Copyright

by

Renato Zanetti

2007

The Dissertation Committee for Renato Zanetti
certifies that this is the approved version of the following dissertation:

Advanced Navigation Algorithms for Precision Landing

Committee:

Robert H. Bishop, Supervisor

David G. Hull

Maruthi R. Akella

Cesar Ocampo

Timothy P. Crain II

Advanced Navigation Algorithms for Precision Landing

by

Renato Zanetti, Ing.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2007

To my wife

Acknowledgments

First and for most I would like to thank my parents. Without their help and support I would not be who I am today, thank you.

The hard part of graduate school is not getting in, is leaving. I want to thank all the graduate students that made my experience so wonderful. I could write a page full of names, but I will limit myself to five. Thanks to my qualifying exam studying partners Christina Chomel, Chad Hanak, and Jeremy Rea. It is not possible to make the qualifying exams a pleasant experience, your company, help, and support made them at least tolerable. Thanks to my officemate and training partner Jorge Alvarez. Thanks to Kyle DeMars for his help and for many interesting conversations.

Thank you to my advisor Dr. Bishop for his lead and for giving me wonderful projects to work on.

Thank you to my wonderful wife, for so many reasons that it would take me more than the 183 pages of this dissertation to list them all.

Renato Zanetti

The University of Texas at Austin
December 2007

Advanced Navigation Algorithms for Precision Landing

Publication No. ____________________

Renato Zanetti, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Robert H. Bishop

A detailed analysis of autonomous navigation algorithms to achieve autonomous precision landing is presented. The problem of integrated attitude determination and inertial navigation is solved. The theoretical results are applied and tested in three different applications. Optimality conditions for constrained quaternion estimation using the Kalman filter are derived.

It is common in spacecraft applications to separate the attitude determination from the inertial navigation system. While this approach has worked in the past, it inevitably degrades the navigation performance when the correlations between the two systems are not correctly accounted for. It is shown how to optimally include an attitude determination algorithm into the Kalman filter. When the conditions to achieve optimality are not met, it is shown how to achieve sub-optimality by properly accounting for the correlation.

The traditional approach to inertial navigation is to employ the inertial measurement unit (IMU) outputs to propagate the estimated states forward in time, rather then use them to update the state. A detailed covariance analysis of dead-reckoning Mars entry navigation is performed. The contribution of various sources of IMU errors are explicitly accounted for and the filter performance is validated through Monte Carlo analysis.

The drawback of dead-reckoning is that this approach prevents the inertial measurements from reducing the uncertainty of the estimated states. While this shortcoming can be compensated by the availability of other measurements, it becomes crucial when the IMU is the only sensor to provide measurements. Such a situation arises, for example, during Mars atmospheric entry. In the second application of this work, IMU measurements from a NASA mission are processed in an extended Kalman filter, and the results are compared to dead-reckoning. It is shown that is possible to reduce the uncertainty of the inertial states by filtering the IMU.

The final application is lunar descent to landing navigation. In this example the IMU is filtered and the algorithms to include an attitude estimate into the Kalman filter are tested. The design performance is confirmed by Monte Carlo analysis.

Contents

Acknowledgments **v**

Abstract **vi**

Chapter 1 Introduction **1**

- 1.1 Contributions of the Dissertation 6
- 1.2 Organization of the Dissertation 7

Chapter 2 Preliminary Notions **9**

- 2.1 Attitude Representations, Kinematics, and Dynamics 9
  - 2.1.1 Attitude Representation 10
  - 2.1.2 Attitude Kinematics 15
  - 2.1.3 Attitude Dynamics 19
- 2.2 Quaternion Estimation in the Kalman Filter 19
  - 2.2.1 Existing Methods of Including Constraints in the Kalman Filter 20
  - 2.2.2 Quaternion Estimation Errors: Additive and Multiplicative . 22
  - 2.2.3 Relationship Between Additive and Multiplicative Error Representations 26
  - 2.2.4 Normalization 32
  - 2.2.5 Norm Constrained Kalman Filtering 35

2.2.6 Conclusions 45

2.3 Davenport Solution to the Wahba Problem 45

2.3.1 QUEST Covariance Analysis 50

2.3.2 Summary 58

2.4 Single Layer Gating Network 58

Chapter 3 Theoretical Foundations 61

3.1 Introduction of Attitude Estimation in Spacecraft Navigation 61

3.1.1 Uncorrelated Sub-filter 62

3.1.2 Star-Tracker Implementation 67

3.1.3 Correlated Sub-Filter and Attitude Sub-Filter Implementation 72

3.1.4 Parallel Filters 76

3.2 Kalman Filter with Uncompensated Biases 79

3.2.1 Discrete Kalman Filter with Uncompensated Biases 80

3.2.2 Continuous Kalman Filter with Uncompensated Biases 87

3.3 Proposed Gating Network 91

Chapter 4 Dead-Reckoning Entry Navigation 95

4.1 Models 97

4.1.1 Dynamics Modeling 98

4.1.2 IMU Error Model 98

4.2 Dead-Reckoning Continuous IMU Measurements 99

4.2.1 Attitude estimation errors 101

4.2.2 Position and velocity estimation errors 103

4.2.3 Estimation error covariance 107

4.2.4 Dead Reckoning Navigation 108

4.2.5 Simulation Results 110

4.2.6 Generating the Accelerometer and Gyro Noises 115

4.2.7 Alternative Approach . . . 116
4.3 Dead-Reckoning Discrete IMU Measurements . . . 117
4.3.1 Attitude estimation errors . . . 120
4.3.2 Position and velocity estimation errors . . . 123
4.3.3 Dead Reckoning Navigation . . . 129
4.3.4 Simulation Results . . . 130
4.4 Conclusions . . . 135

Chapter 5 Adaptive Entry Navigation 137
5.1 Partitioning the Kalman Filter State . . . 139
5.2 Single Extended Kalman Filter . . . 142
5.2.1 Filter Model . . . 143
5.2.2 Atmosphere Model . . . 145
5.2.3 IMU Measurements . . . 146
5.2.4 Measurement Model . . . 146
5.2.5 Filter Summary . . . 147
5.2.6 Simulation Results . . . 148
5.3 Gating Network . . . 150
5.4 Conclusions . . . 152

Chapter 6 Lunar Descent Navigation 153
6.1 Star Camera Model . . . 155
6.2 Extended Kalman Filter . . . 158
6.2.1 Propagation . . . 159
6.2.2 Update . . . 160
6.3 Estimation Error Results . . . 162
6.4 Monte Carlo Analysis . . . 166

Chapter 7 Conclusions **170**

7.1 Future Directions 171

Bibliography **173**

Vita **183**

Chapter 1

Introduction

With the renewed objective of landing men on the Moon as a step to Mars human exploration, the engineering challenge to safely navigate a spacecraft to touchdown on a distant planet has taken center stage. This work is concerned with autonomous navigation of spacecraft performing precision landing. The importance of this study follows from the fact that no mission to date has landed on a distant planet aided by an autonomous navigation system, therefore the need for research in the field is large.

The optimal approach to estimate the spacecraft state (position, velocity, and attitude) is an integrated single estimator, such as the Kalman filter [1–3], or its nonlinear (non-optimal) extension, the extended Kalman filter (EKF) [4]. However, an integrated approach has not been implemented in space missions to date. Space-rated computers with the required computational capability have not been available. But modern sensors have some computational capability and can share the load. For example, a star tracker can identify stars and compute its own attitude in inertial space. It is therefore easier for the central filter to receive an attitude estimate from the star tracker rather than all the raw measurements. While at first glance this appears to be a viable approach, it leads to suboptimal performance of the overall navigation solution. As more powerful computers find their way into service, it is prudent to evaluate the options – what navigation process would be most optimal in terms of landing precision.

An integrated attitude and translation (position and velocity) estimator requires that we consider the nature of the group of rotations in three dimensions, $SO(3)$. Being that $SO(3)$ is not a vector space adds complexity to the process of estimating attitude in a Kalman filter. The fact that no three-dimensional representation of attitude can be globally continuous and non-singular [5] makes it desirable to introduce a higher dimensional representation that results in a constrained state. If the attitude is represented through the quaternion [6] (a parametrization with one redundant parameter) then the constraint is given by the unitary norm of the attitude quaternion. However, the Kalman filter algorithm does not naturally permit the introduction of constraints. So, during the update stage of the estimate process, the attitude quaternion estimate can violate the unitary norm constraint. There will be more discussion of this situstion in the subsequent sections. To avoid poor performance, the constraint should be included in the filter [7, 8]. Modifications to the EKF to estimate the quaternion include the additive EKF [9], the multiplicative EKF [10], and the rotational EKF [11]. This three approaches retain the basic structure of the EKF, relying on linearization to estimate the quaternion.

Other classes of attitude estimation algorithms operate directly on the nonlinear structure of the problem. The Davenport-q algorithm [12] is a nonlinear least-squares solution. While the Davenport-q algorithm is deterministic, it was shown to be a maximum likelihood solution under specific assumptions on the distribution of the measurements [13]. Other deterministic approaches exist, such as TRIAD [14], that determines the rotation matrix directly. Also, nonlinear observers have been investigated [15, 16]. These nonlinear attitude determination algorithms are not easily augmented to include position and velocity states (one such example is Extended QUEST [17]).

The designer is therefore left to choose between estimating the attitude through a linearized approach using a single filter that optimally accounts for the correlation between attitude and other states, or to employ a nonlinear attitude determination algorithm that will decentralized the estimation effort, and by doing so possibly loose optimality. In this dissertation, a method is developed to incorporate the estimate from an attitude determination algorithm in an integrated filter, such as the EKF.

The inertial measurement unit (IMU) is one of the most common navigation aids in aerospace applications. The IMU is composed of accelerometers and a gyros. The accelerometer measures non-field acceleration and the gyro measures rotation rates. These two measurements are often referred to as internal measurements, as opposed to external measurements for which the sensor interacts with the external environment. The traditional approach to navigate the IMU is to employ an algorithm that dead-reckons the IMU outputs of acceleration and attitude rates, i.e. the IMU measurements are used to propagate the spacecraft state (position, velocity, and attitude) using a numerical integration algorithm and a model of gravity. Dead-reckoning is characterized by a dynamic model in which only the gravitational acceleration is present; the attitude dynamics are measured, not modeled. In the case of Mars entry, descent, and landing (EDL), for example, aerodynamic forces and torques are not modeled. Dead-reckoning is a sensible navigation strategy when the aerodynamic models have large uncertainties at the same time time that the IMU hardware is capable of accurately providing measurements of the non-gravitational accelerations. Fortunately, knowledge of the Mars atmosphere has improved over time thanks to data collected from various successful planetary exploration missions. Nevertheless, the lack of predictability of the atmosphere makes the task of modeling the aerodynamic forces challenging. In lunar descent navigation, dead-reckoning the

IMU implies that the thrusters forces and torques need not to be modeled, making the model-based navigation algorithm less complex.

Despite the challenges of processing the IMU data in a model-based navigation algorithm, there are valid reasons for considering abandoning the dead-reckoning approach during the EDL phase of spacecraft navigation. First, the model-based approach provides the ability to accurately navigate through data drop-outs. Although thought to be an unusual event, IMU data dropouts can occur, leading to large state estimation errors that can be mitigated with a model-based approach. Second, the Kalman filtering approach naturally provides a state estimation covariance that accurately represents the state uncertainty, thereby leading to superior estimation accuracies once other external sensors become available, notably the altimeter. Finally, in the case of high uncertainty of Mars atmosphere, if a properly configured filter bank is employed in a multiple-model adaptive estimation (MMAE) architecture, changes in the atmosphere can be detected and accounted for.

It is of fundamental importance to correctly incorporate the correlation of attitude errors with position and velocity errors for precision navigation. A covariance analysis is used to quantify the dead-reckoning performance. Dead-reckoning Mars navigation is also compared to filtering IMU measurements from a NASA Mars landing mission. An extended Kalman filter is employed for this purpose. Previous works [18] have employed sigma point Kalman filters to accomplish similar goals. The EKF will use mission data from Mars Exploration Rover (MER) IMUs. It will be shown that the model-based EKF algorithm leads to better navigation results than dead-reckoning as measured by estimate error uncertainty. The single EKF navigation system is expanded to a MMAE architecture in order to account for different possible atmospheric conditions. In previously reported investigations, a different MMAE scheme [19] was used to filter simulated data. In this work a new filter selection scheme is developed and used to process MER IMU data.

The MMAE is an adaptable estimation technique that consist of a bank of parallel filters. It has been a topic of great interest since Magill's pioneering work [20]. The Magill scheme has been modified to study a variety of problems. The interacting multiple model (IMM) [21] is a MMAE scheme that has received attention in the past years. To avoid the necessity of having a large bank of filters to implement every possible parameter realization, the concept of moving bank was introduced [22]. Methods to enhance the MMAE performance were investigated [23], and conditions for the effective steady-state performance were studied [24]. The MMAE techniques were successfully used for space structures control [25], and actuator-sensor failure detection in various situations, for example on the F-16 [26]. Other applications are tracking maneuvering targets [27], and estimation in presence of switching coefficients [28].

Together with the filter bank, the MMAE has an hypothesis algorithm that weights each filter in the bank. In the Magill scheme case, the weight is given by the conditional probability, and is used to combine the state estimates into a single optimal estimated state. Other possible weighting methods exist, including the single layer gating network [29–32]. The gating network approach is followed here because it is a "winner take all" strategy consistent with our objective of determining the filter producing the "best" state estimate. Each filter in the bank represents a different realization of Mars atmosphere (e.g., one filter represent nominal expected density, another represents possible high density conditions, and so forth). The filter in the bank assigned the highest weight by the gating network indirectly indicates the atmospheric conditions.

The last application developed is Moon descent navigation. For this application the gyro is filtered and attitude estimation is decentralized using a star-tracker that provides a quaternion output rather than raw measurements.

1.1 Contributions of the Dissertation

This work focuses on autonomous navigation for precision landing on distant planets. The major contribution of the dissertation is the detailed theoretical study of two important aspects of the navigation scheme: inclusion of attitude estimation and inertial measurements.

An autonomous vehicle performs its own measurements, which therefore depend on the orientation of the spacecraft. This dependance introduces a correlation between the attitude and translation estimates. Ignoring this correlation results in a nonoptimal navigation solution which deteriorates the system navigation performance. This work researches optimal and suboptimal ways of introducing an attitude estimate into the navigation filter.

Integrating the IMU measurements is a simple solution, and is relatively accurate because of the precision obtainable by modern IMUs. However dead-reckoning is an open loop-method in which the errors of the IMU are directly transmitted to the estimate. A closed-loop solution in which the measurements are compared to a model-based estimate can filter out some of the IMU errors leading to better performance. Detailed applications of both solutions are presented in this work.

Designing a spacecraft with independent attitude and translation filters that dead-reckon the IMU is a proven, reliable method. The need for a very precise navigation estimate however, motivates the search for techniques that can improve the navigation system performance. This dissertation investigates two alternative options which will improve the filter estimate. Quantification of the improvement is mission and hardware specific, and beyond the scope of this theoretical study. However, three applications of relevance to today's aerospace field have been developed to show the possibilities of these techniques.

1.2 Organization of the Dissertation

The organization of this work is as follows: In Chapter 2 the building blocks of the investigation are presented. All relevant equations for the study of attitude (composition, kinematics, Euler's equations) are introduced in Section 2.1. The section focuses on the chosen attitude parametrization, the quaternion-of-rotation, and presents the notation used in this work. Extensive work has been done on the topic of quaternion estimation using Kalman filters. These works are presented in Section 2.2. Section 2.3 introduces an important nonlinear attitude determination algorithm, together with the derivation of the estimation error covariance. Section 2.4 introduces an existing MMAE architecture which will be the bases of the adaptable filter used for Mars entry navigation.

Chapter 3 contains the theoretical contributions of these thesis. It is divided into three section, each containing the theory relevant to one of the three applications presented in Chapters 4–6. Section 3.1 develops the algorithm to incorporate the estimate from an attitude determination algorithm into the EKF. This algorithm is applied to lunar descent navigation (Chapter 6). Section 3.2 develops Kalman filter equations under the assumption that process and measurement noises are composed by a white process and a random bias. Two algorithms are introduced, assuming either discrete (§3.2.1), or continuous (§3.2.2) measurements. The algorithms are applied to Mars entry navigation, specifically they are used in the "classic" navigation approach where the IMU is dead-reckoned. Section 3.3 introduces the modified MMAE scheme, to be used for Mars entry navigation when the accelerometer is filtered (Chapter 5).

Chapter 4 contains a detailed linear covariance analysis of Mars entry navigation. During this phase the IMU provides the only available measurements. In this chapter the dead-reckoning approach is studied. The IMU measurements are modeled including random biases, misalignment errors, and scale factors. These error sources are all considered in the covariance analysis. Both a continuous-time IMU providing directly non-gravitational acceleration and angular velocity (§4.2), and a discrete-time IMU integrating them (§4.3) are considered.

In Chapter 5 the approach to Mars entry navigation is that of filtering the accelerometer. An adaptable filter is successfully implemented to process observations from the Mars Exploration Rover mission. A comparison of the dead-reckoning and filtering approaches is performed, demonstrating the advantages of the filtering approach.

Chapter 6 contains the lunar navigation example, where a star-tracker feeds the Kalman filter with an attitude estimate rather than raw measurements of stars positions. To correctly account for the autocorrelation in the attitude estimate, the star-tracker also provides to the Kalman filter a covariance. The covariance equation used in this chapter is introduced in Section 2.3.1. The proposed filter processes the gyro data to update the state.

A summary of major results and conclusions is given in Chapter 7.

Chapter 2

Preliminary Notions

In this chapter, the basic material to be used in the sequel is introduced. In Section 2.1, the mathematical notation and the equations describing the spacecraft attitude are presented. The quaternion is the chosen representation of the attitude. In Section 2.2, a review of quaternion estimation is presented. Section 2.3 introduces the quaternion estimation algorithm to be used in the star-tracker of Chapter 6. Finally, Section 2.4 introduces the gating network on which the developed MMAE scheme is based.

2.1 Attitude Representations, Kinematics, and Dynamics

Many of the equations presented in this section describing the physics of rotations are due to the Swiss mathematician Leonhard Euler, who is therefore the father of this branch of classical mechanics.

2.1.1 Attitude Representation

Solutions to the problem of representing the rotations in Euclidean 3-space appeared in 1775 when Euler presented two fundamental papers. In the first paper, Euler enunciated his famous theorem stating that all displacements about a fixed point can be represented with a rotation about an axis [33]. From this theorem an attitude representation comes natural: the so called Euler axis and angle $[\hat{\mathbf{n}} \quad \theta]_{b,i}$, where the subscript b, i indicates that rotation is from i to b. Superscripts on vectors will denote the frame in which the components of the vector are calculated. Note that $\hat{\mathbf{n}}^i_{b,i} = \hat{\mathbf{n}}^b_{b,i}$, i.e. the rotation vector has the same components in both frames. The Euler axis and angle are a redundant representation since there is a unitary constraint on the norm of $\hat{\mathbf{n}}$. The associated minimum representation is given by the rotation vector defined as

$$\boldsymbol{\theta} \triangleq \theta\,\hat{\mathbf{n}}.$$

Performing a first rotation from i to c, followed by a second rotation from c to b, can be expressed through a single rotation from i to b. The formula to obtain the total rotation is known as the composition rule. Define the rotation from i to c as rotation one $[\hat{\mathbf{n}}_1 \quad \theta_1]_{c,i}$, the rotation from c to b as rotation two $[\hat{\mathbf{n}}_2 \quad \theta_2]_{b,c}$, and the composed rotation as rotation three $[\hat{\mathbf{n}}_3 \quad \theta_3]_{b,i}$. The composition rule for the Euler axis and angle follows from the quaternion composition discussed later, and is given by [34]

$$\begin{aligned}
\cos\theta_3 &= \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2} - \sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\,\hat{\mathbf{n}}_2\cdot\hat{\mathbf{n}}_1\\
\hat{\mathbf{n}}_3 &= \frac{\sin(\theta_1/2)\cos(\theta_2/2)}{\sin(\theta_3/2)}\hat{\mathbf{n}}_1 + \frac{\sin(\theta_2/2)\cos(\theta_1/2)}{\sin(\theta_3/2)}\hat{\mathbf{n}}_2 +\\
&\quad - \frac{\sin(\theta_1/2)\sin(\theta_2/2)}{\sin(\theta_3/2)}\hat{\mathbf{n}}_2\times\hat{\mathbf{n}}_1.
\end{aligned}$$

In his second 1775 paper, Euler's formula was introduced [35]. In today terminology the formula relates the axis and angle of rotation with the direction cosines matrix. The direction cosines matrix (or rotation matrix) **T**, is a convenient parametrization because it treats rotation of vectors using vector algebra, and the composition of rotations is given by matrix multiplication. Being an orthogonal matrix, **T** has six constraints. Therefore, its highly redundant representation makes it undesirable in estimation algorithms, such as the Kalman filter. In terms of the Euler axis and angle, the rotation matrix is given by

$$\mathbf{T}_i^b = \mathbf{I}_{3\times 3} - \sin\theta\,[\widehat{\mathbf{n}}\times] + (1-\cos\theta)[\hat{\mathbf{n}}\times]^2, \tag{2.1}$$

where $\mathbf{T}_i^b$ is the rotation matrix from frame i to frame b, and the skew-symmetric cross product matrix is defined as

$$[\boldsymbol{\alpha}\times] = \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix}.$$

For small angles $\delta\theta$, Eq. (2.1) is approximately given by

$$\mathbf{T}_i^b \simeq \mathbf{I}_{3\times 3} - \delta\theta\,[\widehat{\mathbf{n}}\times]. \tag{2.2}$$

From the definition of the cross product matrix

$$\boldsymbol{\alpha}\times\boldsymbol{\beta} = [\boldsymbol{\alpha}\times]\boldsymbol{\beta}, \quad \forall\boldsymbol{\alpha},\boldsymbol{\beta}\in\Re^3,$$

the following properties will be useful

$$[(\boldsymbol{\alpha}\times\boldsymbol{\beta})\times] = \boldsymbol{\beta}\boldsymbol{\alpha}^T - \boldsymbol{\alpha}\boldsymbol{\beta}^T, \qquad [\boldsymbol{\alpha}\times]\,[\boldsymbol{\beta}\times] = \boldsymbol{\beta}\boldsymbol{\alpha}^T - \boldsymbol{\beta}^T\boldsymbol{\alpha} I_{3\times 3}. \tag{2.3}$$

In 1770 Euler showed that three angles were sufficient to represent any rotation [36]. This result can be generalized with three parameters that are sufficient to represent a rotation. For example, Euler presented a three dimensional parametrization, the Euler angles [37], that is not of interest in this work. Other three dimensional representations are the Rodrigues parameters [38] given by $\tan(\theta/2)\hat{\mathbf{n}}$ and the modified Rodrigues parameters $\tan(\theta/4)\hat{\mathbf{n}}$. Let $\boldsymbol{\varrho}$ be the vector of Rodrigues parameters. Then the composition rule is given by

$$\boldsymbol{\varrho}_3 = \frac{\boldsymbol{\varrho}_2 + \boldsymbol{\varrho}_1 - \boldsymbol{\varrho}_2 \times \boldsymbol{\varrho}_1}{1 - \boldsymbol{\varrho}_1 \cdot \boldsymbol{\varrho}_2}, \tag{2.4}$$

where rotation one is from frame i to c, rotation two is from c to b, and rotation three is from i to b.

All three-dimensional representations are singular. The first proof is due to Frobenius during his work on abstract algebra (Kuipers [6]). Another proof is given by Stuelpnagel [5]. The singularity of a three-dimensional representation can sometimes be avoided by introducing a discontinuity. Either way the representation may not be satisfactory. In 1940 Hopf proved that the minimum dimension to represent the rotation group in a 1one-to-one global manner is five [39]. Representing the rotation with two redundant elements introduces two constraints, and no five dimensional representation has been found with nice properties. The quaternion is a four-dimensional representation which is not topologically equivalent to the three-dimensional rotation because is a one-to-two representation, but it is sufficient for our purposes.

Quaternion-of-Rotation

Olinde Rodrigues introduced the three dimensional representation that was later study by Gibbs and is often referred as Gibbs vector. Rodrigues also introduced in the same 1840 paper [38] a four dimensional representation that was attributed to

Euler. This representation is known as Euler-Rodrigues symmetric parametrization, but it is more commonly referred as quaternion-of-rotation, or simply quaternion.

In 1843, three years after Rodrigues, Sir William Rowan Hamilton introduced the quaternion. Hamilton's intent was not to parameterize $SO(3)$, instead he invented a new algebra in which the elements were both operators (rotations) and operands (vectors) [40]. Cayley discovered that by defining a quaternion via Euler-Rodrigues parameters, the resulting unitary quaternion represents a rotation, and that the quaternion multiplication is precisely the rotation composition introduced by Rodrigues [41]. Every unitary quaternion represent a rotation, hence are called quaternion-of-rotation. Since in this work the quaternion is used only as a rotation parametrization, *of-rotation* designations will be omitted.

The quaternion used here has the vector first and scalar last, $\overline{\mathbf{q}} = \begin{bmatrix}\mathbf{q}^{\mathrm{T}} & q\end{bmatrix}^{\mathrm{T}}$. The quaternion as a function of Euler axis and angle is given by

$$\overline{\mathbf{q}} = \begin{bmatrix}\mathbf{q} \\ q\end{bmatrix} = \begin{bmatrix}\sin\left(\frac{\theta}{2}\right)\hat{\mathbf{n}} \\ \cos\left(\frac{\theta}{2}\right)\end{bmatrix}.$$

The associated rotation matrix is

$$\mathbf{T} \triangleq \mathbf{T}(\overline{\mathbf{q}}) = \mathbf{I}_{3\times 3} - 2q[\mathbf{q}\times] + 2[\mathbf{q}\times]^2. \tag{2.5}$$

To perform a sequence of two rotations, from i to c to b, the total rotation is

$$\mathbf{T}_i^b = \mathbf{T}_c^b\mathbf{T}_i^c = \mathbf{T}(\overline{\mathbf{q}}_c^b)\mathbf{T}(\overline{\mathbf{q}}_i^c) = \mathbf{T}(\overline{\mathbf{q}}_c^b \otimes \overline{\mathbf{q}}_i^c).$$

The quaternion product $\otimes$ is defined such that the quaternions are multiplied in the same order as the attitude matrices,

$$\overline{\mathbf{q}} \otimes \overline{\mathbf{p}} = \begin{bmatrix} q\mathbf{p} + p\mathbf{q} - \mathbf{q} \times \mathbf{p} \\ qp - \mathbf{q} \cdot \mathbf{p} \end{bmatrix}.$$

Originally Hamilton defined the product in opposite order, Hamilton's product will be denoted by $\circledast$, and in terms of $\otimes$ is given by

$$\overline{\mathbf{p}} \circledast \overline{\mathbf{q}} = \overline{\mathbf{q}} \otimes \overline{\mathbf{p}}.$$

The quaternion product is a bilinear operator, therefore product matrices can be defined as

$$\overline{\mathbf{q}} \otimes \overline{\mathbf{p}} = [\overline{\mathbf{q}} \otimes]\, \overline{\mathbf{p}} = [\overline{\mathbf{p}} \circledast]\, \overline{\mathbf{q}},$$

from which it follows that

$$[\overline{\mathbf{q}} \otimes] = [\boldsymbol{\Psi}(\overline{\mathbf{q}}) \;\; \overline{\mathbf{q}}], \qquad \boldsymbol{\Psi}(\overline{\mathbf{q}}) = \begin{bmatrix} q\mathbf{I}_{3\times3} - [\mathbf{q}\times] \\ -\mathbf{q}^{\mathrm{T}} \end{bmatrix},$$

$$[\overline{\mathbf{q}} \circledast] = [\boldsymbol{\Xi}(\overline{\mathbf{q}}) \;\; \overline{\mathbf{q}}], \qquad \boldsymbol{\Xi}(\overline{\mathbf{q}}) = \begin{bmatrix} p\mathbf{I}_{3\times3} + [\mathbf{q}\times] \\ -\mathbf{q}^{\mathrm{T}} \end{bmatrix}.$$

The matrix $\boldsymbol{\Xi}(\overline{\mathbf{q}})$ is particularly important and possesses the following properties

$$\boldsymbol{\Xi}^{\mathrm{T}}(\overline{\mathbf{q}})\, \boldsymbol{\Xi}(\overline{\mathbf{q}}) = (\overline{\mathbf{q}}^{\mathrm{T}}\, \overline{\mathbf{q}})\mathbf{I}_{3\times3} \qquad \boldsymbol{\Xi}(\overline{\mathbf{q}})\, \boldsymbol{\Xi}^{\mathrm{T}}(\overline{\mathbf{q}}) = (\overline{\mathbf{q}}^{\mathrm{T}}\, \overline{\mathbf{q}})\mathbf{I}_{4\times4} - \overline{\mathbf{q}}\, \overline{\mathbf{q}}^{\mathrm{T}}$$

$$\boldsymbol{\Xi}^{\mathrm{T}}(\overline{\mathbf{q}})\, \overline{\mathbf{q}} = \mathbf{O}_{3\times1} \qquad \boldsymbol{\Xi}^{\mathrm{T}}(\overline{\mathbf{q}})\, \overline{\mathbf{p}} = -\boldsymbol{\Xi}^{\mathrm{T}}(\overline{\mathbf{p}})\, \overline{\mathbf{q}}.$$

The inverse quaternion and the identity quaternion are

$$\overline{\mathbf{q}}^{-1} = \begin{bmatrix} -\mathbf{q}^{\mathrm{T}} & q \end{bmatrix}^{\mathrm{T}} \qquad \bar{\mathbf{i}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}.$$

It is easy to verify that

$$\overline{\mathbf{q}} \otimes \overline{\mathbf{q}}^{-1} = \bar{\mathbf{i}}.$$

A pure quaternion is a quaternion with zero scalar component. A pure quaternion obtained from a three dimensional vector $\mathbf{v}$ is defined as

$$\overline{\mathbf{v}} \triangleq \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}.$$

The rotation of a vector $\mathbf{v}$ can be written in quaternion form as

$$\overline{\mathbf{v}}^{b} = \overline{\mathbf{q}}_{i}^{b} \otimes \overline{\mathbf{v}}^{i} \otimes \left(\overline{\mathbf{q}}_{i}^{b}\right)^{-1},$$

from which is obtained that the rotation matrix $\mathbf{T}(\overline{\mathbf{q}})$ can also be expressed as

$$\mathbf{T}(\overline{\mathbf{q}}) = \boldsymbol{\Xi}(\overline{\mathbf{q}})^{\mathrm{T}} \boldsymbol{\Psi}(\overline{\mathbf{q}}).$$

2.1.2 Attitude Kinematics

The angular velocity of a body rotating with respect to a reference frame i, is defined as

$$\boldsymbol{\omega}(t) \triangleq \lim_{\Delta t \to 0} \frac{\Delta \theta}{\Delta t} \hat{\mathbf{n}}(t). \tag{2.6}$$

A few remarks are important. The Euler axis and angle of Eq. (2.6) represent the rotation of the body frame from time t to time $t + \Delta t$ and should not be confused with the rotation from the reference to the body frame. The definition given in Eq. (2.6) naturally provides the angular velocity in the body, or moving frame, which as before will be denoted with a superscript $\boldsymbol{\omega}^{b}$. We started by stating that the body frame was rotating with respect to a reference frame. When this relation needs to be expressed explicitly, it will be indicated as $\boldsymbol{\omega}_{b,i}^{b}(t)$.

The fact that reference b is rotating with respect to reference i is arbitrary, the same analysis could be carried considering reference i rotating with respect to reference b. Its angular velocity will then be denoted as $\boldsymbol{\omega}_{i,b}$ and

$$\boldsymbol{\omega}_{i,b} = -\boldsymbol{\omega}_{b,i}.$$

Denote by $\Delta\mathbf{T}(t)$ the rotation matrix between the body frame at time t and at time $t+\Delta t$, such that

$$\mathbf{T}_i^b(t+\Delta t) = \Delta\mathbf{T}(t)\,\mathbf{T}_i^b(t). \tag{2.7}$$

Expressing $\Delta\mathbf{T}(t)$ in terms of the Euler axis and angle of Eq. (2.1), it follows that

$$\Delta\mathbf{T}(t) = \mathbf{I}_{3\times3} - \sin\Delta\theta\,[\hat{\mathbf{n}}(t)\times] + (1-\cos\Delta\theta)\,[\hat{\mathbf{n}}(t)\times]^2. \tag{2.8}$$

Using Eqs. (2.7) and (2.8), the derivative of the rotation matrix is obtained as

$$\dot{\mathbf{T}}_i^b(t) \triangleq \lim_{\Delta t\to0}\frac{\mathbf{T}_i^b(t+\Delta t)-\mathbf{T}_i^b(t)}{\Delta t} = \lim_{\Delta t\to0}\frac{-\Delta\theta\,[\hat{\mathbf{n}}(t)\times]}{\Delta t}\,\mathbf{T}_i^b(t) = -[\boldsymbol{\omega}_{b,i}^b\times]\,\mathbf{T}_i^b. \tag{2.9}$$

Knowing the initial orientation of a body and the angular velocity history, is possible to integrate Eq. (2.9) to compute the attitude of the body at any given time. In practice this integration will be done numerically. The numerical integration will introduce roundoff errors which will result in nonorthogonality. One procedure to mitigate the round-off error is to reinstate orthogonality every few integration steps. A much more common and efficient strategy is not to integrate the nine components of the rotation matrix but only the three or four elements of a lower dimensional representation, and to calculate the rotation matrix when needed.

The fact that the kinematics of the quaternion is simple is one of the reason this parametrization is so popular. The derivative of the quaternion is a bilinear form in the angular velocity and the quaternion. The derivative of the quaternion is

$$\dot{\overline{\mathbf{q}}}_i^b(t) \triangleq \lim_{\Delta t \to 0} \frac{\overline{\mathbf{q}}_i^b(t+\Delta t) - \overline{\mathbf{q}}_i^b(t)}{\Delta t}. \tag{2.10}$$

Define $\Delta\overline{\mathbf{q}}$ such that

$$\overline{\mathbf{q}}_i^b(t+\Delta t) = \Delta\overline{\mathbf{q}}(t) \otimes \overline{\mathbf{q}}_i^b(t).$$

The quaternion $\Delta\overline{\mathbf{q}}$ can be written in terms of Euler axis and angle as

$$\Delta\overline{\mathbf{q}} = \begin{bmatrix} \sin\left(\frac{\Delta\theta}{2}\right)\, \hat{\mathbf{n}}(t) \\ \cos\left(\frac{\Delta\theta}{2}\right) \end{bmatrix}.$$

As $\Delta\theta$ goes to zero, it follows that

$$\Delta\overline{\mathbf{q}} \to \begin{bmatrix} \frac{\Delta\theta}{2}\hat{\mathbf{n}}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{\Delta\theta}{2}\hat{\mathbf{n}}(t) \\ 0 \end{bmatrix}.$$

Hence, as $\Delta\theta$ goes to zero,

$$\Delta\overline{\mathbf{q}}(t) \otimes \overline{\mathbf{q}}_i^b(t) = \overline{\mathbf{q}}_i^b(t) + \frac{1}{2} \begin{bmatrix} \Delta\theta\, \hat{\mathbf{n}}(t) \\ 0 \end{bmatrix} \otimes \overline{\mathbf{q}}_i^b(t),$$

from which we find that Eq. (2.10) reduces to

$$\dot{\overline{\mathbf{q}}}_i^b(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(\frac{1}{2} \begin{bmatrix} \Delta\theta\hat{\mathbf{n}}(t) \\ 0 \end{bmatrix} \otimes \overline{\mathbf{q}}_i^b(t) \right) = \frac{1}{2} \begin{bmatrix} \boldsymbol{\omega}_{b,i}^b(t) \\ 0 \end{bmatrix} \otimes \overline{\mathbf{q}}_i^b(t). \tag{2.11}$$

Eq. (2.11) can be rewritten in three equivalent ways:

$$\dot{\overline{\mathbf{q}}}_i^b = \frac{1}{2}\overline{\boldsymbol{\omega}}_{b,i}^b \otimes \overline{\mathbf{q}}_i^b = \frac{1}{2}\boldsymbol{\Xi}(\overline{\mathbf{q}}_i^b)\, \boldsymbol{\omega}_{b,i}^b = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{b,i}^b)\, \overline{\mathbf{q}}_i^b,$$

where the dependency on time is omitted. The matrix $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) \triangleq [\overline{\boldsymbol{\omega}}\otimes],$$

and

$$\overline{\boldsymbol{\omega}}^b_{b,i} \triangleq \begin{bmatrix} \boldsymbol{\omega}^b_{b,i} \\ 0 \end{bmatrix}.$$

According to Shuster [34], Eq. (2.11) was introduced by Cayley [42]. The evolution of the rotation vector is given by the Bortz equation [43],

$$\dot{\boldsymbol{\theta}}^b_i = \boldsymbol{\omega}^b_{b,i} + \frac{1}{2}\boldsymbol{\theta}^b_i \times \boldsymbol{\omega}^b_{b,i} + \frac{1}{\theta^2}[1 - (\theta/2)\cot(\theta/2)]\boldsymbol{\theta}^b_i \times \boldsymbol{\theta}^b_i \times \boldsymbol{\omega}^b_{b,i},$$

where $\theta = \|\boldsymbol{\theta}^b_i\|$.

The following identities will be helpful in the formulation that follows. First

$$\dot{\boldsymbol{\Xi}}\left(\overline{\mathbf{q}}^{\,b}_i\right) = \boldsymbol{\Xi}\left(\dot{\overline{\mathbf{q}}}^{\,b}_i\right) = \frac{1}{2}\boldsymbol{\Xi}\left(\boldsymbol{\Omega}(\boldsymbol{\omega}^b_{b,i})\,\overline{\mathbf{q}}^{\,b}_i\right) = \frac{1}{2}\boldsymbol{\Xi}\left(\begin{bmatrix} \mathbf{q}\times\boldsymbol{\omega}^b_{b,i} + q\,\boldsymbol{\omega}^b_{b,i} \\ -(\boldsymbol{\omega}^b_{b,i})^{\mathrm{T}}\mathbf{q} \end{bmatrix}\right) =$$

$$= \frac{1}{2}\begin{bmatrix} -(\boldsymbol{\omega}^b_{b,i})^{\mathrm{T}}\mathbf{q}\,\mathbf{I}_{3\times 3} + \left[(\mathbf{q}\times\boldsymbol{\omega}^b_{b,i})\times\right] + q[\boldsymbol{\omega}^b_{b,i}\times] \\ (\boldsymbol{\omega}^b_{b,i}\times\mathbf{q})^{\mathrm{T}} - q\,(\boldsymbol{\omega}^b_{b,i})^{\mathrm{T}} \end{bmatrix},$$

where $\mathbf{q}$ and q are the vector and scalar components of $\overline{\mathbf{q}}^{\,b}_i$. Using Eq. (2.3), it can be found that

$$\dot{\boldsymbol{\Xi}}\left(\overline{\mathbf{q}}^{\,b}_i\right) = \frac{1}{2}\boldsymbol{\Xi}\left(\overline{\mathbf{q}}^{\,b}_i\right)\,[\boldsymbol{\omega}^b_{b,i}\times] - \frac{1}{2}\overline{\mathbf{q}}^{\,b}_i\,(\boldsymbol{\omega}^b_{b,i})^{\mathrm{T}}.$$

Also, we have

$$\boldsymbol{\Omega}\left(\boldsymbol{\omega}^b_{b,i}\right)\,\boldsymbol{\Xi}\left(\overline{\mathbf{q}}^{\,b}_i\right) = -\boldsymbol{\Xi}\left(\overline{\mathbf{q}}^{\,b}_i\right)\,[\boldsymbol{\omega}^b_{b,i}\times] - \overline{\mathbf{q}}^{\,b}_i\,(\boldsymbol{\omega}^b_{b,i})^{\mathrm{T}}.$$

2.1.3 Attitude Dynamics

After realizing that the rotation dynamics were independent of the translational dynamics [44], Euler presented the angular momentum law in a fixed reference frame [45]. In 1758, he presented *Euler's equations* [46] which, in modern notation, are given by

$$\mathbf{J}\,\dot{\boldsymbol{\omega}}^b_{b,i} = -\boldsymbol{\omega}^b_{b,i} \times \mathbf{J}\,\boldsymbol{\omega}^b_{b,i} + \mathbf{m}^b, \tag{2.12}$$

where the reference frame i has to be inertial, $\mathbf{m}$ are the external moments. The matrix $\mathbf{J}$ is the moment of inertia expressed in body frame with respect to the center of mass.

Note

When Eneström categorized Euler's works, he gave the years of publication, presentation, and approximate composition. These three dates can sometimes differ substantially from each other. In the bibliography the dates are those of first publication, followed by Eneström index number. All Euler's original works can be viewed at the Euler web archive [47]. The following references were also consulted for historical information [48, 49].

2.2 Quaternion Estimation in the Kalman Filter

Attitude estimation has been the topic of much research and debate over the past two decades [50]. The interest arises from the fact that the representation of the attitude is not a vector space and redundancy is necessary to avoid singularities and discontinuities [5]. For real-time space applications, the quaternion is a favorite attitude representation and will be utilized in this work. In sequential real-time quaternion estimation, two schools of thought receive the most attention: the Additive Extended Kalman Filter (AEKF) [9] and the Multiplicative Extended Kalman

Filter (MEKF) [10]. While the additive approach resembles closely the standard extended Kalman filter (EKF), several shortcomings of the AEKF were pointed out. These shortcomings are:

1. The estimation error does not have a physical meaning.
2. The estimation error covariance becomes ill-conditioned.
3. The algorithm requires a brute force normalization procedure.

Theoretical studies show that the covariance in the AEKF should be nearly singular [11], while practical applications do not reveal the problem [51].

The multiplicative approach defines the estimation error as a rotation. The shortcomings of the MEKF are:

1. The quaternion innovation is obtained through a first- (or second-) order approximation.
2. The quaternion is not estimated directly but the deviation from the nominal is estimated.

Since the quaternion innovation is approximate, the norm of the updated quaternion is not maintained, hence the MEKF necessitates restoring the norm constraint after the update. The most obvious method to accomplish this is to re-scale the updated quaternion by its norm, thereby minimizing the Euclidean distance between the unconstrained and the constrained estimates [52]. The normalization process also provides the unitary estimate with minimum mean square error [53].

2.2.1 Existing Methods of Including Constraints in the Kalman Filter

It is well-known that the Kalman filter provides the *unconstrained* optimal solution of the linear stochastic estimation problem [1–3]. The Kalman filter algorithm has two main phases: the state estimate propagation phase between measurements, and the state estimate update phase when measurements become available. Unconstrained implies that the optimal state estimate is not constrained during the state estimate update phase as the measurements are processed. The Kalman filter provides the optimal state estimate considering n degrees of freedom (that is, the entire vector space $\Re^n$). However, if r state constraints are applied, the degrees of freedom are reduced to $n - r$. Simply projecting the unconstrained solution into the constrained space will not guarantee optimality.

One method of introducing state constraints is to use pseudo-measurements [54,55]. The fundamental idea is to introduce a perfect measurement (hence the use of the term "pseudo-measurement") consisting of the constraint equation into the estimation solution. In the MEKF, the pseudo-measurement is given by

$$\mathbf{y}_{pm} = \overline{\mathbf{q}}^{\mathrm{T}}\overline{\mathbf{q}}$$

and is always equal to one. The residual is therefore given by

$$\boldsymbol{\epsilon} = 1 - \widehat{\overline{\mathbf{q}}}^{\mathrm{T}}\widehat{\overline{\mathbf{q}}},$$

and the measurement mapping matrix is

$$\mathbf{H}_{pm} = 2\widehat{\overline{\mathbf{q}}}^{\mathrm{T}}.$$

Since the norm of the quaternion is truly one, the measurement is perfect, and the measurement error covariance matrix is zero.

This approach has two shortcomings. First, the use of a perfect measurement results in a singular estimation problem known to occur when processing noise-free measurements in a Kalman filter. A small noise can be added to the pseudo-measurement to address the singularity; however with the noise introduced, the constraint is no longer exactly satisfied. Second, since the constraint is nonlinear, after the linearization of the measurement equation consistent with the EKF algorithm, the constraint is no longer satisfied exactly.

One can consider state constraints when considering the optimization problems based on least-squares methods. The solution to the least-squares problem in the presence of linear equality constraints is found in Lawson and Hanson [56]. Another approach is to project the Kalman solution into the desired subspace. Since the projection can be done in different ways, a performance index can be defined to find the optimal projection. The optimal projection for the linear state equality constraint problem is presented in Simon and Chia [57]. The projection of the Kalman solution can be done at any time, not only during the update.

In the multiplicative approach, the attitude is not estimated directly but instead the deviation from the nominal attitude is estimated. This deviation employs a small angle approximation. The quaternion estimate is found by composing the nominal quaternion and the deviation. The small angle approximation results in an approximation of the quaternion norm, i.e. if a first order approximation on the angle is made, the quaternion will have norm one to first order. It is important that the quaternion has exactly norm one, otherwise it will not only rotate vectors, but change their norms too. Therefore, normalization occurs in the MEKF.

The normalization procedure for both AEKF and MEKF consists in rescaling the updated quaternion estimate by its norm.

2.2.2 Quaternion Estimation Errors: Additive and Multiplicative

The additive estimation error is defined as

$$\mathbf{e}_q \triangleq \overline{\mathbf{q}} - \widehat{\overline{\mathbf{q}}}.$$

Because of the unity norm constraints on $\overline{\mathbf{q}}$ and $\widehat{\overline{\mathbf{q}}}$, the issue of computing an estimate with zero mean estimation error is more delicate than in the traditional Kalman filter. For example, if $\overline{\mathbf{q}}$ were deterministic (i.e. no process noise), for $\mathbf{e}_q$ to be zero mean it would imply that

$$\mathrm{E}\left\{\widehat{\overline{\mathbf{q}}}\right\} = \overline{\mathbf{q}},$$

which is generally not possibly because the mean of a distribution over a four-dimension hyper-sphere is necessarily inside the sphere.

In engineering applications, emphasis is placed on the estimation error covariance and how it should be minimized. It is therefore important to notice that minimizing the trace of the covariance follows from the desire to minimize the mean square error (MSE) of an unbiased estimator. The MSE is the the most common way of evaluating estimators, and is defined as

$$\mathrm{MSE} \triangleq \mathrm{E}\left\{\mathbf{e}^{\mathrm{T}}\,\mathbf{e}\right\} = \mathrm{E}\left\{(\mathbf{x}-\widehat{\mathbf{x}})^{\mathrm{T}}\,(\mathbf{x}-\widehat{\mathbf{x}})\right\}.$$

The covariance $\mathbf{P}$ of a random vector $\mathbf{e}$ is defined as

$$\mathbf{P} \triangleq \mathrm{E}\left\{(\mathbf{e}-\mathrm{E}\left\{\mathbf{e}\right\})\,(\mathbf{e}-\mathrm{E}\left\{\mathbf{e}\right\})^{\mathrm{T}}\right\} = \mathrm{E}\left\{\mathbf{e}\,\mathbf{e}^{\mathrm{T}}\right\} - \mathrm{E}\left\{\mathbf{e}\right\}\,\mathrm{E}\left\{\mathbf{e}\right\}^{\mathrm{T}}.$$

The matrix $\mathrm{E}\left\{\mathbf{e}\,\mathbf{e}^{\mathrm{T}}\right\}$ is the mean-square of vector $\mathbf{e}$. When $\mathbf{e}$ is zero mean, covariance and mean-square coincide.

The matrix $\mathbf{P}$ of the Kalman filter is defined from the estimation error $\mathbf{e}$ as

$$\mathbf{P} \triangleq \mathrm{E}\left\{\mathbf{e}\,\mathbf{e}^{\mathrm{T}}\right\}.$$

The definition of $\mathbf{P}$ is that of a mean-square, but $\mathbf{P}$ is referred to as covariance because the Kalman filter is an unbiased estimator (i.e. with zero mean estimation error). By minimizing the trace of the covariance, the Kalman filter minimizes the MSE. Minimizing the covariance is desirable only in the presence of unbiased estimators. Minimizing the covariance implies *shrinking* the estimation error around its mean, which is not necessarily good, since the mean might be large. Figure 2.1 illustrates this concept, the errors of two estimators are plotted. The errors of the biased estimator are represented in blue. The bias is given by $[5 \quad 5]^{\mathrm{T}}$ and the covariance is given by the identity matrix. The errors of the unbiased estimator are shown in green. The covariance is four times larger than the covariance of the biased estimator. However the biased estimator has clearly much larger errors and MSE.

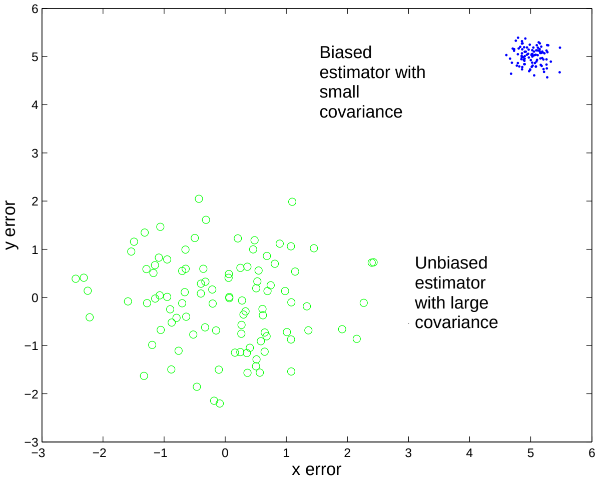


Figure 2.1: Comparison of mean square errors of two estimators.

The estimation error is a way of quantifying the difference between the true state and the estimated state, therefore it does not necessarily have to be physically meaningful. The same holds true for the MSE, which is one measure of the performance of the estimator. A small MSE it is desirable independent of the physics of the problem and independent of the covariance matrix (which might be singular). Like any performance index, its use could be replaced by another measure leading to a different optimization solution. The lack of physical meaning of the additive estimation error is a drawback only because evaluation of the performance of the filter is less intuitive (the errors are not angles). This fact, however, should not be confused with the performance of the AEKF. The additive approach minimizes the standard statistical performance index, which is the MSE (minimum over all linear estimators and conditional to the linearization approximation).

Quaternion estimation has the goal of making the *distance* between the estimated and the estimate variables small. The multiplicative approach defines this *distance* to be a quaternion itself

$$\delta\overline{\mathbf{q}} = \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1}.$$

This approach guarantees a physical interpretation of the estimation error: $\delta\overline{\mathbf{q}}$ is the rotation from the estimated body frame to the true body frame. The relation between the multiplicative error and the additive error is given by

$$\mathbf{e}_q = \overline{\mathbf{q}} - \widehat{\overline{\mathbf{q}}} = \overline{\mathbf{q}} - \delta\overline{\mathbf{q}}^{-1} \otimes \overline{\mathbf{q}} = \overline{\mathbf{q}} - \begin{bmatrix} -\boldsymbol{\Xi}(\overline{\mathbf{q}}) & \overline{\mathbf{q}} \end{bmatrix} \delta\overline{\mathbf{q}}.$$

The true state, $\overline{\mathbf{q}}$, is often treated as deterministic, and therefore taken outside the expectation operation in calculating the theoretical value of the estimation error covariance. This operation is legitimate only in the absence of process noise. In the presence of process noise, $\overline{\mathbf{q}}$ is a random quantity and taking it outside the expected value results in an approximation, therefore only approximate conclusions can be inferred from this procedure.

The *covariance* which is associated with the additive Kalman filter is

$$\mathbf{P}_a = \mathrm{E}\left\{\mathbf{e}_q\, \mathbf{e}_q^{\mathrm{T}}\right\},$$

and for the multiplicative approach

$$\mathbf{P}_{4m} = \mathrm{E}\left\{\delta\overline{\mathbf{q}}\,\delta\overline{\mathbf{q}}^{\mathrm{T}}\right\}.$$

In calculating $\mathbf{P}_a$, the additive Kalman filter does not compensate for the mean of $\mathbf{e}_q$, therefore any theoretical study of $\mathbf{P}_a$ should not contain $\mathrm{E}\left\{\mathbf{e}_q\right\}$ unless it is proven to be zero. It was shown in [7] that when the estimation error is a small rotation, $\mathbf{P}_{4m}$ becomes ill-conditioned. Assuming the true quaternion $\overline{\mathbf{q}}$ is deterministic and relating the two matrices, will result in $\mathbf{P}_a$ being ill-conditioned as well. However, when $\overline{\mathbf{q}}$ is a random vector it cannot be taken outside the expected value, and no conclusions can be made on $\mathbf{P}_a$ from the condition of $\mathbf{P}_{4m}$. The matrix $\mathbf{P}_a$ will depend on the joint distribution of $\overline{\mathbf{q}}$ and $\delta\overline{\mathbf{q}}$ and not solely on $\delta\overline{\mathbf{q}}$.

In the absence of process noise, the Kalman filter covariance will eventually converge to zero. Therefore, the fact that in the absence of process noise $\mathbf{P}_a$ becomes ill-conditioned is an expected characteristic of the AEKF scheme.

2.2.3 Relationship Between Additive and Multiplicative Error Representations

The MEKF is equivalent to the AEKF[*]. Let the measurement $\mathbf{y}$ be related to the quaternion through a nonlinear function $\mathbf{h}$ and noise $\boldsymbol{\eta}$ as

$$\begin{aligned}\mathbf{y} &= \mathbf{h}(\overline{\mathbf{q}}) + \boldsymbol{\eta} = \mathbf{h}(\widehat{\overline{\mathbf{q}}} + \mathbf{e}_q) + \boldsymbol{\eta} = \mathbf{h}(\delta\overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}) + \boldsymbol{\eta} = \mathbf{h}\left(\left[\widehat{\overline{\mathbf{q}}}\circledast\right]\delta\overline{\mathbf{q}}\right) + \boldsymbol{\eta} \\ &\simeq \mathbf{h}\left(\Xi(\widehat{\overline{\mathbf{q}}})\delta\mathbf{q} + \widehat{\overline{\mathbf{q}}}\right) + \boldsymbol{\eta}.\end{aligned}$$

---

[*] Shuster [8] cites Ferraresi [58] to prove the equivalency, Ferraresi's work was not available to the author.

Using a Taylor series expansion and dropping higher order terms yields

$$\begin{aligned}\mathbf{h}(\widehat{\overline{\mathbf{q}}} + \mathbf{e}_q) &\simeq \mathbf{h}(\widehat{\overline{\mathbf{q}}}) + \frac{\partial}{\partial \mathbf{e}_q}\mathbf{h}(\widehat{\overline{\mathbf{q}}} + \mathbf{e}_q)\bigg|_{\mathbf{e}_q=\mathbf{0}} \mathbf{e}_q, \qquad \text{and} \\ \mathbf{h}\left(\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}})\delta\mathbf{q} + \widehat{\overline{\mathbf{q}}}\right) &\simeq \mathbf{h}(\widehat{\overline{\mathbf{q}}}) + \frac{\partial}{\partial \delta\mathbf{q}}\mathbf{h}\left(\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}})\delta\mathbf{q} + \widehat{\overline{\mathbf{q}}}\right)\bigg|_{\delta\mathbf{q}=\mathbf{0}} \delta\mathbf{q}.\end{aligned}$$

Defining $\mathbf{H}(\overline{\mathbf{q}})$ as the jacobian of $\mathbf{h}(\overline{\mathbf{q}})$ and using the chain rule, it follows that

$$\begin{aligned}\mathbf{H}_a(\widehat{\overline{\mathbf{q}}}) &\triangleq \frac{\partial}{\partial \mathbf{e}_q}\mathbf{h}(\widehat{\overline{\mathbf{q}}} + \mathbf{e}_q)\bigg|_{\mathbf{e}_q} = \mathbf{H}(\widehat{\overline{\mathbf{q}}}), \qquad \text{and} \\ \mathbf{H}_m(\widehat{\overline{\mathbf{q}}}) &\triangleq \frac{\partial}{\partial \delta\mathbf{q}}\mathbf{h}\left(\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}})\delta\mathbf{q} + \widehat{\overline{\mathbf{q}}}\right)\bigg|_{\delta\mathbf{q}=\mathbf{0}} = \mathbf{H}(\widehat{\overline{\mathbf{q}}})\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}}) = \mathbf{H}_a(\widehat{\overline{\mathbf{q}}})\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}}).\end{aligned}$$

The multiplicative quaternion error is approximately

$$\delta\overline{\mathbf{q}} \simeq \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix},$$

therefore the multiplicative error is completely defined by its vector part. Let $\mathbf{P}_m \in \Re^{3\times 3} = \mathrm{E}\left\{\delta\mathbf{q}\delta\mathbf{q}^{\mathrm{T}}\right\}$ be the multiplicative error covariance matrix. An AEKF and a MEKF with the same *a priori* estimates and with *a priori* covariances satisfying the following relationship,

$$\mathbf{P}_a^- = \boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}}^{\,-})\mathbf{P}_m^-\boldsymbol{\Xi}(\widehat{\overline{\mathbf{q}}}^{\,-})^{\mathrm{T}}, \tag{2.13}$$

then the additive Kalman gain is (dropping the arguments of the matrix functions)

$$\begin{aligned}\mathbf{K}_a &= \mathbf{P}_a^-\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_a^-\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} = \boldsymbol{\Xi}\mathbf{P}_m^-\boldsymbol{\Xi}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\boldsymbol{\Xi}\mathbf{P}_m^-\boldsymbol{\Xi}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} \\ &= \boldsymbol{\Xi}\mathbf{P}_m^-\mathbf{H}_m^{\mathrm{T}}(\mathbf{H}\mathbf{P}_m^-\mathbf{H}_m^{\mathrm{T}} + \mathbf{R})^{-1} = \boldsymbol{\Xi}\mathbf{K}_m.\end{aligned}$$

The filter residual is $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_a = \boldsymbol{\epsilon}_m = \mathbf{y} - \mathbf{h}(\widehat{\mathbf{q}})$, the *a posteriori* estimates are

$$\widehat{\mathbf{q}}_m^+ = \begin{bmatrix} \mathbf{K}_m\boldsymbol{\epsilon} \\ 1 \end{bmatrix} \otimes \widehat{\mathbf{q}}^- = \boldsymbol{\Xi}\mathbf{K}_m\boldsymbol{\epsilon} + \widehat{\mathbf{q}}^-, \text{ and}$$

$$\widehat{\mathbf{q}}_a^+ = \widehat{\mathbf{q}}^- + \mathbf{K}_a\boldsymbol{\epsilon} = \widehat{\mathbf{q}}^- + \boldsymbol{\Xi}\mathbf{K}_m\boldsymbol{\epsilon} = \widehat{\mathbf{q}}_m^+.$$

The *a posteriori* additive covariance is

$$\begin{aligned}\mathbf{P}_a^+ &= (\mathbf{I} - \mathbf{K}_a\mathbf{H}_a)\mathbf{P}_a^- = (\mathbf{I} - \boldsymbol{\Xi}\mathbf{K}_m\mathbf{H}_a)\boldsymbol{\Xi}\mathbf{P}_m^-\boldsymbol{\Xi}^{\mathrm{T}} = \boldsymbol{\Xi}(\mathbf{I} - \mathbf{K}_m\mathbf{H}_a\boldsymbol{\Xi})\mathbf{P}_m^-\boldsymbol{\Xi}^{\mathrm{T}}. \\ &= \boldsymbol{\Xi}(\mathbf{I} - \mathbf{K}_m\mathbf{H}_m)\mathbf{P}_m^-\boldsymbol{\Xi}^{\mathrm{T}} = \boldsymbol{\Xi}\mathbf{P}_m^+\boldsymbol{\Xi}^{\mathrm{T}}.\end{aligned}$$

It was shown that when the *a priori* covariances of the AEKF and MEKF are related through Eq. (2.13), the state update of the two algorithms is the same, and the *a posteriori* covariance obeys the same relationship.

In between measurements, the quaternion estimates evolve as

$$\frac{d}{dt}\widehat{\mathbf{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}(t)\right)\widehat{\mathbf{q}}(t).$$

The propagation of the covariance between measurements is given by

$$\begin{aligned}\dot{\mathbf{P}}_a(t) &= \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}(t)\right)\mathbf{P}_a(t) - \frac{1}{2}\mathbf{P}_a(t)\boldsymbol{\Omega}\left(\boldsymbol{\omega}(t)\right) + \mathbf{Q}_a(t); \qquad \mathbf{P}_a(t_k) = \mathbf{P}_a^+(t_k) \\ \dot{\mathbf{P}}_m(t) &= -[\boldsymbol{\omega}(t)\times]\mathbf{P}_m(t) + \mathbf{P}_m(t)[\boldsymbol{\omega}(t)\times] + \mathbf{Q}_m(t); \qquad \mathbf{P}_m(t_k) = \mathbf{P}_m^+(t_k)\end{aligned}$$

where $\mathbf{Q}_a$ and $\mathbf{Q}_m$ are the spectral densities of the process noise.

It is now going to be shown that if

$$\mathbf{Q}_a(t) = \boldsymbol{\Xi}\left(\widehat{\mathbf{q}}(t)\right)\mathbf{Q}_m(t)\,\boldsymbol{\Xi}^{\mathrm{T}}\left(\widehat{\mathbf{q}}(t)\right),$$

then

$$\mathbf{P}_a(t) = \boldsymbol{\Xi}\left(\widehat{\bar{\mathbf{q}}}(t)\right)\mathbf{P}_m(t)\boldsymbol{\Xi}^{\mathrm{T}}\left(\widehat{\bar{\mathbf{q}}}(t_k)\right). \tag{2.14}$$

Since the equality holds at the beginning of the propagation, it is sufficient to show that both sides of Eq. (2.14) have the same derivative. For convenience, let

$$\boldsymbol{\Xi} \triangleq \boldsymbol{\Xi}\left(\widehat{\bar{\mathbf{q}}}(t)\right), \quad [\boldsymbol{\omega}\times] \triangleq [\boldsymbol{\omega}(t)\times], \ \text{ and } \ \boldsymbol{\Omega} \triangleq \boldsymbol{\Omega}\left(\boldsymbol{\omega}(t)\right).$$

Taking the derivative of both sides of Eq. (2.14) yields

$$\begin{aligned}
&\frac{1}{2}\boldsymbol{\Omega}\,\mathbf{P}_a - \frac{1}{2}\mathbf{P}_a\boldsymbol{\Omega} + \mathbf{Q}_a = \\
&\qquad = \dot{\boldsymbol{\Xi}}\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} - \boldsymbol{\Xi}[\boldsymbol{\omega}\times]\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} + \boldsymbol{\Xi}\mathbf{P}_m[\boldsymbol{\omega}\times]\boldsymbol{\Xi}^{\mathrm{T}} + \boldsymbol{\Xi}\mathbf{Q}_m\boldsymbol{\Xi}^{\mathrm{T}} + \boldsymbol{\Xi}\mathbf{P}_m\dot{\boldsymbol{\Xi}}^{\mathrm{T}}
\end{aligned} \tag{2.15}$$

where

$$\begin{aligned}
\dot{\boldsymbol{\Xi}} = \boldsymbol{\Xi}\left(\dot{\widehat{\bar{\mathbf{q}}}}\right) = \frac{1}{2}\boldsymbol{\Xi}\left(\boldsymbol{\Omega}\,\widehat{\bar{\mathbf{q}}}\right) &= \frac{1}{2}\boldsymbol{\Xi}\left(\begin{bmatrix}\widehat{\mathbf{q}}\times\boldsymbol{\omega} + \widehat{q}\,\boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}}\widehat{\mathbf{q}}\end{bmatrix}\right) \\
&= \frac{1}{2}\begin{bmatrix}-\boldsymbol{\omega}^{\mathrm{T}}\widehat{\mathbf{q}}\,\mathbf{I}_{3\times3} + [(\widehat{\mathbf{q}}\times\boldsymbol{\omega})\times] + \widehat{q}[\boldsymbol{\omega}\times] \\ (\boldsymbol{\omega}\times\widehat{\mathbf{q}})^{\mathrm{T}} - \widehat{q}\,\boldsymbol{\omega}^{\mathrm{T}}\end{bmatrix}.
\end{aligned}$$

Utilizing the identities from Eq. (2.3), it follows that

$$[(\widehat{\mathbf{q}}\times\boldsymbol{\omega})\times] = \boldsymbol{\omega}\widehat{\mathbf{q}}^{\mathrm{T}} - \widehat{\mathbf{q}}\boldsymbol{\omega}^{\mathrm{T}}, \qquad [\widehat{\mathbf{q}}\times]\,[\boldsymbol{\omega}\times] = \boldsymbol{\omega}\widehat{\mathbf{q}}^{\mathrm{T}} - \boldsymbol{\omega}^{\mathrm{T}}\widehat{\mathbf{q}}\mathbf{I}_{3\times3},$$

it follows that

$$
\begin{aligned}
\dot{\boldsymbol{\Xi}} &= \frac{1}{2}\begin{bmatrix} -\boldsymbol{\omega}^{\mathrm{T}}\widehat{\mathbf{q}}\,\mathbf{I}I_{3\times 3} + \boldsymbol{\omega}\widehat{\mathbf{q}}^{\mathrm{T}} - \widehat{\mathbf{q}}\boldsymbol{\omega}^{\mathrm{T}} + \widehat{q}[\boldsymbol{\omega}\times] \\ -\widehat{\mathbf{q}}^{\mathrm{T}}[\boldsymbol{\omega}\times] - \widehat{q}\,\boldsymbol{\omega}^{\mathrm{T}} \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} -\boldsymbol{\omega}^{\mathrm{T}}\widehat{\mathbf{q}}\,\mathbf{I}I_{3\times 3} + \boldsymbol{\omega}\widehat{\mathbf{q}}^{\mathrm{T}} + \widehat{q}[\boldsymbol{\omega}\times] \\ -\widehat{\mathbf{q}}^{\mathrm{T}}[\boldsymbol{\omega}\times] \end{bmatrix} - \frac{1}{2}\widehat{\bar{\mathbf{q}}}\,\boldsymbol{\omega}^{\mathrm{T}} \\
&= \frac{1}{2}\begin{bmatrix} -[\widehat{\mathbf{q}}\times]\cdot[\boldsymbol{\omega}\times] + \widehat{q}[\boldsymbol{\omega}\times] \\ -\widehat{\mathbf{q}}^{\mathrm{T}}[\boldsymbol{\omega}\times] \end{bmatrix} - \frac{1}{2}\widehat{\bar{\mathbf{q}}}\,\boldsymbol{\omega}^{\mathrm{T}} = \ \frac{1}{2}\boldsymbol{\Xi}[\boldsymbol{\omega}\times] - \frac{1}{2}\widehat{\bar{\mathbf{q}}}\,\boldsymbol{\omega}^{\mathrm{T}}, \qquad (2.16)
\end{aligned}
$$

and

$$
\boldsymbol{\Omega}\,\boldsymbol{\Xi} = \begin{bmatrix} -\widehat{q}[\boldsymbol{\omega}\times] - [\boldsymbol{\omega}\times]\cdot[\widehat{\mathbf{q}}\times] - \boldsymbol{\omega}\widehat{\mathbf{q}}^{\mathrm{T}} \\ -\widehat{q}\boldsymbol{\omega}^{\mathrm{T}} - \boldsymbol{\omega}^{\mathrm{T}}[\widehat{\mathbf{q}}\times] \end{bmatrix} = -\boldsymbol{\Xi}\,[\boldsymbol{\omega}\times] - \widehat{\bar{\mathbf{q}}}\boldsymbol{\omega}^{\mathrm{T}}. \qquad (2.17)
$$

Substituting Eq. (2.16) in Eq. (2.15) yields

$$
\begin{aligned}
\frac{1}{2}\boldsymbol{\Omega}\,\mathbf{P}_a - \frac{1}{2}\mathbf{P}_a\boldsymbol{\Omega} &= \frac{1}{2}\boldsymbol{\Xi}[\boldsymbol{\omega}\times]\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} - \frac{1}{2}\widehat{\bar{\mathbf{q}}}\,\boldsymbol{\omega}^{\mathrm{T}}\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} - \boldsymbol{\Xi}[\boldsymbol{\omega}\times]\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} + \\
&\quad + \boldsymbol{\Xi}\mathbf{P}_m[\boldsymbol{\omega}\times]\boldsymbol{\Xi}^{\mathrm{T}} - \frac{1}{2}\boldsymbol{\Xi}\mathbf{P}_m[\boldsymbol{\omega}\times]\boldsymbol{\Xi}^{\mathrm{T}} - \frac{1}{2}\boldsymbol{\Xi}\mathbf{P}_m\boldsymbol{\omega}\widehat{\bar{\mathbf{q}}}^{\mathrm{T}} \\
&= \frac{1}{2}\left\{-\boldsymbol{\Xi}[\boldsymbol{\omega}\times] - \widehat{\bar{\mathbf{q}}}\,\boldsymbol{\omega}^{\mathrm{T}}\right\}\mathbf{P}_m\boldsymbol{\Xi}^{\mathrm{T}} - \frac{1}{2}\boldsymbol{\Xi}\mathbf{P}_m\left\{[\boldsymbol{\omega}\times]\boldsymbol{\Xi}^{\mathrm{T}} - \boldsymbol{\omega}\widehat{\bar{\mathbf{q}}}^{\mathrm{T}}\right\}
\end{aligned}
$$

which is equivalent because of Eq. (2.17) and Eq. (2.14). The proof is complete.

The above arguments show that every MEKF is equivalent to an AEKF. It does not show the converse. A MEKF designed with

$$
\widehat{\bar{\mathbf{q}}}_m(t_0), \qquad \mathbf{P}_m(t_0), \qquad \mathbf{R}_{m,k}, \quad \text{and } \mathbf{Q}_m(t)
$$

is equivalent to an AEKF with

$$
\begin{aligned}
&\widehat{\bar{\mathbf{q}}}_a(t_0) = \widehat{\bar{\mathbf{q}}}_m(t_0), \quad \mathbf{P}_a(t_0) = \boldsymbol{\Xi}\left(\widehat{\bar{\mathbf{q}}}_0\right)\mathbf{P}_m(t_0)\boldsymbol{\Xi}^{\mathrm{T}}\left(\widehat{\bar{\mathbf{q}}}_0\right), \quad \mathbf{R}_{a,k} = \mathbf{R}_{m,k}, \ \text{ and} \\
&\mathbf{Q}_a(t) = \boldsymbol{\Xi}\left(\widehat{\bar{\mathbf{q}}}(t)\right)\mathbf{Q}_m(t)\,\boldsymbol{\Xi}^{\mathrm{T}}\left(\widehat{\bar{\mathbf{q}}}(t)\right).
\end{aligned}
$$

Every MEKF is equivalent to an AEKF with a singular covariance matrix. This does not imply that every AEKF has a singular covariance. It is sufficient to choose a nonsingular $\mathbf{P}_a(t_0)$. It would be reasonable to ask whether the converse is true: it is possible to design an MEKF equivalent to any given (non-singular) AEKF? The answer is no.

Once more the *a posteriori* estimates are

$$\begin{aligned}\widehat{\mathbf{q}}_m^+ &= \widehat{\mathbf{q}}^- + \mathbf{\Xi}\mathbf{K}_m\boldsymbol{\epsilon}, \quad \text{and}\\ \widehat{\mathbf{q}}_a^+ &= \widehat{\mathbf{q}}^- + \mathbf{K}_a\boldsymbol{\epsilon}.\end{aligned}$$

The filters give the same estimate if

$$\begin{aligned}&(\mathbf{\Xi}\mathbf{K}_m - \mathbf{K}_a)\boldsymbol{\epsilon} = 0,\\ &[\mathbf{\Xi}\mathbf{P}_m\mathbf{H}_m^{\mathrm{T}}(\mathbf{H}_m\mathbf{P}_m\mathbf{H}_m^{\mathrm{T}} + \mathbf{R})^{-1} - \mathbf{P}_a\mathbf{H}_a^{\mathrm{T}}(\mathbf{H}_a\mathbf{P}_a\mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1}]\boldsymbol{\epsilon} = 0, \quad \text{and}\\ &[\mathbf{\Xi}\mathbf{P}_m\mathbf{\Xi}^{\mathrm{T}}\mathbf{H}_a^{\mathrm{T}}(\mathbf{H}_a\mathbf{\Xi}\mathbf{P}_m\mathbf{\Xi}^{\mathrm{T}}\mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1} - \mathbf{P}_a\mathbf{H}_a^{\mathrm{T}}(\mathbf{H}_a\mathbf{P}_a\mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1}]\boldsymbol{\epsilon} = 0.\end{aligned}$$

Unless the residuals have an unusual structure, it is impossible that every realization belongs to the null space of the same matrix, therefore the term in brackets must be zero. If $\mathrm{rank}(\mathbf{P}_a) = 4$

$$\mathbf{\Xi}\mathbf{P}_m\mathbf{\Xi}^{\mathrm{T}} \neq \mathbf{P}_a \quad \forall\mathbf{P}_m.$$

The following equation should be considered

$$\mathbf{\Xi}\mathbf{P}_m\mathbf{\Xi}^{\mathrm{T}}\mathbf{H}_a^{\mathrm{T}}(\mathbf{H}_a\mathbf{\Xi}\mathbf{P}_m\mathbf{\Xi}^{\mathrm{T}}\mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1} = \mathbf{P}_a\mathbf{H}_a^{\mathrm{T}}(\mathbf{H}_a\mathbf{P}_a\mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1}. \tag{2.18}$$

Solving Eq. (2.18) is not always possible. Assume, for example, that the initial orientation is along the reference frame and the initial covariance is

$$\mathbf{P}_a = \kappa \mathbf{I}_{4\times 4}; \qquad \boldsymbol{\Xi} = \begin{bmatrix} \mathbf{I}_{3\times 3} \\ \mathbf{O}_{1\times 3} \end{bmatrix},$$

where κ is a given positive number. Then Eq. (2.18) becomes

$$\begin{bmatrix} \mathbf{P}_m & \mathbf{O}_{3\times 1} \\ \mathbf{O}_{1\times 3} & 0 \end{bmatrix} \mathbf{H}_a^{\mathrm{T}} \left(\mathbf{H}_a \begin{bmatrix} \mathbf{P}_m & \mathbf{O}_{3\times 1} \\ \mathbf{O}_{1\times 3} & 0 \end{bmatrix} \mathbf{H}_a^{\mathrm{T}} + \mathbf{R} \right)^{-1} = \kappa \mathbf{H}_a^{\mathrm{T}} (\kappa \mathbf{H}_a \mathbf{H}_a^{\mathrm{T}} + \mathbf{R})^{-1}.$$

It can be seen that no $\mathbf{P}_m$ will work because the fourth row of the left side of the equation is always going to be zero.

In summary, every design of a MEKF corresponds to a singular AEKF, while not every non-singular AEKF corresponds to a MEKF. Researchers agree that the covariance of the AEKF does not need to be strictly singular [11]. Therefore, there are many possible AEKFs that are not equivalent to a corresponding MEKF.

2.2.4 Normalization

Both multiplicative and additive approaches provide estimates with unit norm to first order [8] with respect to the estimation error. It will be proven that in the multiplicative approach the *a posteriori* estimate norm is always greater than the *a priori* norm, and that the estimate norm is unchanged through propagation. As a consequence, brute force normalization is essential in the multiplicative update to avoid the norm of the estimate becoming arbitrarily large. Of course this can be avoided by using the full nonlinear transformation between the three-dimensional representation of the attitude error and the quaternion. The downside would be that the estimate will depend on the parametrization (rotation vector, Gibbs vector, modified Rodrigues parameters) which is counterintuitive. This approach is referred to as rotational [11].

The *a posteriori* estimate is given by

$$\widehat{\overline{\mathbf{q}}}^{+} = \delta\widehat{\overline{\mathbf{q}}} \otimes \widehat{\overline{\mathbf{q}}}^{-} = \begin{bmatrix} \delta\widehat{q}\widehat{\mathbf{q}}^{-} - [\delta\widehat{\mathbf{q}}\times]\widehat{\mathbf{q}}^{-} + \widehat{q}^{-}\delta\widehat{\mathbf{q}} \\ -\delta\widehat{\mathbf{q}}^{\mathrm{T}}\widehat{\mathbf{q}}^{-} + \delta\widehat{q}\,\widehat{q}^{-} \end{bmatrix}.$$

The norm can be computed to be

$$\|\widehat{\overline{\mathbf{q}}}^{+}\|^2 = \|\widehat{\overline{\mathbf{q}}}^{-}\|^2 \cdot \|\delta\widehat{\overline{\mathbf{q}}}\|^2, \qquad \text{and}$$

$$\|\delta\widehat{\overline{\mathbf{q}}}\|^2 = 1 + \|\delta\widehat{\mathbf{q}}\|^2 > 1.$$

Therefore, the *a posteriori* estimate norm is always greater than the *a priori* estimate norm. During propagation, the norm remains unchanged since the quadratic form of a skew-symmetric matrix is always zero,

$$\frac{d}{dt}\|\overline{\mathbf{q}}\|^2 = \frac{d}{dt}\left(\overline{\mathbf{q}}^{\mathrm{T}}\overline{\mathbf{q}}\right) = 2\overline{\mathbf{q}}^{\mathrm{T}}\dot{\overline{\mathbf{q}}} = 2\overline{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\Omega}(\omega)\overline{\mathbf{q}} = 0.$$

The square of the norm remaining constant implies that the norm remains constant since the norm is always positive.

In summary, it was shown that at every update the norm of the estimate increases, while during propagation the estimate norm remains the same. Hence the norm of the estimate of a MEKF will constantly increase. To avoid this situation, it is necessary to normalize the estimate.

The rate at which the estimate norm increases can be reduced by using the second order MEKF, in which

$$\delta\widehat{\overline{\mathbf{q}}} = \begin{bmatrix} \delta\widehat{\mathbf{q}} \\ 1 - \|\delta\widehat{\mathbf{q}}\|^2/2 \end{bmatrix}.$$

The norm is still greater than one,

$$\|\delta\widehat{\mathbf{q}}\|^2 = \|\delta\widehat{\mathbf{q}}\|^2 + 1 + \|\delta\widehat{\mathbf{q}}\|^4/4 - \|\delta\widehat{\mathbf{q}}\|^2 = 1 + \|\delta\widehat{\mathbf{q}}\|^4/4 > 1.$$

Therefore normalization is still necessary. In the MEKF, the error on the norm is of first order (or second), but always positive, and the cumulative effect after many updates could result in large deviations from unitary norm. In the AEKF, the error in the norm could be either positive or negative, making the normalization necessary but less crucial after some time. Figure 2.2 shows the evolution of the norm in the additive and multiplicative case if the normalization was not enforced after each update. Estimating the quaternion without normalization is not recommended [7]. However, from Fig. 2.2 it should be clear that normalization is an essential part of the MEKF as it is for the AEKF.

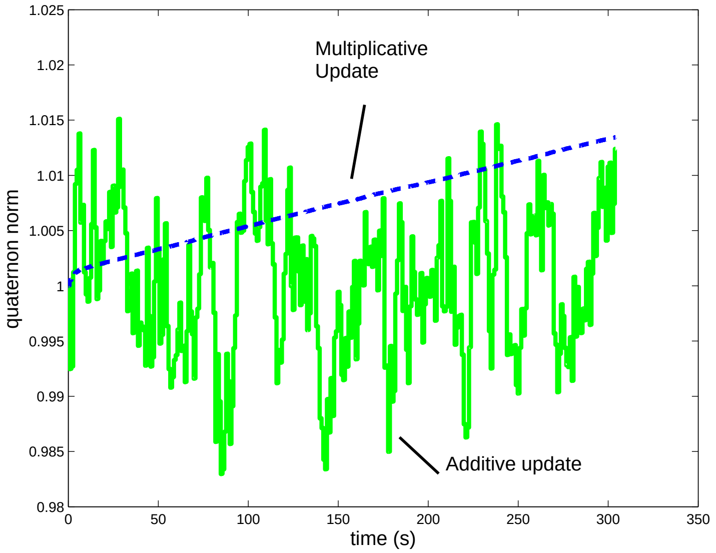


Figure 2.2: Norm evolution without brute force normalization. Dashed line is the multiplicative update, continuous line is the additive update.

Having the norm of the quaternion greater than one results in stretching and not only rotating a vector when the vector is pre-multiplied by $\mathbf{T}(\overline{\mathbf{q}})$. In the accelerometer dead-reckoning navigation scheme, the non-gravitational acceleration measurement is rotated in the inertial frame. It is evident that having the quaternion always greater than one will result in an a bigger effective acceleration measurement, which could rapidly cause divergence because of the high sampling rate of IMUs.

It was shown that defining a physically meaningful attitude estimation error leads to a Kalman filter formulation equivalent to the one defining the error in standard way, which is the Mean Square Error. The same conclusion was reached with a very different approach [59]. Having a small mean square error is desirable, independent of the possible physical interpretation. Arguments were presented to provide an explanation of the fact that in practical applications, the covariance of the AEKF does not become ill-conditioned when process noise is present. Finally, it was shown that brute force normalization is fundamental in both schemes. Theoretically, the norm constraint could be enforced with a perfect measurement of the state norm. A perfect measurement would produce a singular covariance matrix, where the singularity is a byproduct of the linearization process [11] and does not signify that the covariance in the AEKF is singular. A stochastic justification of brute force normalization will be presented next.

2.2.5 Norm Constrained Kalman Filtering

In this section, it will be shown that brute force normalization is optimal in a MSE sense, not only in a geometrical sense as previously shown [52]. Normalization is a nonlinear transformation, therefore similar approximations to those associated with the extended Kalman filter will be made. Optimality does not hold strictly, but conditionally on the above approximations.

Define the *a priori* state estimate $\widehat{\mathbf{x}}_k^-$ to be the state estimate at time t_k just prior to employing the measurement $\mathbf{y}_k$ in the state estimate update algorithm.

Similarly, define the *a posteriori* state estimate $\widehat{\mathbf{x}}_k^+$ to be the state estimate at time t_k just after the state estimate update. The performance index is defined as

$$\mathcal{J}_k = E\left[\left(\mathbf{e}_k^+\right)^{\mathrm{T}} \mathbf{e}_k^+\right], \tag{2.19}$$

where the *a priori* and *a posteriori* estimation errors are given by

$$\begin{aligned} \mathbf{e}_k^- &= \mathbf{x}_k - \widehat{\mathbf{x}}_k^-, \quad \text{and} \\ \mathbf{e}_k^+ &= \mathbf{x}_k - \widehat{\mathbf{x}}_k^+, \end{aligned}$$

respectively. Associated with the estimation errors, define the matrices

$$\begin{aligned} \mathbf{P}_k^- &= \mathrm{E}\left\{\mathbf{e}_k^- \left(\mathbf{e}_k^-\right)^{\mathrm{T}}\right\}, \quad \text{and} \\ \mathbf{P}_k^+ &= \mathrm{E}\left\{\mathbf{e}_k^+ \left(\mathbf{e}_k^+\right)^{\mathrm{T}}\right\}, \end{aligned}$$

before and after the measurement update, respectively. Note that

$$\mathcal{J}_k = \operatorname{trace} \mathbf{P}_k^+ \ .$$

The norm of the state vector is desired to have a predefined value

$$\|\widehat{\mathbf{x}}_k^+\| = \sqrt{l}.$$

This constraint is equivalent to the following scalar quadratic representation

$$(\widehat{\mathbf{x}}_k^+)^{\mathrm{T}} \widehat{\mathbf{x}}_k^+ = l. \tag{2.20}$$

The update is

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\epsilon}_k,$$

where $\boldsymbol{\epsilon}_k = \mathbf{y}_k - \widehat{\mathbf{y}}_k$ is the residual. Substituting the residual into Eq. (2.20), the state constraint can be expressed more conveniently as a control constraint:

$$\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{K}_k^{\mathrm{T}} \mathbf{K}_k \boldsymbol{\epsilon}_k + 2\widehat{\mathbf{x}}_k^{\mathrm{T}} \mathbf{K}_k \boldsymbol{\epsilon}_k + \widehat{\mathbf{x}}_k^{\mathrm{T}} \widehat{\mathbf{x}}_k - l = 0. \tag{2.21}$$

The goal is to find the gain $\mathbf{K}_k$ such that Eq. (2.19) is minimized and the constraint given by Eq. (2.21) is satisfied.

First-Order Condition

The *a posteriori* error mean square is given by[†]

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}},$$

where $\mathbf{P}_k^-$ is the *a priori* state error mean square. Define

$$\mathbf{W}_k \triangleq \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k.$$

Therefore, the Joseph formula can be rewritten as

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^{\mathrm{T}}.$$

The performance index to be minimized is then given by

$$\mathcal{J}_k = \mathrm{trace} \left[\mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^{\mathrm{T}} \right].$$

The Kalman gain should be computed to satisfy the constraint in Eq. (2.21). Matrix $\mathbf{P}_k^-$ is $n \times n$, $\mathbf{K}_k$ is $m \times n$, l is a scalar and the remaining are of appropriate dimensions.

[†] see page 43.

The augmented performance index is

$$\begin{aligned}\mathcal{J}_k = &\operatorname{trace}\left[\mathbf{P}_k^- - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k\mathbf{W}_k\mathbf{K}_k^{\mathrm{T}}\right] + \\ &+ \lambda_k\Big[\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\widehat{\mathbf{x}}_k^- + 2\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\widehat{\mathbf{x}}_k^- + \boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\mathbf{K}_k\boldsymbol{\epsilon}_k - l\Big].\end{aligned}$$

The $n \times m + 1$ optimal values of λ_k and $\mathbf{K}_k$ are obtained solving the $n \times m$ equations resulting from taking the derivative of $\mathcal{J}_k$ with respect to $\mathbf{K}_k$ and setting it to zero. The equations are

$$-2\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + 2\mathbf{K}_k\mathbf{W}_k + 2\lambda_k(\widehat{\mathbf{x}}_k^-\boldsymbol{\epsilon}_k^{\mathrm{T}} + \mathbf{K}_k\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^{\mathrm{T}}) = \mathbf{O}, \tag{2.22}$$

and the scalar constraint Eq. (2.21). Equation (2.22) can be rewritten to obtain the following first-order conditions:

$$\begin{aligned}&\mathbf{K}_k = (\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} - \lambda_k\widehat{\mathbf{x}}_k^-\boldsymbol{\epsilon}_k^{\mathrm{T}})(\mathbf{W}_k + \lambda_k\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^{\mathrm{T}})^{-1}, \quad \text{and} \\ &\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\mathbf{K}_k\boldsymbol{\epsilon}_k + 2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\mathbf{K}_k\boldsymbol{\epsilon}_k + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\left(\widehat{\mathbf{x}}_k^-\right) - l = 0.\end{aligned}$$

Using the matrix inversion lemma, it follows that

$$\begin{aligned}\mathbf{K}_k = &\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1} - \lambda_k\widehat{\mathbf{x}}_k^-\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1} - \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\frac{\lambda_k\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}}{1 + \lambda_k\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\boldsymbol{\epsilon}_k} + \\ &+ \lambda_k\widehat{\mathbf{x}}_k^-\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\frac{\lambda_k\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}}{1 + \lambda_k\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\boldsymbol{\epsilon}_k}.\end{aligned}$$

Substituting into Eq. (2.21), after some manipulations, the following scalar equation with the scalar unknown λ_k is obtained

$$\begin{aligned}&\lambda_k^2\widetilde{\epsilon}_k^2\Big(-\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\widehat{\mathbf{x}}_k^- + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\left(\widehat{\mathbf{x}}_k^-\right) - l\Big) + \lambda_k\widetilde{\epsilon}_k\Big(-2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\widehat{\mathbf{x}}_k^- + 2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\left(\widehat{\mathbf{x}}_k^-\right) - 2l\Big) \\ &+ \Big(\boldsymbol{\epsilon}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\mathbf{H}_k\mathbf{P}_k^-\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\boldsymbol{\epsilon}_k + 2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1}\boldsymbol{\epsilon}_k + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}}\left(\widehat{\mathbf{x}}_k^-\right) - l\Big) = 0,\end{aligned}$$

where

$$\widetilde{\epsilon}_k = \boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k.$$

Therefore, the optimal Lagrange multiplier is

$$\lambda_k = \frac{-b/2 \pm \sqrt{b^2/4 - ac}}{a},$$

where

$$\begin{aligned}
&a = -l\widetilde{\epsilon}_k^2, \qquad b = -2\widetilde{\epsilon}_k l, \text{ and} \\
&c = \boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k + 2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}} \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}} \left(\widehat{\mathbf{x}}_k^-\right) - l.
\end{aligned}$$

Finally, it follows that

$$\lambda_k = \frac{\widetilde{\epsilon}_k l \pm \sqrt{\widetilde{\epsilon}_k^2 l^2 + l\widetilde{\epsilon}_k^2 c}}{-l\widetilde{\epsilon}_k^2} = \frac{1 \pm \sqrt{1 + c/l}}{-\widetilde{\epsilon}_k}.$$

Notice that

$$\begin{aligned}
1 + c/l &= (\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k + 2\left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}} \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}} \left(\widehat{\mathbf{x}}_k^-\right))/l \\
&= (\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}})^{\mathrm{T}} (\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}})/l \geq 0.
\end{aligned}$$

Therefore, λ_k is always a real number and can be rewritten as

$$\lambda_k = \frac{-1}{\widetilde{\epsilon}_k} \pm \frac{\|\widehat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \boldsymbol{\epsilon}_k\|}{\widetilde{\epsilon}_k \sqrt{l}}.$$

Second-Order Condition

Taking the second derivative of the performance index presents some representation issues. Each of the entries of the first derivative could be differentiated again, but this approach will result in $m \times n$ matrix equations. Another approach would be to perturb the gain and show that the perturbation results in an increment of the performance index.

In the case of scalar measurement, the gain $\mathbf{K}_k$ can be partitioned as

$$\mathbf{K}_k = \begin{bmatrix} \mathbf{k}_k \\ k_k \end{bmatrix},$$

where k_k is a scalar. The constraint becomes

$$\epsilon_k^2 \mathbf{k}_k^{\mathrm{T}} \mathbf{k}_k + \epsilon_k^2 k_k^2 + 2\epsilon_k \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \mathbf{k}_k + 2\epsilon_k x_k^- k_k + \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \widehat{\boldsymbol{\chi}}_k^- - l = 0,$$

where

$$\widehat{\mathbf{x}}_k^- = \begin{bmatrix} \widehat{\boldsymbol{\chi}}_k^- \\ \widehat{x}_k^- \end{bmatrix}.$$

Differentiating the constraint, yields

$$2(\epsilon_k^2 \mathbf{k}_k^{\mathrm{T}} + \epsilon_k \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}}) d\mathbf{k}_k + 2(\epsilon_k^2 k_k + \epsilon_k \widehat{x}_k^-) dk_k = 0.$$

Assuming the residual is not zero (if the residual is zero the *a posteriori* estimate is always equal to the *a priori* estimate), it follows that

$$dk_k = -\frac{\epsilon_k \mathbf{k}_k^{\mathrm{T}} + \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}}}{\epsilon_k k_k + \widehat{x}_k^-} d\mathbf{k}_k.$$

The second-order differential is

$$\begin{aligned}
d\mathcal{J}_k^2 &= d\mathbf{K}_k^{\mathrm{T}} G_{KK} d\mathbf{K}_k, \\
G_{KK} &= 2W_k + 2\lambda_k \epsilon_k^2, \text{ and} \\
d\mathcal{J}_k^2 &= (2W_k + 2\lambda_k \epsilon_k^2)(d\mathbf{K}_k^T d\mathbf{K}_k) = (2W_k + 2\lambda_k \epsilon_k^2)(d\mathbf{k}_k^T d\mathbf{k}_k + dk_n^2) \\
&= 2(W_k + \lambda_k \epsilon_k^2) d\mathbf{k}_k^{\mathrm{T}} \left(\mathbf{I} + \frac{\epsilon_k \mathbf{k}_k + \widehat{\boldsymbol{\chi}}_k^-}{\epsilon_k k_k + \widehat{x}_k^-} \, \frac{\epsilon_k \mathbf{k}_k^{\mathrm{T}} + \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}}}{\epsilon k_k + \widehat{x}_k^-} \right) d\mathbf{k}_k.
\end{aligned}$$

The sufficient condition for a minimum is

$$\frac{2(W_k + \lambda_k \epsilon_k^2)}{(\epsilon_k k_k + \widehat{x}_k^-)^2} \left[(\epsilon_k k_k + \widehat{x}_k^-)^2 \mathbf{I} + \epsilon_k^2 \mathbf{k}_k \mathbf{k}_k^{\mathrm{T}} + \epsilon_k \widehat{\boldsymbol{\chi}}_k^- \mathbf{K}_k^{\mathrm{T}} + \epsilon_k \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} + \epsilon_k \mathbf{k}_k \widehat{\boldsymbol{\chi}}_k^- \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \right] > 0$$

$$\mathbf{k}_k = \frac{\left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} - \lambda_k \epsilon_k \widehat{\boldsymbol{\chi}}_k^-}{W_k + \lambda_k \epsilon_k^2}, \qquad k_k = \frac{\mathbf{p}^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} - \lambda_k \epsilon_k \widehat{x}_k^-}{W_k + \lambda_k \epsilon_k^2}, \qquad \mathbf{P}_k^- = \left[\widetilde{\mathbf{P}}_k^- \;\; \mathbf{p}_k \right].$$

Substituting in the gain and eliminating positive scalars, yields

$$\begin{aligned}
&(W_k + \lambda_k \epsilon_k^2) \Bigg\{ (\epsilon_k k_k + \widehat{x}_k^-)^2 \mathbf{I} + \frac{\epsilon_k^2}{(W_k + \lambda_k \epsilon_k^2)^2} \Bigg[\left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \mathbf{H}_k \widetilde{\mathbf{P}}_k^- - \lambda_k \epsilon_k \widehat{\boldsymbol{\chi}}_k^- \mathbf{H}_k \widetilde{\mathbf{P}}_k^- + \\
&- \lambda_k \epsilon_k \left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} + \lambda_k^2 \epsilon_k^2 \widehat{\boldsymbol{\chi}}_k^- \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \Bigg] + \frac{r}{W_k + \lambda_k \epsilon_k^2} \\
&\left[\left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} + \widehat{\boldsymbol{\chi}}_k^- \mathbf{H}_k \widetilde{\mathbf{P}}_k^- - 2\lambda_k \epsilon \widehat{\boldsymbol{\chi}}_k^- \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \right] + \widehat{\boldsymbol{\chi}}_k^- \left(\widehat{\boldsymbol{\chi}}_k^-\right)^{\mathrm{T}} \Bigg\} > 0.
\end{aligned}$$

An equivalent condition is

$$\begin{aligned}
(W_k + \lambda_k \epsilon_k^2) \Bigg\{ &(\epsilon_k k_k + \widehat{x}_k^-)^2 (W_k + \lambda_k \epsilon_k^2)^2 \mathbf{I} + \left[\epsilon_k \left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} + W_k \widehat{\boldsymbol{\chi}}_k^- \right] \\
&\left[\epsilon_k \left(\widetilde{\mathbf{P}}_k^-\right)^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} + W_k \widehat{\boldsymbol{\chi}}_k^- \right]^{\mathrm{T}} \Bigg\} > 0.
\end{aligned}$$

The matrix in brackets is of the form

$$\mu^2\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}},$$

which is positive definite when $\mu \neq 0$. As a consequence, the optimal gain produces a minimum performance index when the scalar $W_k + \lambda_k \epsilon_k^2$ is positive. Since

$$W_k + \lambda_k \epsilon_k^2 = \pm\sqrt{W_k^2 + c/l},$$

the minimum occurs when the plus sign is chosen for the Lagrange multiplier. Also, if the minus sign is chosen, the performance index will be maximized. The same arguments hold true when the measurement is a vector.

Constrained Minimum Solution

The performance index is minimized and the constraint is satisfied when the optimal gain is chosen as

$$\begin{aligned}\mathbf{K}_k^* &= (\mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} - \lambda_k \widehat{\mathbf{x}}_k^- \boldsymbol{\epsilon}_k^{\mathrm{T}})(\mathbf{W}_k + \lambda_k \boldsymbol{\epsilon}_k \boldsymbol{\epsilon}_k^{\mathrm{T}})^{-1}, \qquad \text{where} \\ \lambda_k &= \frac{-1}{\widetilde{\epsilon}_k} + \frac{\|\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- + \left(\widehat{\mathbf{x}}_k^-\right)^{\mathrm{T}} \|}{\widetilde{\epsilon}_k \sqrt{l}}.\end{aligned}$$

The asterisk in $\mathbf{K}_k^*$ was added to distinguish from the unconstrained Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k \mathbf{W}_k^{-1}.$$

The unconstrained *a posteriori* estimate is $\widehat{\mathbf{x}}_k^+$

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\epsilon}_k.$$

The minimizing constrained gain can be rewritten as

$$\mathbf{K}_k^* = \mathbf{K}_k + \left(\frac{\sqrt{l}}{\|\widehat{\mathbf{x}}_k^+\|} - 1 \right) \widehat{\mathbf{x}}_k^+ \frac{\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1}}{\widetilde{\epsilon}_k}.$$

Property 1. *The optimal constrained solution shares the same direction as the optimal unconstrained solution.*

Proof. Let $\widehat{\mathbf{x}}_k^*$ be the optimal constrained estimate. Then it follows that

$$\widehat{\mathbf{x}}_k^* = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k^* \boldsymbol{\epsilon}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\epsilon}_k + \left(\frac{\sqrt{l}}{\|\widehat{\mathbf{x}}_k^+\|} - 1 \right) \widehat{\mathbf{x}}_k^+ \frac{\boldsymbol{\epsilon}_k^{\mathrm{T}} \mathbf{W}_k^{-1}}{\widetilde{\epsilon}_k} \boldsymbol{\epsilon}_k = \frac{\sqrt{l}}{\|\widehat{\mathbf{x}}_k^+\|} \widehat{\mathbf{x}}_k^+.$$

□

So $\widehat{\mathbf{x}}_k^*$ and $\widehat{\mathbf{x}}_k^+$ have the same direction, but different magnitude. Property 1 states that brute force normalization is optimal not only in a geometrical sense, but also in a Mean Square Error sense.

The *a posteriori* estimation error is

$$\mathbf{e}^* = (\mathbf{I} - \mathbf{K}_k^* \mathbf{H}) \mathbf{e}^- + \mathbf{K}_k^* \boldsymbol{\eta}_k.$$

Under the assumption that measurement noise is independent of process noise and initial estimation error, it follows that

$$\mathbf{P}_k^* = \mathrm{E} \left\{ (\mathbf{I} - \mathbf{K}_k^* \mathbf{H}_k) \mathbf{e}^- (\mathbf{e}^-)^{\mathrm{T}} (\mathbf{I} - \mathbf{K}_k^* \mathbf{H}_k)^{\mathrm{T}} \right\} + \mathrm{E} \left\{ \mathbf{K}_k^* \boldsymbol{\eta}_k \boldsymbol{\eta}_k^{\mathrm{T}} (\mathbf{K}^*)^{\mathrm{T}} \right\}. \tag{2.23}$$

The optimal gain is a function of the *a priori* state and the residual, therefore it is a random variable and it should not be taken outside the expectation operator. A similar situation happens in nonlinear Kalman filtering. In the extended Kalman filter, for example, the measurement mapping matrix is a function of the *a priori* state, thus making the gain a function of the *a priori* state as well. The Kalman gain is taken out of the expectation sign, following the EKF solution

$$\mathbf{P}_k^* = (\mathbf{I} - \mathbf{K}_k^*\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k^*\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k^*\mathbf{R}_k\left(\mathbf{K}_k^*\right)^{\mathrm{T}}.$$

Substituting for $\mathbf{K}_k^*$ yields

$$\mathbf{P}_k^* = \mathbf{P}_k^+ + \frac{1}{\widetilde{\epsilon}_k}\left(1 - \frac{\sqrt{l}}{\|\widehat{\mathbf{x}}_k^+\|}\right)^2 \widehat{\mathbf{x}}_k^+\left(\widehat{\mathbf{x}}_k^+\right)^{\mathrm{T}}, \tag{2.24}$$

which is very similar to the correction given by Choukroun *et al.* [60].

When two random variables are related through a nonlinear transformation, it is generally impossible to relate exclusively their second moments but all the moments of the original variable will contribute to the second moment of the transformed variable. Therefore, the correction of the covariance can be accurate or not depending on the distribution. Both the AEKF and MEKF provide estimates with unit norm to first-order [8], therefore the unmodified covariance $\mathbf{P}_k^+$ is an approximation accurate to first-order.

The matrix $\mathbf{P}_k^*$ is an approximation, and like any approximation, might not be satisfactory under certain circumstances. From Eq. (2.24) it can be seen that $\mathbf{P}_k^*$ can be unsatisfactory for small $\widetilde{\epsilon}_k$ and large norm errors of the unconstrained estimate. This situation could arise, for example, in the presence of scalar measurement when the estimation error is large.

The scope of this section was to demonstrate that brute force normalization is optimal in a stochastic sense. The goal was not to derive a correction to the additive *covariance* since a well-performing correction already exists [60]. Also, it is the belief of the author that no correction should be performed to the covariance of the AEKF, because the covariance is accurate to first-order, and so is the EKF. If the AEKF necessitates a covariance adjustment, so does the first-order MEKF since brute force normalization affects the multiplicative error as well.

2.2.6 Conclusions

The arguments previously presented aim to show that the additive and multiplicative approach have equal dignity, that the AEKF does not need to possess a singular covariance, and that brute force normalization is optimal under standard nonlinear filtering assumptions. The multiplicative approach has the advantages of a covariance with smaller dimension and easy physical interpretation of the error, but is otherwise not superior to the additive approach. The other advantage of the multiplicative approach, which is also the reason why it is going to be used for the remaining of this work, is that matrix $\mathbf{P}_m$ is indeed a covariance. This fact makes the tuning of the filter very intuitive and the display of the results very immediate because the estimation errors are angles (or half angles).

2.3 Davenport Solution to the Wahba Problem

The Wahba problem [61] consists in determining the orthogonal matrix $\mathbf{T}$ that minimizes the performance index

$$\mathcal{J}(\overline{\mathbf{q}}) = \frac{1}{2}\sum_{i=1}^{n} w_i \|\widehat{\mathbf{y}}_i - \mathbf{T}\,\widehat{\mathbf{n}}_i\|^2, \tag{2.25}$$

where $\widehat{\mathbf{y}}_i$ are vector observations and $\widehat{\mathbf{n}}_i$ are their representation in the reference frame. This minimization problem can be reformulated for the quaternion, substituting the rotation matrix with $\mathbf{T}(\overline{\mathbf{q}})$ given in Eq. (2.5) and substituting the orthogonality requirement with a unitary norm constrain on $\overline{\mathbf{q}}$. The original solution to this problem is due to Davenport and is given by Keat [12]. Wahba performance index in Eq. (2.25) can be rewritten as

$$\mathcal{J}(\overline{\mathbf{q}}) = \lambda_0 - \mathcal{J}^{\star}(\overline{\mathbf{q}}), \tag{2.26}$$

where

$$\lambda_0 = \frac{1}{2} \sum_{i=1}^{n} w_i \left(\|\widehat{\mathbf{y}}_i\|^2 + \|\widehat{\mathbf{n}}_i\|^2 \right) \tag{2.27}$$

is independent from the quaternion. The minimization of Wahba performance index in Eq. (2.25) is equivalent to maximization of

$$\mathcal{J}^\star(\overline{\mathbf{q}}) = \sum_{i=1}^{n} w_i \mathbf{y}_i^{\mathrm{T}} \mathbf{T}(\overline{\mathbf{q}}) \mathbf{n}_i, \tag{2.28}$$

$$\text{subject to } \|\overline{\mathbf{q}}\|_2 = 1.$$

Scaling the performance index will not affect the solution, therefore often the weights are normalized, i.e. $\sum_{i=1}^{n} w_i = 1$. Vectors $\widehat{\mathbf{y}}_i$ and $\widehat{\mathbf{n}}_i$ are often of unitary norm, under those circumstances $\lambda_0 = 1$.

Defining the 3×3 matrix $\mathbf{B}$ as

$$\mathbf{B} \triangleq \sum_{i=1}^{n} w_i \widehat{\mathbf{y}}_i \widehat{\mathbf{n}}_i^{\mathrm{T}}, \tag{2.29}$$

and using matrix trace properties, it follows that Eq. (2.28) can be written as

$$\mathcal{J}^\star(\overline{\mathbf{q}}) = \operatorname{trace}\left[\mathbf{T}(\overline{\mathbf{q}}) \mathbf{B}^{\mathrm{T}}\right]. \tag{2.30}$$

Substituting Eq. (2.5) in the performance index of Eq. (2.30), and using $\mathbf{B}$ from Eq. (2.29) yields

$$\mathcal{J}^\star = \sigma(q^2 - \mathbf{q}^{\mathrm{T}}\mathbf{q}) + 2\mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{q} - 2q \operatorname{trace}\left[[\mathbf{q}\times]\mathbf{B}^{\mathrm{T}}\right],$$

where

$$\sigma \triangleq \operatorname{trace}(\mathbf{B}).$$

This problem constitutes a quadratic program, i.e. the performance index can be rewritten as

$$\mathcal{J}^{\star}(\overline{\mathbf{q}}) = \overline{\mathbf{q}}^{\mathrm{T}}\mathbf{K}\overline{\mathbf{q}}, \tag{2.31}$$

where the 4×4 matrix $\mathbf{K}$ is now obtained. Define the symmetric matrix $\mathbf{S}$ as

$$\mathbf{S} \triangleq \mathbf{B} + \mathbf{B}^{\mathrm{T}}.$$

Then it follows that

$$2\mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{q} = \mathbf{q}^{\mathrm{T}}\mathbf{S}\,\mathbf{q}.$$

Notice that

$$-2\,\mathrm{trace}\left[[\mathbf{q}\times]\mathbf{B}^{\mathrm{T}}\right] = -2\,\mathrm{trace}\left[\sum_{i=1}^{n} w_i[\mathbf{q}\times]\widehat{\mathbf{n}}_i\widehat{\mathbf{y}}_i^{\mathrm{T}}\right] = 2\sum_{i=1}^{n} w_i\left(\widehat{\mathbf{y}}_i \times \widehat{\mathbf{n}}_i\right)^{\mathrm{T}}\mathbf{q}.$$

Therefore, matrix $\mathbf{K}$ in Eq. (2.31) is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \sigma\mathbf{I}_{3\times 3} & \mathbf{z} \\ \mathbf{z}^{\mathrm{T}} & \sigma \end{bmatrix}, \tag{2.32}$$

where

$$\mathbf{z} \triangleq \sum_{i=1}^{n} w_i\left(\widehat{\mathbf{y}}_i \times \widehat{\mathbf{n}}_i\right).$$

Adjoining the constraint $||\overline{\mathbf{q}}||_2 = 1$ to the performance index with a Lagrange multiplier, denoted by λ, the first-order optimal condition is given by the eigenvalue problem

$$\mathbf{K}\overline{\mathbf{q}} = \lambda\overline{\mathbf{q}}. \tag{2.33}$$

Also using Eq. (2.31) and Eq. (2.33), the performance index can be shown to be

$$\mathcal{J}^\star = \lambda.$$

Since the performance index is to be maximized, the optimal Lagrange multiplier is given by the maximum eigenvalue of $\mathbf{K}$ given in Eq. (2.32), and the optimal quaternion is given by the corresponding unit eigenvector. There is no need to calculate the eigenvector. The vector of Rodrigues parameters is given by

$$\boldsymbol{\varrho} = \mathbf{q}/q.$$

The first three rows of Eq. (2.33) can be expanded to be

$$\left(\mathbf{S} - \sigma \mathbf{I}_{3\times 3}\right)\mathbf{q} + \mathbf{z}q = \lambda \mathbf{q},$$

from which the estimated Gibbs vector is found to be

$$\widehat{\boldsymbol{\varrho}} = [(\sigma + \lambda)\mathbf{I}_{3\times 3} - \mathbf{S}]^{-1}\mathbf{z}. \tag{2.34}$$

The optimal quaternion is given by

$$\widehat{\widehat{\mathbf{q}}} = \frac{1}{\sqrt{1 + \widehat{\boldsymbol{\varrho}}^{\mathrm{T}}\widehat{\boldsymbol{\varrho}}}}\begin{bmatrix}\widehat{\boldsymbol{\varrho}} \\ 1\end{bmatrix}. \tag{2.35}$$

Shuster and Oh [62] show how to handle Eq. (2.34) when matrix $(\sigma + \lambda)\mathbf{I}_{3\times 3} - \mathbf{S}$ is singular. The same paper shows a numerically efficient algorithm to compute the eigenvalue referred to as QUEST. Covariance analysis is also performed in [62] under the assumption of a simplified measurement model, known as the QUEST measurement model. The i^{th} measurement is modeled as

$$\widehat{\mathbf{y}}_i = \mathbf{T}(\overline{\mathbf{q}})\mathbf{n}_i + \widetilde{\mathbf{y}}_i = \mathbf{T}(\overline{\mathbf{q}})\left(\widehat{\mathbf{n}}_i - \widetilde{\mathbf{n}}_i\right) + \widetilde{\mathbf{y}}_i,$$

where $\mathbf{n}_i$ are the true reference vectors while $\widetilde{\mathbf{n}}_i$ and $\widetilde{\mathbf{y}}_i$ are errors. In [62], $\mathbf{y}_i$ and $\mathbf{n}_i$ are assumed to be unit vectors, and the measurement error is given by a rotation $\delta\boldsymbol{\theta}$

$$\widehat{\mathbf{n}}_i = \mathbf{n}_i + \widetilde{\mathbf{n}}_i = \mathbf{T}(\delta\boldsymbol{\theta}_i)\mathbf{n}_i.$$

Using Eq. (2.2) and assuming small angles

$$\widehat{\mathbf{n}}_i \simeq \mathbf{n}_i - [\delta\boldsymbol{\theta}_i\times]\mathbf{n}_i,$$

therefore

$$\mathrm{E}\left\{\widetilde{\mathbf{n}}_i\,\widetilde{\mathbf{n}}_i^{\mathrm{T}}\right\} = [\mathbf{n}_i\times]\,\mathrm{E}\left\{\delta\boldsymbol{\theta}_i\,\delta\boldsymbol{\theta}_i^{\mathrm{T}}\right\}[\mathbf{n}_i\times]^{\mathrm{T}}. \tag{2.36}$$

The QUEST measurement model assumes

$$\mathrm{E}\left\{\delta\boldsymbol{\theta}_i\,\delta\boldsymbol{\theta}_i^{\mathrm{T}}\right\} = \sigma_{n,i}^2\mathbf{I}_{3\times 3},$$

therefore Eq. (2.36) becomes

$$\mathrm{E}\left\{\widetilde{\mathbf{n}}_i\,\widetilde{\mathbf{n}}_i^{\mathrm{T}}\right\} = \sigma_{n,i}^2\left(\mathbf{I}_{3\times 3} - \mathbf{n}_i\,\mathbf{n}_i^{\mathrm{T}}\right).$$

Similarly

$$\mathrm{E}\left\{\widetilde{\mathbf{y}}_i\,\widetilde{\mathbf{y}}_i^{\mathrm{T}}\right\} = \sigma_{y,i}^2\left(\mathbf{I}_{3\times 3} - \mathbf{y}_i\,\mathbf{y}_i^{\mathrm{T}}\right),$$

where $\mathbf{y}_i$ are the true values of the measurements $\mathbf{y}_i = \mathbf{T}(\overline{\mathbf{q}})\,\mathbf{n}_i$. Since $\mathbf{n}_i$ and $\mathbf{y}_i$ are unknown, they have to be replaced by $\widehat{\mathbf{n}}_i$ and $\widehat{\mathbf{y}}_i$ when calculating the two covariances.

Notice that Matrix **B** fully defines the problem, since

$$[\mathbf{z}\times] = \mathbf{B}^{\mathrm{T}} - \mathbf{B}.$$

2.3.1 QUEST Covariance Analysis

The matrix associated with the true observations is defined as

$$\mathbf{B}_{true} \triangleq \sum_{i=1}^{n} w_i \mathbf{y}_i \mathbf{n}_i^{\mathrm{T}},$$

and the matrix associated with the measurement error

$$\delta\mathbf{B} = \sum_{i=1}^{n} w_i \widetilde{\mathbf{y}}_i \mathbf{n}_i^{\mathrm{T}} + \mathbf{T}(\overline{\mathbf{q}}) \sum_{i=1}^{n} w_i \mathbf{y}_i \widetilde{\mathbf{n}}_i^{\mathrm{T}}.$$

Therefore to first-order in the errors

$$\mathbf{B} = \mathbf{B}_{true} + \delta\mathbf{B}.$$

Similar quantities can be defined for $\mathbf{z}$, σ, and $\mathbf{S}$

$$\mathbf{z} = \mathbf{z}_{true} + \delta\mathbf{z}, \qquad \mathbf{S} = \mathbf{S}_{true} + \delta\mathbf{S}, \qquad \sigma = \sigma_{true} + \delta\sigma,$$

obtaining

$$\begin{aligned}
\mathbf{z}_{true} &= \sum_{i=1}^{n} w_i (\mathbf{y}_i \times \mathbf{n}_i) & \delta\mathbf{z} &= \sum_{i=1}^{n} w_i (\mathbf{y}_i \times \widetilde{\mathbf{n}}_i + \widetilde{\mathbf{y}}_i \times \mathbf{n}_i) \\
\mathbf{S}_{true} &= \mathbf{B}_{true} + \mathbf{B}_{true}^{\mathrm{T}} & \delta\mathbf{S} &= \delta\mathbf{B} + \delta\mathbf{B}^{\mathrm{T}} \\
\sigma_{true} &= \operatorname{trace} \mathbf{B}_{true} & \delta\sigma &= \operatorname{trace} \delta\mathbf{B}.
\end{aligned}$$

Provided that at least two independent vector measurements are available, the estimate obtained from $\mathbf{B}_{true}$ using Davenport-q algorithm is the true quaternion. Define

$$\mathbf{M} \triangleq (\sigma + \lambda)\mathbf{I}_{3\times 3} - \mathbf{S},$$

the true Gibbs vector is

$$\boldsymbol{\varrho} = \mathbf{M}_{true}^{-1}\mathbf{z}_{true},$$

where

$$\mathbf{M}_{true} = (\sigma_{true} + \lambda_{true})\mathbf{I}_{3\times 3} - \sum_{i=1}^{n} w_i \mathbf{y}_i \mathbf{n}_i^{\mathrm{T}} - \sum_{i=1}^{n} w_i \mathbf{n}_i \mathbf{y}_i^{\mathrm{T}}. \tag{2.37}$$

The estimated Gibbs vector is

$$\begin{aligned}\widehat{\boldsymbol{\varrho}} &= \left(\mathbf{M}_{true} + \delta\mathbf{M}\right)^{-1}\left(\mathbf{z}_{true} + \delta\mathbf{z}\right) \simeq \left(\mathbf{M}_{true}^{-1} - \mathbf{M}_{true}^{-1}\delta\mathbf{M}\,\mathbf{M}_{true}^{-1}\right)\left(\mathbf{z}_{true} + \delta\mathbf{z}\right)\\ &\simeq \boldsymbol{\varrho} + \mathbf{M}_{true}^{-1}\delta\mathbf{z} - \mathbf{M}_{true}^{-1}\delta\mathbf{M}\,\boldsymbol{\varrho} = \boldsymbol{\varrho} - \widetilde{\boldsymbol{\varrho}},\end{aligned}$$

where a first-order approximation was used. Defining a rotational estimation such that $\mathbf{T}(\overline{\mathbf{q}}) = \mathbf{T}(\delta\overline{\mathbf{q}})\mathbf{T}(\widehat{\overline{\mathbf{q}}})$, and using Eq. (2.4)

$$\delta\mathbf{q} \simeq \delta\boldsymbol{\varrho} = \frac{\mathbf{I}_{3\times 3} + [\boldsymbol{\varrho}\times]}{1 + \boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\varrho} - \boldsymbol{\varrho}^{\mathrm{T}}\widetilde{\boldsymbol{\varrho}}}\widetilde{\boldsymbol{\varrho}}, \tag{2.38}$$

using Mac-Laurin series

$$(1 + \boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\varrho} - \boldsymbol{\varrho}^{\mathrm{T}}\widetilde{\boldsymbol{\varrho}})^{-1} \simeq (1 + \boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\varrho})^{-1} + (1 + \boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\varrho})^{-2}\boldsymbol{\varrho}^{\mathrm{T}}\widetilde{\boldsymbol{\varrho}},$$

substituting in Eq. (2.38), the following first order approximation results

$$\delta\mathbf{q} \simeq \frac{\mathbf{I}_{3\times 3} + [\boldsymbol{\varrho}\times]}{1 + \boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\varrho}}\widetilde{\boldsymbol{\varrho}} = q\left(q\mathbf{I}_{3\times 3} + [\mathbf{q}\times]\right)\widetilde{\boldsymbol{\varrho}},$$

finally

$$\delta\mathbf{q} = \left(q\mathbf{I}_{3\times3} + [\mathbf{q}\times]\right)\,\mathbf{M}_{true}^{-1}\left(\delta\mathbf{M}\,\mathbf{q} - q\delta\mathbf{z}\right). \tag{2.39}$$

Shuster and Oh notice that the covariance should be approximately independent from the the true state, therefore Eq. (2.39) is evaluated at a convenient true state, the identity quaternion, resulting in

$$\delta\mathbf{q} = -\mathbf{M}_{true}^{-1}\delta\mathbf{z}.$$

Assuming $\widetilde{\mathbf{n}}_i$ and $\widetilde{\mathbf{y}}_i$ are uncorrelated from each other, the covariance is

$$\mathbf{P}_{\theta\theta} = 4\mathbf{M}_{true}^{-1}\sum_{i=1}^{n} w_i^2\Big\{[\mathbf{n}_i\times]\mathbf{R}_{y,i}[\mathbf{n}_i\times]^{\mathrm{T}} + [\mathbf{y}_i\times]\mathbf{R}_{n,i}[\mathbf{y}_i\times]^{\mathrm{T}}\Big\}\mathbf{M}_{true}^{-\mathrm{T}},$$

where

$$\mathrm{E}\left\{\widetilde{\mathbf{y}}_i\widetilde{\mathbf{y}}_j^{\mathrm{T}}\right\} = \mathbf{R}_{y,i}\,\delta_{ij}, \quad \mathrm{E}\left\{\widetilde{\mathbf{n}}_i\widetilde{\mathbf{n}}_j^{\mathrm{T}}\right\} = \mathbf{R}_{n,i}\,\delta_{ij}, \quad i,j = 1..n.$$

Since the true quaternion was chosen as the identity quaternion

$$\mathbf{n}_i = \mathbf{y}_i,$$

which substituted into Eq. (2.37) results in

$$\mathbf{M}_{true} = 2\sum_{i=1}^{n} w_i\|\mathbf{n}_i\|^2\,\mathbf{I}_{3\times3} - 2\sum_{i=1}^{n} w_i\mathbf{n}_i\mathbf{n}_i^{\mathrm{T}},$$

because $\sigma_{true} = \sum_{i=1}^{n} w_i\|\mathbf{n}_i\|^2$ and

$$\lambda_{true} = \mathcal{J}^\star(\overline{\mathbf{q}}) = \sigma_{true}.$$

Using vector product matrix properties of Eq. (2.3) it follows that

$$\mathbf{M}_{true} = -2\sum_{i=1}^{n} w_i[\mathbf{n}_i\times]^2, \tag{2.40}$$

using the QUEST measurement model, and choosing the weights such that

$$w_i = \frac{1}{\sigma_n^2 + \sigma_y^2}$$

the covariance becomes

$$\mathbf{P}_{\theta\theta}^{QUEST} = 2\mathbf{M}_{true}^{-1}.$$

In the zero attitude case, $\mathbf{M}_{true}$ is given by Eq. (2.40), or the original definition of Eq. (2.37), or equivalently

$$\mathbf{M}_{true} = -2\sum_{i=1}^{n} w_i[\mathbf{y}_i\times]^2. \tag{2.41}$$

These three different definitions are equivalent for zero attitude, but would result in deferent calculated covariances in the general case of $\overline{\mathbf{q}} \neq \bar{\mathbf{i}}_q$. In their original derivation [62], Shuster and Oh employ Eq. (2.41) without justification. The choice of Eq. (2.41) leads to the QUEST covariance formulation

$$\mathbf{P}_{\theta\theta}^{QUEST} = \left(\sum_{i=1}^{n} w_i[\mathbf{y}_i\times]^2\right)^{-1} = \left\{ \text{trace}\left[\mathbf{T}(\overline{\mathbf{q}})\mathbf{B}_{true}^{\mathrm{T}}\right]\mathbf{I}_{3\times3} - \mathbf{T}(\overline{\mathbf{q}})\mathbf{B}_{true}^{\mathrm{T}}\right\}^{-1}. \tag{2.42}$$

Since $\mathbf{T}(\overline{\mathbf{q}})$ and $\mathbf{B}_{true}$ are unknown, in applying this formula they need to be substituted by $\mathbf{T}(\widehat{\overline{\mathbf{q}}})$ and $\mathbf{B}$. Eight years after the original QUEST covariance analysis, this formulation was proven to be equivalent to the inverse of the Fisher information matrix, which is asymptotically equal to the covariance [13]. Notice however that the QUEST formulation of the covariance is attitude dependent, which is a direct contradiction of the starting assumption. Choosing to represent $\mathbf{M}_{true}$ with

Eq. (2.40) would satisfy the assumption that the covariance is independent from the true quaternion. It turns out that both formulations are correct, they express the covariance in different frames.

The true quaternion $\overline{\mathbf{q}}$ expresses the rotation from a reference frame i to the body frame b. Assuming the covariance is independent from the true attitude, i.e. is independent from the body frame, the body frame can be rotated to coincide with the reference frame for covariance calculation purposes. If that was the case reference vectors $\mathbf{n}_i$ would stay the same, but new measurements $\widehat{\mathbf{y}}_i^*$ would occur

$$\widehat{\mathbf{y}}_i^* = \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\widehat{\mathbf{y}}_i = \mathbf{n}_i + \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\widetilde{\mathbf{y}}_i.$$

Repeating the previous analysis replacing the following quantities

$$\mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\mathbf{y}_i \rightarrow \mathbf{y}_i, \qquad \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\widetilde{\mathbf{y}}_i \rightarrow \widetilde{\mathbf{y}}_i$$

it follows that

$$\delta\boldsymbol{\theta} = -2\mathbf{M}_{true}^{-1}\sum_{i=1}^{n} w_i(\mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\mathbf{y}_i \times \widetilde{\mathbf{n}}_i + \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\widetilde{\mathbf{y}}_i \times \mathbf{n}_i), \tag{2.43}$$

and the following covariance formulation is obtained

$$\mathbf{P}_{\theta\theta} = 4\mathbf{M}_{true}^{-1}\sum_{i=1}^{n} w_i^2\Big\{[\mathbf{n}_i\times]\left(\mathbf{R}_{n,i} + \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}}\mathbf{R}_{y,i}\mathbf{T}(\overline{\mathbf{q}})\right)[\mathbf{n}_i\times]^{\mathrm{T}}\Big\}\mathbf{M}_{true}^{-\mathrm{T}},$$

where $\mathbf{M}_{true}$ is given by Eq. (2.40). If the measurement covariance follows the QUEST measurement model, and if the weights are chosen such that

$$w_i = \frac{1}{\sigma_n^2 + \sigma_y^2},$$

a modification of the QUEST covariance is obtained

$$\mathbf{P}_{\theta\theta}^{MOD} = \left(\sum_{i=1}^{n} w_i [\mathbf{n}_i \times]^2 \right)^{-1} = \left\{ \text{trace} \left[\mathbf{B}_{true}^{\mathrm{T}} \mathbf{T}(\overline{\mathbf{q}}) \right] \mathbf{I}_{3\times 3} - \mathbf{B}_{true}^{\mathrm{T}} \mathbf{T}(\overline{\mathbf{q}}) \right\}^{-1}. \quad (2.44)$$

Like in the previous case, $\mathbf{T}(\overline{\mathbf{q}})$ and $\mathbf{B}_{true}$ are unknown therefore they need to be substituted by $\mathbf{T}(\widehat{\overline{\mathbf{q}}})$ and $\mathbf{B}$. The QUEST formulation returns the covariance in the body frame, while the modified formulation returns the covariance in the inertial frame. Removing the QUEST measurement model assumption from the QUEST covariance formulation, results in the following generalized covariance

$$\mathbf{P}_{\theta\theta}^{GEN} = 4\mathbf{M}_{true}^{-1} \sum_{i=1}^{n} w_i^2 \Big\{ [\mathbf{y}_i \times] \left(\mathbf{T}(\overline{\mathbf{q}}) \mathbf{R}_{n,i} \mathbf{T}(\overline{\mathbf{q}})^{\mathrm{T}} + \mathbf{R}_{y,i} \right) [\mathbf{y}_i \times]^{\mathrm{T}} \Big\} \mathbf{M}_{true}^{-\mathrm{T}}, \quad (2.45)$$

where $\mathbf{M}_{true}$ is given by Eq. (2.41).

Consider the following example, the true quaternion is given by

$$\overline{\mathbf{q}} = [0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5]^{\mathrm{T}},$$

two observations are available

$$\mathbf{n}_1 = [1 \;\; 0 \;\; 0]^{\mathrm{T}}, \qquad \mathbf{n}_2 = [0 \;\; 1 \;\; 0]^{\mathrm{T}},$$

from which

$$\mathbf{T}(\overline{\mathbf{q}}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{y}_1 = [0 \;\; 0 \;\; 1]^{\mathrm{T}}, \quad \mathbf{y}_2 = [1 \;\; 0 \;\; 0]^{\mathrm{T}}.$$

The measurement model is not the QUEST measurement model because the norm of the measurement is allowed to vary

$$\widehat{\mathbf{y}}_i = \mathbf{y}_i + \boldsymbol{\eta}_i, \quad \mathrm{E}\left\{\boldsymbol{\eta}_i \boldsymbol{\eta}_j\right\} = \sigma_y^2 \mathbf{I}_{3\times3}\delta_{ij}, \quad i,j = 1,2.$$

The norm of the estimates of the reference vectors is also allowed to vary

$$\widehat{\mathbf{n}}_i = \mathbf{n}_i + \boldsymbol{\nu}_i, \quad \mathrm{E}\left\{\boldsymbol{\nu}_i \boldsymbol{\nu}_j\right\} = \sigma_n^2 \mathbf{I}_{3\times3}\delta_{ij}, \quad i,j = 1,2.$$

The same covariance formulation as the one using the QUEST measurement model is still valid when

$$w_i = \frac{1}{\sigma_n^2 + \sigma_y^2} \qquad i = 1,2.$$

The standard deviations are chosen as

$$\sigma_n = 0.05 \qquad \sigma_y = 0.1,$$

leading to

$$\mathbf{B}_{true} = \begin{bmatrix} 0 & 80 & 0 \\ 0 & 0 & 0 \\ 80 & 0 & 0 \end{bmatrix}.$$

Using the previously derived equation

$$\mathbf{P}_{\theta\theta}^{MOD} = \left\{ \operatorname{trace}\left[\mathbf{B}_{true}^{\mathrm{T}} \mathbf{T}(\overline{\mathbf{q}})\right] \mathbf{I}_{3\times3} - \mathbf{B}_{true}^{\mathrm{T}} \mathbf{T}(\overline{\mathbf{q}}) \right\}^{-1}$$

the following covariance is obtained

$$\mathbf{P}_{\theta\theta}^{MOD} = \begin{bmatrix} 0.0125 & 0 & 0 \\ 0 & 0.0125 & 0 \\ 0 & 0 & 0.0063 \end{bmatrix}$$

Using five thousand samples from a normal distribution, the statistical covariance defining the error in the inertial frame (as the vector component of $\widehat{\overline{\mathbf{q}}}^{-1} \otimes \overline{\mathbf{q}}$) is

$$\mathbf{P}_{\theta\theta}^{STAT} = \begin{bmatrix} 0.0131 & -0.0001 & 0.0001 \\ -0.0001 & 0.0122 & -0.0000 \\ 0.0001 & -0.0000 & 0.0064 \end{bmatrix}.$$

The QUEST covariance formulation provides

$$\mathbf{P}_{\theta\theta}^{QUEST} = \begin{bmatrix} 0.0125 & 0 & 0 \\ 0 & 0.0063 & 0 \\ 0 & 0 & 0.0125 \end{bmatrix},$$

and the statistical covariance obtained defining the error in the body frame (as the vector component of $\overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1}$) is

$$\mathbf{P}_{\theta\theta}^{STAT} = \begin{bmatrix} 0.0122 & -0.0000 & -0.0001 \\ -0.0000 & 0.0064 & 0.0001 \\ -0.0001 & 0.0001 & 0.0131 \end{bmatrix}.$$

Often times the reference vectors $\widehat{\mathbf{n}}_i$ are functions of the spacecraft position, for example in the case of the magnetometer. In those cases the position estimate needs to be provided by another system, and is useful to derive the cross covariance.

$$\widehat{\mathbf{n}}_i = \widehat{\mathbf{n}}_i(\mathbf{r}) \simeq \mathbf{n}_i + \widetilde{\mathbf{n}}_i + \mathbf{A}_i \mathbf{e}_r, \qquad \mathbf{e}_r \triangleq \mathbf{r} - \widehat{\mathbf{r}}, \qquad \mathbf{A}_i \triangleq \left. \frac{d\widehat{\mathbf{n}}_i}{d\mathbf{r}} \right|_{\mathbf{r}=\hat{\mathbf{r}}}$$

from Eq. (2.43) follows immediately that

$$\mathbf{P}_{\theta r} = 2\mathbf{M}_{true}^{-1} \sum_{i=1}^{n} w_i [\mathbf{n}_i \times] \mathbf{A}_i \mathbf{P}_{rr}, \tag{2.46}$$

assuming $\widetilde{\mathbf{n}}_i$, $\widetilde{\mathbf{y}}_i$, and $\mathbf{e}_r$ are all uncorrelated to each other, and $\mathbf{P}_{rr} = \mathrm{E}\left\{\mathbf{e}_r\mathbf{e}_r^{\mathrm{T}}\right\}$ is provided externally.

2.3.2 Summary

A very common spacecraft attitude determination algorithm was introduced. The original QUEST covariance formulation was re-derived in a different manner and modified to account for (i) a more realistic measurement model that does not necessitate the QUEST measurement model approximation, (ii) a different definition of the attitude estimation error.

2.4 Single Layer Gating Network

The MMAE scheme employed in this work is a modified version of the gating network of Chaer et al. [31, 32]. Figure 2.3[‡] shows the structure of the gating network. The gating network is basically a single layer of cells, each cell receives the same vector of inputs (in navigation applications the inputs are sensor measurements) and computes a weighted sum of the input, which is then passed to the hypothesis testing algorithm which compares the weighted sum to a threshold. The measurement can either pass or fail, in [63] it is shown how this procedure can be interpreted as dividing the hyper-space with a hyper-plane, with one half of the space containing the inputs matching the required pattern and the other half containing the inputs that fail the test. The gating network proposed by Chaer et al. substitutes the threshold with the *softmax* function, which has the advantage of being differentiable. A navigation filter is not desired to be trained beforehand, therefore the inputs weights are continuously updated on-line according with statistical information derived by the filters. The weighted sum of the measurement at time t_k can be written as an inner product $\mathbf{y}_k^{\mathrm{T}}\mathbf{u}_i$, where $\mathbf{u}_i$ is the vector containing the input weights of the i^{th} filter.

[‡] Figure courtesy of R. H. Bishop

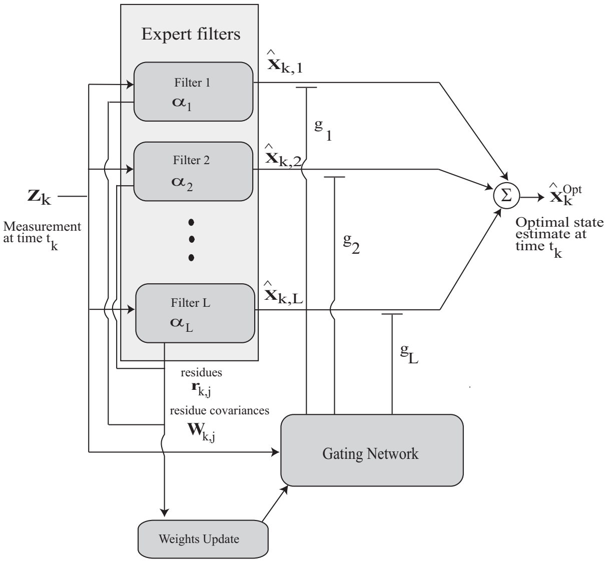


Figure 2.3: The hierarchical gating network architecture.

Considering $\mathbf{u}_i$ as the normal to a hyper-plane, then $\mathbf{y}_k^{\mathrm{T}}\mathbf{u}_i$ will be positive if $\mathbf{y}_k$ lays on the half hyper-space towards where $\mathbf{u}_i$ is pointing, and negative otherwise. The measurement is considered accepted if the inner product returns a positive number, and rejected otherwise. Ideally, after some measurements are processed, the hyper-planes are oriented in such a way that only the best performing filter validates measurements.

As said before the hypothesis algorithm uses a *softmax* function and assigns a weight w_i to each filter in the bank

$$w_i = \frac{e^{\mathbf{y}_k^{\mathrm{T}}\mathbf{u}_i}}{\Sigma_{i=1}^{L} e^{\mathbf{y}_k^{\mathrm{T}}\mathbf{u}_i}}, \tag{2.47}$$

this function is not only differentiable as stated before, but also provides weights that satisfy the characteristics to be interpreted as probabilities

$$0 \leq w_i \leq 1, \;\; \forall i = 1, 2, ..., L \qquad \text{and} \qquad \sum_{i=1}^{L} w_i = 1.$$

The bank of filters is implemented with different values of the unknown vector of parameters $\boldsymbol{\alpha}$. The gating network has to assign a weight w_i to each filter. The probability density of the entire bank is

$$f(\mathbf{y}_k) = \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) P(\boldsymbol{\alpha}_i) \tag{2.48}$$

where $P(\cdot)$ denotes probability and $\boldsymbol{\alpha}_i$ is the unknown parameters realization of the i^{th} filter. Interpreting the weight as a probability, Eq. (2.48) becomes

$$f(\mathbf{y}_k) = \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) w_i.$$

The goal is to maximize the probability of the bank. In order to maximize this probability density, it is easier to work with the natural logarithm of $f(\mathbf{y}_k)$, or

$$l \triangleq \ln f(\mathbf{y}_k) = \ln \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) e^{\mathbf{y}_k^{\mathrm{T}} \mathbf{u}_i} - \ln \sum_{i=1}^{L} e^{\mathbf{y}_k^{\mathrm{T}} \mathbf{u}_i}.$$

Taking the derivative of l with respect to $\mathbf{u}_i$ yields

$$\frac{\partial l}{\partial \mathbf{u}_i} = \left(P(\boldsymbol{\alpha}_i \mid \mathbf{y}_k) - w_i\right) \mathbf{y}_k, \tag{2.49}$$

where

$$P(\boldsymbol{\alpha}_i \mid \mathbf{y}_k) = \frac{f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) w_i}{f(\mathbf{y}_k)}.$$

Eq. (2.49) shows the direction of maximum growth of the function l. The update is accomplished via

$$\mathbf{u}_i \leftarrow \mathbf{u}_i + \lambda \frac{\partial l}{\partial \mathbf{u}_i}$$

where λ is a learning rate parameter. The gating network gains can now be computed with Eq. (2.47).

Chapter 3

Theoretical Foundations

This chapter presents the analytical tools to be used in the applications of the subsequent chapters. Section 3.1 introduces attitude in spacecraft navigation. One of the solutions proposed will be used in Chapter 6 where the star tracker implements an attitude filter to be fused with the central filter. Section 3.2 introduces a way to account for biases in the Kalman filter without explicitly estimating them. This formulation will be applied in the covariance analysis of the IMU dead-reckoning approach for Mars entry navigation in Chapter 4. Finally, Section 3.3 contains a modification to the gating network introduced in § 2.4, this modified adaptive filter will be used in Chapter 5 to account for atmospheric uncertainty during the accelerometer filtering of Mars entry navigation.

3.1 Introduction of Attitude Estimation in Spacecraft Navigation

It is still common on spacecraft missions that the attitude estimation and the spacecraft navigation are handled as independent subsystems. While it could be argued that the attitude does not depend on position and velocity, hence leading to an optimal estimate when independently obtained; the attitude estimation errors naturally enter and affect the translation navigation sub-system. In order to have an optimal estimate it is therefore to account for the effect attitude estimation and position/velocity estimation on the other, and correctly incorporate the correlations. The most straightforward way is to have a single navigation algorithm which estimates both rotational and translational states. Relevant examples of such a solution are given in Chapters 4 and 5. However such a solution is not always desirable either because of heritage reasons, or because some attitude determination algorithms cannot be expanded to estimate other states. The hardware also can affect the decision of having a single navigation system: a star tracker naturally comes with an attitude determination algorithm and provides an attitude estimate rather than raw measurements. In this section various approaches to the decentralized attitude navigation problem will be analyzed. It is possible to obtain an optimal decentralized structure as shown in §3.1.1, other approaches lead to a suboptimal navigation architecture (§3.1.3, §3.1.4). The architectures that are of interest here are those without a master filter. The problem solved is not of fusing the outputs of individual filters. The first architecture investigated here is a filter/sub-filter shown on Figure 3.1. Notice that is not a fusion problem like that depicted in Figure 3.2, which was solved by Carpenter and Bishop [64] for correlated measurements. The problem of an arbitrary number of sub-filters has also been addressed, see for example [65] and [66].

3.1.1 Uncorrelated Sub-filter

Assume the state vector is partitioned into two components as

$$\mathbf{x}^{\mathrm{T}} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} & \mathbf{x}_2^{\mathrm{T}} \end{bmatrix}.$$

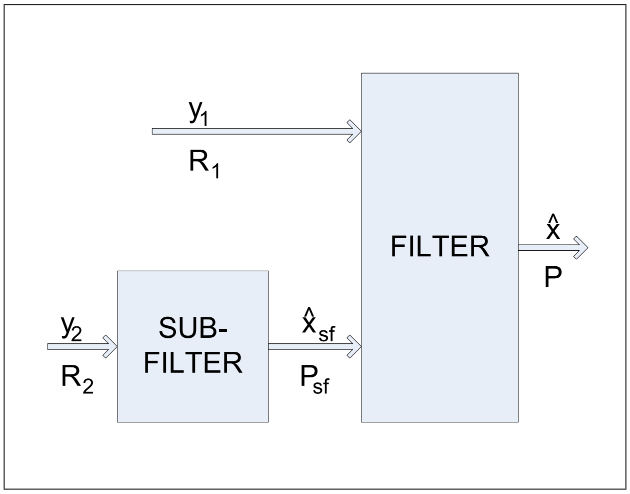


Figure 3.1: Main filter with raw measurements and a filtered state as inputs.

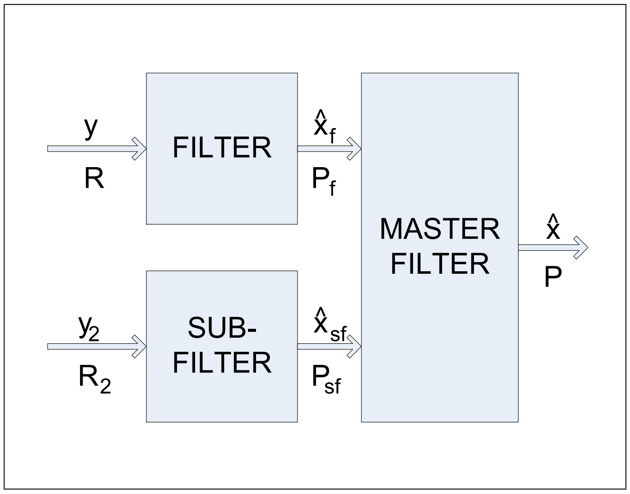


Figure 3.2: Classic filter fusion problem (not considered here).

Consider that the entire vector measurement can be also partitioned into two components. Vector $\mathbf{y}_2$ is independent of $\mathbf{x}_1$, and $\mathbf{y}_1$ is a function of the entire state $\mathbf{x}$.

With the measurement partitioned as

$$\mathbf{y}^{\mathrm{T}} = \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} & \mathbf{y}_2^{\mathrm{T}} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{H}_1\,\mathbf{x} + \boldsymbol{\nu}_1, \quad \mathbf{y}_2 = \mathbf{H}_2\,\mathbf{x}_2 + \boldsymbol{\nu}_2.$$

At each time t_k when the measurement is available, we have the measurement model

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \boldsymbol{\nu}_k$$

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{1,k} \\ \widetilde{\mathbf{H}}_{2,k} \end{bmatrix},$$

where

$$\widetilde{\mathbf{H}}_{2,k} = \begin{bmatrix} \mathbf{O} & \mathbf{H}_{2,k} \end{bmatrix}.$$

Vector $\boldsymbol{\nu}_k$ is zero mean white noise with

$$\mathrm{E}\left\{\boldsymbol{\nu}_k\,\boldsymbol{\nu}_j^{\mathrm{T}}\right\} = \mathbf{R}_k\,\delta_{kj} \quad \forall k, j.$$

Superscript $-$ denotes the *a priori* value, i.e. the value before the measurement is processed. Superscript $+$ denotes the *a posteriori* value, i.e. the value after the measurement is processed.

The optimal global information filter update is [67]

$$(\mathbf{P}^{+})^{-1} = (\mathbf{P}^{-})^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} \tag{3-1a}$$

$$\widehat{\mathbf{x}}^{+} = \mathbf{P}^{+}\left[(\mathbf{P}^{-})^{-1}\widehat{\mathbf{x}}^{-} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}\right], \tag{3-1b}$$

matrix $\mathbf{P}$ being the estimation error covariance. Assuming $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are uncorrelated from each other, i.e.

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_2 \end{bmatrix},$$

Eqs. (3-1a)-(3-1b) become

$$(\mathbf{P}^+)^{-1} = (\mathbf{P}^-)^{-1} + \mathbf{H}_1^{\mathrm{T}}\mathbf{R}_1^{-1}\mathbf{H}_1 + \widetilde{\mathbf{H}}_2^{\mathrm{T}}\mathbf{R}_2^{-1}\widetilde{\mathbf{H}}_2 \tag{3-2a}$$

$$\widehat{\mathbf{x}}^+ = \mathbf{P}^+ \left[(\mathbf{P}^-)^{-1}\widehat{\mathbf{x}}^- + \mathbf{H}_1^{\mathrm{T}}\mathbf{R}_1^{-1}\mathbf{y}_1 + \widetilde{\mathbf{H}}_2^{\mathrm{T}}\mathbf{R}_2^{-1}\mathbf{y}_2\right]. \tag{3-2b}$$

Let first consider the case in which the sub-filter provides estimates based only on the current measurement and not on previous ones, i.e. $\mathbf{P}_{sf}^- = \infty$. Then the sub-filter update is given by

$$\mathbf{P}_{sf}^{-1} = \mathbf{H}_2^{\mathrm{T}}\mathbf{R}_2^{-1}\mathbf{H}_2 \tag{3-3a}$$

$$\widehat{\mathbf{x}}_{sf} = \mathbf{P}_{sf}\mathbf{H}_2^{\mathrm{T}}\mathbf{R}_2^{-1}\mathbf{y}_2. \tag{3-3b}$$

Using Eqs. (3-3a)–(3-3b), the update of the central filter given by Eqs. (3-2a)–(3-2b), can be rewritten as

$$\begin{aligned} (\mathbf{P}^+)^{-1} &= (\mathbf{P}^-)^{-1} + \mathbf{H}_1^{\mathrm{T}}\mathbf{R}_1^{-1}\mathbf{H}_1 + \begin{bmatrix}\mathbf{O} & \mathbf{I}\end{bmatrix}^{\mathrm{T}} \mathbf{P}_{sf}^{-1} \begin{bmatrix}\mathbf{O} & \mathbf{I}\end{bmatrix} \\ \widehat{\mathbf{x}}^+ &= \mathbf{P}^+ \left\{(\mathbf{P}^-)^{-1}\widehat{\mathbf{x}}^- + \mathbf{H}_1^{\mathrm{T}}\mathbf{R}_1^{-1}\mathbf{y}_1 + \begin{bmatrix}\mathbf{O} & \mathbf{I}\end{bmatrix}^{\mathrm{T}} \mathbf{P}_{sf}^{-1}\,\widehat{\mathbf{x}}_{sf}\right\}. \end{aligned}$$

Therefore, treating the sub-filter as a sensor that measures $\mathbf{x}_2$ with covariance $\mathbf{P}_2$ will lead to an optimal linear estimate.

Now suppose that the sub-filter employs all previous measurements to generate the estimate. In this case, there will be correlation between the measurement error and the *a priori* estimation error. The linear update of the central filter is given by

$$\widehat{\mathbf{x}}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\widehat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{y}_k.$$

The associated estimation error is

$$\mathbf{e}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{e}_k^- - \mathbf{K}_k\boldsymbol{\nu}_k,$$

and the estimation error covariance is given by

$$\begin{aligned}\mathbf{P}_k^+ = &(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_k^-\,(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}} - (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathrm{E}\left\{\mathbf{e}_k^-\,\boldsymbol{\nu}_k^{\mathrm{T}}\right\}\mathbf{K}_k^{\mathrm{T}} + \\ &- \mathbf{K}_k\,\mathrm{E}\left\{\boldsymbol{\nu}_k\,(\mathbf{e}_k^-)^{\mathrm{T}}\right\}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}}.\end{aligned}$$

Defining

$$\mathbf{C}_k \triangleq \mathrm{E}\left\{\mathbf{e}_k^-\,\boldsymbol{\nu}_k^{\mathrm{T}}\right\},$$

the gain that minimizes the trace of the *a posteriori* covariance is

$$\mathbf{K}_k = \left(\mathbf{H}_k\mathbf{P}_k^- + \mathbf{C}_k^{\mathrm{T}}\right)\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \mathbf{H}_k\mathbf{C}_k + \mathbf{C}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\right)^{-1}.$$

The *a posteriori* estimation error is a linear combination of the initial estimation error, the measurement noise, and the process noise. Assuming the measurement noise is uncorrelated with the initial state estimate and the process noise, it follows that

$$\mathbf{C}_k = -\sum_{i=1}^{k-1}\mathbf{Z}_i\boldsymbol{\Phi}_i\mathbf{K}_i\,\mathrm{E}\left\{\boldsymbol{\nu}_i\boldsymbol{\nu}_k^{\mathrm{T}}\right\} \tag{3.4}$$

$$\mathbf{Z}_{k-j} = \begin{cases}\mathbf{I} & j = 1\\ \prod_{i=k-1}^{k-j+1}\boldsymbol{\Phi}_i(\mathbf{I} - \mathbf{K}_i\mathbf{H}_i) & j > 1.\end{cases}$$

The standard Kalman filter assumes $\mathrm{E}\left\{\boldsymbol{\nu}_i\boldsymbol{\nu}_k^{\mathrm{T}}\right\} = \mathbf{R}_i\delta_{ik}$, resulting in $\mathbf{C}_k = \mathbf{O}$. For this application the sub-filter is used as a sensor by the central filter, therefore the measurement noise $\boldsymbol{\nu}_k$ is given by the the estimation error of the sub-filter $\mathbf{e}_{sf,k}$. For any Kalman filter, the correlation between the *a posteriori* estimation errors at different times is given by

$$\mathrm{E}\left\{\mathbf{e}_k^+(\mathbf{e}_j^+)^{\mathrm{T}}\right\} = \left[\prod_{i=k}^{j+1} (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\,\boldsymbol{\Phi}_{i-1}\right]\mathbf{P}_j^+, \qquad j < k. \tag{3.5}$$

The measurement noise of the central filter is given by the estimation error of the sub-filter. The central filter computes $\mathbf{C}_k$ using Eq. (3.4), where $\mathrm{E}\left\{\boldsymbol{\nu}_i\boldsymbol{\nu}_k^{\mathrm{T}}\right\}$ is computed by the sub-filter using Eq. (3.5). Clearly, although possible, accounting for the time correlation is difficult, and this implementation with sub-filter dynamics is not very practical for on-board navigation.

3.1.2 Star-Tracker Implementation

While an exhaustive implementation of the above result will be given in Chapter 6, it is useful at this point to provide an illustrative example. The example is a sub-filter that reconstructs the quaternion from star measurements without dynamics. The quaternion is processed by the main filter as a measurement. The main assumption holds, since stars can be considered as infinitely far and therefore the star-tracker measurement can be considerate independent from position, velocity, angular velocity, and any state except the quaternion itself.

The optimality of the previous section was shown for the linear model. However, attitude estimation is inherently nonlinear. Because of the absence of *a priori* information, Eqs. (3-3a) and (3-3b) are the linear least-squares solution. For the nonlinear star-tracker algorithm two solutions are possible. The first is to linearize the problem. The preferred solution is to substitute the linear least-squares formulation with the nonlinear least-squares given by the Davenport-q method. The multiplicative approach is chosen to represent the estimation error, and the covariance is computed using Eq. (2.45). The algorithm to process the star-tracker measurement has only the quaternion and gyro bias as state elements. The state vector could easily be augmented with position, velocity and other states, and other measurements can be also included.

The measurement $\overline{\mathbf{y}} = \widehat{\overline{\mathbf{q}}}_{st}$ is a quaternion. The estimate of the quaternion is computed internally by the star-tracker. The multiplicative approach estimates the deviation between a nominal attitude and the estimated attitude. This deviation is usually expressed as a small rotation vector $\delta\boldsymbol{\alpha}$ but could also be represented with a small Gibbs vector $\delta\boldsymbol{\varrho}$. In the multiplicative extended algorithm, the nominal trajectory is refreshed after every update. Being that the state is a small rotation vector, and the measurement noise covariance, i.e. the star tracker covariance, is also computed for a small rotation vector, it is natural to rewrite the quaternion measurement as rotation vector between the nominal and measured attitude [68]. In doing so, the classical formulation of the Kalman filter is recovered. Suppose a measurement of the rotation between the nominal and measured attitude was available at t_k. Then the state update would be

$$\widehat{\boldsymbol{\alpha}}_k^+ = \widehat{\boldsymbol{\alpha}}_k^- + \mathbf{K}_k \, \delta\boldsymbol{\epsilon}_k,$$

where $\widehat{\boldsymbol{\alpha}}_k^- = \mathbf{0}$ because the *a priori* nominal quaternion coincide with the estimated *a priori* quaternion, and $\delta\boldsymbol{\epsilon}_k$ is the residual, that coincide with the measurement because $\widehat{\boldsymbol{\alpha}}_k^- = \mathbf{0}$. But $\delta\boldsymbol{\epsilon}_k$ is twice the vector part of $\delta\overline{\mathbf{y}}_k$, and $\delta\overline{\mathbf{y}}_k$ is given by

$$\delta\overline{\mathbf{y}} \triangleq \overline{\mathbf{y}} \otimes \widehat{\overline{\mathbf{q}}}^{\,-1}.$$

Let $\overline{\mathbf{q}}_k$ be the true quaternion

$$\overline{\mathbf{q}}_k^{\mathrm{T}} = [\mathbf{q}_k^{\mathrm{T}} \quad q_k].$$

The star tracker measurement is corrupted by a small rotational error $\delta\boldsymbol{\theta}$ whose covariance is given in Eq. (2.45). Since the error is small, the measured quaternion can be modelled as

$$\overline{\mathbf{y}}_k = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_k \\ 1 \end{bmatrix} \otimes \overline{\mathbf{q}}_k,$$

where

$$\mathbf{P}_{st,k} = \mathrm{E}\left\{\delta\boldsymbol{\theta}_k \, \delta\boldsymbol{\theta}_k^{\mathrm{T}}\right\}.$$

Using the above arguments and including the gyro bias in the state vector, the *a posteriori* estimate is given by

$$\begin{bmatrix} \delta\widehat{\boldsymbol{\alpha}}_k^{+} \\ \widehat{\mathbf{b}}_k^{+} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \widehat{\mathbf{b}}_k^{-} \end{bmatrix} + \mathbf{K}_k \, \delta\boldsymbol{\epsilon}_k.$$

The measurement residual is

$$\delta\overline{\mathbf{y}}_k = \overline{\mathbf{y}}_k \otimes \left(\widehat{\overline{\mathbf{q}}}_k^{-}\right)^{-1} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_k \\ 1 \end{bmatrix} \otimes \overline{\mathbf{q}}_k \otimes \left(\widehat{\overline{\mathbf{q}}}_k^{-}\right)^{-1}. \tag{3.6}$$

The true *a priori* deviation is

$$\delta\overline{\mathbf{q}}_k^{-} = \overline{\mathbf{q}}_k \otimes \left(\widehat{\overline{\mathbf{q}}}_k^{-}\right)^{-1}.$$

Substituting into Eq. (3.6) yields

$$\delta\overline{\mathbf{y}}_k = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_k \\ 1 \end{bmatrix} \otimes \delta\overline{\mathbf{q}}_k^{-}.$$

Since the true deviation is approximated by

$$\delta\overline{\mathbf{q}}_k^- \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha}_k^- \\ 1 \end{bmatrix},$$

it follows that the multiplicative residual is

$$\delta\boldsymbol{\epsilon}_k = \delta\boldsymbol{\alpha}_k^- + 2\delta\boldsymbol{\theta}_k \times \delta\boldsymbol{\alpha}_k^- + \delta\boldsymbol{\theta}_k.$$

To first order, we have

$$\delta\boldsymbol{\epsilon}_k \simeq \delta\boldsymbol{\alpha}_k^- + \delta\boldsymbol{\theta}_k. \tag{3.7}$$

The true *a posteriori* deviation is computed to be

$$\delta\overline{\mathbf{q}}_k^+ = \overline{\mathbf{q}}_k \otimes \left(\widehat{\overline{\mathbf{q}}}_k^+\right)^{-1} = \delta\overline{\mathbf{q}}_k^- \otimes \widehat{\overline{\mathbf{q}}}_k^- \otimes \left(\delta\widehat{\overline{\mathbf{q}}}_k^+ \otimes \widehat{\overline{\mathbf{q}}}_k^-\right)^{-1} = \delta\overline{\mathbf{q}}_k^- \otimes \left(\delta\widehat{\overline{\mathbf{q}}}_k^+\right)^{-1}.$$

The vector component of $\delta\overline{\mathbf{q}}_k^+$ is

$$\delta\boldsymbol{\alpha}_k^+ = \delta\boldsymbol{\alpha}_k^- - \frac{1}{2}\delta\boldsymbol{\alpha}_k^- \times (\mathbf{K}_{\alpha,k}\delta\boldsymbol{\epsilon}_k) - \mathbf{K}_{\alpha,k}\delta\boldsymbol{\epsilon}_k.$$

The estimation errors are

$$\mathbf{e}_{\alpha,k} \triangleq \delta\boldsymbol{\alpha}_k - \delta\widehat{\boldsymbol{\alpha}}_k, \qquad \mathbf{e}_{b,k} \triangleq \mathbf{b}_k - \widehat{\mathbf{b}}_k, \qquad \mathbf{e}_k = \begin{bmatrix} \mathbf{e}_{\alpha,k}^{\mathrm{T}} & \mathbf{e}_{b,k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},.$$

Assuming small quantities, truncating to first order, and using Eq. (3.7), it follows that

$$\mathbf{e}_k^+ = (\mathbf{I}_{6\times6} - \mathbf{K}_k\,\mathbf{H}_k)\,\mathbf{e}_k^- - \mathbf{K}_k\,\delta\boldsymbol{\theta}_k,$$

where

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{O}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}.$$

The *a posteriori* error covariance is

$$\mathbf{P}_k = \mathrm{E}\left\{\mathbf{e}_k\, \mathbf{e}_k^{\mathrm{T}}\right\}.$$

Hence, we have

$$\mathbf{P}_k^{+} = \left(\mathbf{I}_{6\times 6} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^{-}\left(\mathbf{I}_{6\times 6} - \mathbf{K}_k\mathbf{H}_k\right)^{\mathrm{T}} + \mathbf{K}_k\mathbf{P}_{st,k}\mathbf{K}_k^{\mathrm{T}},$$

and the optimal Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{\mathrm{T}} + \mathbf{P}_{st,k}\right)^{-1}.$$

The true quaternion propagation is modeled via

$$\frac{d}{dt}\overline{\mathbf{q}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega}\\ 0\end{bmatrix} \otimes \overline{\mathbf{q}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega}^{m} - \delta\boldsymbol{\omega} - \mathbf{b}\\ 0\end{bmatrix} \otimes \overline{\mathbf{q}},$$

where $\delta\boldsymbol{\omega}$ is the gyro noise, $\mathbf{b}$ is the bias, and $\boldsymbol{\omega}^{m}$ is the gyro measurement of body rate. The propagation of the estimate is given by

$$\frac{d}{dt}\widehat{\overline{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega}^{m} - \widehat{\mathbf{b}}\\ 0\end{bmatrix} \otimes \widehat{\overline{\mathbf{q}}}.$$

The quaternion estimation error is $\delta\overline{\mathbf{q}} = \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{\,-1}$, hence its evolution is

$$\frac{d}{dt}\delta\overline{\mathbf{q}} = \frac{d}{dt}\overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1} + \overline{\mathbf{q}} \otimes \frac{d}{dt}\left(\widehat{\overline{\mathbf{q}}}^{-1}\right).$$

The estimation error $\boldsymbol{\alpha}$ is twice the vector component of $\delta\overline{\mathbf{q}}$ and evolves as

$$\frac{d}{dt}\delta\boldsymbol{\alpha} \simeq -(\boldsymbol{\omega}^{m} - \widehat{\mathbf{b}}) \times \delta\boldsymbol{\alpha} - \mathbf{e}_b - \delta\boldsymbol{\omega},$$

valid to first order in the errors. The bias is modeled as

$$\dot{\mathbf{b}} = \boldsymbol{\nu}_b,$$

vector $\boldsymbol{\nu}_b$ is white noise. The propagation of the estimate of the bias is

$$\frac{d}{dt}\widehat{\mathbf{b}} = \mathbf{0}.$$

The propagation of the covariance is

$$\dot{\mathbf{P}} = \mathbf{F}\,\mathbf{P} + \mathbf{P}\,\mathbf{F}^{\mathrm{T}} + \mathbf{Q}$$

where

$$\mathbf{F} = \begin{bmatrix} -\left[(\boldsymbol{\omega}^m - \widehat{\mathbf{b}})\times\right] & -\mathbf{I}_{3\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} \end{bmatrix}$$

and

$$\mathrm{E}\left\{ \begin{bmatrix} \delta\boldsymbol{\omega}(t) \\ \boldsymbol{\nu}_b(t) \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\omega}^{\mathrm{T}}(\tau) & \boldsymbol{\nu}_b^{\mathrm{T}}(\tau) \end{bmatrix} \right\} = \mathbf{Q}\,\delta(t-\tau).$$

This example has shown how to incorporate the estimate of a star camera (which functions as a sub-filter) into the navigation filter. In order to maintain the optimality, the attitude sub-filter needs not only to pass its estimate to the navigation filter, but also the estimation error covariance. Failing to calculate the covariance with the techniques shown in Section 2.3.1, would result in non-optimal performance.

3.1.3 Correlated Sub-Filter and Attitude Sub-Filter Implementation

In the star tracker example of the previous section, the hypothesis that the measurements processed by the sub-filter depend only on the sub-filter state was met.

However this is not the general case. A magnetometer, for example, measures the local magnetic field. Such a measurement is a function of the spacecraft position. If the magnetometer measurement was to be processed in a sub-filter implementing Davenport-q algorithm, the estimated quaternion would be a function of the estimated spacecraft position. If the quaternion estimate obtained through Davenport-q algorithm was processed by the navigation filter as a measurement, this measurement would be correlated to the filter state, and such correlation should be taken into account. Figure 3.3 shows the architecture of this filter/sub-filter case.

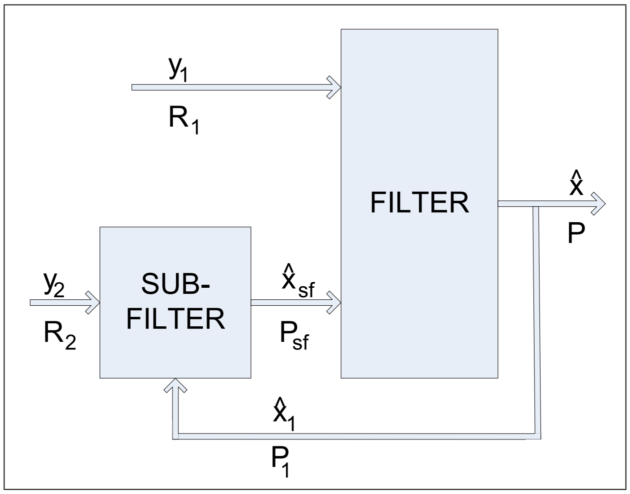


Figure 3.3: Main filter with raw measurements and a filtered state as inputs.

Using the notation of Section 3.1.1, $\mathbf{y}_2$ is now a function of the entire state vector, not of just $\mathbf{x}_2$. This scheme will not be optimal because $\mathbf{y}_2$ contains information on $\mathbf{x}_1$ that will not be used by the sub-filter. However, it can be made sub-optimal by correctly taking into consideration the correlation. Sub-optimal implies that $\widehat{\mathbf{x}}_2$ is globally optimal and $\widehat{\mathbf{x}}_1$ is optimal only given $\mathbf{y}_1$ and $\widehat{\mathbf{x}}_2$. The measurement $\mathbf{y}_2$ is modeled as

$$\mathbf{y}_2 = \mathbf{H}_2\mathbf{x} + \boldsymbol{\nu}_2 = \mathbf{H}_{2,1}\mathbf{x}_1 + \mathbf{H}_{2,2}\mathbf{x}_2 + \boldsymbol{\nu}_2.$$

The sub-filter only estimates $\mathbf{x}_2$, therefore part of the information is ignored, leading to the non-optimality of the estimation of $\mathbf{x}_1$. The component of the state vector $\mathbf{x}_1$ is modeled by the estimate of $\mathbf{x}_1$ from the central filter. The uncertainty associated with the estimate $\widehat{\mathbf{x}}_1$ needs to be added to the measurement noise in order for the sub-filter to be optimal. The estimated measurement is given by

$$\widehat{\mathbf{y}}_2 = \mathbf{H}_{2,1}\widehat{\mathbf{x}}_1 + \mathbf{H}_{2,2}\widehat{\mathbf{x}}_2.$$

The sub-filter only estimates $\mathbf{x}_{sf} = \mathbf{x}_2$. The residual is given by

$$\boldsymbol{\epsilon} = \mathbf{y}_2 - \widehat{\mathbf{y}}_2 = \mathbf{H}_{2,2}\mathbf{e}_{sf} + \mathbf{H}_{2,1}\mathbf{e}_1 + \boldsymbol{\nu}_2.$$

Effectively then, the *measurement* noise of the sub-filter is not only $\boldsymbol{\nu}_2$, but $\mathbf{H}_{2,1}\mathbf{e}_1 + \boldsymbol{\nu}_2$, where $\mathbf{e}_1$ is the estimation error of the central filter associated with $\mathbf{x}_1$. It is assumed that the sub-filter does not use an *a priori* estimate, i.e. $\mathbf{P}_{sf}^{-} = \infty$. Using the information formulation , we find that

$$\mathbf{P}_{sf} = \left[\mathbf{H}_{2,2}^{\mathrm{T}}\left(\mathbf{H}_{2,1}\mathbf{P}_{11}\mathbf{H}_{2,1}^{\mathrm{T}} + \mathbf{R}_2\right)^{-1}\mathbf{H}_{2,2}\right]^{-1} \tag{3-8a}$$

$$\mathbf{K}_{sf} = \mathbf{P}_{sf}\mathbf{H}_{2,2}^{\mathrm{T}}\left(\mathbf{H}_{2,1}\mathbf{P}_{11}\mathbf{H}_{2,1}^{\mathrm{T}} + \mathbf{R}_2\right)^{-1} \tag{3-8b}$$

$$\widehat{\mathbf{x}}_{sf} = \mathbf{K}\boldsymbol{\epsilon}, \tag{3-8c}$$

where $\mathbf{P}_{11}$ is the central filter error covariance associated with $\widehat{\mathbf{x}}_1$. Since the central filter estimation error of $\mathbf{x}_1$ affects the estimate of the sub-filter, there will be a correlation between the sub-filter estimate $\widehat{\mathbf{x}}_{sf}$ and the central filter estimate $\widehat{\mathbf{x}}_1$. The central filter is not going to recover optimality, however sub-optimality can be achieved through the use of the correlation

$$\mathbf{C}_k = \mathrm{E}\left\{\mathbf{e}_k^- \left(\mathbf{e}_{sf,k}\right)^{\mathrm{T}}\right\} = \begin{bmatrix} \mathrm{E}\left\{\mathbf{e}_{1,k}^- \left(\mathbf{e}_{sf,k}\right)^{\mathrm{T}}\right\} \\ \mathbf{O} \end{bmatrix}.$$

The central filter update equations are

$$\begin{aligned} \mathbf{P}_k^+ &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^- \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}} - \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{C}_k\mathbf{K}_k^{\mathrm{T}} \\ &\quad - \mathbf{K}_k\mathbf{C}_k^{\mathrm{T}}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^{\mathrm{T}} && \text{(3-9a)} \\ \mathbf{K}_k &= \left(\mathbf{H}_k\mathbf{P}_k^- + \mathbf{C}_k^{\mathrm{T}}\right)\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \mathbf{H}_k\mathbf{C}_k + \mathbf{C}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\right)^{-1} && \text{(3-9b)} \\ \widehat{\mathbf{x}}_k^+ &= \widehat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\widehat{\mathbf{x}}_{sf,k} - \mathbf{H}_k\widehat{\mathbf{x}}_k^-\right) && \text{(3-9c)} \\ \mathbf{H}_k &= \begin{bmatrix}\mathbf{O} & \mathbf{I}\end{bmatrix}. && \text{(3-9d)} \end{aligned}$$

A spacecraft implementing an attitude sub-filter would use this algorithm when some of the inertial reference vectors $\mathbf{n}_i$ are functions of position, such as the magnetometer and horizon sensor. In this attitude sub-filter example, the sub-filter implements Davenport's algorithm to estimate the attitude from the vector measurements. Therefore Eqs. (3-8a)–(3-8c) are not used, but instead are replaced by Eqs. (2.35) and (2.44). To compute the inertial reference vectors $\mathbf{n}_i$, the main filter passes to the sub-filter the position estimate and the position covariance. The sub-filter outputs the quaternion estimate, together with its covariance and the cross-covariance between the sub-filter quaternion estimate and position. The cross-covariance $\mathbf{P}_{r\theta}$ is calculated with Eq. (2.46). The state vector of the central filter is given by

$$\mathbf{x}^{\mathrm{T}} = \begin{bmatrix}\mathbf{r}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}} & \delta\boldsymbol{\alpha}^{\mathrm{T}}\end{bmatrix},$$

the measurement is given by the sub-filter's quaternion estimate

$$\bar{\mathbf{y}} = \widehat{\bar{\mathbf{q}}}_{sf}.$$

The central filter uses Eqs. (3-9a)–(3-9d) to update the state, the only difference is that replaces the additive residual $\widehat{\mathbf{x}}_{sf} - \mathbf{H}\widehat{\mathbf{x}}^-$ with twice the vector part of the multiplicative residual $\delta\bar{\mathbf{y}}$

$$\delta\bar{\mathbf{y}} = \widehat{\bar{\mathbf{q}}}_{sf} \otimes (\widehat{\bar{\mathbf{q}}}^-)^{-1}.$$

The correlation $\mathbf{C}$ between the measurement and the state is given by

$$\mathbf{C} = \begin{bmatrix} \mathbf{P}_{r\theta} \\ \mathbf{O}_{6\times3} \end{bmatrix}.$$

3.1.4 Parallel Filters

To achieve optimality all measurements and all states need to be processed and estimated in a single integrated filter. It was shown how to lower the burden of the central filter by processing some of the measurements in a sub-filter. This approach still leads to an optimal estimate if such measurements are uncorrelated to the remaining states of the central filter § 3.1.1. When the measurements processed by the sub-filter are correlated to the remaining states of the central filter, only the estimate of common states between the two filters will be optimal § 3.1.3. In this section, a parallel architecture will be presented. In this approach there is not a central filter that contains all the states, but two filters estimating a disjoint part of the state vector, as shown in Figure 3.4. As in the previous sections, the attitude filter employs the Davenport-q method to estimate the attitude, no *a priori* estimate is present and the filter functions exactly like the sub-filter previously presented. The derivation of the non-attitude filter is almost identical to the derivation in the previous section.

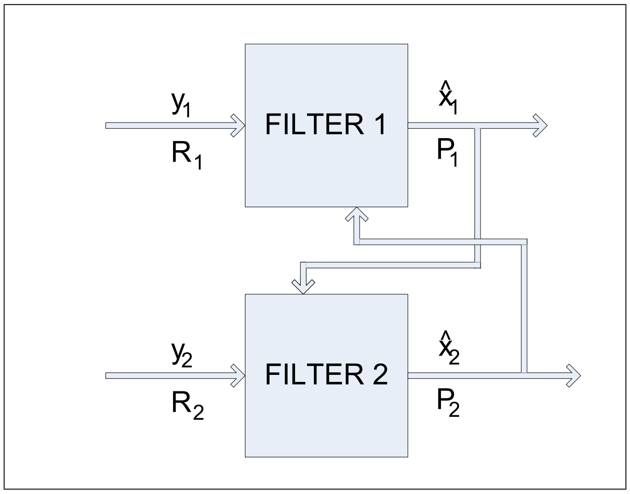


Figure 3.4: Parallel filters architecture.

Let $\mathbf{y}_1$ be the measurement processes by the non-attitude filter (filter 1). It is assumed that $\mathbf{y}_1$ is a function of the attitude, so that

$$\mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1, \overline{\mathbf{q}}) + \boldsymbol{\nu}_1 = \mathbf{h}_1(\mathbf{x}_1, \delta\overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}) + \boldsymbol{\nu}_1,$$

where the small rotation given by $\delta\overline{\mathbf{q}}$ is the actual rotation between the estimated quaternion $\widehat{\overline{\mathbf{q}}}$ and the true quaternion $\overline{\mathbf{q}}$. The estimated measurement is given by

$$\widehat{\mathbf{y}}_1 = \mathbf{h}_1(\widehat{\mathbf{x}}_1, \widehat{\overline{\mathbf{q}}}),$$

where $\widehat{\overline{\mathbf{q}}}$ is passed by the attitude filter. The residual is

$$\boldsymbol{\epsilon}_1 = \mathbf{y}_1 - \widehat{\mathbf{y}}_1 \simeq \mathbf{H}_{11}\mathbf{e}_1 + \mathbf{H}_{12}\delta\boldsymbol{\theta} + \boldsymbol{\nu}_1.$$

The vector $\delta\boldsymbol{\theta}$ is twice the vector component of the quaternion $\delta\overline{\mathbf{q}}$ and

$$\mathbf{H}_1 = [\mathbf{H}_{11}^{\mathrm{T}} \quad \mathbf{H}_{12}^{\mathrm{T}}]^{\mathrm{T}}$$

is the jacobian of $\mathbf{h}_1$. The actual "measurement" noise is given by $\mathbf{H}_{12}\delta\boldsymbol{\theta}+\boldsymbol{\nu}_1$. The update equations are

$$\begin{aligned}
\mathbf{P}_k^+ &= \left(\mathbf{I}-\mathbf{K}_k\mathbf{H}_{11,k}\right)\mathbf{P}_k^-\left(\mathbf{I}-\mathbf{K}_k\mathbf{H}_{11,k}\right)^{\mathrm{T}}+\mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}-\left(\mathbf{I}-\mathbf{K}_k\mathbf{H}_{11,k}\right)\mathbf{C}_k\mathbf{K}_k^{\mathrm{T}}+\\
&\qquad -\mathbf{K}_k\mathbf{C}_k^{\mathrm{T}}\left(\mathbf{I}-\mathbf{K}_k\mathbf{H}_{11,k}\right)^{\mathrm{T}}\\
\mathbf{R}_k &= \mathbf{H}_{12,k}\mathbf{P}_{\theta\theta,k}\mathbf{H}_{12,k}^{\mathrm{T}}+\mathrm{E}\left\{\boldsymbol{\nu}_{1,k}\boldsymbol{\nu}_{1,k}^{\mathrm{T}}\right\}\\
\mathbf{C}_k &= \begin{bmatrix}\mathbf{H}_{12,k}\mathbf{P}_{r\theta,k}^{\mathrm{T}} & \mathbf{O}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\\
\mathbf{K}_k &= \left(\mathbf{H}_{11,k}\mathbf{P}_k^-+\mathbf{C}_k^{\mathrm{T}}\right)\left(\mathbf{H}_{11,k}\mathbf{P}_k^-\mathbf{H}_{11,k}^{\mathrm{T}}+\mathbf{R}_k+\mathbf{H}_{11,k}\mathbf{C}_k+\mathbf{C}_k^{\mathrm{T}}\mathbf{H}_{11,k}^{\mathrm{T}}\right)^{-1}\\
\widehat{\mathbf{x}}_k^+ &= \widehat{\mathbf{x}}_k^-+\mathbf{K}_k\boldsymbol{\epsilon}_{1,k},
\end{aligned}$$

where $\mathbf{P}_{\theta\theta}$ and $\mathbf{P}_{r\theta}$ are provided by the attitude filter. It is assumed that the process and measurement noises are white and uncorrelated to each other. The derivation would become substantially more complex if the attitude filter had dynamics. But in such a situation it would be better to combine the two filters. The advantage of having a dedicated attitude filter is being able to use the nonlinear least-squares approach of the Davenport-q method. If this approach is replaced with a Kalman filter with dynamics it makes more sense to combine it with the Kalman filter for the translational states.

It is common for spacecrafts to have the navigation system independent from the attitude determination. These two systems however share estimates and therefore they output quantities correlated to each other. A common attitude determination system, Davenport-q algorithm, is a point-wise in time estimator, i.e. does not have dynamics. This section showed how to sub-optimally interconnect the two algorithms. Sub-optimality implies that each system is optimal with respect to the measurements it receives. Loss of information occurs because each system is not formulating an estimate using the measurements of the other system. Section 3.1.1 showed how to recovery optimality under two conditions. First the navigation did not only estimate the translation, but also the quaternion. The assumption might seem restrictive (why to have an attitude sub-filter when the navigation filter already estimates the attitude?) but Section 3.1.2 showed a practical example that falls in this category. Star cameras often have their own attitude estimation algorithm and output quaternions rather than star position measurements. The example in § 3.1.2 was a navigation filter having a star camera estimate as a measurement. This example is very much of current interest. Taking advantage of the absence of atmosphere on the Moon, NASA's current intent for human lunar missions is to update the state estimate with quaternions provided by a star tracker. The star camera example also satisfies the second assumption made in § 3.1.2, that measurements processes by the attitude sub-filter were independent from translational states. This second assumption was relaxed in Section 3.1.3.

3.2 Kalman Filter with Uncompensated Biases

One of the fundamental assumptions of the Kalman filter is that measurement and process noise are white. In practice however the error of a sensor can often be modelled more accurately as the sum of a white noise component and a strongly correlated component. The correlated component can either be a constant bias or a walking bias. The most straightforward technique to include the biases in the Kalman filter is to augment the state vector and estimate the biases. In the attempt to decouple the bias estimation from the state estimation, Friedland estimates the state as the bias was not present, and then adds the contribution of the bias. Friedland showed [69] that this approach is equivalent to augmenting the state vector. This technique, known as two-stage Kalman filter or separate-bias Kalman estimation, was then extended to incorporate a walk in the bias forced by white noise [70]. Since the process noise covariance was increased heuristically, optimality conditions were derived [71, 72].

In this section a completely different approach is taken. The effect of a constant random bias in the Kalman filter will be considered without estimating the bias itself. This is important, for example, when the bias is not observable, or when there is not enough information to discern the bias from the measurement. When this situation arises, the classical approach is to tune the filter such that the sample covariance obtained through Monte Carlo analysis matches the predicted covariance. The technique presented here is useful in quantifying the uncertainty due to a random bias in a single run, which would aid in tuning the filter.

The approach taken in this section is different from that of the consider filter [73, 74]. The consider filter can be designed to solve the same problem, and the two algorithms although different are equivalent.

3.2.1 Discrete Kalman Filter with Uncompensated Biases

Consider the stochastic system of difference equations

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{\Upsilon}_k \mathbf{b}_\nu + \boldsymbol{\nu}_k,$$

where $\boldsymbol{\nu}_k$ is process noise assumed to be a zero-mean, white noise sequence with

$$\mathrm{E}\left\{\boldsymbol{\nu}_k\right\} = \mathbf{0} \;\; \forall\, k, \qquad \mathrm{E}\left\{\boldsymbol{\nu}_j\, \boldsymbol{\nu}_k^{\mathrm{T}}\right\} = \mathbf{Q}_k\, \delta_{jk}.$$

Unlike the traditional Kalman filter, a random bias is also considered to be present. The bias has the assumed properties that

$$\mathrm{E}\left\{\mathbf{b}_\nu\right\} = \mathbf{0}, \qquad \mathrm{E}\left\{\mathbf{b}_\nu\, \mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{B}_\nu > \mathbf{O}, \qquad \mathrm{E}\left\{\boldsymbol{\nu}_k \mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{O} \;\; \forall k.$$

The shape matrix $\mathbf{\Upsilon}_k$ is deterministic. Since $\boldsymbol{\nu}_k$ and $\mathbf{b}_\nu$ are zero-mean, an unbiased estimation of the state $\widehat{\mathbf{x}}_{k-1}$ can be propagated forward in time to obtain an unbiased estimate at time t_k

$$\widehat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \widehat{\mathbf{x}}_k^+.$$

The estimation error at t_k before the measurement update is defined as

$$\mathbf{e}_k^- \triangleq \mathbf{x}_k - \widehat{\mathbf{x}}_k^-.$$

At t_k, it is assumed that a measurement is available of the form

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{\Lambda}_k \mathbf{b}_\eta + \boldsymbol{\eta}_k,$$

where

$$\begin{aligned}
&\mathrm{E}\left\{\boldsymbol{\eta}_k\right\} = \mathbf{0} \ \ \forall k, && \mathrm{E}\left\{\boldsymbol{\eta}_j\, \boldsymbol{\eta}_k^{\mathrm{T}}\right\} = \mathbf{R}_k\, \delta_{jk}, && \mathrm{E}\left\{\boldsymbol{\eta}_k \mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{O}, \\
&\mathrm{E}\left\{\mathbf{b}_\eta\right\} = \mathbf{0}, && \mathrm{E}\left\{\mathbf{b}_\eta\, \mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{B}_\eta > \mathbf{O} && \mathrm{E}\left\{\mathbf{b}_\nu \mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{O}, \\
&\mathrm{E}\left\{\boldsymbol{\eta}_j \boldsymbol{\nu}_k\right\} = \mathbf{O}, && \mathrm{E}\left\{\boldsymbol{\nu}_k \mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{O} && \mathrm{E}\left\{\mathbf{b}_\eta \boldsymbol{\eta}_k^{\mathrm{T}}\right\} = \mathbf{O},
\end{aligned}$$

for all k, j. The state update is assumed to be the linear update

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \widehat{\mathbf{y}}_k), \tag{3.10}$$

where

$$\widehat{\mathbf{y}}_k \triangleq \mathbf{H}_k \widehat{\mathbf{x}}_k.$$

The update in Eq. (3.10) provides an unbiased *a posteriori* estimate when the *a priori* estimate is unbiased. After the update, the estimation error is

$$\begin{aligned}
\mathbf{e}_k^+ = \mathbf{x}_k - \widehat{\mathbf{x}}_k^+ &= \mathbf{x}_k - \widehat{\mathbf{x}}_k^- - \mathbf{K}_k \left(\mathbf{H}_k \mathbf{x}_k + \mathbf{\Lambda}_k \mathbf{b}_\eta + \boldsymbol{\eta}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^-\right) \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{e}_k^- - \mathbf{K}_k \mathbf{\Lambda}_k \mathbf{b}_\eta - \mathbf{K}_k \boldsymbol{\eta}_k. \qquad (3.11)
\end{aligned}$$

The covariance update is given by

$$\mathbf{P}_k^+ = (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{\Lambda}_k\mathbf{B}_\eta\mathbf{\Lambda}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}} + \tag{3.12}$$
$$- (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\,\mathrm{E}\left\{\mathbf{e}_k^-\mathbf{b}_\eta^{\mathrm{T}}\right\}\mathbf{\Lambda}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} - \mathbf{K}_k\mathbf{\Lambda}_k\,\mathrm{E}\left\{\mathbf{b}_\eta(\mathbf{e}_k^-)^{\mathrm{T}}\right\}(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}}.$$

assuming $\boldsymbol{\eta}_k$ and $\mathbf{b}_\eta$ are uncorrelated to the initial estimation error (a good assumption).

After propagation to the next measurement at time t_{k+1}, the estimation error is

$$\mathbf{e}_{k+1}^- = \mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{\Upsilon}_k\mathbf{b}_\nu + \boldsymbol{\nu}_k - \mathbf{\Phi}_k\widehat{\mathbf{x}}_k^+ = \mathbf{\Phi}_k\mathbf{e}_k^+ + \mathbf{\Upsilon}_k\mathbf{b}_\nu + \boldsymbol{\nu}_k. \tag{3.13}$$

The covariance propagation is given by

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k\mathbf{P}_k^+\mathbf{\Phi}_k^{\mathrm{T}} + \mathbf{\Upsilon}_k\mathbf{B}\mathbf{\Upsilon}_k^{\mathrm{T}} + \mathbf{Q}_k + \mathbf{\Phi}_k\,\mathrm{E}\left\{\mathbf{e}_k^+\mathbf{b}_\nu^{\mathrm{T}}\right\}\mathbf{\Upsilon}_k^{\mathrm{T}} + \mathbf{\Upsilon}_k\,\mathrm{E}\left\{\mathbf{b}_\nu(\mathbf{e}_k^+)^{\mathrm{T}}\right\}\mathbf{\Phi}_k^{\mathrm{T}},$$

assuming $\boldsymbol{\nu}_k$ and $\mathbf{b}_\nu$ are uncorrelated to the initial estimation error (a good assumption).

Estimation Error

Substituting Eq. (3.11) into Eq. (3.13) yields to the recurrence relation

$$\mathbf{e}_{k+1}^- = \mathbf{\Phi}_k\left[(\mathbf{I}-\mathbf{K}_k\,\mathbf{H}_k)\,\mathbf{e}_k^- - \mathbf{K}_k\,\boldsymbol{\eta}_k - \mathbf{K}_k\,\mathbf{\Lambda}_k\,\mathbf{b}_\eta\right] + \mathbf{\Upsilon}_k\mathbf{b}_\nu + \boldsymbol{\nu}_k.$$

Forming $\mathbf{e}_{k+1}^-\,\mathbf{b}_\eta^{\mathrm{T}}$ and taking the expectation, it follows that

$$\mathrm{E}\left\{\mathbf{e}_{k+1}^-\,\mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{\Phi}_k\,(\mathbf{I}-\mathbf{K}_k\,\mathbf{H}_k)\,\mathrm{E}\left\{\mathbf{e}_k^-\,\mathbf{b}_\eta^{\mathrm{T}}\right\} - \mathbf{\Phi}_k\mathbf{K}_k\,\mathbf{\Lambda}_k\mathbf{B}_\eta. \tag{3.14}$$

Defining

$$\mathrm{E}\left\{\mathbf{e}_k^-\,\mathbf{b}_\eta^{\mathrm{T}}\right\} \triangleq \mathbf{M}_k\,\mathbf{B}_\eta, \tag{3.15}$$

and using Eq. (3.14), the matrix $\mathbf{M}_k$ can be found recursively as

$$\mathbf{M}_{k+1} = \mathbf{\Phi}_k \left[(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{M}_k - \mathbf{K}_k\mathbf{\Lambda}_k\right].$$

If, at the initial time, a propagation occurs such that

$$\mathbf{e}_1^- = \mathbf{\Phi}_0\mathbf{e}_0 + \mathbf{\Upsilon}_0\mathbf{b}_\nu + \boldsymbol{\nu}_0,$$

then from Eq. (3.15)

$$\mathrm{E}\left\{\mathbf{e}_1^-\,\mathbf{b}_\eta^{\mathrm{T}}\right\} = \mathbf{O} \quad \text{implies that} \quad \mathbf{M}_1 = \mathbf{O},\ \text{since}\ \mathbf{B}_\eta > \mathbf{O}.$$

Similarly, using Eqs. (3.11) and (3.13), it follows that

$$\mathbf{e}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\left(\mathbf{\Phi}_k\mathbf{e}_k^+ + \mathbf{\Upsilon}_k\mathbf{b}_\nu + \boldsymbol{\nu}_k\right) - \mathbf{K}_{k+1}\mathbf{\Lambda}_{k+1}\mathbf{b}_\eta - \mathbf{K}_{k+1}\boldsymbol{\eta}_{k+1}.$$

Forming $\mathbf{e}_{k+1}^+\,\mathbf{b}_\nu^{\mathrm{T}}$ and taking the expectation yields

$$\mathrm{E}\left\{\mathbf{e}_{k+1}^+\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\left[\mathbf{\Phi}_k\,\mathrm{E}\left\{\mathbf{e}_k^+\,\mathbf{b}_\nu^{\mathrm{T}}\right\} + \mathbf{\Upsilon}_k\mathbf{B}_\nu\right].$$

Define

$$\mathrm{E}\left\{\mathbf{e}_k^+\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{L}_k\mathbf{B}_\nu.$$

Then

$$\mathrm{E}\left\{\mathbf{e}_{k+1}^+\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\left[\mathbf{\Phi}_k\mathbf{L}_k + \mathbf{\Upsilon}_k\right]\mathbf{B}_\nu = \mathbf{L}_{k+1}\mathbf{B}_\nu,$$

where

$$\mathbf{L}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})(\mathbf{\Phi}_k\mathbf{L}_k + \mathbf{\Upsilon}_k). \tag{3.16}$$

After the first update, we have

$$\mathbf{e}_1^+ = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\,(\boldsymbol{\Phi}_0\mathbf{e}_0 + \boldsymbol{\Upsilon}_0\mathbf{b}_\nu + \boldsymbol{\nu}_0) - \mathbf{K}_1\boldsymbol{\Lambda}_1\mathbf{b}_\eta - \mathbf{K}_1\boldsymbol{\eta}_1.$$

Computing $\mathbf{e}_1^+\,\mathbf{b}_\nu^{\mathrm{T}}$ and taking the expectation yields

$$\mathrm{E}\left\{\mathbf{e}_1^+\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\,\Upsilon_0\mathbf{B}_\nu$$

since $\mathrm{E}\left\{\mathbf{e}_0\mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{O}$. Therefore, we find that

$$\mathbf{L}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\boldsymbol{\Upsilon}_0,$$

which can be obtained using the recursion of Eq. (3.16) for $k = 0$ with $\mathbf{L}_0 = \mathbf{O}$.

Optimal Kalman Gain

Substituting Eq. (3.15) into Eq. (3.12), after some rearrangement, we obtain

$$\begin{aligned}\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\big(\mathbf{H}_k\mathbf{P}_k^- + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\big) - \big(\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}}\big)\mathbf{K}_k^{\mathrm{T}} + \\ + \mathbf{K}_k\big(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \mathbf{H}_k\mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\big)\mathbf{K}_k^{\mathrm{T}}.\end{aligned}$$

Taking the derivative of the trace of $\mathbf{P}_k^+$ with respect to $\mathbf{K}_k$ yields

$$\begin{aligned}\mathcal{J}' = \frac{d}{d\mathbf{K}_k}\,\mathrm{trace}(\mathbf{P}_k^+) = -\big(\mathbf{H}_k\mathbf{P}_k^- + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\big)^{\mathrm{T}} - \big(\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}}\big) + \\ + 2\mathbf{K}_k\big(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \mathbf{H}_k\mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\big).\end{aligned}$$

Setting $\mathcal{J}' = \mathbf{O}$ and solving for $\mathbf{K}_k$ yields the optimal gain,

$$\mathbf{K}_k = \big(\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}}\big)\mathbf{W}_k^{-1}.$$

The matrix $\mathbf{W}_k$ is the covariance of the residuals, as is found to be

$$\begin{aligned}\mathbf{W}_k &\triangleq \mathrm{E}\left\{\boldsymbol{\epsilon}_k\,\boldsymbol{\epsilon}_k^{\mathrm{T}}\right\} = \mathrm{E}\left\{\left(\mathbf{y}-\widehat{\mathbf{y}}\right)\left(\mathbf{y}-\widehat{\mathbf{y}}\right)^{\mathrm{T}}\right\} = \\ &= \mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \mathbf{H}_k\mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}.\end{aligned}$$

Table 3.1 summarizes the discrete-time Kalman filter algorithm with uncompensated bias. Notice that when the biases are absent, the filter reduces to the standard Kalman filter. It was assumed that at the initial time a propagation will occur first, and the first update will follow. If an update occurs at time t_0 before the first propagation, the same algorithm can be used by setting

$$\mathbf{M}_0 = \mathbf{O}, \quad \mathbf{L}_0 = \mathbf{O}.$$

System Model	$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \mathbf{x}_k + \boldsymbol{\Upsilon}_k \mathbf{b}_\nu + \boldsymbol{\nu}_k, \quad \mathrm{E}\left\{\boldsymbol{\nu}_k\right\} = \mathbf{O}$
	$\mathrm{E}\left\{\mathbf{b}_\nu(t_i)\mathbf{b}_\nu^{\mathrm{T}}(t_j)\right\} = \mathbf{B}_\nu > \mathbf{O}, \quad \mathrm{E}\left\{\mathbf{b}_\nu(t_i)\boldsymbol{\nu}_j^{\mathrm{T}}\right\} = \mathbf{O}\ \forall i,j$
	$\mathrm{E}\left\{\boldsymbol{\nu}_i\boldsymbol{\nu}_j^{\mathrm{T}}\right\} = \mathbf{Q}_i\delta_{ij}, \quad \mathrm{E}\left\{\mathbf{b}_\eta\, \mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{O}$
Measurement Model	$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \boldsymbol{\Upsilon}_k\mathbf{b}_\eta + \boldsymbol{\eta}_k, \quad \mathrm{E}\left\{\boldsymbol{\eta}_k\right\} = \mathbf{O}$
	$\mathrm{E}\left\{\mathbf{b}_\eta(t_i)\mathbf{b}_\eta^{\mathrm{T}}(t_j)\right\} = \mathbf{B}_\eta > \mathbf{O}, \quad \mathrm{E}\left\{\mathbf{b}_\eta(t_i)\boldsymbol{\eta}_j^{\mathrm{T}}\right\} = \mathbf{O}\ \forall i,j$
	$\mathrm{E}\left\{\boldsymbol{\eta}_i\boldsymbol{\nu}_j^{\mathrm{T}}\right\} = \mathbf{O}\ \forall i,j, \quad \mathrm{E}\left\{\boldsymbol{\eta}_i\boldsymbol{\eta}_j^{\mathrm{T}}\right\} = \mathbf{R}_i\delta_{ij}$
Initial Conditions	$\widehat{\mathbf{x}}_0 = \mathrm{E}\left\{\mathbf{x}(t_0)\right\}, \quad \mathbf{P}_0 = \mathrm{E}\left\{\mathbf{e}_0\mathbf{e}_0^{\mathrm{T}}\right\}$
Matrices Initialization	$\mathbf{M}_1 = \mathbf{O}, \quad \mathbf{L}_0 = \mathbf{O}$
State Propagation	$\widehat{\mathbf{x}}_k^- = \boldsymbol{\Phi}_{k-1}\widehat{\mathbf{x}}_{k-1}^+$
Covariance Prop.	$\mathbf{P}_k^- = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^{\mathrm{T}} + \boldsymbol{\Upsilon}_{k-1}\mathbf{B}_\nu\boldsymbol{\Upsilon}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} +$
	$+\boldsymbol{\Phi}_{k-1}\mathbf{L}_{k-1}\mathbf{B}_\nu\boldsymbol{\Upsilon}_{k-1}^{\mathrm{T}} + \boldsymbol{\Upsilon}_{k-1}\mathbf{B}_\nu\mathbf{L}_{k-1}^{\mathrm{T}}\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}$
M Calculation	$\mathbf{M}_k = \boldsymbol{\Phi}_{k-1}\left[(\mathbf{I} - \mathbf{K}_{k-1}\mathbf{H}_{k-1})\mathbf{M}_{k-1} - \mathbf{K}_{k-1}\boldsymbol{\Lambda}_{k-1}\right]$
Gain Calculation	$\mathbf{K}_k = \left(\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}}\right)\mathbf{W}_k^{-1}$
	$\mathbf{W}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \mathbf{H}_k\mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} +$
	$+\boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}$
L Calculation	$\mathbf{L}_k = (\mathbf{I} - \mathbf{K}_{k+}\mathbf{H}_k)(\boldsymbol{\Phi}_{k-1}\mathbf{L}_{k-1} + \boldsymbol{\Upsilon}_{k-1})$
State Update	$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-)$
Covariance Update	$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{W}_k\mathbf{K}_k^{\mathrm{T}}$

Table 3.1: Discrete-time Kalman filter with uncompensated bias.

3.2.2 Continuous Kalman Filter with Uncompensated Biases

The continuous time Kalman filter is considered here. The system model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\,\mathbf{x}(t) + \boldsymbol{\Upsilon}(t)\,\mathbf{b}_\nu + \boldsymbol{\nu}(t), \quad \mathrm{E}\left\{\boldsymbol{\nu}(t)\right\} = \mathbf{O}\ \forall t \tag{3-17a}$$

$$\mathrm{E}\left\{\mathbf{b}_\nu(t)\mathbf{b}_\nu(\tau)^{\mathrm{T}}\right\} = \mathbf{B}_\nu\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{b}_\nu(t)\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{B}_\nu\right\} = \mathbf{O} \tag{3-17b}$$

$$\mathrm{E}\left\{\boldsymbol{\nu}(t)\,\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{Q}(t)\delta(t-\tau), \quad \mathrm{E}\left\{\mathbf{b}_\eta\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{O} \tag{3-17c}$$

$$\mathbf{y}(t) = \mathbf{H}(t)\,\mathbf{x}(t) + \boldsymbol{\Upsilon}(t)\,\mathbf{b}_\eta + \boldsymbol{\eta}(t), \quad \mathrm{E}\left\{\boldsymbol{\eta}(t)\right\} = \mathbf{O} \tag{3-17d}$$

$$\mathrm{E}\left\{\boldsymbol{\eta}(t)\,\boldsymbol{\eta}(\tau)^{\mathrm{T}}\right\} = \mathbf{R}(t)\delta(t-\tau), \quad \mathrm{E}\left\{\boldsymbol{\eta}(t)\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\,\forall t,\tau \tag{3-17e}$$

$$\mathrm{E}\left\{\mathbf{b}_\eta(t)\mathbf{b}_\eta^{\mathrm{T}}(\tau)\right\} = \mathbf{B}_\eta\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{b}_\eta(t)\boldsymbol{\eta}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{B}_\eta\right\} = \mathbf{O}. \tag{3-17f}$$

The continuous formulation is obtained from the discrete formulation by making $t_{k+1} \rightarrow t_k + dt$.

The discrete update was given by

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-)$$

in between measurements, the propagation of the estimate is given by

$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{F}(t)\,\widehat{\mathbf{x}}(t), \quad t_k \le t \le t_k + dt,$$

with initial condition $\widehat{\mathbf{x}}_k^+$. As $\mathbf{d}t \rightarrow 0$

$$\widehat{\mathbf{x}}_{k+1}^- = \widehat{\mathbf{x}}(t_k + dt) \rightarrow \widehat{\mathbf{x}}(t_k) + \mathbf{F}(t_k)\,\widehat{\mathbf{x}}(t_k)dt = \left[\mathbf{I} + \mathbf{F}(t_k)dt\right]\ \left[\widehat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-)\right]$$

The derivative of the estimate with continuous measurements is given by

$$\dot{\widehat{\mathbf{x}}}(t_k) = \lim_{dt\rightarrow 0}\frac{\widehat{\mathbf{x}}_{k+1}^- - \widehat{\mathbf{x}}_k^-}{dt} = \mathbf{F}(t_k)\widehat{\mathbf{x}}_k^- + \frac{\mathbf{I} + \mathbf{F}(t_k)dt}{dt}\mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-) \tag{3.18}$$

The residuals covariance $\mathbf{W}_k$ is given by

$$\mathbf{W}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \mathbf{H}_k\mathbf{M}_k\,\mathbf{B}_\eta\boldsymbol{\Lambda}_k^{\mathrm{T}} + \boldsymbol{\Lambda}_k\mathbf{B}_\eta\mathbf{M}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}.$$

Matrix $\mathbf{R}_k$ is a covariance, not a spectral density. Continuous white noise has infinite covariance given by $\mathbf{R}_k(t) = \lim_{dt\to 0}\mathbf{R}(t)\delta(dt)$, where $\mathbf{R}(t_k)$ is a spectral density. It will be shown that $\mathbf{M}(t)$ is bounded, hence the term in the limit dominates and

$$\mathbf{W}(t_k) = \lim_{dt\to 0}\mathbf{R}(t_k)\delta(dt). \tag{3.19}$$

The discrete optimal gain is therefore given by

$$\mathbf{K}_k(t) = \left(\mathbf{P}^-(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{M}(t)\,\mathbf{B}_\eta\boldsymbol{\Lambda}(t)^{\mathrm{T}}\right)\,\left[\mathbf{R}(t)\delta(dt)\right]^{-1}, \tag{3.20}$$

substituting into Eq. (3.18) and replacing t_k with t

$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{F}(t)\widehat{\mathbf{x}}(t) + \mathbf{K}(t)\big[\mathbf{y}(t) - \mathbf{H}(t)\widehat{\mathbf{x}}(t)\big],$$

where

$$\mathbf{K}(t) = \left(\mathbf{P}^-(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{M}(t)\,\mathbf{B}_\eta\boldsymbol{\Lambda}(t)^{\mathrm{T}}\right)\mathbf{R}(t)^{-1} \neq \mathbf{K}_k(t).$$

From Eq. (3.18) the estimation error evolves as

$$\dot{\mathbf{e}}(t) = \mathbf{F}(t)\,\mathbf{e}(t) + \boldsymbol{\Upsilon}(t)\,\mathbf{b}_\nu + \boldsymbol{\nu}(t) \qquad \mathbf{e}(t_k) = \mathbf{e}_k^+. \tag{3.21}$$

Integrating Eq. (3.21) from t_k to $t_{k+1} = t_k + dt$ yields

$$\mathbf{e}_{k+1}^- \triangleq \mathbf{e}(t_k + dt) = \boldsymbol{\Phi}(t_k + dt, t_k)\mathbf{e}_k^+ + \int_{t_k}^{t_k+dt}\boldsymbol{\Phi}(t_k + dt, \tau)\left[\boldsymbol{\Upsilon}(\tau)\mathbf{b}_\nu + \boldsymbol{\nu}(\tau)\right]d\tau.$$

Computing $\mathbf{P}_{k+1}^{-} = \mathrm{E}\left\{\mathbf{e}_{k+1}^{-}(\mathbf{e}_{k+1}^{-})^{\mathrm{T}}\right\}$ yields

$$\begin{aligned}
\mathbf{P}_{k+1}^{-} = & \, \boldsymbol{\Phi}(t_k+dt,t_k)\mathbf{P}(t_k)^{+}\boldsymbol{\Phi}(t_k+dt,t_k)^{\mathrm{T}}+ \\
& + \int_{t_k}^{t_k+dt} \boldsymbol{\Phi}(t_k+dt,\tau)\mathbf{Q}(t)\boldsymbol{\Phi}(t_k+dt,\tau)^{\mathrm{T}}\, d\tau + \\
& + \left[\int_{t_k}^{t_k+dt} \boldsymbol{\Phi}(t_k+dt,\tau)\boldsymbol{\Upsilon}(\tau)d\tau\right]\mathbf{B}_{\nu}\left[\int_{t_k}^{t_k+dt} \boldsymbol{\Phi}(t_k+dt,\tau)\boldsymbol{\Upsilon}(\tau)d\tau\right]^{\mathrm{T}} + \\
& + \boldsymbol{\Phi}(t_k+dt,t_k)\mathbf{L}_k\mathbf{B}_{\nu}\left[\int_{t_k}^{t_k+dt} \boldsymbol{\Phi}(t_k+dt,\tau)\boldsymbol{\Upsilon}(\tau)d\tau\right]^{\mathrm{T}} + \\
& + \left[\int_{t_k}^{t_k+dt} \boldsymbol{\Phi}(t_k+dt,\tau)\boldsymbol{\Upsilon}(\tau)d\tau\right]\mathbf{B}_{\nu}\mathbf{L}_k^{\mathrm{T}}\boldsymbol{\Phi}(t_k+dt,t_k)^{\mathrm{T}}.
\end{aligned}$$

As $dt \to 0$ it follows that

$$\begin{aligned}
\mathbf{P}(t_k+dt)^{-} \to [\mathbf{I}+\mathbf{F}(t_k)\,dt]\Big\{\mathbf{P}_k^{+} + \mathbf{Q}(t_k)\,dt + \boldsymbol{\Upsilon}(t_k)\mathbf{B}_{\nu}\boldsymbol{\Upsilon}(t_k)^{\mathrm{T}}\,dt^2 + \\
+ \mathbf{L}_k\mathbf{B}_{\nu}\boldsymbol{\Upsilon}(t_k)^{\mathrm{T}}\,dt + \boldsymbol{\Upsilon}(t_k)\mathbf{B}_{\nu}\mathbf{L}_k^{\mathrm{T}}dt\Big\}\,[\mathbf{I}+\mathbf{F}(t_k)\,dt]^{\mathrm{T}}. \qquad (3.22)
\end{aligned}$$

The updated covariance is $\mathbf{P}_k^{+} = \mathbf{P}_k^{-} - \mathbf{K}_k\mathbf{W}_k\mathbf{K}_k^{\mathrm{T}}$, it then follows that replacing t_k with t into Eq. (3.22), and using Eqs. (3.19), (3.20)

$$\begin{aligned}
\dot{\mathbf{P}}(t) &= \lim_{dt\to 0}\frac{\mathbf{P}(t+dt)^{-} - \mathbf{P}(t)^{-}}{dt} \\
&= \mathbf{F}(t)\mathbf{P}^{-}(t) + \mathbf{P}^{-}(t)\mathbf{F}(t) + \mathbf{Q}(t) + \mathbf{L}(t)\mathbf{B}_{\nu}\boldsymbol{\Upsilon}(t)^{\mathrm{T}} + \boldsymbol{\Upsilon}(t)\mathbf{B}_{\nu}\mathbf{L}(t)^{\mathrm{T}} + \\
&\quad + \left(\mathbf{P}^{-}(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{M}(t)\,\mathbf{B}_{\eta}\boldsymbol{\Lambda}(t)^{\mathrm{T}}\right)\mathbf{R}(t)^{-1}\left(\mathbf{P}^{-}(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{M}(t)\,\mathbf{B}_{\eta}\boldsymbol{\Lambda}(t)^{\mathrm{T}}\right)^{\mathrm{T}}.
\end{aligned}$$

Similarly

$$\begin{aligned}
\dot{\mathbf{M}}(t) &= \lim_{dt\to 0}\frac{\mathbf{M}(t+dt) - \mathbf{M}(t)}{dt} \\
&= \lim_{dt\to 0}\left\{[\mathbf{I}+\mathbf{F}(t)dt]\,[(\mathbf{I}-\mathbf{K}(t)\mathbf{H}(t))\,\mathbf{M}(t) - \mathbf{K}(t)\boldsymbol{\Lambda}(t)] - \mathbf{M}(t)\right\}/dt \\
&= \mathbf{F}(t)\,\mathbf{M}(t) - \mathbf{K}(t)\left[\mathbf{H}(t)\,\mathbf{M}(t) + \boldsymbol{\Lambda}(t)\right].
\end{aligned}$$

System Model	$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\,\mathbf{x}(t) + \boldsymbol{\Upsilon}(t)\,\mathbf{b}_\nu + \boldsymbol{\nu}(t), \quad \mathrm{E}\left\{\boldsymbol{\nu}(t)\right\} = \mathbf{O}$
	$\mathrm{E}\left\{\mathbf{b}_\nu(t)\mathbf{b}_\nu^{\mathrm{T}}(\tau)\right\} = \mathbf{B}_\nu\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{b}_\nu(t)\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\ \forall t,\tau$
	$\mathrm{E}\left\{\boldsymbol{\nu}(t)\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{Q}(t)\delta_{t-\tau}, \quad \mathrm{E}\left\{\boldsymbol{\eta}(t)\boldsymbol{\nu}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\,\forall t,\tau$
Measurement Model	$\mathbf{y}(t) = \mathbf{H}(t)\,\mathbf{x}(t) + \boldsymbol{\Upsilon}(t)\,\mathbf{b}_\eta + \boldsymbol{\eta}(t), \quad \mathrm{E}\left\{\boldsymbol{\eta}(t)\right\} = \mathbf{O}$
	$\mathrm{E}\left\{\mathbf{b}_\eta(t)\mathbf{b}_\eta^{\mathrm{T}}(\tau)\right\} = \mathbf{B}_\eta\ \forall t,\tau, \quad \mathrm{E}\left\{\mathbf{b}_\eta(t)\boldsymbol{\eta}(\tau)^{\mathrm{T}}\right\} = \mathbf{O}\ \forall t,\tau$
	$\mathrm{E}\left\{\mathbf{b}_\eta\,\mathbf{b}_\nu^{\mathrm{T}}\right\} = \mathbf{O}, \quad \mathrm{E}\left\{\boldsymbol{\eta}(t)\,\boldsymbol{\eta}(\tau)^{\mathrm{T}}\right\} = \mathbf{R}(t)\,\delta(t-\tau)$
Initial Conditions	$\widehat{\mathbf{x}}_0 = \mathrm{E}\left\{\mathbf{x}(t_0)\right\}, \quad \mathbf{P}_0 = \mathrm{E}\left\{\mathbf{e}(t_0)\,\mathbf{e}(t_0)^{\mathrm{T}}\right\}$
	$\mathbf{M}(t_0) = \mathbf{O}, \quad \mathbf{L}(t_0) = \mathbf{O}$
Gain Calculation	$\mathbf{K}(t) = \left(\mathbf{P}^-(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{M}(t)\,\mathbf{B}_\eta\boldsymbol{\Lambda}(t)^{\mathrm{T}}\right)\mathbf{R}(t)^{-1}$
State Estimate	$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{F}(t)\,\widehat{\mathbf{x}}(t) + \mathbf{K}(t)[\mathbf{y}(t) - \mathbf{H}(t)\,\widehat{\mathbf{x}}(t)]$
M Calculation	$\dot{\mathbf{M}}(t) = \mathbf{F}(t)\,\mathbf{M}(t) - \mathbf{K}(t)\left[\mathbf{H}(t)\,\mathbf{M}(t) + \boldsymbol{\Lambda}(t)\right]$
L Calculation	$\dot{\mathbf{L}}(t) = \mathbf{F}(t)\,\mathbf{L}(t) + \boldsymbol{\Upsilon}(t) - \mathbf{K}(t)\,\mathbf{H}(t)\left[\mathbf{L}(t) + \boldsymbol{\Upsilon}(t)\right]$
Covariance	$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t) + \mathbf{Q}(t) + \mathbf{L}(t)\mathbf{B}_\nu\boldsymbol{\Upsilon}(t)^{\mathrm{T}} +$
	$+\boldsymbol{\Upsilon}(t)\mathbf{B}_\nu\mathbf{L}(t)^{\mathrm{T}} + \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}(t)^{\mathrm{T}}$

Table 3.2: Continuous-time Kalman filter with uncompensated bias.

Finally

$$\begin{aligned}
\dot{\mathbf{L}}(t) &= \lim_{dt\to 0} \frac{\mathbf{L}(t+dt) - \mathbf{L}(t)}{dt} \\
&= \lim_{dt\to 0} \left(\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t)\right)\left[\left(\mathbf{I} + \mathbf{F}(t)dt\right)\mathbf{L}(t) + \boldsymbol{\Upsilon}(t)\right]/dt \\
&= \mathbf{F}(t)\,\mathbf{L}(t) + \boldsymbol{\Upsilon}(t) - \mathbf{K}(t)\,\mathbf{H}(t)\left[\mathbf{L}(t) + \boldsymbol{\Upsilon}(t)\right].
\end{aligned}$$

Table 3.2 summarizes the continuous time algorithm. Notice that in the absence of biases the filter reduces to the Kalman-Bucy filter. For finite times, $\mathbf{M}(t)$ will stay bounded as long as $\mathbf{R}(t)$ is non-singular.

3.3 Proposed Gating Network

The MMAE scheme employed here is a modified version of that of Chaer et al. [31,32] introduced in §2.4. The gating network as described before, presents numerical problems due to Eq. (2.49) [75]. In addition, the dependence on $\mathbf{y}_k$ in Eq. (2.47) may create problems when the measurement vector rapidly changes with time [76]. The vectors $\mathbf{u}_i$ should orientate with hyper-planes to create the accept/reject zones, however the measurement will change due to the dynamics of the problem. If the frequency with which the measurements are available is not high enough, the vectors $\mathbf{u}_i$ will not be able to represent the accept/reject zone correctly. As presented in §2.4, the MMAE scheme requires that all measurements are synchronized and available at the same time.

For this work, $\mathbf{y}_k^{\mathrm{T}}\mathbf{u}_i$ of Eq. (2.47) is replaced with a scalar u_i to eliminate the dependence on $\mathbf{y}_k$. Also a scalar voids the requirement that all measurements are synchronized,

$$w_i = \frac{e^{u_i}}{\Sigma_{i=1}^{L} e^{u_i}}. \tag{3.23}$$

The filter weight w_i is interpreted as a probability, hence the probability of the entire bank is

$$f(\mathbf{y}_k) = \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) P(\boldsymbol{\alpha}_i) = \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) w_i,$$

where L is the number of filters in the bank. The goal is to maximize the probability of the bank. In order to maximize this probability density, it is easier to work with the natural logarithm of $f(\mathbf{y}_k)$, or

$$l \triangleq \ln f(\mathbf{y}_k) = \ln \sum_{i=1}^{L} f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) e^{u_i} - \ln \sum_{i=1}^{L} e^{u_i}.$$

Taking the derivative of l with respect to u_i yields

$$\frac{\partial l}{\partial u_i} = P(\boldsymbol{\alpha}_i \mid \mathbf{y}_k) - w_i, \tag{3.24}$$

where

$$P(\boldsymbol{\alpha}_i \mid \mathbf{y}_k) = \frac{f(\mathbf{y}_k \mid \boldsymbol{\alpha}_i) w_i}{f(\mathbf{y}_k)}.$$

Eq. (3.24) shows the direction of maximum growth of the function l. The update is accomplished via

$$u_i \leftarrow u_i + \lambda \frac{\partial l}{\partial u_i}, \tag{3.25}$$

where λ is a learning rate parameter. The gating network gains can now be computed with Eq. (3.23). The scalar u_i can be interpreted as a measure of how likely the i^{th} filter is to be the best performing filter within the bank. The higher the value of u_i, the higher the likelihood that it is the best performing filter. Notice that u_i cannot be interpreted as a probability since $u_i \in \Re$. Eq. (3.25) can be rewritten as

$$u_i \leftarrow u_i + \lambda \left[P(\boldsymbol{\alpha}_i \mid \mathbf{y}_k) - P(\boldsymbol{\alpha}_i) \right]$$

which is intuitive in the following sense: the updated u_i starts from the old value, increases if the probability associated with the last measurement is larger than the old probability, decreases otherwise. The larger the learning rate parameter, the larger the current measurements are weighted. For $\lambda = 0$ the gains do not update, for $\lambda \to \infty$ the filter with higher probability after measurement $\mathbf{y}_k$ will be given probability one, all others will be given probability zero.

Once the filter weights are computed, the state estimate can be chosen to be the state estimate associated with the winning filter, or it can be a weighted average of the L filters in the bank. In the latter case, the estimated state is given by

$$\widehat{\mathbf{x}} = \sum_{i=1}^{L} w_i \widehat{\mathbf{x}}_i.$$

The estimation error associated with the weighted state estimate is

$$\mathbf{e} = \sum_{i=1}^{L} w_i \left(\widehat{\mathbf{x}}_i - \mathbf{x}_i \right) = \sum_{i=1}^{L} w_i \, \mathbf{e}_i,$$

and the estimation error covariance is

$$\mathbf{P} = \mathrm{E}\left\{ \mathbf{e}\,\mathbf{e}^{\mathrm{T}} \right\} = \sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \, \mathrm{E}\left\{ \mathbf{e}_i \; \mathbf{e}_j^{\mathrm{T}} \right\} = \sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \mathbf{P}_{ij},$$

where $\mathbf{P}_{ii}$ is the autocovariance of the i^{th} filter, and $\mathbf{P}_{ij},\ i \neq j$ is the crosscovariance between filters i and j. Recall that

$$\mathbf{e}_i^+ \simeq \mathbf{e}_i^- + \mathbf{K}_i \mathbf{H}_i \mathbf{e}_i^- + \mathbf{K}_i \, \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ is the measurement noise which is a common quantity for all filters in the bank. Then it follows that

$$\mathbf{P}_{ij}^+ = (\mathbf{I} - \mathbf{K}_i \mathbf{H}_i) \mathbf{P}_{ij}^- (\mathbf{I} - \mathbf{K}_j \mathbf{H}_j)^{\mathrm{T}} + \mathbf{K}_i \mathbf{R} \mathbf{K}_j^{\mathrm{T}},$$

where $\mathbf{R}$ is the measurement noise autocovariance, which is also the cross-covariance because the filters share the sensors. Similarly, the propagation is given by

$$\mathbf{P}_{ij}(t) = \boldsymbol{\Phi}_i(t, t_k) \mathbf{P}_{ij}^+(t_k) \boldsymbol{\Phi}_j(t, t_k)^{\mathrm{T}} + \int_{t_k}^{t} \boldsymbol{\Phi}_i(t, \tau) \mathbf{Q}_{ij}(\tau) \boldsymbol{\Phi}_j(t, \tau)^{\mathrm{T}}.$$

To avoid this computation an upper bound of $\mathbf{P}_{ij}$ can be used instead. Since

$$\mathbf{P}_{ij} + \mathbf{P}_{ji} < \mathbf{P}_{ii} + \mathbf{P}_{jj}, \qquad i \neq j$$

a conservative bound can be formulated as follows

$$\begin{aligned}
\mathbf{P} &= \sum_{i=1}^{L} \left[w_i^2 \mathbf{P}_{ii} + \sum_{j=1}^{i-1} w_i w_j \left(\mathbf{P}_{ij} + \mathbf{P}_{ji} \right) \right] \\
&\leq \sum_{i=1}^{L} \left[w_i^2 \mathbf{P}_{ii} + \sum_{j=1}^{i-1} w_i w_j \left(\mathbf{P}_{ii} + \mathbf{P}_{jj} \right) \right] \\
&= \sum_{i=1}^{L} \left[w_i \mathbf{P}_{ii} \left(\sum_{j=1}^{L} w_j \right) \right] = \sum_{i=1}^{L} w_i \mathbf{P}_{ii}.
\end{aligned}$$

Chapter 4

Dead-Reckoning Entry Navigation

One of the most challenging and fascinating problems in spacecraft navigation is atmospheric entry, or re-entry in case of Earth. This is the most dynamically intensive phase and the poorest in available measurements. The high dynamics make direct acquisition of external measurement problematic, ionization blackout prevents ground signals from reaching the spacecraft, even communications with orbiting satellites could be problematic and subject to blackouts.

While re-entry navigation is a challenging task, entry navigation to a distant planet, such as Mars, is an even harder engineering problem. The aids that the Space Shuttle has during its descent are much more than anything achievable on Mars, at least in the medium to long term. Knowledge of the Mars atmosphere, weather, and seasons is improving, but clearly is not as accurate and as easy to predict as on Earth.

All these reasons make Mars entry navigation a very good study case that has and keeps receiving a lot of attention. In this work a detailed study of entry navigation algorithms will be perform, with particular emphasis given to how to optimally introduce the estimation of the attitude. This chapter will use the Kalman filter with uncompensated bias result derived in Section 3.2.

A typical Mars entry, descent, and landing scenario starts at entry interface (EI) which is the time when the spacecrafts switches between the orbit phase to the entry phase, the time when the spacecraft first encounters the sensible atmosphere. In orbit, the spacecraft is mainly tracked by Earth-based resources, such as the Deep Space Network. During EDL, the spacecraft is autonomous and must navigate using on-board resources. Shortly after EI, the IMUs begin providing measurements of all non-gravitational accelerations (i.e., those to due aerodynamic forces). The spacecraft makes a hypersonic/supersonic descent during which only on-board IMUs, and possibly atmospheric measurements (such as stagnation point pressure), are available. The spacecraft is contained within its aeroshell. On future missions requiring precision landing, it is during this upper atmospheric phase that the guidance will be active. For Apollo, the GN&C system modulated the aerodynamic lift direction by banking the capsule during the time in the Earth's atmosphere. Future GN&C algorithms for planetary missions may also modulate the angle of attack.

At about Mach 2+ (or an altitude of approximately 10 km), one or several parachutes are deployed and the aeroshell is jettisoned, allowing ranging instruments on-board to provide a measurement of the proximity to the ground. At this point, more advanced sensors can also map the terrain. Once the heat shield is jettisoned and the altimeter and velocimeter, now exposed to the external environment, provide measurements to the navigation algorithm, the spacecraft is on the parachute and cannot be actively guided using lift modulation. During this phase of EDL, the navigation uncertainty is significantly reduced, but guidance cannot compensate for any existing state errors. Unless there is an active parachute steering control or there is a decision made to fire the engines on the chute, this is not an active guidance phase.

In the last hundreds of meters above the surface, the parachute is jettisoned and the spacecraft lands on its own power. Once the hazard avoidance sensor is available, guidance can actively be utilized to maneuver the vehicle. By this time there is not much ability to make large excursions to hit a pinpoint landing. In the case of MER, a landing bag system was deployed which resulted in a significant bouncing at the final phase. MER did not use active guidance–it was not a precision targeted landing.

The traditional approach to EDL navigation is to employ a filter that dead-reckons the inertial measurement unit, i.e. the IMU measurements are used to propagate the states and not to update them. The advantage of this scheme is the total absence of a model for the aerodynamic forces that act during EDL. The effectiveness of this navigation system to produce a precise navigated state during entry depends almost entirely on the accuracy of the initial spacecraft state knowledge. The magnitude of the IMU errors also contributes to the precision of the estimated state. In this chapter algorithms for dead-reckoning navigation are derived, and a detailed linear covariance analysis is developed for both the continuous time measurement case (section 4.2) and the discrete time measurement case (section 4.3). The original contribution of this analysis lays in the inclusion of attitude estimation and its correlation to translational states estimation through the multiplicative approach, and in considering errors due to uncertainty on the location of the measurement unit with respect to the spacecraft center of mass.

4.1 Models

In this section, common models for the continues and discrete measurement will be presented. The dynamic model for entry navigation is the same for both classes of measurements, as is the IMU model.

4.1.1 Dynamics Modeling

The system dynamics in the inertially-fixed frame are given in the general form

$$\begin{aligned}
\dot{\mathbf{r}}^i &= \mathbf{v}^i \\
\dot{\mathbf{v}}^i &= \mathbf{g}(\mathbf{r}^i + \mathbf{T}_c^i\mathbf{d}^c) + \mathbf{T}_c^i\mathbf{a}^c \\
\dot{\overline{\mathbf{q}}}_i^c &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^c)\overline{\mathbf{q}}_i^c.
\end{aligned}$$

All superscripts indicating the inertial or case frame will be dropped since no confusion can arise because each quantity is consistently expressed in the same frame. The vector $\mathbf{r}$ is the position of the IMU in the inertial frame, $\mathbf{v}$ the velocity of the IMU in the inertial frame, $\overline{\mathbf{q}}$ the quaternion expressing the rotation from inertial to case, therefore $\mathbf{T} := \mathbf{T}_i^c = \mathbf{T}(\overline{\mathbf{q}})$. The vector $\mathbf{g}$ is the acceleration due to gravity, $\mathbf{d}$ is the unknown offset between the IMU and the center of mass which is expressed in the case frame. The true non-gravitational acceleration represented in the IMU case frame is denote by $\mathbf{a}$, and $\boldsymbol{\omega}$ is the relative angular velocity vector of the IMU case frame with respect to the inertial frame expressed in the case frame.

4.1.2 IMU Error Model

The IMU unit contains both an accelerometer package and a gyro package. Only the strapdown implementation of the IMU unit is considered here. The sensor model, whether accelerometer or gyro, continuous-time or discrete-time, has the same form and will be presented in this section.

The accelerometers and gyros produce measurements corrupted by random errors (noise and biases), and systematic errors (misalignment and scale factors). The IMU package produces a measure of the spacecraft non-gravitational accelerations and rotation rate in the IMU case frame. Let $\mathbf{y}_{true}$ be the "true" value of the measurement, and $\mathbf{y}_m$ be the measurement. The measurement error model can be formulated as

$$\mathbf{y}_m = (\mathbf{I}_{3\times3} + \boldsymbol{\Gamma})(\mathbf{I}_{3\times3} + \mathbf{S})(\mathbf{y}_{true} + \mathbf{b} + \boldsymbol{\eta}),$$

where

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} 0 & \gamma_{xz} & -\gamma_{xy} \\ -\gamma_{yz} & 0 & \gamma_{yx} \\ \gamma_{zy} & -\gamma_{zx} & 0 \end{bmatrix}, \qquad \mathbf{S} \triangleq \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}, \tag{4.1}$$

and $(\gamma_{yz}, \gamma_{zy}, \gamma_{zx}, \gamma_{xz}, \gamma_{xy}, \gamma_{yx})$ are nonorthogonality and axes misalignment errors, $\mathbf{b} \in \Re^3$ is the bias, (s_x, s_y, s_z) are scale factor errors, and $\boldsymbol{\eta} \in \Re^3$ is noise. The nonorthogonality and axes misalignment errors, scale factor errors, and bias parameters are all modelled as zero-mean and random constants. The noise $\boldsymbol{\eta}$ is modelled as a zero-mean white random process (or sequence).

Assuming that the various errors are "small," then the following first-order approximation can be made

$$(\mathbf{I}_{3\times3} + \boldsymbol{\Gamma})(\mathbf{I}_{3\times3} + \mathbf{S}) \approx \mathbf{I}_{3\times3} + \boldsymbol{\Gamma} + \mathbf{S}.$$

Defining

$$\boldsymbol{\Delta} \triangleq \boldsymbol{\Gamma} + \mathbf{S} \tag{4.2}$$

yields the measurement model

$$\mathbf{y}_m = (\mathbf{I} + \boldsymbol{\Delta})(\mathbf{y}_{true} + \mathbf{b} + \boldsymbol{\epsilon}). \tag{4.3}$$

4.2 Dead-Reckoning Continuous IMU Measurements

The underlaying assumption of this scheme is that IMU measurements of nongravitational acceleration $\mathbf{a}_m$ and of angular velocity $\boldsymbol{\omega}_m$ are continuously available.

Then the estimated vehicle state at time t is obtained by numerically integrating over the interval $[t_0, t]$ the following equations:

$$\dot{\widehat{\mathbf{r}}} = \widehat{\mathbf{v}} \tag{4-4a}$$

$$\dot{\widehat{\mathbf{v}}} = \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) + \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m \tag{4-4b}$$

$$\dot{\widehat{\overline{\mathbf{q}}}} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}_m)\widehat{\overline{\mathbf{q}}} \tag{4-4c}$$

with initial conditions

$$\widehat{\mathbf{r}}(t_0) = \widehat{\mathbf{r}}_0, \quad \widehat{\mathbf{v}}(t_0) = \widehat{\mathbf{v}}_0, \quad \widehat{\overline{\mathbf{q}}}(t_0) = \widehat{\overline{\mathbf{q}}}_0.$$

The estimate of the rotation matrix is

$$\widehat{\mathbf{T}} = \mathbf{T}(\,\widehat{\overline{\mathbf{q}}}\,).$$

Integration of the navigation equations given in Eqs. (4-4a)–(4-4c) yields the navigated spacecraft position, velocity, and attitude. Since dead-reckoning is essentially an *open-loop* estimation process, the accuracy of the navigated state depends strongly on knowledge of the initial spacecraft state. Also, any measurement errors present in $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ will corrupt the navigation solution.

The estimation errors associated with the dead-reckoning navigation solution is comprised of the attitude estimation errors and the position and velocity estimation errors. As can be seen in Eqs. (4-4a)–(4-4c), integration of the position and velocity equations requires a transformation of the IMU accelerations from the case frame to the inertial frame, which depends, in turn, on the spacecraft attitude estimate. Therefore, any estimation error in the attitude estimate naturally couples into the position and velocity navigation. The attitude estimation does not rely on the position and velocity estimation, hence can be addressed independently.

4.2.1 Attitude estimation errors

Define the multiplicative attitude error, $\delta\overline{\mathbf{q}}$, as

$$\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1} .$$

Computing $\delta\dot{\overline{\mathbf{q}}}$ yields

$$\delta\dot{\overline{\mathbf{q}}} = \dot{\overline{\mathbf{q}}} \otimes \widehat{\overline{\mathbf{q}}}^{-1} + \overline{\mathbf{q}} \otimes \dot{\widehat{\overline{\mathbf{q}}}}^{-1} = \frac{1}{2}\overline{\boldsymbol{\omega}} \otimes \delta\overline{\mathbf{q}} - \frac{1}{2}\delta\overline{\mathbf{q}} \otimes \overline{\boldsymbol{\omega}}_m, \tag{4.5}$$

where the pure quaternion $\overline{\boldsymbol{\omega}}_m$ is defined as

$$\overline{\boldsymbol{\omega}}_m \triangleq \begin{bmatrix} \boldsymbol{\omega}_m \\ 0 \end{bmatrix} .$$

Assuming small angles, the vector part of the quaternion fully represents the attitude

$$\delta\overline{\mathbf{q}} \simeq \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix} ,$$

from Eq. (4.5) it follows that to first-order we have

$$\delta\dot{\mathbf{q}} = -\boldsymbol{\omega}_m \times \delta\mathbf{q} + \frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}_m).$$

It then follows from Eq. (4.3) that

$$\delta\dot{\mathbf{q}} = -\boldsymbol{\omega}_m \times \delta\mathbf{q} + \frac{1}{2}\left[(\mathbf{I} + \boldsymbol{\Delta}_g)^{-1}(\boldsymbol{\omega}_m - \mathbf{b}_g - \boldsymbol{\eta}_g) - \boldsymbol{\omega}_m\right] .$$

To first -order in $\boldsymbol{\Delta}_g$, it follows that

$$(\mathbf{I} + \boldsymbol{\Delta}_g)^{-1} \simeq \mathbf{I} - \boldsymbol{\Delta}_g.$$

Therefore, to first-order we have

$$\delta\dot{\mathbf{q}} = -[\boldsymbol{\omega}_m\times]\delta\mathbf{q} - \frac{1}{2}\left(\boldsymbol{\Delta}_g\boldsymbol{\omega}_m + \mathbf{b}_g + \boldsymbol{\eta}_g\right) \ .$$

With the given definition of $\mathbf{S}_g$ in Eq. (4.1), we can write

$$\mathbf{S}_g\boldsymbol{\omega}_m = \mathbf{D}(\boldsymbol{\omega}_m)\mathbf{s}_g,$$

where

$$\mathbf{s}_g \triangleq \begin{bmatrix} s_{gx} \\ s_{gy} \\ s_{gz} \end{bmatrix} \quad \text{and} \quad \mathbf{D}(\boldsymbol{\omega}_m) = \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix} \ . \tag{4.6}$$

Similarly

$$\boldsymbol{\Gamma}_g\boldsymbol{\omega}_m = \mathbf{N}(\boldsymbol{\omega}_m)\boldsymbol{\gamma}_g \ , \tag{4.7}$$

where

$$\boldsymbol{\gamma}_g \triangleq \begin{bmatrix} \gamma_{g_{xy}} \\ \gamma_{g_{xz}} \\ \gamma_{g_{yx}} \\ \gamma_{g_{yz}} \\ \gamma_{g_{zx}} \\ \gamma_{g_{zy}} \end{bmatrix} \quad \text{and} \quad \mathbf{N}(\boldsymbol{\omega}_m) \triangleq \begin{bmatrix} -\omega_{m_z} & \omega_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{m_z} & -\omega_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_{m_y} & \omega_{m_x} \end{bmatrix} \ .$$

Therefore, it follows that

$$\delta\dot{\mathbf{q}} = -[\boldsymbol{\omega}_m\times]\delta\mathbf{q} - \frac{1}{2}\left[\mathbf{D}(\boldsymbol{\omega}_m)\mathbf{s}_g + \mathbf{N}(\boldsymbol{\omega}_m)\boldsymbol{\gamma}_g + \mathbf{b}_g + \boldsymbol{\eta}_g\right].$$

For small angles, the rotation vector $\boldsymbol{\theta}$ is approximately twice the vector part of the quaternion, therefore $\mathbf{e}_\theta \simeq 2\delta\mathbf{q}$, and

$$\dot{\mathbf{e}}_\theta = -[\boldsymbol{\omega}_m \times]\mathbf{e}_\theta - \mathbf{D}(\boldsymbol{\omega}_m)\mathbf{s}_g - \mathbf{N}(\boldsymbol{\omega}_m)\boldsymbol{\gamma}_g - \mathbf{b}_g - \boldsymbol{\eta}_g. \tag{4.8}$$

4.2.2 Position and velocity estimation errors

The position and velocity estimation error are defined to be

$$\mathbf{e}_r \triangleq \mathbf{r} - \widehat{\mathbf{r}} \qquad \text{and} \qquad \mathbf{e}_v \triangleq \mathbf{v} - \widehat{\mathbf{v}}.$$

Computing the time-derivative of $\mathbf{e}_r$ and $\mathbf{e}_v$ yields, respectively,

$$\dot{\mathbf{e}}_r = \mathbf{e}_v, \tag{4.9}$$

and

$$\dot{\mathbf{e}}_v = \mathbf{g}(\mathbf{r} + \mathbf{T}^{\mathrm{T}}\mathbf{d}) - \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) + \mathbf{T}^{\mathrm{T}}\mathbf{a} - \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m, \tag{4.10}$$

where $\widehat{\mathbf{d}}$ is the estimate of the distance between the IMU and the center of mass. Expanding gravity utilizing a Taylor series, and neglecting higher order terms, it follows that

$$\mathbf{g}(\mathbf{r} + \mathbf{T}^{\mathrm{T}}\mathbf{d}) - \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) \simeq \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) \left(\mathbf{e}_r + \mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}\right),$$

where

$$\mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) \triangleq \left.\frac{\partial \mathbf{g}}{\partial \mathbf{r}}\right|_{\mathbf{r}=\hat{\mathbf{r}}+\hat{\mathbf{T}}^{\mathrm{T}}\hat{\mathbf{d}}}.$$

Since the quaternion error is defined as $\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{\,-1}$ and attitude matrices are multiplied in the same order as quaternions, then $\delta\mathbf{T} = \mathbf{T}\,\widehat{\mathbf{T}}^{\mathrm{T}}$, therefore

$$\mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}} = \widehat{\mathbf{T}}^{\mathrm{T}}\,\delta\mathbf{T}^{\mathrm{T}}(\widehat{\mathbf{d}} + \mathbf{e}_d) - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}},$$

where $\mathbf{e}_d \triangleq \mathbf{d} - \widehat{\mathbf{d}}$. Since

$$\delta\mathbf{T}^{\mathrm{T}} \simeq \mathbf{I}_{3\times 3} + [\mathbf{e}_\theta \times], \tag{4.11}$$

it follows that

$$\mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}} \simeq \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{e}_\theta \times]\widehat{\mathbf{d}} + \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{e}_d = -\widehat{\mathbf{T}}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\mathbf{e}_\theta + \widehat{\mathbf{T}}^{\mathrm{T}}\,\mathbf{e}_d. \tag{4.12}$$

Similarly,

$$\mathbf{T}^{\mathrm{T}}\mathbf{a} - \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m = \widehat{\mathbf{T}}^{\mathrm{T}}\,\delta\mathbf{T}^{\mathrm{T}}\mathbf{a} - \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m,$$

and using Eq. (4.11), we obtain

$$\mathbf{T}^{\mathrm{T}}\mathbf{a} - \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m \simeq \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{e}_\theta \times]\mathbf{a}_m + \widehat{\mathbf{T}}^{\mathrm{T}}\left(\mathbf{a} - \mathbf{a}_m\right). \tag{4.13}$$

Rearranging the terms in the IMU model given in Eq. (4.3) yields

$$\mathbf{a} = (\mathbf{I} + \boldsymbol{\Delta}_a)^{-1}\mathbf{a}_m - \mathbf{b}_a - \boldsymbol{\eta}_a,$$

after some manipulation and using the fact that $(\mathbf{I} + \boldsymbol{\Delta}_a)^{-1} \simeq \mathbf{I} - \boldsymbol{\Delta}_a$ for “small” $\boldsymbol{\Delta}_a$, it follows that

$$\mathbf{a} - \mathbf{a}_m = -\boldsymbol{\Delta}_a\mathbf{a}_m - \mathbf{b}_a - \boldsymbol{\eta}_a. \tag{4.14}$$

Substituting Eqs. (4.12)–(4.14) into Eq. (4.10) and neglecting higher-order terms yields

$$\dot{\mathbf{e}}_v = \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})\left(\mathbf{e}_r - \widehat{\mathbf{T}}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\,\mathbf{e}_\theta + \widehat{\mathbf{T}}^{\mathrm{T}}\,\mathbf{e}_d\right) - \widehat{\mathbf{T}}^{\mathrm{T}}(\boldsymbol{\Delta}_a\mathbf{a}_m + \mathbf{b}_a + \boldsymbol{\eta}_a) - \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{a}_m \times]\mathbf{e}_\theta. \tag{4.15}$$

Keep in mind that in Eq. (4.15), the matrix $\boldsymbol{\Delta}_a$ is comprised of random constants, $\mathbf{b}_a$ is a random constant vector, $\mathbf{e}_d$ is a random constant under the assumption of ballistic entry (no fuel is expended), $\boldsymbol{\eta}_a$ is a random process, and $\mathbf{e}_\theta$ is the attitude estimation error that contributes directly to the uncertainty in the position (through integration) and velocity estimation errors.

Consider the term $\boldsymbol{\Delta}_a \mathbf{a}_m$ more closely. From the definition of $\boldsymbol{\Delta}_a$ given in Eq. (4.2), we have

$$\boldsymbol{\Delta}_a \mathbf{a}_m = \left(\boldsymbol{\Gamma}_a + \mathbf{S}_a\right) \mathbf{a}_m.$$

With the definitions of $\boldsymbol{\Gamma}_a$ and $\mathbf{S}_a$ given in Eq. (4.1), $\boldsymbol{\Delta}_a \mathbf{a}_m$ can also be written as

$$\boldsymbol{\Delta}_a \mathbf{a}_m = \mathbf{D}(\mathbf{a}_m)\mathbf{s}_a + \mathbf{N}(\mathbf{a}_m)\boldsymbol{\gamma}_a,$$

where definitions of $\mathbf{D}(\cdot)$ and $\mathbf{N}(\cdot)$ are equivalent to those of Eqs. (4.6) and (4.7) are used.

Collecting the position, velocity, and attitude estimation error equations from Eqs. (4.8)–(4.10), and writing them in matrix form yields the stochastic linear matrix differential equation

$$\dot{\mathbf{e}} = \mathbf{F}\mathbf{e} + \mathbf{H}_1\mathbf{b} + \mathbf{H}_2\boldsymbol{\eta}, \tag{4.16}$$

where

$$\mathbf{e} \triangleq \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_v \\ \mathbf{e}_\theta \end{bmatrix} \in \Re^9, \qquad \mathbf{b} \triangleq \begin{bmatrix} \mathbf{s}_a \\ \boldsymbol{\gamma}_a \\ \mathbf{b}_a \\ \mathbf{s}_g \\ \boldsymbol{\gamma}_g \\ \mathbf{b}_g \\ \mathbf{e}_d \end{bmatrix} \in \Re^{27}, \qquad \boldsymbol{\eta} \triangleq \begin{bmatrix} \boldsymbol{\eta}_a \\ \boldsymbol{\eta}_g \end{bmatrix} \in \Re^6.$$

The error state matrix $\mathbf{F} \in \Re^{9\times 9}$ is

$$\mathbf{F} = \begin{bmatrix} \mathbf{O}_{3\times 3} & \mathbf{I}_{3\times 3} & \mathbf{O}_{3\times 3} \\ \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) & \mathbf{O}_{3\times 3} & -\mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})\widehat{\mathbf{T}}^{\mathrm{T}}[\widehat{\mathbf{d}}\times] - \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{a}_m\times] \\ \mathbf{O}_{3\times 3} & \mathbf{O}_{3\times 3} & -[\boldsymbol{\omega}_m\times] \end{bmatrix},$$

and the input mapping matrices $\mathbf{H}_1 \in \Re^{9\times 27}$ and $\mathbf{H}_2 \in \Re^{9\times 6}$ are

$$\mathbf{H}_1 \triangleq \begin{bmatrix} \mathbf{O}_{3\times 12} & \mathbf{O}_{3\times 12} & \mathbf{O}_{3\times 3} \\ -\widehat{\mathbf{T}}^{\mathrm{T}}\big[\mathbf{D}(\mathbf{a}_m)\ \ \mathbf{N}(\mathbf{a}_m)\ \ \mathbf{I}_{3\times 3}\big] & \mathbf{O}_{3\times 12} & \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})\widehat{\mathbf{T}}^{\mathrm{T}} \\ \mathbf{O}_{3\times 12} & -\big[\mathbf{D}(\boldsymbol{\omega}_m)\ \ \mathbf{N}(\boldsymbol{\omega}_m)\ \ \mathbf{I}_{3\times 3}\big] & \mathbf{O}_{3\times 3} \end{bmatrix},$$

and

$$\mathbf{H}_2 \triangleq \begin{bmatrix} \mathbf{O}_{3\times 3} & \mathbf{O}_{3\times 3} \\ -\widehat{\mathbf{T}}^{\mathrm{T}} & \mathbf{O}_{3\times 3} \\ \mathbf{O}_{3\times 3} & -\mathbf{I}_{3\times 3} \end{bmatrix}.$$

The components of $\mathbf{b}$ in Eq. (4.16) are the various random constant errors associated with the IMU and c.g. location, where it assumed that

$$\mathrm{E}\{\mathbf{b}\} = \mathbf{0},$$

and $\mathbf{B} \in \Re^{27\times 27}$ is

$$\mathbf{B} \triangleq \mathrm{E}\left\{\mathbf{b}\,\mathbf{b}^{\mathrm{T}}\right\}.$$

The components of $\boldsymbol{\eta}(t)$ in Eq. (4.16) are the random components of the IMU errors, where it is assumed that

$$\mathrm{E}\{\boldsymbol{\eta}(t)\} = \mathbf{0} \qquad \text{and} \qquad \mathrm{E}\left\{\boldsymbol{\eta}(t)\boldsymbol{\eta}(\tau)^{\mathrm{T}}\right\} = \mathbf{V}(t)\delta(t-\tau).$$

4.2.3 Estimation error covariance

Define the state transition matrix $\mathbf{\Phi}(t, t_0) \in \Re^{9\times 9}$ associated with $\mathbf{F}$ as the solution to the matrix differential equation

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{F}\big(\widehat{\mathbf{x}}(t)\big)\,\mathbf{\Phi}(t, t_0), \tag{4.17}$$

where

$$\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{9\times 9}.$$

The solution to Eq. (4.16) is

$$\mathbf{e}(t) = \mathbf{\Phi}(t, t_0)\mathbf{e}(t_0) + \left[\int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_1(\tau)d\tau\right]\mathbf{b} + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_2(\tau)\boldsymbol{\eta}(\tau)d\tau, \tag{4.18}$$

and with the estimation error covariance, $\mathbf{P}(t)$, defined as

$$\mathbf{P}(t) = \mathrm{E}\left\{\mathbf{e}(t)\mathbf{e}^{\mathrm{T}}(t)\right\}, \tag{4.19}$$

the matrix $\mathbf{P}(t)$ can be compute using Eqs. (4.17)–(4.19) yielding

$$\begin{aligned}\mathbf{P}(t) = \mathbf{\Phi}(t, t_0)\mathbf{P}(t_0)\mathbf{\Phi}^{\mathrm{T}}(t, t_0) &+ \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_2(\tau)\mathbf{V}(\tau)\mathbf{H}_2^{\mathrm{T}}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t, \tau)d\tau \\ &+ \left[\int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_1(\tau)d\tau\right]\mathbf{B}\left[\int_{t_0}^{t} \mathbf{H}_1^{\mathrm{T}}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t, \tau)d\tau\right].\end{aligned}$$

As the first step towards a more implementable form of the error covariance, define

$$\begin{aligned}\mathbf{V}_t &\triangleq \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_2(\tau)\mathbf{V}(\tau)\mathbf{H}_2^{\mathrm{T}}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t, \tau)d\tau \\ \mathbf{B}_t &\triangleq \left[\int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{H}_1(\tau)d\tau\right].\end{aligned}$$

Then, taking the time-derivative of $\mathbf{V}_t$ and $\mathbf{B}_t$ yields

$$\dot{\mathbf{V}}_t = \mathbf{F}(t)\mathbf{V}_t + \mathbf{V}_t\mathbf{F}^{\mathrm{T}}(t) + \mathbf{H}_2(t)\mathbf{V}(t)\mathbf{H}_2^{\mathrm{T}}(t), \tag{4.20}$$

and

$$\dot{\mathbf{B}}_t = \mathbf{F}(t)\mathbf{B}_t + \mathbf{H}_1(t), \tag{4.21}$$

respectively. The appropriate initial conditions are $\mathbf{V}_t(t_0) = \mathbf{O}$ and $\mathbf{B}_t(t_0) = \mathbf{O}$. Once the values of $\mathbf{\Phi}(t, t_0)$, $\mathbf{V}_t$ and $\mathbf{B}_t$ have been computed via integration of Eqs. (4.17), (4.20), and (4.21) from t_0 to t, the error covariance $\mathbf{P}(t_0)$ at time t_0 is mapped forward to time t via

$$\mathbf{P}(t) = \mathbf{\Phi}(t, t_0)\mathbf{P}(t_0)\mathbf{\Phi}^{\mathrm{T}}(t, t_0) + \mathbf{V}_t + \mathbf{B}_t\mathbf{B}\mathbf{B}_t^{\mathrm{T}}.$$

4.2.4 Dead Reckoning Navigation

Suppose that the time history of the IMU observations, that is, $\mathbf{a}_m(t)$ and $\boldsymbol{\omega}_m(t)$ are continuously available. Then, dead reckoning navigation, including computing the associated state estimation error covariance, is the process of integrating over the interval $[t_0, t]$ the following equations:

$$\begin{aligned}
\dot{\widehat{\mathbf{r}}} &= \widehat{\mathbf{v}} \\
\dot{\widehat{\mathbf{v}}} &= \widehat{\mathbf{g}} + \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{a}_m \\
\dot{\widehat{\overline{\mathbf{q}}}} &= \frac{1}{2}\mathbf{\Omega}\,\widehat{\overline{\mathbf{q}}} \\
\dot{\mathbf{\Phi}} &= \mathbf{F}\mathbf{\Phi} \\
\dot{\mathbf{B}}_t &= \mathbf{F}\mathbf{B}_t + \mathbf{H}_1 \\
\dot{\mathbf{V}}_t &= \mathbf{F}\mathbf{V}_t + \mathbf{V}_t\mathbf{F}^{\mathrm{T}} + \mathbf{H}_2\mathbf{V}\mathbf{H}_2^{\mathrm{T}},
\end{aligned}$$

and the estimation error covariance at time t_0 is mapped forward to time t via

$$\mathbf{P}(t) = \mathbf{\Phi}\mathbf{P}_0\mathbf{\Phi}^{\mathrm{T}} + \mathbf{V}_t + \mathbf{B}_t\mathbf{B}\mathbf{B}_t^{\mathrm{T}},$$

where $\widehat{\mathbf{g}} \triangleq \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})$ is the modelled gravity, $\widehat{\overline{\mathbf{q}}} = \begin{bmatrix}\widehat{\mathbf{q}}^{\mathrm{T}} & \widehat{q}\end{bmatrix}^{\mathrm{T}}$, $\widehat{\mathbf{T}} = \mathbf{T}(\widehat{\overline{\mathbf{q}}})$, and

$$\mathbf{\Omega} \triangleq \mathbf{\Omega}(\boldsymbol{\omega}_m) = \begin{bmatrix} -[\boldsymbol{\omega}_m\times] & \boldsymbol{\omega}_m \\ -\boldsymbol{\omega}_m^{\mathrm{T}} & 0 \end{bmatrix},$$

$$\widehat{\mathbf{T}}^{\mathrm{T}} \triangleq \mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}} = \mathbf{I}_{3\times3} + 2\widehat{q}[\widehat{\mathbf{q}}\times] + 2[\widehat{\mathbf{q}}\times]^2,$$

$$\mathbf{F} \triangleq \begin{bmatrix} \mathbf{O}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{O}_{3\times3} \\ \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) & \mathbf{O}_{3\times3} & -\mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})\widehat{\mathbf{T}}^{\mathrm{T}}[\widehat{\mathbf{d}}\times] - \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{a}_m\times] \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & -[\boldsymbol{\omega}_m\times] \end{bmatrix},$$

$$\mathbf{H}_1 \triangleq \begin{bmatrix} \mathbf{O}_{3\times12} & \mathbf{O}_{3\times12} & \mathbf{O}_{3\times3} \\ -\widehat{\mathbf{T}}^{\mathrm{T}}\big[\mathbf{D}(\mathbf{a}_m)\ \ \mathbf{N}(\mathbf{a}_m)\ \ \mathbf{I}_{3\times3}\big] & \mathbf{O}_{3\times12} & \mathbf{G}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}})\widehat{\mathbf{T}}^{\mathrm{T}} \\ \mathbf{O}_{3\times12} & -\big[\mathbf{D}(\boldsymbol{\omega}_m)\ \ \mathbf{N}(\boldsymbol{\omega}_m)\ \ \mathbf{I}_{3\times3}\big] & \mathbf{O}_{3\times3} \end{bmatrix},$$

$$\mathbf{H}_2 \triangleq \begin{bmatrix} \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} \\ -\widehat{\mathbf{T}}^{\mathrm{T}} & \mathbf{O}_{3\times3} \\ \mathbf{O}_{3\times3} & -\mathbf{I}_{3\times3} \end{bmatrix},$$

$$\mathbf{D}(\mathbf{a}_m) = \begin{bmatrix} a_{m_x} & 0 & 0 \\ 0 & a_{m_y} & 0 \\ 0 & 0 & a_{m_z} \end{bmatrix}, \qquad \mathbf{D}(\boldsymbol{\omega}_m) = \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix},$$

$$\mathbf{N}(\mathbf{a}_m) = \begin{bmatrix} -a_{m_z} & a_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{m_z} & -a_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{m_y} & a_{m_x} \end{bmatrix},$$

$$\mathbf{N}(\boldsymbol{\omega}_m) = \begin{bmatrix} -\omega_{m_z} & \omega_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{m_z} & -\omega_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_{m_y} & \omega_{m_x} \end{bmatrix},$$

with initial conditions

$$\widehat{\mathbf{r}}(t_0) = \widehat{\mathbf{r}}_0, \quad \widehat{\mathbf{v}}(t_0) = \widehat{\mathbf{v}}_0, \quad \widehat{\bar{\mathbf{q}}}(t_0) = \widehat{\bar{\mathbf{q}}}_0, \quad \boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}, \quad \mathbf{P}(t_0) = \mathbf{P}_0, \quad \mathbf{V}_t(t_0) = \mathbf{0},$$
$$\mathbf{B}_t(t_0) = \mathbf{0}.$$

The sensor models are assumed known and represented by the matrices $\mathbf{V}(t)$ and $\mathbf{B}$.

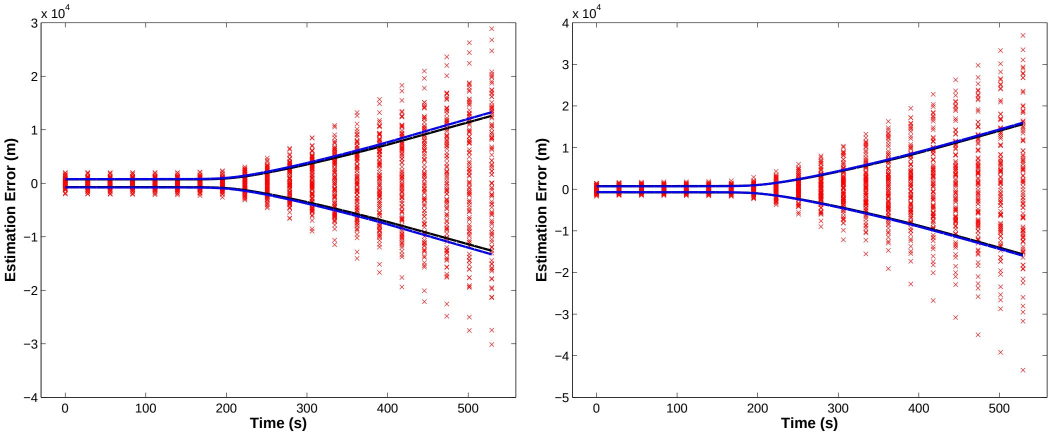
4.2.5 Simulation Results

In the section, the sampled estimation error covariance obtained through Monte Carlo analysis is compared with the linear covariance formulation. Verification of the formulation is made corrupting the "true" measurements and the initial estimate and through judicious use of Monte Carlo. The true trajectory is obtained from NASA's high fidelity SORT simulation. The trajectory follows that of NASA JPL 2005 Mars mission (the mission was eventually cancelled). The true measurements are obtained directly from the true trajectory and corrupted with noise whose characteristic are shown in Table 4.1. In each of the 100 Monte Carlo runs, the random errors are generated from a zero mean gaussian distribution with standard deviation shown in Table 4.1. The 100 trajectories of the estimation error are then used to calculate the sample covariance, which represent the true error characteristics of the estimated state. The filter covariance represents the estimate of the error characteristics. By matching the filter covariance with the sample covariance, we show that the filter accurately represents the estimation error characteristics, hence the filter is well-tuned.

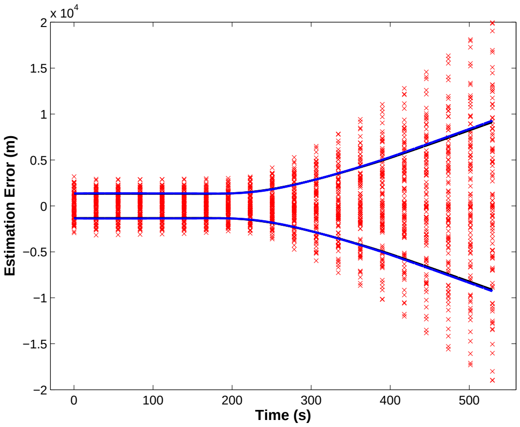
Accelerometer Noise	$\boldsymbol{\eta}_a$	$10\ [\mu g\sqrt{s}]$
Accelerometer Bias	$\mathbf{b}_a$	$0.1\ [mg]$
Accelerometer Scale Factor	$\mathbf{s}_a$	$175\ [ppm]$
Accelerometer Misalignment	$\boldsymbol{\gamma}_a$	$5\ [arcsec]$
Gyro Noise	$\boldsymbol{\eta}_g$	$0.01\ [deg/\sqrt{hr}]$
Gyro Bias	$\mathbf{b}_g$	$0.05\ [deg/hr]$
Gyro Scale Factor	$\mathbf{s}_g$	$5\ [ppm]$
Gyro Misalignment	$\boldsymbol{\gamma}_g$	$5\ [arcsec]$

Table 4.1: Continuous-time IMU errors

Figures 4.1–4.3 show samples of error evolution of the 100 runs (denoted by red x's), the sample covariance (blue lines), and the linear covariance formulation evaluated (black lines). Figures 4.1–4.2 contain the inertial position and velocity errors in the x, y, and z axis respectively. Figure 4.3 contains the three components of the attitude error from estimated body frame to true body frame, the attitude error is represented as a rotation vector.

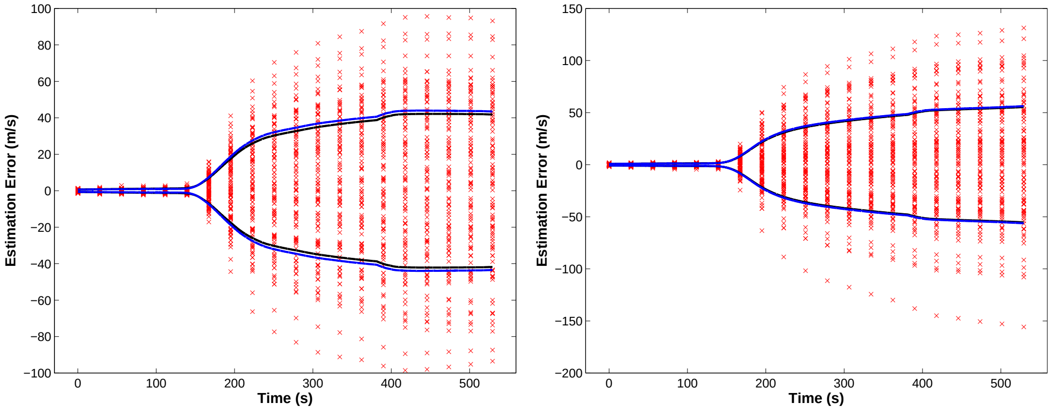


(a) Inertial x axis (b) Inertial y axis

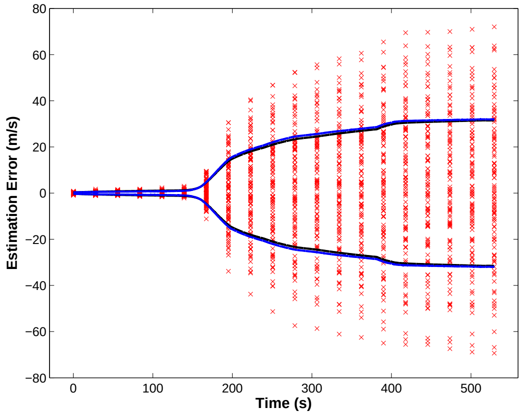


(c) Inertial z axis

Figure 4.1: Position estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.

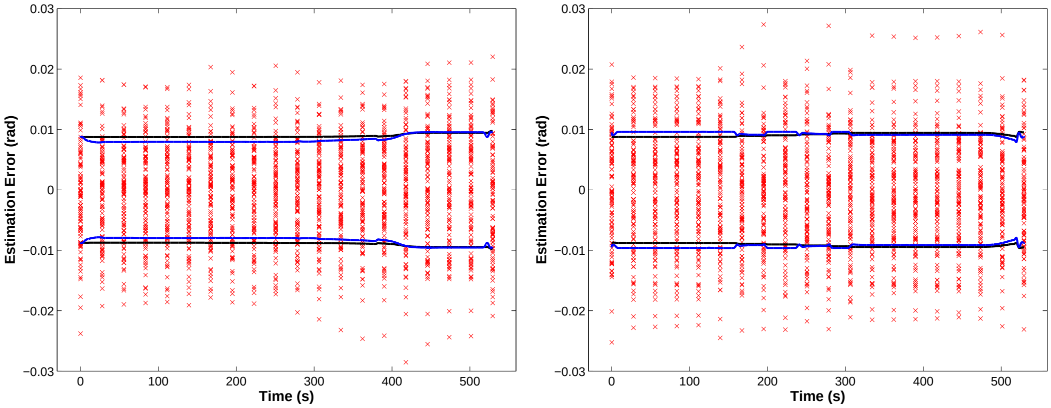


(a) Inertial x axis (b) Inertial y axis

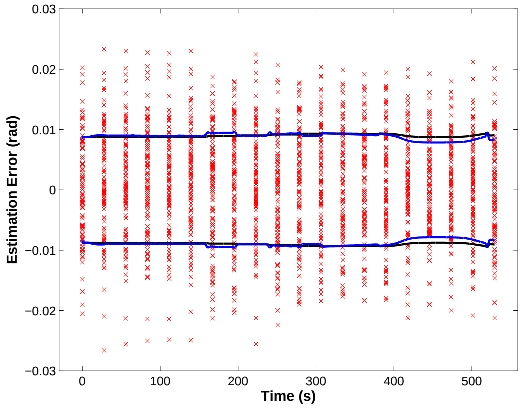


(c) Inertial z axis

Figure 4.2: Velocity estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.



(a) First component (b) Second component



(c) Third component

Figure 4.3: Attitude estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.

4.2.6 Generating the Accelerometer and Gyro Noises

Since the accelerometer and the gyro are dead-reckoned, white noise enters the numerical integration of the estimated state in the simulation. In performing the numerical integration, a white sequence has to be generated to replace the white noise. In this work the white sequence covariance was generated as follows.

Consider the following stochastic differential equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}, \tag{4.22}$$

where $\boldsymbol{\eta}$ is a zero-mean, white noise process, with spectral density $\mathbf{Q}_{spec}$. Eq. (4.22) needs to be simulated, therefore it will be numerically integrated. For simplicity we will consider a single step integrator and small Δt. Each step is given by

$$\mathbf{x}_{k+1} \simeq \mathbf{x}_k + \mathbf{A}_k \mathbf{x}_k \Delta t + \boldsymbol{\eta}_k \Delta t, \tag{4.23}$$

where $\boldsymbol{\eta}_k$ needs to be generated, for example from a gaussian distribution denoted by

$$\boldsymbol{\eta}_k \sim N(\mathbf{0}, \mathbf{R}_k).$$

For small Δt, we make the assumption that $\boldsymbol{\Phi}_k \simeq \mathbf{I} + \mathbf{A}_k \Delta t$. Therefore, the error covariance associated with Eq. (4.23) is

$$\mathbf{P}_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^{\mathrm{T}} + \mathbf{Q}_{cov}, \qquad \mathbf{Q}_{cov} = \mathbf{R}_k \Delta t^2.$$

It is well-known that

$$\dot{\mathbf{Q}}_{cov} = \mathbf{A}\mathbf{Q}_{cov} + \mathbf{Q}_{cov}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_{spec},$$

with $\mathbf{Q}_{cov}(0) = \mathbf{O}$. To first order in Δt, we obtain

$$\mathbf{Q}_{spec} = 2\mathbf{R}_k \Delta t.$$

Hence the zero-mean sequence in Eq. (4.23) is be generated with covariance

$$\mathbf{R}_k \simeq \frac{\mathbf{Q}_{spec}}{2\Delta t}.$$

4.2.7 Alternative Approach

The equations summarized in § 4.2.4 are exactly those used in [77], and equivalent to those in [78] that employed the additive approach. However, an alternative approach is possible from the Kalman filter with uncompensated bias formulation of Section 3.2.2. The equations of Table 3.2 can be used taking into account that no measurement is available, i.e. $\mathbf{K} = \mathbf{O}$. Therefore the equations to be integrate become

$$\begin{aligned}
&\dot{\widehat{\mathbf{x}}} = \mathbf{F}(t)\,\widehat{\mathbf{x}}; \qquad \widehat{\mathbf{x}}(t_0) = \widehat{\mathbf{x}}_0 \\
&\dot{\mathbf{L}} = \mathbf{F}(t)\,\mathbf{L} + \mathbf{H}_1(t); \qquad \mathbf{L}(t_0) = \mathbf{O} \\
&\dot{\mathbf{P}} = \mathbf{F}(t)\,\mathbf{P} + \mathbf{P}\,\mathbf{F}(t) + \mathbf{H}_2(t)\,\mathbf{V}(t)\,\mathbf{H}_2(t)^{\mathrm{T}} + \mathbf{L}(t)\,\mathbf{B}\,\mathbf{H}_1(t)^{\mathrm{T}} + \mathbf{H}_1(t)\,\mathbf{B}\,\mathbf{L}(t); \\
&\hspace{20em} \mathbf{P}(t_0) = \mathbf{P}_0,
\end{aligned}$$

where $\mathbf{F}(t$, $\mathbf{B}$, $\mathbf{V}(t)$, $\mathbf{H}_1(t)$, and $\mathbf{H}_2(t)$ were defined in the previous sections. This alternative approach is more convenient from a computational standpoint because it does not required the integration of as many equations, but is absolutely equivalent from a mathematical standpoint.

4.3 Dead-Reckoning Discrete IMU Measurements

The underlying assumption of this scheme is that discrete IMU measurements of $\Delta\mathbf{v}$'s and $\Delta\boldsymbol{\theta}$'s are available. The measurement model is given by Eq. (4.3) with

$$\mathbf{y}_{true,k} = \Delta\mathbf{v}_k = \int_{t_{k-1}}^{t_k} \mathbf{a}\; dt, \qquad \text{and} \qquad \mathbf{y}_{true,k} = \Delta\boldsymbol{\theta}_k = \int_{t_{k-1}}^{t_k} \boldsymbol{\omega}\; dt,$$

for the accelerometer and gyro, respectively. Once the acceleration and the angular velocity have been integrated by the sensor, their point-wise in time values are not retrievable but only the average values are available. The standard procedure is therefore to assume $\mathbf{a}$ and $\boldsymbol{\omega}$ constant in the time step.

$$\mathbf{a}_{m,k} \triangleq \frac{\Delta\mathbf{v}_{m,k}}{\Delta t}, \qquad \boldsymbol{\omega}_{m,k} \triangleq \frac{\Delta\boldsymbol{\theta}_{m,k}}{\Delta t}, \qquad \forall t \in [t_{k-1}, t_k].$$

The quaternion expressing the rotation from the inertial frame to the case frame at time t_{k-1} is denoted by $\overline{\mathbf{q}}_{k-1}$. Define the quaternion $\Delta\overline{\mathbf{q}}$ expressing the rotation during one time step as

$$\Delta\widehat{\overline{\mathbf{q}}}(t) \triangleq \widehat{\overline{\mathbf{q}}}(t) \otimes \widehat{\overline{\mathbf{q}}}\,_{k-1}^{-1} \qquad t \in [t_{k-1},\, t_k].$$

Its evolution is given by

$$\Delta\dot{\widehat{\overline{\mathbf{q}}}}(t) = \dot{\widehat{\overline{\mathbf{q}}}}(t) \otimes \widehat{\overline{\mathbf{q}}}\,_{k-1}^{-1} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{m,k})\Delta\overline{\mathbf{q}}(t), \qquad t \in [t_{k-1},\, t_k], \qquad \Delta\overline{\mathbf{q}}(t_{k-1}) = \bar{\mathbf{i}}_q. \tag{4.24}$$

Let $\boldsymbol{\theta}$ be the rotation vector parametrization of $\Delta\overline{\mathbf{q}}$. Using this parametrization and assuming small $\boldsymbol{\theta}$ (i.e. small time step), Eq. (4.24) reduces to

$$\dot{\widehat{\boldsymbol{\theta}}}(t) = \boldsymbol{\omega}_{m,k} - \boldsymbol{\omega}_{m,k} \times \widehat{\boldsymbol{\theta}}(t), \qquad t \in [t_{k-1},\, t_k], \qquad \widehat{\boldsymbol{\theta}}(t_{k-1}) = \mathbf{0}. \tag{4.25}$$

The solution of Eq. (4.25) is

$$\widehat{\boldsymbol{\theta}}(t) = \boldsymbol{\omega}_{m,k}(t - t_{k-1}).$$

Therefore, the discrete quaternion update is given by

$$\widehat{\overline{\mathbf{q}}}_k = \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k}) \otimes \widehat{\overline{\mathbf{q}}}_{k-1},$$

where

$$\overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k}) = \Delta\widehat{\overline{\mathbf{q}}}(t_k) = \begin{bmatrix} \sin\left(\frac{1}{2}\|\Delta\boldsymbol{\theta}_{m,k}\|\right) \;\; \Delta\boldsymbol{\theta}_{m,k}/\|\Delta\boldsymbol{\theta}_{m,k}\| \\ \cos(\frac{1}{2}\|\Delta\boldsymbol{\theta}_{m,k}\|) \end{bmatrix}.$$

The estimate of the velocity evolves as

$$\dot{\widehat{\mathbf{v}}}(t) = \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}(t)^{\mathrm{T}}\widehat{\mathbf{d}}) + \widehat{\mathbf{T}}(t)^{\mathrm{T}}\mathbf{a}_{m,k}. \qquad t \in [t_{k-1},\, t_k], \tag{4.26}$$

The estimate of the rotation matrix is

$$\widehat{\mathbf{T}}(t)^{\mathrm{T}} = \mathbf{T}\big(\,\widehat{\overline{\mathbf{q}}}(t)\,\big)^{\mathrm{T}} = \mathbf{T}(\,\widehat{\overline{\mathbf{q}}}_{k-1})^{\mathrm{T}}\;\mathbf{T}(\Delta\widehat{\overline{\mathbf{q}}}(t)\,)^{\mathrm{T}}, \qquad t \in [t_{k-1}, t_k],$$

and to first order we obtain

$$\Delta\widehat{\mathbf{T}}(t) \triangleq \mathbf{T}(\Delta\widehat{\overline{\mathbf{q}}}(t)\,) \simeq \mathbf{I}_{3\times 3} - [\widehat{\boldsymbol{\theta}}(t)\times], \qquad t \in [t_{k-1}, t_k].$$

Using Taylor series, we expand the gravity term

$$\begin{aligned} \widehat{\mathbf{g}}(t) &\triangleq \mathbf{g}(\widehat{\mathbf{r}}(t) + \widehat{\mathbf{T}}(t)^{\mathrm{T}}\widehat{\mathbf{d}}) \simeq \mathbf{g}\big(\widehat{\mathbf{r}}_{k-1} + \widehat{\mathbf{T}}(t)^{\mathrm{T}}\widehat{\mathbf{d}}\big) \\ &\simeq \mathbf{g}\big(\widehat{\mathbf{r}}_{k-1} + \widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\widehat{\mathbf{d}}\big) + \left.\frac{\partial \mathbf{g}(\mathbf{r})}{\partial \mathbf{r}}\right|_{\hat{\mathbf{r}}_{k-1}+\hat{\mathbf{T}}_{k-1}^{\mathrm{T}}\hat{\mathbf{d}}} \Big(\widehat{\mathbf{T}}(t)^{\mathrm{T}} - \widehat{\mathbf{T}}_{k-1}\Big)\,\widehat{\mathbf{d}} \\ &= \widehat{\mathbf{g}}_{k-1} + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\boldsymbol{\theta}}(t)\times]\widehat{\mathbf{d}}, \qquad \forall\, t \in [t_{k-1}, t_k], \end{aligned} \tag{4.27}$$

this can be interpreted as assuming that the acceleration of gravity at the IMU location is constant over the time step. The contribution of $\widehat{\mathbf{G}}_{k-1}[\widehat{\boldsymbol{\theta}}(t)\times]\widehat{\mathbf{d}}$ is small but will be kept for completeness. Using Eq. (4.27), we rewrite Eq. (4.26) as

$$\begin{aligned}\dot{\widehat{\mathbf{v}}}(t) &= \widehat{\mathbf{g}}_{k-1} - \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times]\,\widehat{\boldsymbol{\theta}}(t) + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\Delta\widehat{\mathbf{T}}(t)\mathbf{a}_{m,k} \qquad (4.28)\\ &\simeq \widehat{\mathbf{g}}_{k-1} + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\mathbf{a}_{m,k} - \left(\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times] + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\mathbf{a}_{m,k}\times]\right)\widehat{\boldsymbol{\theta}}(t), \quad t\in[t_{k-1},t_k],\end{aligned}$$

Integrating Eq. (4.28) yields

$$\widehat{\mathbf{v}}_k = \widehat{\mathbf{v}}_{k-1} + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\Delta\mathbf{v}_{m,k} + \widehat{\mathbf{g}}_{k-1}\Delta t - \frac{1}{2}\left(\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times] + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\mathbf{a}_{m,k}\times]\right)\Delta\boldsymbol{\theta}_{m,k}\Delta t. \qquad (4.29)$$

Integrating Eq. (4.29) yields the estimate position

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

In summary, the estimated states are obtained via

$$\begin{aligned}\widehat{\mathbf{r}}_k &= \widehat{\mathbf{r}}_{k-1} + \widehat{\mathbf{v}}_{k-1}\Delta t + \frac{1}{2}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left(\mathbf{I}_{3\times 3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\Delta t + \qquad (4\text{-}30\text{a})\\ &\qquad + \frac{1}{2}\left(\widehat{\mathbf{g}}_{k-1} - \frac{1}{3}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times]\,\Delta\boldsymbol{\theta}_{m,k}\right)\Delta t^2\\ \widehat{\mathbf{v}}_k &= \widehat{\mathbf{v}}_{k-1} + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left(\mathbf{I}_{3\times 3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k} + \qquad (4\text{-}30\text{b})\\ &\qquad + \left(\widehat{\mathbf{g}}_{k-1} - \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times]\Delta\boldsymbol{\theta}_{m,k}\right)\Delta t\\ \widehat{\overline{\mathbf{q}}}_k &= \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k})\otimes\widehat{\overline{\mathbf{q}}}_{k-1}. \qquad (4\text{-}30\text{c})\end{aligned}$$

If it is desired to have a more accurate representation of the gravitational acceleration, the time step can be divided to use a higher-order method, each sub-step will employ an equation similar to Eqs. (4-30a)–(4-30c). Then all contributions will be added together in a weighted average. Notice that only the contribution due to gravity will be represented more accurately. Relying solely on the IMU integral measurements, point-wise in time quantities are not available and discretization errors are unavoidable. A higher order-method would be preferable if the IMU was providing measurements at a low frequency; normally the IMU can function at 10 Hertz or higher, which makes the assumption of constant gravitational acceleration in between measurements very reasonable.

Solution of the navigation equations given in Eqs. (4-30a)–(4-30c) yields the navigated spacecraft position, velocity, and attitude. As in the case of continuous-time measurements, the accuracy of this open-loop navigation architecture is strongly dependent on the knowledge of the initial spacecraft state. Also, measurement errors present in $\Delta\boldsymbol{\theta}_k$ and $\Delta\mathbf{v}_k$ are not filtered, and will corrupt the navigation solution directly.

In order to solve for position and velocity it is necessary to rotate the IMU accelerations from the case frame to the inertial frame. Therefore, any estimation error in the attitude estimate affects the position and velocity estimate. The attitude estimation does not rely on the position and velocity estimation, hence can be addressed independently.

4.3.1 Attitude estimation errors

We assume that the attitude propagates according to

$$\overline{\mathbf{q}}_k = \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{true,k}) \otimes \overline{\mathbf{q}}_{k-1}. \tag{4.31}$$

This is only an approximation but the discretization error will be compensated via process noise. Define the multiplicative attitude error as

$$\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1}.$$

Using Eqs. (4-30c) and (4.31), we obtain

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

which is equivalent to

$$\delta\overline{\mathbf{q}}_k = \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{true,k}) \otimes \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k})^{-1} \otimes \left(\begin{bmatrix} \mathbf{T}(\Delta\boldsymbol{\theta}_{m,k}) & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \delta\overline{\mathbf{q}}_{k-1} \right). \tag{4.32}$$

Assuming small angles, the vector component of the quaternion fully represents the attitude since

$$\delta\overline{\mathbf{q}} \simeq \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix}, \qquad \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{true,k}) \otimes \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k})^{-1} \simeq \begin{bmatrix} \frac{1}{2}\left(\Delta\boldsymbol{\theta}_{true,k} - \Delta\boldsymbol{\theta}_{m,k}\right) \\ 1 \end{bmatrix}.$$

From Eq. (4.32), approximating to first-order yields

$$\delta\mathbf{q}_k = \mathbf{T}(\Delta\boldsymbol{\theta}_{m,k})\,\delta\mathbf{q}_{k-1} + \frac{1}{2}\left(\Delta\boldsymbol{\theta}_{true,k} - \Delta\boldsymbol{\theta}_{m,k}\right).$$

It then follows from Eq. (4.3) that

$$\delta\mathbf{q}_k = \mathbf{T}(\Delta\boldsymbol{\theta}_{m,k})\,\delta\mathbf{q}_{k-1} + \frac{1}{2}\left[(\mathbf{I} + \boldsymbol{\Delta}_g)^{-1}\Delta\boldsymbol{\theta}_{m,k} - \mathbf{b}_g - \boldsymbol{\eta}_{g,k} - \Delta\boldsymbol{\theta}_{m,k}\right].$$

To first-order in $\mathbf{\Delta}_g$ we have

$$(\mathbf{I} + \mathbf{\Delta}_g)^{-1} \simeq \mathbf{I} - \mathbf{\Delta}_g .$$

It then follows that to first-order

$$\delta \mathbf{q}_k = \mathbf{T}(\Delta \boldsymbol{\theta}_{m,k})\, \delta \mathbf{q}_{k-1} - \frac{1}{2}\left(\mathbf{\Delta}_g\, \Delta \boldsymbol{\theta}_{m,k} + \mathbf{b}_g + \boldsymbol{\eta}_{g,k}\right) \ .$$

With the given definition of $\mathbf{S}_g$ in Eq. (4.1), we have the following relationship

$$\mathbf{S}_g \Delta \boldsymbol{\theta}_{m,k} = \mathbf{D}(\Delta \boldsymbol{\theta}_{m,k}) \mathbf{s}_g ,$$

where

$$\mathbf{s}_g \triangleq \begin{bmatrix} s_{gx} & s_{gy} & s_{gz} \end{bmatrix}^{\mathrm{T}} ,$$

and $\mathbf{D}(\cdot)$ is defined in Eq. (4.6). Similarly, we have

$$\mathbf{\Gamma}_g \Delta \boldsymbol{\theta}_{m,k} = \mathbf{N}(\Delta \boldsymbol{\theta}_{m,k}) \boldsymbol{\gamma}_g \ ,$$

where

$$\boldsymbol{\gamma}_g \triangleq \begin{bmatrix} \gamma_{g_{xy}} & \gamma_{g_{xz}} & \gamma_{g_{yx}} & \gamma_{g_{yz}} & \gamma_{g_{zx}} & \gamma_{g_{zy}} \end{bmatrix}^{\mathrm{T}} ,$$

and $\mathbf{N}(\cdot)$ is defined in Eq. (4.7). For small angles the rotation vector $\boldsymbol{\theta}$ is approximately twice the vector part of the quaternion, therefore it follows that the estimation error represented with the rotation vector is given by

$$\boldsymbol{e}_{\theta,k} = \mathbf{T}(\Delta \boldsymbol{\theta}_{m,k})\, \boldsymbol{e}_{\theta,k-1} - \mathbf{D}(\Delta \boldsymbol{\theta}_{m,k}) \mathbf{s}_g - \mathbf{N}(\Delta \boldsymbol{\theta}_{m,k}) \boldsymbol{\gamma}_g - \mathbf{b}_g - \boldsymbol{\eta}_{g,k} . \tag{4.33}$$

4.3.2 Position and velocity estimation errors

The assumption made in Eq. (4.31) is equivalent to assuming constant angular velocity in between measurements. Similarly the gravitational and nongravitational accelerations will be assumed constant during the time step. These assumptions lead to equations for the propagation of the true state equivalent to Eqs. (4-30a)–(4-30c)

$$\begin{aligned}\mathbf{r}_k &= \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t + \frac{1}{2}\mathbf{g}_{k-1}\Delta t^2 + \frac{1}{2}\mathbf{T}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times 3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{true,k}\times\right]\right)\Delta\mathbf{v}_{true,k}\Delta t + \\ &\qquad - \frac{1}{6}\mathbf{G}_{k-1}\mathbf{T}_{k-1}^{\mathrm{T}}[\mathbf{d}\times]\Delta\boldsymbol{\theta}_{true,k}\Delta t^2 \\ \mathbf{v}_k &= \mathbf{v}_{k-1} + \mathbf{g}_{k-1}\Delta t + \mathbf{T}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times 3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{true,k}\times\right]\right)\Delta\mathbf{v}_{true,k} + \\ &\qquad - \frac{1}{2}\mathbf{G}_{k-1}\mathbf{T}_{k-1}^{\mathrm{T}}[\mathbf{d}\times]\Delta\boldsymbol{\theta}_{true,k}\Delta t.\end{aligned}$$

To compensate for the error introduced by the discretization, process noise will be added. The position and velocity estimation error are defined to be

$$\mathbf{e}_{r,k} \triangleq \mathbf{r}_k - \widehat{\mathbf{r}}_k \qquad \text{and} \qquad \mathbf{e}_{v,k} \triangleq \mathbf{v}_k - \widehat{\mathbf{v}}_k.$$

Computing $\mathbf{e}_{r,k}$ yields

$$\begin{aligned}\mathbf{e}_{r,k} &= \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1}\Delta t + \frac{1}{2}(\mathbf{g}_{k-1} - \widehat{\mathbf{g}}_{k-1})\Delta t^2 - \frac{1}{6}\mathbf{G}_{k-1}\mathbf{T}_{k-1}^{\mathrm{T}}[\mathbf{d}\times]\Delta\boldsymbol{\theta}_{true,k}\Delta t^2 + \\ &\quad + \frac{1}{6}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta\boldsymbol{\theta}_{m,k}\Delta t^2 - \frac{1}{2}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times 3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m},k\times\right]\right)\Delta\mathbf{v}_{m,k}\Delta t + \\ &\quad + \frac{1}{2}\mathbf{T}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times 3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{true,k}\times\right]\right)\left(\Delta\mathbf{v}_{true,k}\right)\Delta t,\end{aligned}$$

vector $\widehat{\mathbf{d}}$ is the estimate of the distance between the IMU and the center of mass. Expanding gravity, utilizing a Taylor series and neglecting higher order terms, it follows that

$$\mathbf{g}(\mathbf{r} + \mathbf{T}^{\mathrm{T}}\mathbf{d}) - \mathbf{g}(\widehat{\mathbf{r}} + \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}) \simeq \widehat{\mathbf{G}} \left(\mathbf{e}_r + \mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}}\right),$$

where

$$\widehat{\mathbf{G}} \triangleq \left.\frac{\partial \mathbf{g}}{\partial \mathbf{r}}\right|_{\mathbf{r}=\hat{\mathbf{r}}+\hat{\mathbf{T}}^{\mathrm{T}}\hat{\mathbf{d}}}.$$

Since the quaternion error is defined as $\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{\,-1}$ and attitude matrices are multiplied in the same order as quaternions, then $\delta\mathbf{T} = \mathbf{T}\,\widehat{\mathbf{T}}^{\mathrm{T}}$. Therefore,

$$\mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}} = \widehat{\mathbf{T}}^{\mathrm{T}}\,\delta\mathbf{T}^{\mathrm{T}}(\widehat{\mathbf{d}} + \mathbf{e}_d) - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}},$$

where $\mathbf{e}_d \triangleq \mathbf{d} - \widehat{\mathbf{d}}$. To first-order it follows that

$$\delta\mathbf{T}^{\mathrm{T}} \simeq \mathbf{I}_{3\times3} + [\mathbf{e}_\theta\times].$$

Then,

$$\mathbf{T}^{\mathrm{T}}\mathbf{d} - \widehat{\mathbf{T}}^{\mathrm{T}}\widehat{\mathbf{d}} \simeq \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{e}_\theta\times]\widehat{\mathbf{d}} + \widehat{\mathbf{T}}^{\mathrm{T}}\mathbf{e}_d = -\widehat{\mathbf{T}}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\mathbf{e}_\theta + \widehat{\mathbf{T}}^{\mathrm{T}}\,\mathbf{e}_d.$$

Similarly,

$$\mathbf{T}^{\mathrm{T}}\Delta\mathbf{v}_{true} - \widehat{\mathbf{T}}^{\mathrm{T}}\Delta\mathbf{v}_m = \widehat{\mathbf{T}}^{\mathrm{T}}\,\delta\mathbf{T}^{\mathrm{T}}\Delta\mathbf{v}_{true} - \widehat{\mathbf{T}}^{\mathrm{T}}\Delta\mathbf{v}_m,$$

hence, to first-order, we have

$$\mathbf{T}^{\mathrm{T}}\Delta\mathbf{v}_{true} - \widehat{\mathbf{T}}^{\mathrm{T}}\Delta\mathbf{v}_m \simeq \widehat{\mathbf{T}}^{\mathrm{T}}[\mathbf{e}_\theta\times]\Delta\mathbf{v}_m + \widehat{\mathbf{T}}^{\mathrm{T}}\left(\Delta\mathbf{v}_{true} - \Delta\mathbf{v}_m\right).$$

Finally the position estimation error is obtained to first-order as

$$\begin{aligned}\mathbf{e}_{r,k} = {} & \mathbf{e}_{r,k-1} + \mathbf{e}_{v,k-1}\Delta t - \frac{1}{2}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left[\left(\mathbf{I}_{3\times3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\times\right]\mathbf{e}_{\theta}\Delta t \\ & + \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\left(\mathbf{e}_r + \frac{1}{3}\widehat{\mathbf{T}}^{\mathrm{T}}[(\widehat{\mathbf{d}}\times\Delta\boldsymbol{\theta}_{m,k})\times]\mathbf{e}_{\theta} + \frac{1}{3}\widehat{\mathbf{T}}^{\mathrm{T}}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\mathbf{e}_d\right)\Delta t^2 + \\ & + \frac{1}{2}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\left(\Delta\mathbf{v}_{true,k} - \Delta\mathbf{v}_{m,k}\right)\Delta t - \frac{1}{6}\widehat{\mathbf{U}}_{k-1}\mathbf{e}_r\Delta t^2 + \\ & - \frac{1}{6}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right)(\Delta\boldsymbol{\theta}_{true,k} - \Delta\boldsymbol{\theta}_{m,k})\Delta t.\end{aligned} \tag{4.34}$$

Following a similar pattern, the velocity estimation error is given by

$$\begin{aligned}\mathbf{e}_{v,k} = {} & \mathbf{e}_{v,k-1} + \widehat{\mathbf{G}}_{k-1}\mathbf{e}_r\Delta t - \widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left[\left(\mathbf{I}_{3\times3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\times\right]\mathbf{e}_{\theta} \\ & + \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}\left(\left[\left(\widehat{\mathbf{d}}\times\Delta\boldsymbol{\theta}_{m,k}\right)\times\right]\mathbf{e}_{\theta} + [\Delta\boldsymbol{\theta}_{m,k}\times]\mathbf{e}_d\right)\Delta t + \\ & + \widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\left(\Delta\mathbf{v}_{true,k} - \Delta\mathbf{v}_{m,k}\right) - \frac{1}{2}\widehat{\mathbf{U}}_{k-1}\mathbf{e}_r\Delta t + \\ & - \frac{1}{2}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right)(\Delta\boldsymbol{\theta}_{true,k} - \Delta\boldsymbol{\theta}_{m,k}).\end{aligned} \tag{4.35}$$

In Eqs. (4.34) and (4.35), the ij component of matrix $\widehat{\mathbf{U}}$ is defined as

$$\begin{aligned}\widehat{U}(ij) &\triangleq \sum_{l=1}^{3}\frac{\partial^2 g(i)}{\partial r(j)\,\partial r(l)}\,u(l)\Bigg|_{\mathbf{r}=\hat{\mathbf{r}}+\hat{\mathbf{T}}^{\mathrm{T}}\hat{\mathbf{d}}} \\ \mathbf{u} &\triangleq \widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta\boldsymbol{\theta}_{m,k}.\end{aligned}$$

This term arises from the difference in gravitational acceleration between the center of mass and the IMU location, and should be neglected in any practical application. Rearranging the terms in the IMU model given in Eq. (4.3) yields

$$\Delta\mathbf{v}_m = (\mathbf{I} + \boldsymbol{\Delta}_a)^{-1}\Delta\mathbf{v}_{true} - (\mathbf{b}_a + \boldsymbol{\eta}_a),$$

where, after some manipulation and using the fact that for "small" $\boldsymbol{\Delta}_a$, $(\mathbf{I}+\boldsymbol{\Delta}_a)^{-1} \simeq \mathbf{I} - \boldsymbol{\Delta}_a$, we have

$$\Delta\mathbf{v}_{true} - \Delta\mathbf{v}_m = -\boldsymbol{\Delta}_a \Delta\mathbf{v}_m - (\mathbf{b}_a + \boldsymbol{\eta}_a).$$

Like in the continuous-time case, matrix $\boldsymbol{\Delta}_a$ is comprised of random constants, $\mathbf{b}_a$ is a random constant vector, $\mathbf{e}_d$ is a random constant if entry is ballistic, and $\boldsymbol{\eta}$ is a random sequence. From the definition of $\boldsymbol{\Delta}_a$ given in Eq. (4.2), we have

$$\boldsymbol{\Delta}_a \Delta\mathbf{v}_m = (\boldsymbol{\Gamma}_a + \mathbf{S}_a)\, \Delta\mathbf{v}_m.$$

With the definitions of $\boldsymbol{\Gamma}_a$ and $\mathbf{S}_a$ given in Eq. (4.1), $\boldsymbol{\Delta}_a \mathbf{a}_k$ can also be written as

$$\boldsymbol{\Delta}_a \Delta\mathbf{v} = \mathbf{D}(\Delta\mathbf{v})\mathbf{s}_a + \mathbf{N}(\Delta\mathbf{v})\boldsymbol{\gamma}_a,$$

where definitions equivalent to those of Eqs. (4.6) and (4.7) are used.

Collecting the position, velocity, and attitude estimation error equations from Eqs. (4.33)–(4.35), and writing in matrix form yields the stochastic linear matrix difference equation

$$\mathbf{e}_k = \mathbf{F}_{k-1}\mathbf{e}_{k-1} + \mathbf{H}_{1,k-1}\, \mathbf{b} + \mathbf{H}_{2,k-1}\, \boldsymbol{\eta}_{k-1}, \tag{4.36}$$

where

$$\mathbf{e}_k \triangleq \begin{bmatrix} \mathbf{e}_{r,k} \\ \mathbf{e}_{v,k} \\ \mathbf{e}_{\theta,k} \end{bmatrix} \in \Re^9, \qquad \mathbf{b} \triangleq \begin{bmatrix} \mathbf{s}_a \\ \boldsymbol{\gamma}_a \\ \mathbf{b}_a \\ \mathbf{s}_g \\ \boldsymbol{\gamma}_g \\ \mathbf{b}_g \\ \mathbf{e}_d \end{bmatrix} \in \Re^{27}, \qquad \boldsymbol{\eta}_{k-1} \triangleq \begin{bmatrix} \boldsymbol{\eta}_{a,k} \\ \boldsymbol{\eta}_{g,k} \end{bmatrix} \in \Re^6.$$

The error state matrix $\mathbf{F} \in \Re^{9\times 9}$ is

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{I}_{3\times3} + \frac{1}{2}\left(\widehat{\mathbf{G}}_{k-1} - \frac{1}{3}\widehat{\mathbf{U}}_{k-1}\right)\Delta t^2 & \mathbf{I}_{3\times3}\Delta t & \mathbf{F}_{r\theta,k} \\ \left(\widehat{\mathbf{G}}_{k-1} - \frac{1}{2}\widehat{\mathbf{U}}_{k-1}\right)\Delta t & \mathbf{I}_{3\times3} & \mathbf{F}_{v\theta,k} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{F}_{\theta\theta,k} \end{bmatrix} \tag{4.37}$$

$$\begin{aligned} \mathbf{F}_{r\theta,k} =& \frac{1}{2}\Bigg\{\frac{1}{3}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}[(\widehat{\mathbf{d}} \times \Delta\boldsymbol{\theta}_{m,k})\times]\Delta t + \\ & \qquad\qquad -\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left[\left(\mathbf{I}_{3\times3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\times\right]\Bigg\}\Delta t \\ \mathbf{F}_{v\theta,k} =& \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}\left[\left(\widehat{\mathbf{d}} \times \Delta\boldsymbol{\theta}_{m,k}\right)\times\right]\Delta t - \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left[\left(\mathbf{I}_{3\times3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\times\right] \\ \mathbf{F}_{\theta\theta,k} =& \mathbf{T}(\Delta\boldsymbol{\theta}_{m,k}), \end{aligned}$$

and the input mapping matrices are $\mathbf{H}_{1,k} \in \Re^{9\times27}$ and $\mathbf{H}_{2,k} \in \Re^{9\times6}$ are

$$\mathbf{H}_{1,k-1} = \begin{bmatrix} \mathbf{H}_{1a,k-1}\left[\mathbf{DNI}_v\right] & \mathbf{H}_{1g,k-1}\ \left[\mathbf{DNI}_\theta\right] & \mathbf{H}_{1d,k-1} \end{bmatrix} \tag{4.38}$$

$$
\begin{aligned}
[\mathbf{DNI}_v] &= \begin{bmatrix}\mathbf{D}(\Delta\mathbf{v}_{m,k}) & \mathbf{N}(\Delta\mathbf{v}_{m,k}) & \mathbf{I}_{3\times3}\end{bmatrix} \\
[\mathbf{DNI}_\theta] &= \begin{bmatrix}\mathbf{D}(\Delta\boldsymbol{\theta}_{m,k}) & \mathbf{N}(\Delta\boldsymbol{\theta}_{m,k}) & \mathbf{I}_{3\times3}\end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{H}_{1a,k-1} &= \begin{bmatrix} -\frac{1}{2}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta t \\ -\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right) \\ \mathbf{O}_{3\times3} \end{bmatrix} \\
\mathbf{H}_{1g,k-1} &= \begin{bmatrix} \frac{1}{6}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right) \\ \frac{1}{2}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right) \\ -\mathbf{I}_{3\times3} \end{bmatrix} \\
\mathbf{H}_{1d,k-1} &= \begin{bmatrix} \frac{1}{6}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}[\Delta\boldsymbol{\theta}_{m,k}\times]\Delta t^2 \\ \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}[\Delta\boldsymbol{\theta}_{m,k}\times]\Delta t \\ \mathbf{O}_{3\times3} \end{bmatrix},
\end{aligned}
$$

and

$$
\mathbf{H}_{2,k-1} = \begin{bmatrix}\mathbf{H}_{2a,k-1} & \mathbf{H}_{2g,k-1}\end{bmatrix} \tag{4.39}
$$

$$
\begin{aligned}
\mathbf{H}_{2a,k-1} &= \begin{bmatrix} -\frac{1}{2}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta t \\ -\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}\left(\mathbf{I}_{3\times3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right) \\ \mathbf{O}_{3\times3} \end{bmatrix} \\
\mathbf{H}_{2g,k-1} &= \begin{bmatrix} \frac{1}{6}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right)\Delta t \\ \frac{1}{2}\left(\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\Delta\mathbf{v}_{m,k}\times] + \widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}_{k-1}^{\mathrm{T}}[\widehat{\mathbf{d}}\times]\Delta t\right) \\ -\mathbf{I}_{3\times3} \end{bmatrix}.
\end{aligned}
$$

The components of $\mathbf{b}$ in Eq. (4.36) are the various random constant errors associated with the IMU, where it assumed that

$$\mathrm{E}\left\{\mathbf{b}\right\} = \mathbf{0},$$

and $\mathbf{B} \in \Re^{27\times 27}$ is

$$\mathbf{B} \triangleq \mathrm{E}\left\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\right\}.$$

The components of $\boldsymbol{\eta}_k$ in Eq. (4.36) are the non-constant random components of the IMU errors, where it is assumed that

$$\mathrm{E}\left\{\boldsymbol{\eta}_k\right\} = \mathbf{0} \qquad \text{and} \qquad \mathrm{E}\left\{\boldsymbol{\eta}_i\boldsymbol{\eta}_j^{\mathrm{T}}\right\} = \mathbf{V}_i\delta_{ij}.$$

Finally, it is assumed that $\mathrm{E}\left\{\boldsymbol{\eta}_k\mathbf{b}^{\mathrm{T}}\right\} = \mathbf{O}$ for all k.

The error covariance in the IMU dead-reckoning case can be computed with the technique developed in § 3.2.1. Only the propagation phase needs to be computed since no updates are performed. The following substitutions need to be made from the equations in Table 3.1.

$$\mathbf{Q}_k \leftarrow \mathbf{H}_{2,k}\mathbf{V}_k\mathbf{H}_{2,k}^{\mathrm{T}}, \qquad\qquad \boldsymbol{\Upsilon}_k \leftarrow \mathbf{H}_{1,k}.$$

4.3.3 Dead Reckoning Navigation

Suppose that the IMU observations, $\Delta\mathbf{v}_{m,k}$ and $\Delta\boldsymbol{\theta}_{m,k}$ are available. Then, dead reckoning navigation, including computing the associated state estimation error covariance, is the process of solving the following equations at each time t_k an IMU observation is available:

$$\begin{aligned}
\widehat{\mathbf{r}}_k &= \widehat{\mathbf{r}}_{k-1} + \widehat{\mathbf{v}}_{k-1}\Delta t + \frac{1}{2}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left(\mathbf{I}_{3\times 3} + \frac{1}{3}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k}\Delta t + \\
&\qquad + \frac{1}{2}\left(\widehat{\mathbf{g}}_{k-1} - \frac{1}{3}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times]\,\Delta\boldsymbol{\theta}_{m,k}\right)\Delta t^2 \\
\widehat{\mathbf{v}}_k &= \widehat{\mathbf{v}}_{k-1} + \widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}\left(\mathbf{I}_{3\times 3} + \frac{1}{2}\left[\Delta\boldsymbol{\theta}_{m,k}\times\right]\right)\Delta\mathbf{v}_{m,k} + \\
&\qquad + \left(\widehat{\mathbf{g}}_{k-1} - \frac{1}{2}\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{T}}^{\mathrm{T}}_{k-1}[\widehat{\mathbf{d}}\times]\Delta\boldsymbol{\theta}_{m,k}\right)\Delta t \\
\widehat{\overline{\mathbf{q}}}_k &= \overline{\mathbf{q}}(\Delta\boldsymbol{\theta}_{m,k}) \otimes \widehat{\overline{\mathbf{q}}}_{k-1} \\
\mathbf{L}_k &= \mathbf{F}_{k-1}\mathbf{L}_{k-1} + \mathbf{H}_{k-1} \\
\mathbf{P}_k &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}^{\mathrm{T}}_{k-1} + \mathbf{J}_{k-1}\mathbf{V}_{k-1}\mathbf{J}^{\mathrm{T}}_{k-1} + \mathbf{H}_{k-1}\mathbf{B}\,\mathbf{H}^{\mathrm{T}}_{k-1} + \mathbf{F}_{k-1}\mathbf{L}_{k-1}\mathbf{B}\,\mathbf{H}^{\mathrm{T}}_{k-1} + \\
&\qquad + \mathbf{H}_{k-1}\mathbf{B}\,\mathbf{L}^{\mathrm{T}}_{k-1}\mathbf{F}^{\mathrm{T}}_{k-1}.
\end{aligned}$$

where $\widehat{\mathbf{g}}_k \triangleq \mathbf{g}(\widehat{\mathbf{r}}_k + \widehat{\mathbf{T}}^{\mathrm{T}}_k\widehat{\mathbf{d}})$ is the modeled gravity, $\widehat{\overline{\mathbf{q}}} = \begin{bmatrix}\widehat{\mathbf{q}}^{\mathrm{T}} & \widehat{q}\end{bmatrix}^{\mathrm{T}}$ and

$$\widehat{\mathbf{T}}^{\mathrm{T}}_k \triangleq \mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}} = \mathbf{I}_{3\times 3} + 2\widehat{q}_k[\widehat{\mathbf{q}}_k\times] + 2[\widehat{\mathbf{q}}_k\times]^2,$$

where $\mathbf{F}$, $\mathbf{H}_1$, and $\mathbf{H}_2$ are given in Eqs. (4.37)–(4.39). The initial conditions are

$$\widehat{\mathbf{r}}_0 = \widehat{\mathbf{r}}(t_0), \quad \widehat{\mathbf{v}}_0 = \widehat{\mathbf{v}}(t_0), \quad \widehat{\overline{\mathbf{q}}}_0 = \widehat{\overline{\mathbf{q}}}(t_0), \quad \mathbf{P}_0 = \mathbf{P}(t_0), \quad \mathbf{L}_0 = \mathbf{0}.$$

The IMU provides discrete observations $\Delta\mathbf{v}_{m,k}$ and $\Delta\boldsymbol{\theta}_{m,k}$, and their error models are assumed known and represented by the matrices $\mathbf{V}_k$ and $\mathbf{B}$.

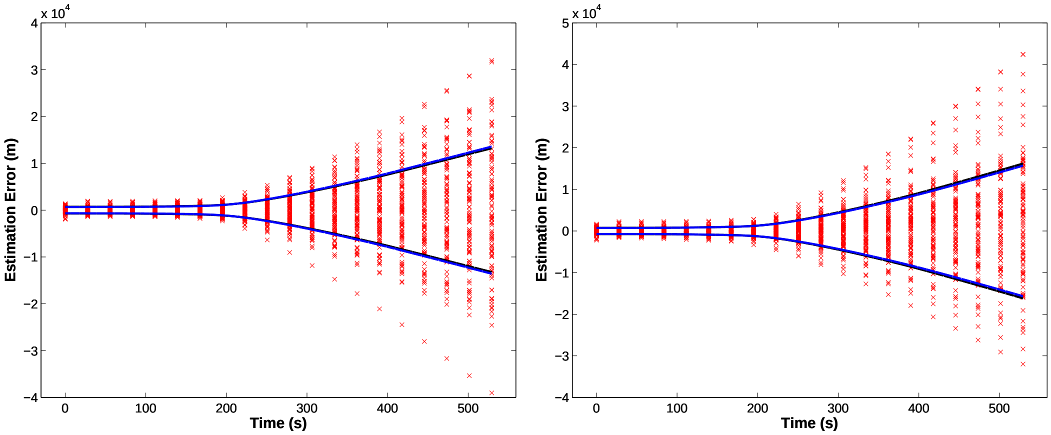
4.3.4 Simulation Results

The simulation with which the discrete algorithm is tested, uses the same trajectory and the same philosophy as the continuous-time case. The only differences (besides the navigation equations) are the true measurements, which are now $\Delta\mathbf{v}$'s and $\Delta\boldsymbol{\theta}$'s, and their errors, which are given in Table 4.2.

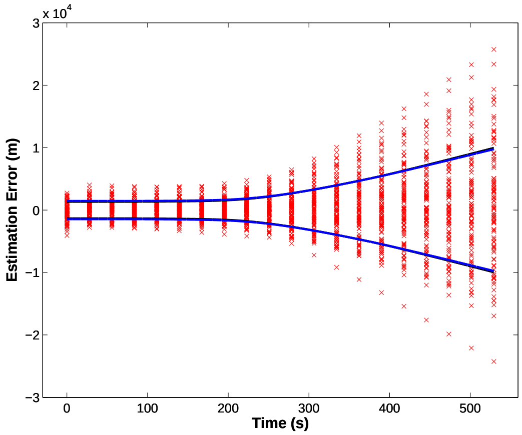
Accelerometer Noise	$\boldsymbol{\eta}_a$	10 $[\mu g\, s]$
Accelerometer Bias	$\mathbf{b}_a$	0.01 $[mg\, s]$
Accelerometer Scale Factor	$\mathbf{s}_a$	175 $[ppm]$
Accelerometer Misalignment	$\boldsymbol{\gamma}_a$	5 $[arcsec]$
Gyro Noise	$\boldsymbol{\eta}_g$	0.01 $[arcsec]$
Gyro Bias	$\mathbf{b}_g$	0.05 $[arcsec]$
Gyro Scale Factor	$\mathbf{s}_g$	5 $[ppm]$
Gyro Misalignment	$\boldsymbol{\gamma}_g$	5 $[arcsec]$

Table 4.2: Discrete-time IMU errors

Figures 4.4–4.6 show samples of error evolution in each of the 100 runs (denoted by red x), the sample covariance (blue lines), and the linear covariance formulation evaluated (black lines). Figures 4.4–4.5 contain the inertial position and velocity errors in the x, y, and z axis respectively. Figure 4.6 contains the three components of the attitude error from estimated body frame to true body frame, the attitude error is represented as a rotation vector. As in the continuous-time case, the sample covariance and the filter covariance match, demonstrating that the linear covariance formulation correctly represent the estimation error.

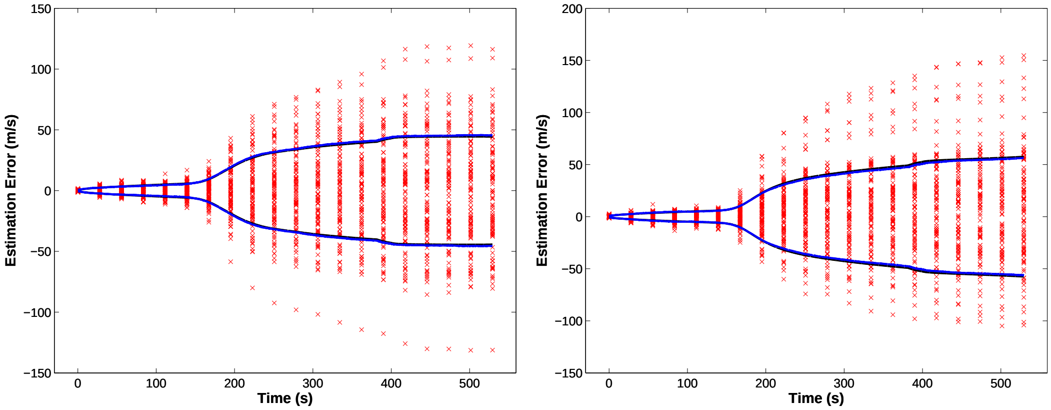


(a) Inertial x axis (b) Inertial y axis



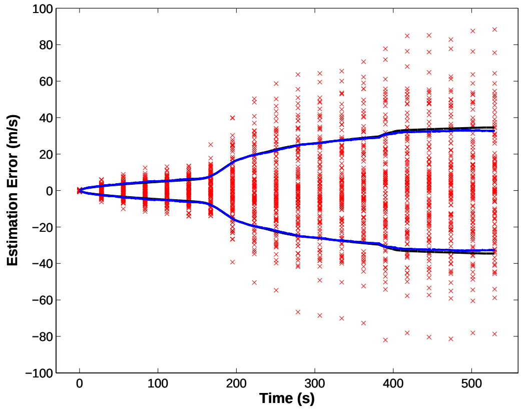
(c) Inertial z axis

Figure 4.4: Position estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.



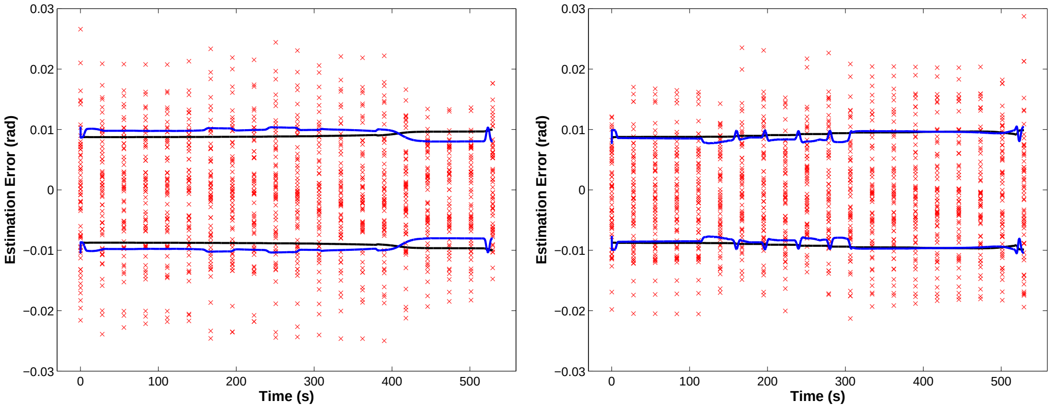
(a) Inertial x axis

(b) Inertial y axis

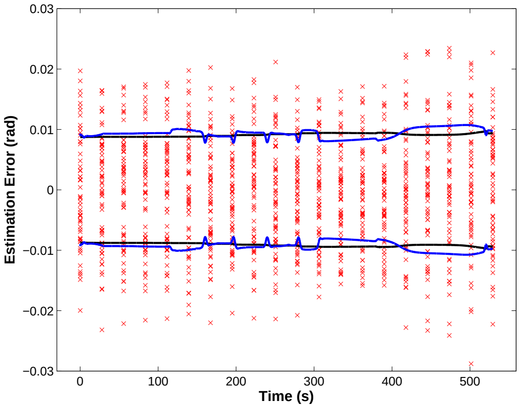


(c) Inertial z axis

Figure 4.5: Velocity estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.



(a) First component (b) Second component



(c) Third component

Figure 4.6: Attitude estimation error. Error denoted by red x, sample covariance by blue line, and calculated covariance by black line.

4.4 Conclusions

In this chapter, the algorithms for precise dead-reckoning navigation were derived to include the state estimation error covariance computation. The underlying error equations were linearized and utilized to develop a formulation of the approximate state estimation error covariance. The correlation of attitude errors with position and velocity errors was explicitly derived. The resulting set of dead-reckoning relationships can be used as an independent verification of Monte Carlo analysis during the verification of the entry filter.

The importance of this example lays in the fact that the vast majority of spacecraft applications use the IMU to propagate the states. During Mars entry, the IMU is the only available sensor, therefore the dead-reckoning approach necessarily results in an increase of the estimates uncertainty. Usually, unmodeled sensors biases are handled by heuristically increasing the filter's tuning parameters: the noise covariances (or spectral densities). While this approach has proven to be reliable, it necessarily introduces an additional level of approximation, which might be incompatible with pin-point landing requirements. The proposed approach explicitly takes into account the effects of uncompensated biases, hence leaving linearization as the only approximation made.

In the phases of EDL following entry, this explicitly accounting for the uncompensated biases errors might not be necessary. One reason is that the availability of other sensors will drive the uncertainty down, and the updates will reduces the effect of the bias errors accumulating through then numerical integration. State updates increase the computational demand of the navigation system, therefore it might be preferable to avoid the added computational complexity necessary to account for the uncompensated biases.

The numbers shown in the plots of this chapter are highly variable depending on the mission and the hardware. What is important then, is not the numbers themselves but the tendencies. It can be seen that the attitude uncertainty stays relatively constant, while the translational uncertainty grows significantly. The reason is readily explained: only gyro errors contribute to attitude uncertainty. Accelerometer errors, unmodeled gravitational acceleration, and attitude estimation errors, all contribute to the uncertainty on translational states. Therefore is the translational states estimation that most needs to be improved, that will be the topic of next chapter. In chapter 5 the accelerometer will be filtered in a model-based Kalman filter to improve the estimate of translation states. This approach is often consider non-practical because the uncertainty on Mars atmosphere is too high to model. Our approach to avoid that situation is to use measurement taken from an actual mission, NASA's Mars Exploration Rovers. Since we do not generate the "truth", we are not able to match the "true" model with the filter model, hence the high uncertainly of the real mission is correctly replicated.

Chapter 5

Adaptive Entry Navigation

The IMU dead-reckoning approach presented in the previous chapter is the most widely used because of its simplicity. Dead-reckoning is a common procedure not only in EDL applications, but in other areas of aerospace engineering, such as missile and aircraft navigation. Mars entry possibly makes an even stronger case for dead-reckoning than Earth-based applications, because the knowledge of the atmospheric conditions is limited, therefore the process noise introduced by aerodynamic forces modeling would most likely be greater than the IMU measurement error. If the propagation model is poor, the Kalman filter should rely heavily on the measurements to formulate its estimate. This occurrence would result in a rapid increase of the covariance during propagation, followed by a big decrease after the update. A very precise measurement (with respect to the process noise) can cause filter divergence because of the linearization approximation. The estimation error covariance in the Kalman filter is obtained through a linear model, justified by the fact that the estimation error should be small. However, if the estimation error covariance drops abruptly the estimation error might not be able to "follow" the covariance, causing divergence of the filter. These implementation problems are very well known [4] and are solved by "tuning" the filter. Therefore, the possible performance advantages of the filter approach are best demonstrated through Monte Carlo analysis. Other advantages, such as data drop-out robustness, do not need to be demonstrated.

In this work, the IMU measurements are not generated through a simulation, but are observation taken from the Mars Exploration Rover mission. The advantage of using real measurements is that the true atmospheric parameters are unknown, therefore the high uncertainties of the actual mission are well simulated since no "true" atmospheric model exists. The disadvantage is that the actual trajectory is also unknown, therefore the error generation is not exact. In this work, a rough best estimated trajectory (BET) was employed as the "true" state.

The approach taken here in the EKF development is to update the position and velocity estimates with the accelerometer measurement, and to dead-reckon the attitude estimate. The reasoning is as follows. Once through the upper atmosphere hypersonic/supersonic phase, other EDL sensors (such as the altimeter and velocimeter) will be available to provide information about translational states. Those measurements can naturally be fused together with the accelerometer measurements within the EKF. On the other hand, it is assumed that there are no other attitude sensors available post-parachute deploy, hence the gyro is the only sensor capable of providing attitude information. There is no tangible benefit to updating the attitude estimate with the gyro data unless accompanied by an attitude dynamics model of sufficient complexity to capture the rotational motion of the spacecraft. The additional complexity of the navigation algorithm due to the attitude dynamics model was deemed to be too great for the potential benefit. If an external attitude sensor should in fact become available, then this issue would necessarily be re-visited. Some information on the attitude could be inferred from the accelerometer measurement, however a single vector measurement is not enough to estimate the attitude, therefore the accuracy of the estimate will depend strongly on the initial estimate – exactly like in the dead-reckoning approach. Some observability would be recovered if the inertial orientation of the non-gravitational acceleration would substantially change during entry, a fact that does not happen during MER entry. MER entry trajectory is without lift, therefore the aerodynamic acceleration is always along the velocity vector. Also, if the information provided by the accelerometer were used to update both the translational states and attitude, the translational state estimate would degrade. The optimal way to update only part of the state is to consider the correlation.

5.1 Partitioning the Kalman Filter State

Suppose that the $n \times 1$ state vector $\mathbf{x}$ is partitioned into $\mathbf{z}$ and $\mathbf{q}$ as

$$\mathbf{x} = \begin{bmatrix} \mathbf{z} \\ \mathbf{q} \end{bmatrix},$$

where $\mathbf{z}$ is $(n-p)\times 1$ and $\mathbf{q}$ is $p \times 1$. The estimation error associated with each partition will be minimized independently. The Kalman gain is partitioned appropriately as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_z \\ \mathbf{K}_q \end{bmatrix},$$

where $\mathbf{K}_z$ is $(n-p)\times m$, $\mathbf{K}_q$ is $p \times m$, m is the dimension of the measurement vector $\mathbf{y}$. At measurement time t_k, a linear update is assumed, where

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \begin{bmatrix} \mathbf{K}_{z,k} \\ \mathbf{K}_{q,k} \end{bmatrix} (\mathbf{y}_k - \widehat{\mathbf{y}}_k).$$

The measurement model is

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\eta}_k = \mathbf{H}_k(\widehat{\mathbf{x}}_k^- + \mathbf{e}_k^-) + \boldsymbol{\eta}_k = \widehat{\mathbf{y}}_k + \mathbf{H}_k \mathbf{e}_k^- + \boldsymbol{\eta}_k.$$

The estimation error covariance before the update can be partitioned as follows

$$\mathbf{P}_k^- = \begin{bmatrix} \mathbf{P}_{1,k} & \mathbf{P}_{2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{zz,k}^- & \mathbf{P}_{zq,k}^- \\ \mathbf{P}_{qz,k}^- & \mathbf{P}_{qq,k}^- \end{bmatrix}; \qquad \mathbf{P}_{2,k} \in \Re^{n\times p}.$$

Note that

$$\begin{aligned}
\mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- &= \begin{bmatrix} \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{1,k} & \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{2,k} \\ \mathbf{K}_{q,k}\mathbf{H}_k\mathbf{P}_{1,k} & \mathbf{K}_{q,k}\mathbf{H}_k\mathbf{P}_{2,k} \end{bmatrix} \\
\mathbf{K}\mathbf{R}_k\mathbf{K}^{\mathrm{T}} &= \begin{bmatrix} \mathbf{K}_{z,k}\mathbf{R}_k\mathbf{K}_{z,k}^{\mathrm{T}} & \mathbf{K}_{z,k}\mathbf{R}_k\mathbf{K}_{q,k}^{\mathrm{T}} \\ \mathbf{K}_{q,k}\mathbf{R}_k\mathbf{K}_{z,k}^{\mathrm{T}} & \mathbf{K}_{q,k}\mathbf{R}_k\mathbf{K}_{q,k}^{\mathrm{T}} \end{bmatrix}.
\end{aligned}$$

The partitioned *a posteriori* covariance is

$$\begin{aligned}
\mathbf{P}_{zz,k}^+ &= \mathbf{P}_{zz,k}^- - \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{1,k} - \mathbf{P}_{1,k}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_{z,k}^{\mathrm{T}} + \mathbf{K}_{z,k}\mathbf{W}_k\mathbf{K}_{z,k}^{\mathrm{T}} \\
\mathbf{P}_{zq,k}^+ &= \mathbf{P}_{zq,k}^- - \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{2,k} - \mathbf{P}_{1,k}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_{q,k}^{\mathrm{T}} + \mathbf{K}_{z,k}\mathbf{W}_k\mathbf{K}_{q,k}^{\mathrm{T}} \\
\mathbf{P}_{qq,k}^+ &= \mathbf{P}_{qq,k}^- - \mathbf{K}_{q,k}\mathbf{H}_k\mathbf{P}_{2,k} - \mathbf{P}_{2,k}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_{q,k}^{\mathrm{T}} + \mathbf{K}_{q,k}\mathbf{W}_k\mathbf{K}_{q,k}^{\mathrm{T}} \\
\mathbf{W}_k &= \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k.
\end{aligned}$$

The matrix $\mathbf{P}_{zz,k}^+$ is only a function of $\mathbf{K}_{z,k}$, and $\mathbf{P}_{qq,k}^+$ is only a function of $\mathbf{K}_{q,k}$. Also, the trace of $\mathbf{P}_k^+$ is equal to the sum of the traces of $\mathbf{P}_{zz,k}^+$ and $\mathbf{P}_{qq,k}^+$. The two facts imply that the minimum of the sum is equal to the sum of the minima, or

$$\begin{aligned}
\min_{\mathbf{K}_k}\left(\operatorname{trace}\mathbf{P}^+\right) &= \min_{\mathbf{K}_{z,k},\,\mathbf{K}_{q,k}}\left(\operatorname{trace}\mathbf{P}_{zz,k}^+ + \operatorname{trace}\mathbf{P}_{qq,k}^+\right) \\
&= \min_{\mathbf{K}_{z,k}}\left(\operatorname{trace}\mathbf{P}_{zz,k}^+\right) + \min_{\mathbf{K}_{q,k}}\left(\operatorname{trace}\mathbf{P}_{qq,k}^+\right).
\end{aligned}$$

The optimal gains are

$$\mathbf{K}_{z,k} = \mathbf{P}_{1,k}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1} \qquad \mathbf{K}_{q,k} = \mathbf{P}_{2,k}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{W}_k^{-1}. \tag{5.1}$$

As was expected, there is no difference in calculating the gains independently or together, because the correlation is taken into account in $\mathbf{P}_{1,k}$ and $\mathbf{P}_{2,k}$:

$$\mathbf{K}_k = \begin{bmatrix} \mathbf{K}_{z,k} \\ \mathbf{K}_{q,k} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{1,k}^{\mathrm{T}} \\ \mathbf{P}_{2,k}^{\mathrm{T}} \end{bmatrix} \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1} = \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1}.$$

There is no advantage in computing the gain via the partition because the full residuals covariance matrix still has to be inverted. Assume, however, that the updates of $\mathbf{q}$ and $\mathbf{x}$ are different. For example $\mathbf{q}$ is dead-reckoned. In this case, the optimal gain can be derived via the partition since the two minimizations will be different. The solution will not be a global optimum, but will be the optimum of all solutions that satisfy the constraint that $\mathbf{q}$ is not updated.

If $\mathbf{q}$ is to be dead-reckoned, then $\mathbf{K}_{q,k} = \mathbf{O}_{p\times m}$ and $\mathbf{K}_{z,k}$ is found with Eq. (5.1). The updated states are

$$\begin{aligned} \mathbf{z}_k^+ =& \mathbf{z}_k^+ + \mathbf{K}_{z,k}(\mathbf{y}_k - \widehat{\mathbf{y}}_k) \\ \mathbf{q}_k^+ =& \mathbf{q}_k^-. \end{aligned}$$

The updated covariance is

$$\begin{aligned} \mathbf{P}_{zz,k}^+ =& \mathbf{P}_{zz,k}^- - \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{1,k} - \mathbf{P}_{1,k}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_{z,k}^{\mathrm{T}} + \mathbf{K}_{z,k}\mathbf{W}_k\mathbf{K}_{z,k}^{\mathrm{T}} = \mathbf{P}_{zz,k}^- - \mathbf{K}_{z,k}\mathbf{W}_k\mathbf{K}_{z,k}^{\mathrm{T}} \\ \mathbf{W}_k =& \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \\ \mathbf{P}_{zq,k}^+ =& \mathbf{P}_{zq,k}^- - \mathbf{K}_{z,k}\mathbf{H}_k\mathbf{P}_{2,k} \\ \mathbf{P}_{qq,k}^+ =& \mathbf{P}_{qq,k}^-. \end{aligned}$$

Notice that the cross-covariance $\mathbf{P}_{zq,k}^+$ is updated. This approach guarantees that $\mathbf{z}^+$ is optimum out of all solutions that dead-reckon $\mathbf{q}$. The result is identical to computing the total gain with the conventional algorithm, and to force the $\mathbf{q}$

partition of the gain to zero, which is the bases of the consider filter [73, 74].

5.2 Single Extended Kalman Filter

In this section, various aspects of the extended Kalman filter are presented. It will be shown that is possible to improve state knowledge using an EKF processing IMU data. The EKF also serves as the main computational building block of the filter bank. Every filter in the bank will be of the form presented in this section, the only difference will be in the realization of the atmospheric model.

Given that the IMU is the only available sensor during the upper atmospheric hypersonic/supersonic phase of the EDL, the dead-reckoning approach discussed in Chapter 4 uses only state integration with given initial conditions. The accuracy of the initial conditions depend on the quality of the spacecraft tracking prior to entry interface. In the terminology of Kalman filtering, dead-reckoning represents state propagation only without any state updates. During the state propagation, the accuracy of the estimate degrades due to random and systematic errors in the IMU. Dependent on the accuracy of the IMU, the state estimation error covariance necessarily increases. Using a model-based EKF approach, the goal is to improve the state estimate sufficiently during the state update to compensate for atmospheric and IMU modelling errors. It is expected that only the estimate of velocity will be substantially improved over time, because position is very poorly observable from aerodynamic acceleration measurements. As the velocity estimation error decreases with time, filtering the IMU data will lead to better overall navigation than with dead-reckoning.

5.2.1 Filter Model

The translation and attitude motion of the spacecraft are modeled via

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v}\\ \dot{\mathbf{v}} &= \mathbf{g}(\mathbf{r}) + \mathbf{a}(\mathbf{r}, \mathbf{v}) + \boldsymbol{\nu}\\ \dot{\overline{\mathbf{q}}} &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\overline{\mathbf{q}},\end{aligned}$$

where $\mathbf{g}$ is the gravitation acceleration, $\mathbf{a}$ is the nongravitational acceleration. Unlike the dead-reckoning case, here $\mathbf{a}(\mathbf{r}, \mathbf{v})$ is not measured by the IMU but instead is modeled by the filter. All translational quantities are expressed in the inertial frame i, and the angular velocity $\boldsymbol{\omega}$ is expressed in the body frame b. Under the standard Kalman filter assumptions, the disturbance $\boldsymbol{\nu}$ is assumed to be a zero-mean, white noise process. The EKF propagation equations are given by

$$\begin{aligned}\dot{\widehat{\mathbf{r}}} &= \widehat{\mathbf{v}}\\ \dot{\widehat{\mathbf{v}}} &= \mathbf{g}(\widehat{\mathbf{r}}) + \mathbf{a}(\widehat{\mathbf{r}}, \widehat{\mathbf{v}})\\ \dot{\widehat{\overline{\mathbf{q}}}} &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m)\widehat{\overline{\mathbf{q}}},\end{aligned}$$

where $\boldsymbol{\omega}_m$ is the gyro measurement. Both the gyro and accelerometer measurements were partially compensated using estimates of their biases obtained in orbit. The IMU gyro measurement is given by

$$\boldsymbol{\omega}_m = (\mathbf{I}_{3\times 3} + \boldsymbol{\Gamma}_g)(\mathbf{I}_{3\times 3} + \mathbf{S}_g)(\boldsymbol{\omega} + \mathbf{b}_g + \boldsymbol{\eta}_g).$$

Unlike the previous chapter, all errors will be accounted as a single source

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + (\boldsymbol{\Gamma}_g + \mathbf{S}_g + \boldsymbol{\Gamma}_g\mathbf{S}_g)\boldsymbol{\omega} + (\mathbf{I}_{3\times 3} + \boldsymbol{\Gamma}_g)(\mathbf{I}_{3\times 3} + \mathbf{S}_g)(\mathbf{b}_g + \boldsymbol{\eta}_g),$$

be re-defining $\boldsymbol{\eta}_g$

$$\boldsymbol{\eta}_g \leftarrow (\boldsymbol{\Gamma}_g + \mathbf{S}_g + \boldsymbol{\Gamma}_g\mathbf{S}_g)\boldsymbol{\omega} + (\mathbf{I}_{3\times3} + \boldsymbol{\Gamma}_g)(\mathbf{I}_{3\times3} + \mathbf{S}_g)(\mathbf{b}_g + \boldsymbol{\eta}_g).$$

Here, the gyro model reduces to

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{\eta}_g.$$

An analogous procedure is performed on the accelerometer. There are good reasons to make those simplifications. Since the true measurements are from a mission and the errors were partially compensated is not possible to know the distribution of the individual errors. Only one set of IMU measurements is available, therefore statistical methods cannot be used to tune each covariance. It makes engineering sense to agglomerate them into a single error source. Also, the goal of this chapter is to show the benefits of filtering over dead-reckoning. Using a more complex model in both cases will not add or detract to the goal of showing the benefits to the filtering approach.

The estimation error is defined with the multiplicative quaternion formulation

$$\mathbf{e} \triangleq \begin{bmatrix}(\mathbf{r} - \widehat{\mathbf{r}})^{\mathrm{T}} & (\mathbf{v} - \widehat{\mathbf{v}})^{\mathrm{T}} & \delta\mathbf{q}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},$$

where $\delta\mathbf{q}$ is the vector component of the quaternion $\delta\overline{\mathbf{q}}$ defined as

$$\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1}.$$

To first-order, the evolution of the estimation error is given by

$$\dot{\mathbf{e}} = \frac{d}{dt}\begin{bmatrix}\mathbf{e}_r \\ \mathbf{e}_v \\ \delta\mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{e}_v \\ \mathbf{G}(\widehat{\mathbf{r}})\mathbf{e}_r + \mathbf{A}_r\mathbf{e}_r + \mathbf{A}_v\mathbf{e}_v + \boldsymbol{\nu} \\ -\boldsymbol{\omega}_m \times \delta\mathbf{q} - 0.5\,\boldsymbol{\eta}_g\end{bmatrix},$$

where

$$\mathbf{A}_r := \left.\frac{\partial\mathbf{a}}{\partial\mathbf{r}}\right|_{r=\hat{r},v=\hat{v}} \quad \text{and} \quad \mathbf{A}_v := \left.\frac{\partial\mathbf{a}}{\partial\mathbf{v}}\right|_{r=\hat{r},v=\hat{v}}.$$

The model for the aerodynamic acceleration expressed in the inertial frame is given by

$$\mathbf{a} = -\frac{c_d S}{2m}\rho\|\mathbf{v}_{rel}\|\mathbf{v}_{rel}.$$

The coefficient of drag, reference surface, and mass (parameters c_d, S, and m, respectively) are assumed to be known. The spacecraft velocity relative to the Mars atmosphere is denoted by $\mathbf{v}_{rel}$. For Mars, the greatest uncertainty is the density ρ.

5.2.2 Atmosphere Model

The atmospheric model used is the simplified COSPAR model based on Viking 1 and 2 and Mariner data [79]. COSPAR was the first attempt at defining a Martian standard atmosphere. It does not take into account longitude, latitude, seasons, and possible dust storms, although it is more precise than a simple exponential density model. The simplified equation for the COSPAR model is a modified exponential

$$\rho = \rho_1 \exp\{-\beta_\rho h + \gamma_\rho \cos(\omega_\rho h) + \delta_\rho \sin(\omega_\rho h)\}, \tag{5.2}$$

where h is the altitude. The nominal coefficients employed in Eq. (5.2) are

$$\rho_1 = 3.49210^6 (\text{g/cm}^3), \quad \beta_\rho = 0.09422, \quad \gamma_\rho = 1.5607$$
$$\delta_\rho = 0.3696, \quad \omega_\rho = \frac{2\pi}{300} (\text{km}^{-1}).$$

5.2.3 IMU Measurements

The measurements used are the IMU data collected during the MER EDL at a frequency of 8 Hertz. The IMU measurements represent the spacecraft change in velocity, $\Delta\mathbf{v}_m$, and change in angle, $\Delta\boldsymbol{\theta}_m$, (partially compensated for biases and misalignments) since the last IMU measurement. These $\Delta\mathbf{v}_m$ and $\Delta\boldsymbol{\theta}_m$ are divided by the time interval to obtain the measured acceleration, $\mathbf{a}_m^c$, and angular velocity, $\boldsymbol{\omega}_m$. The accelerometer measurement expressed in the IMU case frame $\mathbf{a}_m^c$ is rotated into the body frame through a known constant rotation matrix

$$\mathbf{a}_m^b = \mathbf{T}_c^b \, \mathbf{a}_m^c.$$

5.2.4 Measurement Model

The accelerometer measurement has two components: (i) the change in velocity due to nongravitational accelerations, and (ii) the change in velocity due to the offset between the center of mass and the accelerometer location, given by

$$\widehat{\mathbf{a}}_{offset} \simeq \boldsymbol{\omega}_m \times \boldsymbol{\omega}_m \times \mathbf{r}_{offset}.$$

The estimated measurement expressed in the body frame is

$$\widehat{\mathbf{a}}^b = \mathbf{T}^{\mathrm{T}}(\widehat{\mathbf{q}})\widehat{\mathbf{a}} + \widehat{\mathbf{a}}_{offset},$$

The residual $\boldsymbol{\epsilon}$ used to update position and velocity is given by

$$\boldsymbol{\epsilon} = \mathbf{a}_m^b - \widehat{\mathbf{a}}^b,$$

and can be approximated to first-order as

$$\boldsymbol{\epsilon} \simeq \mathbf{T}(\widehat{\mathbf{q}})\mathbf{A}_r\mathbf{e}_r + \mathbf{T}(\widehat{\mathbf{q}})\mathbf{A}_v\mathbf{e}_v + 2[\mathbf{T}(\widehat{\mathbf{q}})\widehat{\mathbf{a}}\times]\delta\mathbf{q} + \boldsymbol{\eta}_a,$$

where $\boldsymbol{\eta}_a$ is the accelerometer noise.

5.2.5 Filter Summary

The propagation equations are

$$\begin{aligned}
\dot{\widehat{\mathbf{r}}} &= \widehat{\mathbf{v}} \\
\dot{\widehat{\mathbf{v}}} &= \mathbf{g}(\widehat{\mathbf{r}}) + \mathbf{a}(\widehat{\mathbf{r}}, \widehat{\mathbf{v}}) \\
\dot{\widehat{\mathbf{q}}} &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m)\widehat{\mathbf{q}}, \\
\dot{\mathbf{P}} &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^{\mathrm{T}} + \mathbf{Q},
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{F} &= \begin{bmatrix} \mathbf{O}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{O}_{3\times3} \\ \mathbf{G}(\widehat{\mathbf{r}}) + \mathbf{A}_r & \mathbf{A}_v & \mathbf{O}_{3\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & -[\boldsymbol{\omega}_m\times] \end{bmatrix} \\
\mathbf{A}_r &= \left.\frac{\partial \mathbf{a}}{\partial \mathbf{r}}\right|_{r=\hat{r}, v=\hat{v}}, \qquad \mathbf{A}_v = \left.\frac{\partial \mathbf{a}}{\partial \mathbf{v}}\right|_{r=\hat{r}, v=\hat{v}} \\
\mathbf{Q}\,\delta(t-\tau) &= \mathrm{E}\left\{ \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\nu}(t) \\ -0.5\boldsymbol{\eta}_g(t) \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\nu}(\tau) \\ -0.5\boldsymbol{\eta}_g(\tau) \end{bmatrix}^{\mathrm{T}} \right\}.
\end{aligned}$$

The update is given by

$$
\begin{aligned}
\widehat{\mathbf{r}}^{+} &= \widehat{\mathbf{r}}^{-} + \mathbf{K}_r \boldsymbol{\epsilon} \\
\widehat{\mathbf{v}}^{+} &= \widehat{\mathbf{v}}^{-} + \mathbf{K}_v \boldsymbol{\epsilon} \\
\widehat{\overline{\mathbf{q}}}^{+} &= \widehat{\overline{\mathbf{q}}}^{-} \\
\mathbf{P}^{+} &= \mathbf{P}^{-} - \mathbf{K}\mathbf{W}\mathbf{K}^{\mathrm{T}},
\end{aligned}
$$

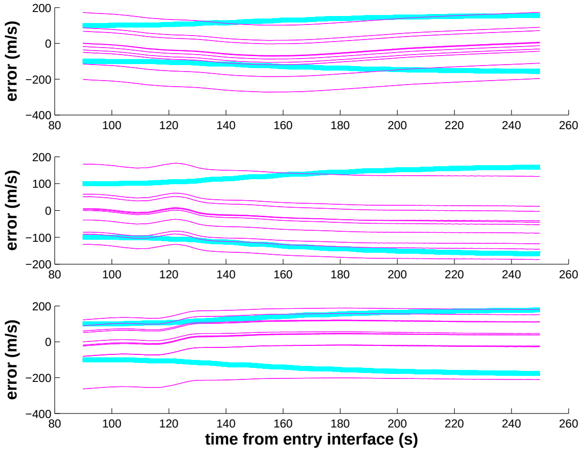
where

$$
\begin{aligned}
\mathbf{K} &= \begin{bmatrix} \mathbf{K}^{\star}(1:6, 1:3)^{\mathrm{T}} & \mathbf{O}_{3\times 3} \end{bmatrix}^{\mathrm{T}} \\
\mathbf{K}^{\star} &= \mathbf{P}^{-}\mathbf{H}^{\mathrm{T}}\mathbf{W}^{-1} \\
\mathbf{H} &= \begin{bmatrix} \mathbf{T}(\widehat{\overline{\mathbf{q}}})\mathbf{A}_r & \mathbf{T}(\widehat{\overline{\mathbf{q}}})\mathbf{A}_v & 2[\mathbf{T}(\widehat{\overline{\mathbf{q}}})\widehat{\mathbf{a}}\times] \end{bmatrix} \\
\mathbf{W} &= \mathbf{H}\mathbf{P}^{-}\mathbf{H}^{\mathrm{T}} + \mathbf{R} \\
\mathbf{R}(t_k)\delta_{k-j} &= \mathrm{E}\left\{ \boldsymbol{\eta}_a(t_k)\boldsymbol{\eta}_a(t_j)^{\mathrm{T}} \right\}.
\end{aligned}
$$

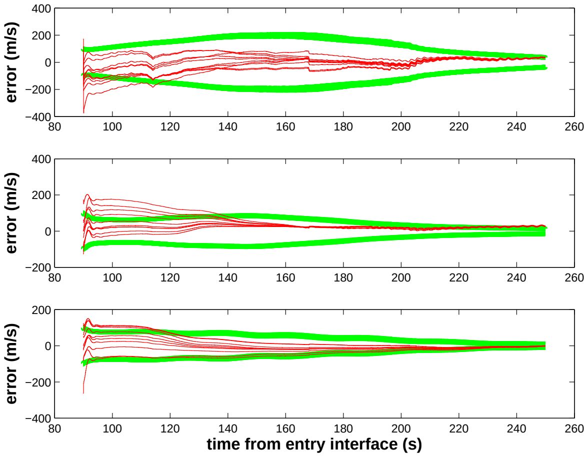
5.2.6 Simulation Results

Simulations show that filtering leads to a more accurate state estimate than dead-reckoning. Figure 5.1 shows the comparison between the EKF state estimates and the dead-reckoning. Every run is preformed with the same set of IMU data from the MER mission. The error history is obtained by comparing a rough best estimated trajectory (BET) to the EKF state estimate. Each run differs in the initial condition generated with a zero-mean, normal distribution whose covariance is diagonal and has standard deviation of 1000m in each position axes, 100m/s in each velocity axes, 3° in attitude, and mean equal to the initial state of the BET.

It can be noticed that the estimates of velocity are roughly parallel to each other. Dead-reckoning is an open-loop procedure, based on adding the measured



(a) Dead-Reckoning



(b) Filtering

Figure 5.1: Comparison of dead-reckoning and EKF filtering approach. Ten runs varying the initial conditions and using the same set of MER IMU measurements. The thick line is the standard deviation.

$\Delta \mathbf{v}_k$ to the previous velocity estimate. Since every run has the same measurement history, it is natural that the errors stay almost parallel to each other and therefore strongly depend on the initial condition. The filtering has a very different result. In an observable system, if the initial estimate is good enough not to violate the linearization assumption, the Kalman filter estimate should become independent of the initial condition. We can see that behavior in Figure 5.1. For each run the estimates converge, revealing that the filter successfully extracts information from the accelerometer, and based its estimate on those measurements (as well as on the model).

5.3 Gating Network

The gating network algorithm described in Section 3.3 was tested with the MER IMU measurements. The bank was composed of three filters, a filter with the nominal COSPAR air density model, a "high" filter with air density 10% higher than nominal, and a "low" filter with air density 10% lower. Figure 5.2 shows the three density models. Figure 5.3 shows the evolution of the gains in the bank, Figure 5.4 shows the total estimation error in position and velocity. In Figure 5.3, the gain associated with the filter modelling a "high" density situation is assigned the highest weight during the initial 180s, then a switch occurs and the filter modelling a "low" density situation is assigned the highest weight. Correspondingly, in Figure 5.4

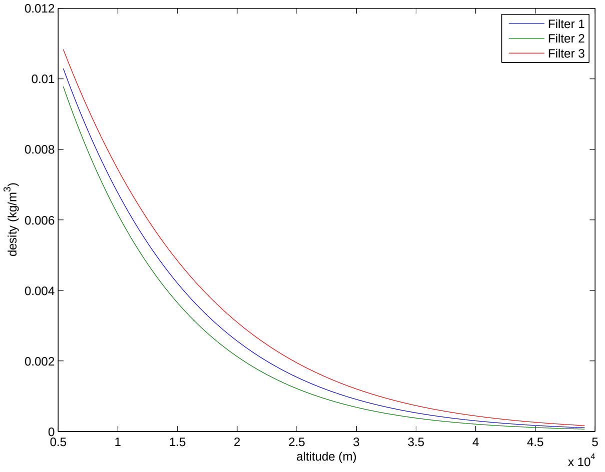


Figure 5.2: Air density of the three filters in the MMAE bank.

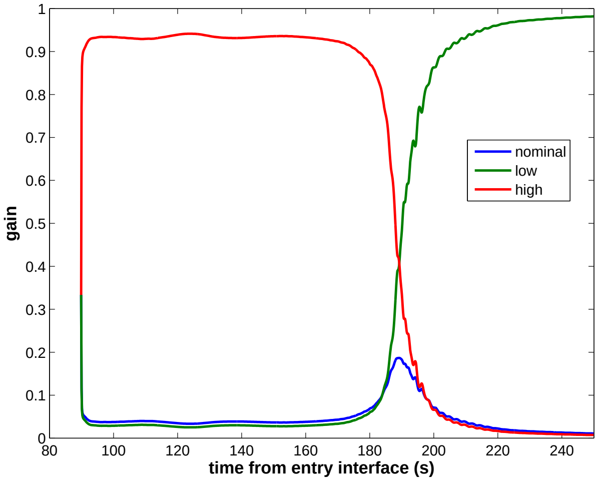


Figure 5.3: Gain evolution.

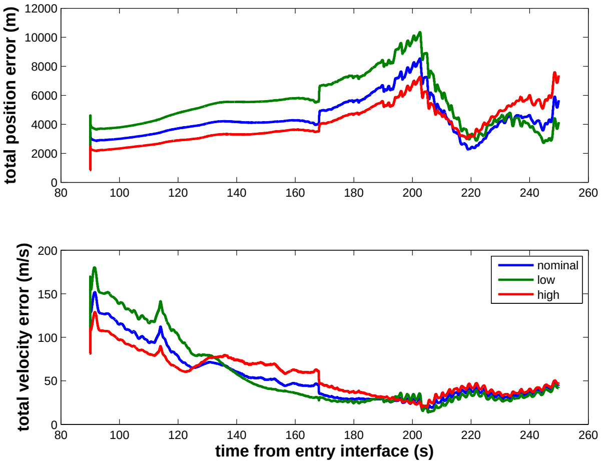


Figure 5.4: Total position and velocity errors of the three filters in the bank.

the position and velocity errors are smallest for the filter modelling the "high" density during the first 180s. After about 200s, the filter modelling the 'low" density produces the smallest estimation errors. This shows that the gating network is capable of correctly selecting the best filter among the ones implemented in the filter bank.

5.4 Conclusions

A modification of an existing multiple model adaptive estimator was successfully implemented for Mars EDL state estimation. It was shown that real flight IMU measurements can be processed as an external measurement in a model-based extended Kalman filter. Filtering the IMU measurements leads to a more accurate state estimate than the dead-reckoning approach.

These results are only indicative of the possibilities of using an EKF for processing the IMU data coupled with the MMAE architecture. This is not an exhaustive investigation, hence no general statements about the MER mission or the atmosphere encountered during the MER EDL should be made. First, the BET used was, in fact, not a final BET (that work is still underway), so that the state estimation errors shown in Figures 5.1 and 5.4 may not reflect the actual estimation errors. Second, and more importantly, the atmospheric density model used in the analysis (based on the COSPAR data) is likely not of sufficient complexity to accurately reflect the expected density variations at Mars. However the goals of this investigation were successfully met

1. The contribution of the attitude estimation error was successfully implemented
2. An adaptable filter implementing a very simple density model was designed to perform better than a dead-reckoning filter. Given that the “true” density is unknown, the premises of this approach are excellent.

Chapter 6

Lunar Descent Navigation

In Chapter 4, a detailed linear covariance analysis for the dead-reckoning approach was developed, and was followed by the filtering of the accelerometer in Chater 5. In this chapter, the gyro will be filtered in an EKF. The scenario chosen for this application is lunar descent. The choice is motivated by the desire to include attitude updates into the EKF. The absence of atmosphere on the Moon allows for the use of a star camera during descent. Star cameras normally have their own estimation algorithm and provide an attitude estimation rather than raw measurements. The technique developed in Section 3.1.1 will be used to fuse the star camera estimate into the Kalman filter.

The scenario considered here begins after the conclusion of the orbital phase when the descent trajectory begins. The descent trajectory is a thrust-coast-thrust. The propulsion system initiates the descent, followed be a no thrust phase. When a predeterminate altitude is reached, the propulsion system will be employed to land the spacecraft. The available sensors are an altimeter, a velocimeter, together with the previously mentioned IMU, star camera, and the IMU. The model of the IMU was presented in Section 4.3. The altimeter provides a measurement of altitude along the local vertical, and the velocimter measures relative velocity with respect to Moon surface. The models for these sensors, together with the true trajectory, are those used by DeMars [80]. Figures 6.1* and 6.2 show the true trajectory. Figure 6.3 shows the measurement times for each external sensor. The star camera model used in this work is presented next.

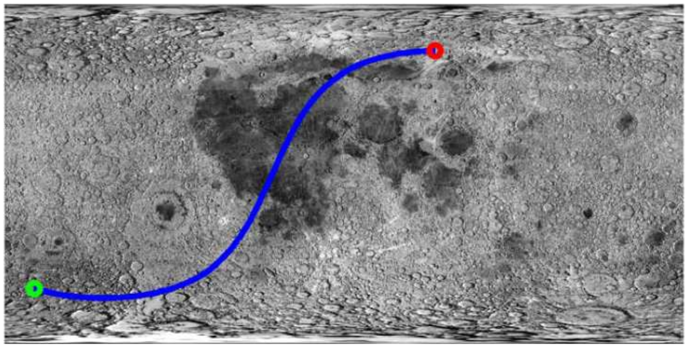


Figure 6.1: Groundtrack of lunar descent to landing trajectory. Green dot is the starting point, red dot the end point.

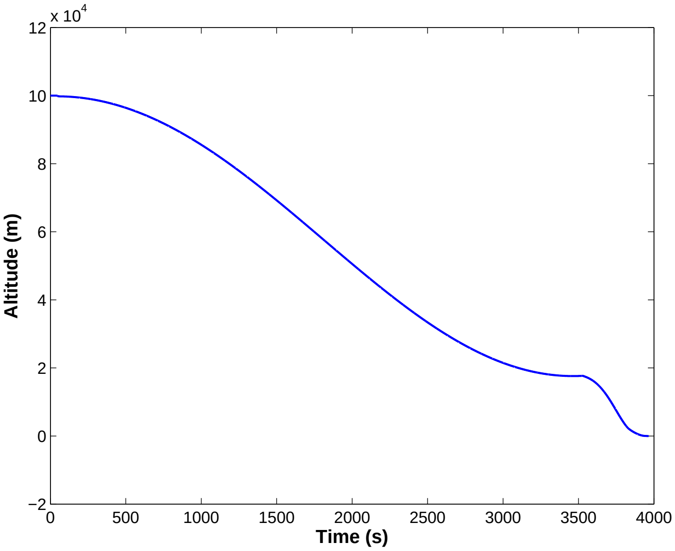


Figure 6.2: Altitude of lunar descent to landing trajectory..

---

*Figure courtesy of K. J. DeMars

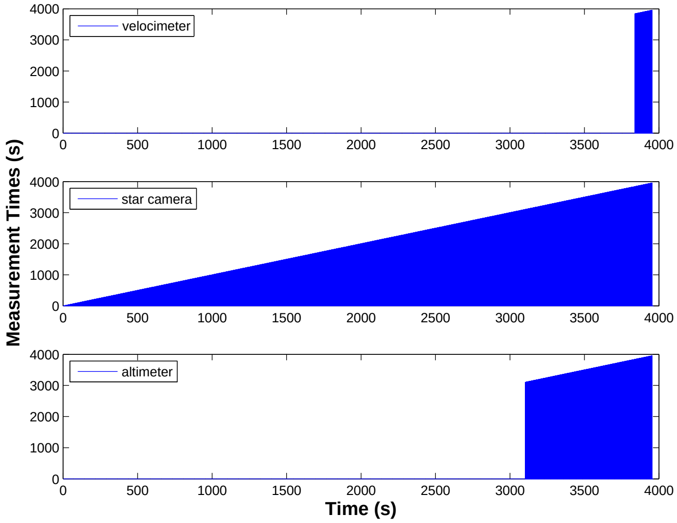


Figure 6.3: Times for external measurements.

6.1 Star Camera Model

The star camera model is based on the measurement of two angles α and β along two perpendicular directions, as shown in Figure 6.4. The reconstructed unit vector measurement for each star, is given by [13]

$$\mathbf{y}_0 = \frac{1}{\sqrt{1+\tan^2\alpha+\tan^2\beta}} \begin{bmatrix} \tan\alpha \\ \tan\beta \\ 1 \end{bmatrix}.$$

Perturbing the angles, it follows that

$$\mathbf{y}_= \frac{1}{\sqrt{1+\tan^2(\alpha+\delta\alpha)+\tan^2(\beta+\delta\beta)}} \begin{bmatrix} \tan(\alpha+\delta\alpha) \\ \tan(\beta+\delta\beta) \\ 1 \end{bmatrix}.$$

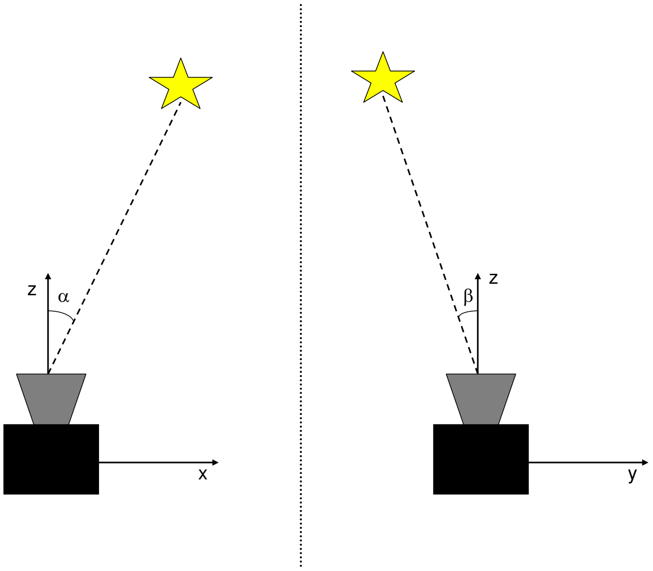


Figure 6.4: Star camera measured angles.

We assume perfect knowledge of the inertial position of the stars. This information is contained in a given star table. From the trigonometric identity

$$\tan(\alpha + \delta\alpha) = \frac{\tan\alpha + \tan\delta\alpha}{1 - \tan\alpha\tan\delta\alpha},$$

we obtain to first-order that

$$\tan(\alpha + \delta\alpha) \simeq \tan\alpha + (1 + \tan^2\alpha)\,\delta\alpha.$$

Hence $\mathbf{y}$ can be approximated to first-order as

$$\mathbf{y} \simeq \mathbf{y}_0 + \delta\mathbf{y},$$

where

$$\delta\mathbf{y} = \begin{bmatrix} \frac{1+\tan^2\beta}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} & -\frac{\tan\alpha\tan\beta}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} \\ -\frac{\tan\alpha\tan\beta}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} & \frac{1+\tan^2\alpha}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} \\ -\frac{\tan\alpha}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} & -\frac{\tan\beta}{(1+\tan^2\alpha+\tan^2\beta)^{3/2}} \end{bmatrix} \begin{bmatrix} (1+\tan^2\alpha)\delta\alpha \\ (1+\tan^2\beta)\delta\beta \end{bmatrix}.$$

It turns out that $y_{0,z} \neq 0$ since the field of view is necessary less than 180 degrees. Therefore, it follows that

$$\begin{aligned} \delta\mathbf{y} &= \frac{1}{y_{0,z}} \begin{bmatrix} y_{0,z}^2 + y_{0,y}^2 & -y_{0,x}y_{0,y} \\ -y_{0,x}y_{0,y} & y_{0,z}^2 + y_{0,x}^2 \\ -y_{0,x}y_{0,z} & -y_{0,y}y_{0,z} \end{bmatrix} \begin{bmatrix} (y_{0,z}^2 + y_{0,x}^2)\delta\alpha \\ (y_{0,z}^2 + y_{0,y}^2)\delta\beta \end{bmatrix} \\ &= \begin{bmatrix} y_{0,z}^2 + y_{0,y}^2 & -y_{0,x}y_{0,y} & -y_{0,x}y_{0,z} \\ -y_{0,x}y_{0,y} & y_{0,z}^2 + y_{0,x}^2 & -y_{0,y}y_{0,z} \\ -y_{0,x}y_{0,z} & -y_{0,y}y_{0,z} & y_{0,y}^2 + y_{0,x}^2 \end{bmatrix} \begin{bmatrix} \frac{y_{0,z}^2 + y_{0,x}^2}{y_{0,z}}\delta\alpha \\ \frac{y_{0,z}^2 + y_{0,y}^2}{y_{0,z}}\delta\beta \\ 0 \end{bmatrix} \\ &= -[\mathbf{y}_0\times]^2 \begin{bmatrix} \frac{y_{0,z}^2 + y_{0,x}^2}{y_{0,z}}\delta\alpha \\ \frac{y_{0,z}^2 + y_{0,y}^2}{y_{0,z}}\delta\beta \\ 0 \end{bmatrix} = -[\mathbf{y}_0\times]^2\,\mathbf{U}\,\delta\boldsymbol{\alpha}, \end{aligned}$$

where

$$\mathbf{U} = \begin{bmatrix} \frac{y_{0,z}^2 + y_{0,x}^2}{y_{0,z}} & 0 \\ 0 & \frac{y_{0,z}^2 + y_{0,y}^2}{y_{0,z}} \\ 0 & 0 \end{bmatrix}, \qquad \delta\boldsymbol{\alpha} = \begin{bmatrix} \delta\alpha \\ \delta\beta \end{bmatrix}.$$

Recall that the QUEST measurement model (§2.3) assumes

$$\delta\mathbf{y} = [\mathbf{y}_0\times]\delta\boldsymbol{\theta},$$

where

$$\mathbf{R}_{\theta\theta} \triangleq \mathrm{E}\left\{\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^{\mathrm{T}}\right\} = \sigma^2\mathbf{I}_{3\times3}.$$

The QUEST model can only be an approximation of the star camera model developed here since for this application

$$\mathbf{R}_{\theta\theta} = -[\mathbf{y}_0\times]\mathbf{U}\,\mathrm{E}\left\{\delta\boldsymbol{\alpha}\,\delta\boldsymbol{\alpha}^{\mathrm{T}}\right\}\mathbf{U}^{\mathrm{T}}[\mathbf{y}_0\times]. \tag{6.1}$$

The star camera employs the Davenport-q algorithm to estimate the quaternion, and calculates the associated covariance with Eq. (2.45) as

$$\mathbf{P}_{st} = 4\mathbf{M}_0^{-1}\sum_{i=1}^{n} w_i^2[\mathbf{y}_i\times]\mathbf{R}_{y,i}[\mathbf{y}_i\times]^{\mathrm{T}}\mathbf{M}_0^{-\mathrm{T}}, \tag{6.2}$$

where

$$\widehat{\mathbf{T}} = \mathbf{T}(\widehat{\overline{\mathbf{q}}}), \qquad \mathbf{M}_0 = -2\sum_{i=1}^{n} w_i[\mathbf{y}_i\times]^2,$$

and $\mathbf{R}_{y,i}$ are calculated for each star using Eq. (6.1) and

$$\mathbf{R}_y = -[\mathbf{y}_0\times]\mathbf{R}_{\theta\theta}[\mathbf{y}_0\times].$$

In calculating the covariance, $\mathbf{y}_0$ needs to be replaced by $\mathbf{y}$.

6.2 Extended Kalman Filter

This section introduces the Kalman filter used to process the altimeter, velocimeter, and gyro measurements, together with the quaternion "measurement" provided by the star camera. The accelerometer is dead-reckoned and used to propagate the state vector.

6.2.1 Propagation

The state propagation consists of numerically integrating the model

$$\frac{d}{dt}\begin{bmatrix}\widehat{\mathbf{r}}\\ \widehat{\mathbf{v}}\\ \widehat{\overline{\mathbf{q}}}\\ \widehat{\boldsymbol{\omega}}\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{v}}\\ \mathbf{g}(\widehat{\mathbf{r}}) + \mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}}\mathbf{a}_m\\ \frac{1}{2}\widehat{\overline{\boldsymbol{\omega}}} \otimes \widehat{\overline{\mathbf{q}}}\\ \mathbf{J}^{-1}\left(\widehat{\boldsymbol{\omega}} \times\ \mathbf{J}\widehat{\boldsymbol{\omega}} + \widehat{\mathbf{m}}\right)\end{bmatrix},$$

where $\widehat{\mathbf{m}}$ is the torque to be applied given by the control system. The components of the estimation error are defined as

$$\mathbf{e}_r \triangleq \mathbf{r} - \widehat{\mathbf{r}}, \quad \mathbf{e}_v \triangleq \mathbf{v} - \widehat{\mathbf{v}}, \quad \mathbf{e}_\theta \triangleq 2\delta\mathbf{q}, \quad \mathbf{e}_\omega \triangleq \boldsymbol{\omega} - \widehat{\boldsymbol{\omega}},$$

where $\delta\overline{\mathbf{q}} \triangleq \overline{\mathbf{q}} \otimes \widehat{\overline{\mathbf{q}}}^{-1}$. The first-order approximation of the evolution of the velocity and attitude components of the error is

$$\begin{aligned}\dot{\mathbf{e}}_v &= \mathbf{G}(\widehat{\mathbf{r}})\mathbf{e}_r + \mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}}[\mathbf{e}_\theta\times]\mathbf{a}_m - \mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}}\boldsymbol{\eta}_a\\ \dot{\mathbf{e}}_\theta &= -[\widehat{\boldsymbol{\omega}}\times]\mathbf{e}_\theta + \mathbf{e}_\omega.\end{aligned}$$

The difference between the modeled torque $\widehat{\mathbf{m}}$ and the actual applied torque $\mathbf{m}$, is treated as process noise $\delta\mathbf{m} \triangleq \mathbf{m} - \widehat{\mathbf{m}}$, from which it follows that assuming perfectly known inertia matrix

$$\dot{\mathbf{e}}_\omega = \mathbf{J}^{-1}[\mathbf{e}_\omega\times]\mathbf{J}\widehat{\boldsymbol{\omega}} + \mathbf{J}^{-1}[\widehat{\boldsymbol{\omega}}\times]\mathbf{J}\mathbf{e}_\omega + \mathbf{J}^{-1}\delta\mathbf{m}.$$

The evolution of the estimation error can then be written in compact matrix form

$$\dot{\mathbf{e}} = \mathbf{F}\mathbf{e} + \boldsymbol{\nu},$$

where

$$\mathbf{F} \triangleq \begin{bmatrix} \mathbf{O}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} \\ \mathbf{G}(\widehat{\mathbf{r}}) & \mathbf{O}_{3\times3} & -\mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}}[\mathbf{a}_m\times] & \mathbf{O}_{3\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & -[\widehat{\boldsymbol{\omega}}\times] & \mathbf{I}_{3\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{J}^{-1}\left\{-[\mathbf{J}\widehat{\boldsymbol{\omega}}\times]+[\widehat{\boldsymbol{\omega}}\times]\mathbf{J}\right\} \end{bmatrix},$$

and

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{T}(\widehat{\overline{\mathbf{q}}})^{\mathrm{T}}\boldsymbol{\eta}_a \\ \mathbf{0} \\ \mathbf{J}^{-1}\delta\mathbf{m} \end{bmatrix} \qquad \mathrm{E}\left\{\boldsymbol{\nu}(t)\right\} = \mathbf{0} \qquad \mathrm{E}\left\{\boldsymbol{\nu}(t)\,\boldsymbol{\nu}^{\mathrm{T}}(\tau)\right\} = \mathbf{Q}(t)\delta(t-\tau).$$

Between measurements, the covariance propagation is given by the continuous-time Riccati equation

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\widehat{\mathbf{x}})\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(\widehat{\mathbf{x}})^{\mathrm{T}} + \mathbf{Q}(t).$$

where $\mathbf{P}(t) = \mathrm{E}\left\{\mathbf{e}\,\mathbf{e}^{\mathrm{T}}\right\}$.

6.2.2 Update

The state update using altimeter and velocimeter is a standard Kalman filter application, and its derivation will not be repeated here because a complete presentation can be found in [80]. This work focuses on the inclusion of the star camera measurement, which is derived here.

The star tracker provides an estimate of the quaternion $\widehat{\overline{\mathbf{q}}}_{st}$ and the associated estimation uncertainty given by the small angle covariance $\mathbf{P}_{st}$. Since the star tracker formulates its estimates based only on the current measurements, it was shown in Section 3.1.1 that is optimal to treat the star tracker as a measurement and its covariance as measurement noise. However, following the multiplicative approach of Section 3.1.2, deviations from rather than quaternions are used. The processed measurement is twice the vector part of the deviation between the "measured" quaternion $\widehat{\widehat{\mathbf{q}}}_{st}$ and the nominal quaternion $\widehat{\widehat{\mathbf{q}}}^{\,-}$ at time t_k. The deviation is given by

$$\overline{\mathbf{y}}_{st,k} = \widehat{\widehat{\mathbf{q}}}_{st,k} \otimes (\widehat{\widehat{\mathbf{q}}}_k^{\,-})^{-1}.$$

The estimated measurement is zero, therefore the star tracker residual $\boldsymbol{\epsilon}_{st,k}$ is

$$\boldsymbol{\epsilon}_{st,k} = 2\mathbf{y}_{st,k}.$$

The state vector for update purposes is given by

$$\mathbf{x}_k^{\mathrm{T}} = \begin{bmatrix} \widehat{\mathbf{r}}_k^{\mathrm{T}} & \widehat{\mathbf{v}}_k^{\mathrm{T}} & \delta\widehat{\boldsymbol{\theta}}_k^{\mathrm{T}} & \widehat{\boldsymbol{\omega}}_k^{\mathrm{T}} \end{bmatrix}, \qquad \delta\widehat{\boldsymbol{\theta}}_k^{-} = \mathbf{0}.$$

The star tracker measurement mapping matrix is,

$$\mathbf{H}_{st,k} = \begin{bmatrix} \mathbf{O}_{3\times 6} & \mathbf{I}_{3\times 3} & \mathbf{O}_{3\times 3} \end{bmatrix}.$$

If other measurements are available at time t_k, they would be included as

$$\boldsymbol{\epsilon}_k = \begin{bmatrix} \boldsymbol{\epsilon}_{st,k} \\ \boldsymbol{\epsilon}_{others,k} \end{bmatrix} \qquad \mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{st,k} \\ \mathbf{H}_{others,k} \end{bmatrix} \qquad \mathbf{R}_k = \begin{bmatrix} \mathbf{P}_{st,k} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{others,k} \end{bmatrix},$$

assuming the other measurements are uncorrelated to the star camera's.

The residuals covariance is given by

$$\mathbf{W}_k = \mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k,$$

the standard Kalman filter update is

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^{\mathrm{T}} \mathbf{W}_k^{-1}, \qquad \widehat{\mathbf{x}}_k^{+} = \widehat{\mathbf{x}}_k^{-} + \mathbf{K}_k \boldsymbol{\epsilon}_k, \qquad \mathbf{P}_k^{+} = \mathbf{P}_k^{-} - \mathbf{K}_k \mathbf{W}_k \mathbf{K}_k^{\mathrm{T}}.$$

The quaternion update is

$$\widehat{\overline{\mathbf{q}}}^{\,+} = \begin{bmatrix} \delta\widehat{\boldsymbol{\theta}}^{+} \\ 1 \end{bmatrix} \otimes \widehat{\overline{\mathbf{q}}}^{\,-},$$

followed by the normalization to restore the unit norm constraint.

NOTE The quaternion is a two to one representation of the attitude, $\overline{\mathbf{q}}$ and $-\overline{\mathbf{q}}$ represent the same rotation. However in processing a quaternion measurement in a EKF, is very important to distinguish between the two. Assume the *a priori* estimate of the attitude is given by

$$\widehat{\overline{\mathbf{q}}}^{\,-} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}},$$

processing a quaternion measurement given by

$$\overline{\mathbf{y}} = \begin{bmatrix} 0 & 0 & \sqrt{1 - .99^2} & -.99 \end{bmatrix}^{\mathrm{T}},$$

would result in an update in the wrong direction. In those situations, it is therefore necessary to process $-\overline{\mathbf{y}}$ in stead of $\overline{\mathbf{y}}$, or, equivalently, process $\overline{\mathbf{y}}$ and update the estimated quaternion with

$$\widehat{\overline{\mathbf{q}}}^{\,+} = \begin{bmatrix} \delta\widehat{\boldsymbol{\theta}}^{+} \\ -1 \end{bmatrix} \otimes \widehat{\overline{\mathbf{q}}}^{\,-}.$$

6.3 Estimation Error Results

In this section the results of a single run will be presented. The analysis shows that the filter is stable for the nominal descent. The estimation error (shown in red) and the filter standard deviation (shown in blue) are shown as functions of time. Table 6.1 show the standard deviations of the measurement errors generated in the simulation.

Accelerometer Noise	10 $[\mu g\sqrt{s}]$
Accelerometer Bias	0.1 $[mg]$
Accelerometer Scale Factor	175 $[ppm]$
Accelerometer Misalignment	5 $[arcsec]$
Gyro Noise	0.01 $[deg/\sqrt{hr}]$
Gyro Bias	0.05 $[deg/hr]$
Gyro Scale Factor	5 $[ppm]$
Gyro Misalignment	5 $[arcsec]$
Altimeter Noise	10 $[m]$
Altimeter Bias	0.5 $[m]$
Velocimeter Noise	0.5 $[m/s]$
Velocimeter Bias	0.5 $[m]$
Star Camera Noise	50 $[arcsec]$
Torque Noise	0.1 $[Nm]$

Table 6.1: Random error standard deviation values

Figures 6.5–6.8 show the evolution of the states, while Figure 6.9 shows the star camera estimate and covariance. The filter covariance clearly shows the time at which the altimeter starts providing measurements (approximately 3200 seconds), and the time at which the velocimeter start providing measurements (approximately 3800 seconds). It can be also noticed that the y component of position is not very observable. This fact is due to its orientation perpendicular to the trajectory. In order to make the y component of position more observable, it is necessary to introduce an additional measurement related to position, like a range measurement (in an appropriate direction) or a full three dimensional position measurement deduced from a terrain camera. It is also noticeable that the angular velocity estimation error reduces to the gyro noise. This fact occurs because of the high frequency of the data and accuracy of the gyro.

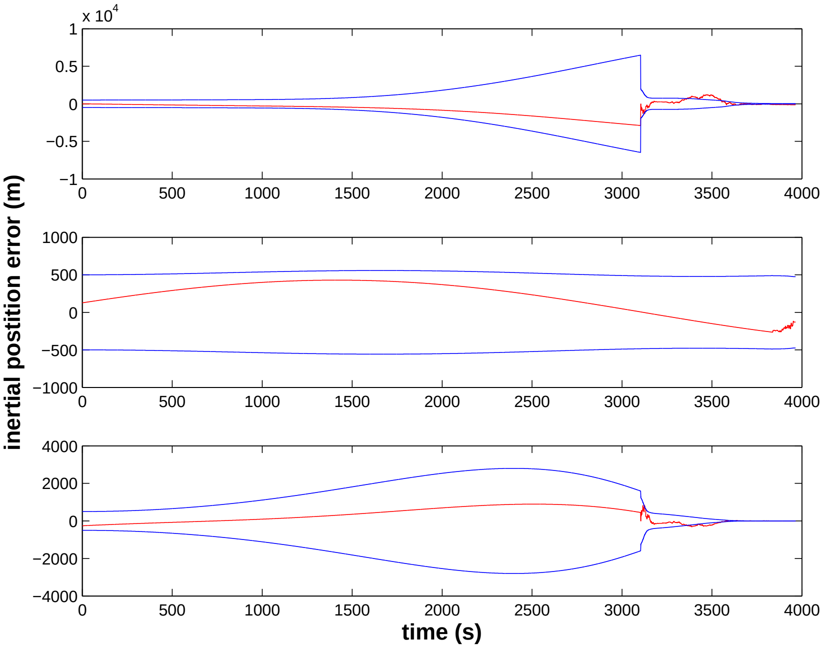


Figure 6.5: Position estimation error.

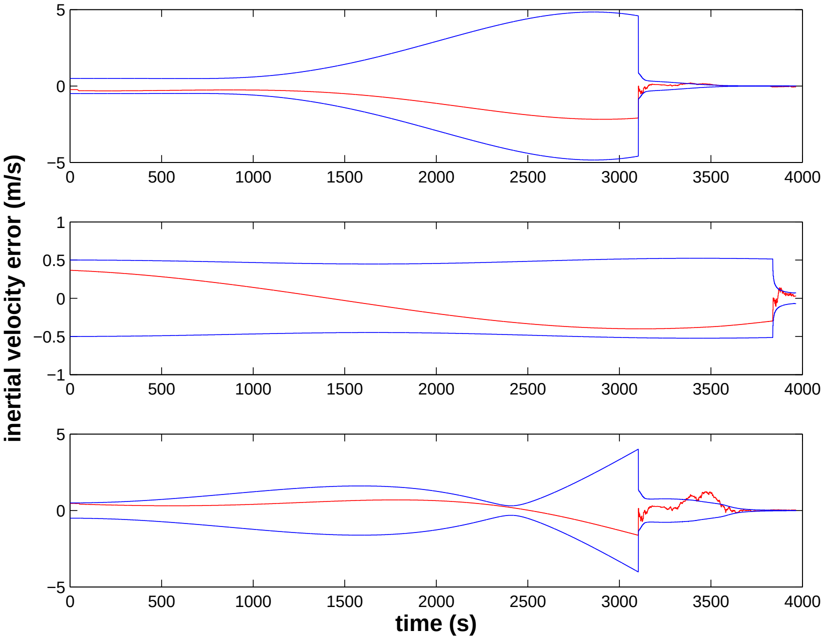


Figure 6.6: Velocity estimation error.

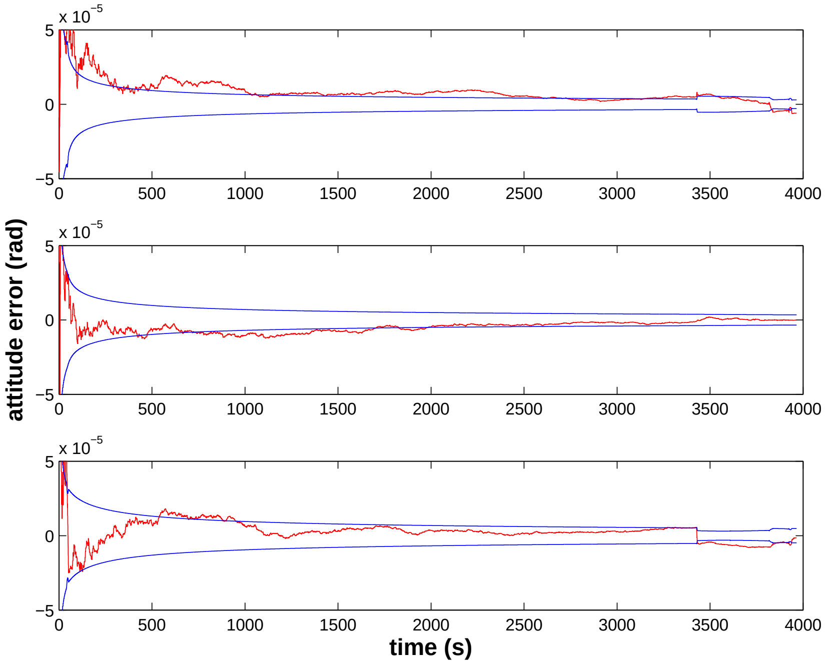


Figure 6.7: Attitude estimation error.

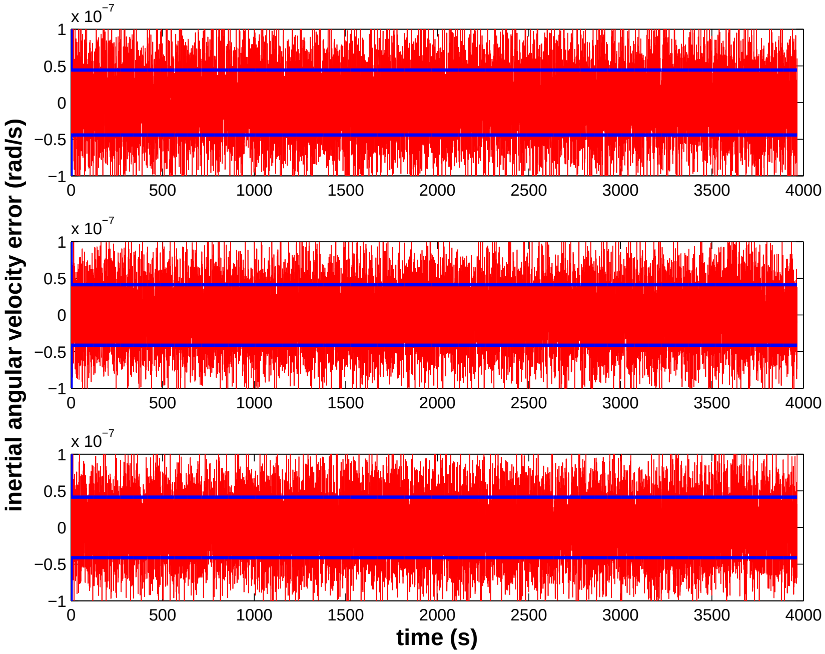


Figure 6.8: Angular velocity estimation error.

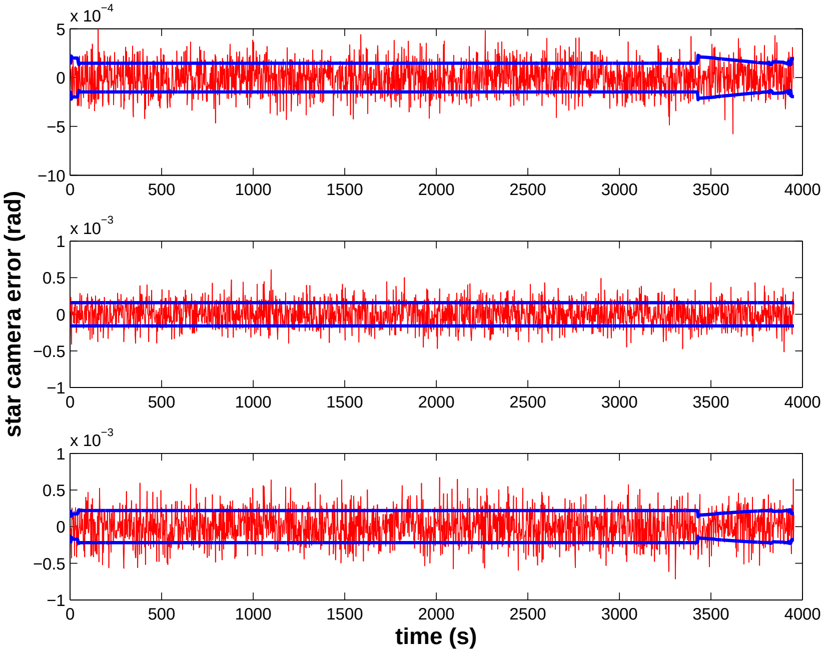


Figure 6.9: Star camera estimation error.

6.4 Monte Carlo Analysis

In order to validate a filter design, is not sufficient to analyze a single run. It is necessary to perform Monte Carlo analysis to confirm that the statistical properties of the estimation error are appropriately represented by the filter covariance. Figures 6.10–6.13 plot the filter covariance (black line) and the sample covariance from 100 runs (blue line). Each run implements different initial estimation error and measurement errors. Figure 6.14 shows the Monte Carlo analysis of the star camera estimate and covariance.

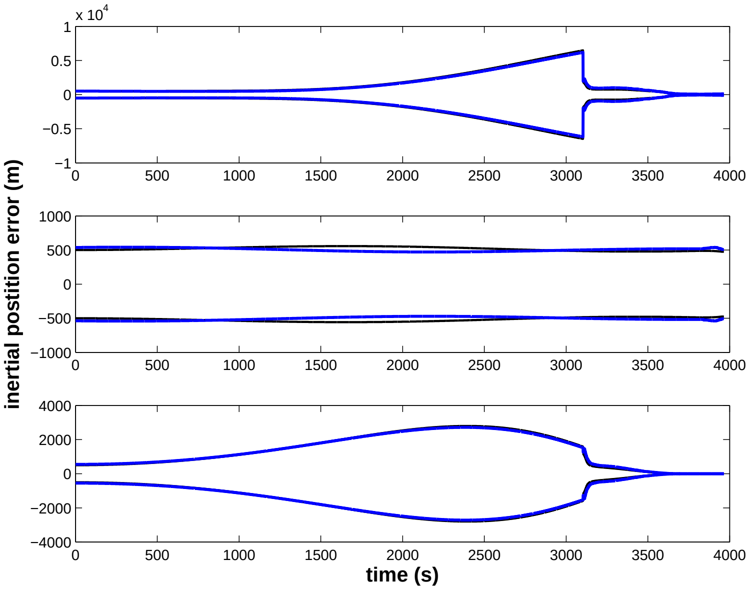


Figure 6.10: Monte Carlo analysis of position estimation error.

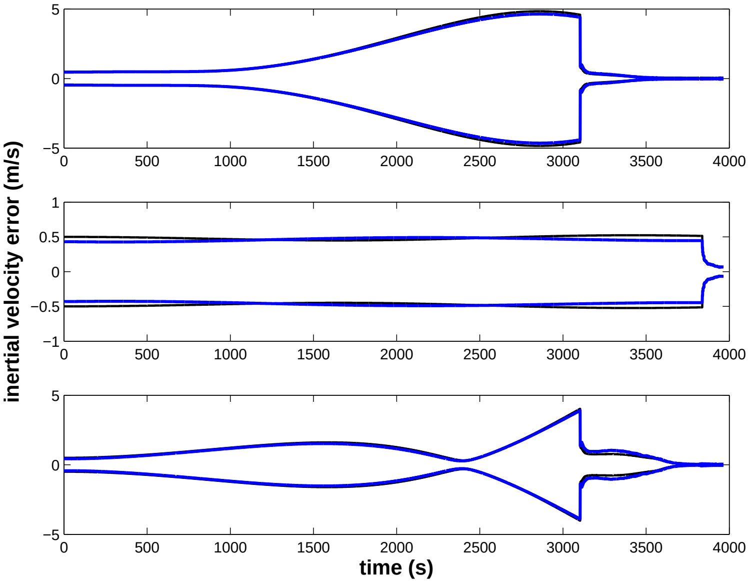


Figure 6.11: Monte Carlo analysis of velocity estimation error.

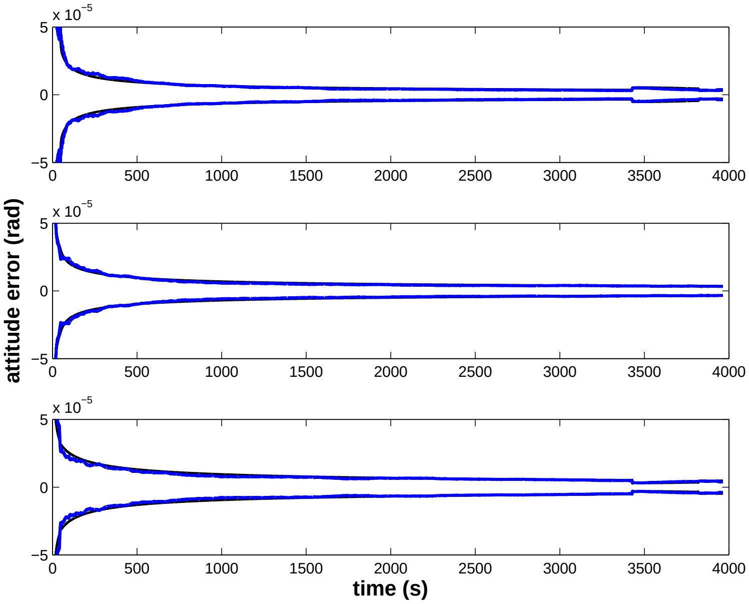


Figure 6.12: Monte Carlo analysis of attitude estimation error.

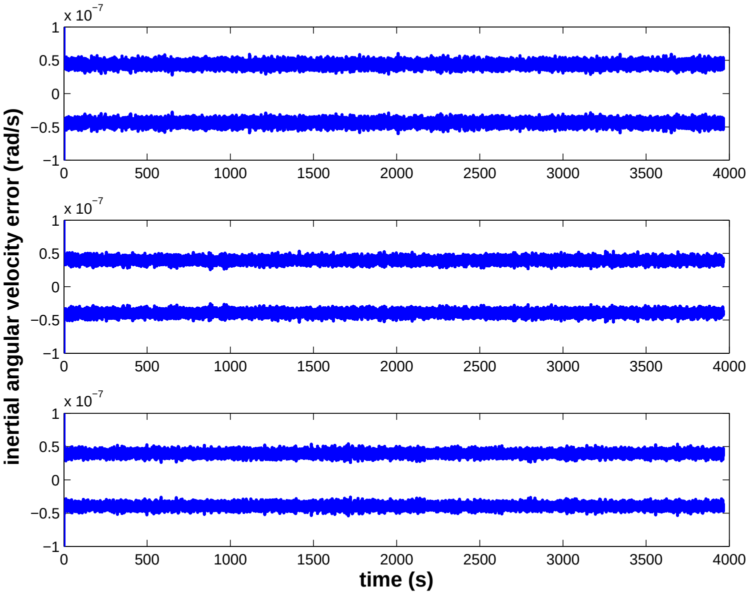


Figure 6.13: Monte Carlo analysis of angular velocity estimation error.

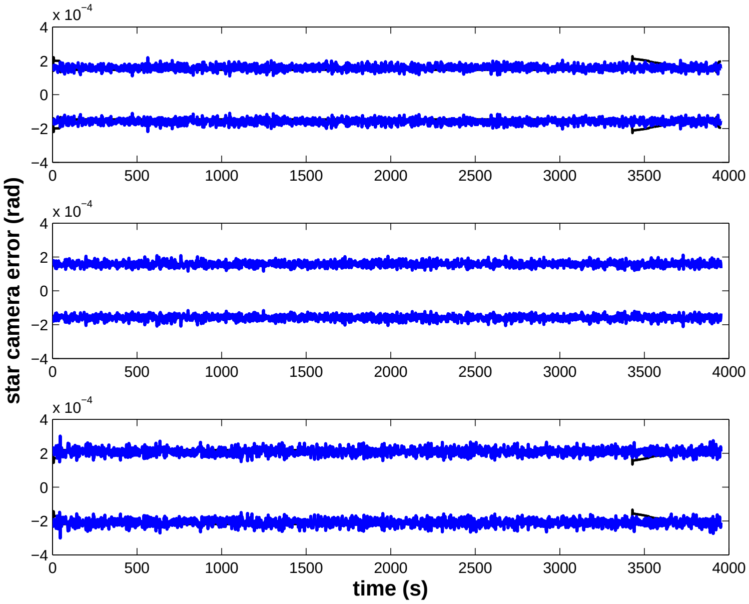


Figure 6.14: Monte Carlo analysis of star camera estimation error.

Chapter 7

Conclusions

A study of precision navigation to support landing on celestial bodies was performed. Particular attention was given to the inclusion of the inertial measurement unit and attitude estimation into the Kalman filter. Examples in which the accelerometer and gyro were successfully used to update the filter were developed. A detailed comparison of the two most used techniques to introduce quaternion estimation into the Kalman filter was performed. The processing of attitude estimates as if they were raw measurements was analyzed, circumstances in which this approach leads to optimal versus sub-optimal estimates were presented. It was found that a star camera can be optimally fused into a Kalman filter and the results were applied to a problem of much current interest, Moon descent to landing navigation. The classical Davenport q-algorithm was discussed. This algorithm is to be used by the star camera to produce its estimate. In order to be optimally coupled with the Kalman filter, the star camera needs to provide together with the estimate of the quaternion, an error covariance. A new covariance formulation was introduced and utilized in the example.

A novel way of introducing random biases in the Kalman filter was also derived and applied to a detailed linear covariance analysis for Mars dead-reckoning navigation. A modification of an existing adaptable Kalman filter was made, this modification was tested in the contest of Mars entry navigation. The three examples provided were used to validate the theoretical contributions by means of statistical analysis through the use of Monte Carlo runs.

Humans have already landed unguided probes to Mars and themselves to the Moon. The next step in advancing distant planets exploration is to be able to land very closely to where desired. This work suggests ways that aim to improve the navigation system for precision landing missions. Filtering the IMU can lead to a reduction of the state's uncertainty, while dead-reckoning the IMU inevitably increases the estimation error covariance. In the absence of external measurements, if the designer chooses to propagate the state with the IMU, accounting for the uncompensated biases leads to a better estimate of the uncertainty. A correct knowledge of the uncertainty is crucial to drive the control decisions that are necessary to achieve pin-point landing. It is fundamental to correctly account for the effects of the attitude error on the navigation filter and the correlation between the two. All these aspects have been presented in dept in this dissertation.

7.1 Future Directions

The conceptual design of algorithms is only the first step towards the implementation of a reliable precise navigation system. The examples in this dissertation provide insight of the benefits of filtering the IMU and correctly incorporate the attitude into the navigation system. High-fidelity simulations, development of more precise models, are necessary to take this theoretical study one step forward towards implementation. More work can be done to overcame the sub-optimality of some of the attitude sub-filter architectures. For example, some measurements could input both the attitude filter and the navigation filter, this approach would necessitate a careful analysis of the resulting correlation.

This work showed the possibilities of an adaptive estimator to overcame the modeling uncertainties of Mars atmosphere. Before a similar solution can be implemented more work needs to be done. Issues like stability and convergence of the adaptable scheme need to be more closely considered.

Bibliography

[1] Rudolf E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, pages 35–45, March 1960.

[2] Rudolf E. Kalman and Richard S. Bucy. New Results in Linear Filtering and Prediction. *Journal of Basic Engineering*, pages 83–95, 1961.

[3] Rudolf E. Kalman. New Methods and Results in Linear Prediction and Filtering Theory. In *Proceedings of the Symposium on Engineering Applications of Random Functions Theory and Probability*, New York, 1961.

[4] Arthur Gelb, editor. *Applied Optimal Estimation*. The MIT press Massachusetts Institute of Technology, Cambridge, MA, 1996.

[5] John Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430, Oct 1964.

[6] Jack B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton, NJ, 1999.

[7] Malcolm D. Shuster. Constraint in attitude estimation: Part I constrained estimation. *The Journal of the Astronautical Sciences*, 51(1):51–74, January-March 2003.

[8] Malcolm D. Shuster. Constraint in attitude estimation: Part II unconstrained

estimation. *The Journal of the Astronautical Sciences*, 51(1):75–101, January-March 2003.

[9] Itzhack Y. Bar-Itzhack and Y. Oshman. Attitude Determination from Vector Observations: Quaternion Estimation. *IEEE Transaction on Aerospace and Electronic Systems*, 21(1):128–135, January 1985.

[10] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. Applied Mathematics and Nonlinear Science. Chapman & Hall/VRV, first edition, 2004.

[11] Mark Pittelkau. An analysis of the quaternion attitude determination filter. *Journal of the Astronautical Sciences*, 51(1):103–120, January-March 2003.

[12] James E. Keat. Analysis of Least-Squares Attittude Determination Routine DOAOP. Technical Report CSC/TM-77/6034, Computer Sciences Corporation, February 1977.

[13] Malcolm D. Shuster. Maximum likelihood estimation of spacecraft attitude. *The Journal of the Astronautical Sciences*, 37(1):79–88, January-March 1989.

[14] James Richard Wertz, editor. *Spacecraft Attitude Determination and Control*. Springer, 1978.

[15] J. Thienel and R. M. Sanner. A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise. *IEEE Transactions on Automatic Control*, 48(11):2011–2015, November 2003.

[16] Maruthi R. Akella, Dongeun Seo, and Renato Zanetti. Actracting Manifolds for Attitude Estimation in Flatland and Otherlands. *to appear in: The Journal of the Astronautical Sciences*.

[17] Mark L. Psiaki. Attitude-determination filtering via extended quaternion estimation. *Journal of Guidance Control and Dynamics*, 23(2):206–214, March-April 2000.

[18] Martin C. Heyne and Robert H. Bishop. Spacecraft Entry Navigation using Sigma Point Kalman Filtering. In *Proceedings of IEEE/ION Position, Location and Navigation Symposium*, pages 71–79, Coronado, CA, April 2006.

[19] Olivier Dubois-Matra and Robert H. Bishop. Multi-model Navigation with Gating Networks for Mars Entry Precision Landing. In *AIAA Atmospheric Flight Mechanics Conference*, Providence RI, August 2004.

[20] D. T. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Systems, Man and Cybernetics*, AC-10:434–439, October 1965.

[21] H. A. P. Blom. An Efficient Filter for Abruptly Changing Systems. In *Proceedings 23rd IEEE Conference on Decision and Control*, 1984.

[22] P. S. Maybeck and K. P. Hentz. Investigation of moving-bank multiple model adaptive algorithma. *AIAA Journal of Guidance, Control, and Dynamics*, 10(1):90–96, January-February 1978.

[23] P. S. Maybeck and P. D. Hanlon. Performance enhancement of a multiple model adaptive estimator. In *Proceedings of the 32nd Conference on Decision and Control*, 1993.

[24] M. J. Caputi. A Necessary Condition for Effective Performance of the Multiple Model Adaptive Estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 31(3):1132–1138, July 1995.

[25] J. A. Gustafson and P. S. Mayback. Flexible Spacestructures Control Via

Moving-Bank Multiple Model Algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):750–, July 1994.

[26] P. Eide and P. Maybeck. An MMAE Failure Detection System for the F-16. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3):1125–1136, July 1996.

[27] W. D. Blair and Y. Bar-Shalom. Traking Maneuvering Targets With Multiple Sensors: Does More Data Always Mean Better Estimates? *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):450–456, January 1996.

[28] H. A. P. Blom and Y. Bar-Shalom. The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

[29] R. A. Jacobs and M. I. Jordan. A competitive modular connectionist architecture. *Advances in Neural Information Processing Systems 3*, pages 767–773, 1991.

[30] R. A. Jacobs and M. I. Jordan. Hierarchies of adaptive experts. *Advances in Neural Information Processing Systems 3*, pages 985–992, 1992.

[31] Wassim S. Chaer, Robert H. Bishop, and Joydeep Ghosh. A Mixture-of-Experts Framework for Adaptive Kalman Filtering. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 27(3):452–464, 1997.

[32] Wassim S. Chaer, Robert H. Bishop, and Joydeep Ghosh. Hierarchical Adaptive Kalman Filtering for Interplanetary Orbit Determination. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):883–896, July 1998.

[33] Leonhard Euler. Formulae generales pro translatione quacunque corporum rigidorum. *Novi Commentarii academiae scientiarum Petropolitanae*, 20:189–207, 1776. E478.

[34] Malcolm D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

[35] Leonhard Euler. Nova methodus motum corporum rigidorum degerminandi. *Novi Commentarii academiae scientiarum Petropolitanae*, 20:208–238, 1776. E479.

[36] Leonhard Euler. Problema algebraicum ob affectiones prorsus singulares memorabile. *Novi Commentarii academiae scientiarum Petropolitanae*, 15:75–106, 1771. E407.

[37] Leonhard Euler. De motu corporum circa punctum fixum mobilium. *Opera Postuma 2*, pages 43–62, 1862. E825.

[38] Benjamin Olinde Rodrigues. Des lois géométriques qui régissent les déplacements dun systéme solide dans l'espace, et de la variation des coordonnées provenant de ses déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 5:380–440, 1840.

[39] Heinz Hopf. Systeme symmetrischer Bilinearformen und euklidische Modelle der projektiven Räume. In *Vierteljahrsschrift der Naturforsch Gesellschaft in Zürich*, volume 85, pages 165–177, 1940.

[40] Simon L. Altmann. Hamilton, Rodriguez, and the Quaternion Scandal. *Mathematics Magazine*, 62(5):291–308, December 1989.

[41] Arthur Cayley. On certain results relating to quaternions. *Philosophical Magazine*, 26:141–145, 1845.

[42] Arthur Cayley. On the Motion of Rotation of a Solid Body. *Cambridge Mathematics Journal*, 3:224–232, 1843.

[43] John E. Borts. A New Mathematical Formulation for Strapdown Inertial Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 7(1):61–66, January 1971.

[44] Leonhard Euler. *Scientia navalis*, volume 1-2. 1749. E110-E111.

[45] Leonhard Euler. Decouverte d'un nouveau principe de Mecanique. *Mémoires de l'acadmie des sciences de Berlin*, 6:185–217, 1752. E177.

[46] Leonhard Euler. Du mouvement de rotation des corps solides autour d'un axe variable. *Mémoires de l'acadmie des sciences de Berlin*, 14:154–193, 1765. E292.

[47] Euler Archive. http://www.math.dartmouth.edu/ euler/.

[48] Hui Cheng and K. C. Gupta. An Historical Note on Finite Rotations. *Journal of Applied Mechanics*, 56(1):139–145, March 1989.

[49] Sandro Caparrini. The Discovery of the Vector Representation of Moments and Angular Velocity. *Archives for History of Exact Sciences*, 56(2):151–181, January 2002.

[50] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman Filtering for Spacecraft Attitude Estimation. *AIAA Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.

[51] Itzhack Y. Bar-Itzhack and Julie K. Thienel. On The Singularity In The Estimation Of The Quaternion-Of-Rotation. In *Guidance Navigation and Control Conference and Exhibit*, Monterrey, CA, August 2002. AIAA. Paper No. AIAA-2002-4831.

[52] Itzhack Y. Bar-Itzhack. Optimum Normalization of a Computed Quaternion of Rotation. *IEEE Transaction on Aerospace and Electronic Systems*, 7(2):401–402, March 1971.

[53] Renato Zanetti and Robert H. Bishop. Quaternion estimation and norm constrained kalman filtering. In *AIAA/AAS Astrodynamics Specialist Conference*, August 21–24 2006.

[54] P. W. Richards. Constrained kalman filtering using pseudo-measurements. In *IEE Colloquium on Algorithms for Target Tracking*, pages 75–79, 16 May 1995.

[55] Itzhack Y. Bar-Itzhack, Julie Deutschmann, and F. Landis Markley. Quaternion Normalization in Adaptive EKF for Spacecraft Attitude Determination. In *AIAA Guidance, Navigation and Control Conference*, New Orleans, Lousiana, August 1991.

[56] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Automatic Computation. Prentice-Hall Inc., Englewood Cliffs, New Jersey, first edition, 1974.

[57] Dan Simon and Tien Li Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):128–136, January 2002.

[58] V. A. Ferraresi. Utilizacão conjunta de sensores inerciais e não-inerciais em determinação de attitude de satélitesvia filtre de kalman. Technical Report INPE-4313-TDL/280, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil, August 1987.

[59] Malcolm D. Shuster. The quaternion in the kalman filter. *Advances in the Astronautical Sciences*, 85:25–37, October-December 1993.

[60] D. Choukroun, Itzhack Y. Bar-Itzhack, and Y. Oshman. A Novel Quaternion Kalman Filter. In *AIAA Guidance Navigation and Control Conference and Exhibit*, 2002.

[61] Grace Wahba. A least square estimate if satellite attitude. *SIAM Review*, 7(3):409, July 1965. Problem 65-1.

[62] Malcolm D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, 1981.

[63] Olivier Dubois-Matra. *Development of Multisensor Fusion Techniques with Gating Networks Applied to Reentry Vehicles*. PhD thesis, The University of Texas at Austin, May 2003.

[64] James R. Carpenter and Robert H. Bishop. Estimate Fusion for Lunar Rendezvous. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, volume 1, pages 1–10, Washington, DC, 1993. AIAA.

[65] Alan S. Willsky, Martin G. Bello, David A. Castanon, Bernard C. Levy, and George C. Verghese. Combining and updating of local estimates and regional maps along sets of one-dimensional tracks. *IEEE Transactions on Automatic Control*, AC-27(4):799–813, 1982.

[66] Neal A. Carlson. Federated Square Root Filter for Decentralized Parallel Processes. *IEEE Transactions on Aerospace and Electronic Systems*, 26(3):517–525, May 1989.

[67] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction To Random Signals And Applied Kalman Filtering*. John Wiley and Sons, third edition, 1997.

[68] Malcolm D. Shuster. The generalized wahba problem. *The Journal of the Astronautical Sciences*, 54(2):245–259, April-June 2007.

[69] Bernard Friedland. Treatment of Bias in Recursive Filtering. *IEEE Transactions on Automatic Control*, 14(4):359–367, August 1969.

[70] Mario B. Ignagni. Separate-Bias Kalman Estimator with Bias State Noise. *IEEE Transactions on Automatic Control*, 35(3):338–341, March 1990.

[71] A. T. Alouani, P. Xia, T. R. Rice, and W. D. Blair. On the Optimality of Two-Stage State Estimation In the Presence of Random Bias. *IEEE Transactions on Automatic Control*, 38(8):1279–1282, August 1993.

[72] Mario (B.) Ignagni. Optimal and Suboptimal Separate-Bias Kalman Estimator for a Stochastic Bias. *IEEE Transactions on Automatic Control*, 45(3):547–551, March 2000.

[73] Gerald J. Bierman. *Factorization Methods for Discrete Sequential Estimation*, volume 128 of *Mathematics in Sciences and Engineering*. Academic Press, 1978.

[74] Byron D. Tapley, Bob E. Schutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[75] Robert H. Bishop and Timothy Price Crain II. Unmodeled impulse detection and identification during MPF cruise: preliminary results. Technical Report CSR-TM-00-02, Center for Space Research - The University of Texas at Austin, 2000.

[76] Robert H. Bishop and Timothy Price Crain II. The Mixture-of-Experts gating network: integration into the ARTSN EKF. Technical Report CSR-TM-99-01, Center for Space Research - The University of Texas at Austin, 1999.

[77] Renato Zanetti, Robert H. Bishop, Kyle J. DeMars, and Timothy P. Crain. Precision entry navigation dead-reckoning error analysis: Theoretical foundations of the continuous-time case. *Submitted for publication*.

[78] T. P. Crain and R. H. Bishop. Mars Entry Navigation: Atmospheric Interface Through Parachute Deploy. In *AIAA Atmospheric Flight Mechanics Conference*, 2002.

[79] Guillaume Gaillet. Preliminary study of a filter for a precision landing on Mars. Technical report, The University of Texas at Austin Center for Space Research, 1997.

[80] Kyle Jordan DeMars. Precision navigation for lunar descent and landing. Master's thesis, The University of Texas at Austin, May 2007.

Vita

Renato Zanetti, the son of Gianfranco Zanetti and Laura Amalia De Carlini, is the second of three children and was born on June 6, 1978, in Olavarria, Buenos Aires, Argentina, but considers the village of Besnate, Varese, in northern Italy as his hometown. At the age of four and a half his family moved to Besnate, where they still reside. After completing middle school in 1992 he enrolled in a technical high school specializing in electronics and telecommunications. In 1997 he graduated from high school and was admitted to Politecnico di Milano, where he joined the aerospace engineering program. He spent the fourth year of college as a reciprocal exchange student at Oklahoma State University, and completed his course requirements back in Milan in July 2002. During fall 2002 he was at the University of Texas at Austin where he met Dr. Robert H. Bishop, who supervised the research that would became his master thesis. Upon returning to Milan, he wrote his thesis, defended it, and graduated in April 2003. That same year he joined the program of Aerospace Engineering at the University of Texas at Austin under the supervision of Dr. Bishop where he has been pursuing a PhD in the area of Guidance, Navigation, and Control.

Renato is married since 2005 to Erika Payàn Zanetti, they live in Austin, Texas.

Permanent Address: 210 East 24th Street, Austin, TX 78712

This dissertation was typeset and printed by the author.