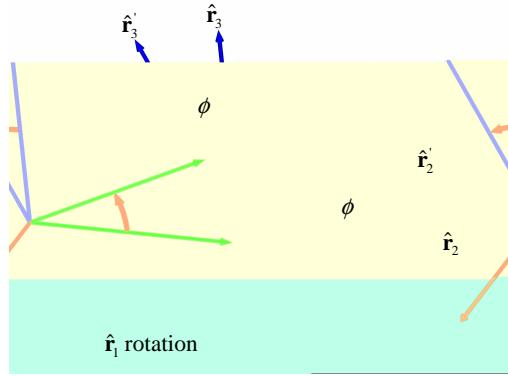


Figure 2.2: First Rotation about the “local” x-axis

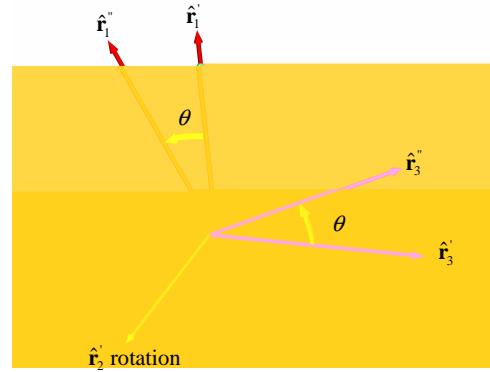


Figure 2.3: Second Rotation about the “local” y-axis

Finally rotate about the third axis ( $z''$ -axis or the  $z$ -axis after the second rotation) of  $\mathbf{r}''$  by  $\psi$  to the body frame  $\mathbf{b}$

$$\mathbf{b} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}'' \quad (2.6)$$

Putting the previous three equations together obtains the attitude transformation matrix that maps from the reference frame to the body frame. From Eq. (2.1) it is obvious from

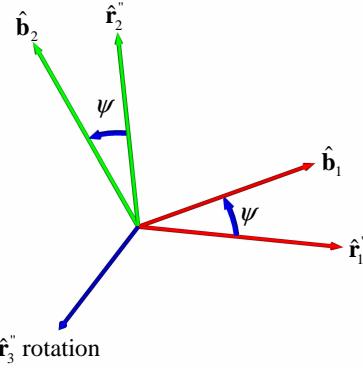


Figure 2.4: Third Rotation about the “local” z-axis

here that

$$A = \begin{bmatrix} c\psi c\theta & s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi - c\psi s\theta c\phi \\ -s\psi c\theta & c\psi c\phi - s\psi s\theta s\phi & c\psi s\phi + s\psi s\theta c\phi \\ s\theta & -c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.7)$$

where  $c\alpha \equiv \cos \alpha$  and  $s\alpha \equiv \sin \alpha$ . Or more generally,

$$A = A_r(\psi)A_q(\theta)A_p(\phi) \quad (2.8)$$

Where the subscripts  $p$ ,  $q$  and  $r$  denote the axis of rotations and as alluded earlier, need not always correspond to the first, second, then third axis (1-2-3 rotation). Since there are  $3 \times 2 \times 2$  possibilities of axes of rotation, there are totally twelve Euler representations. For more details, refer to Ref. [14]. Since a three-dimensional property is represented by a nine-parameter  $3 \times 3$  attitude matrix, there are six constraints required by the orthogonality constraint. An interesting note is that when taking the limit of these angles to zero (small

angle approximation for the sines and cosines), they all approach

$$A \simeq \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} = I_{3 \times 3} - [\alpha \times] \quad (2.9)$$

where  $\alpha = [\phi \ \theta \ \psi]^T$  and  $[\alpha \times]$  is the antisymmetric cross-product matrix with  $\alpha \times \beta = [\alpha \times] \beta$  and

$$[\alpha \times] \equiv \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix} \quad (2.10)$$

## 2.4 Eigenaxis Rotation

**Euler's Theorem:** For every coordinate frame transformation, there is always a *principal angle*,  $\phi$ , that rotates along a *principal axis*,  $\hat{\mathbf{e}}$ , to bring a frame to another.

Leonhard Euler (1707-1783) found that every transformation from one frame to another can always be done with a single axis rotation. This principal axis is also called the *Euler Axis* or *eigenaxis*; the principal angle is sometimes referred as the *Euler angle*. The principal axis has the same components in both frames, i.e.,

$$A\hat{\mathbf{e}} = \hat{\mathbf{e}} \quad (2.11)$$

Thus a subscript to specify the related frames is not needed for the principal axis. Furthermore, this implies that  $\hat{\mathbf{e}}$  is an eigenvector of  $A$  with eigenvalue of 1 and thus the name *eigenaxis*.

The attitude matrix can be obtained directly from the principal axis and the principal angle. It is given by

$$A = (\cos \phi) I_{3 \times 3} + (1 - \cos \phi) \hat{\mathbf{e}} \hat{\mathbf{e}}^T - \sin \phi [\hat{\mathbf{e}} \times] \quad (2.12)$$

From an attitude matrix, we can compute its principal axis and principal angle by

$$\phi = \cos^{-1} \left[ \frac{1}{2} (Tr(A) - 1) \right] \quad (2.13a)$$

$$[\hat{\mathbf{e}} \times] = \frac{1}{2 \sin \phi} (A^T - A) \quad (2.13b)$$

where  $Tr\{\cdot\}$  is the matrix *trace* operator.

## 2.5 Quaternion

Sir William Rowan Hamilton (1805-1865) proposed a uniformly and universally determined (singular-free), four-parameter attitude representation in 1886 which he called the quaternion. The individual quaternion components are also called the Euler parameters. However, it was not until recently that the quaternion sparked a wide spread interest in its implementation and lead to substantial research.

The quaternion is defined as

$$\mathbf{q} \equiv \begin{bmatrix} \boldsymbol{\varrho} \\ q_4 \end{bmatrix} \quad (2.14)$$

where  $\boldsymbol{\varrho} \equiv [q_1 \ q_2 \ q_3]^T = \hat{\mathbf{e}} \sin(\phi/2)$  is the principal axis or vector components and  $q_4 = \cos(\phi/2)$  is the scalar component of a quaternion. Since this is a three-dimensional attitude representation, the four-parameter quaternion components cannot be all independent to each other and a single constraint must exist to reduce its degree-of-freedom by one. The

superfluous degree-of-freedom is reduced with the constraint  $\mathbf{q}^T \mathbf{q} = 1$ , which is equivalent to the orthogonal constraint in an attitude matrix,  $AA^T = 1$ . This is also analogous to  $\hat{\mathbf{e}}$  being a unit vector. An attitude matrix could be easily constructed from the quaternion with

$$A(\mathbf{q}) = \Xi^T(\mathbf{q})\Psi(\mathbf{q}) \quad (2.15)$$

with

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} + [\boldsymbol{\varrho} \times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \quad (2.16a)$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} - [\boldsymbol{\varrho} \times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \quad (2.16b)$$

Computing  $A(\mathbf{q})A^T(\mathbf{q}) = [\Xi^T(\mathbf{q})\Psi(\mathbf{q})][\Xi^T(\mathbf{q})\Psi(\mathbf{q})]^T = I_{3 \times 3}$  from Eq. (2.15) would lead to the conclusion that the quaternion constraint  $\mathbf{q}^T \mathbf{q} = 1$  must hold true. For quaternion multiplication we employ the convention given by Refs. [1] and [15] who multiply the subsequent rotation in the same sequence as the attitude matrix, i.e.  $A(\mathbf{q}')A(\mathbf{q}) = A(\mathbf{q}' \otimes \mathbf{q})$ . This is different from the historical multiplication convention by Refs. [16], [17] and [18]. The quaternion multiplication,  $\otimes$ , is defined with [19] [20]

$$\mathbf{q}' \otimes \mathbf{q} \equiv \begin{bmatrix} \Xi(\mathbf{q}) & \mathbf{q} \end{bmatrix} \mathbf{q}' = \begin{bmatrix} q'_4 \boldsymbol{\varrho} + q_4 \boldsymbol{\varrho}' + [\boldsymbol{\varrho} \times] \boldsymbol{\varrho}' \\ q'_4 q_4 - \boldsymbol{\varrho}'^T \boldsymbol{\varrho} \end{bmatrix} \quad (2.17)$$

To compute quaternion from an attitude matrix, we can use

$$q_4 = \pm \frac{1}{2} \sqrt{1 + Tr(A)} \quad (2.18a)$$

$$\boldsymbol{\varrho} = \frac{1}{4q_4} \begin{bmatrix} A_{23} - A_{32} \\ A_{31} - A_{13} \\ A_{12} - A_{32} \end{bmatrix} \quad (2.18b)$$

The quaternion does not provide an intuitive feeling into the attitude of an object, however, it is more suited for attitude estimation and control since it is singular-free. Besides, the quaternion formulation is also free from computationally expensive transcendental functions. Another interesting note from small angle approximation is that when Eq. (2.9) is used in Eq. (2.18), the vector part of the quaternion converges to half angles (in radians),  $\boldsymbol{\varrho} \simeq \alpha/2$ , and the scalar part approaches unity,  $q_4 \simeq 1$ . This facilitates the use of quaternion in EKF and is demonstrated in Ref. [1].

## 2.6 Generalized Rodrigues Parameters

This section briefly introduces the GRPs. These parameters gives a three-parameter attitude representation. Reference [21] shows that all three-parameter or minimal attitude parameterizations have a singularity or discontinuity somewhere; the GRPs is no exception. These parameters are used in our application to avoid some complications with direct implementation of quaternion in the UF formulation, shown later in Chapter 5. We will use the GRPs for the attitude error representation and quaternions for attitude propagation in our filter implementation. Consequently, only the conversion between quaternions and GRPs is given here.

The GRP, denotes  $\mathbf{p}$ , is obtained from the quaternion by

$$\mathbf{p} \equiv f \frac{\boldsymbol{\varrho}}{a + q_4} \quad (2.19)$$

where  $0 < a < 1$  and  $f$  is a scale factor. Note when  $a = 0$  and  $f = 1$  then Eq. (2.19) gives the Gibbs vector or classical Rodrigues parameters; and when  $a = f = 1$  then Eq. (2.19) gives the standard vector of the MRPs [15]. The MRP formulation has a singularity at  $360^\circ$ , however, in our application, since it will be used to represent the attitude error in the UF formulation, this singularity poses no difficulty.

The conversion from the GRPs to the quaternion is not as straight-forward however. It is given by

$$q_4 = \frac{-a\|\mathbf{p}\|^2 + f\sqrt{f^2 + (1 - a^2)\|\mathbf{p}\|^2}}{f^2 + \|\mathbf{p}\|^2} \quad (2.20a)$$

$$\boldsymbol{\varrho} = f^{-1}(a + q_4)\mathbf{p} \quad (2.20b)$$

## 2.7 Attitude Kinematics with Quaternions

As alluded earlier, singular and transcendental functions free making quaternion ideal for filter implementation. The quaternion kinematics is given by

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\Xi[\mathbf{q}(t)]\boldsymbol{\omega}(t) = \frac{1}{2}\Omega(\boldsymbol{\omega}(t))\mathbf{q}(t) \quad (2.21)$$

where  $\boldsymbol{\omega}$  is the  $3 \times 1$  angular velocity vector and

$$\Omega(\boldsymbol{\omega}) \equiv \begin{bmatrix} -[\boldsymbol{\omega} \times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \quad (2.22)$$

Another advantage in quaternion formulation is its kinematics is bilinear with rotational rate. For more interesting and useful quaternion identities, refer to Ref. [22].

## 2.8 Attitude Error Representation

Unlike linear measurements where the error, difference or distance between two measurements is easily defined, rotational motion does not share such luxury; “distance” has no meaning in rotational coordinates. Thus we need a mechanism that allows us to define the *deviation* between two coordinate frames. We would denote such a difference as the *attitude error*.

The local attitude error in quaternion is denoted as  $\delta \mathbf{q} \equiv [\delta \boldsymbol{\varrho}^T \ \delta q_4]^T$  with

$$\delta \mathbf{q} = \mathbf{q}_\alpha \otimes \mathbf{q}_\beta^{-1} \quad (2.23)$$

where  $\mathbf{q}^{-1}$ , is the quaternion inverse given by  $\mathbf{q}^{-1} = [-\boldsymbol{\varrho} \ q_4]^T$ . Note that  $\mathbf{q} \otimes \mathbf{q}^{-1} = [0 \ 0 \ 0 \ 1]^T$ .

In the attitude error representation, Eq. (2.19) thus becomes

$$\delta \mathbf{p} \equiv f \frac{\delta \boldsymbol{\varrho}}{a + \delta q_4} \quad (2.24)$$

and Eq. (2.20) becomes

$$\delta q_4 = \frac{-a \|\delta \mathbf{p}\|^2 + f \sqrt{f^2 + (1 - a^2) \|\delta \mathbf{p}\|^2}}{f^2 + \|\delta \mathbf{p}\|^2} \quad (2.25a)$$

$$\delta \boldsymbol{\varrho} = f^{-1}(a + \delta q_4) \delta \mathbf{p} \quad (2.25b)$$

Since for small errors  $\delta \boldsymbol{\varrho} \simeq \alpha/2$  and  $\delta q_4 \simeq 1$ , the attitude part of the error covariance is approximated by to the attitude estimation error for any rotational sequence, since the

small angle approximation of any attitude matrix leads to Eq. (2.9). Thus, for small angles, the Gibbs vector linearize to half angles and the MRP linearize to quarter angles. For slight convenience in this conversion, we would choose  $f = 2(a + 1)$  so that  $\|\delta\mathbf{p}\|$  is equal to  $\phi$  for small angles.

## 2.9 Summary

There are many other ways to parameterize an attitude/orientation. In this chapter, four attitude parametrizations were briefly discussed: Euler angles, eigenaxis rotation, quaternions and GRPs. Thus, an attitude matrix can be computed from Eqs. (2.8), (2.12) and (2.15). The GRP was briefly described in the last section to overcome an additive complication with an error quaternion representation that later appears in Chapter 5. Quaternion kinematics is chosen in the filter implementation for its singular-free and free from the computationally burdensome transcendental functions formulation. For the interested readers, Kuipers [18] provides an extensive insight into the quaternion.

# 3 State Estimation

## 3.1 Introduction

This chapter briefly discusses various state estimators. Estimation is defined as extraction of useful information from single/multiple noisy sensor measurements with reference to an assumed mathematical plant model. These filters can be derived in various ways but only one of each will be presented here. We begin with the celebrated Kalman filter (KF). After the derivation of the KF in 1960 [23] Apollo scientists quickly realized its potential. However, most of the real world dynamics are essentially nonlinear in nature, including the highly nonlinear attitude and orbital dynamics. Thus a nonlinear version of KF was developed shortly after. We also present a new approach called the Unscented filter. The Unscented filter promises more accurate estimates with a lower expected error covariance than the extended Kalman filter (EKF). It linearizes to higher order than the standard EKF.

State estimation is needed to “filter” out or to minimize the effect of sensor noise. It is the fusion of various noisy measurement data to obtain (estimate) information about the system states with the help from an assumed system model. Also, as in most of the time, the sensor does not measure all the state components or does not measure the state components directly. Thus an estimator is needed to obtain the missing information that is often important from control point of view.

If the system and sensor/measurement models are “valid” (high correlation between the model and the “truth”), with adequate observability and appropriate assumptions (for example the dominance of first order series expansion to the EKF), the filter should converge towards the true states with decreasing estimated error covariance. However, there is a limit to this filter until a balance is reached between gaining new information from new measurements and loss of information from propagation.

## 3.2 Kalman Filter

### 3.2.1 Discrete Kalman Filter

The truth model is given by

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k \quad (3.1a)$$

$$\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (3.1b)$$

where  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are zero-mean Gaussian white-noise (with no memory, uncorrelated, with the past and future noise) with covariances  $Q_k$  and  $R_k$  respectively,

$$E\{\mathbf{w}_k\} = \mathbf{0} \quad E\{\mathbf{w}_j \mathbf{w}_k^T\} = \delta_{jk} Q_k \quad (3.2a)$$

$$E\{\mathbf{v}_k\} = \mathbf{0} \quad E\{\mathbf{v}_j \mathbf{v}_k^T\} = \delta_{jk} R_k \quad (3.2b)$$

where  $E\{\cdot\}$  denotes the expectation operator and  $\delta_{jk}$  is the Kronecker function

$$\delta_{jk} = \begin{cases} 0 & \text{when } j \neq k \\ 1 & \text{when } j = k \end{cases} \quad (3.3)$$

The cross-correlation covariance is assumed to follow

$$E\{\mathbf{w}_j \mathbf{v}_k^T\} = \mathbf{0} \quad \forall j, k \quad (3.4)$$

Thus, they are uncorrelated with each other. The vector  $\mathbf{w}_k$  is also known as the process noise, it indicates how accurately we know the assumed model or how much the model changes with time. Unfortunately, this usually is not known precisely and some “tuning” effort is required to obtain the desired performance. The vector  $\mathbf{v}_k$  is the measurement noise. It indicates the accuracy of measurements from the sensor. This is usually easily determined or specified by the sensor manufacturer. Let the new state estimate be a linear combination of the propagated values and the sensors measurements:

$$\hat{\mathbf{x}}_k^+ = K'_k \hat{\mathbf{x}}_k^- + K_k \tilde{\mathbf{y}}_k \quad (3.5)$$

where the superscript  $-$  indicates propagated/pre-update value and  $+$  indicates updated value. Define the error states as

$$\tilde{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{x}_k \quad (3.6a)$$

$$\tilde{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k \quad (3.6b)$$

Note that a tilde sign over  $\mathbf{y}$  denotes measurement but a tilde over  $\mathbf{x}$  denotes error. From Eq. (3.6b) and with the substitution of Eqs. (3.5), (3.1b) and (3.6a) for  $\hat{\mathbf{x}}_k^+$ , we have

$$\tilde{\mathbf{x}}_k^+ = (K'_k + K_k H_k - I)\mathbf{x}_k + K_k \mathbf{v}_k + K'_k \tilde{\mathbf{x}}_k^- \quad (3.7)$$

Taking the expectation of the above equation and setting the expectation of the error states to zero for an unbiased estimator (note that  $\mathbf{x}_k$  is not a random variable) leads to

$$K'_k + K_k H_k - I = 0 \text{ or}$$

$$K'_k = I - K_k H_k \quad (3.8)$$

Substituting this result into Eq. (3.5) gives

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-) \quad (3.9)$$

Substituting Eqs. (3.9) and (3.1b) into Eq. (3.6b) gives

$$\begin{aligned} \tilde{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + K_k(H_k \mathbf{x}_k + \mathbf{v}_k - H_k \hat{\mathbf{x}}_k^-) - \mathbf{x}_k \\ &= (I - K_k H_k) \hat{\mathbf{x}}_k^- - (I - K_k H_k) \mathbf{x}_k + K_k \mathbf{v}_k \\ &= (I - K_k H_k) \tilde{\mathbf{x}}_k^- + K_k \mathbf{v}_k \end{aligned} \quad (3.10)$$

Now let's divert our attention to the error covariance. The state propagation equation is simply given by taking the expectation of Eq. (3.1a):

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k \quad (3.11)$$

Substituting Eqs. (3.11) and (3.1a) into the one time-step ahead of Eq. (3.6a) gives

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1}^- &= \Phi_k \hat{\mathbf{x}}_k^+ - \Phi_k \mathbf{x}_k - \Gamma_k \mathbf{w}_k \\ &= \Phi_k \tilde{\mathbf{x}}_k^+ - \Gamma_k \mathbf{w}_k \end{aligned} \quad (3.12)$$

Note that Eq. (3.12) is not a function of the control input  $\mathbf{u}_k$ , it is assumed to be a known or a deterministic property instead of a random variable. Now define the pre-update and

updated error covariances:

$$P_k^- \equiv E\{\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{-T}\} \quad (3.13a)$$

$$P_k^+ \equiv E\{\tilde{\mathbf{x}}_k^+ \tilde{\mathbf{x}}_k^{+T}\} \quad (3.13b)$$

Substituting Eq. (3.12) into the one time-step ahead of Eq. (3.13a) and assuming the statistical independence of  $\mathbf{w}_k$  and  $\tilde{\mathbf{x}}_k^+$ ,  $E\{\mathbf{w}_k \tilde{\mathbf{x}}_k^{+T}\} = E\{\tilde{\mathbf{x}}_k^+ \mathbf{w}_k^T\} = 0$  (this is due to the fact that  $\mathbf{w}_k$  is correlated with  $\tilde{\mathbf{x}}_{k+1}$  instead of  $\tilde{\mathbf{x}}_k$  as obvious from Eq. (3.1a)) leads to

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T \quad (3.14)$$

We then apply Eq. (3.10) into Eq. (3.13b) and again with statistical independence assumption between  $\mathbf{v}_k$  and  $\tilde{\mathbf{x}}_k^-$ ,  $E\{\mathbf{v}_k \tilde{\mathbf{x}}_k^{-T}\} = E\{\tilde{\mathbf{x}}_k^- \mathbf{v}_k^T\} = 0$  (as obvious from Eq. (3.7) that  $\mathbf{v}_k$  is directly correlated with  $\tilde{\mathbf{x}}_k^+$  instead of  $\tilde{\mathbf{x}}_k^-$ ) yields

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (3.15)$$

There are numerous ways to capture the overall deviation of the filter estimates from the truth. The diagonal terms of the error covariance correspond to the error covariance of each state, thus, one way is to minimize the sum of the diagonal terms, or trace, of the error covariance. Let's define the cost function as

$$J(K_k) = \text{Tr}[P_k^+] \quad (3.16)$$

We take the derivative of this cost function and set it to zero to find the extremal. Also,  $\frac{\partial}{\partial A} [\text{Tr}(ABA^T)] = 2AB$  for symmetric  $B$ . Thus, the partial derivative of Eq. (3.16) (note

that the error covariance matrices are symmetric) is

$$\frac{\partial}{\partial K_k} J(K_k) = 0 = -2(I - K_k H_k) P_k^- H_k^T + 2K_k R_k \quad (3.17)$$

Solving the above equation for  $K_k$  gives

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (3.18)$$

Expanding Eq. (3.13b) leads to

$$P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k [H_k P_k^- H_k^T + R_k] K_k^T \quad (3.19)$$

Substituting Eq. (3.18) for the  $K_k$  before the square bracket yields

$$P_k^+ = P_k^- - K_k H_k P_k^- \quad (3.20)$$

$$= [I - K_k H_k] P_k^- \quad (3.21)$$

with initial condition given by  $P_0^+ = E\{\tilde{x}_0^+ \tilde{x}_0^{+T}\}$ . As long as  $R_k$  is positive definite and  $Q_k$  semi-definite, the linear KF is always stable. A summary of the discrete KF algorithm is given in Table 3.1

Table 3.1: Discrete Kalman Filter

Model	$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k$ $\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$ $P_0 = E\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$
Gain	$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-)$ $P_k^+ = [I - K_k H_k] P_k^-$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$

### 3.3 Extended Kalman Filter

#### 3.3.1 Continuous-Discrete Extended Kalman Filter

The continuous nonlinear model equation with discrete-time measurements are

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + G(t) \mathbf{w}(t) \quad (3.22a)$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (3.22b)$$

The noises are again

$$E\{\mathbf{w}(t)\} = \mathbf{0} \quad E\{\mathbf{w}(\tau) \mathbf{w}(t)^T\} = Q(t) \delta(t - \tau) \quad (3.23a)$$

$$E\{\mathbf{v}_k\} = \mathbf{0} \quad E\{\mathbf{v}_j \mathbf{v}_k^T\} = \delta_{jk} R_k \quad (3.23b)$$

where  $Q(t)$  is the power spectral density matrix of the process noise and  $\delta(t - \tau)$  is the Dirac delta function. Applying a Taylor's series expansion on the nonlinear terms in Eq. (3.22) yields

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) + \text{h.o.t.} \quad (3.24a)$$

$$\mathbf{h}_k(\mathbf{x}_k) = \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \text{h.o.t.} \quad (3.24b)$$

where h.o.t. abbreviates “higher order terms”. It is assumed that if the estimate is close to the truth, then the first order term of the Taylor's series expansion dominates. Assuming that the partial derivatives exist, we define

$$F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \quad H_k(\hat{\mathbf{x}}_k^-) \equiv \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \quad (3.25)$$

Note that  $F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)$  is expanded about the current estimate (conditional mean) while  $H_k(\hat{\mathbf{x}}_k^-, t)$  is expanded about the propagated states. Neglecting the h.o.t., then Eq. (3.24) becomes

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \simeq \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)(\mathbf{x} - \hat{\mathbf{x}}) \quad (3.26a)$$

$$\mathbf{h}_k(\mathbf{x}_k) \simeq \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + H_k(\hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \quad (3.26b)$$

Taking the expectation of Eq. (3.26a) yields

$$\hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \quad (3.27)$$

Taking the expectation of Eq. (3.22a) and with Eq. (3.27) leads to the propagation equation for the state vector between measurement times, i.e.  $t_{k-1} \leq t < t_k$

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \quad (3.28)$$

The continuous error covariance propagation equation is given by

$$\dot{P}(t) = F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)P(t) + P(t)F^T(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + G(t)Q(t)G^T(t) \quad (3.29)$$

Now let's consider the update equations for the state vector and error covariance. Motivated by the linear update equation of Eq. (3.5) we again have

$$\hat{\mathbf{x}}_k^+ = K'_k \hat{\mathbf{x}}_k^- + K_k \tilde{\mathbf{y}}_k \quad (3.30)$$

and the definition of the error states of Eq. (3.6)

$$\tilde{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{x}_k \quad (3.31a)$$

$$\tilde{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k \quad (3.31b)$$

Substituting Eq. (3.22b) into Eq. (3.30) and then the resultant into Eq. (3.31b) and with Eq. (3.31a) for  $-\mathbf{x}_k$  leads to

$$\tilde{\mathbf{x}}_k^+ = K'_k \hat{\mathbf{x}}_k^- + K_k \mathbf{h}_k(\mathbf{x}_k) + K_k \mathbf{v}_k + \tilde{\mathbf{x}}_k^- - \hat{\mathbf{x}}_k^- \quad (3.32)$$

Taking the expectation of Eq. (3.32) with  $E\{\tilde{\mathbf{x}}_k^+\} = E\{\tilde{\mathbf{x}}_k^-\} = E\{\mathbf{v}_k\} = 0$  and solving for  $K'_k \tilde{\mathbf{x}}_k^-$  yields

$$K'_k \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - K_k \hat{\mathbf{h}}_k(\mathbf{x}_k) \quad (3.33)$$

Substituting this into Eq. (3.30) gives

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \hat{\mathbf{h}}_k(\mathbf{x}_k)] \quad (3.34)$$

Substituting Eq. (3.33) into Eq. (3.32) and using the approximation of Eq. (3.26b) yields

$$\begin{aligned} \tilde{\mathbf{x}}_k^+ &= \tilde{\mathbf{x}}_k^- + K_k[\mathbf{h}_k(\mathbf{x}_k) - \hat{\mathbf{h}}_k(\mathbf{x}_k)] + K_k \mathbf{v}_k \\ &= \tilde{\mathbf{x}}_k^- + K_k[\mathbf{h}_k(\hat{\mathbf{x}}_k^-) + H_k(\hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \hat{\mathbf{h}}_k(\mathbf{x}_k)] + K_k \mathbf{v}_k \\ &= \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)] \end{aligned} \quad (3.35)$$

The Kalman gain  $K_k$  is determined in the same fashion as before. Again, we choose to minimize the mean-square-error (MSE) by minimizing the trace of the error covariance matrix. Again, the definition of error covariances are

$$P_k^- \equiv E\{\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{-T}\} \quad (3.36a)$$

$$P_k^+ \equiv E\{\tilde{\mathbf{x}}_k^+ \tilde{\mathbf{x}}_k^{+T}\} \quad (3.36b)$$

Using Eq. (3.35) in Eq. (3.36b) with the assumption that  $P_k^+$  and  $\tilde{\mathbf{y}}_k$  are independent from each other gives

$$\begin{aligned} P_k^+ &= P_k^- + K_k E\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)][\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} K_k^T \\ &\quad + E\{\tilde{\mathbf{x}}_k^- [\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} + K_k E\{[\mathbf{h}_k(\mathbf{x}_k) \\ &\quad - \tilde{\mathbf{h}}_k(\mathbf{x}_k)] \tilde{\mathbf{x}}_k^{-T}\} + K_k R_k K_k^T \end{aligned} \quad (3.37)$$

The cost function is to minimize the trace of  $P_k^+$ :

$$J(K_k) = Tr[P_k^+] \quad (3.38)$$

Taking the partial derivative gives

$$\frac{\partial}{\partial K_k} J(K_k) = 0 \quad (3.39)$$

and solving for  $K_k$  with approximation of Eq. (3.26b) yields

$$\begin{aligned} K_k &= -E\{\tilde{\mathbf{x}}_k^- [\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} \\ &\quad \times \left[ E\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)][\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} + R_k \right]^{-1} \\ &= P_k^- H_k^T(\hat{\mathbf{x}}_k^-) [H_k(\hat{\mathbf{x}}_k^-) P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1} \end{aligned} \quad (3.40)$$

Substituting Eq. (3.40) into Eq. (3.37) with approximation of Eq. (3.26b) *mutatis mutandis* leads to

$$\begin{aligned} P_k^+ &= P_k^- + K_k E\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]\tilde{\mathbf{x}}_k^{-T}\} \\ &= [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^- \end{aligned} \quad (3.41)$$

## A Summary of the Continuous-Discre

Table 3.2: Continuous-Discrete Extended Kalman Filter

Model	$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + G(t)\mathbf{w}(t)$ $\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$ $P_0 = E\{\tilde{\mathbf{x}}_0\tilde{\mathbf{x}}_0^T\}$
Gain	$K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-)P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1}$ $H_k(\hat{\mathbf{x}}_k^-) \equiv \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \Big _{\mathbf{x}=\hat{\mathbf{x}}_k^-}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]$ $P_k^+ = [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^-$
Propagation	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)$ $\dot{P}(t) = F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)P(t) + P(t)F^T(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + G(t)Q(t)G^T(t)$ $F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big _{\mathbf{x}=\hat{\mathbf{x}}}$

it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function given a fixed number of parameters [24].

The EKF basically linearizes the system and measurement models to first order with Taylor's series expansion, then simplifies update them with a linear KF. This “first-order” assumption usually works well when the sampling time is small, and when the system behaves in a quasi-linear manner. However, this linearized propagation could easily cause an instability in the filter, which leads to divergence of the filter. The nonlinear transformation, the heart of the UF, called the *unscented transformation* (UT), propagates the states with a minimal set of carefully selected sample points through a nonlinear transformation that captures the posterior mean and covariance accurately to higher order Taylor series expansion [25].

Table 3.3: Discrete Extended Kalman Filter

Model	$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + G_k \mathbf{w}_k$ $\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$ $P_0 = E\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$
Gain	$K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-)P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1}$ $H_k(\hat{\mathbf{x}}_k^-) \equiv \left. \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \right _{\mathbf{x}=\hat{\mathbf{x}}_k^-}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k [\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]$ $P_k^+ = [I - K_k h_k(\hat{\mathbf{x}}_k^-)]P_k^-$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_k, k)$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k + G_k Q_k G_k^T$

The discrete-time system and measurement models with additive noises are assumed to be

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + G_k \mathbf{w}_k \quad (3.42a)$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \quad (3.42b)$$

where  $\mathbf{x}_k$  is the  $n \times 1$  state vector;  $\tilde{\mathbf{y}}_k$  is the  $m \times 1$  measurement vector;  $\mathbf{w}_k$  is the process noise and  $\mathbf{v}_k$  is the measurement noise. As before, we further assume both the process noise and the measurement noise are zero-mean Gaussian noise processes with covariances  $Q_k$  and  $R_k$  respectively, as given by Eqs. (3.2) and (3.4). The KF update scheme is first

rewritten as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \mathbf{v}_k \quad (3.43a)$$

$$P_k^+ = P_k^- - K_k P_k^{vv} K_k^T \quad (3.43b)$$

where  $\mathbf{v}_k$  is the *innovation* given by

$$\mathbf{v}_k \equiv \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, k) \quad (3.44)$$

and  $P_k^{vv}$  is the covariance of  $\mathbf{v}_k$ . The Kalman gain  $K_k$  in this case is then given by

$$K_k = P_k^{xy} (P_k^{vv})^{-1} \quad (3.45)$$

where  $P_k^{xy}$  is the cross-correlation matrix between  $\hat{\mathbf{x}}_k^-$  and  $\hat{\mathbf{y}}_k^-$ .

As mentioned before, the core of the UF is its different propagation that more accurately captures the mean and covariance during the propagation phase with the UT. Instead of a Monte-Carlo style approach to retain the true mean and covariance, the UT generates a discrete distribution with minimum and carefully selected *sigma points* that have the same first and second and possible higher moments. Reference [26] shows that these finite points estimate the true mean and covariance better than the EKF without a computationally-burdensome Monte-Carlo run. With an  $n$ -dimensional Gaussian distribution with error covariance matrix  $P$ , first generate a set of  $2n$  sigma points

$$\boldsymbol{\sigma}_k \leftarrow 2n \text{ columns from } \pm \sqrt{(n + \lambda)[P_k^+ + Q_k]} \quad (3.46a)$$

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+ \quad (3.46b)$$

$$\boldsymbol{\chi}_k(i) = \boldsymbol{\sigma}_k(i) + \hat{\mathbf{x}}_k^+ \quad (3.46c)$$

that have the same covariance from the columns or rows. These sigma points has zero-mean. If the Gaussian distribution has mean  $\mu$ , simply adding  $\mu$  to each of these points would yield the desired mean and covariance [24]. The symmetric nature of the sigma points set guarantee its first three moments to be the same as the original Gaussian distribution. Also, the odd moments are all zero due to this symmetry. For any choice of the scalar  $\lambda$ , the first three moments remain unchanged, however, it is a useful parameter for exploiting knowledge about higher moments. For the Gaussian distribution, choosing  $\lambda$  such that  $n + \lambda = 3$  minimizes the MSE up to fourth order [24]. One efficient method to compute the matrix square-root is the Cholesky decomposition. These sigma points are then propagated with

$$\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}[\boldsymbol{\chi}_k(i), k] \quad (3.47)$$

The predicted mean is given by

$$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n + \lambda} \left\{ \lambda \boldsymbol{\chi}_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\chi}_{k+1}(i) \right\} \quad (3.48)$$

The predicted error covariance is given by

$$\begin{aligned} P_{k+1}^- = & \frac{1}{n + \lambda} \left\{ \lambda [\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-]^T \right. \\ & \left. + \frac{1}{2} \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T \right\} \end{aligned} \quad (3.49)$$

The mean predicted observation is given by

$$\hat{\mathbf{y}}_{k+1}^- = \frac{1}{n + \lambda} \left\{ \lambda \boldsymbol{\gamma}_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\gamma}_{k+1}(i) \right\} \quad (3.50)$$

where

$$\boldsymbol{\gamma}_{k+1}(i) = \mathbf{h}[\boldsymbol{\chi}_{k+1}(i), k] \quad (3.51)$$

The output covariance is given by

$$P_{k+1}^{yy} = \frac{1}{n+\lambda} \left\{ \lambda [\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\} \quad (3.52)$$

Then, the innovation covariance is simply given by

$$P_{k+1}^{vv} = P_{k+1}^{yy} + R_{k+1} \quad (3.53)$$

Finally, the cross-correlation matrix is determined using

$$P_{k+1}^{xy} = \frac{1}{n+\lambda} \left\{ \lambda [\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\} \quad (3.54)$$

Then the Kalman gain is computed with Eq. (3.45), and the state vector can now be updated using Eq. (3.43). A summary of the UF algorithm is presented in Table 3.4.

Notice that the predicted error covariance, Eq. (3.49), has the potential to become non-positive semi-definite when  $\lambda$  is negative. This could potentially cause difficulty in Eq. (3.46a). If this is the case, numerous approaches can be used to guarantee its positive semi-definiteness. One way is to neglect the  $\lambda$  term from Eq. (3.49), Eq. (3.54) and

Table 3.4: Unscented Filter

Model	$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + G_k \mathbf{w}_k$ $\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$ $P_0 = E\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$
Gain	$K_k = P_k^{xy} (P_k^{vv})^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \mathbf{v}_k$ $P_k^+ = P_k^- - K_k P_k^{vv} K_k^T$ $\mathbf{v}_k \equiv \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, k)$
Propagation	$\boldsymbol{\sigma}_k \leftarrow 2n$ columns from $\pm \sqrt{(n + \lambda)[P_k^+ + Q_k]}$ $\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+$ $\boldsymbol{\chi}_k(i) = \boldsymbol{\sigma}_k(i) + \hat{\mathbf{x}}_k^+$ $\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}[\boldsymbol{\chi}_k(i), k]$ $\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n+\lambda} \left\{ \lambda \boldsymbol{\chi}_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\chi}_{k+1}(i) \right\}$ $P_{k+1}^- = \frac{1}{2(n+\lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T \right\}$ $\hat{\mathbf{y}}_{k+1}^- = \frac{1}{n+\lambda} \left\{ \lambda \boldsymbol{\gamma}_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\gamma}_{k+1}(i) \right\}$ $\boldsymbol{\gamma}_{k+1}(i) = \mathbf{h}[\boldsymbol{\chi}_{k+1}(i), k]$ $P_{k+1}^{yy} = \frac{1}{2(n+\lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\}$ $P_{k+1}^{vv} = P_{k+1}^{yy} + R_{k+1}$ $P_{k+1}^{xy} = \frac{1}{2(n+\lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\}$

Eq. (3.52) [19], which leads to

$$P_{k+1}^- = \frac{1}{2(n + \lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T \right\} \quad (3.55)$$

$$P_{k+1}^{yy} = \frac{1}{2(n + \lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\} \quad (3.56)$$

$$P_{k+1}^{xy} = \frac{1}{2(n + \lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-] [\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T \right\} \quad (3.57)$$

For other methods to guarantee the semi-positiveness of the error covariance, refer to Refs. [26] and [27].

The UF generally requires more computational power from the  $2n + 1$  propagations, however, the computational requirement could be comparable to the EKF especially when numerical Jacobians need to be evaluated. Another possible performance improvement of the UF is through parallel processing of the propagation equations. When noises are not additive in the system or measurement model, another method that incorporates these noises into the system states and error covariance is presented in Ref. [26]. This method promises possible higher performance (lower estimated error covariance or better stability), but vastly increase the computational load.

## 3.5 Summary

Various sequential filters are discussed in this chapter: the KF, EKF and UF. The traditional KF applies only for linear systems, however, most of the real world problems are nonlinear in nature. The EKF is the nonlinear extension of the traditional KF. It linearizes the system models so that linear KF equations apply, instead of linearizing the system models. The UF propagates the Gaussian distribution directly, this results in a lower expected error

covariance which leads to higher estimation accuracy. The UF linearizes to higher order than the EKF and offers better treatment of the statistics, hence it is more robust than the EKF when higher nonlinearity is a concern. Also, no Jacobian derivations are needed in the UF formulation, thus the UF is far easier to implement on nonlinear systems.

# 4 Attitude Sensor Models with Errors

## 4.1 Introduction

In this chapter, misalignment models are derived for vector sensors and gyros. Many of the attitude sensors belong to this class of vector sensor, for example, Sun sensors, Earth sensors, Star sensors or sometimes payload sensors. This class of sensors measure the direction or line-of-sight (LOS) of a body (often of celestial origin) with known “position” either from ephemeris data or mathematical model. Finally, the gyro model is presented. The gyro model also includes bias and scale factor errors. Gyros are usually needed as part of the dynamical propagation of attitude kinematics/dynamics model. Since most of the modern sensors are discrete sensors, we will only consider this class of sensors in this work.

## 4.2 Vector Attitude Sensor

Many modern attitude sensors that output a unit vector measurement are modeled by

$$\tilde{\mathbf{b}}_i = A(\mathbf{q})\mathbf{r}_i + \mathbf{v}_i \quad (4.1)$$

where  $\tilde{\mathbf{b}}_i$  denotes the  $i^{th}$   $3 \times 1$  measurement vector,  $\mathbf{r}_i$  is the  $i^{th}$  known  $3 \times 1$  reference vector, and the sensor error-vector  $\mathbf{v}_i$  is Gaussian which satisfies

$$E\{\mathbf{v}_i\} = 0 \quad (4.2)$$

$$E\{\mathbf{v}_i \mathbf{v}_i^T\} = \sigma_i^2 I \quad (4.3)$$

Shuster proved in Ref. [28] that the measurement noise,  $\mathbf{v}_i$ , are random noise concentrated in a very small area about the direction of the  $A(\mathbf{q})\mathbf{r}$  or in other words this area can be approximated orthogonal/tangent plane to the measurement vector. Note that if unit measurement vectors are used then Eq. (4.3) should be appropriately modified. Multiple ( $N$ ) vector measurements can be concatenated to form

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_1 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_N \end{bmatrix}_k + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \end{bmatrix}_k \quad (4.4)$$

$$R_k = \text{diag}[\sigma_1^2 \ \sigma_2^2 \ \dots \ \sigma_N^2] \quad (4.5)$$

where diag denotes a diagonal matrix of appropriate dimension. We should note that any attitude sensor, such as a three-axis magnetometer, star sensor, sun sensor, etc., can be put into the form given by Eq. (4.1). However, most sensors only observe two quantities, such as two angles in star sensors. The resulting form in Eq. (4.1) for these type of sensors has a unity norm constraint in the observation. However, the mean observation given by Eq. (3.50) may not produce an estimate with unit norm. Therefore, it is recommended that the original two quantity observation model be used for these types of sensors in the UF.

We now consider misalignments to this vector sensor model

$$\tilde{\mathbf{b}} = [I - \boldsymbol{\varsigma} \times] A_{s_0, b} A(\mathbf{q}) \mathbf{r}_{ref} \quad (4.6)$$

where  $\boldsymbol{\varsigma}$  is the sensor orthogonal misalignments that follows the small angle approximation presented in Section 2.3,  $A_{s_0, b}$  is the transformation (alignment) matrix from the body to the nominal (assumed or designed) sensor coordinate and  $\mathbf{r}_{ref}$  is known vector in ECI coordinates of the observed object from an ephemeris calculation. Here again we employ a small angle approximation for the misalignment matrix. The estimated observation is then trivial

$$\hat{\mathbf{b}} = [I - \hat{\boldsymbol{\varsigma}} \times] A_{s_0, b} A(\hat{\mathbf{q}}) \mathbf{r}_{ref} \quad (4.7)$$

### 4.3 Gyro

A common sensor that measures the angular rate is the rate-integrating gyro. For this sensor, a widely used model is given by [29]

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) + \boldsymbol{\eta}_v(t) \quad (4.8)$$

$$\dot{\boldsymbol{\beta}}(t) = \boldsymbol{\eta}_u(t) \quad (4.9)$$

where  $\tilde{\boldsymbol{\omega}}(t)$  is the continuous-time measured angular rate, and  $\boldsymbol{\eta}_v(t)$  and  $\boldsymbol{\eta}_u(t)$  are independent zero-mean Gaussian white-noise processes with

$$E\{\boldsymbol{\eta}_v(t)\boldsymbol{\eta}_v^T(\tau)\} = I_{3 \times 3} \sigma_v^2 \delta(t - \tau) \quad (4.10)$$

$$E\{\boldsymbol{\eta}_u(t)\boldsymbol{\eta}_u^T(\tau)\} = I_{3 \times 3} \sigma_u^2 \delta(t - \tau) \quad (4.11)$$

where  $\delta(t - \tau)$  is the Dirac delta function.

In the standard EKF, given a post-update estimate  $\hat{\beta}_k^+$ , the post-update angular velocity and propagated gyro bias follow

$$\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+ \quad (4.12)$$

$$\hat{\beta}_{k+1}^- = \hat{\beta}_k^- \quad (4.13)$$

Given post-update estimates  $\hat{\omega}_k^+$  and  $\hat{\mathbf{q}}_k^+$ , the propagated quaternion is found from the discrete-time equivalent of Eq. (2.21):

$$\hat{\mathbf{q}}_{k+1}^- = \Omega(\hat{\omega}_k^+) \hat{\mathbf{q}}_k^+ \quad (4.14)$$

with

$$\Omega(\hat{\omega}_k^+) \equiv \begin{bmatrix} Z_k & \hat{\varphi}_k^+ \\ -\hat{\varphi}_k^+ & \cos(0.5\|\hat{\omega}_k^+\|\Delta t) \end{bmatrix} \quad (4.15)$$

$$Z_k \equiv \cos(0.5\|\hat{\omega}_k^+\|\Delta t) I_{3 \times 3} - [\hat{\varphi}_k^+ \times] \quad (4.16)$$

$$\hat{\varphi}_k^+ \equiv \sin(0.5\|\hat{\omega}_k^+\|\Delta t) \hat{\omega}_k^+ / \|\hat{\omega}_k^+\| \quad (4.17)$$

where  $\Delta t$  is the sampling interval of the gyro.

The gyro model follows the development from Refs. [5] and [30] with some modifications. The gyro model from Eq. (4.8) is modified with added misalignments and scale factor errors and is given by

$$\tilde{\omega} = (I - \tilde{\Lambda} - \tilde{U})(I - \tilde{\Delta}) A_{g_0, b} \omega + \beta + \eta_v \quad (4.18)$$

where  $\tilde{\Lambda} = \text{diag}(\lambda_x, \lambda_y, \lambda_z)$  is a matrix of symmetric scale factor errors,  $\tilde{U} = \text{diag}(\mu_x \text{sign}(\omega_x), \mu_y \text{sign}(\omega_y), \mu_z \text{sign}(\omega_z))$  is a matrix of asymmetric scale factor errors,  $\tilde{\Delta}$  is a non-orthogonal

small angle misalignment matrix (again, the small angle approximation from Section 2.3 is used; note that the sign changes later as an approximation to the inverse of the matrix), and  $A_{g_0,b}$  is a transformation matrix from body coordinates to nominal gyros coordinates. Note that  $\beta$  and  $\eta_v$  are now in the true (misaligned) gyros coordinate system. Thus the true angular rate in body coordinates is

$$\begin{aligned}\omega &= A_{b,g_0}(I - \tilde{\Delta})^{-1}(I - \tilde{\Lambda} - \tilde{U})^{-1}(\tilde{\omega} - \beta - \eta_v) \\ &\simeq A_{b,g_0}(I + \Delta)(I + \Lambda + U)(\tilde{\omega} - \beta - \eta_v)\end{aligned}\quad (4.19)$$

with  $\Delta \simeq \tilde{\Delta}$ ,  $\Lambda \simeq \tilde{\Lambda}$  and  $U \simeq \tilde{U}$  with a small angle approximation. Since each axis is assumed to be misaligned independently we can sum up their orthogonal small angle rotations of the angular rate about each axis [5]. Let's temporarily define  $\bar{\omega} = (I + \Lambda + U)(\tilde{\omega} - \beta - \eta_v)$ . From Eq. (4.19) we have

$$\begin{aligned}(I + \Delta)\bar{\omega} &= \begin{bmatrix} 1 & -\delta_{xz} & \delta_{xy} \\ \delta_{xz} & 1 & 0 \\ -\delta_{xy} & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{\omega}_x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -\delta_{yz} & 0 \\ \delta_{yz} & 1 & -\delta_{yx} \\ 0 & \delta_{yx} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \bar{\omega}_y \\ 0 \end{bmatrix} \\ &\quad + \begin{bmatrix} 1 & 0 & \delta_{zy} \\ 0 & 1 & -\delta_{zx} \\ -\delta_{zy} & \delta_{zx} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \bar{\omega}_z \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\delta_{yz} & \delta_{zy} \\ \delta_{yz} & 1 & -\delta_{zx} \\ -\delta_{xy} & \delta_{yx} & 1 \end{bmatrix} \bar{\omega}\end{aligned}\quad (4.20)$$

We then perform a QR factorization so that  $I + \Delta = QR$  where  $Q$  is orthogonal and  $R$  is

upper triangle:

$$\begin{aligned}
 I + \Delta &= QR \\
 &= [I + \boldsymbol{\delta} \times] R \\
 &= \begin{bmatrix} 1 & -\delta_z & \delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & \xi_z & \xi_y \\ 0 & 1 & \xi_x \\ 0 & 0 & 1 \end{bmatrix} \\
 &\simeq \begin{bmatrix} 1 & -(\delta_z - \xi_z) & \delta_y + \xi_y \\ \delta_z & 1 & -(\delta_x - \xi_x) \\ -\delta_y & \delta_x & 1 \end{bmatrix} \tag{4.21}
 \end{aligned}$$

Equating Eq. (4.21) with  $I + \Delta$  from Eq. (4.20) we find that

$$\begin{aligned}
 \delta_x &= \delta_{yx} & \xi_x &= \delta_{yx} - \delta_{zx} \\
 \delta_y &= \delta_{xy} & \xi_y &= \delta_{zy} - \delta_{xy} \\
 \delta_z &= \delta_{xz} & \xi_z &= \delta_{xz} - \delta_{yz}
 \end{aligned} \tag{4.22}$$

Since we are using the gyros as the “reference” sensors to calibrate other sensors, we equate the orthogonal components of gyro misalignment to zero. Thus  $\delta_z = \delta_y = \delta_z = 0$  and we obtain

$$\begin{aligned}
 \xi_x &= -\delta_{zx} \\
 \xi_y &= \delta_{zy} \\
 \xi_z &= -\delta_{yz}
 \end{aligned}$$

and

$$\begin{aligned} I + \Delta &= R \\ &= \begin{bmatrix} 1 & \xi_z & \xi_y \\ 0 & 1 & \xi_x \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (4.23)$$

From Eq. (4.19) we have

$$\begin{aligned} \boldsymbol{\omega} &= A_{b,g_0}(I + \Delta)(I + \Lambda + U)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v) \\ &\simeq A_{b,g_0}(I + \Delta + \Lambda + U)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v) \end{aligned} \quad (4.24)$$

Let's define  $M = \Delta + \Lambda + U$  (which will be used later) and  $\check{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\beta}$ . With the latter definition, the previous equation becomes

$$\begin{aligned} \boldsymbol{\omega} &= A_{b,g_0}(I + \Delta + \Lambda + U)(\check{\boldsymbol{\omega}} - \boldsymbol{\eta}_v) \\ &= A_{b,g_0}\check{\boldsymbol{\omega}} - A_{b,g_0}(I + \Delta + \Lambda + U)\boldsymbol{\eta}_v + A_{b,g_0}(\Delta + \Lambda + U)\check{\boldsymbol{\omega}} \\ &= A_{b,g_0}\check{\boldsymbol{\omega}} - A_{b,g_0}(I + \Delta + \Lambda + U)\boldsymbol{\eta}_v \\ &\quad \begin{bmatrix} 0 & \check{\omega}_z & \check{\omega}_y & \check{\omega}_x & 0 & 0 & |\check{\omega}_x| & 0 & 0 \\ \check{\omega}_x & 0 & 0 & 0 & \check{\omega}_y & 0 & 0 & |\check{\omega}_y| & 0 \\ 0 & 0 & 0 & 0 & 0 & \check{\omega}_z & 0 & 0 & |\check{\omega}_z| \end{bmatrix} \\ &\quad \times [\xi_x \ \xi_y \ \xi_z \ \lambda_x \ \lambda_y \ \lambda_z \ \mu_x \ \mu_y \ \mu_z]^T \\ &= A_{b,g_0}\check{\boldsymbol{\omega}} + A_{b,g_0}\Omega_g\boldsymbol{\kappa} - A_{b,g_0}(I + \Delta + \Lambda + U)\boldsymbol{\eta}_v \end{aligned} \quad (4.25)$$

Note that the last three columns from the  $3 \times 9$  matrix from the above equation,  $\check{\omega}_i$  where  $i = x, y, z$ , are an approximation of  $\text{sign}(\omega_i)\check{\omega}_i \simeq |\check{\omega}_i|$ , which is valid assuming that the bias is small. Also,  $\Omega_g$  and  $\boldsymbol{\kappa}$  correspond to the previous matrix of gyro measurements and

vector of gyro misalignments and scale factor errors. Thus our angular velocity estimate is

$$\hat{\boldsymbol{\omega}} = A_{b,g_0} \check{\boldsymbol{\omega}} + A_{b,g_0} \hat{\Omega}_g \hat{\boldsymbol{\kappa}} \quad (4.26)$$

where the hats correspond to their respective estimated values. Note that  $\check{\boldsymbol{\omega}}$  here is a function of both  $\tilde{\boldsymbol{\omega}}$  (the uncompensated, measured gyro rate) and  $\hat{\boldsymbol{\beta}}$  (the estimated gyro bias).

## 4.4 Summary

Common spacecraft sensor models are given in this chapter. First the vector sensor model was given, followed by the gyros model. The vector sensor model was added with misalignments. The gyro model also includes mechanical and electrical errors. The mechanical error is the physical misalignment; the electrical errors include bias drifts (which could be of mechanical origin too, depending on the type of gyros), scale factor errors and sensor noise. Since the gyros are used as the “reference” sensor, we set the orthogonal misalignments of the gyros to zero. Also, the discrete-time quaternion propagation is included.

Another prominent attitude sensor is the three-axis magnetometer (TAM). This sensor is reliable, lightweight with low power consumption. Even if uncalibrated, attitude accuracies of 1–2 degrees is common. Attitude accuracy of 0.1–0.5 degrees and rate accuracy of 0.001 deg/sec – 0.005 deg/sec per axis is possible with careful calibration [31]. Thus, the TAM is widely used in most low Earth orbit (below 1000km altitude) satellites. It serves well for coarse attitude determination and as a contingency sensor. While TAM calibration remains an active research area, and the in-depth background involved, it will not be discussed in this work.

# 5 Attitude and Alignment Filter

## 5.1 Introduction

This section addresses attitude estimation. Attitude estimation is the estimation of the orthogonal rotation matrix that would map the reference observation into the sensor frame [19]. Because of the duality between alignment calibration and attitude estimation in our formulation, this is further extended into an attitude and alignment filter with the inclusion of calibration parameters.

## 5.2 Attitude Estimation

Attitude estimation is divided into two distinct categories: single-frame and multi-frame based methods. Single-frame methods are also called deterministic methods. Multi-frame methods are sometimes referred to as recursive estimation, which involve a continuous span of measurement data. Multi-frame methods, can further be subdivided into batch and sequential estimation. This statistical scheme constantly finds the best agreement between the estimated sensor observation and noise corrupted sensor measured observations.

Some single-frame attitude estimation methods included the Singular Value Decomposition (SVD), TRIAD, Fast Optimal Attitude Estimator (FOAM) and QUEST. Reference [32] has a survey on various algorithms for deterministic three-axis attitude estimation. The advantages of single-frame algorithms are that they require little/poor or no initial condi-

tions at all. Also, they tend to be less computational burdensome. The disadvantages are that they mostly require simultaneous sensor measurements and are less accurate than the multi-frame estimation methods.

The most popular sequential multi-frame attitude estimation method is the EKF. This method uses sequential/subsequent sensor measurements with a dynamics/kinematics model of the system to continuously output the best estimate of the “truth”. Another prominent multi-frame estimation scheme is the UF that linearizes the dynamics model to higher order than the EKF. Multi-frame algorithms like the EKF or UF are able to update the estimated states at sporadic measurement points even at different sampling intervals. Employing a dynamics model not only increases the accuracy of estimation (lower error covariance) but also propagates the states into future time when no measurement is available. However, multi-frame algorithms for nonlinear systems suffer from divergence possibilities, especially with poor initial conditions, and high computational resources.

Multi-frame estimation provides better results since it has memory of the past measurements (and statistical properties) and most of the time is equipped with knowledge of the dynamical model. It also works with more sensors measurements or even with single sensor updates. The use of more measurements is always desired as the random error is inversely proportional to the square root of the number of measurements [33]. The focus of this thesis is on multi-frame methods, especially sequential attitude estimation and alignment calibration. Thus single-frame schemes will not be pursued further. Interested readers are referred to Refs. [34], [35] and [36].

The quaternion is the preferred attitude representation for on board. The quaternion is a four-parameter attitude representation that specifies the rotational axis, also known as eigenaxis, and the corresponding rotational angle. The quaternion kinematics equation is free from computational costly transcendental functions and its relationship with rotational rate is linear making it ideal for onboard implementation. However, since a four-parameter rep-

resentation is used for the three-dimensional world, the quaternion cannot be independent. The quaternion satisfies the norm constraint  $\mathbf{q}^T \mathbf{q} = 1$ . This poses a problem in the EKF formulation where the covariance matrix would become singular. To avoid this problem, Lefferts et al. [1] proposed to estimate only the first three parameters of the error quaternion instead. The scalar part or the fourth component of the error quaternion is assumed to be close to 1 as the “difference” between the propagated and true quaternion is assumed to be “small”. Thus this reduces the state vector of the estimator by one component. The scalar part could be obtained from the constraint,  $q_4 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2}$ .

### 5.3 Wahba's Problem

Grace Wahba, in 1965, proposed an optimization problem for attitude estimation

$$J(\hat{\mathbf{A}}) = \frac{1}{2} \sum_{i=1}^N \sigma_i^{-2} \|\tilde{\mathbf{b}}_i - \hat{\mathbf{A}}\mathbf{r}_i\|^2 \quad (5.1)$$

At first glance, this may seems like a straightforward least-squares minimization problem but it is the following constraint that complicates the solution

$$\hat{\mathbf{A}}\hat{\mathbf{A}}^T = I_{3 \times 3} \quad (5.2)$$

This lead to extensive research in finding the optimal solution.

### 5.4 Calibration

*Calibrate refers to standardize (as a measuring instrument) by determining the deviation from a standard so as to ascertain the proper correction factors* [Merriam-Webster Dictionary]. Calibration is the process of checking and adjusting the accuracy of a measuring

instrument with respect to a standard. However, in our case, the “standard” (or the “truth”) which is the attitude and the rate of attitude change of the sensors or spacecraft in-space, is unknown. If the *truth* is known then attitude sensor calibration would simply be fitting of the calibration-curve with the truth. Instead, the attitude and rate must be estimated together with the calibration parameters. Thus our calibration algorithm uses an attitude estimator. The main focus of this thesis is the UF-based calibrations. But we also include calibrations with the EKF for performance comparisons. Below are the UF and EKF-based calibration derivations for a set of orthogonal gyros and two vector sensors. Let’s assume the first vector sensor as a star sensor and the second one as the payload sensor. The gyro error model includes biases, symmetric scale factor errors and nonorthogonal misalignments. In the UF attitude and alignment filter, we choose to perform the absolute alignment estimation, thus misalignments may be included into both the star sensor and the payload sensor. In the EKF attitude and alignment filter, we choose to follow Ref. [5] and perform the alignment with respect to the payload sensor, thus only the star sensor has misalignments.

### 5.4.1 Unscented Alignment Calibration

The Unscented Alignment Calibration filter is developed in this section. The filter is an extension of the USQUE algorithm developed in Ref. [37] with inclusion of calibration parameters into its state vector. The continuous-time attitude kinematics equation, including

calibration parameters, is given by

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \delta\varrho \\ \delta\beta \\ \delta\kappa \\ \delta\varsigma_1 \\ \delta\varsigma_2 \end{bmatrix} &= \begin{bmatrix} -[\boldsymbol{\omega} \times] & -\frac{1}{2}A_{b,g_0} & -\frac{1}{2}A_{b,g_0}\Omega_g & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\varrho \\ \delta\beta \\ \delta\kappa \\ \delta\varsigma_1 \\ \delta\varsigma_2 \end{bmatrix} \\ &+ \begin{bmatrix} -\frac{1}{2}A_{b,g_0}(I + M) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_v \\ \boldsymbol{\eta}_u \\ \boldsymbol{\eta}_\kappa \\ \boldsymbol{\eta}_{\varsigma_1} \\ \boldsymbol{\eta}_{\varsigma_2} \end{bmatrix} \end{aligned} \quad (5.3)$$

where the continuous process noise covariance (power spectral density) is given by

$$Q = \text{diag} [\sigma_v^2 I_{3 \times 3} \ \sigma_u^2 I_{3 \times 3} \ \sigma_\mu^2 I_{9 \times 9} \ \sigma_{\varsigma_1}^2 I_{3 \times 3} \ \sigma_{\varsigma_2}^2 I_{3 \times 3}] \quad (5.4)$$

The first square matrix on the right hand side of Eq. (5.3) is traditionally referred to as  $F(t)$  and the second square matrix as  $G(t)$ . This equation is generally not used to generate simulation data or filter propagation; however, we need the  $F(t)$  and  $G(t)$  matrices to compute the discrete process noise covariance. The discrete version of Eq. (5.3) with a

small angle approximation is

$$\begin{bmatrix} \delta\varrho \\ \delta\beta \\ \delta\mu \\ \delta\varsigma_1 \\ \delta\varsigma_2 \end{bmatrix}_{k+1} \simeq \begin{bmatrix} I & -\frac{1}{2}tA_{b,g_0} & -\frac{1}{2}tA_{b,g_0}\Omega_g & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \delta\varrho \\ \delta\beta \\ \delta\mu \\ \delta\varsigma_1 \\ \delta\varsigma_2 \end{bmatrix}_k + \Gamma_k \boldsymbol{\eta}_k \quad (5.5)$$

where  $\boldsymbol{\eta}_k$  is a discrete white process noise with  $E(\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T) = I$  and  $\Gamma_k(t)$  is a square root of the discrete process noise that we are going to derive. The first square matrix on the right hand side of the above equation is traditionally referred to as the state transition matrix or  $\Phi(t)$  matrix.

The discrete process noise is related to the continuous process noise by

$$Q_d = \int_0^{\Delta t} \Phi(t) G(t) Q(t) G^T(t) \Phi^T(t) dt \quad (5.6)$$

where  $\Delta t$  is the filter update interval. Substituting  $F(t)$ ,  $G(t)$  and  $\Phi(t)$  with  $M \simeq 0$  (small angle approximation) into the above equation yields [5]

$$Q_d = \Re \begin{bmatrix} \xi_{11} & \xi_{12} & \xi_{13} & 0 & 0 \\ \xi_{21} & \xi_{22} & 0 & 0 & 0 \\ \xi_{31} & \xi_{32} & \xi_{33} & 0 & 0 \\ 0 & 0 & 0 & \xi_{44} & 0 \\ 0 & 0 & 0 & 0 & \xi_{55} \end{bmatrix} \Re^T \quad (5.7)$$

where

$$\begin{aligned}
 \xi_{11} &= \frac{1}{4}\Delta t\sigma_v^2 + \frac{1}{12}\Delta t^3\sigma_u^2 + \frac{1}{12}\Delta t^3\Omega_g\sigma_\kappa^2\Omega_g^T \\
 \xi_{12} &= -\frac{1}{4}\Delta t^2\sigma_u^2 \\
 \xi_{13} &= -\frac{1}{4}\Delta t^2\Omega_g\sigma_\kappa^2 \\
 \xi_{21} &= -\frac{1}{4}\Delta t^2\sigma_u^2 \\
 \xi_{22} &= \Delta t\sigma_u^2
 \end{aligned} \tag{5.8}$$

$$\begin{aligned}
 \xi_{31} &= -\frac{1}{4}\Delta t^2\sigma_\kappa^2\Omega_g^T \\
 \xi_{33} &= \Delta t\sigma_\kappa^2 \\
 \xi_{44} &= \Delta t\sigma_{\varsigma_1}^2 \\
 \xi_{55} &= \Delta t\sigma_{\varsigma_2}^2 \\
 \Re &= \begin{bmatrix} A_{b,g_0} & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}
 \end{aligned} \tag{5.9}$$

For the reasons described in Ref. [37], we choose to propagate and update the quaternion using GRPs. We begin with

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+ \equiv \begin{bmatrix} \boldsymbol{\delta}\hat{\mathbf{p}}_k^+ \\ \hat{\boldsymbol{\beta}}_k^+ \\ \hat{\boldsymbol{\kappa}}_k^+ \\ \hat{\boldsymbol{\varsigma}}_{1,k}^+ \\ \hat{\boldsymbol{\varsigma}}_{2,k}^+ \end{bmatrix} \tag{5.10}$$

Equation (3.46) is partitioned into two parts, the attitude-error part and calibration parameters part:

$$\boldsymbol{\chi}_k(i) \equiv \begin{bmatrix} \boldsymbol{\chi}_k^{\delta p}(i) \\ \dots \\ \boldsymbol{\chi}_k^\beta(i) \\ \boldsymbol{\chi}_k^\kappa(i) \\ \boldsymbol{\chi}_k^{\varsigma_1}(i) \\ \boldsymbol{\chi}_k^{\varsigma_2}(i) \end{bmatrix}, \quad i = 0, 1, \dots, 36 \quad (5.11)$$

where  $\boldsymbol{\chi}_k^\beta$  is the gyro bias,  $\boldsymbol{\chi}_k^\kappa$  is the misalignment and scale factor part,  $\boldsymbol{\chi}_k^{\varsigma_1}$  is the star sensor misalignment part and  $\boldsymbol{\chi}_k^{\varsigma_2}$  is the payload sensor part. First, a new quaternion is generated by multiplying an error quaternion by the current estimate:

$$\hat{\mathbf{q}}_k^+(0) = \hat{\mathbf{q}}_k^+ \quad (5.12)$$

$$\hat{\mathbf{q}}_k^+(i) = \boldsymbol{\delta} \hat{\mathbf{q}}_k^+(i) \otimes \hat{\mathbf{q}}_k^+, \quad i = 1, 2, \dots, 36 \quad (5.13)$$

where  $\boldsymbol{\delta} \mathbf{q}_k^+ \equiv [\boldsymbol{\delta} \boldsymbol{\varrho}_k^{+T}(i) \ \delta q_{4,k}^+(i)]^T$  is represented by Eqs. (2.25a) and (2.25b)

$$\begin{aligned} \delta q_{4,k}^+(i) &= \frac{-\alpha \|\boldsymbol{\chi}_k^{\delta p}(i)\|^2 + f \sqrt{f^2 + (1 - \alpha^2) \|\boldsymbol{\chi}_k^{\delta p}(i)\|^2}}{f^2 + \|\boldsymbol{\chi}_k^{\delta p}(i)\|} \\ i &= 1, 2, \dots, 36 \end{aligned} \quad (5.14)$$

$$\begin{aligned} \boldsymbol{\delta} \boldsymbol{\varrho}_k^+(i) &= f^{-1}[\alpha + \delta q_{4,k}^+(i)] \boldsymbol{\chi}_k^{\delta p}(i) \\ i &= 1, 2, \dots, 36 \end{aligned} \quad (5.15)$$

From Eq. (5.12), it clearly requires that  $\boldsymbol{\chi}_k^{\delta p}(0)$  be zero. Equation (5.13) is then propagated with

$$\hat{\mathbf{q}}_{k+1}^-(i) = \Omega[\hat{\boldsymbol{\omega}}_k^+(i)] \hat{\mathbf{q}}_k^+(i), \quad i = 0, 1, \dots, 36 \quad (5.16)$$

with the propagated angular velocity from Eq. (4.26):

$$\begin{aligned}\hat{\boldsymbol{\omega}}_k^+(i) &= A_{b,g_0} \tilde{\boldsymbol{\omega}}_k - A_{b,g_0} \Omega_g \boldsymbol{\chi}_k^\kappa(i) - (I + \hat{M}(i)) \boldsymbol{\chi}_k^\beta(i) \\ i &= 0, 1, \dots, 36\end{aligned}\quad (5.17)$$

Note that  $\Omega_g$  is a function of  $\tilde{\boldsymbol{\omega}}_k$  defined in Eq. (4.24) and  $\hat{M}$  is a function of  $\boldsymbol{\chi}_k^\kappa(i)$ . The error quaternions are then propagated with

$$\begin{aligned}\boldsymbol{\delta q}_{k+1}^-(i) &= \hat{\mathbf{q}}_{k+1}^-(i) \otimes [\hat{\mathbf{q}}_{k+1}^-(0)]^{-1} \\ i &= 0, 1, \dots, 36\end{aligned}\quad (5.18)$$

Note that  $\boldsymbol{\delta q}_{k+1}^-(0)$  is the identity quaternion. We then convert these propagated quaternions back into sigma points with the GRPs representation using Eq. (2.24):

$$\boldsymbol{\chi}_{k+1}^{\delta p}(0) = \mathbf{0} \quad (5.19)$$

$$\begin{aligned}\boldsymbol{\chi}_{k+1}^{\delta p}(i) &= f \frac{\boldsymbol{\delta q}_{k+1}^-(i)}{a + \delta q_{4,k+1}^-(i)} \\ i &= 1, 2, \dots, 36\end{aligned}\quad (5.20)$$

The calibration parameters are expected to stay at their previous values, thus

$$\begin{aligned}\boldsymbol{\chi}_{k+1}^\beta(i) &= \boldsymbol{\chi}_k^\beta(i) \\ \boldsymbol{\chi}_{k+1}^\kappa(i) &= \boldsymbol{\chi}_k^\kappa(i) \\ \boldsymbol{\chi}_{k+1}^{\varsigma_1}(i) &= \boldsymbol{\chi}_k^{\varsigma_1}(i) \\ \boldsymbol{\chi}_{k+1}^{\varsigma_2}(i) &= \boldsymbol{\chi}_k^{\varsigma_2}(i) \\ i &= 0, 1, \dots, 36\end{aligned}\quad (5.21)$$

The procedure in the UF alignment calibration is as follows. First, we are given initial estimates of attitude ( $\hat{\mathbf{q}}_0^+$ ), gyros biases ( $\beta_0^+$ ), gyros misalignments and scale factor errors ( $\kappa_0^+$ ), star sensor misalignments ( $\varsigma_{1,0}^+$ ) and payload sensor misalignments ( $\varsigma_{2,0}^+$ ) with their respective initial covariances. The UF initial state vector is set to  $\hat{\mathbf{x}}_0^+ = [\mathbf{0}_{3 \times 1}^T \ \beta_0^{+T} \ \kappa_0^{+T} \ \varsigma_{1,0}^{+T} \ \varsigma_{2,0}^{+T}]^T$ . The sigma points are then calculated with Eq. (3.46). The attitude part of the sigma points (the first 3 components of each column of sigma points), is then converted back into quaternion with Eqs. (2.25b) and (2.25a) and propagated with Eq. (5.16). The propagated error quaternion is calculated using Eq. (5.18) and then transformed to propagated attitude sigma points using Eqs. (5.19) and (5.20). The propagated quaternions are used again later to find the mean observations using Eqs. (3.50) and (3.51) with

$$\begin{aligned}\gamma_{k+1}(i) &= \begin{bmatrix} [I - \chi_{k+1}^{S1}(i) \times] A_{s_0,b} A(\hat{\mathbf{q}}^-(i)) \mathbf{r}_{st} \\ [I - \chi_{k+1}^{S2}(i) \times] A_{p_0,b} A(\hat{\mathbf{q}}^-(i)) \mathbf{r}_p \end{bmatrix} \\ i &= 0, 1, \dots, 36\end{aligned}\quad (5.22)$$

where  $A_{s_0,b}$  is the nominal body to star sensor alignment matrix,  $A_{p_0,b}$  is the nominal body to payload sensor alignment matrix,  $\mathbf{r}_{st}$  is vector of star sensor observed object in ECI coordinates and  $\mathbf{r}_p$  is vector of payload sensor observed object in ECI coordinates. The output covariance, innovation covariance and cross-correlation matrix are computed using Eqs. (3.52), (3.53), and (3.54). The predicted mean and covariance can now be computed using Eqs. (3.48) and (3.49). States and covariance are then updated with Eq. (3.43) with  $\hat{\mathbf{x}}_{k+1}^+ \equiv [\delta\hat{\mathbf{p}}_{k+1}^{+T} \ \hat{\beta}_{k+1}^{+T} \ \hat{\kappa}_{k+1}^{+T} \ \hat{\varsigma}_{1,k+1}^{+T} \ \hat{\varsigma}_{2,k+1}^{+T}]^T$ . The error  $\delta\hat{\mathbf{p}}_{k+1}^{+T}$  is then converted to  $\delta\hat{\mathbf{q}}_{k+1}^{+T}$  using Eqs. (2.25a) and (2.25b) and the updated quaternion using

$$\hat{\mathbf{q}}_{k+1}^+ = \delta\mathbf{q}_{k+1}^+ \otimes \hat{\mathbf{q}}_{k+1}^-(0) \quad (5.23)$$

Finally,  $\delta\hat{\mathbf{p}}_{k+1}^+$  is reset to zero for the next propagation.

### 5.4.2 Extended Kalman Filter Alignment Calibration

The alignment filter is developed in a way similar to Ref. [1]. The continuous-time attitude kinematics equation includes calibration parameters as Eq. (5.3) but without the gyro asymmetric scale factor errors. In the EKF formulation, the sensor model sensitivity or observation sensitivity with respect to the states is needed. The observation sensitivity matrix of the vector star sensor with misalignments is given by [1] [5]

$$H_{st} = \begin{bmatrix} 2\hat{A}_{s,b}(\hat{\mathbf{q}})[\mathbf{r}_{st} \times] & \mathbf{0}_{3 \times 9} & [\hat{A}_{s,b} A(\hat{\mathbf{q}})\mathbf{r}_{st}] \times \end{bmatrix} \quad (5.24)$$

where  $\hat{A}_{s,b}$  is the estimated alignment matrix from body to true sensor coordinate. The observation sensitivity matrix of the payload sensor without misalignment is given by

$$H_p = [2A_{p,b}A(\hat{\mathbf{q}})[\mathbf{r}_p \times] \quad \mathbf{0}_{3 \times 9} \quad \mathbf{0}_{3 \times 3}] \quad (5.25)$$

## 5.5 Performance Evaluation

When the filter is tested with data generated from model, i.e. the “truth” is known, filter performance could be evaluated by simply taking the absolute difference between the true states and estimated states based on noisy measurements. This should converge to zero as the states converge. If real measurement data is used, we could take the absolute difference between the noisy sensor measurements and generated measurement (from estimated states). This is also referred to as “residuals”. If the plant model and sensor model accurately capture physical systems and linearization assumption hold true, the residuals are expected to be random Gaussian white noise. If the residuals show systematic errors, this implies further refinement of the filter system model (plant) may be required.

## **5.6 Summary**

The attitude and alignment filters based on the UF and EKF formulations are given in this chapter. We choose to perform absolute alignment calibration (using the gyros as the “reference” sensor) in the UF formulation and relative alignment with respect to the payload sensor in the EKF formulation.

# 6 Results

## 6.1 Introduction

The simulations are divided into different subcomponents. Various simulation parameters are changed to evaluate different scenarios that are accurate representations of real world applications. Alignment filters with the UF formulation developed in the previous chapter will be applied to simulated gyro and attitude sensor measurements. Results from the EKF alignment calibration also will be shown as a comparison. Lastly, telemetry from the Wilkinson Microwave Anisotropy Probe (WMAP) spacecraft will be used to demonstrate the real world application of the UF calibration scheme.

## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

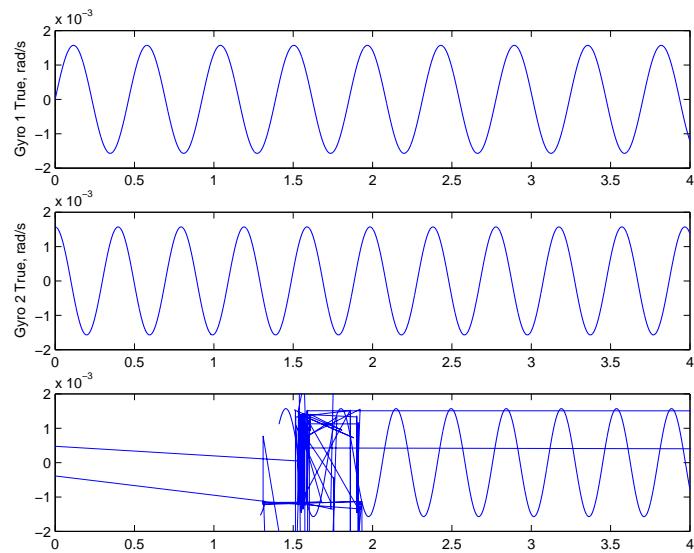
The simulations are performed on an Intel Pentium III 933 Mhz with 256MB RDRAM system. In our simulation, we assume no *a priori* knowledge of the calibration parameters. The simulation parameters used are:

- Star sensor misalignment:  $-100, -100, +100$  arc-s

- Payload sensor misalignment: 150, 150, -100 arc-s
- Initial gyro bias: 0.2, 0.3, 0.2 deg/hr for each axis
- Gyro misalignment:  $\delta_{xz} = \delta_{xy} = \delta_{yz} = 0, \delta_{zx} = 400, \delta_{zy} = 300, \delta_{yz} = 200$  arc-s
- Gyro symmetric scale factor error:  $\lambda_x = 500, \lambda_y = 500, \lambda_z = 500$  ppm
- Gyro asymmetric scale factor error:  $\mu_x = 100, \mu_y = 100, \mu_z = 100$  ppm
- Calibration maneuver: 0.09 des/s sinusoidal in each axis at (0.0006, 0.0007, 0.0008) Hz
- Gyro measurement is simulated with  $\sigma_u = 1.3036 \times 10^{-3} \mu rad/sec^{3/2}$  and  $\sigma_v = 1.45444 \mu rad/sec^{1/2}$
- Initial estimate of all calibration parameters are zero with standard deviation of 5 deg for attitude, 0.5 deg/hr for gyro bias, 500 arc-s for gyro misalignments, 500 ppm for scale factors, 500 arc-s for star sensor misalignments, 500 arc-s for payload sensor misalignments if used
- Star sensor accuracy: 5, 5, 5 arc-s

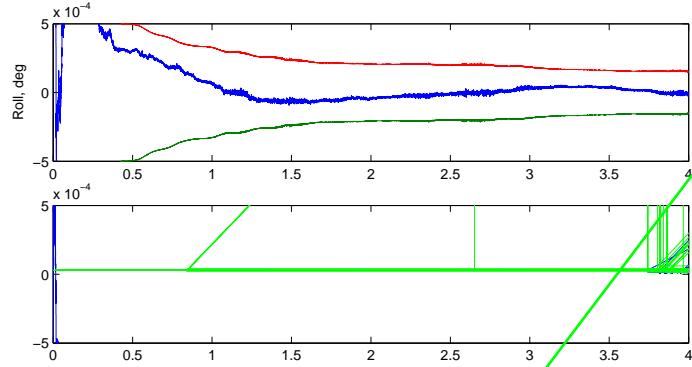
## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---

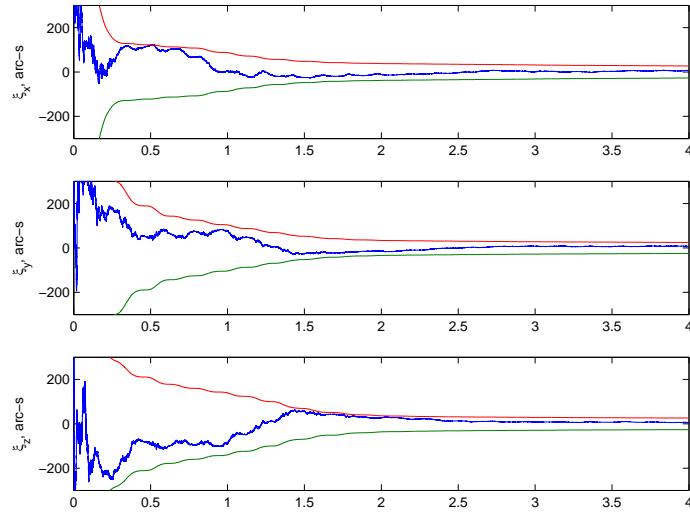


Figure 6.4: UF,  $\Delta t = 0.2$ , Gyros Nonorthogonal Misalignment Errors with  $3\sigma$  Bounds

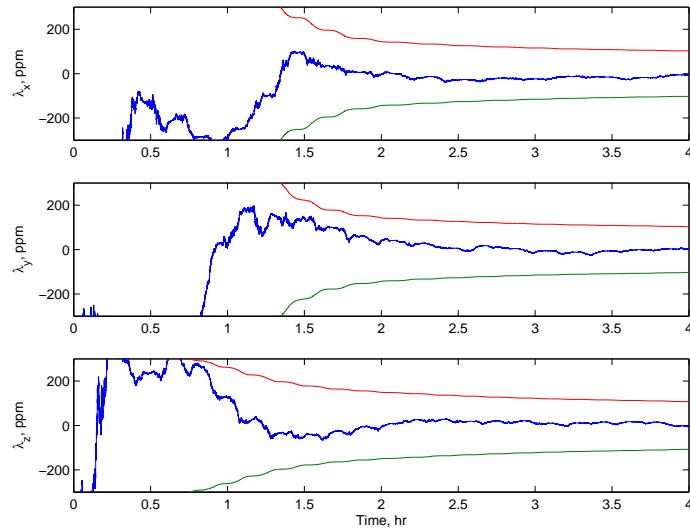


Figure 6.5: UF,  $\Delta t = 0.2$ , Gyros Symmetric Scale Factor Errors with  $3\sigma$  Bounds

## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---

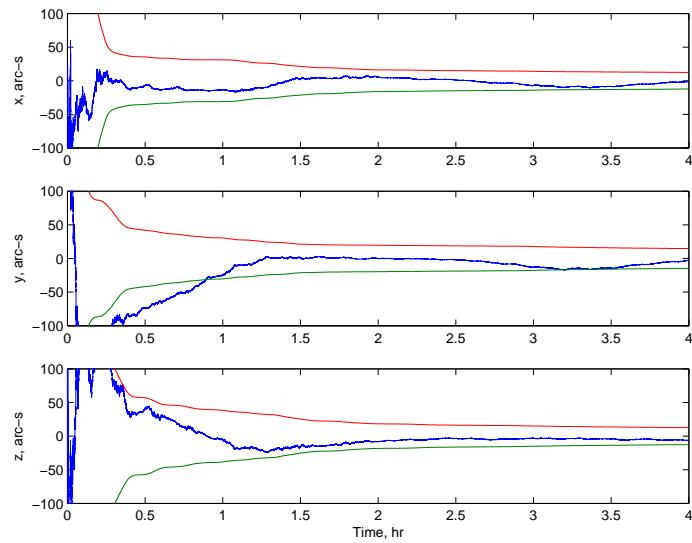
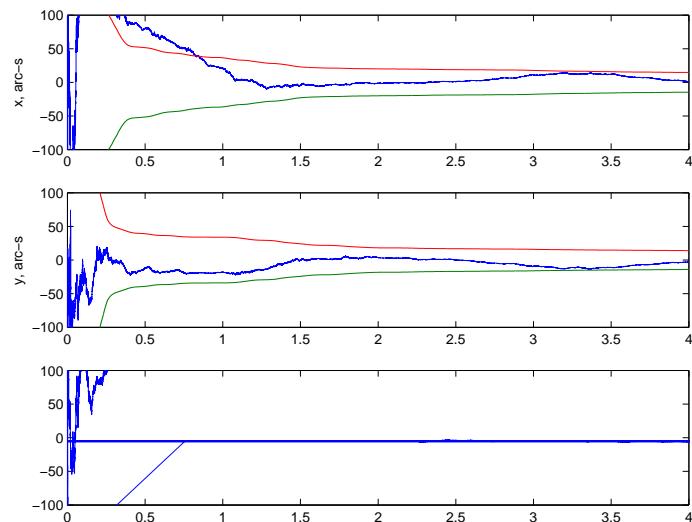


Figure 6.6: UF,  $\Delta t = 0.2$ , Star Sensor Misalignment Errors with  $3\sigma$  Bounds



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---

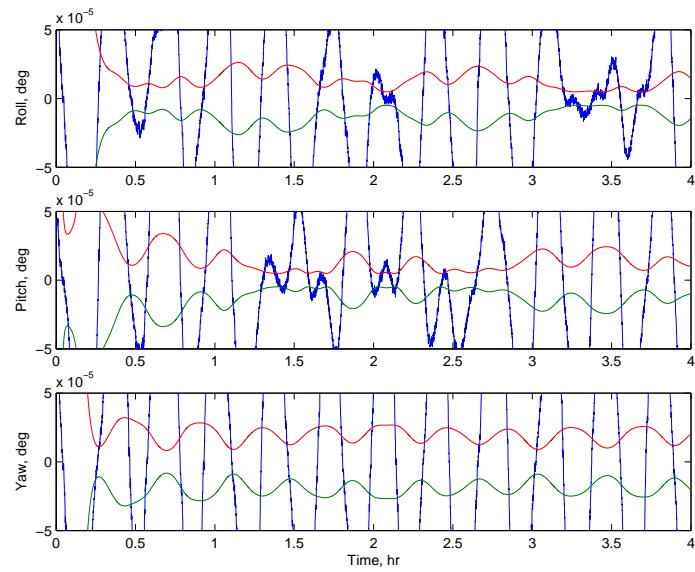
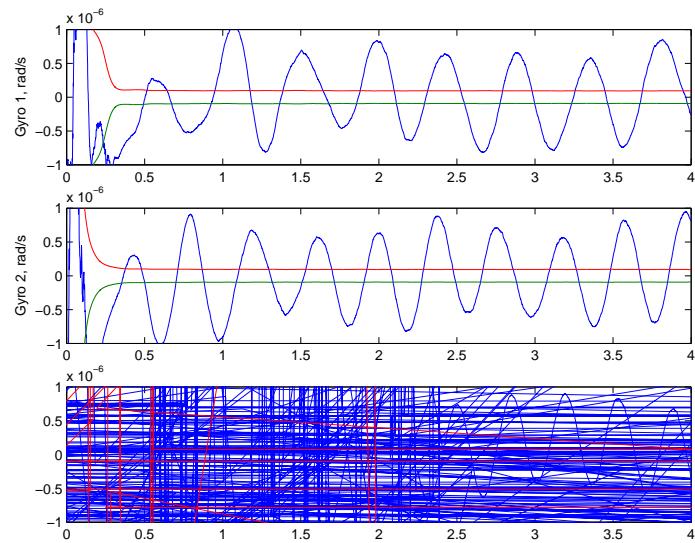
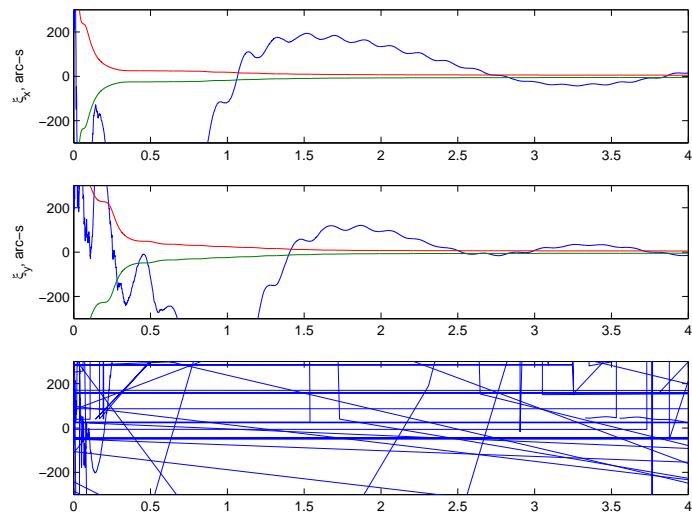


Figure 6.8: EKF,  $\Delta t = 2.0$ , Attitude Errors with  $3\sigma$  Bounds



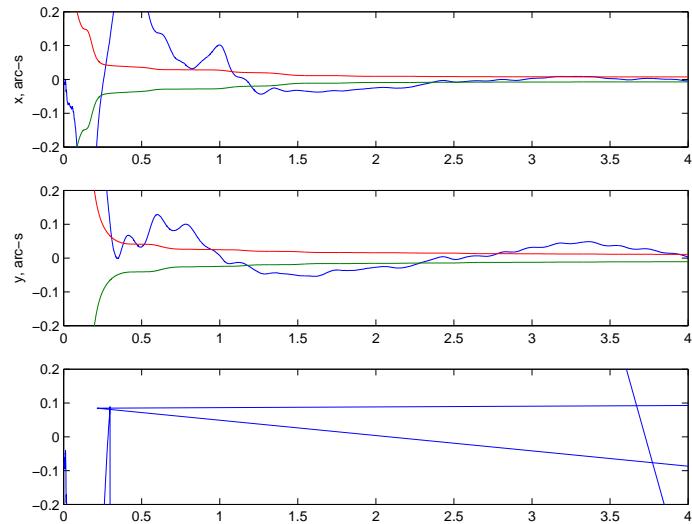
## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---

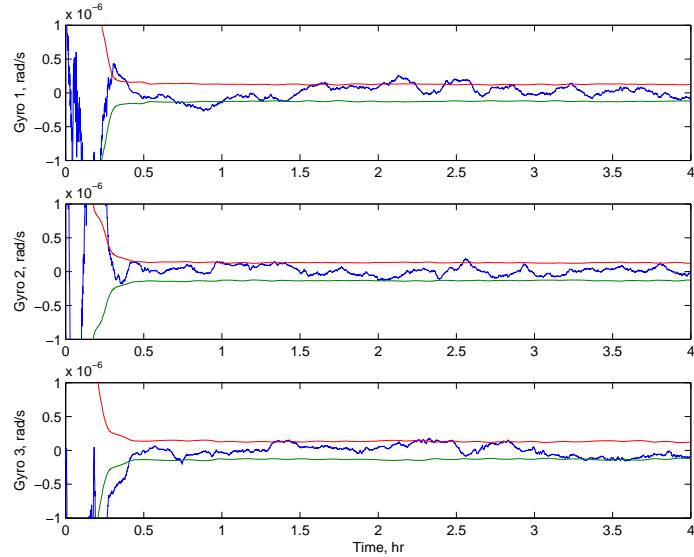


Figure 6.14: UF,  $\Delta t = 2.0$ , Gyros Bias Errors with  $3\sigma$  Bounds

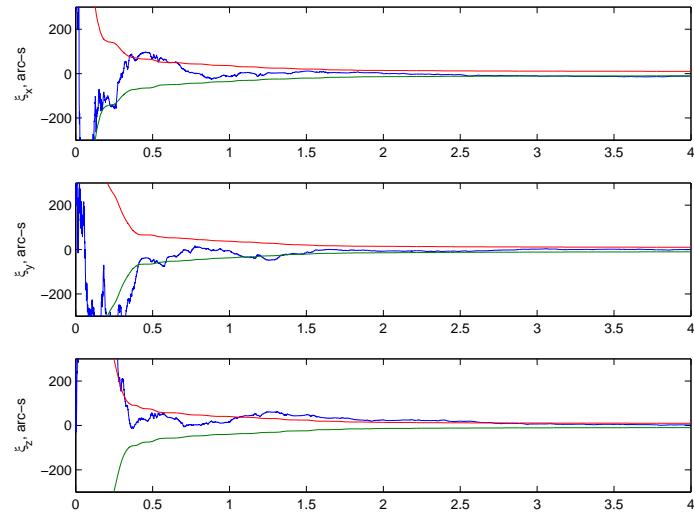


Figure 6.15: UF,  $\Delta t = 2.0$ , Gyros Misalignment Errors with  $3\sigma$  Bounds



## 6.2 Simulated Calibration Results with Extended Kalman Filter and Unscented Filter

---

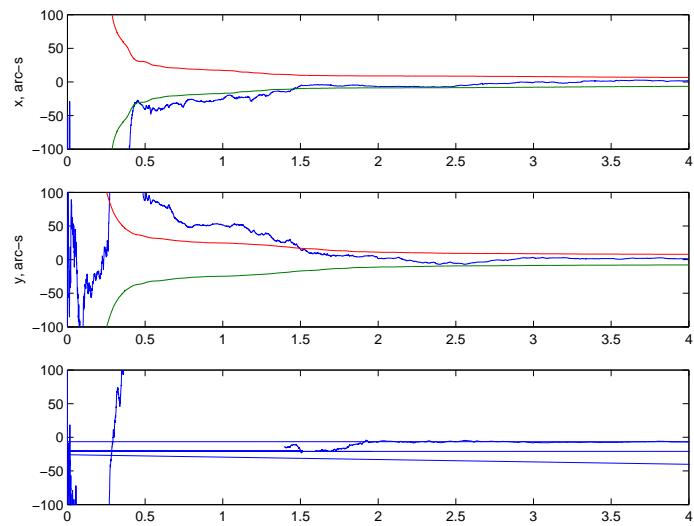


Table 6.1: Filter Interval and Completion Time

Filter	Filter Interval, sec	Completion Time, sec
UF	0.2	4231
UF	2.0	486
EKF	0.2	1141
EKF	0.5	472
EKF	1.0*	260
EKF	2.0*	120

\* filter diverges at this update interval

Table 6.1 shows the time required for a few simulation runs with the UF and EKF for relative performance comparisons. Clearly, the UF requires more computational power for the same filter update interval compared to the EKF. This is mainly due to the covariance decomposition. For the filter update interval of 0.2 sec, the filter convergence properties of UF and EKF are comparable. However, due to linearization, the EKF does not work well at larger filter update intervals. The UF is shown to be able to converge even at an update interval of 2 sec with slight sacrifice in the filter's performance. Compared to the EKF running at 0.2 sec interval, the UF running at 2.0 sec is even less computational intensive. Furthermore, the UF paves way to multiprocessing possibility that could further decrease the computational time. Lower filter update rates also lighten the burden on data telemetry and memory consumption in between update times. Figure 6.19 shows a plot of time required for various EKF simulation runs. The nonlinearity in time required may be attributed to the fact that the simulation consumed up most of the system resources at small interval runs. Therefore, in real world implementation, computational requirements should directly correspond to the size of the filter update interval.

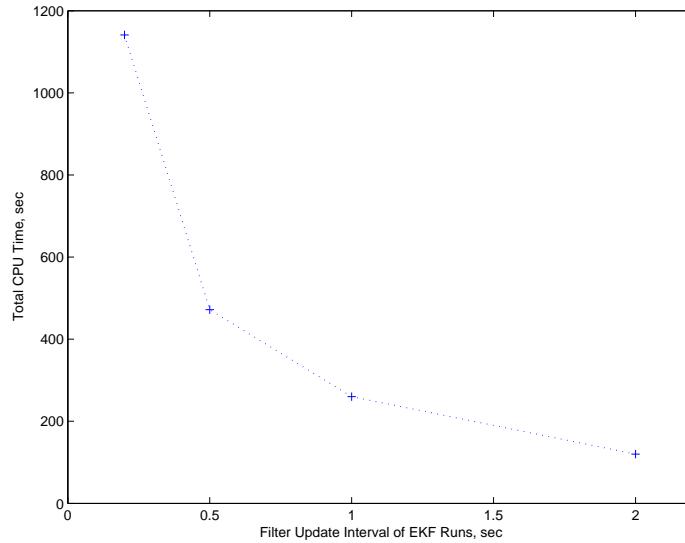


Figure 6.19: CPU Time of EKF Simulation Runs

### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

The Wilkinson Microwave Anisotropy Probe (WMAP or previously MAP) spacecraft was successfully launched at 3:46:46 p.m. EDT (1946:46 GMT) on June 30, 2001 from SLC-17B, Cape Canaveral, Florida, aboard a Boeing Delta 2 7425-10 launch vehicle and has an expected mission lifetime of 27 months. The 840 kg spacecraft is stationed at L2 Halo orbit about a million miles from the Earth in an anti-Sun configuration to allow full skyview of the universe and avoid interferences from the Earth.

WMAP is a successor to the COBE (Cosmic Background Explorer) mission managed by NASA - Goddard Space Flight Center to capture the background radiation of the universe. This two-year mission will capture infant light of the universe from 379,000 years after the Big Bang. If present universe was an 80-year-old person, WMAP captures the image the same person when he/she was barely 19 hours old. The results from WMAP would help

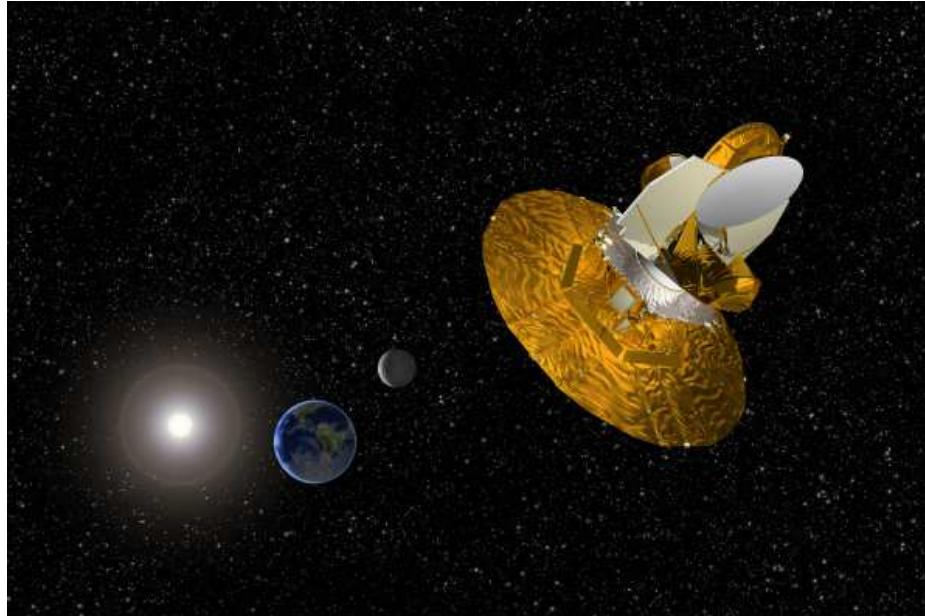


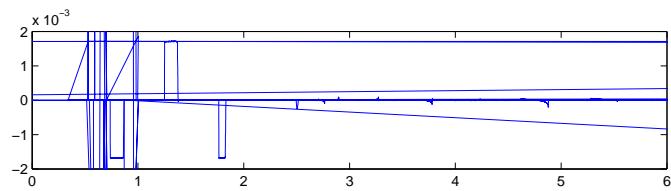
Figure 6.20: Artist Rendering of WMAP Leaving the Earth/Moon Toward the L2 Halo Orbit

scientists in determining the content, composition, shape, history and the ultimate fate of the universe. Some interesting recent findings from the WMAP showed that the age of the universe is 13.7 billion years old with a margin of error of close to 1% and that the universe is “flat”, which means that the high school geometry is valid in the universe.

The gyro data was collected from WMAP on July 1, 2001, while in L2 orbit. This data was then used to validate the applicability of the UF in alignment calibration. Figure 6.21 shows the data collected from the spacecraft. Figure 6.22 shows the estimated quaternion by the UF. Figure 6.24 to 6.27 show the converging behavior of the calibration parameters. This convergence follows the rotational profile shown in Figure 6.21 as the convergence is faster when the rate is higher. This is due to the increase of observability for the calibration parameters when the spacecraft is maneuvering. However, the covariances of gyro bias error estimation increase during maneuvers as shown in Figure 6.23. This is due to the

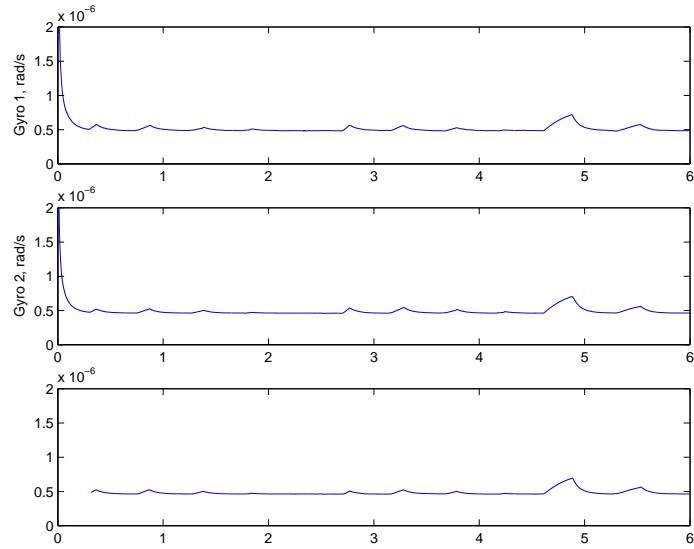
### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---



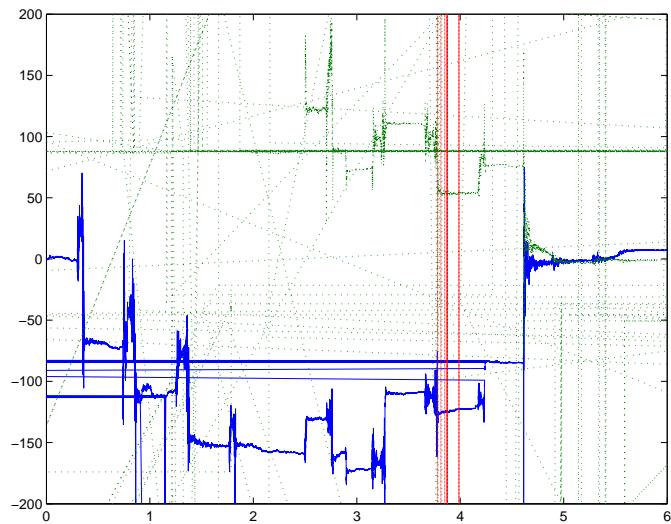
### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---



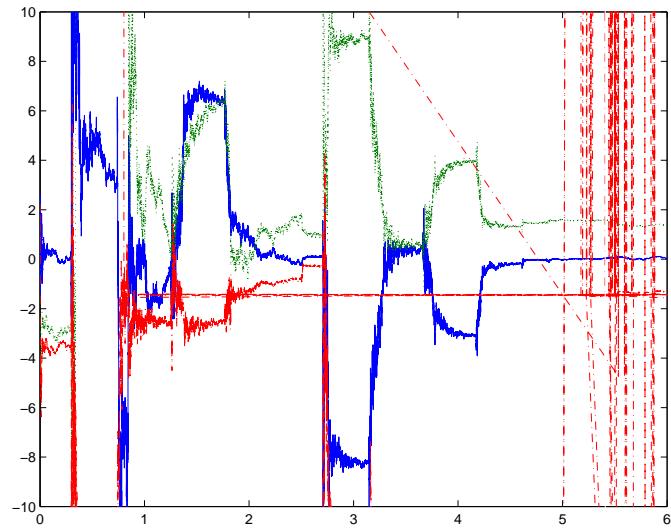
### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---



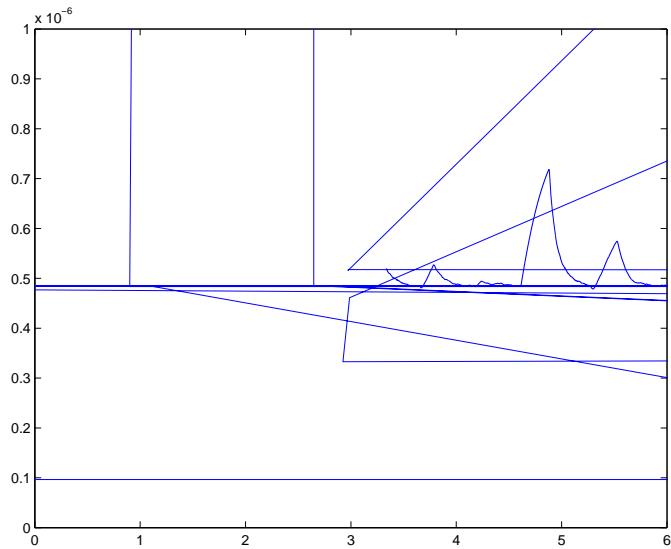
### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---



### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---



### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---

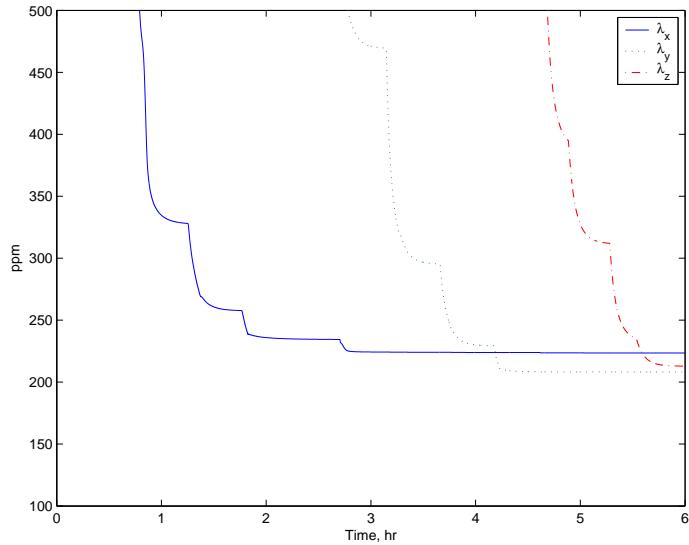


Figure 6.31: WMAP Gyros Symmetric Scale Factor Errors Estimation  $3\sigma$  Bounds

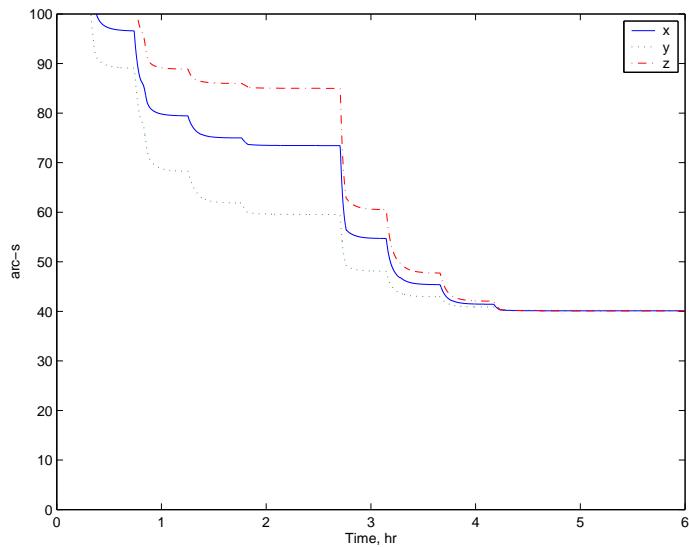


Figure 6.32: WMAP Star Sensor 1 Misalignments Estimation  $3\sigma$  Bounds

### 6.3 Calibration on Actual Attitude Telemetry Data Using Unscented Filter

---

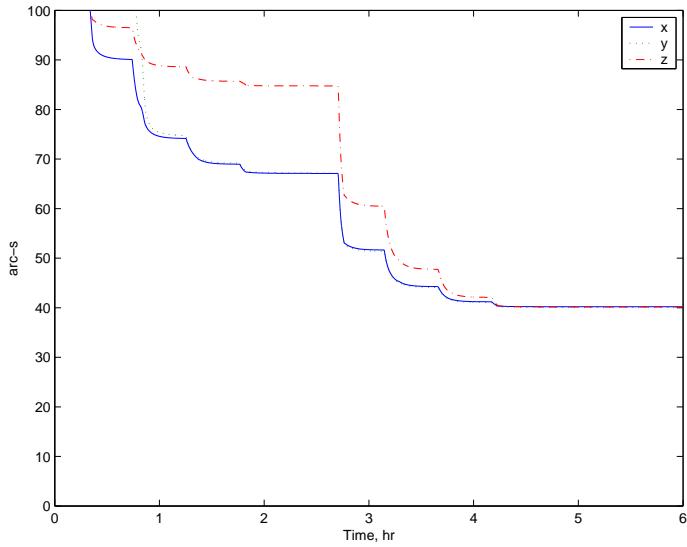
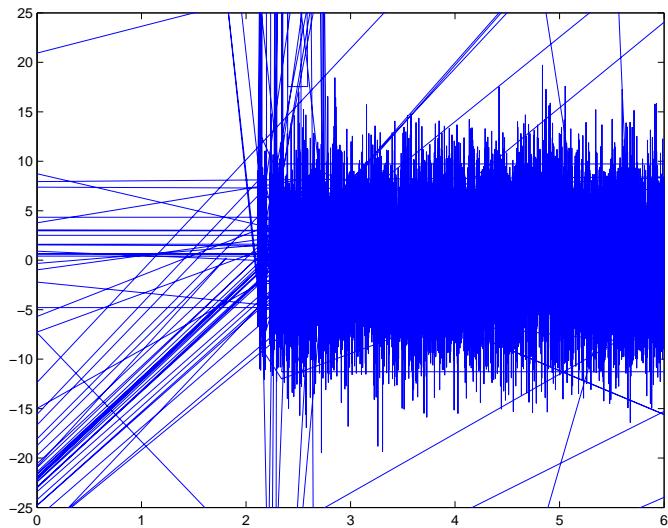


Figure 6.33: WMAP Star Sensor 2 Misalignments Estimation  $3\sigma$  Bounds



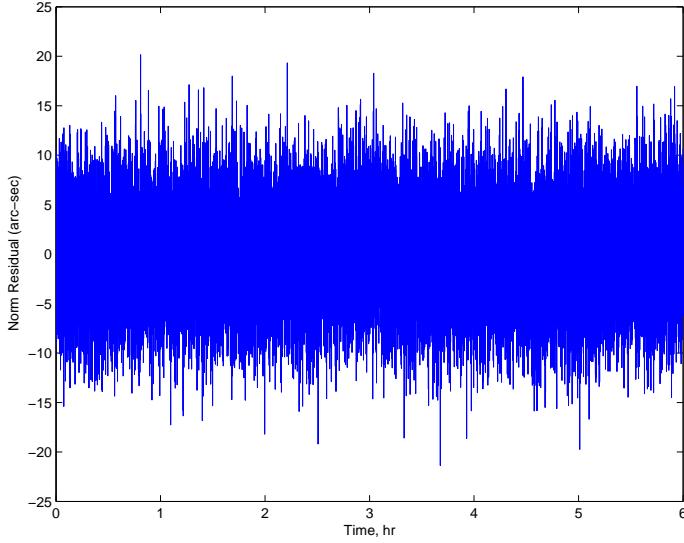


Figure 6.35: WMAP Star Sensor 2 Norm Residual Between Estimated Observation and Sensor Measurement

dominance of nonlinearity at fast maneuvers. Figures 6.28 to 6.33 show the error covariances of attitude and calibration parameters estimation with corresponding  $3\sigma$  bounds. As expected, residual plots of both star sensors shown in Figures 6.34 and 6.35 respectively could be viewed as zero-mean Gaussian noise. This reinstates the UF's performance in real world implementation.

## 6.4 Summary

Calibrations are performed with both the EKF and UF. The UF attitude estimation with calibration parameter estimation is shown to be more robust than the EKF, especially with larger filter update intervals. The UF requires about 2-4 times more computational power than the EKF with the same update interval. However, the performance of UF running at a higher update interval is comparable to the EKF at a smaller update interval, while the UF in this case runs at comparable or faster speed than the EKF. Furthermore, the EKF

diverges at the higher UF update interval due to violation of the lower order linearization assumption in the EKF.

Calibration also performed with gyro telemetry data from the WMAP spacecraft and showed encouraging results. However, the convergence capability of the WMAP data is not as good as the simulated data from sinusoidal rotational profile of the simulated data. Maneuvering spacecraft increases observability of the calibration parameters but introduces nonlinearity into the attitude dynamics. Thus the first-order assumption of the Taylor series expansion in EKF formulation is easily violated and leads to the divergence of the filter at higher filter measurement update interval.

For the same simulation environment, the EKF works well only up to filter update interval of 0.5 sec while UF still able to converge reasonably at interval of 4 sec. Although the UF is slower than the EKF, the performance enhancement of the UF over the EKF may outweighs the increased computational costs. For all these reasons, the UF is more favorable for real-time implementation.

# 7 Conclusion and Future Work

## 7.1 Conclusions

Proper alignment calibration is of prime importance. It reduces fault detection and correction (FDC) that would cause an inadvertent trip into Sun Acquisition or Safehold mode and disrupt nominal spacecraft mission. The UF alignment calibration proved to be more robust than the EKF alignment. The UF alignment filter works even at larger sampling intervals due to higher order approximations than the EKF. Prolonged loss of signal from attitude sensors could easily cause the divergence in the EKF. Furthermore, the ability to the UF to yield reasonable results in contrast of the divergence of the EKF during fast spacecraft maneuvers as shown in the simulation favors its on-board implementation.

## 7.2 Future Work

Calibration is essentially identifying parameters associate with a model. How accurate is a model? We may never know. Different calibration models could yield different results. For example, RWA calibration of EUVE in October 1997 using “motor mode” and “generator mode” yielded different results [10]. Whiteness tests on the residuals between actual measurements and estimated quantities could help to verify the models [19]. Thus here rises a possible “general” calibration approach. Nonphysical model of higher order, for example, polynomial functions, could potentially offer lower residuals or error covariance,

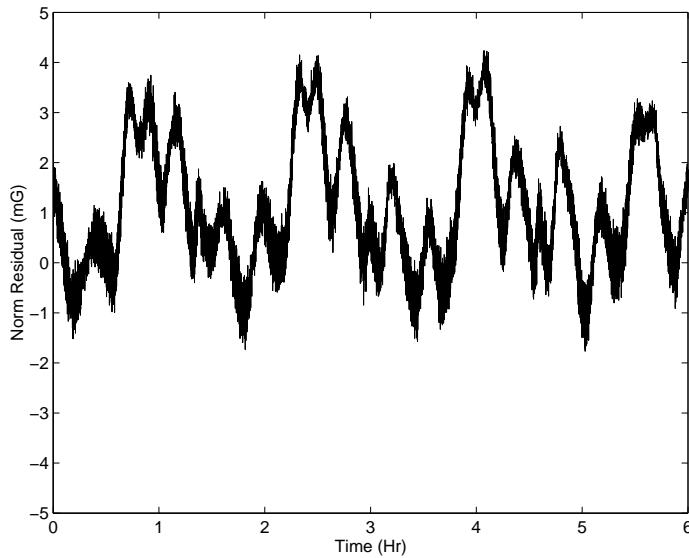


Figure 7.1: Norm Residual Between Body Estimates and Reference Vectors

especially on hard-to-model systems or when the process noise is too high. Reference [38] is an example that a black-box nonlinear model generated without knowledge of the system structure delivered better calibration results for the Hubble Space Telescope. Furthermore, the black-box model could be generated almost on-the-fly without in depth understanding of the physical systems. A disadvantage of this approach also raised in this paper is it may require extensive data sets for model training. Also, it tends to have poor extrapolation properties. Furthermore, if good extrapolation properties are important, this approach could be used to obtain sensor statistical properties for physical model based estimation methods.

Three Axis Magnetometers (TAMs) have been widely implemented in most recent space-craft missions for cheap, robust attitude estimation solutions, especially to Low Earth Orbit (LEO) satellites. However, due to contamination of the magnetic fields around the space-craft (for example from magnetic torque bar, or onboard electronics) they must be compen-

sated the magnetometer measurement. Also, non-Gaussian behavior of the magnetometer measurement either due to reference model error or disturbances in magnetic field proved to be challenging for magnetometer calibration. One quick solution is to include the estimation of the biases [9]. Various magnetometer calibration schemes exist. The more promising ones are attitude-independent methods since attitude estimation accuracy does not affect the calibration results. Reference [39] shows a real-time attitude-independent calibration scheme for TAM with the EKF and UF. An investigation into the residuals between the norm of the estimated vector, using the calibrated parameters, and the geomagnetic-reference vector shows some interesting non-Gaussian components that violates the assumption of most filters, as shown by Figure 7.1. This clearly shows the nonlinear and non-Gaussian nature in magnetic field model.

## Bibliography

- [1] Leffert, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sept.-Oct. 1982.
- [2] Ward, D. K., Davis, G. T., and Donnell, Jr, J. R. O., "The Microwave Anisotropy Probe Guidance, Navigation and Control Hardware Suite," *AIAA GN&C Conference & Exhibit*, Monterey, CA, Aug. 2002, AIAA-2002-4579.
- [3] Andrews, S. F. and Bilanow, S., "Recent Flight Results of the TRMM Kalman Filter," *AIAA GN&C Conference & Exhibit*, Monterey, CA, Aug. 2002, AIAA-2002-5047.
- [4] Andrews, S. F. and Morgenstern, W. M., "Design, Implementation, Testing, and Flight Results of the TRMM Kalman Filter," *Proceedings of the AIAA GN&C Conference & Exhibit*, Boston, MA, Aug. 1998, AIAA-1998-4509, pp. 1911–1918.
- [5] Pittelkau, M. E., "Kalman Filtering for Spacecraft System Alignment Calibration," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, Nov.-Dec. 2001.
- [6] Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," Vol. 3068 of *Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, Florida, April 1997, pp. 477–482.

- [7] Shuster, M. D., Pitone, D. S., and Bierman, G. J., "Batch Estimation of Spacecraft Sensor Alignments I. Relative Alignment Estimation," *Journal of the Astronautical Sciences*, Vol. 39, No. 4, Oct.-Dec. 1991.
- [8] Hartley, J. B. and Hughes, P. M., "Automation of Satellite Operations: Experiences and Future Directions at NASA GSFC," *Third International Symposium on Ground Data Systems for Space Mission Operations*, Munich, Germany, Sept. 1996, SO96.8.007.
- [9] Andrews, S. F. and Morgenstern, W. M., "Initial Flight Results of the TRMM Kalman Filter," *AAS/GSFC 13th International Symposium on Space Flight Dynamics*, NASA-Goddard Space Flight Center, Greenbelt, MD, May 1998, pp. 345–357.
- [10] Bauer, R., Smith, C., and Nevitt, R., "EUVE Reaction Wheel Assembly Calibration for Improved Spacecraft Slews," *AAS/GSFC 13th International Symposium on Space Flight Dynamics*, NASA-Goddard Space Flight Center, Greenbelt, MD, May 1998, pp. 871–885.
- [11] Belur, S. V. and Harman, R., "Calibration of Gyros with Temperature Dependent Scale Factors," *2001 Flight Mechanics Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, June 2001, pp. 163–173.
- [12] Hall, C. D., *Spacecraft Attitude Dynamics and Control*, Lecture Notes for AOE 4140 - Spacecraft Dynamics and Control at Virginia Tech.
- [13] Goldstein, H., *Classical Mechanics*, Addison-Wesley Publishing Company, Reading, MA, 2nd ed., 1980.
- [14] Schaub, H. and Junkins, J. L., *Analytical Mechanics of Aerospace Systems*, AIAA Educational Series, 2003.

- [15] Shuster, M. D., “A Survey of Attitude Representations,” *Journal of the Astronautical Sciences*, Vol. 41, No. 4, Oct.-Dec. 1993, pp. 439–517.
- [16] Hamilton, S. W. R., *Elements of Quaternions*, Longmans, Green and Co, London, 1866.
- [17] Wertz, J. R., editor, *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, Holland, 1978.
- [18] Kuipers, J. B., *Quaternions and Rotation Sequences*, Princeton University Press, Princeton, 1999.
- [19] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, CRC Press, in progress, Boca Raton, FL, 2003.
- [20] Markley, F., “Attitude Estimation or Quaternion Estimation,” *The John L. Junkins Astrodynamics Symposium*, College Station, TX, May 2003, AAS-03-264.
- [21] Stuelpnagel, J., “On the Parameterization of the Three-Dimensional Rotation Group,” *SIAM Review*, Vol. 6, No. 4, 1964, pp. 422–430.
- [22] Crassidis, J. L., “An Omnibus of Quaternion Conundrums and Incantations,” *AIAA/AAS Astrodynamics Specialist Conference*, Denver, CO, August 2000, pp. 294–303.
- [23] Kalman, Rudolph, E., “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME-Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.
- [24] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., “A New Approach for Filtering Nonlinear Systems,” *Proceedings of the American Control Conference*, Seattle, Washington, June 1995, pp. 1628–1632.

- [25] Wan, E. A. and van der Merwe, R., “The Unscented Kalman Filter for Nonlinear Estimation,” in *Proc. of IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, October 2000.
- [26] Wan, E. and van der Merwe, R., “The Unscented Kalman Filter,” *Kalman Filtering and Neural Networks*, edited by S. Haykin, chap. 7, Wiley, 2001.
- [27] Julier, S. J., “The Scaled Unscented Transformation,” *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, pp. 1108–1114.
- [28] Shuster, M. D., “Maximum Likelihood Estimation of Spacecraft Attitude,” *The Journal of the Astronautical Sciences*, Vol. 37, No. 1, Jan.-Mar. 1989, pp. 79–88.
- [29] Farrenkopf, R. L., “Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators,” *Journal of Guidance and Control*, Vol. 1, No. 4, July-Aug. 1978, pp. 282–284.
- [30] Pittelkau, M. E., “Everything is Relative in System Alignment Calibration,” *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Denver, CO, Aug. 2000 AIAA-2000-4246, pp. 361–368.
- [31] Challa, M. and Natanson, G., “Effects of Magnetometer Calibration and Maneuvers on Accuracies of Magnetometer-Only Attitude-and-Rate Determination,” *Proceedings of the AAS/GSFC 13th International Symposium on Space Flight Dynamics*, Vol. 1, (NASA/CP-1998-206858/VOL1) NASA-Goddard Space Flight Center, Greenbelt, MD, May 1998, pp. 389–401.
- [32] Shuster, M. D., “A Broad Look at Deterministic Three-Axis Attitude Determination,” *AIAA/AAS Astrodynamics Specialist Conference*, Denver, CO, August 2000, pp. 294–303.

- [33] Brasoveanu, D. and Hashmall, J., “Spacecraft Attitude Determination Accuracy From Mission Experience,” *Proceedings of the Flight Mechanics/Estimation Symposium*, (NASA/CP-3265) NASA-Goddard Space Flight Center, Greenbelt, MD, May 1994, pp. 153–166.
- [34] Markley, F., “How to Estimate Attitude from Vector Observations,” *AIAA/AAS Astrodynamics Specialist Conference*, Girwood, AK, 1999.
- [35] Markley, F., “Attitude Determination Using Vector Observations and Singular Value Decomposition,” *Journal of the Astronautical Sciences*, , No. 36, pp. 245–258.
- [36] Markley, F., “Attitude Estimation from Vector Observations: A Fast Optimal Matrix Algorithm,” *Journal of the Astronautical Sciences*, Vol. 41, No. 2, 1993, pp. 261–280.
- [37] Crassidis, J. L. and Markley, F. L., “Unscented Filtering for Spacecraft Attitude Estimation,” *AIAA GN&C Conference & Exhibit*, Austin, TX, Aug. 2003 AIAA-2003-5484.
- [38] Barford, L., Tufillaro, N., Usikov, D., Marochnik, L., and McCutcheon, R., “Calibration of Hubble Space Telescope Focal-Length Variations Using the Embedding Technique,” *2001 Flight Mechanics Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, June 2001, pp. 485–495.
- [39] Crassidis, J. L., Lai, K.-L., and Harman, R. R., “Real-Time Attitude-Independent Three-Axis Magnetometer Calibration,” *2003 Flight Mechanics Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, October 2003.
- [40] Golub, G. H. and Van Loan, C. F., *Matrix Computations*, The John Hopkins University Press, Baltimore, MD, 2nd ed., 1989.

## A Matrix Inversion Lemma

This section will present the result derived by Sherman-Morrison-Woodbury [40]. Let

$$F = (A + BCD)^{-1} \quad (\text{A.1})$$

where

$F$  = an arbitrary  $n \times n$  matrix

$A$  = an arbitrary  $n \times n$  matrix

$B$  = an arbitrary  $n \times m$  matrix

$C$  = an arbitrary  $m \times m$  matrix

$D$  = an arbitrary  $m \times n$  matrix

Assuming all the inverses of  $F, A, B, C$  and  $D$  exist, then

$$F = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \quad (\text{A.2})$$

which can be easily proved by showing that  $F^{-1}F = I$ . From inspection, this lemma does not seem to help much in terms of computation requirements or simplifying the original equation. However, for most applications, the subset,  $m$ , is usually smaller than  $n$  and Eq. (A.2) leads to a reduction in the computational requirement.

## B Matlab Code

This appendix includes the Matlab codes used for attitude and misalignments estimation with UF. Below are the Matlab filenames with their respective functions:

Filename	Functions
cross.m	generates a cross matrix from a $3 \times 1$ vector
delta_q.m	finds the quaternion error from two quaternions
dp2dq.m	converts GRPs to quaternions
e2q.m	transforms Euler angle to quaternions
err.m	finds the error between the estimates and the “truth”
m_fun.m	generates the “M” matrix as in the gyro model formulation
mea_gyro	simulates gyro measurements
mea_payload.m	simulates vector measurements of a payload sensor
mea_star2.m	simulates unit vector measurements of a star tracker
om.m	propagates discrete-time attitude quaternion
omega_g.m	generates the “ $\Omega_g$ ” matrix as in the formulation
parameters.m	sensors statistics and true calibration parameters
q2att_mat.m	generates an attitude matrix from a quaternion

<b>Filename</b>	<b>Functions</b>
q_mult.m	performs quaternion multiplication
run.m	The main algorithm that sequentially estimate the current attitude and calibration parameters
state_fun.m	function to propagate the true attitude quaternion and gyro biases
truth.m	propagates the truth and generates measurements
uf.m	estimates current attitude and performs sensors calibration with the UF
xi.m	generates the “ $\Xi$ ” matrix