NOVEL METHODS
FOR
ATTITUDE DETERMINATION
USING VECTOR OBSERVATIONS

Daniel Choukroun

NOVEL METHODS
FOR
ATTITUDE DETERMINATION
USING VECTOR OBSERVATIONS

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Daniel Choukroun

Submitted to the Senate of the Technion - Israel Institute of Technology
Eyar 5763 Haifa May 2003

In the memory of my uncle Claude.

To the future of my little niece Anna.

The Research Thesis was done under the supervision of Prof. Itzhack Bar-Itzhack and Prof. Yaakov Oshman, and with Dr. Haim Weiss as advisor, in the Faculty of Aerospace Engineering.

The generous financial help of the Technion is gratefully acknowledged.

The author is grateful to the Flight Dynamics Analysis Branch of NASA Goddard Space Flight Center for supporting this research (grant NAG5-8770).

The author is also grateful to the Sheffer Foundation for supporting the presentation of results of this research at the Conference GPS ION 2002.

It is my great pleasure to thank my supervisors and my advisor for the professional inspiration, the personal commitment, and the wisdom they show, when introducing me to the exciting world of scientific research.

Contents

Abstract **1**

List of Symbols and Abbreviations **3**

1 Introduction **7**

1.1 Subject of Research 7

1.1.1 Attitude Determination 7

1.1.2 State Matrix Kalman Filter 9

1.2 Literature Survey 10

1.2.1 Optimal Quaternion Estimators 10

1.2.2 Matrix Estimation 16

1.3 The Motivation and Goals of the Present Research 18

1.4 Overview of the Dissertation 19

2 A Novel Quaternion Kalman Filter **23**

2.1 Introduction 23

2.2 Mathematical Model 24

2.2.1 Measurement Truth-model 24

2.2.2 Process Truth-model 27

2.2.3 Truth-model Summary 29

2.3 Quaternion Kalman Filter 30

2.3.1 State Dependent Input Noise Matrices 30

2.3.2 Singular Quaternion Pseudo-measurement 30

2.3.3 Nominal Filter Summary 31

2.4 Adaptive Filter 33
2.4.1 Motivation 33
2.4.2 Approach 34
2.4.3 Solution 35
2.4.4 Quaternion Normalization 37
2.5 Simulation Study 38
2.5.1 Nominal Filter 38
2.5.2 Adaptive Filter 43
2.6 Simulation Study Using GPS Measurements 48
2.6.1 GPS Vectorized Observations 49
2.6.2 Simulation Study 50
2.7 Concluding Remarks 55
3 Optimal-REQUEST Algorithm for Attitude Determination 57
3.1 Introduction 57
3.2 Preliminaries 58
3.2.1 Wahba's Problem 58
3.2.2 The **q**-method 58
3.2.3 The REQUEST Algorithm 59
3.3 Problem Statement 60
3.4 Mathematical Model 60
3.4.1 Solution Approach 60
3.4.2 Measurement Equation 61
3.4.3 Process Equation 62
3.4.4 Pseudo-process Equation 64
3.4.5 Stochastic Models 65
3.4.6 A Measure of Uncertainty 66
3.5 Optimal-REQUEST 68
3.5.1 Measurement Update Stage 68
3.5.2 Optimal Gain 70
3.5.3 Prediction Stage 71

3.5.4 Algorithm Summary . . . 72

3.6 Adaptive Optimal-REQUEST . . . 73

3.6.1 Motivation . . . 73

3.6.2 Measurement Noise Adaptive Filtering . . . 74

3.6.3 Process Noise Adaptive Filtering . . . 76

3.7 Simulation Study . . . 81

3.7.1 Optimal-REQUEST . . . 81

3.7.2 Adaptive Optimal-REQUEST . . . 83

3.8 Concluding Remarks . . . 93

4 A Quadratically Constrained Batch-Recursive Least-Squares Quaternion Filter 95

4.1 Introduction . . . 95

4.2 Time-Invariant Case . . . 96

4.2.1 Model Description . . . 96

4.2.2 Lagrange Multipliers Approach . . . 97

4.2.3 Dynamic Programming Approach . . . 103

4.2.4 Weighting Coefficients α_k . . . 108

4.2.5 Convergence Analysis . . . 109

4.2.6 Algorithm Summary . . . 112

4.2.7 Simulation Study . . . 112

4.2.8 Concluding Remarks . . . 114

4.3 Time-Varying Case . . . 115

4.3.1 Model Description . . . 115

4.3.2 Estimation Problem . . . 118

4.3.3 Dynamic Programming Approach . . . 119

4.3.4 Algorithm Derivation . . . 121

4.3.5 Weighting Coefficients η_k . . . 129

4.3.6 Algorithm Summary . . . 129

4.3.7 Simulation Study . . . 130

4.3.8 Concluding Remarks . . . 132

4.4 Concluding Remarks 133

5 State Matrix Kalman Filter 135

5.1 Introduction 135

5.2 Main Results 136

5.2.1 Problem Formulation 136

5.2.2 Solution Approach 138

5.2.3 State-Matrix Kalman Filter 139

5.2.4 Discussion 140

5.3 Filter Derivation 143

5.3.1 Time Update Stage 143

5.3.2 Measurement Update Stage 145

5.4 Numerical Examples 147

5.4.1 Example 1 147

5.4.2 Example 2 147

5.4.3 Example 3 149

5.5 Concluding Remarks 157

6 Quaternion Estimation from Vector Observations Using a Matrix Kalman Filter 159

6.1 Introduction 159

6.2 Preliminaries 160

6.2.1 The REQUEST Algorithm 160

6.2.2 The Optimal-REQUEST Algorithm 162

6.3 The General Matrix State-Space Model 163

6.4 Mathematical Model 164

6.4.1 Process Equation 164

6.4.2 Measurement Equation 166

6.5 Filter Implementation 168

6.5.1 State-Dependence of the Process Noise 168

6.5.2 Singularity of the Measurement Noise Covariance Matrix 168

6.5.3 Algorithm Summary 168

6.5.4 Reduced Covariance Filter 170
6.5.5 Covariance Matrix of the Quaternion Estimation Error 171
6.6 Constrained Estimation 172
6.6.1 Symmetry Constraint 173
6.6.2 Trace Constraint 176
6.7 Numerical Study 178
6.7.1 Comparison between Optimal-REQUEST and the MKF algorithm . 179
6.7.2 Estimate Properties in the First Stage of the MKF 185
6.7.3 Constrained Matrix Kalman Filter 188
6.8 Discussion 189
6.9 Concluding Remarks 193

7 Matrix Kalman Filtering of the Direction Cosine Matrix 195
7.1 Introduction 195
7.2 Mathematical Model 197
7.2.1 Process Model 197
7.2.2 Measurement Model 199
7.3 Filter Design 199
7.3.1 Full Covariance Filter 200
7.3.2 Reduced Covariance Filter 201
7.4 Orthogonalization 202
7.4.1 Optimal Brute-Force Orthogonalization (OBF) 203
7.4.2 Iterative Brute-Force Orthogonalization (IBF) 203
7.4.3 First Type of Orthogonality Pseudo-Measurement (OPM1) 204
7.4.4 Second Type of Orthogonality Pseudo-Measurement (OPM2) 205
7.5 Augmented State-Matrix Mathematical Model 206
7.6 Simulation Study 209
7.7 Summary and Concluding Remarks 211

8 Discussion 215
8.1 Qualitative Comparison of the Quaternion Optimal Estimators 215

8.2 Quantitative Comparison of the Attitude Optimal Estimators 217

8.3 The QKF Algorithm and the **q**-Method 222

8.4 State Matrix Kalman Filter 222

9 Conclusions 225

10 Recommendations For Further Work 229

A 231

A.1 Property of the Ξ-matrix 231

A.2 Derivation of Eq. (2.71) 231

B 233

B.1 Proof of Eq. (3.20) 233

B.2 Computation of the Matrices $\mathcal{Q}_k$ and $\mathcal{R}_k$ 235

B.2.1 Computation of $\mathcal{Q}_k$ 236

B.2.2 Computation of $\mathcal{R}_k$ 238

B.3 Proof of Eq. (3.75) 240

B.4 Proof of Eqs. (3.78) 242

B.5 Proof of Eq. (3.79) 243

C 245

C.1 Factorization of type A 245

C.2 Factorization of type B 245

C.3 Factorization of type C 246

D 249

D.1 Definitions 249

D.2 Propositions 250

D.3 Information Form of the MKF algorithm 257

D.4 Proof of Prop. 5.2.1 258

D.5 Proof of Prop. 5.2.2 260

E **263**

E.1 Derivation of Eq. (6.17) 263

E.2 Process Noise Covariance Matrix Q_k 266

E.3 Measurement Noise Covariance Matrix R_{k+1} 269

E.4 Proof of the Reduced Covariance Filter 271

F **275**

F.1 Proof of Eq. (7.6) 275

F.2 Proof of Eq. (7.8) 276

F.3 Derivation of the DCM Reduced Covariance Filter 277

F.4 Proof of Proposition 7.5.1 279

F.5 Proof of Proposition 7.5.2 281

G **283**

G.1 Proof of Eq. (8.3) 283

Bibliography **285**

List of Figures

1.1 Logical Structure of the Dissertation. (The numbers are chapters numbers). 21

2.1 Monte-Carlo mean of the quaternion estimate in the nominal filter over 100 runs. 39

2.2 Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error in the nominal filter over 100 runs. 40

2.3 Monte-Carlo mean and $\pm 1\sigma$-envelope of the quaternion estimation error in the nominal filter over 100 runs. 41

2.4 Standard deviation of the estimation error in the nominal filter for a sample run. 42

2.5 Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error in the adaptive filter over 100 runs. 44

2.6 Monte-Carlo mean and $\pm 1\sigma$ envelope of the quaternion estimation error in the adaptive filter over 100 runs. 45

2.7 Standard deviation of the estimation error in the adaptive filter for a sample run. 46

2.8 Monte-Carlo mean of the adaptive gyro noise standard deviation over 100 runs. 47

2.9 GPS case. Monte-Carlo mean of the angular estimation error in the nominal filter over 100 runs. 52

2.10 GPS case. Quaternion estimation error variances in the nominal filter for a sample run. 53

2.11 GPS case. Monte-Carlo mean angular estimation error in the adaptive filter over 100 runs. 54

3.1 Nominal filter. Monte-Carlo mean of the angular estimation error $\delta\phi$. . . . 83

3.2 Nominal filter. Time histories of the filter gain ρ and of the process noise level η on a sample run. 84

3.3 Nominal filter. Comparison of Optimal REQUEST and REQUEST with constant gains. 85

3.4 Adaptive measurement noise filter. Difference between the predicted and the current squared residuals on a sample run. 86

3.5 Adaptive measurement noise filter. Filter levels of noise and Filter gain on a sample run. 87

3.6 Adaptive measurement noise filter. Angular estimation error $\delta\phi$ on a sample run. 88

3.7 Adaptive process noise filter. Case 1. Difference between the predicted and the current squared residuals on a sample run. 89

3.8 Adaptive process noise filter. Case 1. Filter levels of noise and Filter gain on a sample run. 90

3.9 Adaptive process noise filter. Case 1. Angular estimation error $\delta\phi$ on a sample run. 91

3.10 Adaptive process noise filter. Case 2. Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error $\delta\phi$. 92

4.1 Time-invariant case. Simulation S_1. Angular estimation error for a sample run. 114

4.2 Time-invariant case. Simulation S_2. Angular estimation error for a sample run. 115

4.3 Time-varying case. Angular estimation error for a sample run. 134

5.1 Example 1 - Estimation errors for a sample run. 148

5.2 Example 1 - Monte-Carlo and Filter Means and $\pm 1\sigma$-envelopes of the estimation errors for 100 runs - Blue thick line: mean, blue dotted line: $\pm 1\sigma$, red thin line: Filter $\pm 1\sigma$. 149

5.3 Example 2 - Estimated matrix state elements for a sample run. 150

5.4 Example 2 - Monte-Carlo and Filter Means and $\pm 1\sigma$-envelopes of the estimation errors for 100 runs - Blue thick line: mean, blue dotted line: $\pm 1\sigma$, red thin line: Filter $\pm 1\sigma$. 151

5.5 Example 3 - Sample run - Time histories of the estimation errors. 155

5.6 Example 3 - Monte-Carlo simulation - Mean and $\pm 1\,\sigma$-envelope of the estimation errors. Blue thick line: mean, blue dotted line: MC 1σ, red thin line: Filter 1σ. 156

6.1 Monte-Carlo mean and $\pm 1\sigma$ envelope of ΔX_{11}. 180

6.2 Monte-Carlo mean and $\pm 1\sigma$ envelope of ΔX_{13}. 181

6.3 Monte-Carlo mean and $\pm 1\sigma$ envelope of ΔX_{14}. 181

6.4 Monte-Carlo mean of ρ_{OPREQ} and ρ_{MKF}. 182

6.5 Monte-Carlo mean and $\pm 1\sigma$ envelope of δe_1. 182

6.6 Monte-Carlo mean and $\pm 1\sigma$ envelope of δe_2. 183

6.7 Monte-Carlo mean and $\pm 1\sigma$ envelope of δe_3. 183

6.8 Monte-Carlo mean and $\pm 1\sigma$ envelope of $1 - \delta q$. 184

6.9 Monte-Carlo mean and $\pm 1\sigma$ envelope of $\delta\phi$. 184

6.10 Covariance matrix of $[\Delta X_{11}\ \Delta X_{21}\ \Delta X_{12}\ \Delta X_{22}]^T$. 187

6.11 Constrained Estimation. Monte-Carlo mean of δe_1. 188

6.12 Constrained Estimation. Monte-Carlo mean of δe_2. 189

6.13 Constrained Estimation. Monte-Carlo mean of δe_3. 189

6.14 Constrained Estimation. Monte-Carlo mean of $1 - \delta q$. 190

6.15 Constrained Estimation. Monte-Carlo mean of $\delta\phi$. 190

7.1 Monte-Carlo means of the estimation convergence index J_c. 211

7.2 Monte-Carlo means of the orthogonality index J_o. 212

8.1 Comparison of the optimal attitude estimators. 220

List of Tables

7.1 Monte-Carlo means of $J_c \cdot 10^5$ at the final time 210

7.2 Monte-Carlo means of J_o at the final time 210

7.3 Qualitative Comparison of Orthogonalization Techniques 213

8.1 Properties of the Quaternion Optimal Filters 217

8.2 Sampling Times and Noises Stochastic Models 219

8.3 Steady-state Monte-Carlo Means and Standard Deviations of $\delta\phi$ for 100 runs221

D.1 Correspondence between u and (i,j,k,l) 252

Abstract

Quaternions of rotation have become popular for the purpose of attitude parametrization because they yield a minimal parameter global attitude representation and describe the kinematics of a rigid body through a linear ordinary differential equation. Over the last forty years optimal quaternion estimators using vector observations have been developed following two main approaches that have complementary properties. One approach has its origin in a special deterministic estimation problem, the so-called "Wahba problem", from which arose the known "**q**-method" for constrained least-squares deterministic estimation of the quaternion. The other approach consists in modelling the quaternion as a stochastic state vector, and developing Extended Kalman filters for minimum variance estimation of the quaternion. Both approaches are in the realm of the linear least squares estimation theory, where the term "least squares" means both the classical and the minimum-variance settings of the theory.

In the present work we introduce four novel optimal quaternion estimators (filters) which are based on the two approaches. The performance of the filters are demonstrated by means of simulations. The first algorithm is a linear Kalman filter that operates on a special linear quaternion pseudo-measurement model. The measurement modelling is done without using the traditional linearization procedure. Adaptive filtering theory is applied to cope with unmodeled parameters in the process noise. The special case where the filter processes vectorized GPS (Global Positioning System) phase measurements is investigated. The second algorithm is an optimized REQUEST algorithm. REQUEST is a recursive algorithm for least squares fitting of the attitude quaternion to vector measurements, which uses a heuristic fading memory factor to filter propagation noises. A central feature in REQUEST is the so-called K-matrix. In order to estimate that matrix, an optimal estimation strategy is proposed here by embedding the REQUEST algorithm in a Kalman filter framework. Borrowing principles from adaptive filtering theory, separate filters for process noise and measurement noise adaptive estimation are developed. A third algorithm is developed as a batch-recursive least-squares estimator. The derivation approach for the time-varying algorithm relies on the Dynamic Programming technique, and extends it to include the quaternion unit-norm constraint. The forth algorithm is a Kalman filter of the K-matrix. That algorithm eliminates two deficiencies of the second algorithm; namely, the conservative estimation cost and the limitation on the filter gain to be a scalar. An analysis provides a rigorous justification of the performance improvement, which is evident in the simulations.

In the second part of the dissertation, the problem of matrix estimation is addressed in a broad perspective with respect to past research works. A general state-matrix Kalman filter is presented which operates on plants described naturally by matrix equations. The matrix Kalman filter developed here retains the statistical properties of the classical Kalman filter and, at the same time, has the advantages of a compact matrix notation. The matrix Kalman filter is applied to two attitude determination problems; namely the estimation of the K-matrix, mentioned before, and the estimation of the Direction Cosine Matrix. The analysis of the two algorithms shows that special important assumptions on the filter noise stochastic models yield simplified equations and substantially lower computation loads, while maintaining satisfying levels of performance. Pseudo-measurement techniques are utilized to implement special constraints like symmetry, zero-trace, or orthogonality of the estimated state-matrix. As shown by simulations, combining these techniques with ad-hoc iterative procedures seems to be a promising technique for constrained Kalman filtering.

List of Symbols and Abbreviations

Cartesian Coordinate Frames:

$\mathcal{B}$	Body frame.
$\mathcal{I}$	Inertial frame.
$\mathcal{R}$	Reference frame.

Scalars:

f, g, L, J	Performance indexes.
$a_i,\ \alpha_i,\ \beta_i$	Positive weights in least squares cost functions.
$\eta_i, \gamma_i,\ \boldsymbol{\nu}_i,\ \rho_i$	Positive weights in least squares functions.
Δt	Time increment.
$\lambda,\ \mu$	Lagrange Multipliers.
λ_{max}	Largest positive eigenvalue of the K-matrix.
q	Scalar part of the attitude quaternion.
$\widehat{\eta},\ q_\epsilon$	Process noise level adaptive estimate.
$\widehat{\rho}$	Measurement noise level adaptive estimate.
$\sigma_b,\ \sigma_\epsilon$	Standard deviations of $\boldsymbol{\delta b}$ and $\boldsymbol{\epsilon}$.

Vectors:

$\mathbf{b}$	Observation unit-norm vector.
$\mathbf{e}$	3×1 vector part of the attitude quaternion.
$\mathbf{q}$	Attitude quaternion.
$\mathbf{r}$	Reference unit-norm vector.
$\mathbf{v}$	Measurement noise vector.
$\mathbf{w}$	Process noise vector.
$\widetilde{\mathbf{y}}$	Measurement residual vector.
$\mathbf{z}$	3×1 vector computed in the q-method.
$\boldsymbol{\delta b}$	Measurement error in $\mathbf{b}$.
$\boldsymbol{\epsilon}$	Measurement error in $\boldsymbol{\omega}$.
$\boldsymbol{\omega}$	Angular velocity vector of $\mathcal{B}$ w.r.t $\mathcal{R}$ along $\mathcal{B}$.

Matrices:

A, D	Attitude matrix, DCM matrix.
B	3×3 Attitude profile matrix.
$G, K, \mathcal{K}$	Kalman gain matrices.
H, $\mathcal{H}$	Observation matrices.
I_n	$n \times n$ Identity matrix.
K	4×4 K-matrix of the $\mathbf{q}$-method.
$O_{m \times n}$	$m \times n$ null matrix.
$P, \mathcal{P}$	Estimation error covariance matrices.

Q, R	Noise covariance matrices.
$\mathcal{Q}$, $\mathcal{R}$	Noise uncertainty matrices in Optimal REQUEST.
$S, \mathcal{S}$	Measurement residual covariance matrix.
V	Measurement noise matrix.
W	Process noise matrix.
X	State-matrix, matrix of state variables.
Y	Matrix of measured variables.
Ω	Dynamics skew-symmetric matrix.
Φ, Ψ, Θ	State transition matrices.

Other Symbols:

cov	Covariance operator.
E	Expectation operator.
vec	Vec operator.
$\otimes$	Kronecker product.
tr	Trace operator.
$\widehat{(\cdot)}_{k/k}$	A posteriori estimate of the variable $(\cdot)$ at time t_k given k measurements.
$\widehat{(\cdot)}_{k+1/k}$	A priori estimate of the variable $(\cdot)$ at time t_{k+1} given k measurements.
$[\mathbf{e}\times]$	Cross-product matrix of the 3×1 column-vector $\mathbf{e}$, defined as $\begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$

Abbreviations:

AD	Attitude Determination.
BLUE	Best Linear Unbiased Estimator.
BRLS	Batch-Recursive Least-Squares.
CMKF	Constrained Matrix Kalman Filter.
DCM	Direction Cosine Matrix.
DMKF	Direction cosine Matrix Kalman Filter.
DP	Dynamic Programming.
EKF	Extended Kalman Filter.
KF	Kalman Filter.
KMKF	K-Matrix Kalman Filter.
LS	Least Squares.
MEKF	Multiplicative Extended Kalman Filter.
MKF	Matrix Kalman Filter.
OPREQ	OPtimal-REQuest.
PM	Pseudo-Measurement.
QKF	Quaternion Kalman Filter.
RLS	Recursive Least Squares.

Chapter 1

Introduction

1.1 Subject of Research

1.1.1 Attitude Determination

The first part of this dissertation is concerned with the development of optimal algorithms for attitude quaternion estimation using vector observations. We will begin by introducing the general problem of attitude determination and defining the notion of vector observation. We will focus on a specific mathematical representation of attitude, namely the quaternion-of-rotation, which is a very popular attitude parametrization especially, in the inertial navigation and space communities. A selected family of optimal algorithms for quaternion estimation will be reviewed. The survey, which covers 35 years of algorithm developments, will point out the cornerstones of their evolution and will lead to the starting point of the present research. The survey will essentially show that optimal quaternion estimators typically fall into two main categories that have complementary attributes. These properties will be extensively discussed in the ensuing. The main guideline that we kept in mind when developing the quaternion filters proposed here was to get "the best of both worlds".

The problem of attitude determination (AD) from vector observations is defined as follows. Consider a rigid body in space and define a Cartesian coordinate system, $\mathcal{B}$, that is fixed to the body. The fundamental AD problem is to specify the orientation of the axes of $\mathcal{B}$ with respect to a given reference Cartesian coordinate system, $\mathcal{R}$. Let us resolve a generalized[1] vector $\mathbf{x}$ along the axes of $\mathcal{B}$ and $\mathcal{R}$, and denote by $\mathbf{b}$ and $\mathbf{r}$, respectively, its 3×1 column-matrices of components. We denote by A the rotation matrix that brings the axes of $\mathcal{R}$ onto the axes of $\mathcal{B}$. This matrix is known as the *attitude* matrix (also called the Direction Cosine Matrix or DCM [1, p. 410]). Typical vector observations on-board spacecrafts are the Earth Magnetic Field, the direction to the Sun, to the Moon, or to other celestial bodies. In general, the reference vectors, $\mathbf{r}$, are known quite accurately from tables or almanacs, while the body vectors, $\mathbf{b}$, are obtained from measurements corrupted by errors. As a result, an attitude matrix measurement model is described by the following linear equation

$$\mathbf{b} \;=\; A\,\mathbf{r} + \boldsymbol{\delta b} \tag{1.1}$$

where $\boldsymbol{\delta b}$ is the error in $\mathbf{b}$. In an AD problem when using vector observations, we want to estimate A given a sequence of pairs $(\,\mathbf{b},\, \mathbf{r})$. Note that, given two non-collinear observations[2], it is possible to estimate A by means of deterministic algorithms, e.g. [2]. However, a single observation is not enough to yield an unambiguous attitude matrix [3].

The A-matrix is one parametrization of attitude among several others. Each of them presents advantages and disadvantages, which are discussed in extensive surveys in the literature (see e.g. [1, pp. 410–420]). Another popular attitude representation is the quaternion of rotation, denoted by $\mathbf{q}$. The quaternion $\mathbf{q}$ is a real 4×1 column-matrix with a vector part, $\mathbf{e}$, and a scalar part, q, which is defined on the unit sphere of $\mathbb{R}^4$ [1, pp. 758,759]. The relation between $\mathbf{q}$ and A, as given in [1, p. 414], is

$$A\,(\mathbf{q}) = \left(\,q^2 - \mathbf{e}^T\,\mathbf{e}\right)\; I_3 \;+\; 2\;\mathbf{e}\,\mathbf{e}^T \;-\; 2\;q\;\left[\,\mathbf{e}\times\right] \tag{1.2}$$

where I_3 is the 3×3 identity matrix, $\mathbf{e}^T$ denotes the transpose of $\mathbf{e}$, and the cross-product matrix $[\,\mathbf{e}\times]$ is defined by

$$[\,\mathbf{e}\times] \triangleq \begin{bmatrix} 0 & -\,e_3 & e_2 \\ e_3 & 0 & -\,e_1 \\ -\,e_2 & e_1 & 0 \end{bmatrix} \tag{1.3}$$

where e_i, $i = 1, 2, 3$, are the components of $\mathbf{e}$.

[1] A generalized vector is characterized by a pointing direction and a length.

[2] By two non-collinear observations we mean two observed generalized-vectors that are non-collinear.

It is known since Euler[3] that three independent parameters are needed to specify the orientation of a rigid body in space [1, p. 417]. By definition, quaternions of rotation consist of four elements; that is, the quaternion is a 4 parameter rotation specifier and has, thus, one redundant component. Another shortcoming of the quaternion is that it is not a 1-1 representation of the rotation group.[4] Indeed, both quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same attitude (they yield the same attitude matrix in Eq. 1.2). Nevertheless, using the quaternion is very convenient because it gives a global attitude representation; that is, there is no singularity problem at any orientation of the rigid body. Furthermore, the equations of motion of a rigid body are linear differential equations in the components of $\mathbf{q}$ [1, p. 512]. The latter is a desirable property when developing estimation and control algorithms. Moreover, the quaternion is a relatively computationally efficient parametrization since, as opposed to Euler Angles, it does not involve trigonometric functions to compute the attitude matrix, and has *only* one redundant parameter, as opposed to the six redundant elements of the attitude matrix.

1.1.2 State Matrix Kalman Filter

The second part of the dissertation presents the development and some applications of a general discrete-time Kalman filter for state matrix estimation. The classical Kalman filter (KF) was developed to estimate a state column-vector given a sequence of column-vector measurements. As a result the KF formulation is not naturally suited to the case of matrix random variables. Indeed, there are systems in which the dynamics of the variables is naturally described by matrix differential/difference equations. For instance, the Direction Cosine Matrix (Eq. 1.2) is a matrix whose dynamics is governed by a matrix differential equation [1, p. 512]. In the past, research has been continuously conducted in order to design algorithms that estimate matrix parameters *without breaking the original matrix structure of the system.* Contributing to that research, the State Matrix Kalman filter is developed here to operate on variables that are naturally described by matrix equations.

[3] Euler first showed in 1776 that the group of rotation of Euclidean 3-space is itself a 3-dimensional manifold.

[4] It was shown in [4] that the minimal 1-1 global attitude parametrization needs a set of 5 elements.

1.2 Literature Survey

1.2.1 Optimal Quaternion Estimators

Optimal attitude determination (AD) algorithms have been developed following two main approaches: the classic constrained least squares approach, based on the so-called Wahba problem, and the minimum-variance (Kalman filtering) approach.

Constrained Least Squares Approach

In 1965, Wahba formulated the AD problem using vector observations as a constrained least squares estimation problem [5]:
Given the two sets of n vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ *and* $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, $n \geq 2$, *where each pair* $(\mathbf{r}_i, \mathbf{b}_i)$ *corresponds to a generalized vector,* $\mathbf{x}_i$, *find the proper orthogonal matrix, A, which brings the first set into the best least squares coincidence with the second. That is, find A which minimizes*

$$\sum_{i=1}^{n} \| \mathbf{b}_i - A\,\mathbf{r}_i \|^2 \tag{1.4}$$

subject to the constraints that A be orthogonal and that $\det(A) = 1$.

This problem is generally known as Wahba's problem. Note that in the present formulation Wahba's problem is a *single-frame* attitude determination problem; that is, it assumes that the vector measurements that are processed to estimate the attitude have been obtained for a constant attitude. Normally this takes place at a single time point. One family of solutions of Wahba's problem is concerned with the determination of that optimal matrix A (see [6–10] for earlier solutions and [11–14] for more recent methods), while another family is concerned with the determination of the corresponding optimal quaternion. We will treat here the latter family.

Single-Frame Quaternion Estimators. In 1968, Davenport devised a method for computing the optimal single-frame quaternion $\mathbf{q}$. This method, known in the literature as the $\mathbf{q}$-method (see e.g. [1, pp. 426–428]), is summarized as follows. Given a batch of n simultaneous observations $(\mathbf{b}_i, \mathbf{r}_i)$ and the corresponding weights a_i, define the following

3×3 matrices B and S, the 3×1 vector $\mathbf{z}$ and the scalar σ

$$\begin{aligned} B &\triangleq \textstyle\sum_{i=1}^{n} a_i\, \mathbf{b}_i\, \mathbf{r}_i^T & \qquad \mathbf{z} &\triangleq \textstyle\sum_{i=1}^{n} a_i\, (\mathbf{b}_i \times \mathbf{r}_i) \\ S &\triangleq B + B^T & \qquad \sigma &\triangleq tr(B) \end{aligned} \tag{1.5}$$

where "*tr*" denotes the trace operator. Then, construct the 4×4 symmetric matrix K

$$K \triangleq \begin{bmatrix} S - \sigma I_3 & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix} \tag{1.6}$$

(Note that the trace of K is equal to zero). As reported in [3], using Eq. (1.2), Davenport showed that the weighted Wahba's cost function, Eq. (1.4), can be transformed into a quadratic function of the quaternion as follows

$$\frac{1}{2} \sum_{i=1}^{n} a_i \| \mathbf{b}_i - A\, \mathbf{r}_i \|^2 \; = \; 1 - \mathbf{q}^T K \mathbf{q} \tag{1.7}$$

where, without loss of generality, it is assumed that the a_i's add to one. Thus, Wahba's problem is equivalent to the *maximization* of a quadratic form of $\mathbf{q}$. It is known that the problem of determining the stationary values of a quadratic form on the unit sphere leads to the solution of an eigenvalue problem [15, p. 31]. As a result, the eigenvector of K that belongs to the largest positive eigenvalue is the optimal quaternion estimate. This method is an algorithm for a single-frame attitude determination that has become very popular and has inspired numerous algorithms.

In [3] a modified version of the Power Method (see [16, pp. 405–407]) was proposed to compute the sought eigenvector.

The well-known QUEST (QUaternion ESTimator) algorithm was developed in [17] on the basis of the $\mathbf{q}$-method. The author utilized the Rodriguez parameters vector $\mathbf{y}$, also called Gibbs vector, which is defined as the vector part of the quaternion, $\mathbf{e}$, scaled by the scalar part, q [1, p. 416]. Efficient closed-form expressions for the estimate of $\mathbf{y}$ and for the corresponding estimation error covariance matrix were obtained. Much attention was given to the singular case of a π-rotation between the coordinate frames, $\mathcal{B}$ and $\mathcal{R}$, because the norm of $\mathbf{y}$ becomes unbounded for such a rotation [1, p. 416]. QUEST is a very popular algorithm for single-frame attitude quaternion estimation.

Two algorithms, ESOQ (EStimator of the Optimal Quaternion) and ESOQ2, were presented in [18] and [19]. Both algorithms provide closed-form solutions of the $\mathbf{q}$-method.

ESOQ2 actually estimates the Euler Angle/Axis, which have known relations to the quaternion [1, p. 414].

All the algorithms presented so far are single-frame estimators. Therefore, they need at least two simultaneous measurements to completely estimate attitude. Moreover, the information contained in past measurements is lost. These shortcomings have been recognized and methods have been devised in order to relax the requirement for the measurements to be acquired simultaneously. These methods require knowing the angular velocity of $\mathcal{B}$ with respect to $\mathcal{R}$.

In [20] a batch algorithm is developed based on the attitude matrix kinematics equation. It is assumed that the angular velocity was measured by a triad of gyroscopes. The highlights of that algorithm are that it provides an optimal predictor of the attitude matrix, and that it can estimate, as well, a set of constant disturbance parameters, such as gyro biases. This algorithm, however, has the inconveniences of batch algorithms.

Recursive Quaternion Estimators. Recursive estimators are often more convenient than batch estimators, since they do not require storage of past data and allow real-time processing of new incoming observations.

Reference [21] presents a recursive filter that estimates the attitude quaternion and other parameters, which can have arbitrary dynamics. The update stage of the filter involves an extended version of the **q**-method. That new method relies on the spectral decomposition of the K-matrix (see Eq. 1.6). The requirement of computing the eigenfactors of the K-matrix yields a substantial increase in the filter computational burden.

The QUEST filter, a recursive discrete-time computationally simple algorithm is developed in [22]. The variable that is sequentially propagated and updated is the B-matrix, as defined in Eq. (1.5). In order to account for modelling errors in the process equation, a fading memory factor [23, pp. 285–287] is introduced in the predicting stage.

In [24], the K-matrix was considered when developing another recursive discrete-time filter, REQUEST. It is well known (see, e.g., [1, p. 512]) that the body angular motion can be described in terms of the attitude quaternion by the following differential equation

$$\dot{\mathbf{q}} = \frac{1}{2}\,\Omega\mathbf{q} \tag{1.8}$$

where Ω is a 4×4 skew symmetric matrix and is a function of the 3×1 vector of the angular velocity of the body with respect to the reference frame, resolved in body frame and denoted by $\boldsymbol{\omega}$. The expression for Ω is given by

$$\Omega = \begin{bmatrix} -\,[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\,\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{1.9}$$

The solution of Eq. (1.8) yields

$$\mathbf{q}(t_{k+1}) = \Phi(t_{k+1}, t_k)\mathbf{q}(t_k) \tag{1.10}$$

where, because Ω is skew symmetric, the transition matrix, $\Phi(t_{k+1}, t_k)$, is a 4 × 4 orthogonal matrix. Algorithms for computing $\Phi(t_{k+1}, t_k)$ can be found in standard control theory or state estimation texts, (see, e.g. [23, pp. 296–298]). Ideally, when Ω is known perfectly, the matrix $\Phi(t_{k+1}, t_k)$ propagates the attitude quaternion from time t_k to time t_{k+1}. For simplicity of notation, we denote it by Φ_k, and in the same manner, $\mathbf{q}(t_k)$ is denoted by $\mathbf{q}_k$ for any k. Consequently, Eq. (1.8) becomes

$$\mathbf{q}_{k+1} \;=\; \Phi_k \mathbf{q}_k \tag{1.11}$$

Based on Eq. (1.11), an optimal attitude prediction step was devised in terms of the K-matrix. The author denoted by $K_{i/j}$ the K-matrix that represents the attitude at t_i *and* that is constructed from the measurements up to t_j. The REQUEST propagation stage from t_k to t_{k+1} is formulated as follows

$$K_{k+1/k} \;=\; \Phi_k\, K_{k/k}\, \Phi_k^T \tag{1.12}$$

Assuming that a single observation is acquired at t_{k+1}; that is, $\mathbf{r}_{k+1}$ and $\mathbf{b}_{k+1}$ are known, one can construct the corresponding K-matrix, denoted by δK_{k+1}, as follows. First, define

$$\begin{aligned} B_{k+1} &\triangleq \mathbf{b}_{k+1}\,\mathbf{r}_{k+1}^T & \qquad S_{k+1} &\triangleq B_{k+1} + B_{k+1}^T \\ \mathbf{z}_{k+1} &\triangleq \mathbf{b}_{k+1} \times \mathbf{r}_{k+1} & \qquad \sigma_{k+1} &\triangleq tr\left(B_{k+1}\right) \end{aligned} \tag{1.13}$$

then compute δK_{k+1} as

$$\delta K_{k+1} = \begin{bmatrix} S_{k+1} - \sigma_{k+1}\, I_3 & \mathbf{z}_{k+1} \\ \mathbf{z}_{k+1}^T & \sigma_{k+1} \end{bmatrix} \tag{1.14}$$

Denoting by a_i the scalar weighting coefficient of the i^{th} observation, the following scalars are recursively computed

$$m_0 = 0 \qquad\qquad m_{k+1} = m_k + a_{k+1} \tag{1.15}$$

The update stage of REQUEST is of the form

$$K_{k+1/k+1} = \frac{\rho_{k+1} m_k}{m_{k+1}} K_{k+1/k} + \frac{a_{k+1}}{m_{k+1}} \delta K_{k+1} \tag{1.16}$$

The coefficients m_k are used to normalize the weights a_i, such that the largest eigenvalue of $K_{k+1/k+1}$ stays close to 1. The coefficient ρ_{k+1} is a fading memory factor, which is chosen to be equal to 1, if Φ_k is error-free, and between 0 and 1 for filtering propagation noises. The choice of ρ_{k+1} is heuristic, making the REQUEST algorithm a *suboptimal* filter.

To summarize, the central highlight of Wahba's approach, is that it yields elegant closed-form algorithms, which need no a priori estimate of the quaternion. Another highlight is that the unit-norm constraint on $\mathbf{q}$ is built-in in the formulation of the problem.

On the other hand, Wahba's constrained least squares approach yields estimators that lack probabilistic significance. Actually, the right choice of the weighting coefficient in Wahba's cost function is an inherent pittfall of that approach. Usually, the weights are chosen as scalars equal to the inverse of the variance of the associated vector measurement. This choice is, however, heuristic. Efforts were made in order to recast Wahba's problem as a Maximum Likelihood estimation problem [25]. This was achieved, however, by assuming very special measurement noise probability distributions. Another shortcoming of the estimators developed following this approach was the difficulty to include parameters other than attitude, their sub-optimality with regards to time propagation noises, and the fact that an estimation error covariance analysis was generally uneasy.

Minimum-Variance Approach

Following the minimum variance approach the attitude determination problem has been embedded in the Kalman filtering theory. The attitude quaternion is considered as a state variable. The state-space process equation is derived from the quaternion kinematics equation, as given in Eqs. (1.8) and (1.9), and the measurement equation is obtained by substituting $A(\mathbf{q})$ of Eq. (1.2) into Eq. (1.1).

The stochastic inputs to that system are the error vector, $\boldsymbol{\epsilon}$, in the measured angular velocity, $\boldsymbol{\omega}$, and the attitude sensors noise vector, $\boldsymbol{\delta b}_{k+1}$. The latter enters the measurement equation (1.2) as an additive noise. It is assumed that $\boldsymbol{\omega}$ is the output of a triad of body-mounted gyros that have an additive error $\boldsymbol{\epsilon}$. This error acts as a multiplicative noise in Eq. (1.8). The common approach (see e.g. [26, 27]) is to model the dynamics process as a stochastic process as follows

$$\dot{\mathbf{q}} = \frac{1}{2}\,\Omega\mathbf{q} + \mathbf{w} \tag{1.17}$$

where the process noise vector, denoted by the 4×1 vector $\mathbf{w}$, is added to take in account the imperfect knowledge in the dynamics matrix Ω. An approximation to $\mathbf{w}$ can be derived as a linear function of $\boldsymbol{\epsilon}$, and of the state $\mathbf{q}$. Consequently, the statistical model of $\mathbf{w}$ can be easily deduced from the statistical model of $\boldsymbol{\epsilon}$. Equation (1.17) together with the measurement equation

$$\mathbf{b}_{k+1} \; = \; A(\mathbf{q}_{k+1})\,\mathbf{r}_{k+1} \; + \; \boldsymbol{\delta b}_{k+1} \tag{1.18}$$

form the state-space model of the time-varying attitude quaternion system. Note that the expression $A(\mathbf{q}_{k+1})$ in Eq. (1.2) is non-linear in $\mathbf{q}_{k+1}$. Therefore, the *extended* Kalman filter (EKF), rather than the linear Kalman filter, must be applied to this system provided that the statistical models of $\mathbf{w}$ and $\boldsymbol{\delta b}_{k+1}$ are known [26–28]. Furthermore, the state-dependence in the statistical model of $\mathbf{w}$ is handled by substituting the estimated state instead of the true state in the expression for $\mathbf{w}$. This is a known feature of the EKF.

Using the quaternion as a state-vector created a special issue in the development of the various filters. This issue is the interdependence among the four components of the *true* quaternion, which is a unit-norm vector. According to the logic that the quaternion estimate should also be a unit-norm vector, various approaches were followed to develop constrained Kalman filters.

It should be noted that using the unit-norm constraint to reduce the dimension of the filter would yield a non-linear dynamics model, which is inherent to all three-dimensional attitude parametrizations. Therefore, solutions were sought to handle the unit-norm constraint while keeping the full quaternion.

The most direct response to the normalization question is “brute force” normalization of the quaternion after the measurement update stage of the EKF; namely, the division of the updated quaternion estimate by its euclidean norm. Although optimal in some sense, as shown in [29], this normalization step is performed outside the filter algorithm, and thus perturbs the stochastic estimation process. It was shown, however, that normalizing the quaternion estimate only affects the covariance computation to second order, which is outside the purview of the EKF [27]. A different approach is adopted in the so-called “multiplicative” EKF (MEKF), which is presented in [26]. In that work, the estimation error is modelled, within a good approximation, as a quaternion of rotation. The update stage of the MEKF features the multiplication of that quaternion estimation error with the a priori quaternion estimate, which is also a quaternion of rotation. That update stage yields a quaternion of rotation, and thus the estimate preserves to a certain extent the unit norm. Other approaches that are in the framework of the classical EKF, employ pseudo-measurements of the quaternion or of its norm [28], which has the advantage of avoiding the ad-hoc normalization procedures described before.

A common feature of all the developed quaternion Kalman filters is that they operate on a linearized measurement model. The linearization procedure induces undesirable effects such as biases in the estimation errors and sensitivity to a priori estimate. It is known indeed that high initial estimation errors in an EKF might lead to divergence of the estimation process.

1.2.2 Matrix Estimation

The classical Kalman filter (KF) [30, 31] was developed to operate on plants which are governed by vector differential/difference equations. In addition, the measured variable was assumed to be a vector. The description of plant dynamics and related measurement models by vector equations is a very common one; however, there are problems in which the time-propagation of the variables is most naturally described by means of matrix differential/difference equations. Similarly there are systems where the measured variables are conveniently described by matrices.

Examples of such systems can be found in various areas of sciences and engineering. In the area of mechanical systems a central issue is that of structure identification, which leads to algorithms for estimating the inertia, stiffness, and damping matrices of a given structural system (see e.g. [32]). The field of attitude determination is, in its own right, concerned with matrix estimation through the identification problem of the

Direction Cosine Matrix (DCM). The dynamics of the DCM is moreover governed by a matrix differential equation [1, p. 512]. Another famous example in the field of Optimal Estimation and Control is that of the matrix differential Riccati equation that describes the time propagation of the estimation error covariance matrix in a Kalman-Bucy filter [31].

Continuous research efforts have been conducted to develop estimation and control algorithms that naturally operate on *matrix* plant models. It should be noted that the tools for tackling the matrix estimation problem were already available. A possible approach belongs to the linear filtering theory in infinite dimensional Hilbert spaces. A general Kalman filter for infinite dimensional random processes is presented in [33], and can be conceptually applied to the special case of finite dimensional Euclidean space of $m \times n$ random matrices. In particular, that approach provides an elegant way of defining the covariance matrix of a matrix random process. A more direct approach consists of decomposing a given matrix plant into a set of vector equations and proceed with the application of the conventional Kalman filter. This approach is adopted in [34] for estimation of the 3×3 DCM from 3×1 vector measurements. A similar approach is followed in [35] where the vectorized form of the 4×4 K-matrix is estimated in a conventional KF. However for a matrix of high dimensions this approach results in an excessive number of equations, and it may be almost impossible to determine any structure and properties of the solution. This complication motivated the development of methods and algorithms that handle matrix systems in a natural manner.

Considering the above mentioned Riccati differential equation as a matrix plant, the known Kalman-Bucy filter was derived in [36] as an optimal controller for that plant. For that purpose, special tools from matrix differential calculus were used, such as *the Gradient Matrix* [37]. In Statistics, one is often interested in estimating the covariance matrix of a multivariate distribution (e.g. Gaussian distribution in [38]). Also from Statistics, the "Total Least-Squares" method handles the problem of matrix estimation (see [16, p. 595] and [39]). That method is a generalization of the classical least squares technique. It is a method of data fitting, which is based on the standard linear model (see e.g. [40, p. 10]), and which is appropriate when there are errors in both the observation vector and in the measurement matrix. Actually, it seeks optimal estimates of the observation vector and of the measurement matrix that fit the model to the data. The issue of estimating a matrix of deterministic parameters was also addressed in [38, 41–43] using a probabilistic approach. Reference [38] considered the problem of obtaining a maximum-likelihood estimate of the covariance matrix of a multivariate normal distribution. Reference [41] presented a batch maximum-likelihood estimator of unknown matrices of parameters. In that work, the processed measurement was modelled as a linear combination of the unknown matrices, and the elements of the measurement matrix were assumed to be zero-mean, independent, normally distributed random variables. In [43] a matrix version of the Best Linear Unbiased Estimator (BLUE, see e.g. [40, p. 121]) was derived. The estimator was a batch algorithm that processed a single matrix measurement, whose rows were assumed independently identically distributed. It was also shown under what conditions the matrix BLUE reduced to the classical vector BLUE. A special matrix recursive least squares algorithm, called Simplified Multivariate Least-Squares Algorithm, was derived in [44, p. 96]. Under special conditions, that algorithm features an elegant matrix extension of the classical recursive least squares algorithm.

A key-concept, which is central in [36–38, 41], is that of matrix differentiation. Indeed, this concept and the associated theorems make the considered matrix estimation problems tractable using a compact notation. A systematic notational approach for dealing with matrix differentiation of matrix functions was developed in [42] and was applied to an adaptive decision problem.

1.3 The Motivation and Goals of the Present Research

The present work is concerned, in its first part, with the development of novel optimal quaternion filters in the framework of Least-Squares Estimation Theory[5]. The main guideline is to unify the two complementary approaches mentioned before; that is, the constrained least squares approach (Wahba's approach) and the minimum variance approach (Kalman filtering), such as to get "the best of both worlds".

[5] The word "Least-Squares" is used here in a general sense that includes both the classical and the minimum-variance settings of the theory.

In the second part of the dissertation, the problem of matrix estimation is addressed in a broad perspective, with respect to the past research efforts. A general *state-matrix Kalman filter* is developed in this work. The goal is to develop the state matrix Kalman filter such that it retains the statistical properties of the classical Kalman filter, and at the same time, has the advantages of a compact matrix notation.

1.4 Overview of the Dissertation

This dissertation is divided into 10 chapters and their appendices. Chapters 2, 3, and 4, constitute the first part of the work, and are devoted to the development of new quaternion optimal filters. Chapter 2 presents a quaternion Kalman filter which is based on a novel measurement model. After the system mathematical modelling, we address the issues of filter implementation and adaptive filtering. A simulation study with a GPS example illustrates the effectiveness of that filter. In Chapter 3, the optimization problem of the fading memory factor of REQUEST is formulated and solved. That problem is addressed using a system state-space modelling and is solved through a KF-like estimation strategy, yielding a new algorithm, called Optimal-REQUEST. A third estimator, which is developed in Chapter 4, is a quadratically-constrained batch-recursive least-squares quaternion estimator. The time-invariant and the time-varying cases are treated separately. Two different methods are utilized: the Lagrange Multipliers method and the Dynamic Programming method.

The second part of the dissertation begins with Chapter 5, which is concerned with the development of a general discrete-time state-matrix Kalman filter. The algorithm is first summarized and extensively discussed. Then the proof of the algorithm is presented. Several numerical examples illustrate the validity of the state-matrix Kalman filter. Chapters 6 and 7 propose two applications of that algorithm to the problem of attitude determination. The first application is a matrix Kalman filter of the K-matrix. That work has its motivation in the conclusions of Chapter 3; namely, the filter developed in Chapter 6 aims at improving the performance of Optimal-REQUEST, which is discussed in Chapter 3. In Chapter 7 we develop a matrix Kalman filter of the Direction Cosine Matrix. In both applications we apply the elegant technique of pseudo-measurement for constrained estimation, and investigate the filters performance through extensive simulations. Chapter 8 proposes a comparative discussion of the various quaternion optimal filters developed in Chapters 2, 3, and 4, and of the matrix Kalman filters of Chapters 6, 7. We present the conclusions of this work in Chapter 9, and propose recommendations for further work in the last chapter. The chart in Fig. 1.1 presents the logical structure of the dissertation.

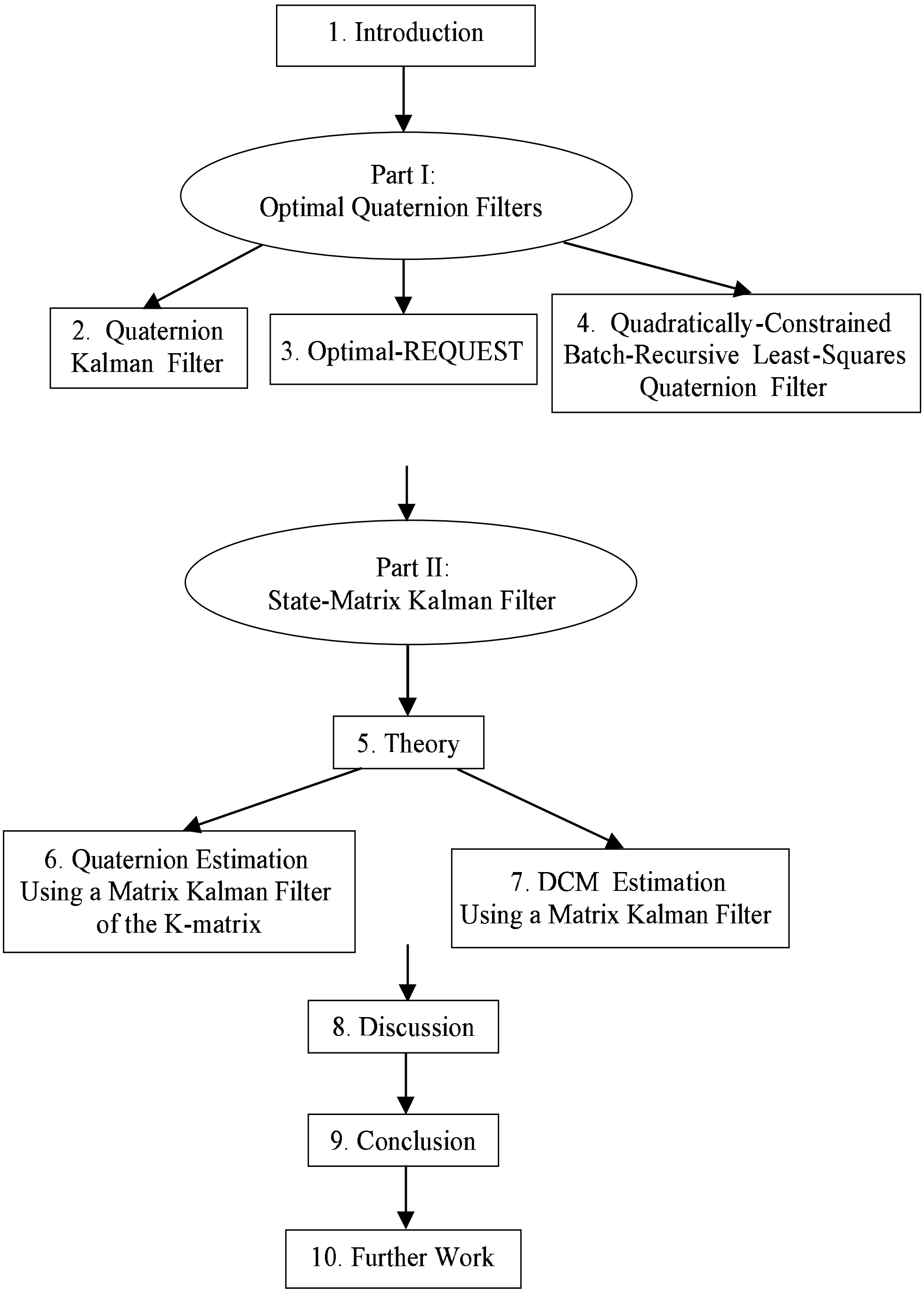

1. Introduction
Part I:
Optimal Quaternion Filters
2. Quaternion Kalman Filter
3. Optimal-REQUEST
4. Quadratically-Constrained Batch-Recursive Least-Squares Quaternion Filter
Part II:
State-Matrix Kalman Filter
5. Theory
6. Quaternion Estimation Using a Matrix Kalman Filter of the K-matrix
7. DCM Estimation Using a Matrix Kalman Filter
8. Discussion
9. Conclusion
10. Further Work

Fig. 1.1: Logical Structure of the Dissertation. (The numbers are chapters numbers).

Chapter 2

A Novel Quaternion Kalman Filter

This work was presented as Paper AIAA 2002-4460 at the 42th AIAA Guidance, Navigation, and Control Conference, Monterey, Aug. 2002 [45].

2.1 Introduction

In this chapter a novel Kalman filter (KF) for estimating the attitude quaternion from vector measurements is presented. As mentioned in Chapter 1, the common vector measurement model for the quaternion is described by the following 3×1 vector equation (see Eq. 1.1)

$$\mathbf{b} = \mathbf{b}^o + \boldsymbol{\delta b} \tag{2.1}$$

where

$$\mathbf{b}^o = A(\mathbf{q})\,\mathbf{r} \tag{2.2}$$

and $A(\mathbf{q})$ is the attitude matrix, as given in Eq. (1.2), which is rewritten here

$$A(\mathbf{q}) = (\,q^2 - \mathbf{e}^T\,\mathbf{e})\,I_3 + 2\,\mathbf{e}\,\mathbf{e}^T - 2\,q\,[\,\mathbf{e}\times] \tag{2.3}$$

In Eq. (2.3), $\mathbf{e}$ and q are, respectively, the vector and the scalar part of the attitude quaternion, $\mathbf{q}$. Substituting Eq. (2.3) into Eq. (2.2), and using the resulting expression in Eq. (2.2) yields a measurement equation which is quadratic in $\mathbf{q}$. That equation is usually linearized and the linearized model is implemented in an Extended Kalman filter. As mentioned before, the linearization procedure has undesirable consequences like biases in the estimation errors and sensitivity of the filter to initial conditions.

The present work introduces a novel model for quaternion vector measurements. This model avoids the linearization procedure. Using this approach, though, the measurement noise input matrix is quaternion dependent. An exact expression for this matrix is developed here. It should be mentioned that the quaternion process equations, which were developed in previous successful algorithms (see e.g. [26, 27]), also had a quaternion-dependent noise input matrix. An ordinary linear KF is designed here to operate on the novel quaternion state-space model, where the state-dependent coefficients are computed using the best available state estimate. In order to focus on the AD issue, it is assumed that the angular-rate is measured and that all the noises in the systems are zero-mean white noises. This imposes no limitation because there are several techniques that can be used to compensate for modelling errors like unmodeled gyro biases. One of these techniques, which is based on adaptive filtering theory, will be utilized here.

In the next section we derive the state-space *Truth-model* of the quaternion system. We begin by developing the novel quaternion measurement model, then, for the sake of completeness, the classic quaternion dynamics model is developed. In section 2.3 we describe the *Design-model* and present the corresponding quaternion Kalman filter. In section 2.4 an adaptive filter based on the Design-model is developed. Monte-Carlo simulation results for the non-adaptive and for the adaptive algorithms are presented in section 2.5. Section 2.6 is concerned with the special case where the vector observations are constructed using GPS differential phase measurements. The performance of the various filters using GPS are demonstrated by simulations. In the last section, we present the conclusions derived from this work.

2.2 Mathematical Model

2.2.1 Measurement Truth-model

We consider the pair of 3×1 unit column-matrices $\mathbf{b}^o$ and $\mathbf{r}$ obtained by resolving a physical vector, at a given time, in the body frame, $\mathcal{B}$, and in the reference frame, $\mathcal{R}$, respectively. (The time subscripts are omitted in the following development for clarity.)

We define the 4×1 quaternion vectors $\mathbf{b}^o_q$ and $\mathbf{r}_q$ as follows

$$\mathbf{b}^o_q \triangleq \left\{ \begin{matrix} \mathbf{b}^o \\ 0 \end{matrix} \right\} \qquad \mathbf{r}_q \triangleq \left\{ \begin{matrix} \mathbf{r} \\ 0 \end{matrix} \right\} \tag{2.4}$$

It is well known that $\mathbf{b}^o_q$ and $\mathbf{r}_q$ are related by the quaternion of rotation $\mathbf{q}$ as follows

$$\mathbf{b}^o_q = \mathbf{q}^{-1} \otimes \mathbf{r}_q \otimes \mathbf{q} \tag{2.5}$$

where $\otimes$ is the quaternion product and $\mathbf{q}^{-1}$ is the quaternion inverse [1, pp. 758, 759]. The definition of the quaternion product yields the identities

$$\mathbf{q} \otimes \mathbf{b}^o_q \equiv \begin{bmatrix} -[\mathbf{b}^o \times] & \mathbf{b}^o \\ -\mathbf{b}^{oT} & 0 \end{bmatrix} \mathbf{q} \tag{2.6a}$$

$$\mathbf{r}_q \otimes \mathbf{q} \equiv \begin{bmatrix} [\mathbf{r} \times] & \mathbf{r} \\ -\mathbf{r}^T & 0 \end{bmatrix} \mathbf{q} \tag{2.6b}$$

Premultiplying Eq. (2.5) by $\mathbf{q}$ and substituting Eqs. (2.6) into the resulting equation leads to the following 4×1 vector equation

$$\begin{bmatrix} -[(\mathbf{b}^o + \mathbf{r}) \times] & \mathbf{b}^o - \mathbf{r} \\ -(\mathbf{b}^o - \mathbf{r})^T & 0 \end{bmatrix} \mathbf{q} = \mathbf{0} \tag{2.7}$$

Defining the 4×1 vectors $\mathbf{s}^o$, $\mathbf{d}^o$ and the 4×4 skew-symmetric matrix H^o as follows

$$\mathbf{s}^o \triangleq \frac{1}{2} (\mathbf{b}^o + \mathbf{r}) \tag{2.8a}$$

$$\mathbf{d}^o \triangleq \frac{1}{2} (\mathbf{b}^o - \mathbf{r}) \tag{2.8b}$$

$$H^o \triangleq \begin{bmatrix} -[\mathbf{s}^o \times] & \mathbf{d}^o \\ -\mathbf{d}^{oT} & 0 \end{bmatrix} \tag{2.8c}$$

and using Eq. (2.8) in Eq. (2.7) yields

$$H^o \mathbf{q} = \mathbf{0} \tag{2.9}$$

This equation, which is linear with respect to the quaternion, is the model equation of an error-free quaternion measurement, where the observation matrix, H^o, is a function of the data $\mathbf{b}^o$, $\mathbf{r}$. Obviously, this measurement, which will be called pseudo-measurement in the following, is equal to the 4×1 null vector. In other words, Eq.(2.9) shows that $\mathbf{q}$ belongs to the null space of the matrix H^o. Notice that there is no limitation on the length of $\mathbf{q}$ in Eq. (2.9); in particular, $\mathbf{q}$ needs not be a unit vector in order to satisfy this equation.

In general, for an observation at time t_{k+1}, the vector $\mathbf{b}_{k+1}$ is the output of a sensor, e.g. magnetometer, earth sensor, sun sensor, etc. The $\mathbf{r}_{k+1}$ vector is known from tables or almanac and, thus, is relatively accurate. The measurement noise, denoted by $\boldsymbol{\delta b}_{k+1}$, is defined by

$$\boldsymbol{\delta b}_{k+1} \triangleq \mathbf{b}_{k+1} - \mathbf{b}^o_{k+1} \tag{2.10}$$

where $\mathbf{b}^o_{k+1}$ is the true value of the observation. Using $\boldsymbol{\delta b}_{k+1}$ we define a 4×4 skew-symmetric matrix ΔH_{k+1}

$$\Delta H_{k+1} \triangleq \frac{1}{2} \begin{bmatrix} -\left[\boldsymbol{\delta b}_{k+1}\times\right] & +\boldsymbol{\delta b}_{k+1} \\ -\boldsymbol{\delta b}^T_{k+1} & 0 \end{bmatrix} \tag{2.11}$$

which is the error in the knowledge of the true H-matrix due to the measurement noise. Denoting by H^o_{k+1} and H_{k+1} the true and the measured value of H, respectively, we write

$$H^o_{k+1} = H_{k+1} - \Delta H_{k+1} \tag{2.12}$$

Then, substituting this H^o_{k+1} in the following noise-free quaternion pseudo-measurement equation (at time t_{k+1})

$$\mathbf{0} = H^o_{k+1}\mathbf{q}_{k+1} \tag{2.13}$$

results in

$$\mathbf{0} = H_{k+1}\mathbf{q}_{k+1} - \Delta H_{k+1}\mathbf{q}_{k+1} \tag{2.14}$$

Using the property described in Eq. (A.1.2) of Appendix A.1, where $\mathbf{q} = \mathbf{q}_{k+1}$ and $\mathbf{x} = \boldsymbol{\delta b}_{k+1}$, Eq. (2.14) can be written as

$$\mathbf{0} = H_{k+1}\mathbf{q}_{k+1} - \frac{1}{2}\,\Xi_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{2.15}$$

where

$$\Xi_{k+1} = \begin{bmatrix} \left[\mathbf{e}_{k+1}\times\right] + q_{k+1}\, I_3 \\ -\mathbf{e}^T_{k+1} \end{bmatrix} \tag{2.16}$$

Equation (2.15) describes the quaternion pseudo-measurement model. The signal term, $H_{k+1}\mathbf{q}_{k+1}$, is a linear function of the quaternion. The noise term, $-\frac{1}{2}\,\Xi_{k+1}\,\boldsymbol{\delta b}_{k+1}$, is an additive 4×1 quaternion-dependent vector. It is assumed that the 3×1 measurement noise vector $\boldsymbol{\delta b}_{k+1}$ is a zero-mean, white-noise process with a covariance matrix R_{k+1}; that is

$$E\{\,\boldsymbol{\delta b}_{k+1}\} = \mathbf{0} \qquad\qquad E\{\,\boldsymbol{\delta b}_{k+1}\,\boldsymbol{\delta b}^T_{k+1}\} = R_{k+1} \tag{2.17}$$

We notice that the noise term, $-\frac{1}{2}\,\Xi_{k+1}\,\boldsymbol{\delta b}_{k+1}$, is a 4×1 vector function of the 3×1 vector $\boldsymbol{\delta b}_{k+1}$; the latter fact will yield a singular measurement covariance matrix in the *design-model* of the Kalman filter, as shown in the next section.

The pseudo-measurement equation (2.15) has special features. First, the "measurement" itself, which is the left side of the equation, is known a priori and second, its value is zero. In spite of these features, this model is a legitimate model suitable for use in a Kalman filter. This is so because there is no reason why the "measurement" should not be known a priori and there is no reason why zero should not be a legitimate value like any other number.

2.2.2 Process Truth-model

The quaternion process truth-model has been already developed in previous works [26, 27]. The development, which is redone here for the sake of completeness, follows the same general lines. There is though one diversion from the previous works. It is the fact that here the discrete time process equation is developed rather than its continuous time version. We consider the following discrete process of the attitude kinematics [1, pp. 511,512]

$$\mathbf{q}_{k+1} \;=\; \Phi^o_k \mathbf{q}_k \tag{2.18}$$

for $k = 0, 1, \ldots$, with initial conditions $\mathbf{q}_0$. The vector $\mathbf{q}_k$ is the quaternion of the rotation from a given reference frame $\mathcal{R}$ onto the body frame $\mathcal{B}^o$. Using $\boldsymbol{\omega}^o_k$, the angular velocity vector of $\mathcal{B}^o$ with respect to $\mathcal{R}$ resolved in $\mathcal{B}^o$, define

$$[\boldsymbol{\omega}^o_k \times] \triangleq \begin{bmatrix} 0 & -\omega^o_3 & \omega^o_2 \\ \omega^o_3 & 0 & -\omega^o_1 \\ -\omega^o_2 & \omega^o_1 & 0 \end{bmatrix} \tag{2.19a}$$

$$\Omega^o_k \triangleq \frac{1}{2}\begin{bmatrix} -\,[\boldsymbol{\omega}^o_k \times] & \boldsymbol{\omega}^o_k \\ -\,\boldsymbol{\omega}^{oT}_k & 0 \end{bmatrix} \tag{2.19b}$$

Then, assuming that $\boldsymbol{\omega}^o_k$ is constant during the time increment Δt, the 4×4 orthogonal transition matrix Φ^o_k is expressed by [1, pp. 511, 512]

$$\Phi^o_k = \exp\left(\Omega^o_k \Delta t\right) \tag{2.20}$$

In practice, $\boldsymbol{\omega}^o_k$ is not known, but it is rather measured or estimated. In this work, it is assumed that a triad of gyroscopes measures the angular velocity $\boldsymbol{\omega}^o_k$ where

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}^o_k + \boldsymbol{\epsilon}_k \tag{2.21}$$

The measured angular velocity vector is denoted by $\boldsymbol{\omega}_k$, and $\boldsymbol{\epsilon}_k$ is the gyro measurement noise vector. A "measured" transition matrix can be computed by using $\boldsymbol{\omega}_k$ instead of $\boldsymbol{\omega}^o_k$ in Eqs. (2.19) and (2.20). This matrix, which is denoted by Φ_k, differs from Φ^o_k by $\Delta\Phi_k$, that is,

$$\Phi_k = \Phi^o_k + \Delta\Phi_k \tag{2.22}$$

The error matrix $\Delta\Phi_k$ can be expressed as a matrix power series where the norm of the n^{th} term is of order $\mathcal{O}(\| \boldsymbol{\epsilon}_k \Delta t\|^n)$, for $n = 1, 2 \ldots$. Assuming that the noise $\boldsymbol{\epsilon}_k$ and the time increment Δt are small enough, the matrix $\Delta\Phi_k$ is approximated by its first order term only; that is

$$\Delta\Phi_k \simeq \mathcal{E}_k \Delta t \tag{2.23}$$

where the expression for the 4×4 matrix $\mathcal{E}_k$ is obtained by substituting $\boldsymbol{\epsilon}_k$ into $\boldsymbol{\omega}^o_k$ of Eqs. (2.19). Inserting Eq. (2.23) in Eq. (2.22) and using the resulting equation in the nominal discrete dynamics equation (2.18), yields

$$\mathbf{q}_{k+1} \simeq \Phi_k \mathbf{q}_k - \Delta t\, \mathcal{E}_k \mathbf{q}_k \tag{2.24}$$

Then, using the property described in Eq. (A.1.2) of Appendix A.1, where $\mathbf{q} = \mathbf{q}_k$ and $\mathbf{x} = \boldsymbol{\epsilon}_k$, we obtain

$$\mathbf{q}_{k+1} \simeq \Phi_k \mathbf{q}_k - \frac{\Delta t}{2} \Xi_k \boldsymbol{\epsilon}_k \tag{2.25}$$

The discrete process truth-model is finally defined by

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k + \mathbf{w}_k \tag{2.26}$$

where

$$\mathbf{w}_k = -\frac{\Delta t}{2}\,\Xi_k\,\boldsymbol{\epsilon}_k \qquad\qquad \Xi_k = \begin{bmatrix} [\mathbf{e}_k \times] + q_k\, I_3 \\ -\,\mathbf{e}_k^T \end{bmatrix} \tag{2.27}$$

Equation (2.26) is a first order approximation in $\boldsymbol{\epsilon}_k$ and Δt of the exact process equation (Eq. 2.18). One can notice that Eq. (2.26) does not preserve the unit-norm property of the quaternion. However, simulations show that for sufficiently small values of Δt and $\boldsymbol{\epsilon}_k$; that is, $\Delta t = 0.1\,\text{sec}$ and $\|\,\boldsymbol{\epsilon}_k\| = 10^{-3}\,\text{rad/sec}$, the difference between the $\mathbf{q}_k$'s of Eq. (2.18) and Eq. (2.26) is negligible for all practical purposes. As stated earlier, the noise $\boldsymbol{\epsilon}_k$ is assumed to be a zero-mean white noise process with a known covariance matrix Q_k^{ϵ}; that is

$$E\,[\,\boldsymbol{\epsilon}_k] = \mathbf{0} \qquad\qquad E\left[\,\boldsymbol{\epsilon}_k\,\boldsymbol{\epsilon}_k^T\right] = Q_k^{\epsilon} \tag{2.28}$$

2.2.3 Truth-model Summary

The process and measurement equations of the system are as follows:

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k - \frac{\Delta t}{2}\,\Xi_k\,\boldsymbol{\epsilon}_k \tag{2.29}$$

$$\mathbf{0} = H_{k+1}\mathbf{q}_{k+1} - \frac{1}{2}\,\Xi_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{2.30}$$

The transition matrix Φ_k is computed from the measured angular velocity $\boldsymbol{\omega}_k$,

$$\Omega_k = \frac{1}{2}\begin{bmatrix} -\,[\boldsymbol{\omega}_k \times] & \boldsymbol{\omega}_k \\ -\,\boldsymbol{\omega}_k^T & 0 \end{bmatrix} \tag{2.31}$$

$$\Phi_k = \exp\left(\,\Omega_k\,\Delta t\right) \tag{2.32}$$

The observation matrix H_{k+1} is computed from the vector observation at t_{k+1}, $(\,\mathbf{b}_{k+1}, \mathbf{r}_{k+1})$,

$$\mathbf{s}_{k+1} = \frac{1}{2}\,\left(\mathbf{b}_{k+1} + \mathbf{r}_{k+1}\right) \tag{2.33a}$$

$$\mathbf{d}_{k+1} = \frac{1}{2}\,\left(\mathbf{b}_{k+1} - \mathbf{r}_{k+1}\right) \tag{2.33b}$$

$$H_{k+1} = \begin{bmatrix} -\,\left[\mathbf{s}_{k+1}\times\right] & \mathbf{d}_{k+1} \\ -\mathbf{d}_{k+1}^T & 0 \end{bmatrix} \tag{2.33c}$$

The expressions for Ξ_k and Ξ_{k+1} are given, respectively, in Eqs. (2.27) and (2.16). The system noise vectors $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\delta b}_{k+1}$ are zero-mean white noise sequences, with respective covariance matrices, Q_k^ϵ and R_{k+1}. They are uncorrelated with one another and uncorrelated with the initial quaternion $\mathbf{q}_0$. The case of biased error will be treated in a later section.

2.3 Quaternion Kalman Filter

Two issues must be addressed in order to implement a Kalman filter employing the model described in the preceding section:

1. The state-dependence of the input noise matrices Ξ_k, Ξ_{k+1} in Eqs. (2.29), (2.30).

2. The singularity in the quaternion pseudo-measurement covariance matrix.

These issues will be resolved in the ensuing which will lead to an efficient Kalman filter algorithm.

2.3.1 State Dependent Input Noise Matrices

The input noise matrices Ξ_k and Ξ_{k+1} in Eqs. (2.29) and (2.30) depend on the true quaternion. The true quaternion is unknown; therefore, we replace the true quaternion by its best estimate. Hence the coefficient matrices Ξ_k and Ξ_{k+1} become functions of the best available state estimate. Consequently, the system equations of the design-model on which the filter operates are as follows:

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k - \frac{\Delta t}{2}\,\widehat{\Xi}_k\,\boldsymbol{\epsilon}_k \tag{2.34}$$

$$\mathbf{0} = H_{k+1}\mathbf{q}_{k+1} - \frac{1}{2}\,\widehat{\Xi}_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{2.35}$$

where the matrices $\widehat{\Xi}_k$ and $\widehat{\Xi}_{k+1}$ are computed using Eqs. (2.27) and (2.16), respectively. In Eq. (2.27) $\mathbf{q}$ is replaced by $\widehat{\mathbf{q}}_{k+1/k}$, the a priori estimate at t_{k+1}, and in Eq. (2.16) $\mathbf{q}$ is replaced by $\widehat{\mathbf{q}}_{k/k}$, the a posteriori estimate at t_k.

2.3.2 Singular Quaternion Pseudo-measurement

Denote by $\mathbf{v}_{k+1}$ the 4×1 noise vector in the pseudo-measurement of Eq. (2.35); that is,

$$\mathbf{v}_{k+1} = -\frac{1}{2}\,\widehat{\Xi}_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{2.36}$$

The expression for the covariance matrix of $\mathbf{v}_{k+1}$ is

$$R^q_{k+1} \triangleq E\left[\mathbf{v}_{k+1}\mathbf{v}^T_{k+1}\right] = \frac{1}{4}\,\widehat{\Xi}_{k+1}\,R_{k+1}\,\widehat{\Xi}^T_{k+1} \tag{2.37}$$

where $\widehat{\Xi}_{k+1}$ has dimensions 4×3 and R_{k+1} is the 3×3 covariance matrix of $\boldsymbol{\delta b}_{k+1}$. Due to the latter, the rank of the 4×4 matrix R^q_{k+1} is at most three, thus, R^q_{k+1} is singular. There are several techniques to cope with the issue of a singular measurement covariance matrix [40, p. 354]. The way chosen in the present algorithm to circumvent the problem is to add small values to the main diagonal of R^q_{k+1}, which has a stabilizing effect on the numerics of the Kalman filter. That is, choose a small α relatively to the assumed level of the noise, and compute

$$R^q_{k+1} = \frac{1}{4}\,\widehat{\Xi}_{k+1}\,R_{k+1}\,\widehat{\Xi}^T_{k+1} + \alpha\, I_4 \tag{2.38}$$

2.3.3 Nominal Filter Summary

Using the previous developments the nominal quaternion Kalman filter is summarized as follows.

- *Filter Initialization*

  Choose the appropriate values for Q^ϵ_k and R_{k+1}, where

$$Q^\epsilon_k = E\left[\boldsymbol{\epsilon}_k\,\boldsymbol{\epsilon}^T_k\right] \tag{2.39}$$

$$R_{k+1} = E\left[\boldsymbol{\delta b}_{k+1}\,\boldsymbol{\delta b}^T_{k+1}\right] \tag{2.40}$$

  and choose an approximate value for the initial estimate of the state vector. In the absence of such initial estimate, choose

$$\widehat{\mathbf{q}}^T_{0/0} = [0\;0\;0\;1] \tag{2.41}$$

- *Time propagation*

$$\Omega_k \triangleq \frac{1}{2}\begin{bmatrix} -\left[\boldsymbol{\omega}_k\times\right] & \boldsymbol{\omega}_k \\ -\boldsymbol{\omega}^T_k & 0 \end{bmatrix} \tag{2.42}$$

$$\Phi_k = \exp\left(\Omega_k\,\Delta t\right) \tag{2.43}$$

$$\widehat{\mathbf{q}}_{k+1/k} = \Phi_k\,\widehat{\mathbf{q}}_{k/k} \tag{2.44}$$

$$Q^q_k = \left(\frac{\Delta t}{2}\right)^2 \widehat{\Xi}_k Q^\epsilon_k\,\widehat{\Xi}_k^{\;T} \tag{2.45}$$

$$P_{k+1/k} = \Phi_k\,P_{k/k}\,\Phi^T_k + Q^q_k \tag{2.46}$$

- *Measurement update*

$$\begin{aligned}
\mathbf{s}_{k+1} &= \frac{1}{2}\left(\mathbf{b}_{k+1} + \mathbf{r}_{k+1}\right) && (2.47)\\
\mathbf{d}_{k+1} &= \frac{1}{2}\left(\mathbf{b}_{k+1} - \mathbf{r}_{k+1}\right) && (2.48)\\
H_{k+1} &= \begin{bmatrix} -\left[\mathbf{s}_{k+1}\times\right] & \mathbf{d}_{k+1} \\ -\mathbf{d}_{k+1}^T & 0 \end{bmatrix} && (2.49)\\
R^q_{k+1} &= \frac{1}{4}\,\widehat{\Xi}_{k+1}\,R_{k+1}\,\widehat{\Xi}^T_{k+1} + \alpha\, I_4 && (2.50)\\
S_{k+1/k} &= H_{k+1}\,P_{k+1/k}\,H^T_{k+1} + R^q_{k+1} && (2.51)\\
K_{k+1} &= P_{k+1/k}\,H^T_{k+1}\,S^{-1}_{k+1/k} && (2.52)\\
\widehat{\mathbf{q}}_{k+1/k+1} &= \left(I_4 - K_{k+1}\,H_{k+1}\right)\,\widehat{\mathbf{q}}_{k+1/k} && (2.53)\\
P_{k+1/k+1} &= \left(I_4 - K_{k+1}\,H_{k+1}\right)\,P_{k+1/k}\,\left(I_4 - K_{k+1}\,H_{k+1}\right)^T \\
&\quad + K_{k+1}\,R^q_{k+1}\,K^T_{k+1} && (2.54)
\end{aligned}$$

The matrices $\widehat{\Xi}_k$ and $\widehat{\Xi}_{k+1}$ are computed according to the format

$$\Xi \triangleq \begin{bmatrix} [\mathbf{e}\times] + q I_3 \\ -\mathbf{e}^T \end{bmatrix} \tag{2.55}$$

when substituting the components of $\mathbf{q}$ by the components of $\widehat{\mathbf{q}}_{k/k}$ and $\widehat{\mathbf{q}}_{k+1/k}$, respectively.

Notice that contrary to previous quaternion Kalman filters [26–28] the present filter does not produce a quaternion estimate whose length is equal to or even close to unity. As will be shown by extensive Monte-Carlo simulation, this does not impair the efficiency of the filter. This can be explained by recalling that the error-free pseudo-measurement equation is (see Eq. 2.9)

$$H^o\mathbf{q} = \mathbf{0} \tag{2.56}$$

According to Eq. (2.56), any $\mathbf{q}$ that belongs to the null space of H^o is a legitimate true quaternion; therefore, as mentioned before, this model does not imply any restriction upon the length of the true quaternion. As a consequence, when this model is implemented in the Kalman filter, the filter produces a feedback on *the direction* of the quaternion estimate only, not on its *length.* This can be understood as follows. Re-write Eq. (2.53) in the form

$$\widehat{\mathbf{q}}_{k+1/k+1} = \widehat{\mathbf{q}}_{k+1/k} + K_{k+1}\left(\mathbf{0} - H_{k+1}\,\widehat{\mathbf{q}}_{k+1/k}\right) \tag{2.57}$$

This is the KF measurement update formula of the state vector. Normally the first element in the brackets is the measurement. In our special case the "measurement" is zero. The difference between the measurement and the expected measurement, $H_{k+1}\,\widehat{\mathbf{q}}_{k+1/k}$, serves as an error signal for correcting the a prioris estimate $\widehat{\mathbf{q}}_{k+1/k}$. This is the feedback feature of the KF. Obviously, the true $\mathbf{q}_{k+1}$ is close to the null-space of H_{k+1}. (The proximity depends on how noisy the data $\mathbf{b}_{k+1}$ is (see Eqs. 2.47, 2.48, and 2.49). As mentioned above, in the case of noise-free data, $\mathbf{q}_{k+1}$ would lie exactly in the null-space of H_{k+1}.) Then any estimate $\widehat{\mathbf{q}}_{k+1/k}$ that lies in the direction of $\mathbf{q}_{k+1}$ is also close to the null-space of H_{k+1} and almost no correction signal is generated in Eq. (2.57) that changes the value of $\widehat{\mathbf{q}}_{k+1/k}$. As a result the estimated quaternion converges to the true quaternion in direction but not in length. Nevertheless, if there is a need for a unit-norm estimate of the quaternion in order to transform vectors between the body and reference coordinate systems or for display purposes then the updated quaternion can be normalized outside the estimation process.

2.4 Adaptive Filter

2.4.1 Motivation

It is well known that the KF is an optimal globally convergent state estimator under the assumptions that the system truth-model is linear and the design-model is identical to the truth-model. However, the KF becomes suboptimal and sensitive to initial conditions if there are errors in the design-model. This is the case of the current model, where the gain noise matrix, Ξ, is state-dependent, and where the gyro outputs are assumed to be unbiased. An adequate approach to *on line* enhancement of the filter performance is adaptive filtering. In this section we present an adaptive filter that processes the measurement residuals in order to optimally compensate for modelling errors according to some performance index. The modelling errors include, for instance, unmodeled gyro biases and erroneous system noise covariance matrices. The case of *process noise* adaptive estimation is considered in the following. However, a similar approach can be followed to develop a *measurement noise* adaptive filter.

2.4.2 Approach

The approach presented herein is inspired by Jazwinski [46, p. 311] , which considered the case of a scalar Gaussian measurement noise with one predicted residual processed by an adaptive filter. In this work we consider vector measurements and use a *sample mean* of N predicted residuals in the adaptive algorithm. The use of an *N-sample mean* gives the algorithm a higher statistical significance. Assume that a sample of N measurements was acquired from t_k to t_{k+N}. The l-step measurement residuals process at t_{k+l} given measurements up to t_k is denoted by $\boldsymbol{\nu}_{k+l}$ and is defined as

$$\boldsymbol{\nu}_{k+l} = \mathbf{y}_{k+l} - H_{k+l}\,\widehat{\mathbf{q}}_{k+l/k} \tag{2.58}$$

for $l = 1, 2 \ldots N$, where $\mathbf{y}_{k+l}$ is the measurement at t_{k+l}, H_{k+l} is computed from the single vector measurement at t_{k+l}, and $\widehat{\mathbf{q}}_{k+l/k}$ is the a priori quaternion estimate at t_{k+l} given measurements up to t_k. In the present model, $\mathbf{y}_{k+1}$ is the null vector (see Eq. 2.35), so that Eq. (2.58) becomes

$$\boldsymbol{\nu}_{k+l} = -\,H_{k+l}\,\widehat{\mathbf{q}}_{k+l/k} \tag{2.59}$$

The sample mean of N predicted residuals, denoted by $\bar{\boldsymbol{\nu}}^N$, is defined by

$$\bar{\boldsymbol{\nu}}^N \triangleq \frac{1}{N}\sum_{l=1}^{N} \boldsymbol{\nu}_{k+l} \tag{2.60}$$

Furthermore, the gyros noise covariance matrix, denoted by Q_k^ϵ, is modelled by

$$Q_k^\epsilon \; = \; \eta\, I_3 \tag{2.61}$$

where η is an unknown positive parameter to be estimated. The residuals sample second moment matrix, denoted by M^N is computed as

$$M^N = \frac{1}{N}\sum_{l=1}^{N} \boldsymbol{\nu}_{k+l}\,\boldsymbol{\nu}_{k+l}^T \tag{2.62}$$

Let $S^{k+N/k}$ denote the covariance matrix of $\bar{\boldsymbol{\nu}}^N$. This matrix is computed in the filter and is a function of the value of η assumed in the filter. Given the above premisses, we propose to solve the following minimization problem:

$$\min_{\eta \geq 0} \left\{ J(\eta) \; = \; \|M^N \; - \; S^{k+N/k}(\eta)\|^2 \right\} \tag{2.63}$$

where $\|M\|$ denotes the Frobenius norm of M; that is $\|M\|^2 \stackrel{\triangle}{=} tr(M^T M)$. The value of η that is computed from Eq. (2.63) is the optimal estimate of the gyros noise variance.

The latter approach is explained as follows. The matrix M^N is the residuals sample covariance matrix while $S^{k+N/k}$ is the covariance matrix of $\bar{\boldsymbol{\nu}}^N$, as computed using the filter design-model parameters. In particular, $S^{k+N/k}$ is a function of the a priori value of η. Hence, a good value of η is one that brings *consistency* between the sample covariance matrix M^N and its predicted value, $S^{k+N/k}$. This consistency condition is satisfied through the minimization described in Eq. (2.63).

2.4.3 Solution

For the sake of clarity of presentation, the solution is first developed for the case of a one-step residuals process and then extended to the case of a sample mean of N predicted residuals. Denoting by $P_{k/k}$ and $\widehat{\mathbf{q}}_{k/k}$, respectively, the a posteriori estimation error covariance matrix and the a posteriori quaternion estimate at t_k, and using the assumption that $Q_k^\epsilon = \eta\, I_3$, the covariance propagation step from t_k to t_{k+1} is given by

$$P_{k+1/k} = \Phi_{k+1} P_{k/k} \Phi_{k+1}^T + \eta\, \widehat{\Xi}_{k+1} \widehat{\Xi}_{k+1}^T \left(\frac{\Delta t}{2}\right)^2 \tag{2.64}$$

At t_{k+1} the residual vector $\boldsymbol{\nu}_{k+1}$ and its predicted covariance matrix are (see Eqs. 2.50 and 2.51) computed from

$$\boldsymbol{\nu}_{k+1} = -H_{k+1} \widehat{\mathbf{q}}_{k+1/k} \tag{2.65}$$

$$S_{k+1/k} = H_{k+1} P_{k+1/k} H_{k+1}^T + \frac{1}{4} \widehat{\Xi}_{k+1/k} R_{k+1} \widehat{\Xi}_{k+1/k}^T + \alpha\, I_4 \tag{2.66}$$

Using Eq. (2.64) in Eq. (2.66) yields

$$S_{k+1/k} = M_1 + \eta\, L_1 \tag{2.67}$$

where

$$M_1 \stackrel{\triangle}{=} H_{k+1} \Phi_k P_{k/k} \Phi_k^T H_{k+1}^T + \frac{1}{4} \widehat{\Xi}_{k+1} R_{k+1} \widehat{\Xi}_{k+1}^T + \alpha\, I_4 \tag{2.68}$$

$$L_1 \stackrel{\triangle}{=} H_{k+1} \widehat{\Xi}_k \widehat{\Xi}_k^T H_{k+1}^T \left(\frac{\Delta t}{2}\right)^2 \tag{2.69}$$

Then performing the minimization

$$\min_{\eta \geq 0} \left\{ J(\eta) = \| \boldsymbol{\nu}_{k+1}\ \boldsymbol{\nu}_{k+1}^T - S_{k+1/k}(\eta) \|^2 \right\} \tag{2.70}$$

yields the following minimizing η

$$\widehat{\eta}^{*} = \frac{tr\left[\left(\boldsymbol{\nu}_{k+1}\,\boldsymbol{\nu}^{T}_{k+1} - M_{1}\right)\,L^{T}_{1}\right]}{tr\left(L_{1}\,L^{T}_{1}\right)} \tag{2.71}$$

The derivation of Eq. (2.71) is provided in Appendix A.2, in which

$$A = \boldsymbol{\nu}_{k+1}\boldsymbol{\nu}^{T}_{k+1}, \qquad B = S_{k+1/k}, \qquad C = M_{1}, \qquad D = L_{1} \tag{2.72}$$

The extension of the algorithm to the case of a sample mean of N predicted residuals is straightforward. The algorithm is given in the following in a recursive form.

Algorithm: Assume that, at time t_k, the a posteriori quaternion estimate and the covariance matrix of the a posteriori estimation error have already been computed; that is, $\widehat{\mathbf{q}}_{k/k}$ and $P_{k/k}$ are known. Then, the following computations are performed:

1. Start with $l = 1$ and

$$P'_{k/k} = P_{k/k},\, \bar{\boldsymbol{\nu}}^{0} = \mathbf{0} \tag{2.73a}$$

$$M_{0} = O_{4},\, L_{0} = O_{4}, M^{0} = O_{4} \tag{2.73b}$$

2. Compute $\widehat{\Xi}_{k+l-1}$ using

$$\widehat{\Xi}_{k+l-1} = \begin{bmatrix} \left[\widehat{\mathbf{e}}_{k+l-1/k}\times\right] + \widehat{q}_{k+l-1/k}\, I_{3} \\ -\widehat{\mathbf{e}}_{k+l-1/k} \end{bmatrix} \tag{2.74}$$

3. Time propagate

$$\widehat{\mathbf{q}}_{k+l/k} = \Phi_{k+l-1}\, \widehat{\mathbf{q}}_{k+l-1/k} \tag{2.75}$$

$$P'_{k+l/k} = \Phi_{k+l-1}\, P'_{k+l-1/k}\, \Phi^{T}_{k+l-1} \tag{2.76}$$

4. Compute

$$\boldsymbol{\nu}_{k+l} = -\,H_{k+l}\,\widehat{\mathbf{q}}_{k+l/k} \tag{2.77}$$

$$M^{l} = \left(\frac{l-1}{l}\right) M^{l-1} + \frac{1}{l}\,\boldsymbol{\nu}_{k+l}\,\boldsymbol{\nu}^{T}_{k+l} \tag{2.78}$$

5. Compute $\widehat{\Xi}_{k+l}$, R^{q}_{k+l} and Ψ_{l} using

$$\widehat{\Xi}_{k+l} = \begin{bmatrix} \left[\widehat{\mathbf{e}}_{k+l/k}\times\right] + \widehat{q}_{k+l/k}\, I_{3} \\ -\widehat{\mathbf{e}}_{k+l/k} \end{bmatrix} \tag{2.79}$$

$$R^{q}_{k+l} = \frac{1}{4}\,\widehat{\Xi}_{k+l}\,R_{k+l}\,\widehat{\Xi}^{T}_{k+l} + \alpha\, I_{4} \tag{2.80}$$

$$\Psi_{l} = \left(\Phi_{k+l-1}\widehat{\Xi}_{k+l-1}\widehat{\Xi}^{T}_{k+l-1}\,\Phi^{T}_{k+l-1} + \widehat{\Xi}_{k+l}\widehat{\Xi}^{T}_{k+l}\right)\left(\,\frac{\Delta t}{2}\,\right)^{2} \tag{2.81}$$

6. Compute M_l and L_l using

$$M_l = M_{l-1} + H_{k+l} \, P'_{k+l/k} \, H^T_{k+l} + R^q_{k+l} \tag{2.82}$$

$$L_l = L_{l-1} + H_{k+l} \, \Psi_l \, H^T_{k+l} \tag{2.83}$$

7. Increase l by 1, if $l \leq N$ go to step 2.

When N predicted residuals have been processed following the above algorithm, the a posteriori estimate of η at time t_{k+N}, is computed by

$$\widehat{\eta}^* = \frac{tr\left[\left(M^N - M_N\right) L^T_N\right]}{tr\left(L_N \, L^T_N\right)} \tag{2.84}$$

If $\widehat{\eta}^*$ is positive, the a priori estimate of η is replaced by $\widehat{\eta}^*$ and the latter value is used to update the covariance matrix of the gyros noise, Q^ϵ_k, according to the assumption made in Eq. (2.61). Thus,

$$Q^\epsilon_k = \widehat{\eta}^* \, I_3 \tag{2.85}$$

Then, the quaternion process noise covariance matrix, Q^q_k, is computed by inserting Eq. (2.85) in Eq. (2.45) which yields

$$Q^q_k = \widehat{\eta}^* \left[\left(\frac{\Delta t}{2}\right)^2 \widehat{\Xi}_k \, \widehat{\Xi}^T_k \right] \tag{2.86}$$

If the computed value of $\widehat{\eta}^*$ is negative, then the process noise covariance matrix is set to zero; that is $Q^q_k = O_4$.

Equations (2.73) through (2.86) describe the adaptive part of the quaternion Kalman filter. The result of this algorithm, that is the value of Q^q_k, is substituted into Eq. (2.45) in the nominal algorithm. Note that the computations in Eqs. (2.73) through (2.86) are performed *in parallel* to the nominal algorithm since they use the quantities H_k, Φ_k, R_{k+1}, which are computed by the nominal algorithm.

2.4.4 Quaternion Normalization

Contrary to the nominal filter, where the quaternion estimate was not and did not have to be a unit-norm vector, the adaptive filter needs a unit-norm estimate in order to operate. Thus, the quaternion estimate is now normalized after each update stage as follows

$$\widehat{\mathbf{q}}_{k+1/k+1} = \frac{\widehat{\mathbf{q}}_{k+1/k+1}}{\| \, \widehat{\mathbf{q}}_{k+1/k+1} \|} \tag{2.87}$$

Equation (2.87) is inserted in the adaptive algorithm after Eq. (2.53). The quaternion estimate is normalized in order to prevent its norm from becoming too small. Such decrease in norm in the nominal filter does indeed occur as was observed in the Monte-Carlo simulations. One consequence of the decrease in the norm of $\widehat{\mathbf{q}}_{k+1/k}$ is that the innovations, $\boldsymbol{\nu}_{k+1}$, becomes very small too, as seen from Eq. (2.59); thus, even if the direction of the vector estimate, $\widehat{\mathbf{q}}_{k+1/k}$, is very inaccurate, $\boldsymbol{\nu}_{k+1}$ may stay within its predicted standard deviation envelope, which avoids the adaptive algorithm from starting. This spurious convergence is avoided by keeping the norm of the quaternion estimate equal to unity; thus an error in the direction of the vector estimate can be sensed and fed back into the estimation process.

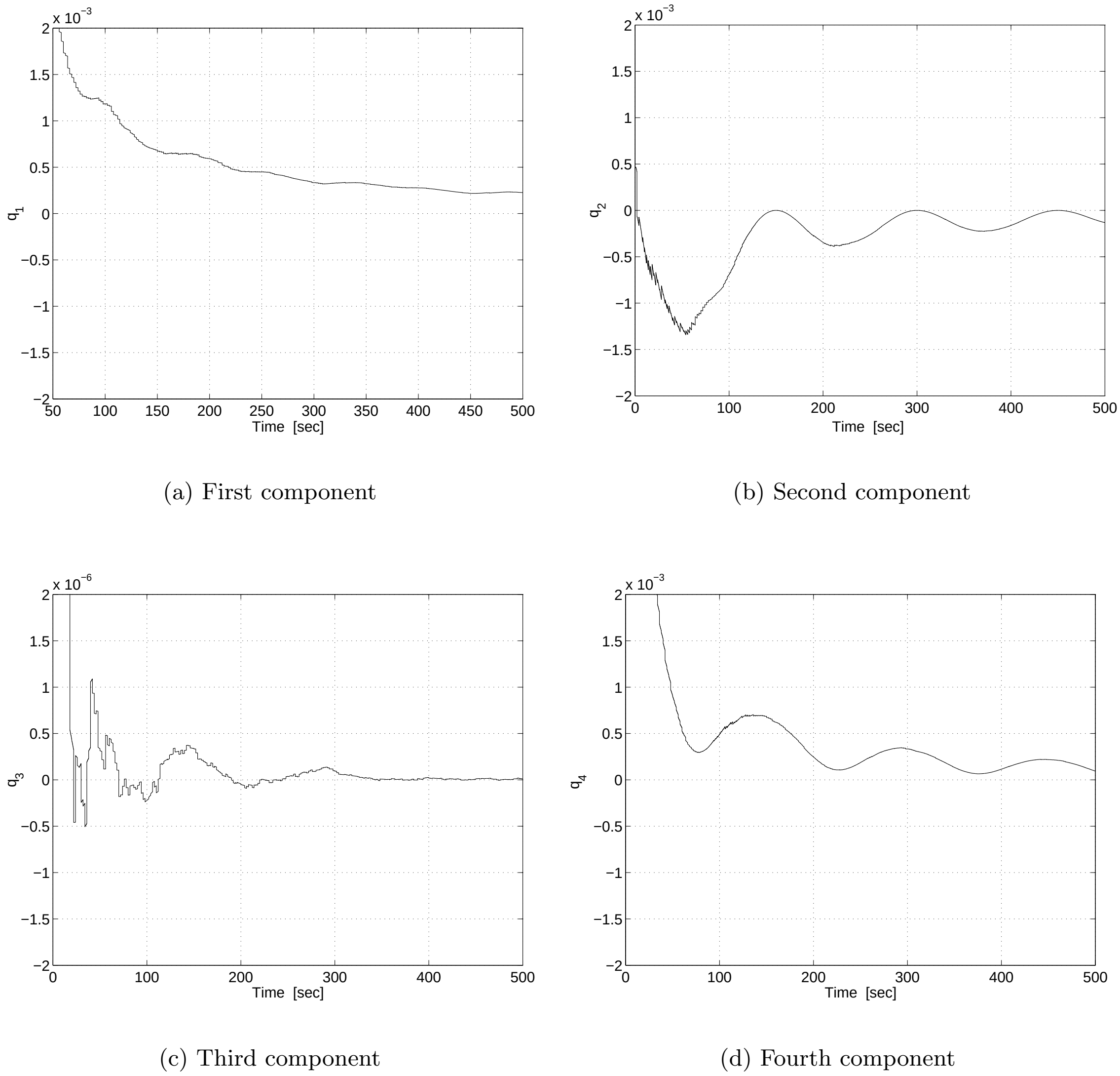
2.5 Simulation Study

Extensive Monte-Carlo simulations were performed in order to test the non-adaptive (nominal) and the adaptive estimators of the preceding sections.

2.5.1 Nominal Filter

A first series of simulations was run to test the quaternion filter in the case where the Truth and the Design models of the quaternion system were identical. This filter is *the nominal filter* presented in section 2.3. In the simulation, the reference coordinate system, $\mathcal{R}$, was assumed to be inertial. The body coordinate system, $\mathcal{B}$, was rotating with the following angular velocity, $\boldsymbol{\omega}^o_k = 0.01\ \sin(2\pi\, t_k/150) \cdot [1, 1, 1]^T$. Three body-mounted gyroscopes measured $\boldsymbol{\omega}^o_k$ at a sampling rate of 10 Hz. The gyros noises were Gaussian zero-mean white noises with a standard deviation of 200 mdeg/hr in each axis. Single vector observations were sequentially acquired at a rate of 0.5 Hz. The vector observation noise was a Gaussian zero-mean white-noise with an angular standard deviation of 1/200 rad. Each simulated time span was 500 sec. We ran 100 runs with different seeds and averaged the results at each time point to obtain ensemble averages and ensemble standard deviations.

The results for the nominal filter are summarized in Figures 2.1 to 2.4. In Figure 2.1 the time history of the Monte-Carlo average of the quaternion estimate is presented. Each



(a) First component

(b) Second component

(c) Third component

(d) Fourth component

Fig. 2.1: Monte-Carlo mean of the quaternion estimate in the nominal filter over 100 runs.

subfigure corresponds to one component. These figures show that the components of the estimated quaternion converge to very low steady-state levels (10^{-4}, 10^{-6}). As a result, the norm of the quaternion estimate is very small i.e. of the order of 10^{-4}, and not equal or even close to unity. The convergence of the estimated quaternion, $\widehat{\mathbf{q}}$, to the correct direction is illustrated in Figures 2.2 and 2.3. For the purpose of comparing $\widehat{\mathbf{q}}$ with the true quaternion, $\mathbf{q}$, we first normalize $\widehat{\mathbf{q}}$; that is, we compute $\widehat{\mathbf{q}}' = \widehat{\mathbf{q}}/\|\widehat{\mathbf{q}}\|$. Using the

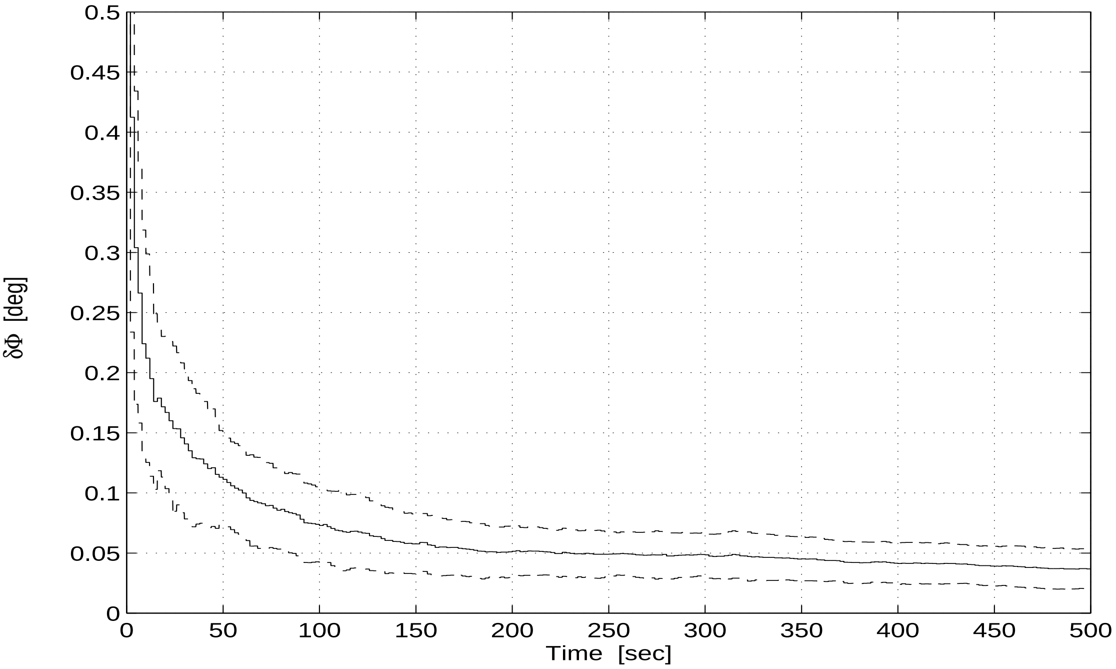
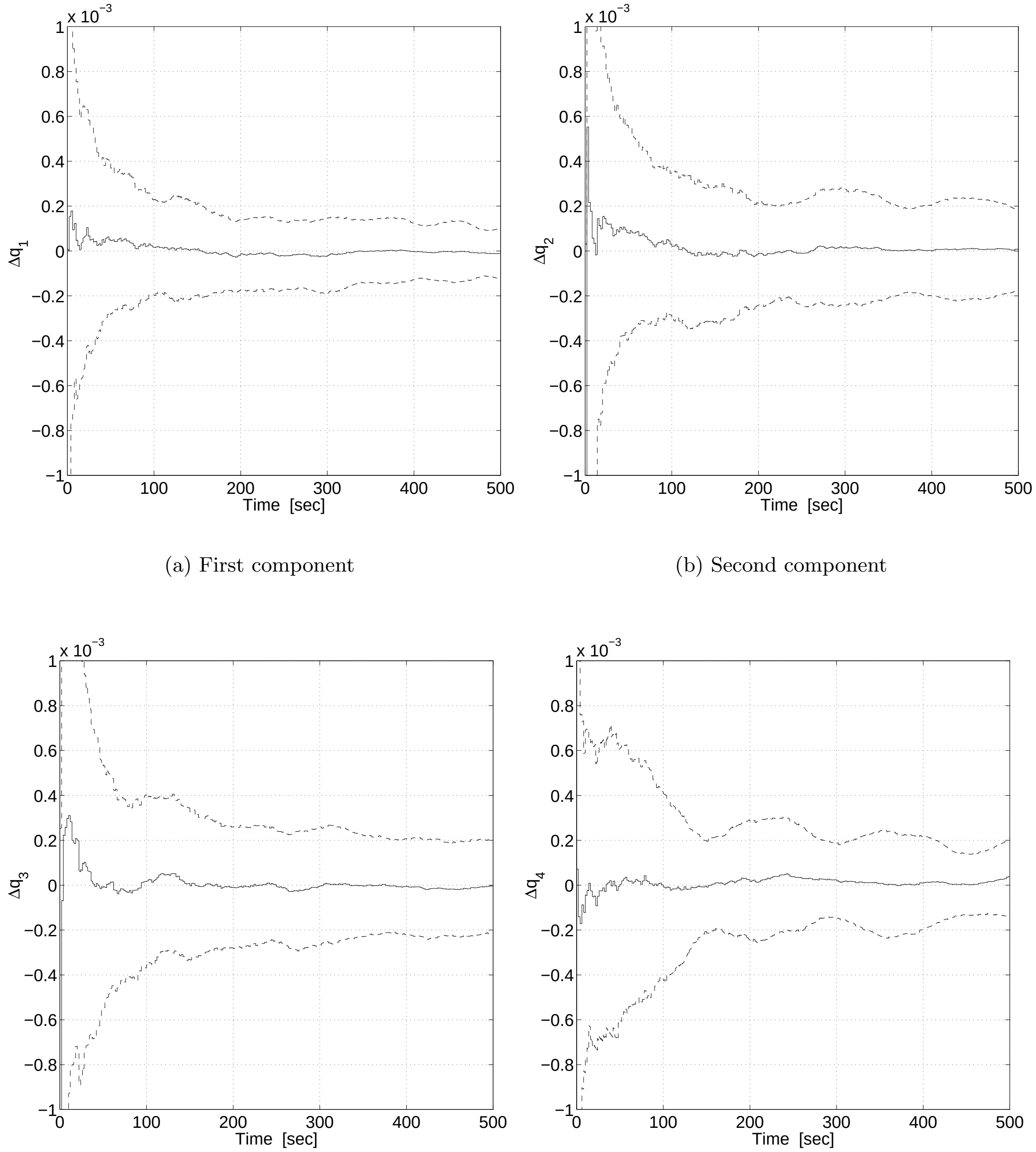


Fig. 2.2: Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error in the nominal filter over 100 runs.

unit-norm estimate $\widehat{\mathbf{q}}'$ we consider two types of estimation errors, which are commonly used in quaternion estimators. The first one is the so-called multiplicative estimation error, which is denoted here by $\boldsymbol{\delta q}$ and is defined as $\boldsymbol{\delta q} \overset{\triangle}{=} (\mathbf{q})^{-1} \otimes \widehat{\mathbf{q}}'$, where $\otimes$ and $\cdot^{-1}$ denote, respectively, the quaternion product and inverse [1, pp. 758,759]. The 4×1 unity vector $\boldsymbol{\delta q}$ is itself a quaternion of rotation that represents the small rotation which brings the axis of the estimated body frame onto those of the true body frame. From the scalar component of $\boldsymbol{\delta q}$, denoted by δq, we extract the value of the rotation angle, $\delta\phi$, using the known relation $\delta\phi = 2\ \arccos(\delta q)$. Figure 2.2 shows the time history of the Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error, $\delta\phi$, over the 100 runs. On this plot we observe a transient of approximately 50 sec, where $\delta\phi$ decreases from its initial value ($\simeq 10$ deg) down to 0.1 deg, and a steady-state stage where $\delta\phi$ is slowly decreasing and is eventually reaching 0.038 deg at the end of the simulation run. The Monte-Carlo standard deviation is approximately constant during the steady-state stage and is about 0.02 deg.



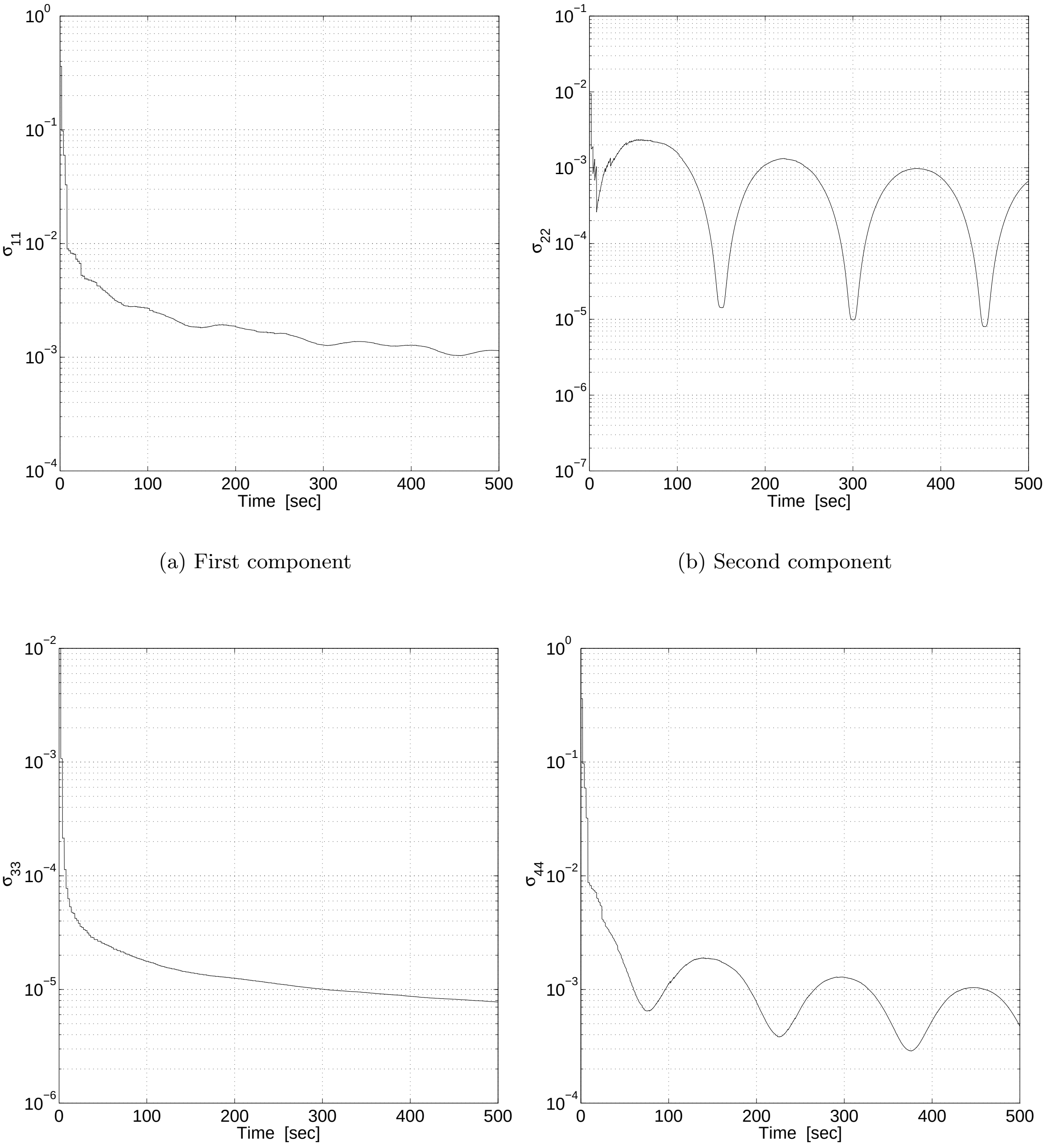
(a) First component

(b) Second component

(c) Third component

(d) Fourth component

Fig. 2.3: Monte-Carlo mean and $\pm 1\sigma$-envelope of the quaternion estimation error in the nominal filter over 100 runs.



(a) First component

(b) Second component

(c) Third component

(d) Fourth component

Fig. 2.4: Standard deviation of the estimation error in the nominal filter for a sample run.

The second type of estimation error is the additive estimation error, $\boldsymbol{\Delta q}$, which is defined as $\boldsymbol{\Delta q} \overset{\triangle}{=} \mathbf{q} - \widehat{\mathbf{q}}'$. Figure 2.3 depicts the time history of the Monte-Carlo mean and $\pm 1\sigma$ envelope of $\boldsymbol{\Delta q}$ over 100 runs. Each one of the four subfigures corresponds to a component of $\boldsymbol{\Delta q}$. The four components exhibit similar behaviors. Looking at the Monte-Carlo means we see that they all settle on zero after transition stages of about 50 sec. This shows the asymptotic unbiasedness of the estimate. All four Monte-Carlo standard deviations are slightly affected by the body dynamics, and are oscillating around steady-state values of $2\,10^{-4}$.

Figure 2.4 depicts the time histories of the estimation error standard deviations *as computed in the filter* during a single typical simulation run. The four filter standard deviations, which are denoted by σ_{11}, σ_{22}, σ_{33}, and σ_{44}, are obtained by computing the square of the four diagonal elements of the estimation error covariance matrix, $P_{k/k}$ (see Eq.2.54). All the filter standard deviations reach a steady-state behavior after a quick transient of about 10 sec. The body dynamics particularly affects the variations of σ_{22} and σ_{44} (Figs. 2.4b and 2.4d) while σ_{11} and σ_{33} seem to be monotonously decreasing (Figs. 2.4a and 2.4c).

2.5.2 Adaptive Filter

In order to test the adaptive filter we added a constant bias of 20 deg/hr to the outputs of the three gyroscopes. These biases were not added to the design-model which was implemented in the Kalman filter; thus the filter had the task of compensating for the unmodeled gyro biases by properly adapting the weight it gave to the process noise level. The adaptive algorithm was turned on at $t = 300$ sec, and operated autonomously thereafter. All the other conditions of simulation were identical to the case of the nominal filter.

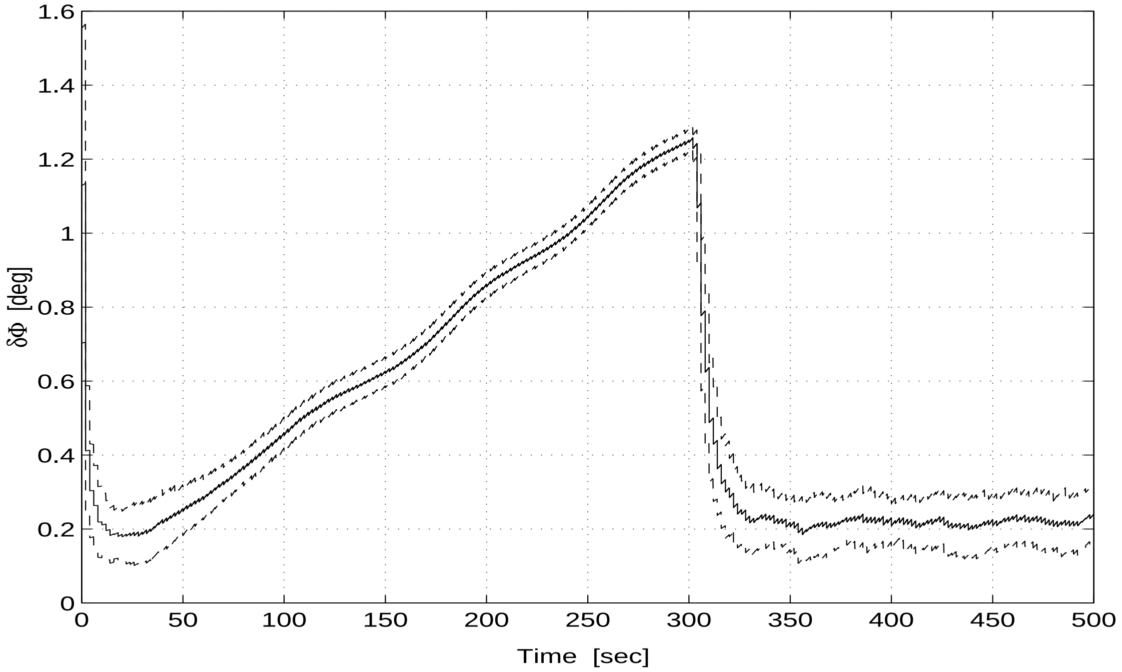
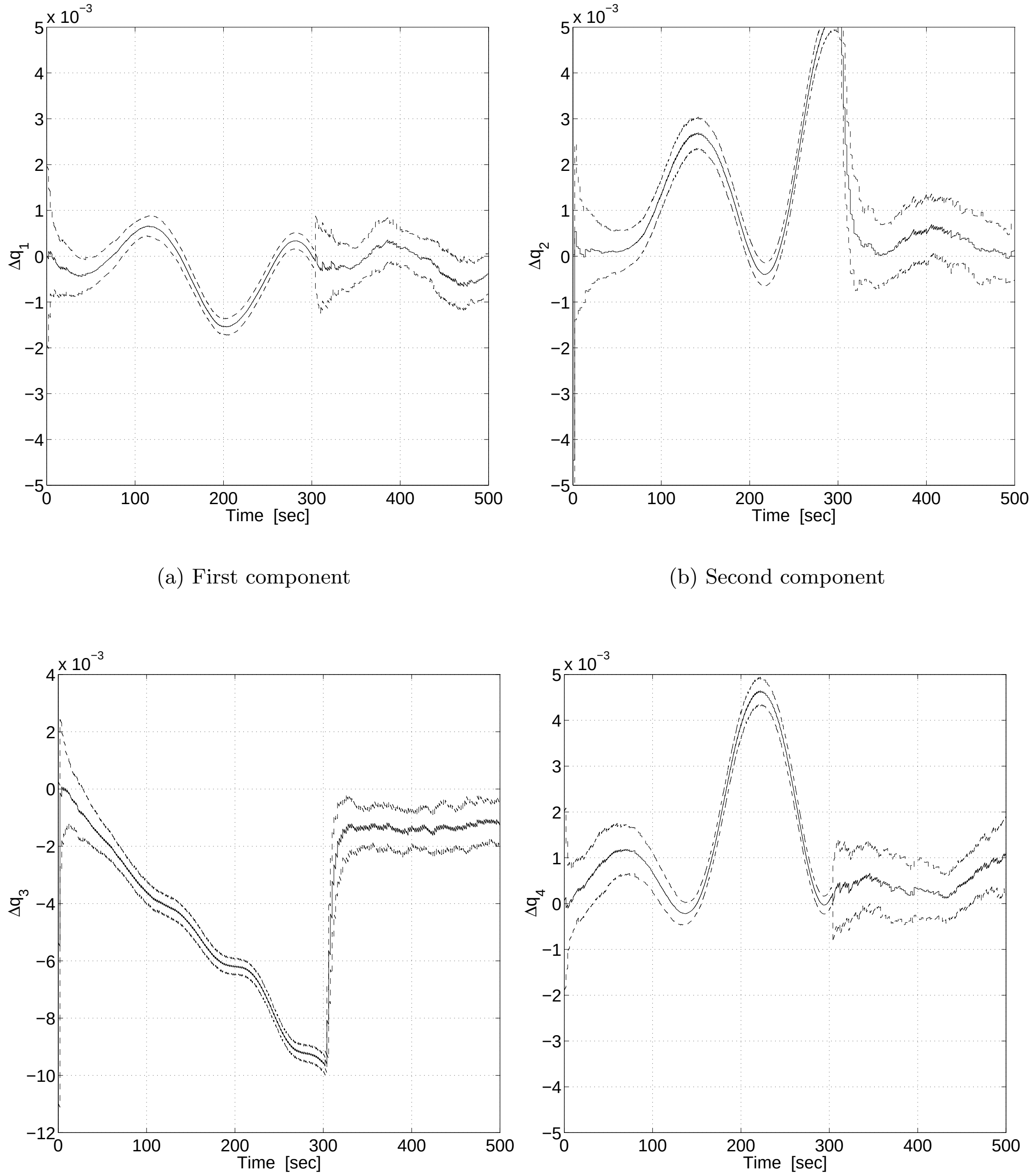


Fig. 2.5: Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error in the adaptive filter over 100 runs.

The results of 100 Monte-Carlo runs are presented in Figures 2.5 to 2.8. Figure 2.5 shows the time history of the Monte-Carlo mean and $\pm 1\sigma$ envelope of the angular estimation error, $\delta\phi$. We observe four distinct stages. First, there is a quick transient stage of approximately 20 sec where $\delta\phi$ decreases from its initial value down to 0.2 deg. At this stage the filter behaves like the nominal filter because the unmodeled gyro biases had no much considerable effect on the prediction error yet, and the measurement updates were still accurate enough to enforce the convergence of the estimate. Next we observe a divergence stage were $\delta\phi$ grows with the time at a rate equal to the value of the biases, 20 deg/hr. At $t = 300$ sec, we see another transient stage where $\delta\phi$ is brought from around 1.25 deg down to around 0.2 deg. This is the instant were the adaptive algorithm began to work. Finally, there was a steady-state stage where the filter maintained the Monte-Carlo average of $\delta\phi$ on a level of 0.21 deg, and the Monte-Carlo standard deviation at about 0.05 deg.

Figure 2.6 presents the performance of the adaptive filter when considering the additive



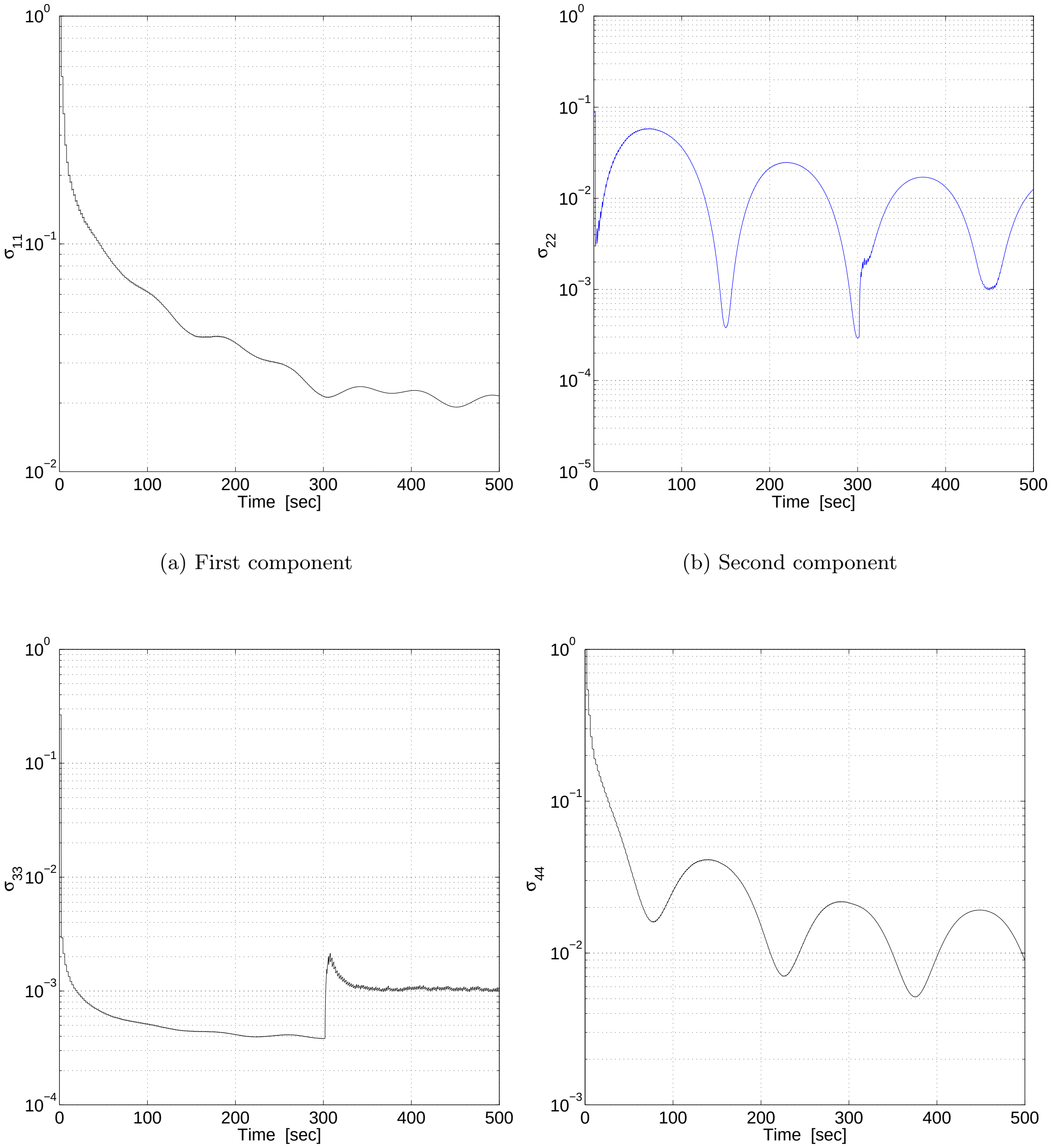
(a) First component

(b) Second component

(c) Third component

(d) Fourth component

Fig. 2.6: Monte-Carlo mean and $\pm 1\sigma$ envelope of the quaternion estimation error in the adaptive filter over 100 runs.



(a) First component

(b) Second component

(c) Third component

(d) Fourth component

Fig. 2.7: Standard deviation of the estimation error in the adaptive filter for a sample run.

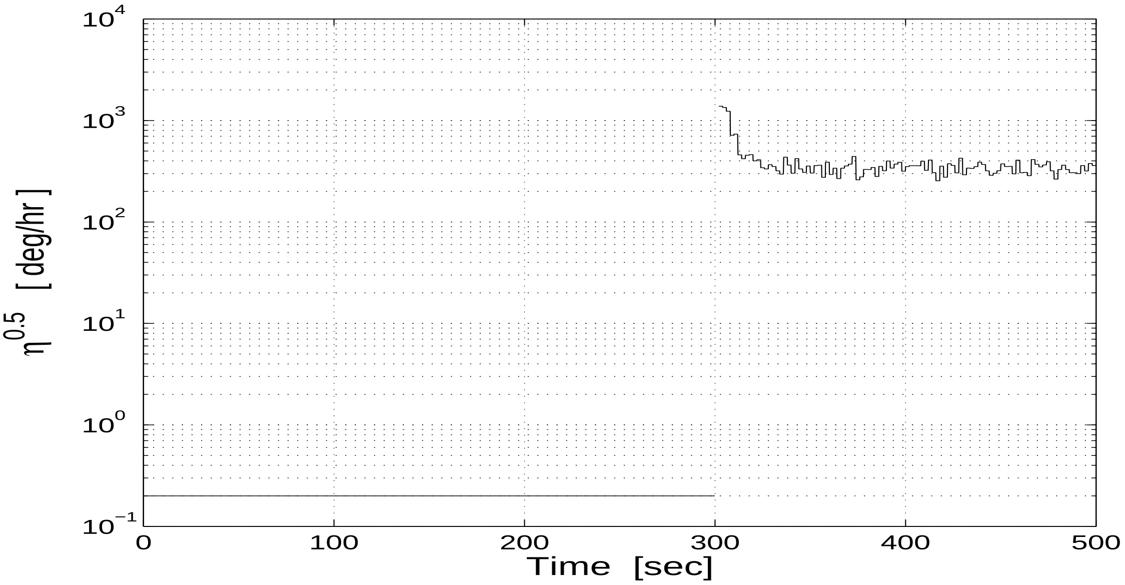


Fig. 2.8: Monte-Carlo mean of the adaptive gyro noise standard deviation over 100 runs.

estimation error, $\boldsymbol{\Delta q}$. The four subfigures, 2.6a to 2.6d, depict the time histories of the Monte-Carlo averages and $\pm 1\sigma$ envelopes of the four components of $\boldsymbol{\Delta q}$. All four plots show that during the non-adaptive stage the filter does not succeed in keeping the quaternion estimation error small. On the other hand, as soon as the adaptive filter is activated it maintains the estimation errors averages very close to zero, from the activation time $t = 300$ sec until the simulation end. Actually, these averages are not strictly zero, they are rather of the order of $5\,10^{-4}$. The Monte-Carlo standard deviations, though, show a steady-state values of $5\,10^{-4}$. It is clear that the divergence due to the unmodeled gyro biases has been avoided on the expense of the estimation performances.

Figure 2.7 present the standard deviations of the components of $\boldsymbol{\Delta q}$ as computed in the filter during a single typical run. We see that during the non-adaptive part of the run ($t \leq 300$ sec) the standard deviations are greater by two orders of magnitude than those of the nominal case (Fig. 2.4). This is so because the covariance computation in the filter depends on the estimated quaternion (see Eqs. 2.45, 2.50, and 2.55), and because here the estimated quaternion is of unit-norm. In contrast, the steady-state length of the estimated quaternion in the nominal filter is much smaller (about 10^{-4}), which yields smaller values in the filter standard deviations. We can see that the adaptive filter ($t \geq 300$ sec) affects the filter covariance computation by increasing the steady-state levels of the standard deviations, σ_{11}, σ_{22}, σ_{33}, while σ_{44} seems to be unaffected. This shows that the adaptive filter worked as expected: it compensated for the unmodeled biases by increasing the uncertainty in the a priori estimate. This allowed the filter to stay in an “watchfull” state, and to weigh more favorably the new incoming measurements.

Figure 2.8 shows the time history of the Monte-Carlo mean of the adaptive gyro noise standard deviation, denoted by $\sqrt{\eta}$, over 100 runs. Recall that this quantity is an optimal estimate of the gyro noise level, as computed by the filter using the assumptions presented in the preceding section. Before $t = 300$ sec, $\sqrt{\eta}$ equals 20 mdeg/hr, which is the standard deviation of the gyro output white noises. After $t = 300$ sec, we see that $\sqrt{\eta}$ reaches a maximum of about 1500 deg/hr before settling on a steady-state of about 350 deg/hr. We see then that the adaptive filter compensated for the unmodeled gyro biases of 20 deg/hr by increasing the filter process noise to a level of about 350 deg/hr.

2.6 Simulation Study Using GPS Measurements

This section presents an application of the novel Kalman filter for estimating the attitude-quaternion from vectorized GPS phase measurements.

Spacecraft attitude determination methods using Global Positioning System (GPS) carrier signals have been intensively investigated in recent years [47–50]. Much attention was given to concept, hardware and algorithm development, as well as to testing. Algorithms for GPS attitude determination can be classified into two main classes: 1. integer resolution and 2. attitude calculations. Several methods for integer resolution were presented in the literature (see [47, 51]). In this work we assume that the integer ambiguity is solved, and we are concerned only with the second part; namely, the attitude calculation. To make use of the novel Kalman filter, the GPS phase measurements have first to be converted to vector measurements. In the next section we review the method for vectorizing GPS phase measurements.

2.6.1 GPS Vectorized Observations

The current implementation calls for three antennas, which are designated as antenna 0, 1 and 2. The three antennas are so installed as to form a right angle with antenna 0 at the vertex. The antenna coordinate system is defined as follows. The origin of the system is at 0, the system x axis is along the line connecting antenna 0 and antenna 1, and the y axis is along the line connecting antenna 0 and antenna 2. The third axis is normal to the antenna's plane and is directed according to the right-hand rule. We assume the antenna system to be rigidly attached to a structure (body), on which a Cartesian coordinate frame (the body frame, $\mathcal{B}$) is defined. It is also assumed that the orientation of the antenna system with respect to $\mathcal{B}$ is known. In the following we assume, without loss of generality, that the antenna system and the body-frame coincide. Let us denote by $\mathcal{R}$ a Reference Cartesian coordinate frame. Consider a unit vector in the direction of an observed satellite. Let $\mathbf{b}^o$ and $\mathbf{r}$ denote the projections of this unit vector along $\mathcal{B}$ and $\mathcal{R}$ coordinates respectively. Then $\mathbf{b}^o$ and $\mathbf{r}$ satisfy

$$\mathbf{b}^o = A\ \mathbf{r} \tag{2.88}$$

where the matrix A is the attitude matrix; that is, the rotation matrix bringing the axes of $\mathcal{R}$ onto the axes of $\mathcal{B}$. The component b^o_j, $j = 1, 2$ of $\mathbf{b}^o$ is equal to $\cos(\alpha_j)$, where α_j is the angle between the line of sight to the GPS satellite and the axis (j) of $\mathcal{B}$:

$$b^o_j = \cos(\alpha_j) \tag{2.89}$$

The component b^o_j is related to the phase difference between simultaneous phase measurements at antenna 0 and antenna j of the carrier broadcasted by the GPS satellite. This phase difference ϕ_j is given in carrier wavelength and can be written as follows:

$$\phi_j = \phi'_j + N_j \tag{2.90}$$

where ϕ'_j is a fraction of wavelength, N_j is an integer. The phase measurement is performed using carrier-phase differentiating, which yields ϕ'_j, but there are techniques [52,53] for quick determination of N_j. Therefore, knowing ϕ'_j is equivalent to knowing ϕ_j, and it is assumed that ϕ_j is the measured quantity. It is easy to show that

$$b^o_j = \phi_j \left(\frac{\lambda}{a_j} \right) \tag{2.91}$$

where a_j is the distance between antenna 0 and antenna j and λ is the carrier wavelength. The vector measurement from GPS phase observables is a unit norm 3×1 column-matrix constructed as follows:

$$\mathbf{b}^o = \begin{bmatrix} b_1^o \\ b_2^o \\ \sqrt{1 - (\ b_1^o)^2 - (\ b_2^o)^2} \end{bmatrix} \tag{2.92}$$

where b_1^o and b_2^o are computed from Eq. (2.91). Note that the third component of $\mathbf{b}^o$ is chosen to be the positive square-root. This was done because only the signals of those GPS satellites that are above the antenna plane, and thus in the positive direction of the third axis of $\mathcal{B}$, are received by the antennas. The measurement of ϕ_j introduces measurement errors in b_j^o that stem from structure flexure, baseline uncertainties, electronic delays, multipath errors, etc. The measurement error is δb_j and is assumed to be additive:

$$b_j = b_j^o + \delta b_j \tag{2.93}$$

This yields the following vector measurement equation:

$$\mathbf{b} = A\ \mathbf{r} + \boldsymbol{\delta b} \tag{2.94}$$

Choosing the reference frame $\mathcal{R}$ to coincide with the Earth Centered Earth Inertial frame (ECEI) the vector $\mathbf{r}$ is easily computed because both the satellite and the vehicle positions are known in Earth coordinates and the Earth inertial rate of rotation is accurately known, too.

One recognizes in Equation (2.94) the standard measurement equation from which the novel quaternion measurement model was derived (see section 2.2).

2.6.2 Simulation Study

The present section present the results of the simulations that were performed in order to test the nominal (non-adaptive) and the adaptive estimators.

Nominal filter

A first simulation was run to test the quaternion filter in the case where the Truth and the Design-Models of the quaternion system were identical. This filter is *the nominal* *filter*. The body coordinates system $\mathcal{B}$ was assumed to be fixed with respect to the inertial coordinates system $\mathcal{R}$. The quaternion of rotation from $\mathcal{B}$ to $\mathcal{R}$ is $[0,\ -0.7071,\ 0,\ 0.7071]$. The GPS phase measurements were performed at a rate of 0.5 Hz. They were corrupted by a Gaussian zero-mean white noise with an equivalent angle standard deviation of 1 deg. The tested scenario assumed that two GPS satellites (SV) were tracked. During the first half-hour of the simulation, the tracked SVs had an angular separation of approximately 10 deg and were on a descending phase. At $t = 0.5$ hr one of the satellites disappeared while a third SV emerged in the field of view of the antennae. It was assumed that the switch between the tracked SVs is immediate. At $t = 0.5$ hr, the initial angular separation between the tracked SVs was 79.3 deg. Three body-mounted gyroscopes measured the angular velocity of $\mathcal{B}$ with respect to $\mathcal{R}$. Because system $\mathcal{B}$ did not rotate with respect to system $\mathcal{R}$ the gyro measurements contained only gyro noises. The gyro noises were Gaussian zero-mean white noises with a standard deviation of 0.2 mdeg/hr in each axis. The sampling rate of the gyro measurements was 10 Hz. The simulation time lasted two hours. The initial angular estimation error $\delta\phi_{0/0}$ was 120 deg. The angular estimation error at time t_k, $\delta\phi_{k/k}$, is defined as the angle of the rotation that brings the axes of the estimated body frame $\widehat{\mathcal{B}}$ onto those of the true body frame $\mathcal{B}$. This angle is obtained as follows. First, the quaternion of the rotation from $\widehat{\mathcal{B}}$ to $\mathcal{B}$, denoted by $\widetilde{\mathbf{q}}_{k/k}$, is evaluated, then the rotation angle $\delta\phi_{k/k}$ is computed from the scalar component of $\widetilde{\mathbf{q}}_{k/k}$ using the known relation $\delta\phi_{k/k} = 2\ \arccos\left[\widetilde{\mathbf{q}}_{k/k}(4)\right]$.

The results of Monte-Carlo simulations over 100 runs are summarized in Figures 2.9 and 2.10. Figure 2.9 shows the time history of the Monte-Carlo mean of the angular estimation error, $\delta\phi_{k/k}$. The mean error converges within approximately 500 seconds to a level of about 2 deg where it reaches a slow decrease rate. When the GPS receiver locks on the couple of SVs with high angular separation, at $t = 1800$ sec, there is a dramatic decrease in the estimation mean error. This decrease is a direct consequence of the more favorable geometry of the tracked SVs. Similarly to the Dilution Of Precision (DOP) that is defined for the navigation solution from GPS pseudo-range measurements, one could define an *ADOP*, that is, Attitude Dilution Of Precision, in the present attitude solution from GPS phase measurements. After 500 sec, the mean error reaches a steady-state around 0.05 deg. Figure 2.10 shows the time variations of the estimation error variances for a sample run,

P_{ii} for $i = 1, 2, 3, 4$, of the four quaternion components. The settling times of P_{11} and P_{33} are about 500 seconds, while those of P_{22} and P_{44} are close to a few seconds. The steady state values of P_{11} and P_{33} are 10^{-7} and 10^{-8} $[(\text{rad})^2]$. The steady state values of P_{22} and P_{44} are much higher, close to $0.2[(\text{rad})^2]$. The effect of the switch between tracked SVs is clearly seen on the graphs of P_{11} and P_{33}, while it is almost indistinct for P_{22} and P_{44}. One should remember that the first and third component of the true attitude quaternion, as well as the absolute values of the second and fourth components, are equal. This may explain the similarities of the related estimation error variances.

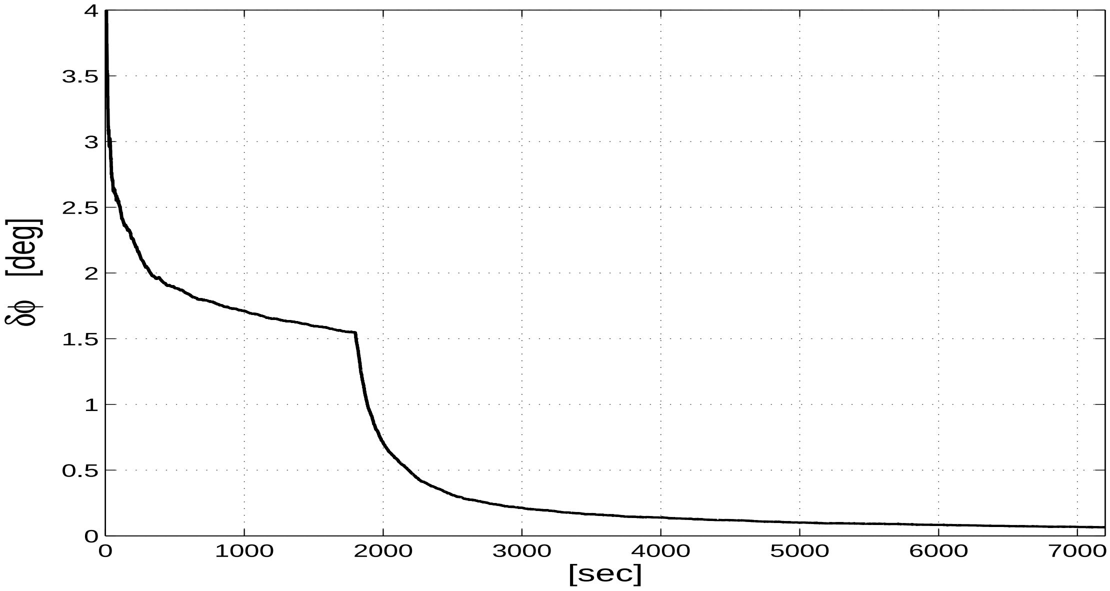
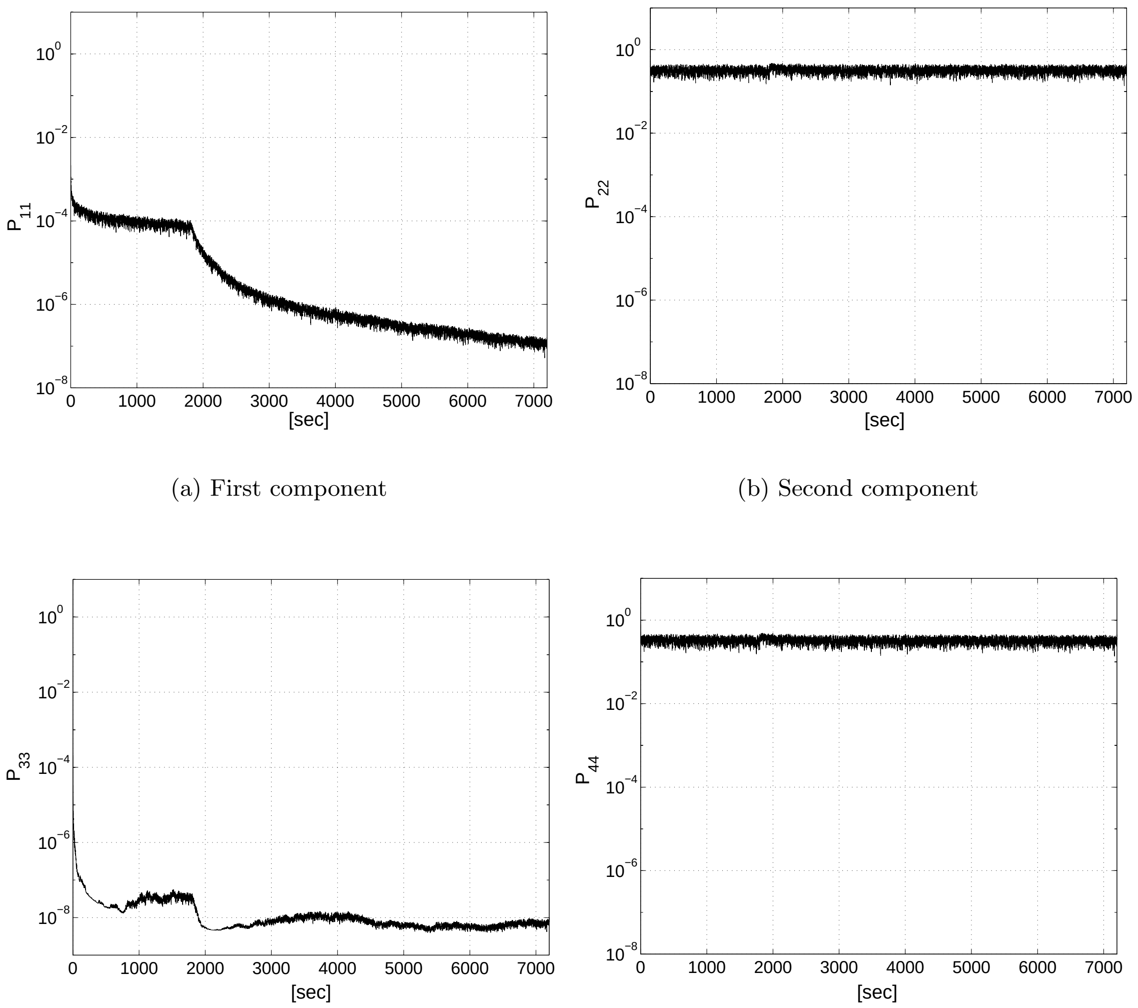


Fig. 2.9: GPS case. Monte-Carlo mean of the angular estimation error in the nominal filter over 100 runs.

Adaptive filter

A second simulation was run to test the adaptive filter performance. For this purpose a constant bias of 20 deg/hr along the three body axes was added to the gyro measurement in the Truth model but *not* in the Design model. The sampling rates of the GPS measurements and of the angular rate measurements are similar to those of the nominal case, that is, 0.5 Hz and 10 Hz respectively. The motivation was to allow the filter propagation error to develop between two measurement updates and to observe whether the adaptive



(a) First component

(b) Second component

(c) Third component

(d) Forth component

Fig. 2.10: GPS case. Quaternion estimation error variances in the nominal filter for a sample run.

estimation process converged. The sample mean of the N predicted residuals $\bar{\boldsymbol{\nu}}^N$ was computed from a sample of 10 one-step residuals. The adaptive algorithm was activated at time $t = 400\,$sec. The same configuration of GPS satellites as for the nominal filter was simulated. The switch between the tracked couples of SVs happened at $t = 2200\,$sec. The runs lasted one hour.

Figure 2.11 shows the time evolution of the Monte-Carlo mean angular estimation error over 100 runs. The upper plots shows the two phases of the estimation process: a diverging phase and a steady-state converging phase. During the first phase, from $t = 0\,\text{sec}$ to $t = 400\,\text{sec}$, the mean estimation error linearly diverges because of the unmodeled constant bias in the gyro output. At $t = 400\,\text{sec}$ the mean error reaches a maximum of 3.8 deg. At this instant, the adaptive algorithm is activated which causes a decrease of the mean error from 3.8 deg down to approximately a steady-state level of 0.25 deg with peaks up to 1 deg. The lower plot in Fig. 2.11 shows a zoom on the variations of the estimation error during the steady-state period. One can clearly see that in spite of the propagation of the error between two measurement updates, the adaptive filter succeeds in bounding the mean error on an average level of 0.25 deg. One notices that there is no effect of the switch between the tracked SVs (at $t = 2200\,\text{sec}$) on the mean estimation error.

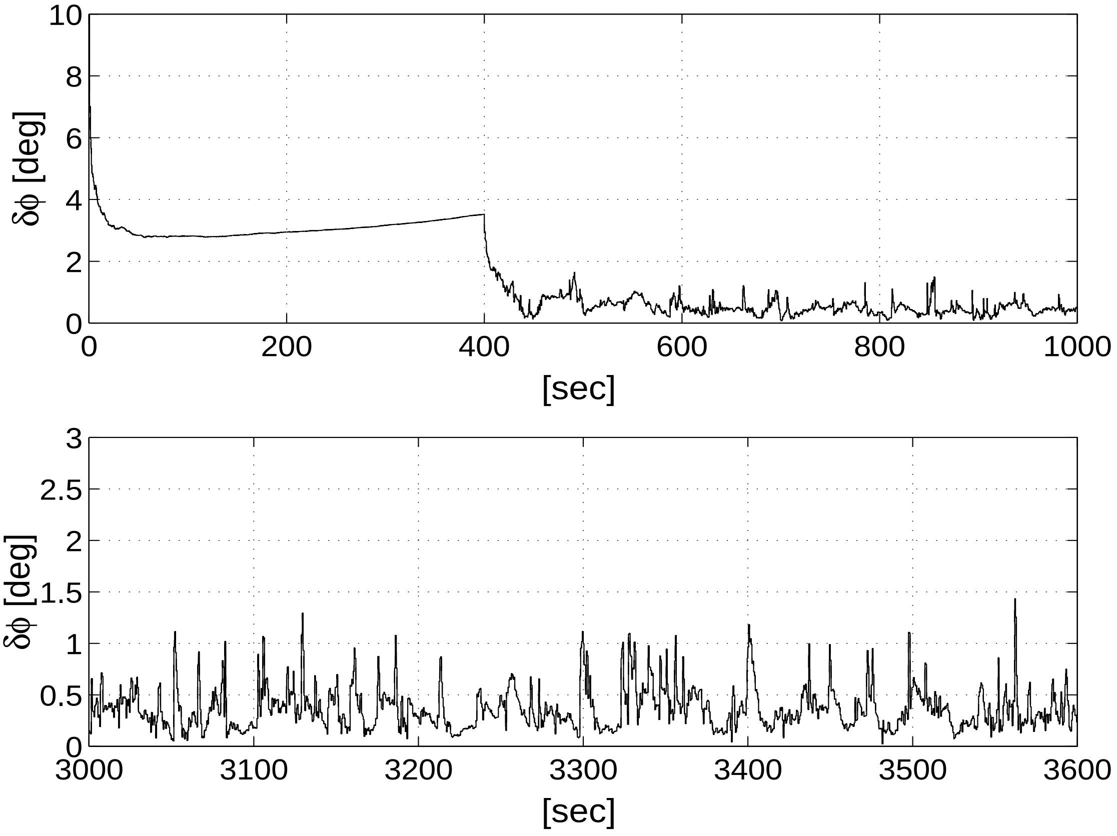


Fig. 2.11: GPS case. Monte-Carlo mean angular estimation error in the adaptive filter over 100 runs.

2.7 Concluding Remarks

In this chapter, a novel Kalman filter for estimating the quaternion of rotation using vector measurements was presented. The special feature of the algorithm was the measurement model. This model was obtained by a certain manipulation of the classical vector measurement equation which resulted in a pseudo-measurement equation featuring a linear signal term. This enabled the use of a linear Kalman filter. The measurement noise, however, was state dependent, but this did not impair the efficiency of the filter. The proposed KF did not enforce the normality of the quaternion, yet Monte-Carlo simulation runs showed that the quaternion estimate converged to the right direction in the four-dimensional space. Normalization was required only if the estimated quaternion was needed for vector transformation, for display and other purposes. This was done externally to the KF algorithm. To account for inaccuracy in the knowledge of the gyro noises statistical model, in particular to account for the case of constant biases in the gyro outputs, an adaptive version of the filter was introduced. In order to ensure the convergence of the adaptive filter, normalization of the quaternion estimate was necessary. Monte-Carlo simulations runs were made that demonstrated the efficiency of the adaptive filter as well. A case study was presented where the vector measurements were constructed using GPS phase measurements.

Chapter 3

Optimal-REQUEST Algorithm for Attitude Determination

This work was presented as Paper AIAA 2001-4153 at the 41th AIAA Guidance, Navigation, and Control Conference, Montreal, Aug. 2001 [54].

3.1 Introduction

In 1996, REQUEST was presented in [24]. As mentioned in Chapter 1, this algorithm belongs to the family of attitude determination estimators that originates from Wahba's problem. REQUEST is a recursive algorithm for least squares fitting of the attitude quaternion of a rigid body to vector measurements. It relies on the ability to perfectly propagate the so-called K-matrix given noise-less angular rates, and is, therefore, suboptimal when filtering propagation noises. The algorithm presented in this chapter is an optimized REQUEST procedure, which optimally filters measurement as well as propagation noises. The solution approach is based on state space modelling of the K-matrix system and uses Kalman filtering techniques in order to estimate the optimal K-matrix. Then, the attitude quaternion is extracted from the estimated K-matrix using a classical method. The special case of a zero-mean white propagation noise is considered when developing a nominal version of the optimal algorithm. The case of a biased noise is treated too by applying an adaptive filtering technique.

In section 3.2, we briefly review Wahba's problem, the **q**-method, and the REQUEST algorithm. The problem that is solved in the present work is stated in section 3.3. Then, in section 3.4 we lay the foundation necessary for the development of an optimized REQUEST algorithm. The Optimal-REQUEST algorithm is presented in section 3.5. In section 3.6 we present two adaptive versions of Optimal-REQUEST. The first algorithm is an adaptive *measurement noise* estimator, the second algorithm is an adaptive *process noise* estimator. In section 3.7 we demonstrate the performance of the non-adaptive and of the adaptive filters through extensive Monte-Carlo simulations. Finally, we present our conclusions in section 3.8.

3.2 Preliminaries

3.2.1 Wahba's Problem

As mentioned in chapter 1, the attitude determination problem from vector observations was first formulated by Wahba in 1965 as a least squares estimation problem in the so-called Wahba problem [5]:

Given the two sets of n vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ *and* $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, $n \geq 2$, *where each pair* $(\mathbf{r}_i, \mathbf{b}_i)$ *corresponds to a generalized vector,* $\mathbf{x}_i$, *find the proper orthogonal matrix, A, which brings the first set into the best least squares coincidence with the second. That is, find A which minimizes*

$$\sum_{i=1}^{n} \| \mathbf{b}_i - A\,\mathbf{r}_i \|^2 \tag{3.1}$$

subject to the constraints $A^T A = I_3$ *and* $\det(A) = 1$.

3.2.2 The q-method

As reported in [3, pp. A.1-A.11] the famous **q**-method was devised by Davenport in 1968. It provides a solution of Wahba's problem in terms of the quaternion of rotation. The rational of the **q**-method is based on the following identity

$$\frac{1}{2} \sum_{i=1}^{n} a_i \| \mathbf{b}_i - A\,\mathbf{r}_i \|^2 = 1 - \mathbf{q}^T K \mathbf{q} \tag{3.2}$$

where the K matrix in the right hand-side of Eq. (3.2) is function of the vector measurements as shown next. The algorithm is as follows. First, a batch of n simultaneous observations $\mathbf{b}_i$, $\mathbf{r}_i$, $i = 1, 2, \ldots, n$, is acquired. Let the corresponding weights in the weighted Wahba's cost function be denoted by a_i, and assume, with no loss of generality, that they add to one; that is, $\sum_{i=1}^{n} a_i = 1$. Then, the 3×3 matrices B and S, the 3×1 column matrix $\mathbf{z}$, and the scalar σ are defined as

$$\begin{array}{ll} B \triangleq \sum_{i=1}^{n} a_i \, \mathbf{b}_i \, \mathbf{r}_i^T & S \triangleq B + B^T \\ \mathbf{z} \triangleq \sum_{i=1}^{n} a_i \, \mathbf{b}_i \times \mathbf{r}_i & \sigma \triangleq tr(B) \end{array} \tag{3.3}$$

where $tr(\cdot)$ denotes the trace operator. In the next step, the 4×4 symmetric matrix K of Eq. (3.2) is computed as

$$K = \begin{bmatrix} S - \sigma I_3 & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix} \tag{3.4}$$

Davenport showed that the optimal quaternion estimate is the eigenvector of K that belongs to its largest positive eigenvalue. The highlights of the **q**-method are that it optimally filters the measurement noises, and that it requires neither linearization of some equation model nor an initial quaternion estimate. The **q**-method, however, as any single-frame attitude estimator, is a memoryless algorithm; that is, the information contained in measurements of past attitudes is lost.

3.2.3 The REQUEST Algorithm

As mentioned before, Reference [24] introduced a discrete-time recursive algorithm named REQUEST that propagates and updates the K-matrix. This algorithm is summarized next. Let Φ_k denote the 4×4 orthogonal transition matrix in the quaternion dynamics difference equation

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k \tag{3.5}$$

According to the REQUEST algorithm, the propagation of the K-matrix, $K_{k/k}$, from t_k to t_{k+1}, which yields $K_{k+1/k}$, is given by

$$K_{k+1/k} = \Phi_k \, K_{k/k} \, \Phi_k^T \tag{3.6}$$

Given a single observation at time t_{k+1}, $\mathbf{r}_{k+1}$ and $\mathbf{b}_{k+1}$, one can construct the corresponding incremental K-matrix, denoted by δK_{k+1}, as follows. First define

$$\begin{array}{ll} B_{k+1} \triangleq a_{k+1} \, \mathbf{b}_{k+1} \, \mathbf{r}_{k+1}^T & S_{k+1} \triangleq B_{k+1} + B_{k+1}^T \\ \mathbf{z}_{k+1} \triangleq a_{k+1} \, \mathbf{b}_{k+1} \times \mathbf{r}_{k+1} & \sigma_{k+1} \triangleq tr\left(B_{k+1}\right) \end{array} \tag{3.7}$$

then compute δK_{k+1} as

$$\delta K_{k+1} = \frac{1}{a_{k+1}} \begin{bmatrix} S_{k+1} - \sigma_{k+1} I_3 & \mathbf{z}_{k+1} \\ \mathbf{z}_{k+1}^T & \sigma_{k+1} \end{bmatrix} \tag{3.8}$$

where a_{k+1} is the scalar weighting coefficient of the $(k+1)^{th}$ observation. The update stage of REQUEST is of the form

$$K_{k+1/k+1} = \frac{\rho_{k+1} m_k}{m_{k+1}} K_{k+1/k} + \frac{1}{m_{k+1}} \delta K_{k+1} \tag{3.9}$$

where m_k and m_{k+1} are scalar coefficients that keep the largest eigenvalue of $K_{k+1/k+1}$ close to one. The coefficient ρ_{k+1} is a fading memory factor, which is equal to 1 if Φ_k is error-free, and is otherwise set between 0 and 1 according to the propagation noise. (Note that although REQUEST can handle a batch of new measurements, for our purpose we consider only one new measurement.)

3.3 Problem Statement

The choice of ρ_{k+1} is heuristic, making the filter REQUEST suboptimal. Moreover ρ_{k+1} is determined by only considering the accuracy of the propagation stage and not of the measurement. We wish to optimize REQUEST by computing an optimal value of the parameter ρ_{k+1} in the update stage of REQUEST. The optimal value of ρ_{k+1} is the value that achieves an optimal blending of the a priori estimate of the K-matrix, $K_{k+1/k}$, and its new observation, δK_{k+1}.

3.4 Mathematical Model

3.4.1 Solution Approach

The approach to computing an optimal gain, ρ_k^*, consists of embedding REQUEST in the framework of Kalman filtering. With this purpose in mind, we modify the formulation of the attitude estimation problem. The central idea is that we would know the value of the true attitude quaternion at any time if we knew the value of the true K-matrix at any time. The true K-matrix means the K-matrix that does not contain any error, neither from vector measurement noise, nor from propagation noise. Because we cannot know the true K-matrix, we propose to estimate it in some optimal way.

A notation that is consistent with that of REQUEST is used here. Denote by $K^o_{i/j}$ the true K-matrix related to the attitude at t_i that is based on the ideal noise-free vector measurements up to t_j. The *estimate* of this matrix is denoted by $K_{i/j}$. Similarly, the true K-matrix that is based on the noise-free vector measurements acquired at t_i is denoted by δK^o_i. The measured K-matrix, which is computed using the *noisy* vector measurements acquired at t_i, is denoted by δK_i.

In our approach we consider the true K-matrix as a state-matrix variable and derive the dynamics and measurement equations that describe the state K-matrix system. To the conventional model of a state-space system, which includes the dynamics and measurement equations, we add another equation. This supplementary equation models a deterministic linear combination of the noise-free K-matrices $K^o_{k+1/k}$ and δK^o_{k+1} at time t_{k+1}. As will be shown, this combination yields another noise-free K-matrix, $K^o_{k+1/k+1}$, which is the K-matrix related to the true attitude at t_{k+1} and computed using all the imaginary noise-free vector measurements up to t_{k+1}. (We refer to these vectors as imaginary vectors since in reality only the noisy measurements are available to us.) Note that the structure of the computation of $K^o_{k+1/k+1}$ fits the structure of the update stage equation in REQUEST (Eq. 3.9). The reason for this computation will become clear when defining the estimation errors and deriving their recursive equations. As will be shown later, the updated estimation error is used to define a special cost function. This cost function will be minimized with respect to the scalar gain ρ_k, yielding the sought optimal gain ρ^*_k. This gain will be used in the update stage of the K-matrix estimation process, and, finally, an estimate of the attitude quaternion will be computed as the eigenvector that belongs to the largest eigenvalue of the updated K-matrix estimate.

3.4.2 Measurement Equation

The measurement equation of the K-matrix is derived as follows. Consider the measurement equation for a single vector observation at time t_{k+1}, and assume that the reference unit vector $\mathbf{r}_{k+1}$ is known exactly. Denoting by A_{k+1} the attitude matrix at t_{k+1}, and by

$\boldsymbol{\delta b}_{k+1}$ the noise vector that corrupts the measurement $\mathbf{b}_{k+1}$, we have

$$\mathbf{b}_{k+1} \;=\; A_{k+1}\,\mathbf{r}_{k+1} + \boldsymbol{\delta b}_{k+1} \tag{3.10}$$

Define the following quantities

$$\begin{array}{ll} B_b \triangleq a_{k+1}\,\boldsymbol{\delta b}_{k+1}\,\mathbf{r}^T_{k+1} & \mathcal{S}_b \triangleq {B_b}^T + B_b \\ \mathbf{z}_b \triangleq a_{k+1}\,\boldsymbol{\delta b}_{k+1} \times \mathbf{r}_{k+1} & \sigma_b \triangleq tr(\,B_b) \end{array} \tag{3.11}$$

then using these quantities we can define a 4 × 4 symmetric matrix, denoted by V_{k+1}, as follows:

$$V_{k+1} \;=\; \frac{1}{a_{k+1}} \begin{bmatrix} \mathcal{S}_b - \sigma_b\, I_3 & \mathbf{z}_b \\ \mathbf{z}^T_b & \sigma_b \end{bmatrix} \tag{3.12}$$

It can be shown that V_{k+1} is the error in the δK measurement equation (see Eqs. 3.7 and 3.8) at t_{k+1} because

$$\delta K_{k+1} = \delta K^o_{k+1} + V_{k+1} \tag{3.13}$$

where, as mentioned before, δK_{k+1} and δK^o_{k+1} are, respectively, the noisy and the noise-free K-matrices of the new vector measurements available at t_{k+1}. Thus, δK_{k+1} is computed using the noisy observation, $\mathbf{b}_{k+1}$, $\mathbf{r}_{k+1}$, (see Eqs. 3.7 and 3.8), while δK^o_{k+1} contains the imaginary noise-free vector measurements at t_{k+1}; that is, δK^o_{k+1} is expressed using Eqs. (3.7) and (3.8) for the noise-free observations (i.e., $\boldsymbol{\delta b} = \mathbf{0}$). Note that V_{k+1} is a linear function of $\boldsymbol{\delta b}_{k+1}$ and $\mathbf{r}_{k+1}$, and, since $\boldsymbol{\delta b}_{k+1}$ is random, V_{k+1} is also random. The linearity of V_{k+1} in $\boldsymbol{\delta b}_{k+1}$ will be used to derive a stochastic model for V_{k+1} from the stochastic model of $\boldsymbol{\delta b}_{k+1}$.

3.4.3 Process Equation

Let us denote by $\mathcal{B}_k$ and $\mathcal{R}$, respectively, the body frame at time t_k and the constant reference frame. Let the 3 × 1 vector $\boldsymbol{\omega}^o_k$ denote the body axes angular rate vector of $\mathcal{B}_k$ with respect to $\mathcal{R}$, and let $K^o_{0/0}$ denote the noise-free (true) K-matrix at time t_0. If $K^o_{0/0}$ and $\boldsymbol{\omega}^o_k$ are known, the true K-matrix can be propagated using

$$K^o_{k+1/k} \;=\; \Phi^o_k\; K^o_{k/k}\; \Phi^{oT}_k \tag{3.14}$$

where $K^o_{k/k}$ is related to the true attitude at t_k and contains all the noise-free observations up to t_k, $K^o_{k+1/k}$ is related to the true attitude at t_{k+1} and contains all the noise-free observations up to t_k, and Φ^o_k is the 4×4 transition matrix corresponding to $\boldsymbol{\omega}^o_k$ (see Eq. 3.5). In practice $\boldsymbol{\omega}^o_k$ is not known but is, rather, measured (or estimated). The transition matrix, Φ_k, which is computed from the measured angular rate, $\boldsymbol{\omega}_k$, contains an error term, $\Delta\Phi_k$; that is

$$\Phi^o_k = \Phi_k - \Delta\Phi_k \tag{3.15}$$

Using Eq. (3.15) in Eq. (3.14) yields

$$K^o_{k+1/k} = \Phi_k \, K^o_{k/k} \, \Phi_k + W^o_k \tag{3.16}$$

where

$$W^o_k \triangleq - \Phi_k \, K^o_{k/k} \, \Delta\Phi^T_k - \Delta\Phi^T_k \, K^o_{k/k} \, \Phi^T_k + \Delta\Phi_k \, K^o_{k/k} \, \Delta\Phi^T_k \tag{3.17}$$

The expression for W^o_k in Eq. (3.17) is exact but not useful since we cannot infer from it any statistical property of W^o_k. To determine a useful approximation to W^o_k we assume that the error in $\boldsymbol{\omega}_k$ is additive. Denoting this error by $\boldsymbol{\epsilon}_k$ we have

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}^o_k + \boldsymbol{\epsilon}_k \tag{3.18}$$

Based on this model we expand the expression for W^o_k in a Taylor series about $\boldsymbol{\omega}^o_k$, and, assuming that $\| \boldsymbol{\epsilon}_k \|$ and Δt are small enough, retain only the first order term in $\boldsymbol{\epsilon}_k$ and Δt. Let $B^o_{k/k}$ denote the true B-matrix at time t_k (see Eq. 3.3); that is, $B^o_{k/k}$ corresponds to the error-free vector observations up to t_k. Defining B_ϵ, S_ϵ, σ_ϵ and $\mathbf{z}_\epsilon$ as

$$\begin{array}{ll} B_\epsilon \triangleq [\boldsymbol{\epsilon}_k \times] \; B^o_{k/k} & S_\epsilon \triangleq B_\epsilon + B_\epsilon{}^T \\ [\mathbf{z}_\epsilon \times] \triangleq B_\epsilon{}^T - B_\epsilon & \sigma_\epsilon \triangleq tr(B_\epsilon) \end{array} \tag{3.19}$$

the first order approximation in $\boldsymbol{\epsilon}_k$ for W^o_k is

$$W_k = \begin{bmatrix} S_\epsilon - \sigma_\epsilon I_3 & \mathbf{z}_\epsilon \\ \mathbf{z}^T_\epsilon & \sigma_\epsilon \end{bmatrix} \Delta t \tag{3.20}$$

where $\Delta t = t_{k+1} - t_k$. The proof of Eq. (3.20) is presented in Appendix B.1. To summarize, we have derived an approximate propagation equation of the true K-matrix, which is

$$K^o_{k+1/k} = \Phi_k \, K^o_{k/k} \, \Phi^T_k + W_k \tag{3.21}$$

where Φ_k corresponds to $\boldsymbol{\omega}_k$ and W_k is an additive matrix noise. The initial true K-matrix is denoted by $K^o_{0/0}$. The expression for W_k is provided in Eqs. (3.19) and (3.20). From these equations it is realized that W_k contains the noise vector $\boldsymbol{\epsilon}_k$ as well as all the noise-free vector pairs $(\mathbf{b}^o, \mathbf{r})$ up to t_k. Note that W_k is random since $\boldsymbol{\epsilon}_k$ is random. Also note that W_k is linear in the error $\boldsymbol{\epsilon}_k$. This property will become useful when deriving a stochastic model for W_k. Extensive simulations showed that the propagation model described by Eq. (3.21), when W_k rather than W_k^o was used, was a good approximation to the exact propagation model described by Eq. (3.16).

3.4.4 Pseudo-process Equation

As mentioned earlier, a supplementary equation is added to the K-matrix model equations (3.13), (3.21). Motivated by the REQUEST algorithm (see Eq. 3.9) we define the following L matrix

$$L \overset{\triangle}{=} (1-\alpha_{k+1})\frac{m_k}{m_{k+1}} K^o_{k+1/k} + \alpha_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\delta K^o_{k+1} \tag{3.22}$$

where α_{k+1} is any real number in the interval $[0,1)$, δm_{k+1} is a positive weight, and the scalars m_{k+1} are recursively computed by $m_{k+1} = (1-\alpha_{k+1})m_k + \alpha_{k+1}\delta m_{k+1}$ for $k = 0, 1, \ldots$ and $m_0 = \delta m_0$. Note that Eq. (3.22), which we call *pseudo-process equation*, has a structure which is similar to that of the update stage of REQUEST, given in Eq. (3.9). The pseudo-process equation will be central in the development of Optimal-REQUEST. Next we show that L, the result of the pseudo-process equation (3.22), is a legitimate K-matrix.

Proposition. *For any value of* α_{k+1} *in the interval* $[0, 1)$ *the matrix* L *given in Eq.* (3.22) *is a K-matrix related to the true attitude at time* t_{k+1}.

Proof. By assumption, the two matrices $K^o_{k+1/k}$ and δK^o_{k+1} contain error-free vector observations, therefore one of their eigenvectors is the true attitude quaternion $\mathbf{q}_{k+1}$, and it belongs to their maximal eigenvalue, which can be made equal to one by proper scaling. Thus,

$$K^o_{k+1/k}\mathbf{q}_{k+1} = \mathbf{q}_{k+1} \tag{3.23}$$

$$\delta K^o_{k+1}\mathbf{q}_{k+1} = \mathbf{q}_{k+1} \tag{3.24}$$

Using Eq. (3.22) and the last two equations, we find that

$$\begin{aligned}
L\mathbf{q}_{k+1} &= \left[\left(1-\alpha_{k+1}\right)\frac{m_k}{m_{k+1}}K^o_{k+1/k} + \alpha_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\delta K^o_{k+1}\right]\mathbf{q}_{k+1} \\
&= \left(1-\alpha_{k+1}\right)\frac{m_k}{m_{k+1}}K^o_{k+1/k}\mathbf{q}_{k+1} + \alpha_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\delta K^o_{k+1}\mathbf{q}_{k+1} \\
&= \left(1-\alpha_{k+1}\right)\frac{m_k}{m_{k+1}}\mathbf{q}_{k+1} + \alpha_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\mathbf{q}_{k+1} \\
&= \frac{(1-\alpha_{k+1})m_k + \alpha_{k+1}\delta m_{k+1}}{m_{k+1}}\mathbf{q}_{k+1} \\
&= \mathbf{q}_{k+1} \qquad (3.25)
\end{aligned}$$

From Eq. (3.25) we conclude that $\mathbf{q}_{k+1}$ is an eigenvector of L associated with the eigenvalue 1. Consequently, the matrix L is a legitime K-matrix. □

As stated in the Proposition, the matrix L is related to the true attitude at time t_{k+1}; this matrix is indeed a correct K-matrix at time t_{k+1} containing all the noise-free vector measurements up to time t_{k+1}. Adopting a consistent notation we denote this matrix by $K^o_{k+1/k+1}$. The pseudo-process equation is written here again

$$K^o_{k+1/k+1} = (1-\alpha_{k+1})\frac{m_k}{m_{k+1}}K^o_{k+1/k} + \alpha_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\delta K^o_{k+1} \qquad (3.26)$$

It should be emphasized that Eq. (3.26) was developed in order to match the structure of the update stage of REQUEST. As will be seen later, this matching is crucial for defining the estimation errors, designing a cost function and, finally, minimizing this cost function with respect to the gain. The pseudo-process of Eq. (3.26), and the process and measurement equations (Eqs. 3.21 and 3.13), which are

$$K^o_{k+1/k} = \Phi_k K^o_{k/k}\Phi_k^T + W_k \qquad (3.27)$$

$$\delta K_{k+1} = \delta K^o_{k+1} + V_{k+1} \qquad (3.28)$$

(with an initial condition $K^o_{0/0}$), form the model for the K-matrix system.

3.4.5 Stochastic Models

The purpose of this subsection is to describe the stochastic models of the system noise matrices W_k and V_k. In order to derive the stochastic models for W_k and V_k, we need the stochastic models for $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\delta b}_k$. For the moment, only basic models are considered in this work; thus, the vector process $\boldsymbol{\epsilon}_k$ is modelled as a zero-mean white noise vector process whose components are identically distributed with variance η_k; that is

$$E\left[\boldsymbol{\epsilon}_k\right] = \mathbf{0} \qquad E\left[\boldsymbol{\epsilon}_k\, \boldsymbol{\epsilon}_{k+i}^T\right] = \eta_k\, I_3\, \delta_{k,k+i} \tag{3.29}$$

for $k = 1, 2, \ldots$, and $\delta_{k,k+i}$ is Kronecker's delta function. Assuming that the unit vector measurements, $\mathbf{b}_k$, are axi-symmetrically distributed about their true value, we employ a unit vector error model [17] that provides approximate but quite accurate expressions for the mean and the covariance of $\boldsymbol{\delta b}_k$. The first and second moments of this model are

$$E\left[\boldsymbol{\delta b}_k\right] = \mathbf{0} \qquad E\left[\boldsymbol{\delta b}_k\, \boldsymbol{\delta b}_{k+i}^T\right] = \mu_k \left(I_3 - \mathbf{b}_k\, \mathbf{b}_{k+i}^T\right) \delta_{k,k+i} \tag{3.30}$$

for $k = 1, 2, \ldots$, where μ_k is the variance of the component of $\mathbf{b}_k$ along a direction normal to $E\left[\mathbf{b}_k\right]$. Furthermore, it is assumed that $\boldsymbol{\delta b}_k$ and $\boldsymbol{\epsilon}_k$ are mutually uncorrelated. The exact expressions for the two moments can be found in [12].

3.4.6 A Measure of Uncertainty

The goal of the following analysis is to present a model for the uncertainty that W_k and V_k introduce into the K-matrix system of Eqs. (3.27) and (3.28), respectively. For scalar and vector processes, the covariance is a measure of the uncertainty associated with the process error. In the case of a matrix process, like the one presented in Eqs. (3.27) and (3.28), we introduce a special measure of uncertainty as follows.

Definition 3.4.1. *For the zero-mean matrix process X, a measure of uncertainty P_{XX} is defined as*

$$P_{XX} \overset{\triangle}{=} E\left[XX^T\right] \tag{3.31}$$

As an example, consider the real 2 × 2 symmetric matrix of zero-mean processes x_{11}, x_{12}, and x_{22}, given by

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \tag{3.32}$$

The corresponding P_{XX} has the following form

$$P_{XX} = \begin{bmatrix} E\left[x_{11}^2\right] + E\left[x_{12}^2\right] & E\left[x_{11}\, x_{12}\right] + E\left[x_{12}\, x_{22}\right] \\ E\left[x_{11}\, x_{12}\right] + E\left[x_{12}\, x_{22}\right] & E\left[x_{12}^2\right] + E\left[x_{22}^2\right] \end{bmatrix} \tag{3.33}$$

Discussion

P_{XX} is a symmetric positive semi-definite matrix. The variances of each element of X are on the main diagonal of P_{XX}, while the off-diagonal elements of P_{XX} contain only cross-covariance terms.

Consider the 4×1 vector $\mathrm{vec}X$, which is defined as

$$\mathrm{vec}X \triangleq \begin{bmatrix} x_{11} & x_{12} & x_{12} & x_{22} \end{bmatrix}^T \tag{3.34}$$

and construct its 4×4 covariance matrix

$$\mathrm{cov}\,(\,\mathrm{vec}X) \triangleq E\left[\mathrm{vec}X\,(\,\mathrm{vec}X)^T\right] \tag{3.35}$$

then the trace of the matrix P_{XX} is identical to the trace of the covariance matrix in Eq. (3.35); that is

$$tr(\,P_{XX}) \equiv tr[\,\mathrm{cov}\,(\,\mathrm{vec}X)] \tag{3.36}$$

We realize that although the matrix P_{XX} is not a covariance matrix, it has some desired features in a compact convenient form. Thus, it will be used as a measure of uncertainty for the matrix error processes that are considered in this work.

Using the expressions for W_k and V_k, in Eqs. (3.20) and (3.12), respectively, it can be shown that these processes are zero-mean mutually uncorrelated white-noise processes. Thus,

$$\begin{aligned} E\left[W_k\,W_{k+i}{}^T\right] &= O_4 \\ E\left[V_k\,V_{k+i}{}^T\right] &= O_4 \\ E\left[W_k\,V_{k+i}{}^T\right] &= O_4 \end{aligned} \tag{3.37}$$

for $i \neq 0$. Denote by $\mathcal{Q}_k$ and $\mathcal{R}_k$, respectively, the P-matrices of W_k and V_k; that is

$$\mathcal{Q}_k \triangleq E\left[W_k\,W_k{}^T\right] \qquad \mathcal{R}_k \triangleq E\left[V_k\,V_k{}^T\right] \tag{3.38}$$

then explicit expressions for $\mathcal{Q}_k$ and $\mathcal{R}_k$ can be derived using the assumptions that we made on $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\delta b}_k$. In the computation of $\mathcal{Q}_k$, one must address the issue of the dependence of the process noise matrix, W_k, on the noise-free matrix $B^o_{k/k}$ (Eqs. 3.19, 3.20), which is unknown. To overcome this difficulty $B^o_{k/k}$ is replaced by its best available estimate, $B_{k/k}$. The latter is computed from the estimated K-matrix, $K_{k/k}$, using the definition of the K-matrix given in Eqs. (3.3) and (3.4). The computations of the matrices $\mathcal{Q}_k$ and $\mathcal{R}_k$ are derived in Appendix B.2.

3.5 Optimal-REQUEST

The algorithm derivation in this section follows an approach that was used in a direct derivation of the Kalman-Bucy filter for the case of vector processes [36]. A similar derivation of the discrete Kalman filter can be found in the literature [23, pp. 105-110]. The approach consists of three steps:

1. The update of the estimate is formulated as a linear combination of the predicted estimate and the new observation.
2. The a posteriori and a priori estimates are forced to be unbiased.
3. The optimal filter gain is computed by minimizing the variance of the a posteriori estimation error.

In the third step, instead of the variance, we will use the measure of uncertainty introduced in the preceding section.

3.5.1 Measurement Update Stage

The proposed update stage is a slightly modified version of REQUEST; namely, unlike the REQUEST updating formula of Eq. (3.9), here the updated estimate $K_{k+1/k+1}$ is chosen to be a convex combination of the predicted estimate $K_{k+1/k}$ and the new observation δK_{k+1}; that is

$$K_{k+1/k+1} = \left(1 - \rho_{k+1}\right) \frac{m_k}{m_{k+1}} K_{k+1/k} + \rho_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} \delta K_{k+1} \tag{3.39}$$

where δm_{k+1} is a positive scalar weight and m_{k+1} is computed recursively by

$$m_{k+1} = (1 - \rho_{k+1}) m_k + \rho_{k+1} \delta m_{k+1} \tag{3.40}$$

for $k = 0, 1 \ldots$ and $m_0 = \delta m_0$. The scalar $\rho_{k+1} \in [0, 1)$ has the role of a gain in Eq. (3.39). (We exclude the case where $\rho_{k+1} = 1$ for practical reasons since, in that case, the filter would be memoryless.) Due to the linearity of Eq. (3.39) and to the fact that ρ_{k+1} is a scalar, the properties of symmetry and of zero-trace of the estimated K-matrix are preserved. This is a matter of importance since the computation of the attitude quaternion, which uses the **q**-method, depends on these properties.

The estimation errors of the filter are defined by

$$\Delta K_{k+1/k} \overset{\triangle}{=} K^o_{k+1/k} - K_{k+1/k} \tag{3.41a}$$

$$\Delta K_{k+1/k+1} \overset{\triangle}{=} K^o_{k+1/k+1} - K_{k+1/k+1} \tag{3.41b}$$

where $\Delta K_{k+1/k}$ and $\Delta K_{k+1/k+1}$ denote, respectively, the a priori and the a posteriori estimation errors. It is assumed that the a priori estimate $K_{k+1/k}$ is unbiased, that is $E\left[\Delta K_{k+1/k}\right] = 0$. This assumption will be justified when developing the prediction stage.

The expression for $K^o_{k+1/k+1}$ is known from Eq. (3.26), where α_{k+1} is arbitrary in $[0, 1)$. For a reason that will be clear in the ensuing we choose $\alpha_{k+1} = \rho_{k+1}$, where ρ_{k+1} is that which was introduced in Eq. (3.39). Consequently, Eq. (3.26) becomes

$$K^o_{k+1/k+1} = \left(1 - \rho_{k+1}\right) \frac{m_k}{m_{k+1}} K^o_{k+1/k} + \rho_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} \delta K^o_{k+1} \tag{3.42}$$

Subtracting Eq. (3.39) from Eq. (3.42) yields the following relation between the a priori and the a posteriori errors

$$\Delta K_{k+1/k+1} = \left(1 - \rho_{k+1}\right) \frac{m_k}{m_{k+1}} \Delta K_{k+1/k} + \rho_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} V_{k+1} \tag{3.43}$$

where V_{k+1} is the measurement error defined in Eq. (3.12). Taking the expectation of both sides of Eq. (3.43) yields

$$E\left[\Delta K_{k+1/k+1}\right] = \left(1 - \rho_{k+1}\right) \frac{m_k}{m_{k+1}} E\left[\Delta K_{k+1/k}\right] + \rho_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} E\left[V_{k+1}\right] \tag{3.44}$$

Using the assumptions that the measurement error V_{k+1} and the a priori estimation error $\Delta K_{k+1/k}$ are zero-mean, one finds that the a posteriori estimation error $\Delta K_{k+1/k+1}$ is zero-mean too, as desired.

The P-matrices corresponding to both estimation errors are defined as follows

$$P_{k+1/k} \overset{\triangle}{=} E\left[\Delta K_{k+1/k} \Delta K^T_{k+1/k}\right] \tag{3.45}$$

$$P_{k+1/k+1} \overset{\triangle}{=} E\left[\Delta K_{k+1/k+1} \Delta K^T_{k+1/k+1}\right] \tag{3.46}$$

Using Eq. (3.43), we compute the following product

$$\begin{aligned} \Delta K_{k+1/k+1} \Delta K^T_{k+1/k+1} \quad = \quad & \left[\left(1 - \rho_{k+1}\right) \frac{m_k}{m_{k+1}}\right]^2 \Delta K_{k+1/k} \Delta K^T_{k+1/k} \\ & - \left(1 - \rho_{k+1}\right) \rho_{k+1} \frac{m_k \delta m_{k+1}}{m^2_{k+1}} \left(\Delta K_{k+1/k} V^T_{k+1} + V_{k+1} \Delta K^T_{k+1/k}\right) \\ & + \left(\rho_{k+1} \frac{\delta m_{k+1}}{m_{k+1}}\right)^2 V_{k+1} V^T_{k+1} \end{aligned} \tag{3.47}$$

Before computing the expectations of the expressions in Eq. (3.47), we consider the following expectation, $E\left[\Delta K_{k+1/k} V_{k+1}^T\right]$. From Eq. (3.30) $\boldsymbol{\delta b}$ is a zero-mean white noise process, therefore (see Eqs. 3.10-3.12) V is a zero-mean white-noise process too. Moreover, the random variable $\Delta K_{k+1/k}$ depends on the sequences $\{W_i\}$ and $\{V_i\}$ where i takes the integer values from 1 to k only. Therefore δK_{k+1} and V_{k+1} are uncorrelated, thus

$$E\left[\Delta K_{k+1/k} V_{k+1}^T\right] = O_4 \tag{3.48}$$

Taking the transpose of Eq. (3.48) yields a similar result. Taking the expectation of both sides of Eq. (3.47), and only retaining the non-zero terms yields

$$P_{k+1/k+1} = \left((1-\rho_{k+1})\frac{m_k}{m_{k+1}}\right)^2 E\left[\Delta K_{k+1/k}\Delta K_{k+1/k}^T\right] + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 E\left[V_{k+1}V_{k+1}^T\right] \tag{3.49}$$

One identifies the matrices $P_{k+1/k}$ and $\mathcal{R}_{k+1}$ in the first and second terms of the right hand-side of Eq. (3.49), thus we can write

$$P_{k+1/k+1} = \left((1-\rho_{k+1})\frac{m_k}{m_{k+1}}\right)^2 P_{k+1/k} + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 \mathcal{R}_{k+1} \tag{3.50}$$

The last equation expresses the uncertainty update in the K-matrix estimation process, when a new measurement is acquired.

3.5.2 Optimal Gain

When a new observation is processed, we would like the estimation uncertainty to decrease as much as possible. From the earlier discussion of the properties of the P-matrix we saw that its trace was a suitable measure of the uncertainty. Thus, we define the following cost function

$$J_{k+1} \triangleq tr\left(E\left[\Delta K_{k+1/k+1}\Delta K_{k+1/k+1}^T\right]\right) = tr\left(P_{k+1/k+1}\right) \tag{3.51}$$

Then, the design problem of the filter gain, ρ_{k+1}, reduces to solving the following minimization problem with respect to ρ_{k+1},

$$\min_{\rho_{k+1}}\left\{J_{k+1} = tr(P_{k+1/k+1})\right\} \tag{3.52}$$

Inserting Eq. (3.50) into the expression for J_{k+1} in Eq. (3.51) yields, after some manipulation, the expression

$$J_{k+1}\left(\rho_{k+1}\right) = \left(\left(1-\rho_{k+1}\right)\frac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 tr\left(\mathcal{R}_{k+1}\right) \qquad (3.53)$$

The first-order necessary condition for extremum of J_{k+1} is

$$\begin{aligned} \frac{d\,J_{k+1}}{d\,\rho_{k+1}} = 2\left[\left(\frac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) + \left(\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 tr\left(\mathcal{R}_{k+1}\right)\right]\rho_{k+1} \\ -2\left(\frac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) = 0 \end{aligned} \qquad (3.54)$$

yielding the following condition for ρ^*_{k+1} to be a stationary point

$$\rho^*_{k+1} = \frac{m_k^2\, tr\left(P_{k+1/k}\right)}{m_k^2\, tr\left(P_{k+1/k}\right) + \delta m_{k+1}^2\, tr\left(\mathcal{R}_{k+1}\right)} \qquad (3.55)$$

Note that ρ^*_{k+1}, as computed from Eq. (3.55), is in the interval $[0,1)$ for any $m_k \neq 0$ and $\delta m_k \neq 0$. Using the sufficiency condition for this point to be a minimum, it can be verified that the cost function J_{k+1} indeed reaches a minimum at ρ^*_{k+1}. *Note that, in the case of a scalar process, Eq. (3.55) yields the exact expression of the Kalman filter gain.* Even in the general case the filter gain, ρ^*_{k+1}, still has the feature of a Kalman filter gain; namely, for a high uncertainty in the a priori estimate, with respect to the uncertainty in the measurement, the gain is close to 1 and weighs more favorably the new measurement in the update stage described in Eq. (3.39). On the other hand, for a high uncertainty in the measurement, with respect to the uncertainty in the a priori estimate, the gain is close to 0, and the filter assigns a low weight to the new measurement.

3.5.3 Prediction Stage

We require that the predicted $K_{k+1/k}$ be linear in $K_{k/k}$ and that it produces an unbiased predicted estimate. These requirements yield the prediction formula of the REQUEST algorithm (Eq. 3.6)

$$K_{k+1/k} = \Phi_k\, K_{k/k}\, \Phi_k^T \qquad (3.56)$$

Using the K-matrix process Eq. (3.21), the prediction Eq. (3.56) and the definitions of the estimation errors Eqs. (3.41) yields the error propagation equation

$$\Delta K_{k+1/k} = \Phi_k \Delta K_{k/k}\, \Phi_k^T + W_k \qquad (3.57)$$

Using the latter expression for $\Delta K_{k+1/k}$, the stochastic models of the noise, and the orthogonality property of the Φ_k matrix, the propagation equation for the P-matrix is obtained as follows

$$P_{k+1/k} \;=\; \Phi_k \, P_{k/k} \, \Phi_k^T + \mathcal{Q}_k \tag{3.58}$$

Equation (3.58) is similar to the covariance propagation equation in the Kalman filter algorithm.

To conclude, similarly to the Kalman filter, the algorithm contains two parallel channels. One channel is for the computation of the signal estimate, which here is the K-matrix estimate, and the other one is for the uncertainty propagation of the estimation process, which in the Kalman filter is expressed by the covariance matrix, and is needed for the computation of the optimal filter-gain.

3.5.4 Algorithm Summary

The Optimal-REQUEST algorithm presented in this section can be summarized as follows.

1. Initialization:

$$K_{0/0} = \delta K_0 \tag{3.59}$$

where δK_0 is computed using the first vector measurement according to Eq. (3.8)

$$P_{0/0} = \mathcal{R}_0 \tag{3.60}$$

$$m_0 = \delta m_0 \tag{3.61}$$

where δm_0 is a positive weighting factor.

2. Time Update equations:

$$K_{k+1/k} = \Phi_k \, K_{k/k} \, \Phi_k^T \tag{3.62}$$

$$P_{k+1/k} = \Phi_k \, P_{k/k} \, \Phi_k^T + \mathcal{Q}_k \tag{3.63}$$

where the matrix $\mathcal{Q}_k$ is computed according to Eqs. (3.19), (3.20) and (3.38).

3. Measurement Update equations:

$$\rho^*_{k+1} = \frac{m_k^2 \, tr\left(P_{k+1/k}\right)}{m_k^2 \, tr\left(P_{k+1/k}\right) + \delta m_{k+1}^2 \, tr\left(\mathcal{R}_{k+1}\right)} \tag{3.64}$$

$$m_{k+1} = \left(1 - \rho^*_{k+1}\right) m_k + \rho^*_{k+1} \delta m_{k+1} \tag{3.65}$$

$$K_{k+1/k+1} = \left(1 - \rho^*_{k+1}\right) \frac{m_k}{m_{k+1}} K_{k+1/k} + \rho^*_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} \delta K_{k+1} \tag{3.66}$$

$$P_{k+1/k+1} = \left[\left(1 - \rho^*_{k+1}\right) \frac{m_k}{m_{k+1}}\right]^2 P_{k+1/k} + \left(\rho^*_{k+1} \frac{\delta m_{k+1}}{m_{k+1}}\right)^2 \mathcal{R}_{k+1} \tag{3.67}$$

where the matrix $\mathcal{R}_{k+1}$ is computed according to Eqs. (3.11), (3.12) and (3.38). The output of the algorithm, $\widehat{\mathbf{q}}_{k+1/k+1}$, is the eigenvector of $K_{k+1/k+1}$ which belongs to its maximal eigenvalue.

3.6 Adaptive Optimal-REQUEST

3.6.1 Motivation

If the filter is constructed on the basis of an erroneous model, "it learns the wrong state too well when it operates over many data" [46, p. 302]. Suppose that the actual measurement noise variance is higher than the one that is assumed in the filter, or that the actual process noise contains an unmodeled bias, then the filter matrices, $\mathcal{Q}_k$ and $\mathcal{R}_k$, are not accurate any more. Eventually, the matrix $P_{k+1/k}$ becomes very small, the filter gain, ρ^*_{k+1}, is therefore small, and subsequent observations have little effects on the estimate. The latter fact, together with the difference between the filter and the actual dynamical models, might yield divergence of the estimates sequence. In order to compensate for model errors *in the context of real-time filtering*, the adaptive filtering technique seems to be adequate.

In the following we use an approach developed by Jazwinski for the case of a Gaussian scalar measurement process [46, p. 311]. Because of the particular nature of the present system, we adopt Jazwinski's approach to design ad-hoc criterions, as will be shown. Essentially, the technique is to look at measurement residuals and "to let the noise levels *adapt* to the residuals".

A first difficulty appears when considering the definition of the residual. Actually, we can not use the K-matrix in a "classic" definition of a measurement residual, that is, by subtracting the predicted K-matrix, $K_{k+1/k}$, from the actual measured K-matrix, δK_{k+1}, as follows

$$\mathit{Residual} \overset{\triangle}{=} \delta K_{k+1} - K_{k+1/k} \tag{3.68}$$

The reason is that δK_{k+1} is constructed from observations made at t_{k+1} , while $K_{k+1/k}$ is a predicted estimate of the observations acquired up to time t_k. Therefore, the two matrices, δK_{k+1} and $K_{k+1/k}$ may be different. This would yield a non-zero residual even in the ideal case of noise-free data !

A second difficulty is inherent to the adaptive filtering technique : how can we know whether the filter divergence is due to a process noise modelling error or to a measurement noise modelling error? This question leads to the practical question: when observing the filter divergence (high values of the residuals), do we have to increase or to decrease the filter gain ? We will show how the first difficulty can be overcome by designing specific residuals based on the geometrical properties of the K-matrix. Furthermore, the necessity of defining special measurement residuals appears to be a "blessing disguised", since it provides also a solution to the second difficulty.

3.6.2 Measurement Noise Adaptive Filtering

Residual definition

We define the measurement residual as the following scalar

$$r^{\lambda}_{k+1} \overset{\triangle}{=} 1 - \widehat{\mathbf{q}}^{T}_{k+1/k}\, \delta K_{k+1}\, \widehat{\mathbf{q}}_{k+1/k} \tag{3.69}$$

where δK_{k+1} is the new observation of the K-matrix, and $\widehat{\mathbf{q}}_{k+1/k}$ is the predicted estimate of the quaternion at time t_{k+1}, given measurements up to time t_k. We define r^{λ}_{k+1} as a measurement residual for the following reason: we know, from the $\mathbf{q}$-method, that the true quaternion is the eigenvector of the error-free measurement matrix δK^{o}_{k+1}. Therefore, if there is no estimation error in $\widehat{\mathbf{q}}_{k+1/k}$, and if there is no measurement error in δK_{k+1}, then the quadratic expression $\widehat{\mathbf{q}}^{T}_{k+1/k}\, \delta K_{k+1}\, \widehat{\mathbf{q}}_{k+1/k}$ is equal to the eigenvalue of δK_{k+1} related to $\widehat{\mathbf{q}}_{k+1/k}$. Moreover, we have normalized the weights in δK_{k+1}, so that the largest eigenvalue of the error-free matrix δK_{k+1} is equal to 1. As a consequence, the expression r^{λ}_{k+1} is equal to zero. Thus, r^{λ}_{k+1} is a legitime residual.

Adaptive estimation of the measurement noise level μ_{k+1}

Assuming that $\mathcal{R}_{k+1} = \mu_{k+1} I_4$, we compute an estimate of μ_{k+1} by requiring

$$E\left[\left(r^{\lambda}_{k+1}\right)^2\right] = \left(r^{\lambda}_{k+1}\right)^2 \tag{3.70}$$

The criterion, proposed in Eq. (3.70) follows the following approach. Let us denote the estimate of μ_{k+1} by μ^*_{k+1}. Since the predicted variance of the residual, $E\left[\left(r^{\lambda}_{k+1}\right)^2\right]$, is function of the filter measurement noise level, μ^*_{k+1}, and since the actual residual squared, $\left(r^{\lambda}_{k+1}\right)^2$, is function of the actual measurement noise level, μ_{k+1}, then Eq. (3.70) will be satisfied if μ^*_{k+1} and μ_{k+1} are equal. Using Jazwinski's words, "the estimate of μ_{k+1} produces *consistency* between the residuals and their predicted statistics" [46, p. 312]. Turning to the computation of μ^*_{k+1}, we analyze the statistics of r^{λ}_{k+1}. It is shown in Appendix B.3 that the residual, r^{λ}_{k+1}, has the following first approximation in the measurement error, V_{k+1},

$$r^{\lambda}_{k+1} = -\mathbf{q}^T_{k+1} V_{k+1} \mathbf{q}_{k+1} \tag{3.71}$$

Equation (3.71) shows that the residual r^{λ}_{k+1} is selective. It is proportional to the measurement noise in a first order, while the process noise is entering only higher order terms. This fact motivates the choice of the residual r^{λ}_{k+1} for applying a *measurement noise* adaptive filtering technique. Moreover, it is shown in Appendix B.3 that the residual, as expressed in Eq. (3.71), is unbiased.

The predicted variance of r^{λ}_{k+1} is computed as follows. We assume that the matrix $\mathcal{R}_k$ is of the form

$$\mathcal{R}_k = \mu_k I_4 \tag{3.72}$$

and that there is a single observation at t_{k+1}, which is related to the reference unit vector $\mathbf{r}_{k+1}$. Defining the following real 4×3 matrix $\widehat{\Theta}_{k+1}$ as

$$\widehat{\Theta}_{k+1} \triangleq \begin{bmatrix} \mathbf{r}^T_{k+1} \widehat{\mathbf{e}}_{k+1/k} I_3 + \mathbf{r}_{k+1} \widehat{\mathbf{e}}^T_{k+1/k} - \widehat{\mathbf{e}}_{k+1/k} \mathbf{r}^T_{k+1} - \widehat{q}_{k+1/k} \left[\mathbf{r}_{k+1} \times\right] \\ \widehat{\mathbf{e}}^T_{k+1/k} \left[\mathbf{r}_{k+1} \times\right] + \widehat{q}_{k+1/k} \mathbf{r}^T_{k+1} \end{bmatrix} \tag{3.73}$$

where $\widehat{\mathbf{q}}^T_{k+1/k} = \left[\widehat{\mathbf{e}}^T_{k+1/k} \; \widehat{q}_{k+1/k}\right]$, then an approximation to the predicted residual variance is given by

$$E\left[\left(r^\lambda_{k+1}\right)^2\right] = \mu_{k+1} \; \widehat{\mathbf{q}}^T_{k+1/k} \widehat{\Theta}_{k+1} \left(I_3 - \mathbf{b}_{k+1} \, \mathbf{b}^T_{k+1}\right) \widehat{\Theta}^T_{k+1} \, \widehat{\mathbf{q}}_{k+1/k} \tag{3.74}$$

Using Eqs. (3.71) and (3.74) in Eq. (3.70), and solving for μ_{k+1}, yields the estimate of the measurement noise level, μ^*_{k+1},

$$\mu^*_{k+1} = \frac{\left(r^\lambda_{k+1}\right)^2}{\widehat{\mathbf{q}}^T_{k+1/k} \widehat{\Theta}_{k+1} \left(I_3 - \mathbf{b}_{k+1} \, \mathbf{b}^T_{k+1}\right) \widehat{\Theta}^T_{k+1} \, \widehat{\mathbf{q}}_{k+1/k}} \tag{3.75}$$

subject to the constraint that μ^*_{k+1} is positive. The detailed proof of Eq. (3.75) is presented in Appendix B.3. To conclude, Eq. (3.75) provides a real-time estimate of the measurement noise level at time t_{k+1}, given the residual r^λ_{k+1}, the pair of unit vectors, $\mathbf{b}_{k+1}$, $\mathbf{r}_{k+1}$, and the predicted estimate of the quaternion $\widehat{\mathbf{q}}^T_{k+1/k}$. μ^*_{k+1} is substituted to μ_k in the computation of the matrix $\mathcal{R}_k$, Eq. (3.72). Note that, in order to compute μ^*_{k+1}, we need to compute $\widehat{\mathbf{q}}_{k+1/k}$.

3.6.3 Process Noise Adaptive Filtering

Residual definition

We consider now the case of process noise modelling errors. We want to design a residual that is sensitive to these errors. We define the following residual vector

$$\mathbf{r}^q_{k+1} \overset{\triangle}{=} \left(I_4 - \delta K_{k+1}\right) \widehat{\mathbf{q}}_{k+1/k} \tag{3.76}$$

The motivation for the above residual stems from the fact that the true quaternion is an eigenvector with eigenvalue 1 of the error-free new K-matrix. Thus, if there is no estimation error in $\widehat{\mathbf{q}}_{k+1/k}$, and if the matrix δK_{k+1} is error-free, then $\widehat{\mathbf{q}}_{k+1/k}$ belongs to the null space of $I_4 - \delta K_{k+1}$ and the residual, $\mathbf{r}^q_{k+1}$, vanishes. Given the residuals $\mathbf{r}^q_{k+1}$, we want the filter to adapt, in some optimal way, its process noise level to the actual process noise level.

Adaptive estimation of the process noise level η_k

For this purpose, we extend the criterion proposed in the case of a scalar residual to the case of a vector residual. Consider the following minimization problem over η_k, which is the covariance of the gyro outputs noise,

$$\min_{\eta_k \geq 0} \left\{ J\left(\eta_k\right) = \left\| \mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T - E\left[\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T\right] \right\|_F \right\} \tag{3.77}$$

where $\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T$ is the squared residual, $E\left[\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T\right]$ is the predicted covariance matrix of this residual, and $\|M\|_F$ denotes the Frobenius norm of the matrix M; that is $\|M\|_F \overset{\triangle}{=} \sqrt{tr(MM^T)}$. The best estimate of η_k, denoted by η^*_k, is that particular value of η_k for which the predicted residual covariance matrix is the closest to the actual squared residual. It is shown in Appendix B.5 that the residual $\mathbf{r}^q_{k+1}$ is unbiased. Let P_{rr} denote the covariance matrix of $\mathbf{r}^q_{k+1}$; that is $P_{rr} = E\left[\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T\right]$. The computation of P_{rr} is more cumbersome than in the scalar case and needs some intermediary results. It requires the computation of the quaternion estimation error covariance matrices. Let $P^{\mathbf{q}}_{k/k}$ and $P^{\mathbf{q}}_{k+1/k}$ denote the covariance matrices of the quaternion additive estimation errors $\boldsymbol{\Delta q}_{k/k}$ and $\boldsymbol{\Delta q}_{k+1/k}$, respectively, and let Q^ϵ_k denote the covariance matrix of $\boldsymbol{\epsilon}_k$ (see Eq. 3.29). Assuming that $P^{\mathbf{q}}_{k/k}$ and Q^ϵ_k are known, it is shown in Appendix B.4 that $P^{\mathbf{q}}_{k+1/k}$ is computed as follows.

$$P^{\mathbf{q}}_{k+1/k} = \Phi_k P^{\mathbf{q}}_{k/k} \Phi^T_k + Q^q_k \tag{3.78a}$$

$$Q^q_k = \widehat{\Xi}_k Q^\epsilon_k \widehat{\Xi}^T_k (\frac{\Delta t}{2})^2 \tag{3.78b}$$

$$\widehat{\Xi}_k = \begin{bmatrix} [\widehat{\mathbf{e}}_k \times] + \widehat{q}_k I_3 \\ -\widehat{\mathbf{e}}^T_k \end{bmatrix} \tag{3.78c}$$

Based on Eqs. (3.78) it is shown in Appendix B.5 that P_{rr} is computed as follows

$$\begin{aligned} P_{rr} \;=\; & \left(I_4 - \delta K_{k+1}\right) P^{\mathbf{q}}_{k+1/k} \left(I_4 - \delta K_{k+1}\right)^T \\ & +\mu_{k+1} \widehat{\Theta}_{k+1} \left(I_3 - \mathbf{b}_{k+1} \mathbf{b}^T_{k+1}\right) \widehat{\Theta}^T_{k+1} \end{aligned} \tag{3.79}$$

where $\widehat{\Theta}_{k+1}$ was defined in Eq. (3.73). Substituting Eq. (3.78a) into Eq. (3.79) and assuming $Q^q_k = \eta_k (\frac{\Delta t}{2})^2 I_4$ yields

$$\begin{aligned} P_{rr} \;=\; & \left(I_4 - \delta K_{k+1}\right) \Phi_k P^{\mathbf{q}}_{k/k} \Phi^T_k \left(I_4 - \delta K_{k+1}\right)^T \\ & +\mu_{k+1} \widehat{\Theta}_{k+1} \left(I_3 - \mathbf{b}_{k+1} \mathbf{b}^T_{k+1}\right) \widehat{\Theta}^T_{k+1} \\ & +\eta_k \left(I_4 - \delta K_{k+1}\right)^2 (\frac{\Delta t}{2})^2 \end{aligned} \tag{3.80}$$

Defining the matrices M and N as follows

$$\begin{aligned} N &= \left(I_4 - \delta K_{k+1}\right)^2 \left(\tfrac{\Delta t}{2}\right)^2 \\ M &= \left(I_4 - \delta K_{k+1}\right) \Phi_k \, P^{\mathbf{q}}_{k/k} \, \Phi_k^T \left(I_4 - \delta K_{k+1}\right)^T \\ &\quad + \mu_{k+1} \widehat{\Theta}_{k+1} \left(I_3 - \mathbf{b}_{k+1} \mathbf{b}_{k+1}^T\right) \widehat{\Theta}_{k+1}^T \end{aligned} \tag{3.81}$$

then Eq. (3.80) is rewritten as

$$P_{rr} = M + \eta_k N \tag{3.82}$$

Recalling that the minimization problem of Eq. (3.77) is

$$\min_{\eta_k \geq 0} \left\{ J(\eta_k) = \| \mathbf{r}^q_{k+1} (\mathbf{r}^q_{k+1})^T - P_{rr}(\eta_k) \|^2 \right\} \tag{3.83}$$

we define the scalars α and β as

$$\alpha \stackrel{\triangle}{=} tr\left[\left(\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T - M\right)\right] N^T \tag{3.84a}$$

$$\beta \stackrel{\triangle}{=} tr\left(N^2\right) \tag{3.84b}$$

so that the cost function in Eq. (3.83) is rewritten

$$J(\eta_k) = \beta \, \eta_k^2 - 2 \, \alpha \, \eta_k + \text{const} \tag{3.85}$$

The first derivative of the cost function with respect to η_k is

$$\frac{\partial J}{\partial \eta_k} = 2 \, (\beta \, \eta_k - \alpha) \tag{3.86}$$

Equating Eq. (3.86) to zero, one gets the optimal value of η_k, η_k^* :

$$\eta_k^* = \frac{\alpha}{\beta} \tag{3.87}$$

Substituting the values of the coefficients α and β, Eq. (3.84) into Eq. (3.87) yields

$$\eta_k^* = \frac{tr\left[\left(\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T - M\right) N^T\right]}{tr(N^2)} \tag{3.88}$$

From Eq. (3.86), we see that the second derivative of the cost function with respect to η_k is equal to $2\,\beta$. As seen from Eq. (3.84b), β is a positive scalar. So, the cost function reaches a minimum at η_k^*. To summarize, it has been shown that, given the new observation δK_{k+1}, the predicted quaternion estimate $\widehat{\mathbf{q}}_{k+1/k}$, and the covariance matrix of the filtered quaternion estimation error $P^{\mathbf{q}}_{k/k}$, a best estimate of the process noise η_k^* could be computed in Eq. (3.88). This value is then used to update the matrix Q_k^{ϵ} in Eq. (3.29), and the matrix $\mathcal{Q}_k$ (Appendix B.2).

Computation of $P^{\mathbf{q}}_{k/k}$

Since the estimation update stage is performed on the K-matrix process, rather than on the quaternion process, we can not deduce an exact recursive relation between the a priori and the a posteriori estimates of the quaternion. Thus, we can not perform an exact covariance computation of the update stage in order to compute $P^{\mathbf{q}}_{k/k}$. However, we propose the approach of approximating the matrix $P^{\mathbf{q}}_{k/k}$. We begin by writing the spectral decomposition of the real symmetric matrix $K_{k+1/k}$ [15, p. 64]. Given the four eigenvalues $\lambda^0_{k+1/k} \geq \lambda^1_{k+1/k} \geq \lambda^2_{k+1/k} \geq \lambda^3_{k+1/k}$ and the associated four eigenvectors $\widehat{\mathbf{q}}^0_{k+1/k}$, $\widehat{\mathbf{q}}^1_{k+1/k}$, $\widehat{\mathbf{q}}^2_{k+1/k}$, $\widehat{\mathbf{q}}^3_{k+1/k}$, we have

$$\begin{aligned} K_{k+1/k} = {} & \lambda^0_{k+1/k}\,\widehat{\mathbf{q}}^0_{k+1/k}(\widehat{\mathbf{q}}^0_{k+1/k})^T + \lambda^1_{k+1/k}\,\widehat{\mathbf{q}}^1_{k+1/k}(\widehat{\mathbf{q}}^1_{k+1/k})^T \\ & + \lambda^2_{k+1/k}\,\widehat{\mathbf{q}}^2_{k+1/k}(\widehat{\mathbf{q}}^2_{k+1/k})^T + \lambda^3_{k+1/k}\,\widehat{\mathbf{q}}^3_{k+1/k}(\widehat{\mathbf{q}}^3_{k+1/k})^T \end{aligned} \tag{3.89}$$

Assuming that the error in the eigenvalues are negligible with respect to the error in the eigenvectors, defining

$$\boldsymbol{\Delta q}^i_{k+1/k} \overset{\triangle}{=} \mathbf{q}^i_{k+1} - \widehat{\mathbf{q}}^i_{k+1/k} \tag{3.90}$$

for $i = 0, 1, 2, 3$, and using Eq. (3.90) in Eq. (3.89), we see that the term associated with $\lambda^0_{k+1/k}$ becomes

$$\lambda^0_{k+1/k}\left[\mathbf{q}^0_{k+1}{\mathbf{q}^0_{k+1}}^T - \mathbf{q}^0_{k+1}(\boldsymbol{\Delta q}^0_{k+1/k})^T - \boldsymbol{\Delta q}^0_{k+1/k}{\mathbf{q}^0_{k+1}}^T + \boldsymbol{\Delta q}^0_{k+1/k}(\boldsymbol{\Delta q}^0_{k+1/k})^T\right] \tag{3.91}$$

where $\mathbf{q}^0_{k+1}$ is the true attitude quaternion at time t_{k+1}. A similar development can be made for the three other terms of Eq. (3.89). We recall here that

$$\Delta K_{k+1/k} = K^o_{k+1/k} - K_{k+1/k} \tag{3.92}$$

Since $K^o_{k+1/k}$ is constructed from the true data its spectral decomposition includes the true eigenvalues, λ^i_{k+1}, and the true eigenvectors, $\mathbf{q}^i_{k+1}$, as follows

$$K^o_{k+1/k} = \lambda^0_{k+1}\mathbf{q}^0_{k+1}{\mathbf{q}^0_{k+1}}^T + \lambda^1_{k+1}\mathbf{q}^1_{k+1}{\mathbf{q}^1_{k+1}}^T + \lambda^2_{k+1}\mathbf{q}^2_{k+1}{\mathbf{q}^2_{k+1}}^T + \lambda^3_{k+1}\mathbf{q}^3_{k+1}{\mathbf{q}^3_{k+1}}^T \tag{3.93}$$

Using Eqs. (3.93) and (3.89) in Eq. (3.92), and assuming that $\lambda^i_{k+1/k} = \lambda^i_{k+1}$, $i = 0, 1, 2, 3$, we can express $\Delta K_{k+1/k}$ as the sum of four terms, where the first one is

$$\lambda^0_{k+1}\left[-\mathbf{q}^0_{k+1}(\boldsymbol{\Delta q}^0_{k+1/k})^T - \boldsymbol{\Delta q}^0_{k+1/k}{\mathbf{q}^0_{k+1}}^T + \boldsymbol{\Delta q}^0_{k+1/k}(\boldsymbol{\Delta q}^0_{k+1/k})^T\right] \tag{3.94}$$

and the other three terms are similar expressions with the superscript taking the values $1, 2, 3$. Each term is a partial predicted estimation error of the K-matrix. We define $\Delta K^0_{k+1/k}$ as

$$\Delta K^0_{k+1/k} \triangleq \lambda^0_{k+1} \left[-\mathbf{q}^0_{k+1} (\boldsymbol{\Delta q}^0_{k+1/k})^T - \boldsymbol{\Delta q}^0_{k+1/k} \mathbf{q}^0_{k+1}{}^T + \boldsymbol{\Delta q}^0_{k+1/k} (\boldsymbol{\Delta q}^0_{k+1/k})^T \right] \tag{3.95}$$

Taking the expectations of both sides of Eq. (3.95) and assuming that $\boldsymbol{\Delta q}^0_{k+1/k}$ is zero-mean yields

$$E\left[\Delta K^0_{k+1/k}\right] = -\lambda^0_{k+1/k} E\left[\boldsymbol{\Delta q}^0_{k+1/k} (\boldsymbol{\Delta q}^0_{k+1/k})^T\right] \triangleq -\lambda^0_{k+1/k} P^{\mathbf{q}}_{k+1/k} \tag{3.96}$$

where $\lambda^0_{k+1/k}$ was substituted to λ^0_{k+1}. From Eq. (3.96), we conclude that the expectation of the partial predicted error $\Delta K^0_{k+1/k}$ is proportional to the covariance matrix of the predicted error in the corresponding eigenvector.

The same analysis (Eq. 3.95 through Eq. 3.101) holds for the new observation matrix δK_{k+1}. In this case, the expectation of the "partial measurement error" is proportional to a quaternion measurement error covariance matrix R^q_{k+1}. The expression for R^q_{k+1} was developed in Chapter 2, and is given here

$$R^q_{k+1} = \mu_{k+1} \widehat{\Xi}_{k+1/k+1} \left(I_3 - \mathbf{b}_{k+1} \mathbf{b}^T_{k+1} \right) \widehat{\Xi}^T_{k+1/k+1} \tag{3.97a}$$

$$\widehat{\Xi}_{k+1/k+1} \triangleq \begin{bmatrix} \left(\widehat{\mathbf{e}}_{k+1/k+1} \times \right) + \widehat{q}_{k+1/k+1} I_3 \\ -\widehat{\mathbf{e}}_{k+1/k+1} \end{bmatrix} \tag{3.97b}$$

Considering the Measurement Update stage of the Optimal-REQUEST, Eq.(3.66), we see that the filtered estimate $K_{k+1/k+1}$ is a linear combination of $K_{k+1/k}$ and δK_{k+1} with scalar coefficients. The error equation is given by

$$\Delta K_{k+1/k+1} = \left(1 - \rho^*_{k+1}\right) \frac{m_k}{m_{k+1}} \Delta K_{k+1/k} - \rho^*_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} V_{k+1} \tag{3.98}$$

where the optimal gain ρ^*_{k+1} is substituted for ρ_{k+1}. Taking the expectation of Eq.(3.98) gives a linear equation in the respective expectations. We know that the "global" expectations, namely $E\left[\Delta K_{k+1/k+1}\right]$, $E\left[\Delta K_{k+1/k}\right]$ and $E\left[V_{k+1}\right]$ are equal to zero. But we are actually interested in the expectations of the "partial errors", in $\Delta K_{k+1/k+1}$, $\Delta K_{k+1/k}$ and V_{k+1}, that are associated with the largest eigenvalue of each matrix. Following the argument developed above, each "partial expectation" is proportional to a covariance matrix of the related estimated eigenvector. If we consider the largest eigenvalue, then the estimated eigenvector is the estimated quaternion of rotation. In the following we will call "largest eigenvector", the eigenvector related to the largest eigenvalue. Let us assume that the predicted quaternion $\widehat{\mathbf{q}}_{k+1/k}$ and the quaternion associated with the new observation are closed to each other. This is justified if the measurement and process noises are small. Thus, we realize, that the update stage Eq.(3.66) leaves the largest eigenvector in the estimated K-matrix relatively unchanged, while the other eigenvectors may vary dramatically. Assuming that this "independence" in the behavior of the largest eigenvector happens also for the corresponding error, and thus for the corresponding error covariance matrix, we propose the following heuristic update stage of $P^{\mathbf{q}}_{k+1/k+1}$,

$$P^{\mathbf{q}}_{k+1/k+1} = \left(1 - \rho^*_{k+1}\right) \frac{m_k}{m_{k+1}} P^{\mathbf{q}}_{k+1/k} - \rho^*_{k+1} \frac{\delta m_{k+1}}{m_{k+1}} R^q_{k+1} \tag{3.99}$$

To conclude this section, the adaptive process noise algorithm is more cumbersome than the nominal algorithm or than the adaptive measurement noise algorithm. In order to compute the estimate of the process noise level η^*_k, using Eqs.(3.81), (3.84) and (3.87), an approximate covariance analysis of the quaternion estimation has to be performed through Eqs. (3.78), (3.97) and (3.99). Note that the same estimated value η^*_k is used for evaluating the process noise covariance matrices, Q^ϵ_k and $\mathcal{Q}_k$.

3.7 Simulation Study

3.7.1 Optimal-REQUEST

An extensive Monte-Carlo simulation study was performed in order to test the attitude estimator in the presence of process and measurement noises. Different single vector observations were acquired at each sampling time. The body coordinate system $\mathcal{B}$ was assumed to be fixed with respect to an inertial coordinate system $\mathcal{R}$. The vector observation noise was a zero-mean Gaussian white noise with an angular standard deviation of one degree, which is a typical accuracy obtained using magnetometers. Three body-mounted gyroscopes measured the angular velocity of $\mathcal{B}$ with respect to $\mathcal{R}$. Because the system $\mathcal{B}$ did not rotate with respect to system $\mathcal{R}$ the nominal body rates were zero, hence the gyro measurements contained only gyro noises. The gyro noises were Gaussian zero-mean white noises with a standard deviation of 0.2 deg/hr in each axis. The noise models in the system and in the filter were identical. The sample rate was 10 Hz, both in the measured directions and in the gyro measurements. Each simulation time lasted 100 seconds. We ran 100 runs with different seeds and averaged the results at each time point to obtain the ensemble averages.

The results are summarized in Figs. 3.1 and 3.2. Figure 3.1 presents the time history of the Monte-Carlo mean of $\delta\phi$, the angular estimation error. The angle $\delta\phi$ is defined as the angle of the small rotation that brings the estimated body frame, $\widehat{\mathcal{B}}$, onto the true body frame, $\mathcal{B}$. This angle is obtained as follows. First, the quaternion of the rotation from $\widehat{\mathcal{B}}$ to $\mathcal{B}$, denoted by $\boldsymbol{\delta q}$, is evaluated, then the rotation angle $\delta\phi$ is computed from the scalar component of $\boldsymbol{\delta q}$, δq, using the known relation [1, p. 414] $\delta\phi = 2 \arccos(\delta q)$. The error reaches a steady state of approximately 0.06 degree. Figure 3.2 shows the time histories of two parameters of the filter, ρ and η. The scalar η is defined as the Frobenius norm of the matrix $\mathcal{Q}_k$. Thus, it quantifies the process noise level modelled in the filter. The random variations of η reflect the dependence of the process noise W_k on the estimate of the K-matrix (see Eqs. 3.19, 3.20). The optimal filter gain, ρ, decreases exponentially from 0.5 down to 0.001. As expected, the Optimal-REQUEST algorithm behaves like a Kalman filter; that is, initially, it weighs more favorably the new observations in the estimate, and, as the number of processed data grows, it progressively turns to be a pure predictor of the estimate; that is, it weighs less the incoming measurements. The attitude estimated by the Optimal-REQUEST filter converges successfully to the true attitude in the particular case where the noise models of the system are accurately known. Figure 3.3 shows the performance of Optimal-REQUEST compared to that of various cases of REQUEST where the gain, ρ, was chosen as a constant. We chose three different values for ρ; namely, 0.1, 0.01, and 0.001, which, as seen in Fig. 3.3-a are typical values in the span of the optimal gain, ρ^*. Figure 3.3-b depicts the variations of the various MC-means of the angular estimation error, $\delta\phi$. For $\rho = 0.1$, the filter weighs relatively much the measurements, so that after a quick transient the error remains on a relatively high steady-state (0.45 deg), and shows random variations. On the other hand, a very low gain ($\rho = 0.001$) yields a smooth but very slow convergence of the error. As seen in Fig. 3.3-b, the optimal gain

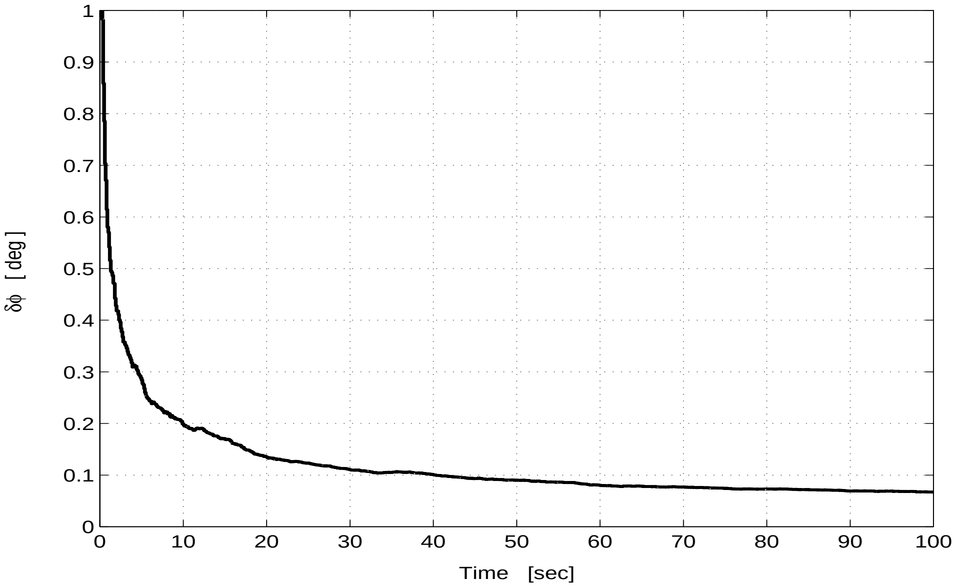


Fig. 3.1: Nominal filter. Monte-Carlo mean of the angular estimation error $\delta\phi$.

in Optimal REQUEST yields a lower bound for the various MC-means of $\delta\phi$, during the transient as well as in the steady-state.

3.7.2 Adaptive Optimal-REQUEST

The results of three Monte-Carlo simulations are presented. The first simulation illustrates the application of the adaptive measurement noise algorithm. The other two simulations are concerned with adaptive process noise estimation.

Adaptive measurement noise estimation

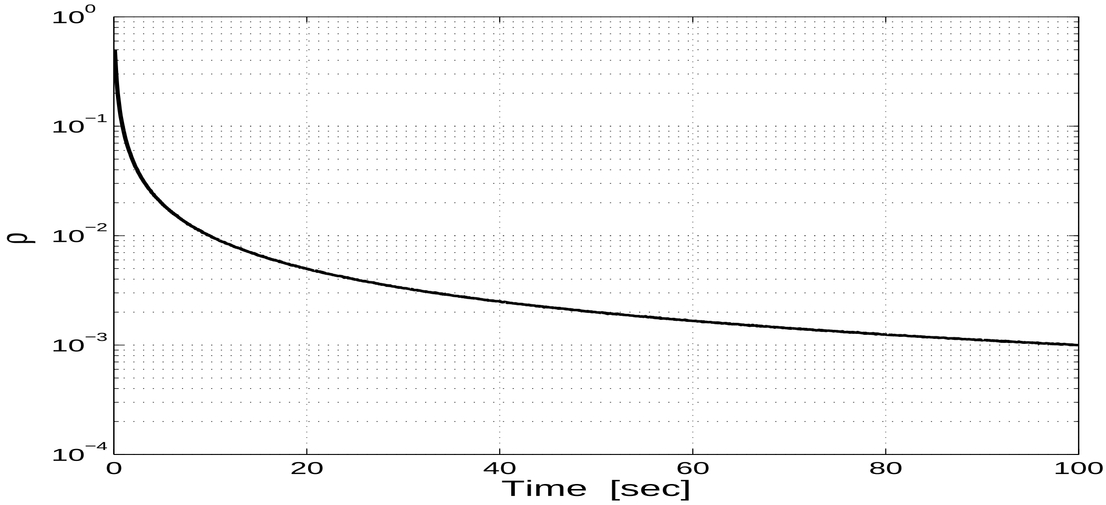
In the present simulation, the true models of the noises are

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-8}\,\text{rad}/sec\right)^2 I_3\right) \qquad \boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-2}\,\text{rad}\right)^2 I_3\right) \tag{3.100}$$

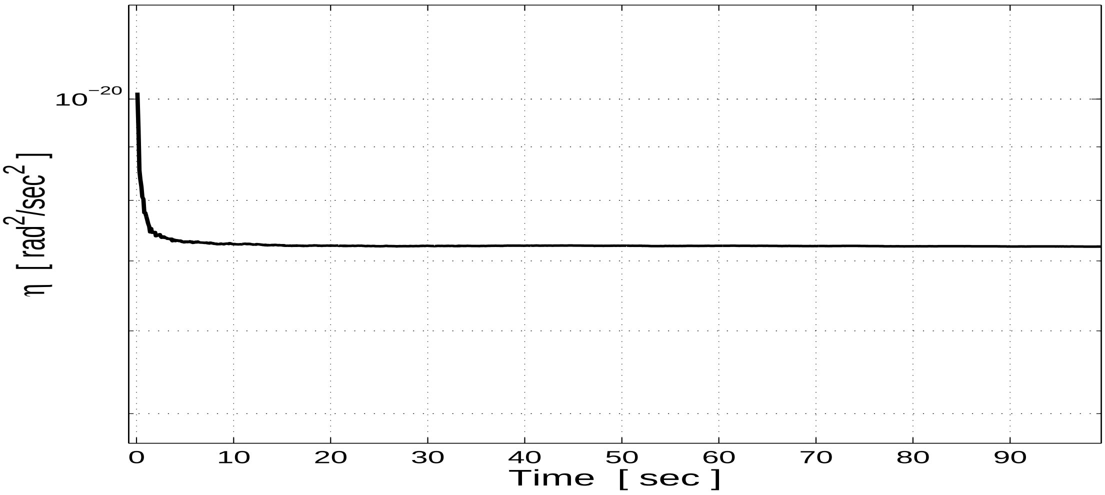
and the filter models of the noises are

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-6}\,\text{rad}/sec\right)^2 I_3\right) \qquad \boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-3}\,\text{rad}\right)^2 I_3\right) \tag{3.101}$$

We want the filter to think that the measurements are very accurate, while the process is very noisy. Thus, we expect that it will stay in a permanent “opened eye” state, with



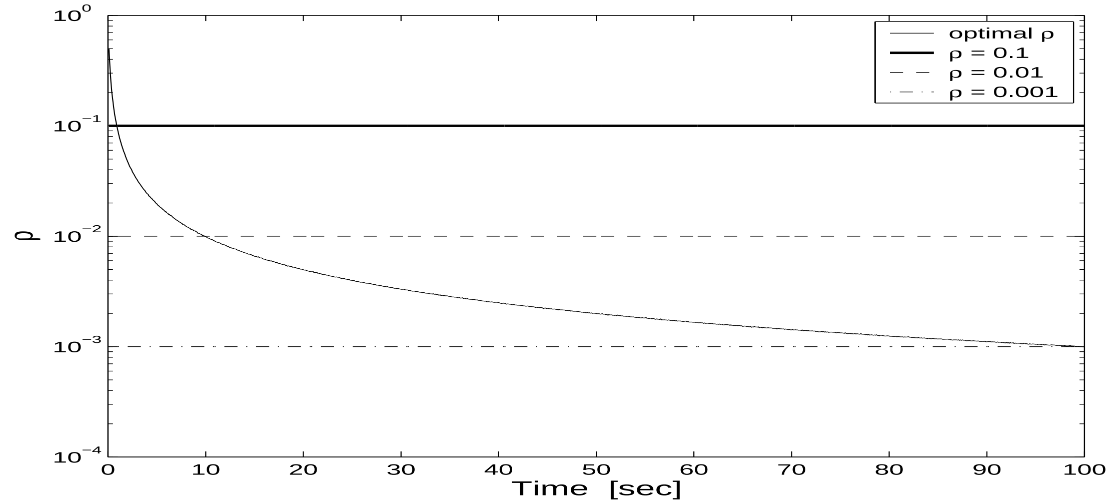
(a) Filter gain



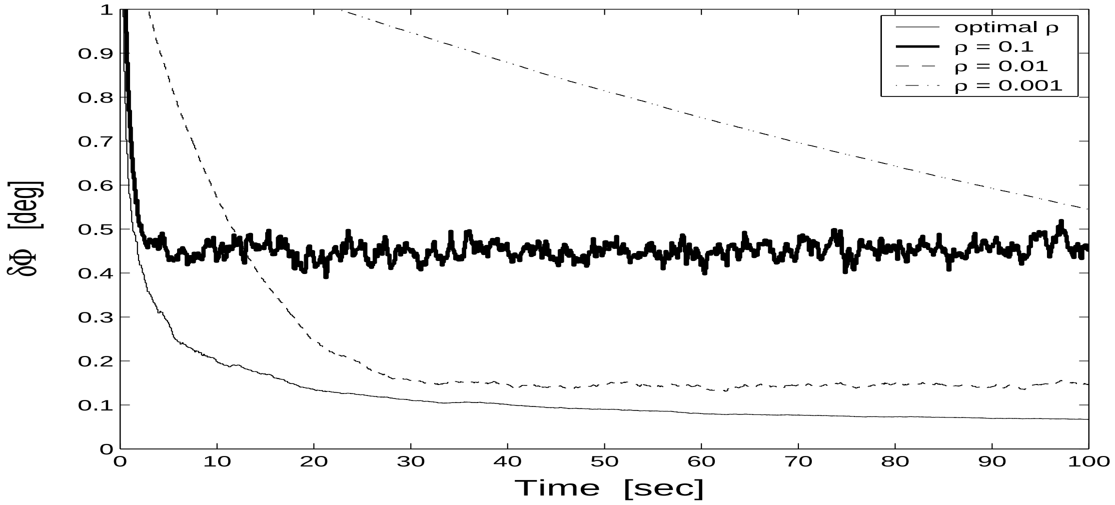
(b) Process noise level

Fig. 3.2: Nominal filter. Time histories of the filter gain ρ and of the process noise level η on a sample run.

a high valued gain weighting preferentially the new observation in the filtered estimate. Then, at $t = 30$ seconds, we apply the adaptive algorithm and expect that the filter will adapt its measurement noise level to the residuals.



(a) Time histories of the gains



(b) MC-means of $\delta\phi$ for 100 runs

Fig. 3.3: Nominal filter. Comparison of Optimal REQUEST and REQUEST with constant gains.

Figure 3.4 shows the time histories of the current squared residuals and of their predicted variance. During the non-adaptive phase of the algorithm, the discrepancy between both processes is manifest. The filter is too “optimistic”, predicting a much smaller level of noise than the current value. When estimating the level of noise, we see that the prediction

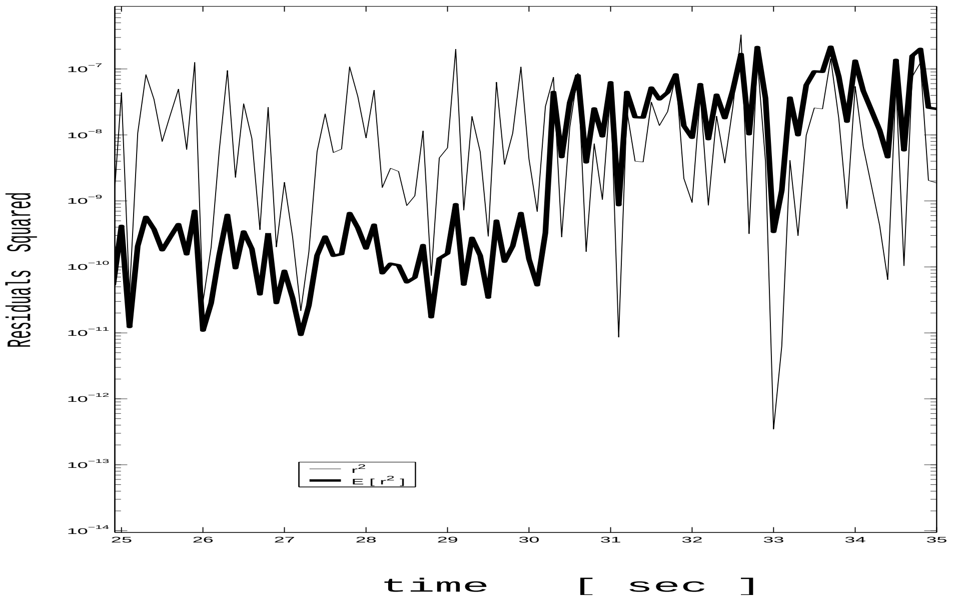


Fig. 3.4: Adaptive measurement noise filter. Difference between the predicted and the current squared residuals on a sample run.

follows more or less the actual history of the squared residuals. Figure 3.5 corroborates the preceding results. It shows the value of the measurement noise, of the process noise in the filter and of the filter gain. We see that the filter succeeds in estimating the true value of the measurement noise, "jumping" from about 0.1 degree up to about 1 degree, in a few seconds. Then, once it has estimated the right value of the noise, it stays at that value. As a consequence, the filter dramatically decreases its gain (see the lower graph). We see that, until $t = 30$ seconds, the gain remained at a high level of about 0.8, that is, 80% of the filtered estimate was constituted of the new observation. but, when the adaptive algorithm is applied, the gain drops and reaches a steady state of about 0.003. Thus the new observation represents only 0.3% in the adaptive filtered estimate. Moreover, the curve tracing the history of the filter process noise becomes less noisy. This can be explained by the fact the $\mathcal{Q}_k$ matrix is constructed using the estimated K-matrix which has become less noisy thanks to the adaptive algorithm. Figure 3.6 summarizes the present simulation by showing the time-wise evolution of the angular estimation error. The non-adaptive phase is characterized by noisy fluctuations of the estimation error about a mean of 0.6 degree. The maxima reach amplitudes of about one degree which is the level

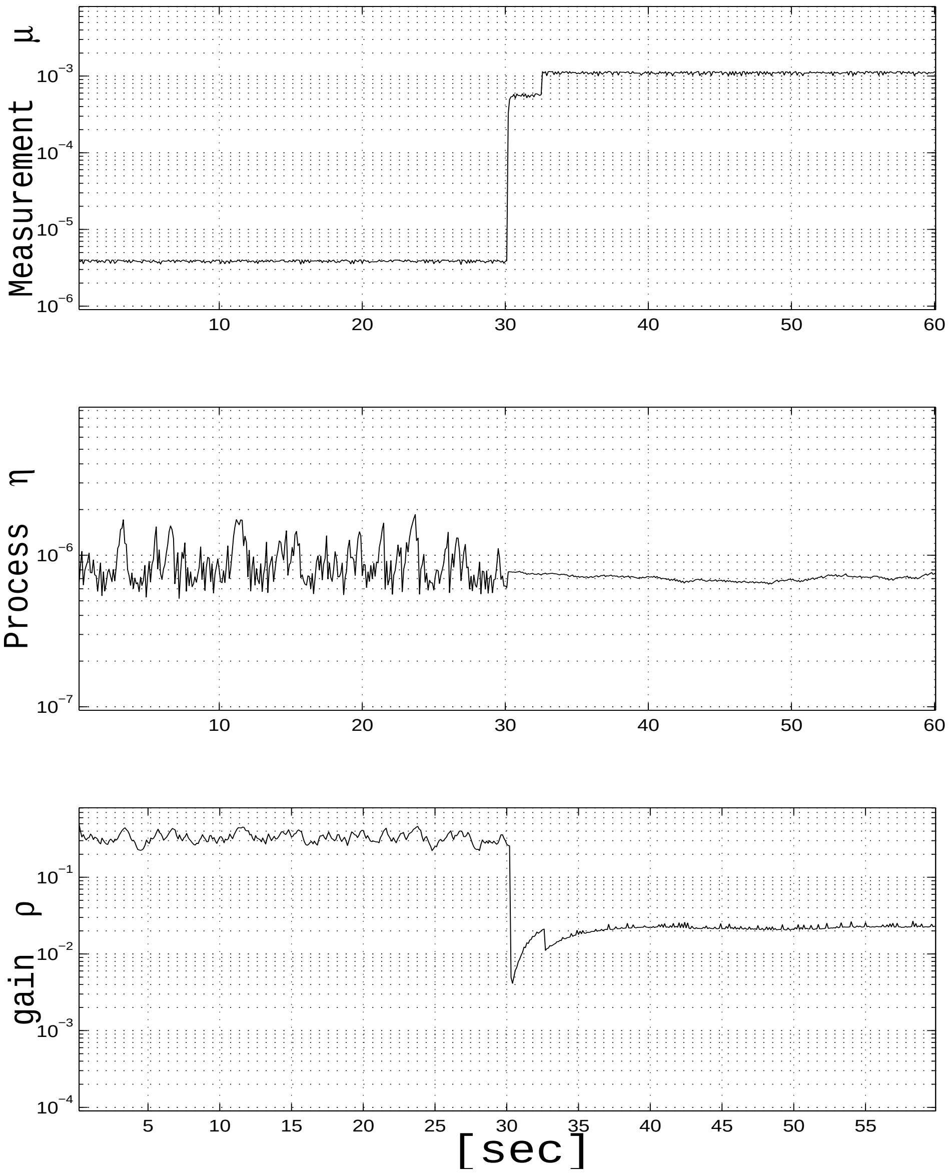


Fig. 3.5: Adaptive measurement noise filter. Filter levels of noise and Filter gain on a sample run.

of uncertainty in the measurement. However, in the adaptive phase, the filter estimation error is decreased down to a steady-state of about 0.15 degree. This level is higher than the nominal one, of 0.1 degree, but shows clearly how the filter succeeded in improving its performances by the use of an adaptive filtering technique.

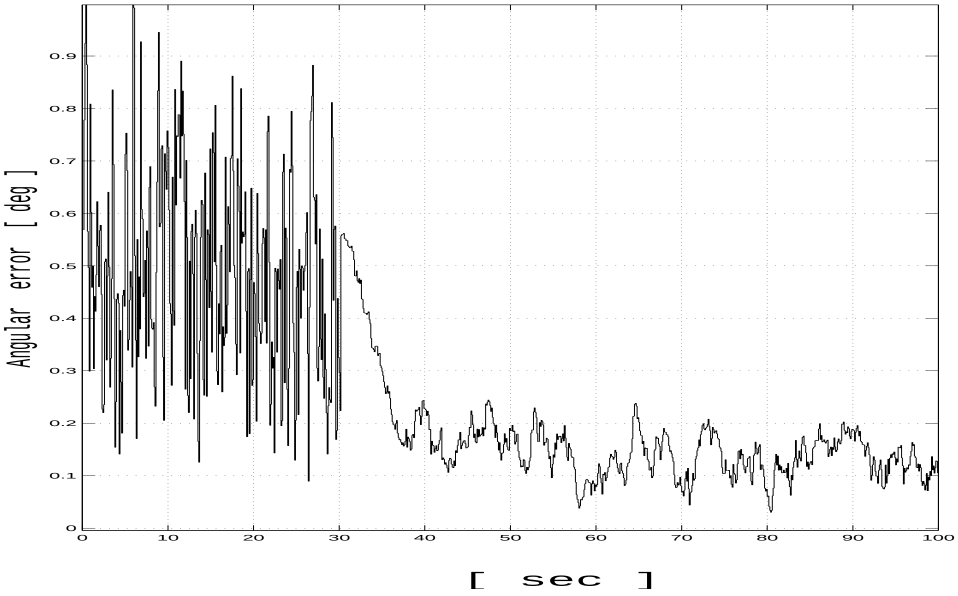


Fig. 3.6: Adaptive measurement noise filter. Angular estimation error $\delta\phi$ on a sample run.

Adaptive process noise estimation

Case 1. In the present simulation, the true models of the noises are

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-5}\,\mathrm{rad}/sec\right)^2 I_3\right) \qquad \boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-2}\,\mathrm{rad}\right)^2 I_3\right) \tag{3.102}$$

and the filter models of the noises are

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-6}\,\mathrm{rad}/sec\right)^2 I_3\right) \qquad \boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, \left(10^{-2}\,\mathrm{rad}\right)^2 I_3\right) \tag{3.103}$$

The filter knows the true level of the measurement noise, but is “optimistic” relative to the process noise level. We expect that it will rapidly reach a steady-state, where it will reject the new observations and, thus, perform the estimation procedure by essentially predicting the estimate. Note that the huge values in the standard deviations of the process noise are chosen in order to check the feasibility of the algorithm over short time runs.

Figures 3.7, 3.8 and 3.9 summarize the simulation results. Figure 3.7 shows the curves of the current residuals squared and of their predicted variance. For the sake of clarity, we plotted only the trace of the related matrices. It appears that the approximation in the

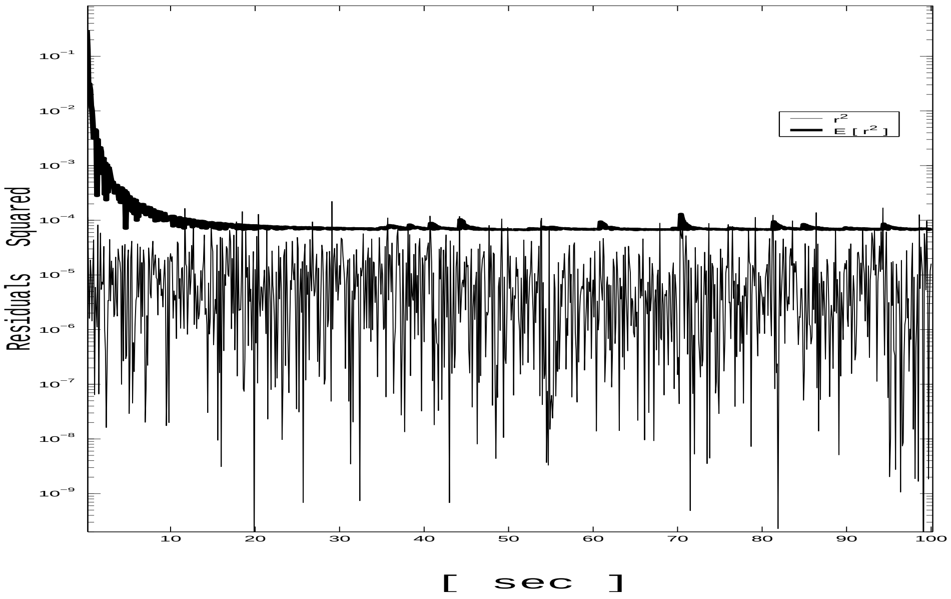


Fig. 3.7: Adaptive process noise filter. Case 1. Difference between the predicted and the current squared residuals on a sample run.

quaternion covariance analysis is quite conservative since the predicted residuals covariance remains on a high valued level relative to the actual residuals covariance level. However, the maxima sometimes overcome their predicted value, so that the adaptive algorithm can work. Figure 3.8 shows indeed, that the filter process noise level stays at the initial value of $0.1^o/hr$, except for some discrete "jumps" where it reaches levels of $10^o/hr$. If we look at the curve of the filter gain, we see that, as expected, it decreases with the time. However, the increase in the estimated process noise level η_k^* yielding an increase in the P-matrix of the predicted estimate, the gain filter is itself increased, at discrete time points. After each "impulse" , the gain follows again its decreasing trend. However, the frequency of the impulses is such, that the gain is maintained at relative high values, about 0.05, thus keeping the filter in an opened eye state. The angular estimation error is plotted in Figure 3.9. The divergence trend is manifest during the non-adaptive phase of the simulation. The filter has learned the wrong K-matrix too well and propagates the estimate using a false dynamic model. This divergence is stopped at $t = 30sec.$ and the error remains on a level of 0.3 degrees during the rest of the run. The filter indeed avoids divergence.

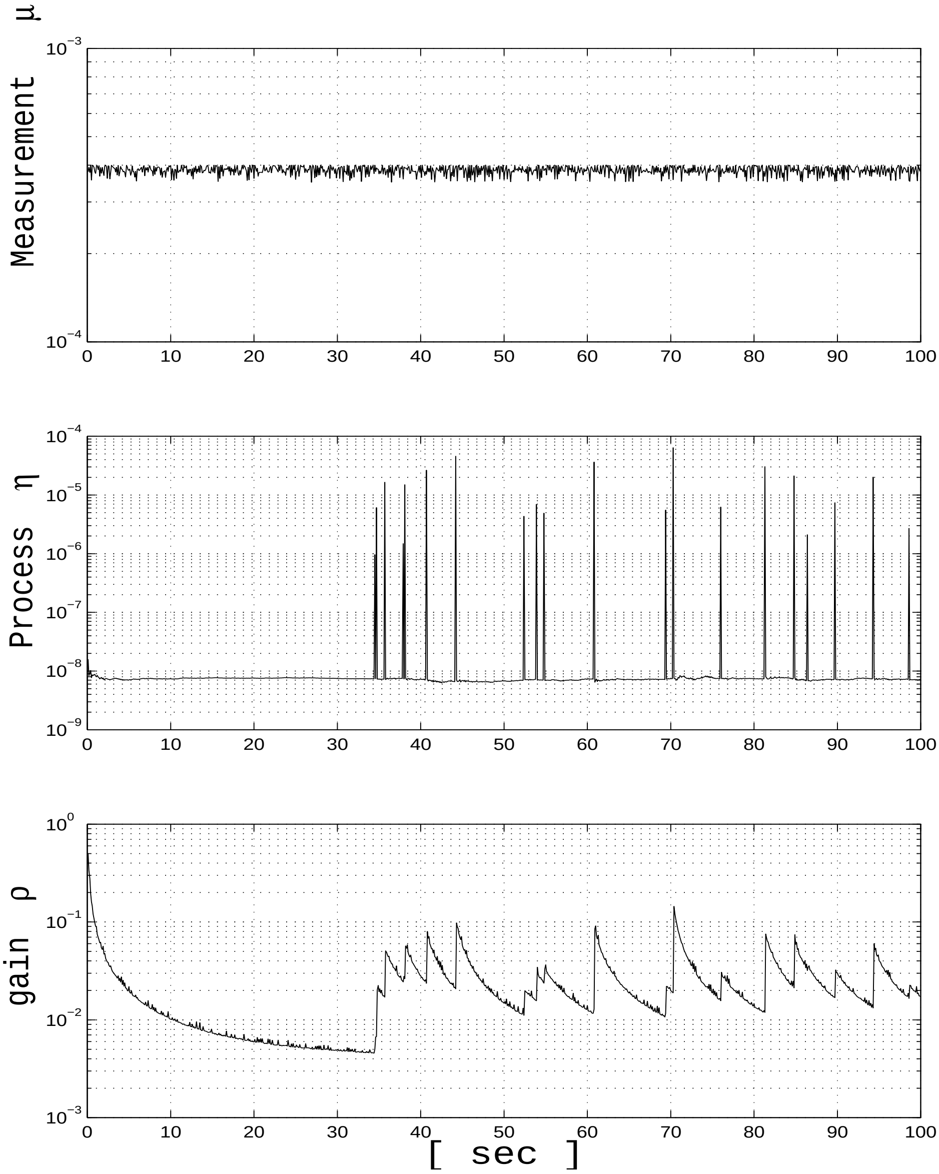


Fig. 3.8: Adaptive process noise filter. Case 1. Filter levels of noise and Filter gain on a sample run.

Case 2. In this simulation we consider the case of unmodeled biases in the gyro outputs. Namely, the gyro outputs are assumed to be corrupted by unknown constant biases of equal magnitudes, 20 deg/hr, and by zero-mean white noises with known standard deviations equal to 0.2 deg/hr. The vector measurement error is modelled as a zero-mean white noise with a known angular standard deviation of 0.5 deg. The sampling rates of the gyros and of

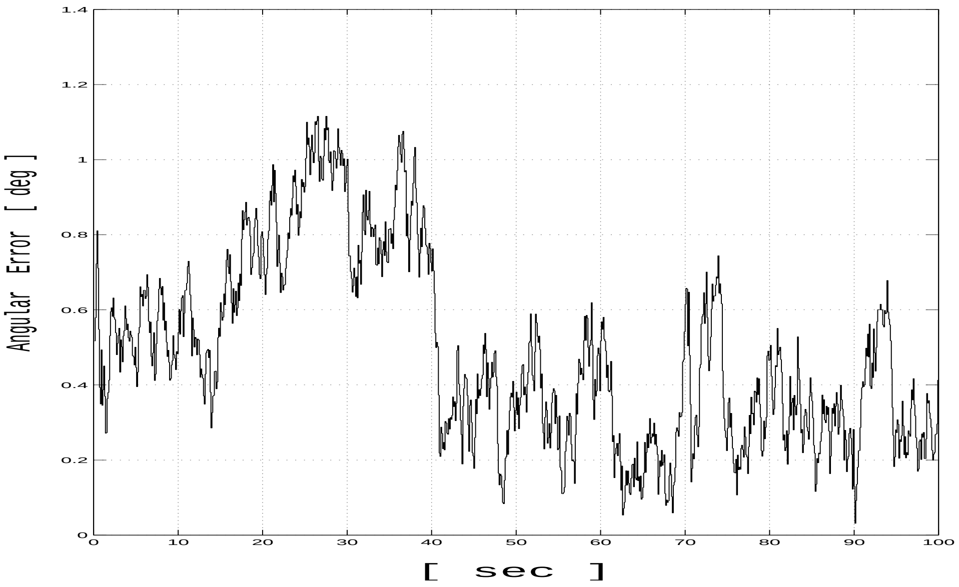
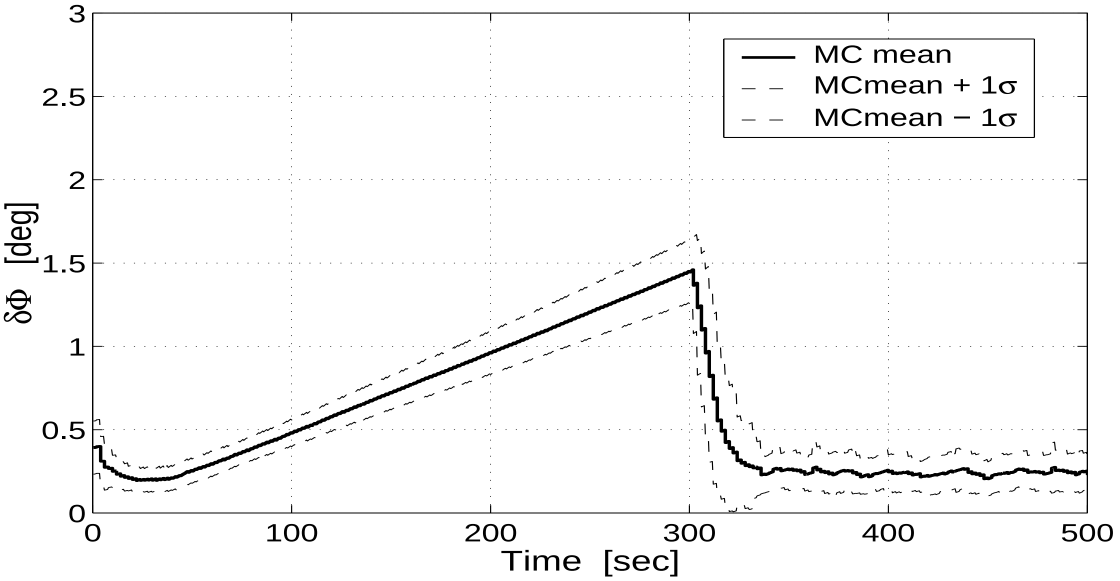
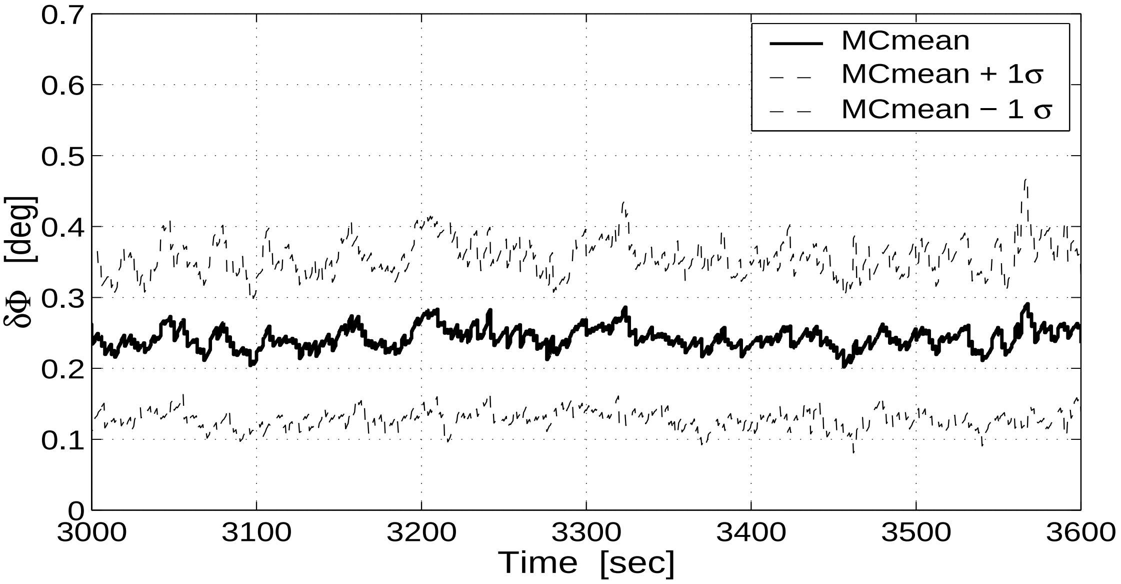


Fig. 3.9: Adaptive process noise filter. Case 1. Angular estimation error $\delta\phi$ on a sample run.

the vector measurement sensor are equal to 0.5 Hz. We ran a Monte-carlo simulation over 100 runs, where each run lasted one hour. For each run the non-adaptive filter is operating during the first 300 seconds, then the adaptive filter is turned on and is operating until the end of the run. The results of this simulation are summarized in Figure 3.10. The figure depicts the time variations of the Monte-Carlo mean and of the $\pm 1\sigma$-envelope of the angular estimation error, $\delta\phi$. Figure 3.10-a shows the non-adaptive phase as well as the converging transient due to the adaptive algorithm. The divergence of the estimation error during the non-adaptive phase is clearly linear in time; this is due to the unmodeled constant gyro biases. After having reached a maximum of 1.5 deg at $t = 300$ sec, the Monte-Carlo mean drops and is stabilizing around a steady-state value of 0.25 deg. The steady-state phase is detailed in Figure 3.10-b. This simulation demonstrates again the effectiveness of the process noise adaptive estimator. It avoids divergence and yields satisfying estimation performance in spite of unmodeled gyro biases.



(a) Non-adaptive phase and transient of the adaptive phase



(b) Adaptive steady-state phase

Fig. 3.10: Adaptive process noise filter. Case 2. Monte-Carlo mean and ±1σ envelope of the angular estimation error $\delta\phi$.

3.8 Concluding Remarks

In this chapter a new recursive optimal estimator for estimating the quaternion-of-rotation from vector measurements was presented. This algorithm, named Optimal-REQUEST, was developed by optimizing the original fading-memory factor of the REQUEST algorithm.

The proposed filter was derived using a unique variance-like performance criterion, which yields a stochastic basis to the estimation strategy. Like a Kalman filter, the proposed algorithm optimally filters both measurement and process noises; thus, it covers a deficiency of REQUEST, where the process noise is filtered in an empirical manner. The special case of a zero-mean white-noise process was considered for the nominal version of Optimal-REQUEST. The approach of adaptive filtering was adopted to cope with modelling errors. Both cases of process noise modelling errors, like unmodeled gyro biases, and measurement noise modelling errors were treated here separately.

Optimal-REQUEST retains also all the features of REQUEST; that is, it is a time varying recursive attitude quaternion estimator, the quaternion unit-norm property is optimally preserved, and attitude is updated even when a single vector observation is processed at each time point. The effectiveness of the nominal and of the adaptive filters were demonstrated through Monte-Carlo simulations.

Chapter 4

A Quadratically Constrained Batch-Recursive Least-Squares Quaternion Filter

4.1 Introduction

This chapter is concerned with the development of a quadratically constrained (QC) batch-recursive quaternion filter in the framework of classical least squares (LS) theory. As mentioned in Chapter 1 an attractive feature of the LS approach is that it allows to embed the quaternion unit norm quadratic constraint in the formulation of the estimation problem. It was also noted that the elegant **q**-method and some of its derivatives [3, 17, 24] become suboptimal when filtering propagation noises. Besides, the various solutions devised in order to optimally filter the propagation noises [20, 21] generally include heavy numerical procedures. In the present work a simple closed-form QC LS quaternion estimator is proposed that optimally filters vector measurement and propagation noises. It also estimates parameters other than attitude, like constant biases in gyro outputs, and angular velocity. For the latter purpose a simple smoothing procedure is presented that assume a priori knowledge of the biases. The smoothing technique avoids the classical state-augmentation technique, which would increase the filter computational burden.

The general idea in the present filter is that the incoming data is processed in a two-stage algorithm. In the first stage, a batch of n simultaneous vector measurements ($n \geq 2$)

is acquired at each sampling time and processed to yield a single-frame estimate of the quaternion[1]. This quaternion estimate is henceforth considered as a quaternion measurement. Using these measurements along with *scalar* weights, which reflect their respective accuracy, a QC weighted LS problem is formulated. In order to solve this problem in the case of time-invariant systems, two different approaches are adopted: the Lagrange Multipliers (LM) approach, and the Dynamic Programming (DP) approach. The latter is inspired by the Dynamic Programming theory of optimization as developed by Bellman (see [15, p. 140]). The application of DP to optimal state estimation was first introduced in [55]. We extend that application here in order to incorporate the quaternion unit-norm quadratic constraint. The case of time-varying systems is treated via the DP approach.

In section 4.2 we treat the case of a time-invariant system using both the LM and the DP methods. The performance of the static filter is examined using simulations. The case of a time-varying system is handled in section 4.3 following the use of the DP method for estimation of the optimal quaternion. The efficiency of the dynamic filter estimate of attitude and constant gyro biases is illustrated through simulation. Finally the conclusions of this chapter are proposed in the last section.

4.2 Time-Invariant Case

4.2.1 Model Description

We begin with the case of a time-invariant system, that is, the Body frame $\mathcal{B}$ is in a constant attitude relative to the Reference frame $\mathcal{R}$. We assume, that at each sampling time t_k, n vector measurements are performed ($n \geq 2$), so that a unique unit-norm estimate of the attitude quaternion can be computed. Let $\{\overline{\mathbf{q}}_k\}$ be the unit-norm single-frame estimate at time t_k. We write

$$\overline{\mathbf{q}}_k = \mathbf{q}_k + \overline{\mathbf{v}}_k \tag{4.1}$$

[1] The single-frame estimator can be a batch estimator, like the $\mathbf{q}$-method or QUEST, or a recursive filter, like the static REQUEST algorithm.

where $\mathbf{q}_k$ is the true quaternion at time t_k (which is here constant), and $\overline{\mathbf{v}}_k$ is the error in $\overline{\mathbf{q}}_k$. The following quaternion state-space model is defined

$$\mathbf{q}_{k+1} = \mathbf{q}_k \tag{4.2}$$

$$\overline{\mathbf{q}}_{k+1} = \mathbf{q}_{k+1} + \overline{\mathbf{v}}_{k+1} \tag{4.3}$$

Assuming that N successive quaternion measurements are acquired at N successive epoch times a performance index, denoted by $J(\mathbf{q})$, is defined as

$$J(\mathbf{q}) = \frac{1}{2}\sum_{k=1}^{N} \alpha_k \|\overline{\mathbf{q}}_k - \mathbf{q}\|^2 \tag{4.4}$$

where $\|M\|$ denotes the Euclidean norm, that is, $\|\mathbf{x}\| \overset{\triangle}{=} \sqrt{\mathbf{x}^T\mathbf{x}}$, and α_k, $k = 1, 2, \ldots N$ are N positive weighting scalars. Since any quaternion defining the attitude of a coordinate frame has a unit norm, it is desired to minimize $J(\mathbf{q})$ over the unit sphere. We begin by deriving a batch solution of this estimation problem. Then we convert it to its recursive form. The derivation is closely related to the classical derivation of the unconstrained recursive least squares estimator (see [40, p. 30]).

4.2.2 Lagrange Multipliers Approach

Batch Estimator

We assume that, at time t_N, a sequence of estimates $\overline{\mathbf{q}}_k$, $k = 1 \ldots N$, was computed and that the respective coefficients α_k were evaluated. The cost function in Eq. (4.4) becomes after expansion

$$J_N(\mathbf{q}_N) = \frac{1}{2}\alpha_1\|\overline{\mathbf{q}}_1 - \mathbf{q}_N\|^2 + \frac{1}{2}\alpha_2\|\overline{\mathbf{q}}_2 - \mathbf{q}_N\|^2 + \ldots + \frac{1}{2}\alpha_N\|\overline{\mathbf{q}}_N - \mathbf{q}_N\|^2 \tag{4.5}$$

Note that the subscript N in $\mathbf{q}_N$ denotes the fact that $\mathbf{q}$ is estimated at time t_N. The true quaternion actually does not change. Nevertheless this notation is adopted in preparation to the time varying case. In order to minimize the cost $J_N(\mathbf{q}_N)$ for $\mathbf{q}_N$ over the unit sphere, the Lagrange Multipliers method is used. Let f_N be the quadratic constraint

$$f_N(\mathbf{q}_N) = \frac{1}{2}\left(\mathbf{q}_N^T\mathbf{q}_N - 1\right) = 0 \tag{4.6}$$

Let μ_N denotes the associated Lagrange multiplier. Then we form the Lagrangian, L_N, as follows

$$L_N(\mathbf{q}_N) = J_N + \mu_N\, f_N \qquad \mu_N \,\epsilon\, \mathrm{R} \tag{4.7}$$

The stationarity conditions are expressed as

$$\frac{\partial L_N}{\partial \mathbf{q}_N}(\mathbf{q}_N^*, \mu_N^*) = \mathbf{0} \tag{4.8}$$

$$\frac{\partial L_N}{\partial \mu_N}(\mathbf{q}_N^*, \mu_N^*) = 0 \tag{4.9}$$

which means that the Jacobian of the Lagrangian is necessarily null at an extremum point. Solving Eq. (4.8) for $\mathbf{q}_N$ yields the optimal estimate, $\mathbf{q}_N^*$, as a function of the optimal Lagrange multiplier, μ_N^*,

$$\mathbf{q}_N^* = \frac{\alpha_1 \,\overline{\mathbf{q}}_1 \,+\, \alpha_2 \,\overline{\mathbf{q}}_2 \,+\, \ldots \,+\, \alpha_N \,\overline{\mathbf{q}}_N}{\alpha_1 \,+\, \alpha_2 \,+\, \ldots\, \alpha_N \,+\, \mu_N^*} \tag{4.10}$$

Substituting Eq. (4.10) into Eq. (4.9) and rearranging shows that μ_N can have two optimal values corresponding to the two equations

$$(\alpha_1 \,+\, \ldots \,+\, \alpha_N \,+\, \mu_N^*) = \pm \,\|\alpha_1 \,\overline{\mathbf{q}}_1 \,+\, \ldots \,+\, \alpha_N \,\overline{\mathbf{q}}_N\| \tag{4.11}$$

Turning to the sufficient condition one obtains

$$\left[\frac{\partial^2 J}{\partial \mathbf{q} \partial \mathbf{q}^T} + \mu \,\frac{\partial^2 f}{\partial \mathbf{q} \partial \mathbf{q}^T}\right]_{\mathbf{q}^*, \mu^*} > \mathbf{0} \tag{4.12}$$

which means that, in order for an extremum to be a local minimum, the Hessian of the Lagrangian has to be positive definite at that point. The subscript N has been omitted in the above condition for the sake of clarity. The sufficient condition is then

$$(\alpha_1 \,+\, \ldots \,+\, \alpha_N \,+\, \mu_N^*) \,>\, 0 \tag{4.13}$$

Using Eq. (4.13) the ambiguity of Eq. (4.11) vanishes and the solution is unique. As a conclusion, the optimal values for μ_N, $\mathbf{q}_N$ are

$$\mathbf{q}_N^* \;=\; \frac{\alpha_1 \,\overline{\mathbf{q}}_1 \,+\, \ldots \,+\, \alpha_N \,\overline{\mathbf{q}}_N}{\|\alpha_1 \,\overline{\mathbf{q}}_1 \,+\, \ldots \,+\, \alpha_N \,\overline{\mathbf{q}}_N\|} \tag{4.14}$$

$$\mu_N^* \;=\; \|\alpha_1 \,\overline{\mathbf{q}}_1 \,+\, \ldots \,+\, \alpha_N \,\overline{\mathbf{q}}_N\| - (\alpha_1 \,+\, \ldots \,+\, \alpha_N) \tag{4.15}$$

The vector $\mathbf{q}_N^*$, which is the optimal batch estimate of the attitude quaternion $\mathbf{q}_N$ for a given sequence $\{\overline{\mathbf{q}}_k\}$, $k = 1 \ldots N$, will be denoted in the following by $\widehat{\mathbf{q}}_{N/N}$. In order to check the validity of the estimator, we assume that $\overline{\mathbf{q}}_k = \mathbf{q}$ for all k in Eq. (4.14), then using the unit-norm property of $\mathbf{q}$ yields $\widehat{\mathbf{q}}_{N/N} = \mathbf{q}_N^* = \mathbf{q}$. This shows the unbiasedness of the estimator.

Recursive Estimator

State Estimate Recursion In order to derive the recursive estimator, it is assumed, in a first step, that an estimate $\widehat{\mathbf{q}}_N$ has been computed using Eq. (4.14) at time t_N and all α's have been computed. Let $\widehat{\mathbf{q}}_{N/N}$ be the estimate of the attitude quaternion at time t_N, given a sequence of N measurements $\{\overline{\mathbf{q}}_k\}$, $k = 1 \ldots N$, then from Eq. (4.14)

$$\widehat{\mathbf{q}}_{N/N} = \frac{\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N}{\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\|} \tag{4.16}$$

It is assumed that at time t_{N+1} n new observations ($n \geq 2$) are acquired. Using these new observations a new measurement of the quaternion, $\overline{\mathbf{q}}_{N+1}$, is computed. The associated weight, α_{N+1}, is given. The new cost function is now

$$\begin{aligned} J_N(\mathbf{q}_{N+1}) \;&= \\ &\tfrac{1}{2}\,\alpha_1\,\|\overline{\mathbf{q}}_1 - \mathbf{q}_{N+1}\|^2\; + \ldots + \;\tfrac{1}{2}\,\alpha_N\,\|\overline{\mathbf{q}}_N - \mathbf{q}_{N+1}\|^2\; + \;\tfrac{1}{2}\,\alpha_{N+1}\,\|\overline{\mathbf{q}}_{N+1} - \mathbf{q}_{N+1}\|^2 \end{aligned} \tag{4.17}$$

Using again the batch formula of Eq. (4.14), an estimate of $\mathbf{q}$ at time t_{N+1}, based on $(N+1)$ measurements, is obtained by

$$\widehat{\mathbf{q}}_{N+1/N+1} = \frac{\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}}{\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\|} \tag{4.18}$$

The ratio in Eq. (4.18) can be separated into the sum of two ratios

$$\begin{aligned} \widehat{\mathbf{q}}_{N+1/N+1} = &\frac{\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N}{\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\|} \\ + &\frac{\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}}{\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\|} \end{aligned} \tag{4.19}$$

and using Eq. (4.16),

$$\begin{aligned} &\widehat{\mathbf{q}}_{N+1/N+1} = \\ &\frac{\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\|}{\|\,(\,\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\|\,)\;\widehat{\mathbf{q}}_{N/N}\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\,\|}\;\widehat{\mathbf{q}}_{N/N}\; + \\ &\frac{\alpha_{N+1}}{\|\,(\,\|\alpha_1\,\overline{\mathbf{q}}_1\, + \,\ldots\, + \,\alpha_N\,\overline{\mathbf{q}}_N\|\,)\;\widehat{\mathbf{q}}_{N/N}\, + \,\alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\,\|}\;\overline{\mathbf{q}}_{N+1} \end{aligned} \tag{4.20}$$

According to Eq. (4.20) it seems that the computation of $\widehat{\mathbf{q}}_{N+1/N+1}$ requires the simultaneous knowledge of the sequence of measurements $\{\overline{\mathbf{q}}_k\}$, $k = 1 \ldots N$. Thus, the desired feature of a recursive estimator, i.e., to process *only* the former estimate and the new measurement for computing the new estimate, is not achieved yet. This difficulty will be overcome next.

Weights Recursion The recursion algorithm would be complete if the weighting coefficients in Eqs. (4.20) and (4.20) were recursively obtained. For this purpose a sequence of positive scalars, $\beta_{k/k}$, $k = 1 \ldots N$, is defined by

$$\beta_{k/k} \overset{\triangle}{=} \|\alpha_1 \overline{\mathbf{q}}_1 \,+\, \ldots \,+\, \alpha_k \overline{\mathbf{q}}_k\| \tag{4.21}$$

It appears, from Eqs.(4.21) and (4.16), that this sequence can be recursively defined as follows

$$\beta_{k/k} \overset{\triangle}{=} \|\beta_{k-1/k-1}\, \widehat{\mathbf{q}}_{k-1/k-1} \,+\, \alpha_k \overline{\mathbf{q}}_k\| \tag{4.22}$$

and, at time t_{N+1}, $\beta_{N+1/N+1}$ is computed by

$$\beta_{N+1/N+1} = \|\beta_{N/N}\, \widehat{\mathbf{q}}_{N/N} \,+\, \alpha_{N+1} \overline{\mathbf{q}}_{N+1}\| \tag{4.23}$$

Initialization Assume that, at time t_1, n vector observations ($n \geq 2$) are acquired and processed to yield the estimate $\overline{\mathbf{q}}_1$. Assume furthermore that the associated weighting coefficient α_1 is given. Consider the problem of minimizing the function

$$J_1(\mathbf{q}_1) = \frac{1}{2}\alpha_1 \|\overline{\mathbf{q}}_1 - \mathbf{q}_1\|^2 \tag{4.24}$$

over the unit sphere. Substituting $\overline{\mathbf{q}}_1$ for $\mathbf{q}_1$ in Eq. (4.24) brings the cost to 0. Since the cost function is non-negative for all $\mathbf{q}_1$, then 0 is its global minimum and

$$\widehat{\mathbf{q}}_{1/1} = \overline{\mathbf{q}}_1 \tag{4.25}$$

That is, $\overline{\mathbf{q}}_1$ is the filtered estimate at time t_1. Using the definition of the sequence $\{\beta_{k/k}\}$ (Eq. 4.21) at $k = 1$ yields

$$\beta_{1/1} = \|\, \alpha_1 \overline{\mathbf{q}}_1 \,\| = \alpha_1 \|\overline{\mathbf{q}}_1\| = \alpha_1 \tag{4.26}$$

Summary of the State Estimate and Weight Coefficients Recursions

1. At time t_1

$$\begin{aligned} \beta_{1/1} &= \alpha_1 \\ \widehat{\mathbf{q}}_{1/1} &= \overline{\mathbf{q}}_1 \end{aligned} \tag{4.27}$$

2. At time t_{N+1}

$$\begin{aligned} \beta_{N+1/N+1} &= \|\, \beta_{N/N}\, \widehat{\mathbf{q}}_{N/N} \,+\, \alpha_{N+1} \overline{\mathbf{q}}_{N+1} \,\| \\ \widehat{\mathbf{q}}_{N+1/N+1} &= \tfrac{\beta_{N/N}}{\beta_{N+1/N+1}}\, \widehat{\mathbf{q}}_{N/N} \,+\, \tfrac{\alpha_{N+1}}{\beta_{N+1/N+1}} \overline{\mathbf{q}}_{N+1} \end{aligned} \tag{4.28}$$

Minimal Costs Recursion It is of some interest, although not essential to the estimation process, to determine the sequence of the minimal costs. Let the minimal cost $\widehat{J}_{N/N}$ be the value of the cost function in Eq. (4.5) when $\mathbf{q}_N = \widehat{\mathbf{q}}_{N/N}$, that is

$$\widehat{J}_{N/N} = J_N(\widehat{\mathbf{q}}_{N/N}) \tag{4.29}$$

The sequence $\{\widehat{J}_{k/k}\}$, $k = 1 \ldots N$, is the sequence of positive real numbers that is to be evaluated. One way to evaluate $\widehat{J}_{N+1/N+1}$ is, for instance, to substitute $\widehat{\mathbf{q}}_{N+1/N+1}$ for $\mathbf{q}_{N+1}$ in Eq. (4.17) and to compute all the terms of the sum. But this requires knowing simultaneously the whole sequences of past measurements and of their relative weights. Therefore the advantage of having a recursive estimation algorithm would be lost. However the sequence can be recursively computed as follows. At time t_N, the minimal cost is

$$\begin{aligned}\widehat{J}_{N/N} &= \frac{1}{2}\alpha_1\|\overline{\mathbf{q}}_1 - \widehat{\mathbf{q}}_{N/N}\|^2 + \ldots + \frac{1}{2}\alpha_N\|\overline{\mathbf{q}}_N - \widehat{\mathbf{q}}_{N/N}\|^2 &(4.30)\\ &= \frac{1}{2}\alpha_1(\overline{\mathbf{q}}_1^T\overline{\mathbf{q}}_1 - 2\overline{\mathbf{q}}_1^T\widehat{\mathbf{q}}_{N/N} + \widehat{\mathbf{q}}_{N/N}^T\widehat{\mathbf{q}}_{N/N}) + \ldots &(4.31)\end{aligned}$$

Using the unit property of $\overline{\mathbf{q}}_k$ and of $\widehat{\mathbf{q}}_{N/N}$, and the expression for $\widehat{\mathbf{q}}_{N/N}$ given in Eq. (4.16) yields

$$\begin{aligned}\widehat{J}_{N/N} &= \frac{1}{2}\alpha_1(2 - 2\overline{\mathbf{q}}_1^T\widehat{\mathbf{q}}_{N/N}) + \ldots + \frac{1}{2}\alpha_N(2 - 2\overline{\mathbf{q}}_N^T\widehat{\mathbf{q}}_{N/N}) &(4.32)\\ &= (\alpha_1 + \ldots + \alpha_N) - (\alpha_1\overline{\mathbf{q}}_1 + \ldots + \alpha_N\overline{\mathbf{q}}_N)^T\widehat{\mathbf{q}}_{N/N} &(4.33)\\ &= (\alpha_1 + \ldots + \alpha_N) - \|\alpha_1\overline{\mathbf{q}}_1 + \ldots + \alpha_N\overline{\mathbf{q}}_N\| &(4.34)\\ &= -\mu_N^\star &(4.35)\end{aligned}$$

where $\mu_N^\star$ is the optimal Lagrange multiplier shown in Eq. (4.15). Using the definition of the β-coefficient, as given in Eq. (4.21), yields

$$\widehat{J}_{N/N} = (\alpha_1 + \ldots + \alpha_N) - \beta_{N/N} = -\mu_N^\star \tag{4.36}$$

At time t_{N+1}, identical computations lead to the following result

$$\widehat{J}_{N+1/N+1} = (\alpha_1 + \ldots + \alpha_N + \alpha_{N+1}) - \beta_{N+1/N+1} \tag{4.37}$$

Subtracting Eq. (4.36) from Eq. (4.37) yields

$$\widehat{J}_{N+1/N+1} = \widehat{J}_{N/N} + \alpha_{N+1} - (\beta_{N+1/N+1} - \beta_{N/N}) \tag{4.38}$$

Equation (4.38) is the sought for recursion on the minimal costs. The minimal cost initialization is done by recalling that at time t_1

$$J_1(\mathbf{q}_1) = \frac{1}{2}\,\alpha_1\,\|\overline{\mathbf{q}}_1 - \mathbf{q}_1\|^2 \tag{4.39}$$

which yields

$$\widehat{J}_{1/1} = J_1(\overline{\mathbf{q}}_1) = 0 \tag{4.40}$$

Summary of the Filter using the Lagrange Multiplier Approach

1. Initialization Stage:

$$\widehat{\mathbf{q}}_{1/1} = \overline{\mathbf{q}}_1 \tag{4.41a}$$

$$\beta_{1/1} = \alpha_1 \tag{4.41b}$$

$$\widehat{J}_{1/1} = 0 \tag{4.41c}$$

2. Measurement Update Stage:

$$\beta_{N+1/N+1} = \|\,\beta_{N/N}\,\widehat{\mathbf{q}}_{N/N} \;+\; \alpha_{N+1}\,\overline{\mathbf{q}}_{N+1}\,\| \tag{4.42a}$$

$$\widehat{\mathbf{q}}_{N+1/N+1} = \frac{\beta_{N/N}}{\beta_{N+1/N+1}}\,\widehat{\mathbf{q}}_{N/N} \;+\; \frac{\alpha_{N+1}}{\beta_{N+1/N+1}}\,\overline{\mathbf{q}}_{N+1} \tag{4.42b}$$

$$\widehat{J}_{N+1/N+1} = \widehat{J}_{N/N} + \alpha_{N+1} - (\beta_{N+1/N+1} - \beta_{N/N}) \tag{4.42c}$$

Relation to the Unconstrained Least-Squares Problem Assume that the model in Eq. (4.2) for the system and for the measurement equations is given and that the minimization problem in Eq. (4.4) is to be solved without the quadratic constraint of Eq. (4.6). This is the unconstrained least squares problem (see for instance [40, p. 30]), which solution is here

$$\widehat{\mathbf{q}}^{LS}_{N/N} = \frac{\alpha_1\,\overline{\mathbf{q}}_1 \;+\; \ldots \;+\; \alpha_N\,\overline{\mathbf{q}}_N}{(\alpha_1 \;+\; \ldots \;+\; \alpha_N)} \tag{4.43}$$

The solution of the quadratic constrained problem is seen to be the normalized solution of the unconstrained problem. Note that normalization is here embedded in the optimization procedure.

4.2.3 Dynamic Programming Approach

Introduction

In the present section we propose an alternate approach, namely a Dynamic Programming (DP) approach to tackle the problem of minimizing the cost function, $J(\mathbf{q})$, which is rewritten here

$$J(\mathbf{q}) = \frac{1}{2}\sum_{k=1}^{N} \alpha_k \|\overline{\mathbf{q}}_k - \mathbf{q}\|^2 \tag{4.44}$$

One central virtue of this approach is the inherent recursive formulation of the optimization problem. The latter fact yields in a natural way a recursive algorithm for the solution of the problem. In this work we use the DP technique, as introduced in [55] for the purpose of state variables estimation, and extend it to incorporate the quaternion estimate unit-norm constraint.

The main purpose of this section is to validate the DP-based method by comparing its solution with the one obtained using the Lagrange Multipliers. The DP method is described next.

Dynamic Programming Method

Filtering Problem at time t_1 Assume that, at t_1, n independent vector observations ($n \geq 2$) are acquired. That is, α_1 and $\overline{\mathbf{q}}_1$ are given. Let the cost function $J_{1/1}(\mathbf{q}_1)$ be

$$J_{1/1}(\mathbf{q}_1) = \frac{1}{2}\,\alpha_1\,\|\overline{\mathbf{q}}_1 - \mathbf{q}_1\|^2 \tag{4.45}$$

Then the filtering problem, at time t_1, is formulated as

$$\min_{\mathbf{q}_1^T\,\mathbf{q}_1=1} \{J_{1/1}(\mathbf{q}_1)\} \tag{4.46}$$

The vector $\overline{\mathbf{q}}_1$ brings the cost $J_{1/1}$ to zero, which is its global minimum. Since $\overline{\mathbf{q}}_1$ has a unit-norm, it is the solution of (4.46) so that

$$\widehat{\mathbf{q}}_{1/1} = \overline{\mathbf{q}}_1 \tag{4.47}$$

that is, the filtered estimate of $\mathbf{q}_1$ at time t_1 is $\overline{\mathbf{q}}_1$. Then, defining

$$\gamma_{1/1} \overset{\triangle}{=} \alpha_1 \tag{4.48}$$

the cost function (4.45) can be rewritten

$$J^{\star}_{1/1}(\mathbf{q}_1) = \frac{1}{2}\gamma_{1/1} \, \| \, \widehat{\mathbf{q}}_{1/1} - \mathbf{q}_1 \|^2 \; + \; \widehat{J}_{1/1} \tag{4.49}$$

where

$$\widehat{J}_{1/1} = 0 \tag{4.50}$$

Finally, the problem stated in Eq. (4.46) is equivalent to the following problem

$$\min_{\mathbf{q}_1^T \, \mathbf{q}_1 = 1} \; \{ \, \frac{1}{2}\gamma_{1/1} \, \| \, \widehat{\mathbf{q}}_{1/1} - \mathbf{q}_1 \|^2 \, \} \tag{4.51}$$

The procedure of rewriting the cost function as a function of $\mathbf{q}_1$ and of $\widehat{\mathbf{q}}_{1/1}$, its a posteriori estimate at time t_1, *is the cornerstone of the DP-method.* As will be shown in the following, this allows a sequential formulation and solution of the estimation problem. Note that the weighing coefficient $\gamma_{1/1}$ of $\widehat{\mathbf{q}}_{1/1}$ is defined throughout this procedure.

Comparison between LM-method and DP-method at this stage:
It can be seen, from Eqs.(4.27) and (4.47),(4.48), that, at time t_1, the first estimate and the coefficients β and γ are identical in both methods. Moreover, the minimal cost vanishes in both methods (Eqs. 4.41,4.50).

Filtering Problem at time t_2 Assume $\widehat{\mathbf{q}}_{1/1}$ and $\gamma_{1/1}$ to be known. Assume that n new vector observations ($n \geq 2$)are acquired, so that α_2 and $\overline{\mathbf{q}}_2$ are also given. Let the cost function at time t_2 be

$$J_{2/2}(\mathbf{q}_2) = J^{\star}_{1/1}(\mathbf{q}_2) \; + \; \frac{1}{2}\alpha_2 \, \|\overline{\mathbf{q}}_2 - \mathbf{q}_2\|^2 \tag{4.52}$$

Then the filtering problem at time t_2 is formulated as

$$\min_{\mathbf{q}_2^T \mathbf{q}_2 = 1} \; \{ \, J_{2/2}(\mathbf{q}_2) = J^{\star}_{1/1}(\mathbf{q}_2) \; + \; \frac{1}{2}\alpha_2 \, \|\overline{\mathbf{q}}_2 - \mathbf{q}_2\|^2 \, \} \tag{4.53}$$

Substituting for $J^{\star}_{1/1}$ its expression as obtained from Eq. (4.49), one gets

$$J_{2/2}(\mathbf{q}_2) = \frac{1}{2}\gamma_{1/1} \, \| \, \widehat{\mathbf{q}}_{1/1} - \mathbf{q}_2 \|^2 \; + \; \frac{1}{2}\alpha_2 \, \|\overline{\mathbf{q}}_2 - \mathbf{q}_2\|^2 \; + \; \widehat{J}_{1/1} \tag{4.54}$$

Remark 1: The cost appears to be the sum of three terms. The first one is related to the a posteriori estimate of the attitude quaternion at time t_1, the second term refers to the new measurement, at time t_2. Each term is weighted according to the respective reliability (evaluated by $\gamma_{1/1}$ and α_2 respectively) of the aposteriori estimate and of the new measurement. The third term is the optimized cost at time t_1. Actually, Eq. (4.53) shows how the estimation problem is sequentially formulated. It will be shown later that the same approach helps in formulating the prediction/filtering problem in the dynamic case.

Remark 2: There are two classic methods for solving the minimization problem Eq. (4.53). The first one is the Lagrange Multiplier method. The second one is by direct substitution of the constraint Eq. (4.6) in the cost function. The second method will be used here. Furthermore, the search for the extremum can be done either by differentiation or by inspection of the cost function. The second technique will be used in the following.

We formulate the cost function $J_{2/2}$ as a sum of squares as follows. First, rewrite Eq. (4.54) as

$$2\; J_{2/2}(\mathbf{q}_2) = \; \gamma_{1/1} \, \| \, \widehat{\mathbf{q}}_{1/1} - \mathbf{q}_2 \|^2 \; + \; \alpha_2 \, \| \overline{\mathbf{q}}_2 - \mathbf{q}_2 \|^2 \; + \; 2 \; \widehat{J}_{1/1} \tag{4.55}$$

Expanding Eq. (4.55) and using the property that $\widehat{\mathbf{q}}_{1/1}$, $\overline{\mathbf{q}}_2$ and $\mathbf{q}_2$ are unit-norm vectors yields

$$2\; J_{2/2}(\mathbf{q}_2) = \; -2 \, (\gamma_{1/1} \, \widehat{\mathbf{q}}_{1/1} + \alpha_2 \, \overline{\mathbf{q}}_2)^T \, \mathbf{q}_2 \; + \; 2 \, (\gamma_{1/1} + \alpha_2) \; + \; 2 \; \widehat{J}_{1/1} \tag{4.56}$$

Let the vector $\mathbf{x}_2$ be defined by

$$\mathbf{x}_2 \overset{\triangle}{=} \gamma_{1/1} \, \widehat{\mathbf{q}}_{1/1} + \alpha_2 \, \overline{\mathbf{q}}_2 \tag{4.57}$$

When multiplying Eq. (4.56) by $\|\mathbf{x}_2\|/\|\mathbf{x}_2\|$, one gets

$$2\; J_{2/2}(\mathbf{q}_2) = \|\mathbf{x}_2\| \; \left[-2 \, \frac{\mathbf{x}_2^T}{\|\mathbf{x}_2\|} \, \mathbf{q}_2 \right] \; + \; 2 \, (\gamma_{1/1} + \alpha_2) \; + \; 2 \; \widehat{J}_{1/1} \tag{4.58}$$

Then, adding and subtracting 1 inside of the brackets, and noting that the vector $\mathbf{x}_2/\|\mathbf{x}_2\|$ has a unit norm, one obtains

$$2\; J_{2/2}(\mathbf{q}_2) = \|\mathbf{x}_2\| \; \left[\underbrace{\left\| \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \right\|^2}_{=1} \; - 2 \, \frac{\mathbf{x}_2^T}{\|\mathbf{x}_2\|} \, \mathbf{q}_2 \right] \; - \; \|\mathbf{x}_2\| \; + \; 2 \, (\gamma_{1/1} + \alpha_2) \; + \; 2 \; \widehat{J}_{1/1} \tag{4.59}$$

The expression in the brackets can be completed to a complete square. This yields :

$$2\, J_{2/2}(\mathbf{q}_2) = \|\mathbf{x}_2\| \, \left\| \left(\frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \right) \; - \; \mathbf{q}_2 \right\|^2 \; + \; 2\,(\gamma_{1/1} + \alpha_2 - \|\mathbf{x}_2\|) \; + \; 2\, \widehat{J}_{1/1} \tag{4.60}$$

Finally, defining the positive coefficient

$$\gamma_{2/2} \overset{\triangle}{=} \|\mathbf{x}_2\| \tag{4.61}$$

and dividing Eq. (4.60) by 2 on both sides, the cost function $J_{2/2}$ becomes

$$J_{2/2}(\mathbf{q}_2) = \frac{1}{2}\, \gamma_{2/2} \, \left\| \left(\frac{\mathbf{x}_2}{\gamma_{2/2}} \right) \; - \; \mathbf{q}_2 \right\|^2 \; + \; (\gamma_{1/1} + \alpha_2 - \gamma_{2/2}) \; + \; \widehat{J}_{1/1} \tag{4.62}$$

Only the first term of the cost $J_{2/2}$ in Eq. (4.62) is a function of $\mathbf{q}_2$, hence $\mathbf{q}_2$ that minimizes $J_{2/2}$ is the same $\mathbf{q}_2$ that minimizes the following cost function

$$\min_{\mathbf{q}_2^T \mathbf{q}_2 = 1} \; \{ \; \overline{J}_{2/2}(\mathbf{q}_2) = \frac{1}{2}\, \gamma_{2/2} \, \left\| \left(\frac{\mathbf{x}_2}{\gamma_{2/2}} \right) - \mathbf{q}_2 \right\|^2 \; \} \tag{4.63}$$

Since the vector $\mathbf{x}_2/\gamma_{2/2}$ minimizes $\overline{J}_{2/2}$ and satisfies the normality constraint, it is the optimal solution and will be noted by $\widehat{\mathbf{q}}_{2/2}$; that is,

$$\widehat{\mathbf{q}}_{2/2} = \frac{\mathbf{x}_2}{\gamma_{2/2}} \tag{4.64}$$

As a conclusion, the cost function $J_{2/2}(\mathbf{q}_2)$ has been expressed in the form

$$J^{\star}_{2/2}(\mathbf{q}_2) = \frac{1}{2}\, \gamma_{2/2} \, \| \, \widehat{\mathbf{q}}_{2/2} - \mathbf{q}_2 \|^2 \; + \; \widehat{J}_{2/2} \tag{4.65}$$

where

$$\widehat{J}_{2/2} = \; \widehat{J}_{1/1} + \alpha_2 - (\gamma_{2/2} - \gamma_{1/1}) \tag{4.66}$$

The estimate $\widehat{\mathbf{q}}_{2/2}$ is the optimal filtered estimate of $\mathbf{q}_2$ at time t_2. Note that Eq. (4.65) is similar to that obtained in Eq. (4.49) at time t_1. Using Eq. (4.64), the estimation problem can then be formulated as

$$\min_{\mathbf{q}_2^T \, \mathbf{q}_2 = 1} \; \{ \; \frac{1}{2}\, \gamma_{2/2} \, \| \, \widehat{\mathbf{q}}_{2/2} - \mathbf{q}_2 \|^2 \; \} \tag{4.67}$$

A similar result was obtained at time t_1 [see Eq. (4.51)]. Based on the preceding, the development, that is presented from Eq. (4.49) to (4.67), can be done in order to demonstrate further the recursive formulation of the estimation problem. The coefficient $\gamma_{2/2}$ appears to be the weighing coefficient of $\widehat{\mathbf{q}}_{2/2}$ in the filtering problem at time t_3.

Comparison between method 1 *and* method 2 *at this stage:*
Equations (4.42),(for $N = 1$), (4.61), and (4.64) show that the filtered estimates at time t_2 are identical in both methods and that $\gamma_{2/2} = \beta_{2/2}$. Moreover, Eq. (4.66) shows that the minimal costs are identical.

Filtering Problem at time t_{N+1} Assume that the a posteriori estimate of the attitude quaternion, $\widehat{\mathbf{q}}_{N/N}$, and its weighing coefficient $\gamma_{N/N}$ were computed at time t_N, and that they are identical to the results of the LM-method. Furthermore, assume that the cost function at time t_N was expressed as

$$J^{\star}_{N/N}(\mathbf{q}_N) = \frac{1}{2}\,\gamma_{N/N}\,\|\,\widehat{\mathbf{q}}_{N/N} - \mathbf{q}_N\|^2 \;+\; \widehat{J}_{N/N} \tag{4.68}$$

At time t_{N+1}, $\overline{\mathbf{q}}_{N+1}$ and α_{N+1} are given. Defining the new cost function by (see Eq. 4.52)

$$J_{N+1/N+1}(\mathbf{q}_{N+1}) = J^{\star}_{N/N}(\mathbf{q}_{N+1}) \;+\; \frac{1}{2}\,\alpha_{N+1}\,\|\overline{\mathbf{q}}_{N+1} - \mathbf{q}_{N+1}\|^2 \tag{4.69}$$

the filtering problem, at time t_{N+1}, is

$$\min_{\mathbf{q}_2^T\mathbf{q}_2=1} \{\; J_{N+1/N+1}(\mathbf{q}_{N+1}) = J^{\star}_{N/N}(\mathbf{q}_{N+1}) \;+\; \frac{1}{2}\,\alpha_{N+1}\,\|\overline{\mathbf{q}}_{N+1} - \mathbf{q}_{N+1}\|^2 \;\} \tag{4.70}$$

Since the formulation of the estimation problem and the expressions of the cost functions are similar at time t_{N+1} and at time t_2, the same computation steps, as performed in Eqs.(4.55) through (4.65), are applied, which lead to the following problem formulation, at time t_{N+1}

$$\min_{\widehat{\mathbf{q}}_{N+1}^T\widehat{\mathbf{q}}_{N+1}=1} \{\; J_{N+1/N+1}(\mathbf{q}_{N+1}) = \frac{1}{2}\,\gamma_{N+1/N+1}\,\left\|\frac{\mathbf{x}_{N+1}}{\gamma_{N+1/N+1}} - \mathbf{q}_{N+1}\right\|^2 \;+\; \widehat{J}_{N+1/N+1}\;\} \tag{4.71}$$

where

$$\mathbf{x}_{N+1} \overset{\triangle}{=} \gamma_{N/N}\,\widehat{\mathbf{q}}_{N/N} \;+\; \alpha_{N+1}\,\overline{\mathbf{q}}_{N+1} \tag{4.72a}$$

$$\gamma_{N+1/N+1} \overset{\triangle}{=} \|\mathbf{x}_{N+1}\| \tag{4.72b}$$

$$\widehat{J}_{N+1/N+1} = \widehat{J}_{N/N} + \alpha_{N+1} - (\gamma_{N+1/N+1} - \gamma_{N/N}) \tag{4.72c}$$

Thus, $\widehat{\mathbf{q}}_{N+1/N+1}$, the a posteriori estimate at time t_{N+1}, is given by

$$\widehat{\mathbf{q}}_{N+1/N+1} = \frac{\mathbf{x}_{N+1}}{\gamma_{N+1/N+1}} \tag{4.73}$$

and, finally, the minimal cost function is rewritten as

$$J^{\star}_{N+1/N+1}(\mathbf{q}_{N+1}) = \frac{1}{2}\,\gamma_{N+1/N+1}\,\|\,\widehat{\mathbf{q}}_{N+1/N+1} - \mathbf{q}_{N+1}\|^2 \;+\; \widehat{J}_{N+1/N+1} \tag{4.74}$$

Comparing Eq. (4.74) with Eq. (4.68), it is seen that the recursion is achieved.

Comparison between the LM-method and the DP-method at this stage:
Considering Eqs.(4.42),and Eqs.(4.72a), (4.72b), (4.72c), it is seen that the algorithms derived from both methods are identical.

Conclusion It was proved that the set of recursive relations, (Eqs. 4.72a,4.72b,4.72c), is the optimal solution of the recursive estimation problem, (Eqs. 4.70,4.68), at time t_1 and at time t_{N+1}. Therefore, the previous relations represent the optimal solution at any time t_k, $k = 1, 2, \ldots$ Both methods, the LM-method and the DP-method yield the same estimation algorithm.

4.2.4 Weighting Coefficients $\alpha_{_k}$

As a matter of fact, the weighting coefficients are the degrees of freedom of any weighted least squares estimator. Since the derivation of the estimator does not assume any particular values for the weights, they can be arbitrarily chosen. However, their choice has a strong influence on the convergence rate of the estimation process. if the quaternion measurement, $\overline{\mathbf{q}}$ is constructed using the **q**-method, then one may choose $\alpha_{_k}$ equal to the maximal eigenvalue of the associated K-matrix. The reason for this choice is as follows. We know, from the **q**-method, that the quaternion measurement $\overline{\mathbf{q}}_{_k}$ is the eigenvector of a K-matrix associated with the largest eigenvalue λ_{max_k}. Furthermore, we know that λ_{max_k} is very close to 1. For a given angular variance $\sigma^2_{_k}$ in the vector measurements at $t_{_k}$ it is easy to show that (see [17])

$$\lambda_{max_k} \;=\; 1 - \mathcal{O}\left(\sigma^2_{_i}\right)$$

So, the more accurate the vector measurements, the smaller $\sigma_{_k}$, and the closer λ_{max_k} from 1. Therefore, we can interpret λ_{max_k} as a reliability factor of the computed quaternion $\overline{\mathbf{q}}_{_k}$. A first possibility is then to define

$$\alpha_{_k} \overset{\triangle}{=} \lambda_{max_k}$$

Another choice is to use the level of noise of the vector measurements in a direct manner. For a given batch of n vector observations at time t_k, the corresponding coefficient α_k is defined as

$$\alpha_k \triangleq \sum_{i=1}^{n} \frac{1}{\sigma_i^2[k]} \tag{4.75}$$

where $\sigma_i[k]$ denotes the standard deviation of the i^{th} vector observation at time t_k. This choice is justified as follows. The reliability of the measurements $\overline{\mathbf{q}}_k$ increases with the size of the batch of observations n and decreases with the variance of the given observations.

4.2.5 Convergence Analysis

Introduction

It is important to analyze the convergence of the recursive algorithm given in Eqs.(4.72a),(4.72b), and (4.72c). The recursive algorithm is expressed by

$$\gamma_{k+1/k+1} = \| \gamma_{k/k}\, \widehat{\mathbf{q}}_{k/k} + \alpha_{k+1}\, \overline{\mathbf{q}}_{k+1} \| \tag{4.76a}$$

$$\widehat{\mathbf{q}}_{k+1/k+1} = \Psi\left(\widehat{\mathbf{q}}_{k/k}, \overline{\mathbf{q}}_{k+1} \right) \tag{4.76b}$$

where the vector function $\Psi\left(\widehat{\mathbf{q}}_{k/k}, \overline{\mathbf{q}}_{k+1} \right)$ is defined by

$$\Psi\left(\widehat{\mathbf{q}}_{k/k}, \overline{\mathbf{q}}_{k+1} \right) \triangleq \frac{\gamma_{k/k}\, \widehat{\mathbf{q}}_{k/k} + \alpha_{k+1}\, \overline{\mathbf{q}}_{k+1}}{\gamma_{k+1/k+1}} \tag{4.77}$$

We use here two types of errors, namely the a posteriori estimation error at t_k, denoted by $\widehat{\mathbf{q}}_{k/k}$, and the measurement error at t_{k+1}, denoted by $\mathbf{v}_{k+1}$, which are defined as

$$\widehat{\mathbf{q}}_{k/k} = \mathbf{q} + \tilde{\mathbf{q}}_{k/k} \tag{4.78a}$$

$$\overline{\mathbf{q}}_{k+1} = \mathbf{q} + \mathbf{v}_{k+1} \tag{4.78b}$$

Development of Convergence Conditions

Coefficient $\gamma_{k+1/k+1}$ Substituting Eqs.(4.78) into Eq. (4.76) yields

$$\gamma_{k+1/k+1} = \left(\gamma_{k/k} + \alpha_{k+1} \right) \|\mathbf{q} + \mathbf{u}\| \tag{4.79}$$

where the 4×1 vector $\mathbf{u}$ is defined as

$$\mathbf{u} \triangleq \frac{\gamma_{k/k}\, \tilde{\mathbf{q}}_{k/k} + \alpha_{k+1}\, \mathbf{v}_{k+1}}{\left(\gamma_{k/k} + \alpha_{k+1} \right)} \tag{4.80}$$

Using the unit-norm property of $\mathbf{q}$, Eq. (4.79) is expressed by

$$\gamma_{k+1/k+1} = \left(\gamma_{k/k} + \alpha_{k+1}\right) \left(1 + 2\,\mathbf{q}^T\mathbf{u} + \mathbf{u}^T\mathbf{u}\right)^{\frac{1}{2}} \tag{4.81}$$

Taking the inverse of Eq. (4.79) we have

$$\gamma^{-1}_{k+1/k+1} = \left(\gamma_{k/k} + \alpha_{k+1}\right)^{-1} \left(1 + 2\,\mathbf{q}^T\,\mathbf{u} + \mathbf{u}^T\,\mathbf{u}\right)^{-\frac{1}{2}} \tag{4.82}$$

Defining the variable δ as

$$\delta \overset{\triangle}{=} 2\,\mathbf{q}^T\mathbf{u} + \mathbf{u}^T\mathbf{u} \tag{4.83}$$

and using the Taylor series expansion of $(1+\delta)^m$ around 0 to second order, that is, $(1+\delta)^m \simeq 1 + \,m\,\delta + \frac{1}{2}\,m\,(m-1)\,\delta^2$ yields

$$(1+\delta)^{-\frac{1}{2}} \simeq 1 - \frac{1}{2}\,\delta + \frac{3}{8}\,\delta^2 \tag{4.84}$$

Substituting the right-hand side of Eq. (4.84) in Eq. (4.82) and retaining the terms of second order in $\mathbf{u}$ yields

$$\gamma^{-1}_{k+1/k+1} \simeq \left(\gamma_{k/k} + \alpha_{k+1}\right)^{-1} \left(1 - \mathbf{q}^T\,\mathbf{u} - \frac{1}{2}\,\mathbf{u}^T\,\mathbf{u} + \frac{3}{2}\left(\mathbf{q}^T\mathbf{u}\right)^2\right) \tag{4.85}$$

Taylor Series Expansion of Ψ around $(\mathbf{q},\mathbf{q})$ to Second Order in $\mathbf{u}$ Substituting Eqs.(4.78) into Eq. (4.77) gives

$$\Psi\left(\mathbf{q} + \tilde{\mathbf{q}}_{k/k}\,,\,\mathbf{q} + \mathbf{v}_{k+1}\right) = \gamma^{-1}_{k+1/k+1}\left(\gamma_{k/k} + \alpha_{k+1}\right)\left(\mathbf{q} + \mathbf{u}\right) \tag{4.86}$$

Substituting Eq. (4.85) into Eq. (4.86) and arranging terms yields

$$\begin{aligned}\Psi\left(\mathbf{q} + \tilde{\mathbf{q}}_{k/k}\,,\,\mathbf{q} + \mathbf{v}_{k+1}\right) &\simeq \mathbf{q} + \left(\,I_4 - \mathbf{q}\,\mathbf{q}^T\,\right)\,\mathbf{u} \\ &+ \left[\left((-\frac{1}{2})\mathbf{u}^T\,\mathbf{u} + \frac{3}{2}\left(\mathbf{q}^T\mathbf{u}\right)^2\right)\,I_4 + \left(\mathbf{u}\,\mathbf{u}^T\right)\right]\,\mathbf{q}\end{aligned} \tag{4.87}$$

First Order Approximation When the first order term is not zero, it is the dominant term. So, to first order in $\mathbf{u}$, the estimation error $\tilde{\mathbf{q}}_{k+1/k+1}$ is given by

$$\tilde{\mathbf{q}}_{k+1/k+1} \simeq \left(I_4 - \mathbf{q}\mathbf{q}^T\right)\,\mathbf{u} \tag{4.88}$$

That is, the orthogonal projection of $\mathbf{u}$ on the orthogonal complementary subspace of $\mathbf{q}$ is the approximated estimation error. Denoting by $M\,(\mathbf{q})$ the projection matrix $\left(I_4 - \mathbf{q}\mathbf{q}^T\right)$ and substituting Eq. (4.80) into Eq. (4.88) yields

$$\tilde{\mathbf{q}}_{k+1/k+1} = M(\mathbf{q})\frac{\gamma_{k/k}}{\gamma_{k/k} + \alpha_{k+1}}\tilde{\mathbf{q}}_{k/k} + M(\mathbf{q})\frac{\alpha_{k+1}}{\gamma_{k/k} + \alpha_{k+1}}\mathbf{v}_{k+1} \tag{4.89}$$

It is seen, that, if the recursive process converges, then it is of first order in $\tilde{\mathbf{q}}$. The measurement error $\mathbf{v}$ acts like a forcing function in the discrete dynamics of the error, $\tilde{\mathbf{q}}$. The convergence depends on the eigenvalues of the dynamics matrix, which is the M-matrix multiplied by a scalar.

Convergence Conditions The M-matrix is symmetric and has got the following spectral decomposition

$$M = T \begin{bmatrix} I_3 & \\ & 0 \end{bmatrix} T^T \tag{4.90}$$

where I_3 is the 3×3 identity matrix and T is the orthogonal matrix of the eigenvectors. It is seen, that the dynamics matrix has a triple eigenvalue at $\frac{\gamma_{k/k}}{\gamma_{k/k}+\alpha_{k+1}}$ and a simple one at 0. Since the scalars $\gamma_{k/k}$ and α_{k+1} are positive, these eigenvalues are inside the unit circle, which ensures convergence.

Initial Error $\tilde{\mathbf{q}}_{0/0}$ Response In the case of a sequence of ideal measurements, that is $\mathbf{v} = 0$ for all k, the filter response to an initial error would be

$$\tilde{\mathbf{q}}_{k+1/k+1} = T \begin{bmatrix} I_3 & \\ & 0 \end{bmatrix} T^T \left(\frac{\gamma_{k/k}}{\gamma_{k/k} + \alpha_{k+1}} \right) \cdots \left(\frac{\gamma_1}{\gamma_1 + \alpha_2} \right) \left(\frac{\alpha_0}{\alpha_0 + \alpha_1} \right) \tilde{\mathbf{q}}_0 \tag{4.91}$$

It appears that the diagonal matrix has its fourth entry constantly equal to 0 while the first three entries remain equal to 1. This phenomenon would prevent convergence unless the product of the scalar ratios did not enforce it. Since each of these ratios is smaller than one, the N-terms product tends to 0 when N tends to ∞.

Conclusion

A sufficient condition of convergence for the recursive algorithm is that the weights α_k are positive. The rate of convergence is of the first order in $\tilde{\mathbf{q}}$. From an initial error, without noise in the measurements, the filter converges asymptotically to the true state. In the general case, the measurement noise acts as a forcing function, that generates a non zero, but bounded term, in the sequence $\tilde{\mathbf{q}}$.

4.2.6 Algorithm Summary

1. Initialization Stage:

$$\widehat{\mathbf{q}}_{0/0} = \overline{\mathbf{q}}_0 \tag{4.92a}$$

$$\gamma_{0/0} = \alpha_0 \tag{4.92b}$$

$$\widehat{J}_{0/0} = 0 \tag{4.92c}$$

2. Filtering Stage:

$$\gamma_{k/k} = \| \, \gamma_{k-1/k-1} \, \widehat{\mathbf{q}}_{k-1/k-1} + \alpha_k \overline{\mathbf{q}}_k \, \| \tag{4.93a}$$

$$\widehat{\mathbf{q}}_{k/k} = \frac{\gamma_{k-1/k-1} \, \widehat{\mathbf{q}}_{k-1/k-1} + \alpha_k \overline{\mathbf{q}}_k}{\gamma_{k/k}} \tag{4.93b}$$

$$\widehat{J}_{k/k} = \alpha_k - \left(\gamma_{k/k} - \gamma_{k-1/k-1} \right) + \widehat{J}_{k-1/k-1} \tag{4.93c}$$

4.2.7 Simulation Study

Simulation Description

The simulations of the time invariant-case have two purposes. The first one is to check the validity of the static filter developed previously. The second one is to illustrate how particular choices of the weights α_k are influencing the estimation process. We consider the following model of the simulated vector measurement noise $\boldsymbol{\delta b}_k$,

$$\boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, (\sigma_b)^2 I_3\right)$$

The sequence of unit vector measurements $\mathbf{b}_k$ is computed by adding $\boldsymbol{\delta b}_k$ to the true measurement vector and by normalizing the result. At each sampling time, two simultaneous measurements are made. Using these two measurements, a K-matrix is constructed and its largest eigenvalue, λ_{max_k}, and related eigenvector $\overline{\mathbf{q}}_k$ are computed. The choice of the coefficients α_k is twofold,

$$\begin{array}{lccc} \textit{Simulation } S_1 & \alpha_k & = & \lambda_{max_k} \\ \textit{Simulation } S_2 & \alpha_k & = & \frac{2}{\sigma_b^2} \end{array}$$

In each of the two simulations, S_1 and S_2, sensors failures are simulated. It is assumed then that the damaged sensors still acquires measurement however with a decreased accuracy, which is quantified as follows,

$$\begin{array}{lclcccl} \sigma_{_b} & = & 0.01\,\text{rad} & 0 & \leq & t & \leq & 10 sec. \\ \sigma_{_b} & = & 0.1\,\text{rad} & 10 & \leq & t & \leq & 40 sec. \\ \sigma_{_b} & = & 0.01\,\text{rad} & 40 & \leq & t & \leq & 100 sec. \end{array}$$

These values are typical for Magnetometers and coarse Earth Horizon Sensors. The simulations are run over 100 seconds. The vector measurement sampling time is 2 seconds.

Results Analysis

Figures 4.1 and 4.2 show the results of the two simulations. The time history of the angular estimation error for Simulation S_1 is plotted in Figure 4.1. We observe three phases: first, a convergent phase from $t = 0$ to $t = 10$ sec., then a noisy estimation phase from $t = 10$ to $t = 40$ sec. and finally a slowly convergent phase from $t = 40$ to $t = 100$ sec. During the noisy estimation phase, the angular error has a mean level of about 1 degree and reaches maxima of about 1.6 degrees. This arises from the new incoming observations which have a standard deviation of a few degrees. However, the algorithm succeeds in filtering these observations and reducing the uncertainty down to a level of 1 degree. We realize that the new noisy observations have a relative important weight in the filtered estimate. This arises from the fact that the maximal eigenvalue of the K-matrix is close to one within an order of $\sigma_{_b}^2$. That is, when processing the relative accurate measurements ($\sigma_{_b}$ about one degree), the coefficients $\alpha_{_k}$ are of the order $\left(1 - 10^{-4}\right)$. On the other hand, for the noisy measurements, the $\alpha_{_k}$'s are of the order $1 - 10^{-2}$. We see that these values are very close one to each other, and do not allow the filter to make the difference between the accurate and inaccurate measurements. Therefore the convergence rate in the estimation error is slow. The error level is still 0.6 deg after 100 sec.

The time history of the angular estimation error for Simulation S_2 is shown in Figure 4.2. Similarly to the previous simulation three phases are observed. The first one corresponds to a converging estimation error. During the second one, from $t = 10$ seconds to $t = 40$ seconds, the error stays at a relative constant level. The third phase is a convergent phase down to the steady-state value of about 0.15 degree. We understand that the particular choice of the coefficients $\alpha_{_k}$ had a beneficial influence on the estimation procedure.

This is explained as follows. The weights of the accurate measurements are two orders of magnitude higher than the weights of the coarse measurements. Thus the accurate measurements have an important impact on the filtered estimate. This is depicted by the rapid recovery in the third phase of the run.

4.2.8 Concluding Remarks

Two methods were used in order to develop the static estimator, namely the Lagrange Multipliers method, and a Dynamic Programming method. The performance of the static estimator were demonstrated through sample runs simulations. Sensors failures were simulated that evidenced the importance of an adequate choice for the weights in the least squares cost function.

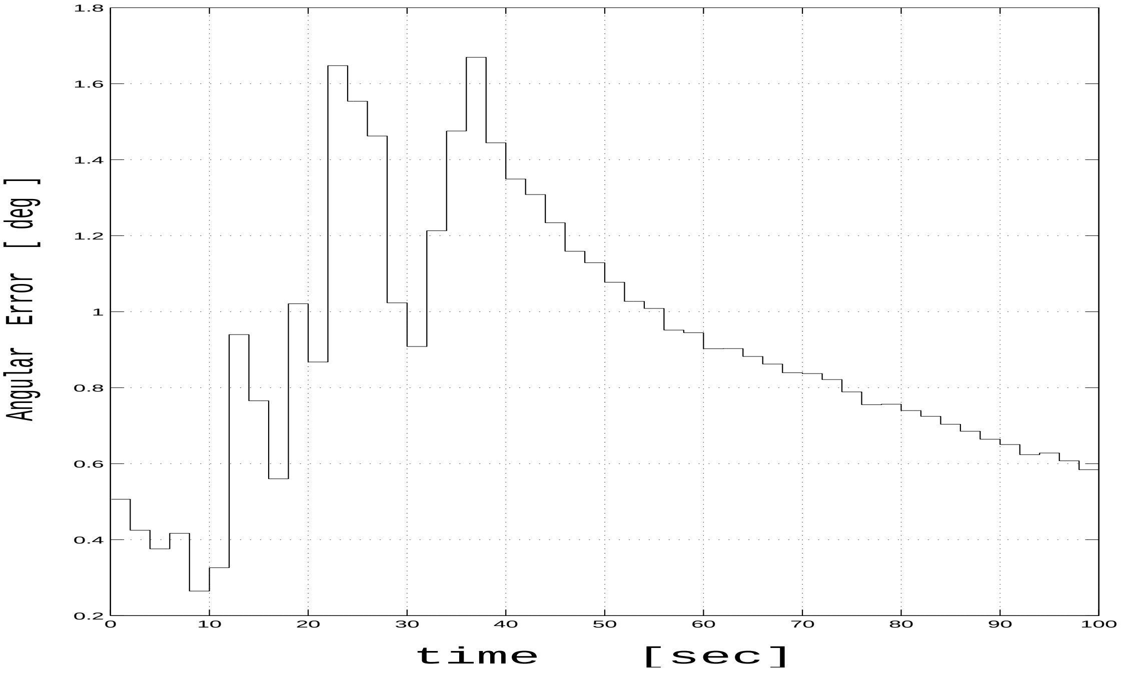


Fig. 4.1: Time-invariant case. Simulation S_1. Angular estimation error for a sample run.

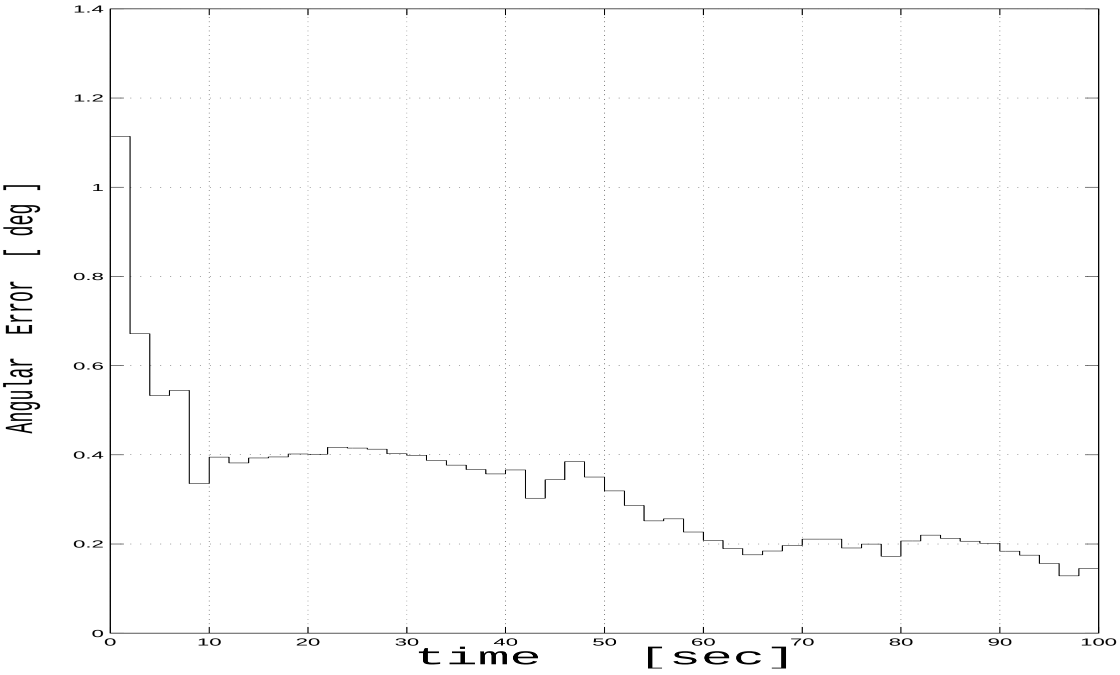


Fig. 4.2: Time-invariant case. Simulation S_2. Angular estimation error for a sample run.

4.3 Time-Varying Case

4.3.1 Model Description

In this section, we consider the quaternion discrete process which is described by the following vector difference equation[2]

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k + \mathbf{w}_k \tag{4.94}$$

for $k = 0, 1, \ldots$, where $\mathbf{q}_k$ is the 4×1 state vector. $\mathbf{q}_k$ is the quaternion of the rotation from the reference frame $\mathcal{R}$ to the body frame $\mathcal{B}$ at time t_k, so it is a unit norm vector. Thus, the state vector process of Eq. (4.94) is subject to the quadratic constraint

$$\mathbf{q}_k^T \mathbf{q}_k = 1 \tag{4.95}$$

for $k = 1, 2, \ldots$. The 4×4 orthogonal matrix Φ_k is computed from gyroscopes measurements of the body angular velocity vector. Using the computed value of Φ_k in the

[2] This model is developed in details in chapter 2

quaternion kinematic equation Eq. (4.94) yields an erroneous propagation of the quaternion process. The process error can be modelled as an additive error, denoted by $\mathbf{w}_k$, and expressed as a function of the gyroscopes outputs error $\boldsymbol{\epsilon}_k$ (see details in chapter 2). Partitioning the estimate of the attitude quaternion at t_k, $\widehat{\mathbf{q}}_k$, using its vector and scalar parts, as follows

$$\widehat{\mathbf{q}}_k = \left\{ \begin{array}{c} \widehat{\mathbf{e}}_k \\ \widehat{q}_k \end{array} \right\} \tag{4.96}$$

and defining the matrix $\widehat{\Xi}_k$ as

$$\widehat{\Xi}_k \triangleq \left[\begin{array}{c} [\widehat{\mathbf{e}}_k \times] + \widehat{q}_k \; I_3 \\ -\widehat{\mathbf{e}}_k^T \end{array} \right] \tag{4.97}$$

then a first order approximation of the process noise $\mathbf{w}_k$ is expressed by

$$\mathbf{w}_k = -\frac{\Delta t}{2} \widehat{\Xi}_k \, \boldsymbol{\epsilon}_k \tag{4.98}$$

where Δt is the incremental time in the discretized quaternion process of Eq. (4.94). In the present work it is assumed that $\{ \boldsymbol{\epsilon}_k \}$ is a random bias sequence and that the components of $\boldsymbol{\epsilon}_k$ are identically distributed with standard deviation σ_ϵ. That is,

$$E[\boldsymbol{\epsilon}_k] = \bar{\boldsymbol{\epsilon}}_k \qquad \operatorname{cov}(\boldsymbol{\epsilon}_k) = \sigma_\epsilon^2 \, I_3 \tag{4.99}$$

Using the model Eq. (4.99), we can write $\boldsymbol{\epsilon}_k$ as the sum of its expectation and of a zero-mean white noise process, denoted by $\mathbf{n}_{\epsilon_k}$, which has the same covariance matrix as $\boldsymbol{\epsilon}_k$.

$$\boldsymbol{\epsilon}_k = \bar{\boldsymbol{\epsilon}}_k + \mathbf{n}_{\epsilon_k} \tag{4.100}$$

Using Eq. (4.100) into Eq. (4.98) yields

$$\mathbf{w}_k = -\frac{\Delta t}{2} \widehat{\Xi}_k \left(\bar{\boldsymbol{\epsilon}}_k + \mathbf{n}_{\epsilon_k} \right) \tag{4.101}$$

$$= -\frac{\Delta t}{2} \widehat{\Xi}_k \, \bar{\boldsymbol{\epsilon}}_k - \frac{\Delta t}{2} \widehat{\Xi}_k \, \mathbf{n}_{\epsilon_k} \tag{4.102}$$

The expectation of $\mathbf{w}_k$ is obtained by taking the expectation of both sides in Eq. (4.101), and by using the facts that the variable $\widehat{\Xi}_k$ is deterministic and that the process $\mathbf{n}_{\epsilon_k}$ is zero-mean.

$$\bar{\mathbf{w}}_k \triangleq E[\mathbf{w}_k] = -\frac{\Delta t}{2} \widehat{\Xi}_k \, \bar{\boldsymbol{\epsilon}}_k \tag{4.103}$$

Then, defining the process $\mathbf{n}_{w_k}$ as

$$\mathbf{n}_{w_k} \triangleq \frac{\Delta t}{2}\, \widehat{\Xi}_k\, \mathbf{n}_{\epsilon_k} \tag{4.104}$$

we can rewrite Eq. (4.101) as follows

$$\mathbf{w}_k = \bar{\mathbf{w}}_k - \mathbf{n}_{w_k} \tag{4.105}$$

Finally, Eq. (4.105) can be written equivalently

$$\bar{\mathbf{w}}_k = \mathbf{w}_k + \mathbf{n}_{w_k} \tag{4.106}$$

which is the usual way of writing a measurement equation of the variable $\mathbf{w}_k$. To summarize, we assume that the expected value of $\mathbf{w}_k$, $\bar{\mathbf{w}}_k$, can be computed using Eq. (4.103) and we consider this a priori value a “measurement” of the variable $\mathbf{w}_k$. This approach is well known and is used in the development of the Maximum A Posteriori Estimator (see [40, p. 192]). In the context of a non-stochastic least squares estimator, the measurement equation Eq. (4.106), also called data equation in [56, p. 116], is used for defining the corresponding measurement residuals. We also consider the quaternion measurement equation introduced in the preceding section.

$$\bar{\mathbf{q}}_k = \mathbf{q}_k + \mathbf{v}_k \tag{4.107}$$

where $\bar{\mathbf{q}}_k$ is the single-frame estimate of $\mathbf{q}_k$ at t_k and $\mathbf{v}_k$ denotes the associated error.

Model Summary

We consider the linear time varying system that is described by the process equation

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k + \mathbf{w}_k \tag{4.108}$$

and by the observation equations

$$\begin{aligned} \overline{\mathbf{q}}_k &= \mathbf{q}_k + \mathbf{v}_k \\ \bar{\mathbf{w}}_k &= \mathbf{w}_k + \mathbf{n}_{w_k} \end{aligned} \tag{4.109}$$

where $\mathbf{v}_k$, and $\mathbf{n}_{w_k}$ are assumed to be zero-mean white noise sequences, uncorrelated with one another and with the initial quaternion $\mathbf{q}_0$.

4.3.2 Estimation Problem

The estimation problem consists of determining the sequences $\{\mathbf{q}_N\}$ and $\{\mathbf{w}_N\}$, $N = 1, 2, \ldots$, given the sequences of measurements $\{\overline{\mathbf{q}}_N\}$ and $\{\bar{\mathbf{w}}_N\}$, $N = 1, 2, \ldots$, subject to the dynamic constraint $\mathbf{q}_{N+1} = \Phi_N \mathbf{q}_N + \mathbf{w}_N$, and the quadratic constraint $\mathbf{q}_N^T \mathbf{q}_N = 1$. Since the statistical properties of the noise sequences $\{\mathbf{w}_k\}$, $\{\mathbf{v}_k\}$ and $\{\mathbf{n}_{w_k}\}$ are not exactly known, a least squares estimation approach is taken. Based on the measurement equations Eqs. (4.109), a cost function is designed as follows

$$2\ J_{N/N}\left(\mathbf{q}_0, \ldots, \mathbf{q}_N, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\right) = \alpha_0 \|\mathbf{q}_0 - \overline{\mathbf{q}}_0\|^2 + \sum_{k=1}^{N} \alpha_k \|\mathbf{q}_k - \overline{\mathbf{q}}_k\|^2 + \eta_{k-1} \|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 \tag{4.110}$$

where α_0, α_k and η_{k-1}, $k = 1, 2, \ldots, N$, are prescribed positive real scalars. They determine how close the estimated values of $\mathbf{q}_k$ and $\mathbf{w}_{k-1}$ will be from the corresponding measurement at time t_k, $\overline{\mathbf{q}}_k$ and $\bar{\mathbf{w}}_{k-1}$. They are the reliability factors of the measurements $\overline{\mathbf{q}}_k$ and $\bar{\mathbf{w}}_{k-1}$. The specific estimation problem of interest here is the filtering problem of $\mathbf{q}_N$, given the sequences $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_N\}$ and $\{\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{N-1}\}$. For this purpose, the cost function $J_{N/N}$ will be expressed as a function of $\mathbf{q}_N$ only and of the sequence $\{\mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\}$ as follows. The dynamic constraints of Eq. (4.94) are rewritten :

$$\mathbf{q}_k = \Phi_k^T \left(\mathbf{q}_{k+1} - \mathbf{w}_k\right) \tag{4.111}$$

$k = 1, 2, ...$, and substituted in Eq. (4.110). Thus, $J_{N/N}$ contains only terms in $\mathbf{q}_N$ and in the $\mathbf{w}_k$'s. The particular value $\{\mathbf{q}_N, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\}$ for which the cost function is minimum, subject to the constraint Eq. (4.95), is the estimate of $\{\mathbf{q}_N, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\}$, given the observations $\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_N$, $\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{N-1}$. Note that the estimates of $\mathbf{q}_N$ and $\mathbf{w}_{N-1}$ are filtered estimates, while the estimates of $\mathbf{w}_0, \ldots, \mathbf{w}_{N-2}$ are smoothed estimates.

A usual approach for solving the constrained minimization of $J_{N/N}$ is the variational approach of the Lagrange Multipliers method. When applying this method, the optimal value of $\{\mathbf{q}_N, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\}$ is obtained by solving a set of non-linear equations in $\{\mathbf{q}_N, \mathbf{w}_0, \ldots, \mathbf{w}_{N-1}\}$. Considering the optimal value of $\mathbf{q}_N$, it can be shown that it is a function of the sequences $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{N-1}\}$ and $\{\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{N-1}\}$. So, the estimated value of $\mathbf{q}_N$, using this method, is a batch estimate. However, as stated and shown by Cox in [55], "the possibility arises of proceeding sequentially via a dynamic programming formulation in the filtering problem". Cox considered dynamic models, which are constituted of non linear process and measurement equations, with additive white Gaussian noises. In the following, we apply the method suggested by Cox to the case of a linear system, and adapt it in order to include the quadratic constraint $\mathbf{q}_N^T\mathbf{q}_N = 1$.

4.3.3 Dynamic Programming Approach

The functional equation technique of dynamic programming will be applied as follows. We consider the sequence of minimization problems of the cost function $J_{N/N}$, defined in Eq. (4.110), over the sequence $\{\mathbf{q}_k\}$ and the sequence $\{\mathbf{w}_k\}$, subject to the constraints of Eq. (4.111) and Eq. (4.95). We introduce the sequence of functions $\{F_N(\mathbf{q}_N)\}$, where $\mathbf{q}_N$ is unit norm and $N = 1, 2, \ldots$, defined by the relations

$$F_0(\mathbf{q}_0) \triangleq \frac{1}{2}\alpha_0\|\mathbf{q}_0 - \overline{\mathbf{q}}_0\|^2 \tag{4.112}$$

$$F_N(\mathbf{q}_N) \triangleq \min_{\mathbf{w}_0,\ldots,\mathbf{w}_{N-1}} \left\{ J_{N/N} \right\} \tag{4.113}$$

The minimization in Eq. (4.113) is subject to the constraints Eq. (4.111) and Eq. (4.95). To derive a relation between $F_N(\mathbf{q}_N)$ and $F_{N-1}(\mathbf{q}_{N-1})$ for $N = 1, 2, \ldots$ we proceed as follows

$$F_N(\mathbf{q}_N) =$$
$$\min_{\mathbf{w}_{N-1}} \left\{ \min_{\mathbf{w}_0,\ldots,\mathbf{w}_{N-2}} \left\{ \frac{1}{2}\alpha_0\|\mathbf{q}_0 - \overline{\mathbf{q}}_0\|^2 + \frac{1}{2}\sum_{k=1}^{N-1} \eta_{k-1}\|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 + \alpha_k\|\mathbf{q}_k - \overline{\mathbf{q}}_k\|^2 \right\} \right.$$
$$\left. + \frac{1}{2}\eta_{N-1}\|\mathbf{w}_{N-1} - \bar{\mathbf{w}}_{N-1}\|^2 \right\} + \frac{1}{2}\alpha_N\|\mathbf{q}_N - \overline{\mathbf{q}}_N\|^2 \tag{4.114}$$

Using the definition of $F_{N-1}(\mathbf{q}_{N-1})$ (Eq. 4.113) yields

$$F_N(\mathbf{q}_N) = \tag{4.115}$$
$$\min_{\mathbf{w}_{N-1}} \left\{ F_{N-1}(\mathbf{q}_{N-1}) + \frac{1}{2}\eta_{N-1}\|\mathbf{w}_{N-1} - \bar{\mathbf{w}}_{N-1}\|^2 \right\} + \frac{1}{2}\alpha_N\|\mathbf{q}_N - \overline{\mathbf{q}}_N\|^2$$

where the minimization over $\mathbf{w}_{N-1}$ is subject to the constraint of Eq. (4.111). Equation (4.115) is the sought recursive relation between $F_N(\mathbf{q}_N)$ and $F_{N-1}(\mathbf{q}_{N-1})$. The constrained minimization problem in Eq. (4.114) can be solved either by the Lagrange

Multipliers method, or by substituting the constraint in the cost function. The second method will be used here. Hence, writing the dynamic constraint at t_N,

$$\mathbf{q}_{N-1} = \Phi_{N-1}^T \left(\mathbf{q}_N - \mathbf{w}_{N-1}\right) \tag{4.116}$$

and substituting the right hand side of Eq. (4.116) to $\mathbf{q}_{N-1}$ in Eq. (4.115) yields the functional equation

$$\begin{aligned} &F_N\left(\mathbf{q}_N\right) = \\ &\min_{\mathbf{w}_{N-1}} \left\{ F_{N-1}\left(\Phi_{N-1}^T\left(\mathbf{q}_N - \mathbf{w}_{N-1}\right)\right) + \frac{1}{2}\,\eta_{N-1}\|\,\mathbf{w}_{N-1} - \bar{\mathbf{w}}_{N-1}\|^2\right\} + \frac{1}{2}\,\alpha_N\|\mathbf{q}_N - \overline{\mathbf{q}}_N\|^2 \end{aligned} \tag{4.117}$$

where $\mathbf{q}_N$ satisfies the constraint of Eq. (4.95). Eq. (4.117) is the functional equation that has to be solved.

Remark 1: The minimization of the function $F_0\left(\mathbf{q}_0\right)$ over $\mathbf{q}_0$, subject to the unit norm constraint, provides the filtered estimate of $\mathbf{q}_0$ at t_0, denoted by $\widehat{\mathbf{q}}_{0/0}$. It can be seen from Eq. (4.113) that $\widehat{\mathbf{q}}_{0/0}$ is simply equal to the measurement $\overline{\mathbf{q}}_0$.

Remark 2: At each step, the minimization of the function $F_N\left(\mathbf{q}_N\right)$ over $\mathbf{q}_N$, subject to the unit norm constraint, provides a filtered estimate of $\mathbf{q}_N$, denoted by $\widehat{\mathbf{q}}_{N/N}$, given the sequence of measurements $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{N-1}\}$ and $\{\,\bar{\mathbf{w}}_0, \ldots,\ \bar{\mathbf{w}}_{N-1}\}$. So, Eq. (4.117) shows how the filtering problem is sequentially formulated : assuming that the function $F_{N-1}\left(\mathbf{q}_{N-1}\right)$ has been computed, we can compute the function $F_N\left(\mathbf{q}_N\right)$ by solving the functional equation (4.117). Then, this function $F_N\left(\mathbf{q}_N\right)$ is minimized to compute $\widehat{\mathbf{q}}_{N/N}$ and so on.

Remark 3: However, it is possible to decompose each filtering step into two steps, namely a predicting step and an update step, as is commonly done in sequential estimation algorithms of time varying parameters. We define the sequence of functions $G_N\left(\mathbf{q}_N\right)$, $N = 1, 2, \ldots$

$$G_N\left(\mathbf{q}_N\right) \triangleq \min_{\mathbf{w}_{N-1}} \left\{ F_{N-1}\left(\Phi_{N-1}^T\left(\mathbf{q}_N - \mathbf{w}_{N-1}\right)\right) + \frac{1}{2}\,\eta_{N-1}\|\,\mathbf{w}_{N-1} - \bar{\mathbf{w}}_{N-1}\|^2\right\} \tag{4.118}$$

For any particular value of $\mathbf{q}_N$, the number $G_N\left(\mathbf{q}_N\right)$ may be interpreted as the cost of the estimated sequence $\{\mathbf{q}_0, \ldots, \mathbf{q}_N\}$ in which $\mathbf{q}_N$ takes that particular value, given the sequence of observations $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{N-1}\}$ and $\{\,\bar{\mathbf{w}}_0, \ldots,\ \bar{\mathbf{w}}_{N-1}\}$. Choosing as an estimate of $\mathbf{q}_N$, the value of $\mathbf{q}_N$ for which the function $G_N\left(\mathbf{q}_N\right)$ is minimum corresponds to predicting the $\mathbf{q}_N$, denoted by $\widehat{\mathbf{q}}_{N/N-1}$ that will achieve the minimal cost, given the observations.

Substituting $G_N(\mathbf{q}_N)$ to its expression in Eq. (4.117), yields

$$F_N(\mathbf{q}_N) = G_N(\mathbf{q}_N) + \frac{1}{2}\,\alpha_N \|\mathbf{q}_N - \overline{\mathbf{q}}_N\|^2 \tag{4.119}$$

Since the minimization of $G_N(\mathbf{q}_N)$ over $\mathbf{q}_N$, subject to the unit norm constraint , gives an a priori estimate of $\mathbf{q}_N$ at t_N, it is seen from Eq. (4.119) that, minimizing $F_N(\mathbf{q}_N)$ over $\mathbf{q}_N$, corresponds to finding the best combination between the a priori estimate of $\mathbf{q}_N$ and the new observation, $\overline{\mathbf{q}}_N$, according to its reliability factor α_N.

Remark 4: It was shown, so far, how the predicting/filtering problem can be sequentially formulated and solved, but we do not know explicitly the expressions for the predicted/filtered estimates. Actually, in the prediction step, we wish to compute the a priori estimate $\widehat{\mathbf{q}}_{N/N-1}$ as a function of $\widehat{\mathbf{q}}_{N-1/N-1}$, the a posteriori estimate of $\mathbf{q}_{N-1}$ at t_{N-1}, and of $\bar{\mathbf{w}}_{N-1}$ only. For this purpose, the function $F_{N-1}\left(\mathbf{q}_{N-1}\right)$ will be expressed using the estimate $\widehat{\mathbf{q}}_{N-1/N-1}$ before computing $G_N(\mathbf{q}_N)$ from Eq. (4.118). We will detail this manipulation when developing the algorithm, in the next section.

The DP-method for the time-varying case is summarized next.

1. The initialization step consists in minimizing $F_0(\mathbf{q}_0)$ with respect to the vector $\mathbf{q}_0$ over the unit sphere. This provides the initial filtered estimate of $\mathbf{q}_0$, $\widehat{\mathbf{q}}_{0/0}$. Then, $F_0(\mathbf{q}_0)$ is expressed using the estimate $\widehat{\mathbf{q}}_{0/0}$.

2. The quaternion predicting step at time t_N is as follows. Given the function $F_{N-1}\left(\mathbf{q}_{N-1}\right)$, the function $G_N(\mathbf{q}_N)$ is computed by solving Eq. (4.118). The predicted estimate of $\mathbf{q}_N$, $\widehat{\mathbf{q}}_{N/N-1}$ given the sequences $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{N-1}\}$ and $\{\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{N-1}\}$, is computed by minimizing $G_N(\mathbf{q}_N)$ over the unit norm vector $\mathbf{q}_N$. The minimal cost $G_N(\mathbf{q}_N)$ is then expressed using the estimate $\widehat{\mathbf{q}}_{N/N-1}$.

3. The quaternion filtering step at time t_N is as follows. The function $F_N(\mathbf{q}_N)$ is computed from Eq. (4.119). The filtered estimate of $\mathbf{q}_N$, $\widehat{\mathbf{q}}_{N/N}$, is computed by minimizing $F_N(\mathbf{q}_N)$ with respect to the vector $\mathbf{q}_N$ over the unit sphere.

4.3.4 Algorithm Derivation

Remark about the notation: A particular notation of the estimated variables is used here. The estimates of $\mathbf{q}_k$ and $\mathbf{w}_k$ are denoted by $\widehat{\mathbf{q}}_{k/i}$ and $\widehat{\mathbf{w}}_{k/i}$, where the index i

only refers to the *quaternion* measurement sequence. For instance, $\widehat{\mathbf{w}}_{k/k-1}$ represents an estimate of $\mathbf{w}_k$ given $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{k-1}\}$.

Initialization Step at t_0

It is assumed that, at t_0, the quaternion measurement $\overline{\mathbf{q}}_0$ is performed and its reliability factor α_0 is given. Let us rewrite the function $F_0(\mathbf{q}_0)$, as defined from Eq. (4.113).

$$F_0(\mathbf{q}_0) \overset{\triangle}{=} \frac{1}{2}\alpha_0 \|\mathbf{q}_0 - \overline{\mathbf{q}}_0\|^2 \tag{4.120}$$

The filtering problem is to minimize $F_0(\mathbf{q}_0)$ over the unitary vector $\mathbf{q}_0$. It is seen, from Eq. (4.119), that choosing the unitary vector $\overline{\mathbf{q}}_0$ as the estimate of $\mathbf{q}_0$ satisfies the unit norm constraint and gives to the cost function F_0 the value 0. Therefore, $\overline{\mathbf{q}}_0$ is the sought estimate. Defining

$$\widehat{\mathbf{q}}_{0/0} \overset{\triangle}{=} \overline{\mathbf{q}}_0 \tag{4.121}$$

$$\gamma_{0/0} \overset{\triangle}{=} \alpha_0 \tag{4.122}$$

$$\widehat{F}_0 \overset{\triangle}{=} F_0\left(\widehat{\mathbf{q}}_{0/0}\right) \tag{4.123}$$

and using Eqs. (4.121) through (4.123) in Eq. (4.120) yields

$$F_0(\mathbf{q}_0) = \frac{1}{2}\gamma_{0/0}\|\mathbf{q}_0 - \widehat{\mathbf{q}}_{0/0}\|^2 + \widehat{F}_0 \tag{4.124}$$

In order to distinguish the new expression of the function $F_0(\mathbf{q}_0)$, as given in Eq. (4.124), from the former expression of $F_0(\mathbf{q}_0)$, as given in Eq. (4.120), we will define the function $F_0^\star(\mathbf{q}_0)$ by

$$F_0^\star(\mathbf{q}_0) \overset{\triangle}{=} \frac{1}{2}\gamma_{0/0}\|\mathbf{q}_0 - \widehat{\mathbf{q}}_{0/0}\|^2 + \widehat{F}_0 \tag{4.125}$$

Remark 1: The coefficient $\gamma_{0/0}$ can be interpreted as the reliability factor of the filtered estimate $\widehat{\mathbf{q}}_{0/0}$. The positive scalar $\widehat{F}_0$ is the minimal cost of filtering $\mathbf{q}_0$.

Quaternion Predicting Step at t_{k-1}

Let us assume that the quaternion filtering problem was solved at t_{k-1}, so that the function $F_{k-1}^\star\left(\mathbf{q}_{k-1}\right)$ was computed and is now of the form

$$F_{k-1}^\star\left(\mathbf{q}_{k-1}\right) = \frac{1}{2}\gamma_{k-1/k-1}\|\mathbf{q}_{k-1} - \widehat{\mathbf{q}}_{k-1/k-1}\|^2 + \widehat{F}_{k-1} \tag{4.126}$$

where $\widehat{\mathbf{q}}_{k-1/k-1}$, $\gamma_{k-1/k-1}$ and $\widehat{F}_{k-1}$ are known. The vector $\widehat{\mathbf{q}}_{k-1/k-1}$ denotes the a posteriori estimate of $\mathbf{q}_{k-1}$, given the measurements $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{k-1}\}$ and $\{\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{k-2}\}$. The scalar $\gamma_{k-1/k-1}$ is the reliability factor of $\widehat{\mathbf{q}}_{k-1/k-1}$. The scalar $\widehat{F}_{k-1}$ is the minimal cost of filtering $\mathbf{q}_{k-1}$. Moreover, we assume that the measurement $\bar{\mathbf{w}}_{k-1}$ was acquired and that its reliability factor, η_{k-1}, is given. The function $G_k(\mathbf{q}_k)$ is defined as follows

$$G_k(\mathbf{q}_k) \overset{\triangle}{=} \min_{\mathbf{w}_{k-1}} \left\{ F^\star_{k-1}\left(\Phi_{k-1}{}^T\left(\mathbf{q}_k - \mathbf{w}_{k-1}\right)\right) + \frac{1}{2}\eta_{k-1}\|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 \right\} \tag{4.127}$$

Optimization with respect to $\mathbf{w}_{k-1}$ Using Eq. (4.126) in Eq. (4.127) gives

$$\begin{aligned} &G_k(\mathbf{q}_k) = \qquad (4.128)\\ &\min_{\mathbf{w}_{k-1}} \left\{ \frac{1}{2}\gamma_{k-1/k-1}\|\Phi_{k-1}{}^T\left(\mathbf{q}_k - \mathbf{w}_{k-1}\right) - \widehat{\mathbf{q}}_{k-1/k-1}\|^2 + \frac{1}{2}\eta_{k-1}\|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 \right\} + \widehat{F}_{k-1} \end{aligned}$$

Then, *using the orthogonality property of the matrix* Φ_{k-1}, Eq. (4.128) can be rearranged as follows

$$\begin{aligned} &G_k(\mathbf{q}_k) = \qquad (4.129)\\ &\min_{\mathbf{w}_{k-1}} \left\{ \frac{1}{2}\gamma_{k-1/k-1}\|\mathbf{w}_{k-1} - \left(\mathbf{q}_k - \Phi_{k-1}\widehat{\mathbf{q}}_{k-1/k-1}\right)\|^2 + \frac{1}{2}\eta_{k-1}\|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 \right\} + \widehat{F}_{k-1} \end{aligned}$$

We now turn to the minimization problem over $\mathbf{w}_{k-1}$. Let us rewrite the cost function of Eq. (4.129),

$$\frac{1}{2}\gamma_{k-1/k-1}\|\mathbf{w}_{k-1} - \left(\mathbf{q}_k - \Phi_{k-1}\widehat{\mathbf{q}}_{k-1/k-1}\right)\|^2 + \frac{1}{2}\eta_{k-1}\|\mathbf{w}_{k-1} - \bar{\mathbf{w}}_{k-1}\|^2 \tag{4.130}$$

The optimal value of $\mathbf{w}_{k-1}$ can be computed either by means of differentiation of the cost function Eq. (4.130) or by means of factorization of the cost function. The second method is applied here for reasons that will appear in the following. The factorization procedure for the expression in Eq. (4.130) depending on $\mathbf{w}_{k-1}$ is referred to as type A. It is developed in Appendix C.1. Using the result of Appendix C.1, Eq. (4.130) is expressed as follows.

$$\begin{aligned} &\frac{1}{2}\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)\|\mathbf{w}_{k-1} - \frac{\gamma_{k-1/k-1}\left(\mathbf{q}_k - \Phi_{k-1}\widehat{\mathbf{q}}_{k-1/k-1}\right) + \eta_{k-1}\bar{\mathbf{w}}_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\|^2 \\ &+ \frac{1}{2}\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\|\mathbf{q}_k - \left(\Phi_{k-1}\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\right)\|^2 \end{aligned} \tag{4.131}$$

Equation (4.131) is now written as a sum of squares where only the first square contains $\mathbf{w}_{k-1}$. Thus, the value $\mathbf{w}_{k-1}$ for which Eq. (4.131) is minimal is found by inspection, which yields

$$\mathbf{w}^{\star}_{k-1}\left(\mathbf{q}_k\right) = \frac{\gamma_{k-1/k-1}\left(\mathbf{q}_k - \Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1}\right) + \eta_{k-1}\,\bar{\mathbf{w}}_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)} \tag{4.132}$$

For any particular value $\mathbf{q}_k$, Eq. (4.132) provides a smoothed estimate of $\mathbf{w}_{k-1}$. This estimate is the least squares fit of $\mathbf{w}_{k-1}$ to its measurement $\bar{\mathbf{w}}_{k-1}$, and to the value that satisfies the dynamic constraint. The degree of uncertainty in this smoothed estimate is reflected in the sensitivity of Eq. (4.131) to changes in $\mathbf{w}_{k-1}$. Thus, the scalar $\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)$ can be interpreted as the reliability factor of the smoothed estimate of $\mathbf{w}_{k-1}$.

Optimization with respect to $\mathbf{q}_k$ Substituting $\mathbf{w}^{\star}_{k-1}\left(\mathbf{q}_k\right)$ from Eq. (4.132) to $\mathbf{w}_{k-1}$ in Eq. (4.131) and writing the result in Eq. (4.128) yields

$$G_k\left(\mathbf{q}_k\right) = \frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\|\mathbf{q}_k - \left(\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\right)\|^2 + \widehat{F}_{k-1} \tag{4.133}$$

The optimal predicted value of $\mathbf{q}_k$, given the measurements $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_{k-1}\}$ and $\{\bar{\mathbf{w}}_0, \ldots, \bar{\mathbf{w}}_{k-1}\}$ is the value for which the cost $G_k\left(\mathbf{q}_k\right)$ is minimum, subject to the constraint Eq. (4.95). The minimization problem is solved as follows. The cost function Eq. (4.133) is factorized with respect to $\mathbf{q}_k$, subject to the unit norm constraint, as exposed in Appendix C.2. This factorization problem is referred as a problem of type B. Let us consider the part of Eq. (4.133) that depends on $\mathbf{q}_k$,

$$\frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\|\mathbf{q}_k - \left(\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\right)\|^2 \tag{4.134}$$

Applying the result of Appendix C.2 in Eq. (4.134) yields

$$\begin{aligned}&\frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\,\|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\|\;\|\mathbf{q}_k - \frac{\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}}{\|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\|}\|^2\\ &+ \frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\left(1 - \|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\|\right)^2\end{aligned} \tag{4.135}$$

Then, the unitary value of $\mathbf{q}_k$ that minimizes Eq. (4.135) is found by inspection, which yields the following expression for $\widehat{\mathbf{q}}_{k/k-1}$

$$\widehat{\mathbf{q}}_{k/k-1} = \frac{\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}}{\|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\|} \tag{4.136}$$

Remark 1: We can see, from Eq. (4.133), that the unconstrained least squares predicted estimate of $\mathbf{q}_k$ is $\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}$. Thus, the unitary estimate $\widehat{\mathbf{q}}_{k/k-1}$ is obtained by normalizing the unconstrained least squares estimate.

Remark 2: The minimal cost of the estimate $\widehat{\mathbf{q}}_{k/k-1}$, as given in Eq. (4.136), can be interpreted as a measure of how far the unconstrained least squares solution is from the quadratic constraint. In other words it is the cost of normalization.

Remark 3: In order to check the result Eq. (4.136), we use it in Eq. (4.132). Since, we obtain $\bar{\mathbf{w}}_{k-1}$, the expression Eq. (4.136) is correct.

We wish to express $G_k\,(\mathbf{q}_k)$ using the estimate $\widehat{\mathbf{q}}_{k/k-1}$. We can see from Eq. (4.135) that this was almost achieved. Let us define $\gamma_{k/k-1}$ and $\widehat{G}_k$ as follows,

$$\gamma_{k/k-1} \triangleq \frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\,\|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\| \tag{4.137}$$

$$\widehat{G}_k \triangleq \frac{1}{2}\,\frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)}\,\left(1 - \|\,\Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1} + \bar{\mathbf{w}}_{k-1}\|\right)^2 + \widehat{F}_{k-1} \tag{4.138}$$

Using Eqs. (4.136) through (4.138) in Eq. (4.133), the function $G_k\,(\mathbf{q}_k)$ can be expressed as follows

$$G_k\,(\mathbf{q}_k) = \frac{1}{2}\,\gamma_{k/k-1}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k-1}\|^2 + \widehat{G}_k \tag{4.139}$$

Defining $G^{\star}_k\,(\mathbf{q}_k)$ as

$$G^{\star}_k\,(\mathbf{q}_k) \triangleq \frac{1}{2}\,\gamma_{k/k-1}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k-1}\|^2 + \widehat{G}_k \tag{4.140}$$

we are ready to tackle the filtering problem at t_k.

Quaternion Filtering Step at t_k

Assuming that the function $G^{\star}_k\,(\mathbf{q}_k)$ in Eq. (4.140) was determined, and that a quaternion measurement $\overline{\mathbf{q}}_k$ was acquired at t_k as well as its reliability factor α_k. The function

$F_k(\mathbf{q}_k)$ is defined using Eq. (4.119) as

$$F_k(\mathbf{q}_k) \triangleq G_k^\star(\mathbf{q}_k) + \frac{1}{2}\alpha_k \|\mathbf{q}_k - \overline{\mathbf{q}}_k\|^2 \tag{4.141}$$

Substituting Eq. (4.140) in Eq. (4.141) yields

$$F_k(\mathbf{q}_k) = \frac{1}{2}\gamma_{k/k-1}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k-1}\|^2 + \frac{1}{2}\alpha_k\|\mathbf{q}_k - \overline{\mathbf{q}}_k\|^2 + \widehat{G}_k \tag{4.142}$$

We wish to minimize Eq. (4.142) over the unitary vector $\mathbf{q}_k$. The cost function Eq. (4.142) is the sum of three squares. The first one is related to the predicted estimate $\widehat{\mathbf{q}}_{k/k-1}$ and is weighted by $\gamma_{k/k-1}$. Thus, $\gamma_{k/k-1}$ can be interpreted as the reliability factor of the predicted estimate. The second term is related to the new observation $\overline{\mathbf{q}}_k$. The third term is the cost of the estimate $\widehat{\mathbf{q}}_{k/k-1}$ and is constant with respect to $\mathbf{q}_k$.

In order to solve the minimization problem of interest, we proceed in a similar way as for the time invariant algorithm. The part of the cost in Eq. (4.142) that is dependent on $\mathbf{q}_k$ is

$$\frac{1}{2}\gamma_{k/k-1}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k-1}\|^2 + \frac{1}{2}\alpha_k\|\mathbf{q}_k - \overline{\mathbf{q}}_k\|^2 \tag{4.143}$$

The expression in Eq. (4.143) is factorized with respect to $\mathbf{q}_k$, subject to the constraint Eq. (4.95) as developed in Appendix C.3 (factorization of type C). The factorized form of Eq. (4.143) is as follows

$$\begin{aligned}&\frac{1}{2}\|\,\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k\|\;\|\mathbf{q}_k - \frac{\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k}{\|\,\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k\|}\|^2 \\ &+ \left(\gamma_{k/k-1} + \alpha_k - \|\,\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k\|\right)\end{aligned} \tag{4.144}$$

By inspection of Eq. (4.144), we find the value of $\mathbf{q}_k$ for which Eq. (4.144) is minimum. This value, denoted by $\widehat{\mathbf{q}}_{k/k}$, is the filtered estimate of $\mathbf{q}_k$, given the sequences of measurements $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_k\}$ and $\{\,\bar{\mathbf{w}}_0, \ldots,\ \bar{\mathbf{w}}_{k-1}\}$. It is equal to

$$\widehat{\mathbf{q}}_{k/k} = \frac{\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k}{\|\,\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k\overline{\mathbf{q}}_k\|} \tag{4.145}$$

Remark 1: The new observation $\overline{\mathbf{q}}_k$ is used to modify the predicted value of $\mathbf{q}_k$ to obtain the current estimate of $\mathbf{q}_k$. This is the desired recursive feature of the algorithm.

We wish to express $F_k\,(\mathbf{q}_k)$ using the current estimate $\widehat{\mathbf{q}}_{k/k}$. Defining $\gamma_{k/k}$ and $\widehat{F}_k$ as

$$\begin{aligned} \gamma_{k/k} &\triangleq \|\,\gamma_{k/k-1}\,\widehat{\mathbf{q}}_{k/k-1} + \alpha_k \overline{\mathbf{q}}_k\,\| && (4.146)\\ \widehat{F}_k &\triangleq \left(\gamma_{k/k-1} + \alpha_k - \gamma_{k/k}\right) + \widehat{G}_k && (4.147) \end{aligned}$$

and using Eqs. (4.144) through (4.147) in the expression of $F_k\,(\mathbf{q}_k)$ in Eq. (4.142) yields

$$F_k\,(\mathbf{q}_k) = \frac{1}{2}\,\gamma_{k/k}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k}\|^2 + \widehat{F}_k \tag{4.148}$$

Finally, we define $F^\star_k\,(\mathbf{q}_k)$ as

$$F^\star_k\,(\mathbf{q}_k) = \frac{1}{2}\,\gamma_{k/k}\|\mathbf{q}_k - \widehat{\mathbf{q}}_{k/k}\|^2 + \widehat{F}_k \tag{4.149}$$

Comparing the cost functions $F^\star_{k-1}\left(\mathbf{q}_{k-1}\right)$ in Eq. (4.126) and $F^\star_k\,(\mathbf{q}_k)$ in Eq. (4.149) it can be seen that the recursion has been achieved.

Remark 1: The function $F^\star_k\,(\mathbf{q}_k)$ will be imbedded in the following prediction cost function. $\gamma_{k/k}$ will be the weighting factor of the filtered estimate $\widehat{\mathbf{q}}_{k/k}$ in this prediction step. $\widehat{F}_k$ is the cost of the estimate $\widehat{\mathbf{q}}_{k/k}$.

Remark 2: An estimate of $\mathbf{w}_{k-1}$ can be computed by substituting $\widehat{\mathbf{q}}_{k/k}$ to $\widehat{\mathbf{q}}_k$ in Eq. (4.132). The smoothed estimate of $\mathbf{w}_{k-1}$, denoted $\widehat{\mathbf{w}}_{k-1/k}$, given the measurements $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_k\}$ and $\{\,\bar{\mathbf{w}}_0, \ldots,\,\bar{\mathbf{w}}_{k-1}\}$ is computed next.

Process Bias Smoothing Step at t_k

Once the filtered estimate of the quaternion, $\widehat{\mathbf{q}}_{k/k}$, has been computed from Eq. (4.145) we can substitute it for $\mathbf{q}_k$ in Eq. (4.132) and obtain a smoothed estimate of $\mathbf{w}_{k-1}$, denoted by $\widehat{\mathbf{w}}_{k-1/k}$ as follows.

$$\widehat{\mathbf{w}}_{k-1/k} = \frac{\gamma_{k-1/k-1}\left(\widehat{\mathbf{q}}_{k/k} - \Phi_{k-1}\,\widehat{\mathbf{q}}_{k-1/k-1}\right) + \eta_{k-1}\,\bar{\mathbf{w}}_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)} \tag{4.150}$$

Denoting by $\widehat{\boldsymbol{\epsilon}}_{k-1/k}$ the smoothed estimate of the gyro outputs error, $\boldsymbol{\epsilon}_{k-1}$, we can compute it using Eq. (4.98). We begin by rewriting this equation

$$\mathbf{w}_k \;=\; -\frac{\Delta t}{2}\,\widehat{\Xi}_k\,\boldsymbol{\epsilon}_k \tag{4.151}$$

At this stage, we define the matrix $\widehat{\Xi}_{k/k}$ by substituting $\widehat{\mathbf{q}}_{k/k}$ to $\widehat{\mathbf{q}}_k$ in Eq. (4.97). Then, multiplying Eq. (4.151) by $\left(-\frac{2}{\Delta t}\,\widehat{\Xi}^T_{k/k}\right)$, on the left, and using the following identity (see Appendix A.1)

$$\widehat{\Xi}^T_{k/k}\,\widehat{\Xi}_{k/k} \;\equiv\; I_3 \tag{4.152}$$

yields

$$\boldsymbol{\epsilon}_k \;=\; -\frac{2}{\Delta t}\,\widehat{\Xi}^T_{k/k}\;\mathbf{w}_k \tag{4.153}$$

We see that substituting $\widehat{\mathbf{w}}_{k-1/k}$ to $\mathbf{w}_k$ into Eq. (4.153) provides a smoothed estimate of the gyroscopes error, $\widehat{\boldsymbol{\epsilon}}_{k-1/k}$.

$$\widehat{\boldsymbol{\epsilon}}_{k-1/k} \;=\; -\frac{2}{\Delta t}\,\widehat{\Xi}^T_{k/k}\;\widehat{\mathbf{w}}_{k-1/k} \tag{4.154}$$

Angular Velocity Estimation at t_k

Since the process $\boldsymbol{\epsilon}_k$ was modelled as random bias process, the knowledge of $\widehat{\mathbf{w}}_{k-1/k}$ is theoretically of no use, since it refers to the value of $\boldsymbol{\epsilon}_k$ at the *previous* time epoch. However, in practice, the error process $\boldsymbol{\epsilon}_k$ can be modelled as a constant bias (see e.g. [23, p. 138]). So, extrapolating the smoothed estimate $\widehat{\boldsymbol{\epsilon}}_{k-1/k}$ at t_k yields

$$\widehat{\boldsymbol{\epsilon}}_{k/k} \;=\; \widehat{\boldsymbol{\epsilon}}_{k-1/k} \tag{4.155}$$

When a measurement of the angular velocity $\boldsymbol{\omega}_k$ at t_k is made, we use $\widehat{\boldsymbol{\epsilon}}_{k/k}$ to compute an estimate of $\boldsymbol{\omega}_k$, denoted by $\widehat{\mathbf{w}}_{k/k}$.

$$\widehat{\boldsymbol{\omega}}_{k/k} \;=\; \boldsymbol{\omega}_k - \widehat{\boldsymbol{\epsilon}}_{k/k} \tag{4.156}$$

The vector $\widehat{\boldsymbol{\omega}}_{k/k}$ is an a priori estimate of $\boldsymbol{\omega}_k$ given the sequence of observations $\{\overline{\mathbf{q}}_0, \ldots, \overline{\mathbf{q}}_k\}$ and $\{\,\bar{\mathbf{w}}_0, \ldots,\ \bar{\mathbf{w}}_{k-1}\}$. An a priori value of the transition matrix Φ_k, denoted by $\widehat{\Phi}_{k/k}$, can then be computed using the estimate $\widehat{\boldsymbol{\omega}}_{k/k}$. The predicting stage Eq. (4.136) is then replaced by

$$\widehat{\mathbf{q}}_{k+1/k} \;=\; \widehat{\Phi}_{k/k}\;\widehat{\mathbf{q}}_{k/k} \tag{4.157}$$

Equation (4.157) is an alternate way to compute a predicted estimate of $\mathbf{q}_k$ at t_{k+1}. It has the advantage of relaxing the a priori knowledge of the gyro biases. Indeed, we can set $\bar{\boldsymbol{\epsilon}}_k = \mathbf{0}$, which gives $\bar{\mathbf{w}}_k = \mathbf{0}$ and work with $\widehat{\mathbf{w}}_{k-1/k}$ as computed from Eq. (4.150). On the other hand, Eq. (4.157) has not been developed in the context of an optimal estimator, and we can not derive the computation steps relative to the weighting factors. However, using the heuristic argument that both a priori values of $\mathbf{w}_k$, namely $\bar{\mathbf{w}}_k$ and $\widehat{\mathbf{w}}_{k-1/k}$ are close one to each other, we may conclude that their respective weights should have also close values. This yields the following computation for $\gamma_{k+1/k}$

$$\gamma_{k+1/k} = \frac{\gamma_{k/k}\eta_k}{\gamma_{k/k} + \eta_k} \| \widehat{\Phi}_{k/k}\, \widehat{\mathbf{q}}_{k/k} \| = \frac{\gamma_{k/k}\eta_k}{\gamma_{k/k} + \eta_k} \tag{4.158}$$

4.3.5 Weighting Coefficients η_k

Given the standard deviation, σ_ϵ, of the gyro noise, $\boldsymbol{\epsilon}_k$, [see Eq. (4.99], the coefficients η_k are chosen as

$$\eta_k = \frac{1}{(\,\sigma_\epsilon\, \Delta t)^2} \tag{4.159}$$

We know that $(\,\sigma_\epsilon\, \Delta t)^2$ is of the order of the variance of the measurement $\bar{\mathbf{w}}_k$ (Eqs. 4.99 and 4.98). Thus, the choice expressed in Eq. (4.159) is meaningful; each $\mathbf{w}_k$-measurement residual in the least squares cost function of Eq. (4.110) is weighted according to the accuracy of the associated measurement.

4.3.6 Algorithm Summary

1. Initialization Stage:

$$\widehat{\mathbf{q}}_{0/0} = \overline{\mathbf{q}}_0 \tag{4.160a}$$

$$\gamma_{0/0} = \alpha_0 \tag{4.160b}$$

$$\widehat{F}_0 = 0 \tag{4.160c}$$

$$\widehat{\Phi}_{0/0} = \Phi_0 \tag{4.160d}$$

2. Predicting Stage

$$\widehat{\mathbf{q}}_{k/k-1} = \widehat{\Phi}_{k-1/k-1}\, \widehat{\mathbf{q}}_{k-1/k-1} \tag{4.161a}$$

$$\gamma_{k/k-1} = \frac{\gamma_{k-1/k-1}\eta_{k-1}}{\left(\gamma_{k-1/k-1} + \eta_{k-1}\right)} \tag{4.161b}$$

$$\widehat{G}_k = \frac{1}{2}\,\gamma_{k/k-1} + \widehat{F}_{k-1} \tag{4.161c}$$

3. Quaternion Filtering Stage:

$$\gamma_{k/k} = \| \, \gamma_{k/k-1} \, \widehat{\mathbf{q}}_{k/k-1} + \alpha_k \overline{\mathbf{q}}_k \| \tag{4.162a}$$

$$\widehat{\mathbf{q}}_{k/k} = \frac{\gamma_{k/k-1} \, \widehat{\mathbf{q}}_{k/k-1} + \alpha_k \overline{\mathbf{q}}_k}{\gamma_{k/k}} \tag{4.162b}$$

$$\widehat{F}_k = \alpha_k - \left(\gamma_{k/k} - \gamma_{k/k-1} \right) + \widehat{G}_k \tag{4.162c}$$

4. Gyro Biases Smoothing Stage:

$$\widehat{\mathbf{w}}_{k-1/k} = \frac{\gamma_{k-1/k-1} \left(\widehat{\mathbf{q}}_{k/k} - \Phi_{k-1} \, \widehat{\mathbf{q}}_{k-1/k-1} \right) + \eta_{k-1} \, \bar{\mathbf{w}}_{k-1}}{\gamma_{k-1/k-1} + \eta_{k-1}} \tag{4.163a}$$

$$\widehat{\boldsymbol{\epsilon}}_{k-1/k} = -\frac{2}{\Delta t} \, \widehat{\Xi}^T_{k/k} \, \widehat{\mathbf{w}}_{k-1/k} \tag{4.163b}$$

5. Angular Velocity Estimation:

$$\widehat{\boldsymbol{\omega}}_{k/k} = \boldsymbol{\omega}_k - \widehat{\boldsymbol{\epsilon}}_{k-1/k} \tag{4.164a}$$

$$\widehat{\Omega}_{k/k} = \frac{1}{2} \begin{bmatrix} -\left[\widehat{\boldsymbol{\omega}}_{k/k} \times \right] & \widehat{\boldsymbol{\omega}}_{k/k} \\ -\widehat{\boldsymbol{\omega}}^T_{k/k} & 0 \end{bmatrix} \tag{4.164b}$$

$$\widehat{\Phi}_{k/k} = \exp\left(\widehat{\Omega}_{k/k} \, \Delta t \right) \tag{4.164c}$$

4.3.7 Simulation Study

Simulation Description

In order to focus on the quaternion and gyro biases estimation problems we simulate the case of a static body frame, $\mathcal{B}$, with respect to some referential frame, $\mathcal{R}$. The output of the direction sensors are corrupted by the vector error process $\boldsymbol{\delta b}_k$, which is computed as a Gaussian zero-mean white noise with the following statistics (typical for Magnetometers and coarse Earth Horizon Sensors)

$$\boldsymbol{\delta b}_k \sim \mathcal{N}\left(\mathbf{0}, (10^{-2}\,\mathrm{rad})^2 \, I_3 \right)$$

We assume that the outputs of the different sensors are uncorrelated one with each other. Each quaternion measurement is computed by the $\mathbf{q}$-method from a batch of two simultaneous vector measurements. Using the standard deviation of 10^{-2}, we have $\alpha_k = 2\;10^4$.

The output error of the gyroscope is modelled as a Gaussian sequence with the following statistics

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}\left(10^{-4}\,\mathrm{rad}/s,\left(10^{-6}\,\mathrm{rad}/s\right)^2\;I_3\right)$$

We use relative high values in the model of $\boldsymbol{\epsilon}_k$ in order to check the efficiency of the algorithm within short time runs. The incremental time Δt is equal to 0.1 second. The weights η_k are $\eta_k = 10^{14}$. The gyro sampling time is 0.1 sec. The vector measurement sampling time is 2 sec. Each simulation run lasts 1000 sec. In the first phase of the simulation (150 sec.) the filter is run without estimating the gyro biases. The bias smoothing procedure is turned on at $t = 150$ sec. and is maintained until the simulation stops.

Results Analysis

Figure 4.3 summarizes the results of the time varying simulation. It shows the time history of the angular estimation error. The evolution of the error can be analyzed in four phases. During the first phase, from $t = 0$ to $t = 40$ seconds, the error decreases from about 1 degree down to 0.2 degree. During the second phase, from $t = 40$ to $t = 150$ seconds, the estimation error is increasing linearly with the time and reaches its maximum at $t = 150$ seconds, at a value of 0.76 degree, as is seen on the zoomed plot. The third phase, from $t = 150$ to $t = 600$ seconds, consists of a slow monotonous decrease in the error, that reaches the steady-state phase, from $t = 600$ seconds until the end, and remains on a level of approximately 0.06 degree (see the zoomed plot).

First Phase. At the beginning of the run the filter converges. In spite of a very badly modelled process equation (the unmodeled bias produces an error of $10^o/hr$), the filter succeeds in reducing the error level by filtering new observations. These observation have an uncertainty level of about 1^o. Thus the error starts from a level of 1^o and decreases as long as the measurements are processed, like in the time invariant case. As a result of the increasing number of processed measurements, the weight $\gamma_{k/k}$ of the filtered estimate $\widehat{\mathbf{q}}_{k/k}$ is also increasing. At $t = 40$ seconds, after 20 update stages, the ratio $\alpha_{k+1}/\,\gamma_{k/k}$ is about 1/20, so that the filter becomes less sensitive to the incoming observation.

Second Phase. The filter is diverging. Two main causes may be responsible for the filter divergence. The first one is the error in the dynamics model. Indeed, the propagation step is performed using a noisy Φ_k-matrix, which is computed from the biased measurement, $\boldsymbol{\omega}_k$. The second cause is that the weights η_k are several order of magnitude higher than the weights α_k. This arises from the fact that the given gyroscopes are much more accurate sensors than the given direction sensors. As a result, the coefficient $\gamma_{k+1/k}$ is almost identical to $\gamma_{k/k}$ and the predicting stage is performed as if there was no propagation noise. So, we are in the case of a mismodelled dynamics together with a low level of process noise in the filter model. As a consequence the filter gets locked on the a state estimate which is highly drifting. The zoomed plot in Figure 4.3, centered at $t = 150$ seconds, clearly shows how the unmodeled bias is linearly increasing the error between two measurements, and how the update stage of the estimate does not succeed in compensating the predicting error.

Third Phase. Once the gyroscopes biases and the angular velocity are being estimated the convergence of the filter is recovered with a slow rate. The zoomed plot at $t = 150$ seconds manifests the difference in the predicting error process, before and after that the bias is estimated. After $t = 150$ seconds, the error is constant between measurements, since the estimated angular velocity is close to its true value ($\mathbf{0}\,\mathrm{rad}/s.$). However the rate of convergence is slow, since the relative weight of the estimate is still increasing while the weight of the new observation stays at the same value. After approximately 600 seconds, that is 300 observations fixes, the error reaches and remains at the level of 0.06 deg.

4.3.8 Concluding Remarks

The algorithm derived for the time-invariant case was extended to the time-varying case using the Dynamic Programming approach. Developed in the framework of constrained least squares estimation, the proposed algorithm optimally filters measurement noises as well as process noises. Moreover, the quaternion normalization is optimally preserved along the estimation procedure.

The filter shows good performance in a relative highly noisy environment. The estimation error steady-state is about 0.15 deg. for a level of 1 deg. in the measurement noise and $10\,\mathrm{deg}/hr$ in the process noise.

Severe errors in the dynamics model, like unmodeled gyro biases, lead the estimation error to divergence. Divergence can be prevented by estimating the gyro biases, which are modelled as constant parameters with a known a priori value. The biases estimation step consists of a simple smoothing procedure; this procedure, as opposed to the classical state augmentation procedure does not increase the filter dimension. Although divergence is avoided the convergence rate during recovery from high estimation errors is considerably low.

4.4 Concluding Remarks

In this chapter a quadratically constrained batch-recursive least squares filter for attitude quaternion estimation was introduced. The Dynamic Programming approach, together with basic factorization techniques, allowed the derivation of the recursive relations for the time-varying filter in a straightforward manner.

The proposed filter is able to estimate parameters other than attitude, like constant gyro biases and body angular velocity. The estimation process is a two-stage procedure. The first stage embeds a standard closed-form batch algorithm for single-frame estimation of the attitude quaternion. That stage relies on the ability of the attitude sensors to acquire a batch of at least two simultaneous vector measurements. The second stage includes a recursive least squares procedure, where the quaternion estimates of the first-stage are optimally blended under the unit-norm constraint of the quaternion estimate. The constant gyro biases are estimated through a smoothing procedure which does not increase the filter dimension, as opposed to the known state augmentation procedure. All the computational steps in the filter have analytical expressions which keep the algorithm computationally simple. Simulations show the efficiency of the proposed filter for both time-invariant and time-varying systems.

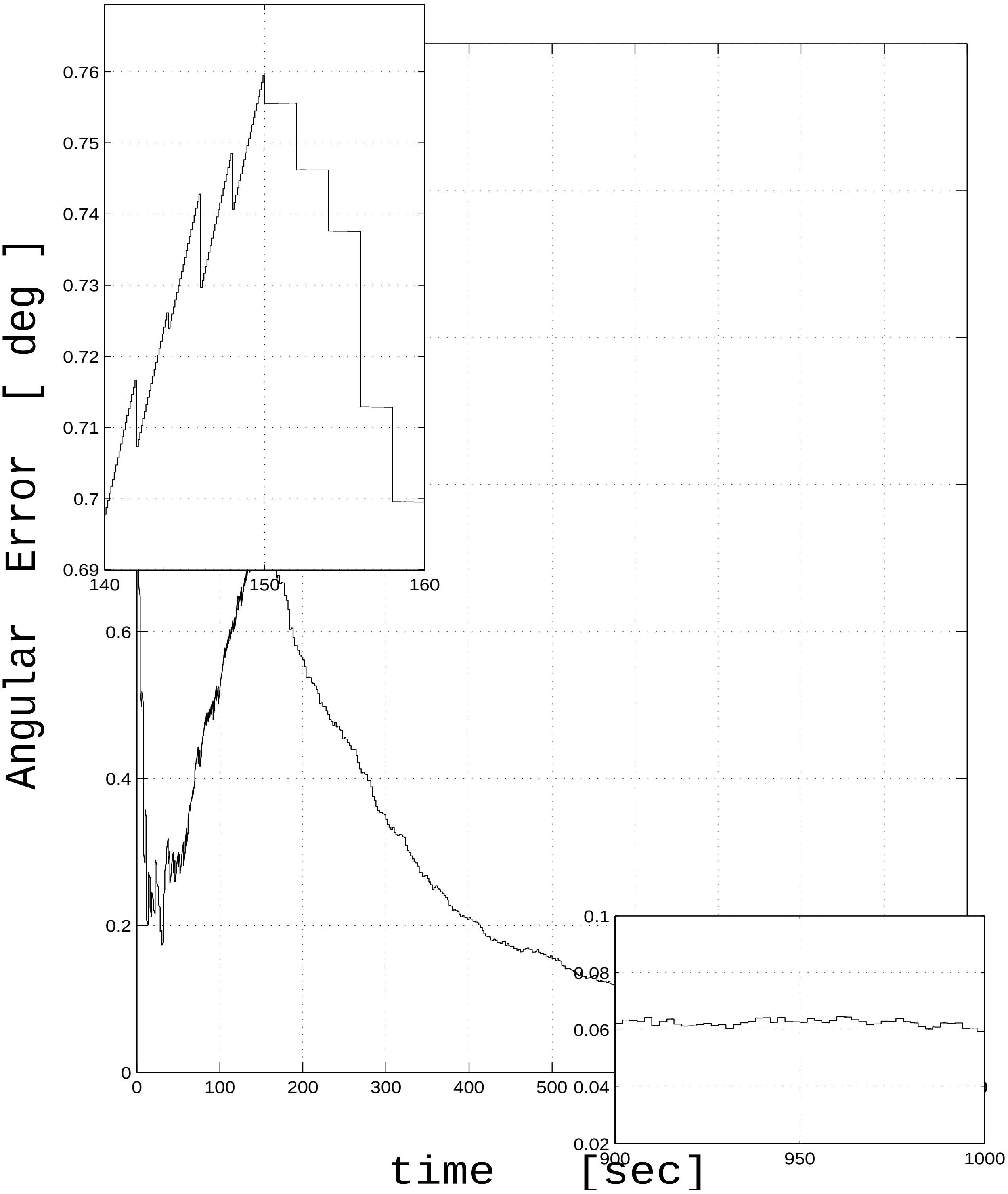


Fig. 4.3: Time-varying case. Angular estimation error for a sample run.

Chapter 5

State Matrix Kalman Filter

5.1 Introduction

This chapter is concerned with the development of a general discrete-time *state-matrix* Kalman filter. As mentioned in chapter 1, there are systems that are most conveniently described by matrix processes and matrix measurement equations, and research has been continuously conducted to develop estimation algorithms that fit naturally those systems [32, 36–39, 41–44].

The matrix Kalman filter (MKF) developed here is a new algorithm that can implement in a natural way plant models that are described by matrix equations. Indeed, this algorithm yields the state matrix estimate and the estimation error covariance matrix in terms of the original system matrices. A compact notation is used in the MKF formulation in order to aid both intuition and the mathematical manipulation.

In this work, we consider a linear discrete-time plant with a state matrix process and matrix measurements. The state matrix dynamics is described by a stochastic difference equation with an additive matrix zero-mean white noise. The measurement is modelled as a linear matrix-valued function of the state matrix with an additive matrix zero-mean white noise. All matrices are defined over the real field. Operating on this plant the matrix Kalman filter provides a sequence of optimal matrix estimates and, as the vector KF, yields the covariance of the estimation error.

Section 5.2 presents the main results of this work. It describes the formulation of the problem, the solution approach, and summarizes the MKF algorithm. It shows that the MKF algorithm is a direct generalization of the classical discrete vector Kalman filter. It also shows how previous matrix filters are included as special cases of the MKF algorithm. The derivation of the filter is detailed in section 5.3. The derivation relies on definitions and propositions that are stated and proved in Appendix D. Numerical examples are proposed in section 5.4 to illustrate the validity of the MKF. The last section presents the conclusions of this chapter.

5.2 Main Results

5.2.1 Problem Formulation

Matrix State-Space Model

Consider a linear discrete-time stochastic dynamic system governed by the difference equation

$$X_{k+1} = \sum_{r=1}^{R} \Theta^r_k \, X_k \, \Psi^r_k + W_k \tag{5.1}$$

where $X_k \in \mathbb{R}^{m\times n}$ is the *matrix* state variable at time t_k, $\Theta^r_k \in \mathbb{R}^{m\times m}$ and $\Psi^r_k \in \mathbb{R}^{n\times n}$ are "transition matrices", and W_k is a $m \times n$ noise matrix (see Definition D.1.4). The *matrix* measurement equation of the matrix plant is

$$Y_{k+1} = \sum_{s=1}^{S} H^s_{k+1} \, X_{k+1} \, G^s_{k+1} + V_{k+1} \tag{5.2}$$

where $Y_{k+1} \in \mathbb{R}^{p\times q}$ is the matrix measurement of X_{k+1} at time t_{k+1}, $H^s_{k+1} \in \mathbb{R}^{p\times m}$ and $G^s_{k+1} \in \mathbb{R}^{n\times q}$ are the observation matrices, and $V_{k+1} \in \mathbb{R}^{p\times q}$ is a noise matrix.

Equation (5.1) is the general expression for a linear matrix difference equation. This is shown by applying Proposition D.2.3 of Appendix D.2 where

$$Z = X_{k+1} \ , \ X = X_k \ , \ A^u = \Theta^r_k \ , \ B^u = \Psi^r_k$$
$$u = r \ , \ v = R \ , \ i_1 = i_3 = m \ , \ i_2 = i_4 = n$$

The system equation of any linear plant that is described by a matrix difference equation will be a particular case of Eq. (5.1). In the general case, the number of terms in the sum of Eq. (5.1) may reach $(mn)^2$; that is $R = (mn)^2$. However, particular plants may need less terms in the system equation, so that R is left as a design parameter.

Equation (5.2) is shown to be the general expression for a linear matrix measurement model of the matrix plant by applying Proposition D.2.3 where

$$Z = Y_{k+1} \ , \ X = X_{k+1} \ , \ A^u = H^s_{k+1} \ , \ B^u = G^s_{k+1}$$
$$u = s \ , \ v = S \ , \ i_1 = p \ , \ i_2 = q \ , \ i_3 = m \ , \ i_4 = n$$

The parameter S in Eq. (5.2) can also be considered as a design parameter.

Statistical Models

The $m \times n$ matrix sequence W_k is assumed to be a Gaussian zero-mean white noise sequence. The $mn \times mn$ covariance matrix of W_k (defined using Definition D.1.4) is denoted by Q_k and is assumed to be known. The statistical assumptions on W_k are formulated in concise notation as follows

$$W_k \sim \mathcal{N}\{\, O_{m\times n}, Q_k\} \tag{5.3}$$

where $O_{m\times n}$ denotes the $m \times n$ zero-matrix.

The $p \times q$ sequence V_k is assumed to be a Gaussian zero-mean white noise sequence. Similarly, the $pq \times pq$ covariance matrix of V_k, that is denoted by R_k, is assumed to be known. The statistical model of V_k is therefore

$$V_k \sim \mathcal{N}\{\, O_{p\times q}, R_k\} \tag{5.4}$$

where $O_{p\times q}$ denotes the $p \times q$ zero-matrix.

The initial state matrix X_0 is assumed to be Gaussian with mean $\bar{X}_0$ and $mn \times mn$ covariance matrix Π_0; that is

$$X_0 \sim \mathcal{N}\{\, \bar{X}_0, \Pi_0\} \tag{5.5}$$

It is further assumed, as in the basic vector state-space model, that W_k and V_k are not correlated with one another, and with X_0.

Estimation Problem

The matrix Kalman filter is the unbiased minimum variance estimator of the $m \times n$ matrix state X_k at t_k, given a sequence of $p \times q$ matrix observations up to t_k, $\{Y_l\}$,

$l = 1 \ldots k$. Let $\widehat{X}_{k/k}$ and $\Delta X_{k/k}$ denote, respectively, the a posteriori state estimate and the a posteriori estimation error, where $\Delta X_{k/k} \overset{\triangle}{=} X_k - \widehat{X}_{k/k}$. Let $P_{k/k}$ denote the a posteriori estimation error covariance matrix, where $P_{k/k} = \mathrm{cov}(\Delta X_{k/k}) \overset{\triangle}{=} \mathrm{cov}(\mathbf{\Delta x}_{k/k})$, and $\mathbf{\Delta x}_{k/k} \overset{\triangle}{=} \mathrm{vec}\,\Delta X_{k/k}$ (see Definitions D.1.1 and D.1.2 of Appendix D.1). Then, the filtering problem is related to the following minimization problem

$$\min_{X_k} \left\{ tr\left(P_{k/k} \right) \right\} \tag{5.6}$$

subject to the Eqs. (5.1) and (5.2) and to the stochastic assumptions of section 5.2.1. In the case where the state and the measurement are represented as vectors the solution of the above minimization problem leads to the standard Kalman filter.

5.2.2 Solution Approach

The solution approach consists of three principal steps. In the first step we apply the vec operator on the matrix plant described by Eqs. (5.1) and (5.2). The vec operator is defined in Definition D.1.2 of Appendix D.1. Thus, the matrix equations (5.1) and (5.2) are transformed into equivalent *vector* equations. The result of this step is a standard state-space model, where the state is a mn-dimensional vector and the measurement is a pq-dimensional vector. In the following, we refer to this system as the “Vec-system”.

In the second step Kalman filter theory is applied to the “Vec-system”. The Time Update and Measurement Update stages are developed using the classical Kalman filter algorithm. As a result we obtain two sets of equations: the first set computes the mn-dimensional state estimate of the “Vec-system” and the second set computes the $mn \times mn$ estimation error covariance matrix.

The third step aims at retrieving the matrix formalism of the original problem. This is done by applying the inverse of the vec-operator on the Kalman filter of the “Vec-system”. It stems from Definition D.1.4 that the covariance computations of the “Vec-system” and of the matrix system are identical. Thus *the covariance computation performed in the second step is left in its current extended form.* However, the state estimate equations, which are mn-dimensional vector equations, are transformed into equivalent $m \times n$ matrix equations. Hence, a matrix innovations sequence is naturally defined and the Time Update and Measurement Update stages are expressed in terms of the original matrices. These manipulations yield a state-matrix Kalman filter in a compact notation.

5.2.3 State-Matrix Kalman Filter

The state-matrix Kalman filter is summarized in the following. The proof of the algorithm is provided in section 5.3. The symbol ⊗ used in Eqs. (5.9) and (5.12) denotes the Kronecker product (see Definition D.1.3 of Appendix D.1).

1. Initialization:

$$\widehat{X}_{0/0} = \bar{X}_0 \qquad P_{0/0} = \Pi_0 \qquad \text{(Given)} \tag{5.7}$$

2. Time Update:

$$\widehat{X}_{k+1/k} = \sum_{r=1}^{\mu} \Theta_k^r \, \widehat{X}_{k/k} \, \Psi_k^r \tag{5.8}$$

$$\Phi_k = \sum_{r=1}^{\mu} \left[\left(\Psi_k^r \right)^T \otimes \Theta_k^r \right] \tag{5.9}$$

$$P_{k+1/k} = \Phi_k \, P_{k/k} \, \Phi_k^T + Q_k \tag{5.10}$$

3. Measurement Update:

$$\widetilde{Y}_{k+1} = Y_{k+1} - \sum_{s=1}^{\nu} H_{k+1}^s \, \widehat{X}_{k+1/k} \, G_{k+1}^s \tag{5.11}$$

$$\mathcal{H}_{k+1} = \sum_{s=1}^{\nu} \left[\left(G_{k+1}^s \right)^T \otimes H_{k+1}^s \right] \tag{5.12}$$

$$S_{k+1} = \mathcal{H}_{k+1} \, P_{k+1/k} \, \mathcal{H}_{k+1}^T + R_{k+1} \tag{5.13}$$

$$K_{k+1} = P_{k+1/k} \, \mathcal{H}_{k+1}^T \, S_{k+1}^{-1} \tag{5.14}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \sum_{j=1}^{n} \sum_{l=1}^{q} K_{k+1}^{jl} \, \widetilde{Y}_{k+1} \, E^{lj} \tag{5.15}$$

where K_{k+1}^{jl} is a $m \times p$ submatrix of the $mn \times pq$ matrix K_{k+1} defined by

$$K_{k+1} = \underbrace{\begin{bmatrix} K_{k+1}^{11} & \cdots & K_{k+1}^{1l} & \cdots \\ \vdots & \ddots & \vdots & \ddots \\ K_{k+1}^{j1} & \cdots & K_{k+1}^{jl} & \cdots \\ \vdots & \ddots & \vdots & \ddots \end{bmatrix}}_{\text{q matrices}} \left.\vphantom{\begin{bmatrix} K \\ \vdots \\ K \\ \vdots \end{bmatrix}}\right\} \text{n matrices} \tag{5.16}$$

and E^{lj} is a $q \times n$ matrix with 1 at position (lj) and 0 elsewhere.

$$P_{k+1/k+1} = \left(I_{mn} - K_{k+1} \, \mathcal{H}_{k+1} \right) P_{k+1/k} \left(I_{mn} - K_{k+1} \, \mathcal{H}_{k+1} \right)^T + K_{k+1} \, R_{k+1} K_{k+1}^T \tag{5.17}$$

where I_{mn} is the $mn \times mn$ identity matrix.

The variance and the covariance of $\Delta X[i,j]$ (the element (ij) in the matrix ΔX) are (see Proposition D.2.8 of Appendix D.2)

$$\text{var}\{\,\Delta X[i,j]\} = P\,[(j-1)m+i,(j-1)m+i] \tag{5.18}$$

$$\text{cov}\{\,\Delta X[i,j],\,\Delta X[k,l]\} = P[(j-1)m+i,(l-1)m+k] \tag{5.19}$$

where $i,k = 1\ldots m$, and $j,l = 1\ldots n$. The variable ΔX denotes either the a posteriori or the a priori estimation error as applicable, and P is the associated covariance matrix.

5.2.4 Discussion

The matrix Kalman filter algorithm, which is described in Eqs. (5.7) to (5.19), operates on the matrix state-space system given in Eqs. (5.1) and (5.2). Under the noise stochastic model assumptions of section 5.2.1, the matrix Kalman filter yields the unbiased minimum variance estimate of the state-matrix. The proposed filter behaves like the conventional linear Kalman filter (KF). It is a time varying algorithm for optimal recursive estimation of the state process using a sequence of measurements. Similarly to the conventional KF, the computations of the a priori state estimate (Eq. 5.8) and of the innovations sequence (Eq. 5.11) are directly derived from the system process and measurement equations (5.1) and (5.2) respectively.

It should be emphasized that the outputs of the matrix KF, e.g. the values of the estimate and of the error covariance matrix, are identical to those of a conventional KF operating on the “Vec-system”. The formulation of the state estimation equations is however different (Eqs. 5.8, 5.11 and 5.15). In the matrix KF the algorithm for the Time Update stage and the Measurement Update stage of the state is formulated using the coefficients of the original matrix state-space model. However, the covariance computation in the MKF is the same as in the KF of the “Vec-system”. The interpretation of the a posteriori estimation-error covariance-matrix in terms of the state-matrix elements is described in equations (5.18) and (5.19). The latter equations are based on Proposition D.2.8 of Appendix D, where $X = \Delta X_{k/k}$ and $P = P_{k/k}$.

The proposed matrix KF is a natural extension of the conventional KF. The two new dimensions are the number of columns in the state-matrix, n, and the number of columns in the matrix measurement, q. The ordinary vector Kalman filter is a special case of the matrix KF, since taking $n = q = 1$ in the proposed algorithm yields *exactly* the Kalman filter.

From MKF to matrix BLUE

The matrix Best Linear Unbiased Estimator (BLUE), which is proposed in [43], is a special case of the MKF algorithm. It is well known (see e.g. [40, p. 125]) that in the vector case the BLUE algorithm is a special case of the Weighted Least-Squares (WLS) algorithm, where the weighted matrix is chosen as the inverse of the measurement covariance matrix. It is also well-known that the WLS is a special case of the linear time-invariant Kalman filter, where there is no a priori estimate. Thus, conceptually the matrix case has the same properties. We show here that the MKF algorithm does indeed lead to the matrix BLUE algorithm. We first review the matrix BLUE of [43]. Consider the following multivariate model.

$$Y = HX + V \tag{5.20}$$

$$E\left[\mathbf{v}_i\right] = \mathbf{0} \qquad i = 1, 2, \ldots, p \tag{5.21}$$

$$\text{cov}\left[\mathbf{v}_i \mathbf{v}_j\right] = \delta_{ij} R \tag{5.22}$$

where Y is $p \times n$, H is a full-rank $p \times n$ matrix, X is a $m \times n$ matrix of unknown parameters, V is a $p \times n$ noise matrix, $\mathbf{v}_i$ is the i^{th} row of V, δ_{ij} is the delta of Kronecker, and R is a $n \times n$ positive definite matrix. The matrix BLU estimator of X, denoted by $\overline{X}$, is given as (see [43])

$$\overline{X} = \left(H^T H\right)^{-1} H^T Y \tag{5.23}$$

This proves the well-known fact that the BLUE of X is the simple least squares (LS) estimator derived equation by equation.

Since BLUE, as given in Eq. (5.23) is related to the Information form of the LS estimator (see [40, p. 59]), we will use the Information form of the MKF algorithm, which is described in Appendix D.3. Consider the special case of a time-invariant matrix plant, where the measurement model is described by Eq. (5.20), the measurement noise stochastic model is described by Eqs. (5.21) and (5.22), and there is no a priori estimate. Note that in the present case $q = n$. The above assumptions yield

$$\mathcal{H} = I_n \otimes H \tag{5.24}$$

$$\operatorname{cov}[V] = R \otimes I_p \tag{5.25}$$

$$\widehat{X}_0 = O_{m\times n} \tag{5.26}$$

$$P_0^{-1} = O_{mn} \tag{5.27}$$

where $\otimes$ denotes the Kronecker product, and O_{mn} denotes the mn-dimensional matrix of zeros.

Proposition 5.2.1. *Under the assumptions detailed in Eqs.* (5.24) *to* (5.27)*, the MKF Information algorithm, as given in Appendix D.3, yields the matrix BLUE that is given in Eq.* (5.23)*.*

Proposition 5.2.1 is proved in Appendix D.4.

From MKF to the Simplified Multivariate Least-Squares Algorithm

The matrix filter proposed in [44, p. 96], called Simplified Multivariate Least-Squares (SMLS) Algorithm, can be derived as a special case of the MKF algorithm. We begin by reviewing the SMLS algorithm. That algorithm was developed in the framework of Deterministic Parameter Estimation Theory. The model used in [44, p. 96] is as follows

$$\mathbf{y}_{k+1} = X^T \mathbf{h}_{k+1} \tag{5.28}$$

$$P_0 = I_n \otimes \overline{P}_0 \tag{5.29}$$

$$R = I_n \tag{5.30}$$

where $\mathbf{y}_{k+1}$ is $n \times 1$, X is a $m \times n$ matrix of unknown parameters, $\mathbf{h}_{k+1}$ is $m \times 1$, P_0 is the $mn \times mn$ initial matrix used in the recursion, and R is the inverse of the weighting matrix associated with the $(k+1)^{th}$ measurement. Under the above assumptions, the filter SMLS is formulated as follows

$$X_{k+1} = X_k + P_k \mathbf{h}_{k+1} \left(\mathbf{y}_{k+1}^T - \mathbf{h}_{k+1}^T X_k\right) / \left(1 + \mathbf{h}_{k+1}^T P_k \mathbf{h}_{k+1}\right) \tag{5.31}$$

$$P_{k+1} = P_k - P_k \mathbf{h}_{k+1} \mathbf{h}_{k+1}^T P_k / \left(1 + \mathbf{h}_{k+1}^T P_k \mathbf{h}_{k+1}\right) \tag{5.32}$$

Taking the transpose of both sides of Eq. (5.28), and adding a noise term to the right-hand side yields the following measurement model, on which the MKF can operate.

$$\mathbf{y}_{k+1}^T = \mathbf{h}_{k+1}^T X + \mathbf{v}_{k+1}^T \tag{5.33}$$

where $\mathbf{v}_{k+1}$ is a $n \times 1$ zero-mean white noise process with known covariance $R_{k+1} = I_n$. Assume, moreover, that the n columns of the initial estimation error are independently identically distributed with $m \times m$ covariance matrix $\overline{P}_0$, then we have

$$R_{k+1} = I_n \tag{5.34}$$

$$P_0 = I_n \otimes \overline{P}_0 \tag{5.35}$$

Proposition 5.2.2. *Under the model assumptions detailed in Eqs.* (5.33) *to* (5.35)*, the time-invariant MKF algorithm, as given in Eqs.* (5.11) *to* (5.17)*, yields the SMLS algorithm that is formulated in Eqs.* (5.31) *and* (5.32)*.*

Proposition 5.2.2 is proved in Appendix D.5.

5.3 Filter Derivation

We turn now to the development of the matrix Kalman filter, which is described in Eqs. (5.7) through (5.19). The development uses Propositions that are stated and proved in Appendix D.2.

5.3.1 Time Update Stage

Assume that $\widehat{X}_{k/k}$ and $P_{k/k}$ have been computed at time t_k, and that Q_k is known. Applying the Vec-operator to Eq. (5.1) yields

$$\operatorname{vec} X_{k+1} = \left\{ \sum_{r=1}^{R} \left[(\Psi_k^r)^T \otimes \Theta_k^r \right] \right\} \operatorname{vec} X_k + \operatorname{vec} W_k \tag{5.36}$$

where $\otimes$ denote the Kronecker product (see Definition D.1.3). Equation (5.36) is obtained using Proposition D.2.5 with

$$Y = X_{k+1}\ ,\ X = X_k\ ,\ A^u = \Theta_k^r\ ,\ B^u = \Psi_k^r\ ,\ u = r\ ,\ v = R$$

The second term in the right-hand side of Eq. (5.36) is due to the linearity property of the Vec-operator. Let define the $mn \times mn$ matrix Φ_k as

$$\Phi_k \triangleq \sum_{r=1}^{R} \Big[\big(\Psi_k^r \big)^T \otimes \Theta_k^r \Big] \tag{5.37}$$

and let $\mathbf{x}_{k+1} \in \mathbb{R}^{mn}$, $\mathbf{x}_k \in \mathbb{R}^{mn}$, and $\mathbf{w}_k\, \mathbb{R}^{mn}$ denote the Vec-transforms of X_{k+1}, X_k, and W_k, respectively. Using this notation Eq. (5.36) is rewritten as

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \tag{5.38}$$

Equation (5.38) is the process equation of the “Vec-system”. It is in the format of a standard vector state-space equation of dimension mn with $\mathbf{x}_k$ as a state vector, Φ_k as a transition matrix, and $\mathbf{w}_k$ as a Gaussian zero-mean white-noise input. Thus, the state estimate associated with the Time Update stage of the Kalman filter is [40, p. 228]

$$\widehat{\mathbf{x}}_{k+1/k} = \Phi_k \widehat{\mathbf{x}}_{k/k} \tag{5.39}$$

where $\widehat{\mathbf{x}}_{k+1/k} = \operatorname{vec} \widehat{X}_{k+1/k}$ and $\widehat{\mathbf{x}}_{k/k} = \operatorname{vec} \widehat{X}_{k/k}$. It is a matter of fact that Eq. (5.39) is equivalent to

$$\operatorname{vec} \widehat{X}_{k+1/k} = \left\{ \sum_{r=1}^{R} \Big[\big(\Psi_k^r \big)^T \otimes \Theta_k^r \Big] \right\} \operatorname{vec} \widehat{X}_{k/k} \tag{5.40}$$

where Φ_k has been replaced by its definition (Eq. 5.37). In order to recover the matrix format from Eq. (5.40), we apply Proposition D.2.5 again (but this time in the inverse direction) with

$$Y = \widehat{X}_{k+1/k} \;,\; X = \widehat{X}_{k/k} \;,\; A^u = \Theta_k^r \;,\; B^u = \Psi_k^r \;,\; u = r \;,\; v = R$$

As a result the Time Update stage for state matrix estimation is expressed as

$$\widehat{X}_{k+1/k} = \sum_{r=1}^{R} \Theta_k^r \, \widehat{X}_{k/k} \, \Psi_k^r \tag{5.41}$$

It can be seen from Eq. (5.41) that the a priori estimate at time t_{k+1} is computed using only the matrix-plant parameters (see Eq. 5.1) and the a posteriori matrix estimate at time t_k.

Reminding that the covariance matrix of the *state matrix* estimation error is defined as the covariance matrix of its Vec-transform (Definition D.1.4), the Time Update stage for the estimation error covariance is that of the mn-dimensional "Vec-system". That is, the a priori estimation error covariance matrix at time t_{k+1} is computed from [40, p. 229]

$$P_{k+1/k} = \Phi_k \, P_{k/k} \, \Phi_k^T + Q_k \tag{5.42}$$

where Φ_k is computed from Eq. (5.37).

5.3.2 Measurement Update Stage

Assume that $\widehat{X}_{k+1/k}$ and $P_{k+1/k}$ have been computed, and that a new matrix measurement, Y_{k+1}, is performed at time t_{k+1} with known measurement error covariance matrix, R_{k+1}. Operating on both sides of Eq. (5.2) with the Vec-operator and applying Proposition D.2.5 with

$$Y = Y_{k+1} \;,\; X = X_{k+1} \;,\; A^u = H^s_{k+1} \;,\; B^u = G^s_{k+1} \;,\; u = s \;,\; v = S$$

yields the pq-dimensional vector equation

$$\operatorname{vec} Y_{k+1} = \left\{ \sum_{s=1}^{S} \left[\left(G^s_{k+1} \right)^T \otimes H^s_{k+1} \right] \right\} \operatorname{vec} X_{k+1} + \operatorname{vec} V_{k+1} \tag{5.43}$$

Let $\mathbf{y}_{k+1}$ and $\mathbf{v}_{k+1}$ denote the Vec-transforms of Y_{k+1} and V_{k+1}, respectively, and let define the $pq \times mn$ matrix $\mathcal{H}_{k+1}$ as

$$\mathcal{H}_{k+1} = \sum_{s=1}^{S} \left[\left(G^s_{k+1} \right)^T \otimes H^s_{k+1} \right] \tag{5.44}$$

Then, Eq. (5.43) is rewritten as

$$\mathbf{y}_{k+1} = \mathcal{H}_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1} \tag{5.45}$$

Equation (5.45) is the measurement equation of the "Vec-system". It has the standard form of a measurement equation in a vector state-space model. Using the measurement model of Eq. (5.45) we apply the classical Kalman filter theory. Let the innovations sequence be denoted by $\widetilde{\mathbf{y}}_{k+1}$; it is computed from

$$\widetilde{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1} - \mathcal{H}_{k+1} \widehat{\mathbf{x}}_{k+1/k} \tag{5.46}$$

Defining the *matrix* innovations sequence, $\widetilde{Y}_{k+1}$, such that $\widetilde{\mathbf{y}}_{k+1} = \operatorname{vec} \widetilde{Y}_{k+1}$, and replacing $\mathcal{H}_{k+1}$ by its definition (Eq. 5.44) yields

$$\operatorname{vec} \widetilde{Y}_{k+1} = \operatorname{vec} Y_{k+1} - \left\{ \sum_{s=1}^{S} \left[\left(G^s_{k+1} \right)^T \otimes H^s_{k+1} \right] \right\} \operatorname{vec} \widehat{X}_{k+1/k} \tag{5.47}$$

Equation (5.47) is transformed into the equivalent matrix equation by applying Proposition D.2.5 with

$$Y = \widetilde{Y}_{k+1} \ , \ X = \widehat{X}_{k+1/k} \ , \ A^u = H^s_{k+1} \ , \ B^u = G^s_{k+1} \ , \ u = s \ , \ v = S$$

such that the expression for the matrix innovations sequence, $\widetilde{Y}_{k+1}$, is given by

$$\widetilde{Y}_{k+1} = Y_{k+1} - \sum_{s=1}^{S} H^s_{k+1} \widehat{X}_{k+1/k} G^s_{k+1} \tag{5.48}$$

The covariance matrix of $\widetilde{Y}_{k+1}$, denoted by S_{k+1}, is defined as the covariance matrix of $\widetilde{\mathbf{y}}_{k+1}$. Thus, from Kalman filter theory, S_{k+1} is computed from

$$S_{k+1} = \mathcal{H}_{k+1} P_{k+1/k} \mathcal{H}^T_{k+1} + R_{k+1} \tag{5.49}$$

Note that S_{k+1} is a $pq \times pq$ matrix. In order to update the state estimate of the mn-dimensional system, the $mn \times pq$ Kalman gain matrix is computed from [40, p. 246]

$$K_{k+1} = P_{k+1/k} \mathcal{H}^T_{k+1} S^{-1}_{k+1} \tag{5.50}$$

and the mn-dimensional state estimate is updated as follows

$$\widehat{\mathbf{x}}_{k+1/k+1} = \widehat{\mathbf{x}}_{k+1/k} + K_{k+1} \widetilde{\mathbf{y}}_{k+1} \tag{5.51}$$

Equation (5.51) can be rewritten as

$$\operatorname{vec} \widehat{X}_{k+1/k+1} = \operatorname{vec} \widehat{X}_{k+1/k} + K_{k+1} \operatorname{vec} \widetilde{Y}_{k+1} \tag{5.52}$$

Consider the matrix K_{k+1} as a $mn \times pq$ array of nq submatrices of dimension $m \times p$. Let K^{jl} denote the $m \times p$ submatrix of K_{k+1} at position (jl). Then, applying Proposition D.2.7 to the second term in the right-hand side of Eq. (5.52) with

$$A = K_{k+1} \ , \ \mathbf{x} = \operatorname{vec} \widetilde{Y}_{k+1}$$

yields

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \sum_{j=1}^{n} \sum_{l=1}^{q} K^{jl}_{k+1} \widetilde{Y}_{k+1} E^{lj} \tag{5.53}$$

Equation (5.53) is the Measurement Update equation for the matrix state estimate at time t_{k+1}. The covariance matrix of the a posteriori estimation error at time t_{k+1} is

$$P_{k+1/k+1} = \left(I_{mn} - K_{k+1} \mathcal{H}_{k+1}\right) P_{k+1/k} \left(I_{mn} - K_{k+1} \mathcal{H}_{k+1}\right)^T + K_{k+1} R_{k+1} K^T_{k+1} \tag{5.54}$$

as known from the classic Kalman filter [40, p. 245].

5.4 Numerical Examples

5.4.1 Example 1

In this example we deal with the estimation of a 2 × 2 state-matrix from 2 × 2 matrix measurements. The system is described by the following process and measurement equations:

$$X_{k+1} = \begin{bmatrix} 0.5 & 0.1 \\ 0 & 0.5 \end{bmatrix} X_k \begin{bmatrix} 0.8 & -1 \\ 0.3 & 0.8 \end{bmatrix} + W_k \tag{5.55}$$

$$Y_{k+1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} X_{k+1} \begin{bmatrix} 1 & 2 \\ -1 & 1 \end{bmatrix} + V_{k+1} \tag{5.56}$$

The initial state X_0 is Gaussian with mean $\bar{X}_0 = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$ and covariance $\Pi_0 = 0.2\, I_4$. The process noise W_k is a Gaussian zero-mean white sequence with covariance matrix $Q_k = 10^{-2} I_4$. The measurement noise V_k is a Gaussian zero-mean white sequence with covariance matrix $R_k = I_4$.

The results of the simulation for $\widehat{X}_0 = O_4$ are summarized in Figures 5.1 and 5.2. Figure 5.1 shows a sample run of the four state estimation errors. Figure 5.2 depicts the results of 100 Monte-Carlo runs as well as the predicted filter performances. As expected from a converging Kalman filter, the MC-mean almost coincides with 0 in the steady-state and the filter is unbiased. Moreover, the Monte-Carlo and the filter 1σ-envelopes are very close, which shows the consistency of the filter estimation process.

5.4.2 Example 2

This example illustrates the ability of the matrix Kalman filter to estimate a 2 × 3 matrix of constant unknown parameters from a sequence of 2 × 1 vector measurements. The measurement equation is described by:

$$\mathbf{y}_k = X\, \boldsymbol{g}_k + \mathbf{v}_k \tag{5.57}$$

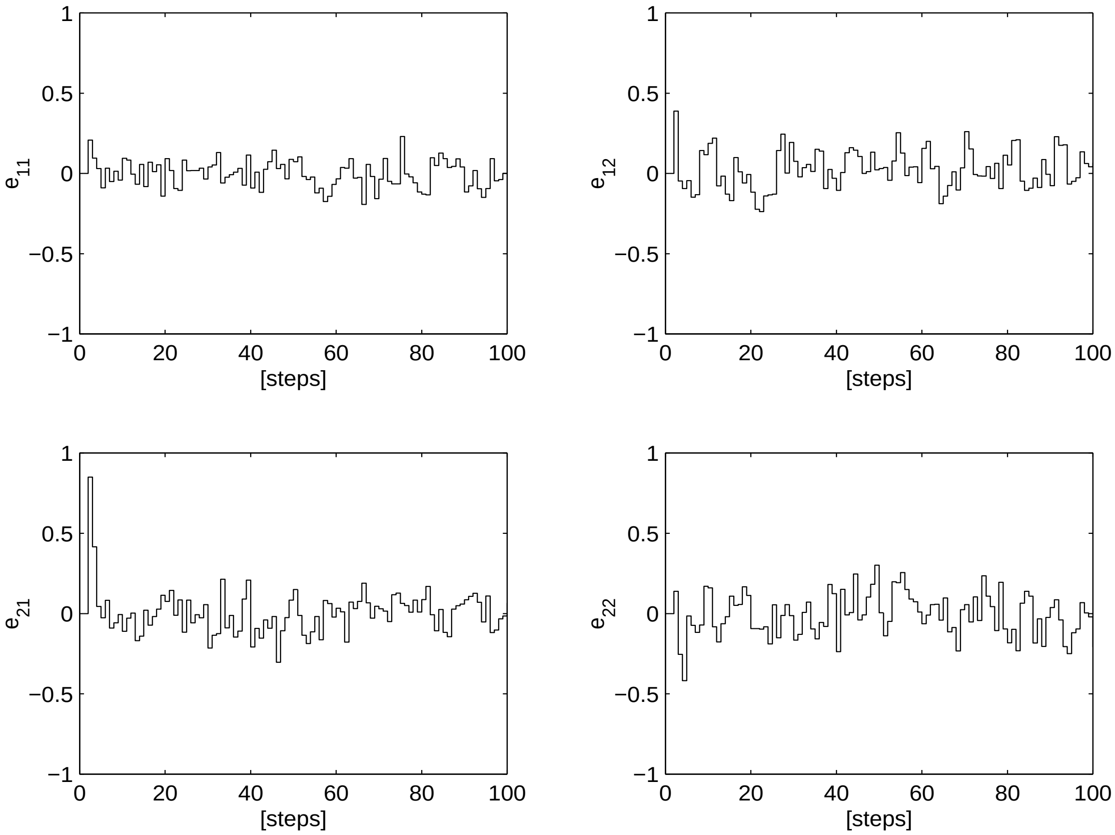


Fig. 5.1: Example 1 - Estimation errors for a sample run.

The matrix of unknown parameters is the 2×3 matrix

$$X = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix} \tag{5.58}$$

The measurement column-matrix $\boldsymbol{g}_k$ has the following vector values, which are repeated periodically

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \; , \; \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \; and \; \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.59}$$

The measurement noise vector $\mathbf{v}_k$ is a Gaussian zero-mean white sequence with covariance matrix $R_k = 10^{-2}$ diag$(2, 4)$.

The simulation results for $\widehat{X}_0 = O_{2,3}$ are summarized in Figures 5.3 and 5.4. Figure 5.3 shows the time variations of the six state estimates. They all converge to the correspond-

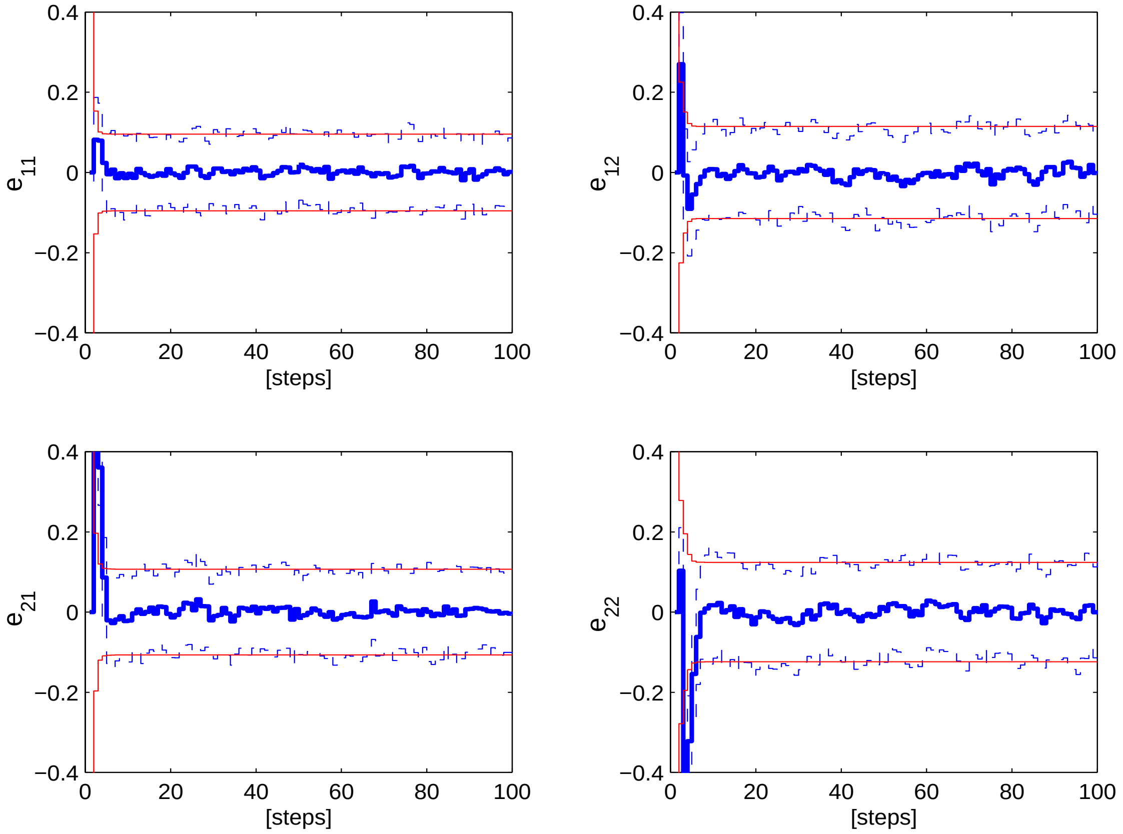


Fig. 5.2: Example 1 - Monte-Carlo and Filter Means and $\pm 1\sigma$-envelopes of the estimation errors for 100 runs - Blue thick line: mean, blue dotted line: $\pm 1\sigma$, red thin line: Filter $\pm 1\sigma$.

ing true constant parameters within approximately 300 steps. A 100-runs Monte-Carlo simulation was performed. The estimation errors results are depicted in Figure 5.4. In this example, as in Example 1, the MC simulation clearly shows the unbiasedness and the consistency of the filter.

5.4.3 Example 3

Introduction

In this example we consider the application of the matrix Kalman filter as a "prefilter" of vector measurements in a spacecraft attitude determination problem. Consider the case where vector measurements are simultaneously acquired. As will be shown, stacking

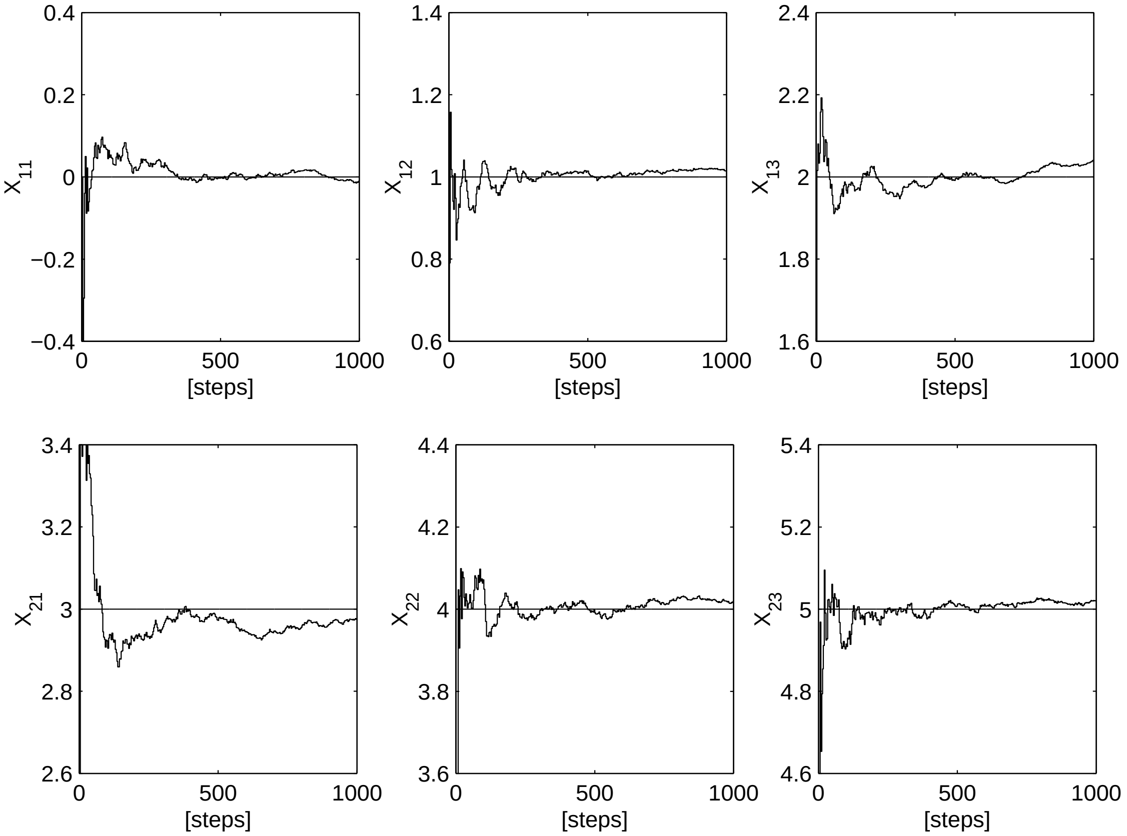


Fig. 5.3: Example 2 - Estimated matrix state elements for a sample run.

them one near the other, as columns of a matrix, yields a *matrix measurement*. The idea underlying the present example is that prefiltering this data by processing the matrix measurements in a MKF enhances the attitude estimator performance. For this purpose, two steps are necessary. In the first step, the data is modelled as a state-matrix of a time-varying matrix plant, and the MKF is applied to estimate that state-matrix. In the second step, the attitude is computed from the estimated state matrix using known methods (see e.g. [24]). For the sake of brevity of the example and relevance to the MKF algorithm we focus on the prefiltering operation (first step).

Model

In this section we develop the linear time-varying discrete-time matrix model of the matrix system of interest. The vector measurement model is described by the following

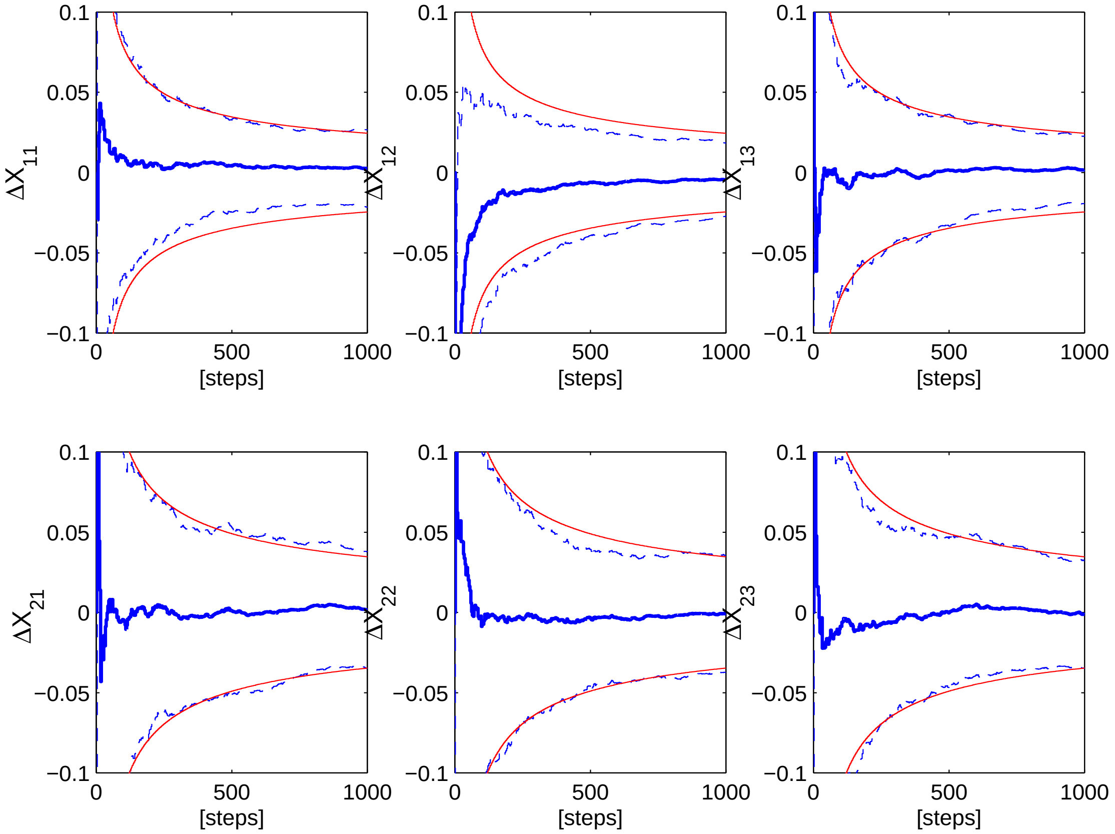


Fig. 5.4: Example 2 - Monte-Carlo and Filter Means and ±1σ-envelopes of the estimation errors for 100 runs - Blue thick line: mean, blue dotted line: ±1σ, red thin line: Filter ±1σ.

equation

$$\mathbf{b}_k = \mathbf{b}^o_k + \mathbf{v}_k \tag{5.60}$$

where the 3×1 unit vector $\mathbf{b}_k$ contains the noisy body components of the measured direction, the 3×1 unit vector $\mathbf{b}^o_k$ contains the error-free body components of this direction, and $\mathbf{v}_k$ is the measurement noise vector. Assuming that a set of N simultaneous observations is acquired, the N vector equations of type Eq. (5.60) are combined in a $3 \times N$ matrix equation as follows.

$$\begin{bmatrix} \mathbf{b}_{k,1} & \mathbf{b}_{k,2} & \cdots & \mathbf{b}_{k,N} \end{bmatrix} = \begin{bmatrix} \mathbf{b}^o_{k,1} & \mathbf{b}^o_{k,2} & \cdots & \mathbf{b}^o_{k,N} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{k,1} & \mathbf{v}_{k,2} & \cdots & \mathbf{v}_{k,N} \end{bmatrix} \tag{5.61}$$

Let Y_k denote the $3 \times N$ matrix of the vector measurements on the left-hand side of Eq. (5.61), X_k denote the $3 \times N$ matrix of the error-free body components of the N vector measurements, and V_k denote the $3 \times N$ noise matrix on the right-hand side of Eq. (5.61). Then Eq. (5.61) is rewritten as

$$Y_k = X_k + V_k \tag{5.62}$$

Equation (5.62) has the standard expression for a matrix measurement model, which can be implemented in the matrix Kalman filter. *The state matrix, X_k, is thus defined as the $3 \times N$ matrix whose columns are the true vector observations.* The approach for deriving the process equation is outlined in the following. Let A_k denote the attitude matrix[1] at time t_k, and let $\mathbf{r}$ denote the 3×1 unit column-matrix that is the decomposition of the observed star direction in an inertial reference Cartesian coordinate frame. Then $\mathbf{b}^o_k$, A_k, and $\mathbf{r}$ are related by

$$\mathbf{b}^o_k = A_k \, \mathbf{r} \tag{5.63}$$

While the body (spacecraft) coordinate frame can have arbitrary kinematics, the rate of change of the reference vector, $\mathbf{r}$, is here assumed to be negligible[2]. Thus, the time subscript in $\mathbf{r}$ is dropped. The kinematics law of a rigid body in terms of the attitude matrix is described by the well-known difference equation [1, p. 512]

$$A_{k+1} = \Theta^o_k \, A_k \tag{5.64}$$

where the transition matrix Θ^o_k is a 3×3 orthogonal matrix function of the body-resolved angular velocity vector, $\boldsymbol{\omega}^o_k$. When the measurement of the *same* direction is acquired at t_{k+1}, then in analogy with Eq. (5.63) we write

$$\mathbf{b}^o_{k+1} = A_{k+1} \, \mathbf{r} \tag{5.65}$$

Inserting Eq. (5.64) into Eq. (5.65) and using Eq. (5.63) in the resulted equation yields

$$\begin{aligned} \mathbf{b}^o_{k+1} &= \left(\Theta^o_k \, A_k \right) \mathbf{r} && (5.66) \\ &= \Theta^o_k \, (A_k \, \mathbf{r}) && (5.67) \\ &= \Theta^o_k \, \mathbf{b}^o_k && (5.68) \end{aligned}$$

[1] The attitude matrix, also called Direction Cosine Matrix, is the rotation matrix that brings the axes of the spacecraft Cartesian coordinate frame onto the axes of some reference Cartesian coordinate frame.

[2] This is for instance the case when stars are the observe outputs because stars are assumed to be fixed in an inertial reference frame.

When N simultaneous vector measurements are acquired at t_k, then N equations of type Eq. (5.68) are simultaneously written as follows

$$\begin{bmatrix} \mathbf{b}^o_{k+1,1} & \mathbf{b}^o_{k+1,2} & \cdots & \mathbf{b}^o_{k+1,N} \end{bmatrix} = \begin{bmatrix} \Theta^o_k \, \mathbf{b}^o_{k,1} & \Theta^o_k \, \mathbf{b}^o_{k,2} & \cdots & \Theta^o_k \, \mathbf{b}^o_{k,N} \end{bmatrix} \tag{5.69}$$

Note that at each time point we observe the same N directions in space. On the left-hand side of Eq. (5.69), we identify as X_{k+1} the state-matrix at time t_{k+1}. Factoring out Θ^o_k in the right-hand side of Eq. (5.69) leads to the state-matrix dynamics equation

$$X_{k+1} = \Theta^o_k \, X_k \tag{5.70}$$

Note that Eq. (5.70) is a deterministic equation, which requires the knowledge of the angular velocity $\boldsymbol{\omega}^o_k$. In the lack of the exact angular velocity, we customarily substitute the *measured* angular velocity, $\boldsymbol{\omega}_k$, for the true angular velocity, $\boldsymbol{\omega}^o_k$, in the expression for Θ^o_k, which is then denoted by Θ_k. This leads to the model of Eq. (5.71) where the noise in the gyro outputs produces a noise term W_k.

$$X_{k+1} = \Theta_k \, X_k + W_k \tag{5.71}$$

Henceforth, the state matrix X_k is modelled as a stochastic process whose dynamics is governed by the difference equation (5.71). The covariance matrices of the process noise and of the measurement noise denoted, respectively, by Q_k and R_k, are usually determined by filter tuning based on Monte-Carlo simulations. Here though analytical expressions for Q_k and R_k are proposed. They provide good initial values before adjusting them when tuning the filter. The developments of these expressions are straightforward and are given without proof. The matrix state-space dynamics and measurement models are summarized as follows:

$$X_{k+1} = \Theta_k \, X_k + W_k \tag{5.72}$$

$$Y_{k+1} = X_{k+1} + V_{k+1} \tag{5.73}$$

where Θ_k is computed from

$$[\boldsymbol{\omega}_k \times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{5.74}$$

$$\Theta_k = \exp\{- \; [\boldsymbol{\omega}_k \times] \; \Delta t\} \tag{5.75}$$

where $\boldsymbol{\omega}_k$ is the angular velocity vector at time point t_k, and ω_i, $i = 1, 2, 3$, are its components in body axes, and Δt is the time increment; that is, $\Delta t = t_{k+1} - t_k$. For the case of two simultaneous vector observations (N=2), R_k, the 6×6 covariance matrix of V_k is computed as

$$R_k = \begin{bmatrix} R_{k,1} & O_{3\times3} \\ O_{3\times3} & R_{k,2} \end{bmatrix} \tag{5.76}$$

where $R_{k,1}$ and $R_{k,2}$ are, respectively, the covariance matrices of the measurement errors in the first and second vector observations. The gyros measure the angular velocity vector, $\boldsymbol{\omega}_k$, with an additive error, $\boldsymbol{\epsilon}_k$, which is modelled as a zero-mean white noise sequence with a known covariance matrix, Q^ϵ_k. The 6×6 covariance matrix of W_k, Q_k, is computed as follows.

$$T_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \qquad T_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad T_3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.77}$$

$$T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{5.78}$$

$$Q'_k = T\, Q^\epsilon_k\, T^T \tag{5.79}$$

$$\Gamma_k = \Delta t\, \left[(\widehat{X}_{k/k})^T \otimes I_3 \right] \tag{5.80}$$

$$Q_k = \Gamma_k\, Q'_k\, \Gamma^T_k \tag{5.81}$$

where $\widehat{X}_{k/k}$ denotes the a posteriori estimate of X_k at t_k. Note that the filter covariance computation is state-dependent due to the use of $\widehat{X}_{k/k}$ in the expression for Q_k.

Simulation Study

The tested scenario is of a spin-stabilized spacecraft that undergoes nutation. The spin velocity is 0.464 *rpm*, the nutation rate is 1 *rph*, and the nutation angle is 22.5^o. The *same* two directions are simultaneously observed at each sampling time. This is the case of a star-tracker tracking the same stars all the time. The reference vector-components of

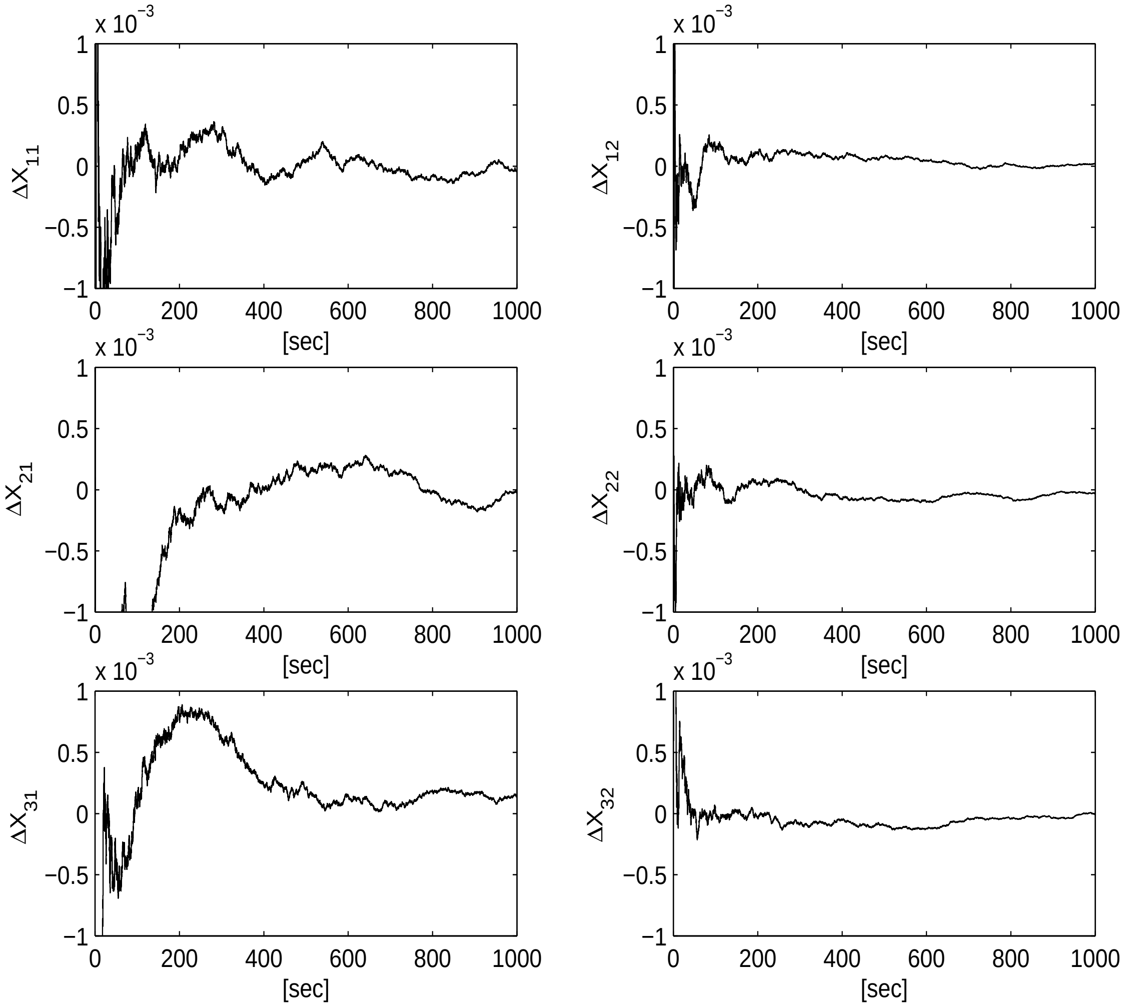


Fig. 5.5: Example 3 - Sample run - Time histories of the estimation errors.

the two observed directions are chosen as

$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{r}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.82}$$

The vector observation noises are assumed to be zero-mean, white sequences with covariance matrices $R_{k,1} = 4 \cdot 10^{-4} I_3[\text{rad}^2]$ and $R_{k,2} = 4 \cdot 10^{-5} I_3[\text{rad}^2]$. As a special case, the gyro noise is assumed to be a zero-mean white sequence with covariance matrix $Q_k^\epsilon = (0.01 \text{ deg/hr})^2 I_3$. The initial value of the state, X_0, is a Gaussian matrix random

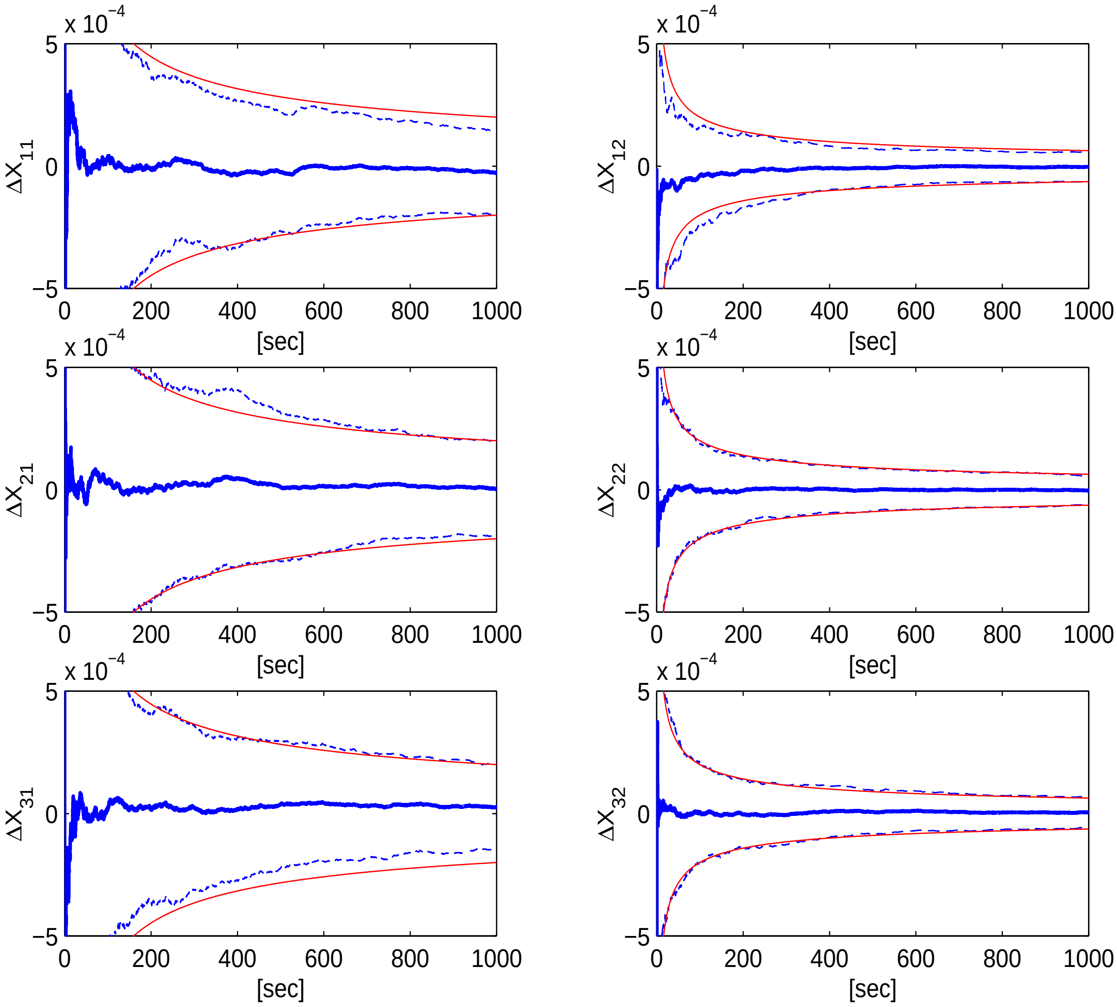


Fig. 5.6: Example 3 - Monte-Carlo simulation - Mean and $\pm 1\,\sigma$-envelope of the estimation errors. Blue thick line: mean, blue dotted line: MC 1σ, red thin line: Filter 1σ.

variable with mean $\bar{X}_0$ and covariance matrix Π_0 given as

$$\bar{X}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.83}$$

$$\Pi_0 = 0.2\, I_6 \tag{5.84}$$

where I_6 denotes the 6×6 identity matrix. The vector measurement noises, the gyro noise, and the initial state are assumed to be uncorrelated with one another. The MKF algorithm is initialized with $\widehat{X}_0 = O_{3\times 2}$, and $P_{0/0} = 5\,I_6$.

Figures 5.5 and 5.6 summarize the simulation results. Figure 5.5 presents a sample run of the time histories of the estimation errors. The left-hand column of plots corresponds to the first observed direction, while the plots in the right-hand column correspond to the second direction. It can be seen that the estimation errors are convergent. Because of the higher accuracy in the second vector observation, the estimator performance is better in the right-hand column, e.g., the transition phases are shorter and the steady-state values are smaller. This analysis is confirmed when looking at the 100-runs Monte-Carlo simulation results (Fig. 5.6). The 1σ-envelopes are clearly narrower in the right-hand column than in the left-hand column. Moreover, Fig. 5.6 shows that there is consistency between the actual and the predicted value of the filter performance. Note that this happens in spite of the state-dependence introduced in the filter covariance computation (see expression for Q_k Eqs. 5.77 to 5.81).

5.5 Concluding Remarks

It has been shown that discrete-time systems, which are described by linear matrix difference equations and are observed through linear matrix measurement equations, can be estimated by a matrix version of the ordinary Kalman filter. It has been proved that the matrix Kalman filter is a natural and straightforward generalization of the ordinary Kalman filter, and includes, as special cases, other matrix filters that were developed in the past. The proposed estimation algorithm uses a compact matrix notation to produce the matrix estimate and the estimation error covariance matrix in terms of the original plant matrices. The validity and efficiency of the matrix Kalman filter was demonstrated by means of Monte-Carlo simulations for three estimation problems.

Chapter 6

Quaternion Estimation from Vector Observations Using a Matrix Kalman Filter

6.1 Introduction

The Optimal-REQUEST algorithm, which was introduced in Chapter 3, is an optimized REQUEST algorithm operating in a two-stage mode. The first stage is a linear optimal filter of the K-matrix, and the second stage consists of a **q**-method-based procedure where **q** is extracted from the filtered K-matrix. The first stage was developed using state-space modelling techniques and a Kalman filter strategy to optimize the fading memory factor of REQUEST. It was shown that Optimal-REQUEST performed well under nominal conditions, and adaptive filtering techniques were devised to cope with modelling errors, like constant bias in the gyro outputs. However, the performance of that algorithm were limited by two facts: the optimal gain was a scalar, and the estimation performance index was more conservative than the usual estimation error variance.

In the present work, in order to enhance the performance of the Optimal-REQUEST algorithm, we replace its first stage with a matrix Kalman filter (MKF) of the K-matrix. Since the MKF, as developed in Chapter 5, implements a matrix gain and is a minimum-variance estimator, we eliminate the two above-mentioned deficiencies of Optimal-REQUEST. Besides, using a matrix Kalman filter is adequate for the K-matrix plant, since that plant is described by linear matrix equations. A pseudo-measurement technique, whose concept is presented in [28] for quaternion normalization in an EKF, is proposed here to enforce the symmetry and zero-trace constraints upon the estimated K-matrix. Note that the linearity of the constraints preserve the linearity of the filter. As mentioned before, the K-matrix MKF serves as the first stage of the proposed quaternion estimator. As usual, the second stage of the estimator consists of the extraction of the quaternion, whenever it is needed, from the updated K-matrix.

The development of the MKF algorithm for quaternion attitude determination is presented as follows. Section 6.2 briefly reviews the REQUEST and the Optimal-REQUEST algorithms. Section 6.3 briefly reviews the state-space model on which the MKF algorithm can operate. Section 6.4 presents the development of the mathematical model for the K-matrix system. The stochastic models for the sensor noises (gyros and vector measurement sensors) are described, and analytical approximations to the system noise covariance matrices are developed. Issues related to the implementation of the filter are addressed in section 6.5. The special issue of linearly constrained estimation is addressed in section 6.6 using the pseudo-measurement technique, which allows the formulation of a linearly constrained Kalman filter (CMKF). Section 6.7 presents a numerical study where the performance of the MKF is compared, by means of Monte-Carlo simulations, to that of Optimal-REQUEST and of CMKF. Section 6.8 investigates the conditions under which the MKF of the K-matrix reduces to the Optimal-REQUEST algorithm. The conclusions are presented in the last section.

6.2 Preliminaries

6.2.1 The REQUEST Algorithm

The REQUEST algorithm [24] performs recursive estimation of time-varying attitude. It essentially propagates and updates the K-matrix. The propagation of the K-matrix, $K_{k/k}$, from t_k to t_{k+1}, which yields $K_{k+1/k}$, is given by

$$K_{k+1/k} = \Phi_k K_{k/k} \Phi_k^T \tag{6.1}$$

where Φ_k is the transition matrix in the difference equation that governs the following discrete-time dynamics of the quaternion [1, p. 564]

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k \tag{6.2}$$

The transition matrix, Φ_k, for a piecewise constant angular velocity vector, $\boldsymbol{\omega}_k$, over time intervals of span $\Delta t = t_{k+1} - t_k$, is given by

$$\Phi_k = e^{\frac{1}{2}\,\Omega_k\,\Delta t} \tag{6.3}$$

where [1, p. 559]

$$\Omega_k = \begin{bmatrix} -\,[\boldsymbol{\omega}_k \times] & \boldsymbol{\omega}_k \\ -\boldsymbol{\omega}_k^T & 0 \end{bmatrix} \tag{6.4}$$

and $\boldsymbol{\omega}_k$ is the gyro measured angular velocity resolved in body coordinates. Given a single observation[1], $\mathbf{b}_{k+1}$, $\mathbf{r}_{k+1}$, at time t_{k+1}, one can construct the corresponding incremental K-matrix, denoted by δK_{k+1}, as follows

$$\delta K_{k+1} = \begin{bmatrix} S_{k+1} - \sigma_{k+1} I_3 & \mathbf{z}_{k+1} \\ \mathbf{z}_{k+1}^T & \sigma_{k+1} \end{bmatrix} \tag{6.5}$$

where

$$\begin{aligned} B_{k+1} &= \mathbf{b}_{k+1}\,\mathbf{r}_{k+1}^T & \qquad S_{k+1} &= B_{k+1} + B_{k+1}^T \\ \mathbf{z}_{k+1} &= \mathbf{b}_{k+1} \times \mathbf{r}_{k+1} & \qquad \sigma_{k+1} &= tr\big(B_{k+1}\big) \end{aligned} \tag{6.6}$$

The update stage of REQUEST is formulated as

$$K_{k+1/k+1} = \frac{\rho_k m_k}{m_{k+1}}\,K_{k+1/k} + \frac{1}{m_{k+1}}\,\delta K_{k+1} \tag{6.7}$$

where the sequence of scalars m_k, $k = 1, 2, \ldots$, aims at keeping the largest eigenvalue of $K_{k+1/k+1}$ close to 1. The coefficient ρ_k is a fading memory factor, which is equal to 1 if Φ_k is error-free, and is otherwise set between 0 and 1 according to the propagation noise. *The heuristic choice of ρ_k makes the filter REQUEST suboptimal.*

[1] Reference [24] treated the general case of multiple observations at time t_{k+1}. The case of a single observation is presented here for the sake of simplicity

6.2.2 The Optimal-REQUEST Algorithm

Optimal-REQUEST, as presented in Chapter 3, optimizes the coefficient ρ_k in Eq. (6.7) according to a special estimation cost function. The central idea in the development of that filter is that REQUEST is considered an estimator *of the K-matrix.* That is, each K-matrix, $K_{i/j}$, is considered the *estimate* of a true K-matrix, $K^o_{i/j}$. The true matrix $K^o_{i/j}$ is related to the true attitude at t_i, and is constructed using the ideal error-free vector measurements up to t_j. Doing so, an estimation error, ΔK_{ij}, is defined as $\Delta K_{ij} \overset{\triangle}{=} K^o_{i/j} - K_{i/j}$. Similarly, the matrix δK_i, which is computed using the *noisy* measurements acquired at time t_i, is the measurement of the true matrix δK^o_i, which is constructed using the ideal noise-free measurements at time t_i. The K-matrix dynamics and measurement models are described by the following equations.

$$K^o_{k+1/k} = \Phi_k K^o_{k/k} \Phi^T_k + W_k \tag{6.8}$$

$$\delta K_{k+1} = \delta K^o_{k+1} + V_{k+1} \tag{6.9}$$

where W_k and V_{k+1} are the process noise and measurement noise, respectively. Assuming that the vector measurement at t_{k+1} is affected by an additive error, $\boldsymbol{\delta b}_{k+1}$, it is shown in Chapter 3 that V_{k+1} has the following expression

$$V_{k+1} = \begin{bmatrix} \mathcal{S}_b - \sigma_b I_3 & \mathbf{z}_b \\ \mathbf{z}^T_b & \sigma_b \end{bmatrix} \tag{6.10}$$

where

$$\begin{aligned} B_b &= \boldsymbol{\delta b}_{k+1} \mathbf{r}^T_{k+1} & \mathcal{S}_b &= B_b + B_b{}^T \\ \mathbf{z}_b &= \boldsymbol{\delta b}_{k+1} \times \mathbf{r}_{k+1} & \sigma_b &= tr(B_b) \end{aligned} \tag{6.11}$$

Assuming that the measured angular velocity, $\boldsymbol{\omega}_k$, has an additive error, $\boldsymbol{\epsilon}_k$, a good approximation to the noise matrix W_k is given by

$$W_k \simeq \begin{bmatrix} S_\epsilon - \sigma_\epsilon I_3 & \mathbf{z}_\epsilon \\ \mathbf{z}^T_\epsilon & \sigma_\epsilon \end{bmatrix} \Delta t \tag{6.12}$$

where

$$\begin{aligned} B_\epsilon &= [\boldsymbol{\epsilon}_k \times]\, B_{k/k} & S_\epsilon &= B_\epsilon + B_\epsilon{}^T \\ \mathbf{z}_\epsilon &= B_\epsilon{}^T - B_\epsilon & \sigma_\epsilon &= tr(B_\epsilon) \end{aligned} \tag{6.13}$$

and $B_{k/k}$ is computed using the value of $K_{k/k}$, and the relationships described in Eqs. (3.3) and (3.4). A particular measure of uncertainty is introduced in order to model and quantify the uncertainty of matrix processes, such as the noises W_k and V_{k+1}, and the updated estimation error, $\Delta K_{k/k}$. Similarly to the classical Kalman filter, the optimal value of the gain, ρ_k^*, is obtained by minimizing that uncertainty in the updated estimation error. Optimal-REQUEST is a two-stage estimator of the attitude where the first stage consists of a recursive linear estimator of a time varying K-matrix, and the second stage consists of the computation of the optimal quaternion, subject to the unity constraint on the quaternion length. As mentioned before, the shortcomings of Optimal-REQUEST are that the measure of uncertainty is more conservative than the classical trace of the covariance matrix, and the gain is limited to be scalar.

6.3 The General Matrix State-Space Model

The matrix Kalman filter, developed in Chapter 5, can handle plants that are described by the following linear matrix equations

$$X_{k+1} = \sum_{r=1}^{\mu} \Theta_k^r \, X_k \, \Psi_k^r + W_k \tag{6.14}$$

$$Y_{k+1} = \sum_{s=1}^{\nu} H_{k+1}^s \, X_{k+1} \, G_{k+1}^s + V_{k+1} \tag{6.15}$$

where $X_k \in \mathbb{R}^{m \times n}$ is the state matrix, $\Theta_k^r \in \mathbb{R}^{m \times m}$, and $\Psi_k^r \in \mathbb{R}^{n \times n}$, $r = 1, 2, \ldots, \mu$, are transition matrices, $W_k \in \mathbb{R}^{m \times n}$ is the process noise; the matrix $Y_{k+1} \in \mathbb{R}^{p \times q}$ is the measurement, $H_{k+1}^s \in \mathbb{R}^{p \times m}$, and $G_{k+1}^s \in \mathbb{R}^{n \times q}$, $s = 1, 2, \ldots, \nu$, are measurement matrices, and $V_{k+1} \in \mathbb{R}^{p \times q}$ is the measurement noise. The scalars μ and ν are problem-dependent. The usual assumptions concerning the noise stochastic models are adopted. That is, the system noises, W_k and V_k, are zero-mean white Gaussian sequences; they are uncorrelated with one another, and uncorrelated with the initial state X_0. Also, the covariances of the noises are known. The covariance of a matrix sequence, say U_k, is defined here as the covariance of its vec-transform, denoted by $\mathrm{vec}\,(U_k)$, where vec is the vec-operator (see Appendix D.1.2). The vec-operator operates on an arbitrary matrix, $M \in \mathbb{R}^{m \times n}$, by stacking the columns of M one over the other, and returning the mn-dimensional column-vector, $\mathrm{vec}\,(M)$.

The matrix Kalman filter combines the statistical properties of an ordinary Kalman filter with the advantage of a compact notation. It produces a Kalman filter matrix estimate in terms of the original plant matrices. The algorithm, its proof, and application examples are presented in Chapter 5.

6.4 Mathematical Model

In this section the state-space model equations of the K-matrix system are formulated, and explicit expressions for the system noise covariance matrices are provided.

6.4.1 Process Equation

The process equation is the first-order stochastic matrix equation that governs the dynamics of the true K-matrix. In order to work with the usual symbols, the true K-matrix, which is the state variable, is denoted by the symbol X; thus, X_k denotes the true K-matrix at time t_k. Substituting in Eq. (6.8) X_k and X_{k+1} for $K^o_{k/k}$ and $K^o_{k+1/k}$, respectively, yields

$$X_{k+1} = \Phi_k \, X_k \, \Phi_k^T + W_k \tag{6.16}$$

Equation (6.16) is a special case of the general process equation described in Eq. (6.1). Indeed, substituting in Eq. (6.1) Φ_k and Φ_k^T for Θ_k^r and Ψ_k^r, respectively, and taking $\mu = 1$, yields Eq. (6.16). Therefore, Eq. (6.16) is readily implementable in a matrix Kalman filter.

The expression for the process noise matrix, W_k, in Eq. (6.16) is given in Eqs. (6.12) and (6.13). As shown in Appendix E.1, an equivalent expression for W_k is

$$W_k = (\, X_k \, \mathcal{E}_k - \mathcal{E}_k \, X_k) \, \Delta t \tag{6.17}$$

where X_k is the state matrix at t_k, the time increment is denoted by Δt, and $\mathcal{E}_k$ is the following 4×4 skew-symmetric matrix

$$\mathcal{E}_k = \frac{1}{2} \begin{bmatrix} -\,[\boldsymbol{\epsilon}_k \times] & \boldsymbol{\epsilon}_k \\ -\,\boldsymbol{\epsilon}_k^T & 0 \end{bmatrix} \tag{6.18}$$

The 3×1 vector $\boldsymbol{\epsilon}_k$ denotes the additive error in the measured value of the body angular velocity vector. The reason for using Eqs. (6.17), (6.18) instead of Eqs. (6.12), (6.13), is that it facilitates the analysis of the stochastic model for W_k. In this work we consider the special case where $\boldsymbol{\epsilon}_k$ is a zero-mean white noise process with covariance matrix Q_k^ϵ; that is,

$$E\{\boldsymbol{\epsilon}_k\} = \mathbf{0} \tag{6.19a}$$

$$E\left\{\boldsymbol{\epsilon}_k\,\boldsymbol{\epsilon}_l^T\right\} = Q_k^\epsilon\,\delta_{kl} \tag{6.19b}$$

where δ_{kl} is the Kronecker delta. Moreover, it is assumed that $\boldsymbol{\epsilon}_k$ is uncorrelated with the initial state, X_0. Noting that the well-known state augmentation technique [40, p. 350] can easily handle the case of biased gyros, we use this basic model in order to focus on the attitude estimation problem. It can be shown that W_k is a zero-mean process. Using Eq. (6.19) and the expression for W_k given in Eq. (6.17) yields

$$\begin{aligned}\left(\frac{1}{\Delta t}\right) E\{W_k\} &= E\{\mathcal{E}_k\,X_k - X_k\,\mathcal{E}_k\} \\ &= E\{\mathcal{E}_k\,X_k\} - E\{X_k\,\mathcal{E}_k\} \\ &= O_4\end{aligned} \tag{6.20}$$

The second equality in Eq. (6.20) arises from the linearity property of the expectation operator, while the third equality is the result of the Markoff property of the process X_k, together with the assumptions that $\boldsymbol{\epsilon}_k$ is white and is uncorrelated with the initial state. Taking advantage of the linear structure of W_k, with respect to $\boldsymbol{\epsilon}_k$, we seek an analytic expression for its covariance matrix, Q_k. The fact of the state-dependence of W_k is handled by replacing the state with its best available estimate. In the present case, X_k is replaced by the updated estimate $\widehat{X}_{k/k}$. The computation of Q_k is summarized in the following. The detailed proof of the validity of this computation is given in Appendix E.2. Compute off-line the following 16 × 3 matrix M

$$M^T = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6.21}$$

Given $\widehat{X}_{k/k}$ and Q_k^ϵ at time t_k, compute the 16 × 3 matrix Γ_k

$$\Gamma_k = \left(\frac{\Delta t}{2}\right)\left[\left(I_4 \otimes \widehat{X}_{k/k}\right) - \left(\widehat{X}_{k/k}^T \otimes I_4\right)\right] M \tag{6.22}$$

where $\otimes$ denotes the Kronecker product [58, p.243], then compute

$$Q_k = \Gamma_k Q_k^\epsilon \Gamma_k^T \tag{6.23}$$

where Q_k^ϵ denotes the covariance matrix of $\boldsymbol{\epsilon}_k$.

6.4.2 Measurement Equation

The state matrix is measured at each sampling time, t_{k+1}. The measurement model is that of Optimal-REQUEST, Eq. (6.9); that is

$$Y_{k+1} = X_{k+1} + V_{k+1} \tag{6.24}$$

where Y_{k+1} is the measured K-matrix constructed using the noisy vector observations acquired at time t_{k+1}, and V_{k+1} denotes the measurement noise. However, unlike Optimal-REQUEST, we assume here that *the same directions* are observed at all the sampling times, $t_0, t_1, \ldots, t_{k+1}, \ldots$. This is necessary since X_{k+1} is the result of the time-propagation of X_0, and X_0 is constructed using the ideal noise-free vector observations at t_0. To clarify the latter fact consider the case of ideal noise-free matrix measurements of a static system. The state matrix, thus, is constant, and so must be the measurement matrix since it measures exactly the state. The meaning of this is that the same vector measurement must be acquired at each sampling time in order to be consistent with the proposed state-space model. Looking at the general measurement model described in Eq.(6.15) one realizes that Eq. (6.24) is a particular case of Eq. (6.15) where $H_{k+1}^s = G_{k+1}^s = I_4$ and $\nu = 1$. Thus Eq. (6.24) is readily implementable in the matrix Kalman filter.

Given a batch of vector observations, $\mathbf{b}_i$, $\mathbf{r}_i$, $i = 1, 2, \ldots, m$ ($m > 1$), acquired at t_{k+1}, the expression for the noise matrix V_{k+1} in the measurement equation, (6.24) is

$$V_{k+1} = \sum_{i=1}^{m} \alpha_i V_{k+1}^i \tag{6.25}$$

where $\alpha_i \triangleq a_i / \sum_{i=1}^m a_i$. Each matrix V_{k+1}^i in Eq. (6.25) is expressed using a pair, $\boldsymbol{\delta b}_i(t_{k+1})$, $\mathbf{r}_i(t_{k+1})$ according to Eqs. (6.10) and (6.11), where the 3×1 vector $\boldsymbol{\delta b}_i(t_{k+1})$ is the error in the i^{th} vector observation. It is assumed that the sequence $\boldsymbol{\delta b}_i(t_{k+1})$, $i = 1, \ldots, m$, is a zero-mean, white sequence with a known covariance matrix, R_{k+1}^i.

Moreover, the vector observations acquired at the same time are assumed to be uncorrelated with one another. That is,

$$E\left\{\,\boldsymbol{\delta b}_i(t_{k+1})\right\} = \mathbf{0} \tag{6.26a}$$

$$E\left\{\,\boldsymbol{\delta b}_i(t_{k+1})\,\boldsymbol{\delta b}_i(t_l)^T\right\} = R^i_{k+1}\,\delta_{k+1,l} \tag{6.26b}$$

$$E\left\{\,\boldsymbol{\delta b}_i(t_{k+1})\,\boldsymbol{\delta b}_j(t_{k+1})^T\right\} = R^i_{k+1}\,\delta_{ij} \tag{6.26c}$$

where $i, j = 1, 2, \ldots m$, $k, l = 1, 2 \ldots$ and $\delta_{k+1,l}$, δ_{ij} denote Kronecker deltas. In addition, it is also assumed that the vector observations are uncorrelated with the process noise, $\boldsymbol{\epsilon}_k$, and with the initial state, X_0. From Eqs. (6.10) and (6.11), we see that the elements of V^i_{k+1} are linear functions of the zero-mean measurement noise. Thus, using Eq. (6.26a) in Eqs. (6.10) and (6.11), the sequence V^i_{k+1} is shown to be zero-mean. Since V_{k+1} is a weighted sum of zero-mean sequences V^i_{k+1} (see Eq. 6.25), then V_{k+1} is also zero-mean. Using Eqs. (6.26), an analytic expression for the covariance matrix of V_{k+1} , denoted by R_{k+1}, can be derived. A summary of the computation of the 16×16 matrix R_{k+1} is given next. Its proof is lengthy but straightforward and is provided in Appendix E.3. For the sake of clarity the symbol t_{k+1} is dropped from the following equations; however, it should be remembered that all the computations carry the time tag t_{k+1}. Given $\mathbf{r}_i$, R^i and a_i, $i = 1, 2, \ldots m$, compute

$$\widetilde{R}_i = \begin{bmatrix} [\mathbf{r}_i\times] & \mathbf{r}_i \\ -\mathbf{r}_i^T & 0 \end{bmatrix} \tag{6.27a}$$

$$\Lambda^i = \left(\widetilde{R}_i \otimes I_4\right) M \tag{6.27b}$$

$$R_i = \Lambda^i\, R^i\, \Lambda^{iT} \tag{6.27c}$$

$$\alpha_i = \frac{a_i}{\sum_{i=1}^m a_i} \tag{6.27d}$$

$$R_{k+1} = \sum_{i=1}^{m} \alpha_i^2 R_i \tag{6.27e}$$

where, as before, $\otimes$ denotes the Kronecker product and the matrix M is defined in Eq. (6.21).

6.5 Filter Implementation

6.5.1 State-Dependence of the Process Noise

As mentioned earlier, and as shown in Eq. (6.17), the process noise, W_k, is a function of the state, X_k, thus, in order to compute an approximate value of the process noise covariance matrix, Q_k, the state is replaced by its best available estimate, $\widehat{X}_{k/k}$. This step is similar to what is done in an Extended Kalman filter (EKF); however, unlike the EKF, where the non-linearities are linearized in the state, here the linearization is in the time step, Δt. As a result, if $\boldsymbol{\epsilon}_k$ is very large, Δt can be taken small to offset the error.

6.5.2 Singularity of the Measurement Noise Covariance Matrix

The measurement noise covariance matrix, R_{k+1}, which is computed in Eq. (6.27) is singular. Indeed, assuming that the covariance matrix in each vector-measurement error, R^i, is of full rank (rank three), then every 16×16 matrix R_i, (see Eq. 6.27c), is at most of rank three, and is, therefore, singular. Thus, if there is a single vector measurement ($m = 1$) at t_{k+1} the rank of R_{k+1} is three. Simulations showed that if two non-collinear vector observations are acquired at t_{k+1}, the rank of R_{k+1} increases to six, and, in the case of three or more than three non-collinear measurements ($m \geq 3$) the rank equals nine. In no case, the 16×16 matrix R_{k+1} is of rank greater than nine. There are several techniques to cope with the issue of a singular measurement covariance matrix (see e.g. [40, p. 354]). To circumvent the problem here, we add small values to the main diagonal of R_{k+1}, which has a stabilizing effect on the numerics of the Kalman filter. This is done by choosing a small β relatively to the assumed level of the noise, and computing R_{k+1} as follows

$$R_{k+1} = \sum_{i=1}^{m} \alpha_i^2 R_i + \beta I_{16} \tag{6.28}$$

where I_{16} denotes the 16×16 identity matrix.

6.5.3 Algorithm Summary

The matrix Kalman filter of the K-matrix is summarized in the following.

1. Initialization: $$\widehat{X}_{0/0} = Y_0 \qquad P_{0/0} = R_0 \tag{6.29}$$

2. Time Update:

$$\widehat{X}_{k+1/k} = \Phi_k \, \widehat{X}_{k/k} \, \Phi_k^T \tag{6.30}$$

$$\mathcal{F}_k = \Phi_k \otimes \Phi_k \tag{6.31}$$

$$P_{k+1/k} = \mathcal{F}_k \, P_{k/k} \, \mathcal{F}_k^T + Q_k \tag{6.32}$$

where ⊗ denotes the Kronecker product [58, p.243].

3. Measurement Update:

$$\widetilde{Y}_{k+1} = Y_{k+1} - \widehat{X}_{k+1/k} \tag{6.33}$$

$$S_{k+1} = P_{k+1/k} + R_{k+1} \tag{6.34}$$

$$\mathcal{K}_{k+1} = P_{k+1/k} \, S_{k+1}^{-1} \tag{6.35}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \sum_{j=1}^{4} \sum_{l=1}^{4} \mathcal{K}_{k+1}^{jl} \, \widetilde{Y}_{k+1} \, E^{lj} \tag{6.36}$$

where $\mathcal{K}_{k+1}^{jl}$ is a 4 × 4 submatrix of the 16 × 16 matrix $\mathcal{K}_{k+1}$ defined by

$$\mathcal{K}_{k+1} = \underbrace{\begin{bmatrix} \mathcal{K}_{k+1}^{11} & \cdots & \mathcal{K}_{k+1}^{14} \\ \vdots & \ddots & \vdots \\ \mathcal{K}_{k+1}^{41} & \cdots & \mathcal{K}_{k+1}^{44} \end{bmatrix}}_{\text{4 submatrices}} \left.\vphantom{\begin{bmatrix} a \\ a \\ a \end{bmatrix}}\right\} \text{4 submatrices} \tag{6.37}$$

and E^{lj} is a 4 × 4 matrix with 1 in the element (lj) and 0 elsewhere.

$$P_{k+1/k+1} = \left(I_{16} - \mathcal{K}_{k+1}\right) P_{k+1/k} \left(I_{16} - \mathcal{K}_{k+1}\right)^T + \mathcal{K}_{k+1} \, R_{k+1} \mathcal{K}_{k+1}^T \tag{6.38}$$

The variance and the covariance of $\Delta X[i,j]$ [the element (ij) in the matrix ΔX] are

$$\text{var}\{\,\Delta X[i,j]\} = P\,[(j-1)m+i,(j-1)m+i] \tag{6.39}$$

$$\text{cov}\{\,\Delta X[i,j],\,\Delta X[k,l]\} = P[(j-1)m+i,(l-1)m+k] \tag{6.40}$$

where $i,k = 1,2,3,4$, and $j,l = 1,2,3,4$. The variable ΔX denotes either the a posteriori or the a priori estimation error as applicable, and P is the associated covariance matrix. The matrix Kalman filter that is described in Eqs.(6.29) to (6.38) constitutes the first stage of the attitude estimation process; namely, the Kalman filtering of the K-matrix. The second stage consists of the computation of the eigenvector of K that belongs to the largest eigenvalue.

6.5.4 Reduced Covariance Filter

In this section we show how to reduce the computational complexity of the filter by doing further reasonable assumptions on the statistics of the system's noises. Namely, we assume for our Design-model that the rows of the matrix process noise, W_k are independently identically distributed with 4×4 covariance matrix $\overline{Q}_k$. Similar assumptions are done with respect to the matrix measurement noise, V_k, and the initial estimation error, $\Delta X_{0/0}$. These assumptions are expressed by

$$Q_k = \overline{Q}_k \otimes I_4 \tag{6.41a}$$

$$R_k = \overline{R}_k \otimes I_4 \tag{6.41b}$$

$$P_{0/0} = \overline{P}_{0/0} \otimes I_4 \tag{6.41c}$$

where $\otimes$ denotes the Kronecker product. As shown in Appendix E.4, the use of Eqs. (6.41) in the equations of the filter (Eqs. 6.29 to 6.38) reduces the covariance computation from 16×16 to 4×4 matrix equations. Moreover, it enables more compact notation for the state measurement update stage. The reduced covariance filter is summarized next.

1. Initialization:

$$\widehat{X}_{0/0} = Y_0 \qquad \overline{P}_{0/0} = \overline{R}_0 \tag{6.42}$$

2. Time Update:

$$\widehat{X}_{k+1/k} = \Phi_k \widehat{X}_{k/k} \Phi_k^T \tag{6.43}$$

$$\overline{P}_{k+1/k} = \Phi_k \overline{P}_{k/k} \Phi_k^T + \overline{Q}_k \tag{6.44}$$

where $\overline{P}_{k/k}$ and $\overline{P}_{k+1/k}$ are 4×4 matrices.

3. Measurement Update:

$$\widetilde{Y}_{k+1} = Y_{k+1} - \widehat{X}_{k+1/k} \tag{6.45}$$

$$\overline{S}_{k+1} = \overline{P}_{k+1/k} + \overline{R}_{k+1} \tag{6.46}$$

$$\overline{K}_{k+1} = \overline{P}_{k+1/k}(\overline{S}_{k+1})^{-1} \tag{6.47}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \widetilde{Y}_{k+1}\overline{K}_{k+1}^T \tag{6.48}$$

$$\overline{P}_{k+1/k+1} = \left(I_4 - \overline{K}_{k+1}\right)\overline{P}_{k+1/k}\left(I_4 - \overline{K}_{k+1}\right)^T + \overline{K}_{k+1}\overline{R}_{k+1}\overline{K}_{k+1}^T \tag{6.49}$$

where $\overline{P}_{k+1/k+1}$, $\overline{S}_{k+1}$, and $\overline{K}_{k+1}$ are 4×4 matrices.

As a result of assumptions (6.41) the computation of the filter covariance is identical for each row of the estimation error matrix. This is why only a reduced 4 × 4 covariance computation is needed. The computation complexity is reduced thus by a factor of $4^4 = 256$. If the full covariance matrices are needed they are readily computed as shown in Appendix E.4 by

$$P_{k/k} = \overline{P}_{k/k} \otimes I_4 \tag{6.50a}$$

$$P_{k+1/k} = \overline{P}_{k+1/k} \otimes I_4 \tag{6.50b}$$

$$S_{k+1} = \overline{S}_{k+1} \otimes I_4 \tag{6.50c}$$

$$\mathcal{K}_{k+1} = \overline{K}_{k+1} \otimes I_4 \tag{6.50d}$$

6.5.5 Covariance Matrix of the Quaternion Estimation Error

In this section we use and extend a result of the QUEST algorithm presented in [17] in order to evaluate the covariance matrix of the quaternion estimation error. We will show how to extract this matrix from the covariance matrix $P_{k/k}$ computed in Eq. (6.38).

We begin by reviewing the result of interest. Denote by $\boldsymbol{\delta}\mathbf{z}$ the measurement error in the 3×1 vector $\mathbf{z}$ of a K-matrix, as given in Eq. (3.3); let P_{zz} be the covariance matrix of $\boldsymbol{\delta}\mathbf{z}$. Denote by $\boldsymbol{\delta q}$ the estimation error in the QUEST solution according to the $\mathbf{q}$-method; here $\boldsymbol{\delta q}$ is defined as a multiplicative error; that is

$$\boldsymbol{\delta q} = \widehat{\mathbf{q}} \star \mathbf{q}^{-1} \tag{6.51}$$

where $\mathbf{q}$ is the true quaternion, $\widehat{\mathbf{q}}$ is the estimate, $\star$ and $\widehat{\mathbf{q}}^{-1}$ denote the operations of quaternion product and quaternion inverse, respectively (see [1, p. 758]). Thus, $\boldsymbol{\delta q}$ itself is a quaternion; it has a vector part, $\boldsymbol{\delta e}$, and a scalar part, δq. This error-quaternion is related to the rotation that brings the true body frame onto the estimated body frame. Assuming that the angle of this rotation is very small, δq is approximated by 1, as done in [17]. The uncertainty then is concentrated in $\boldsymbol{\delta e}$. It is shown in [17] that a good approximation, P_{ee}, to the covariance matrix of $\boldsymbol{\delta e}$ is computed as

$$P_{ee} = N P_{zz} N \tag{6.52}$$

where

$$N = \left\{ 2 \sum_{i=1}^{m} \left(I_3 - \mathbf{r}_i \mathbf{r}_i^T \right) \right\}^{-1} \tag{6.53}$$

and $\mathbf{r}_i$, $i = 1, 2, \ldots, m$, is the batch of observed directions acquired at a particular epoch time, and resolved in the reference frame.

In order to use Eq. (6.52) in our algorithm, we consider $\boldsymbol{\delta}\mathbf{z}$ as an estimation error rather than a measurement error. Then the extraction of the covariance matrix of $\boldsymbol{\delta}\mathbf{z}$, P_{zz}, from the estimation error covariance matrix, $P_{k/k}$, is done in a straightforward manner. Since

$$\boldsymbol{\delta}\mathbf{z} = \begin{bmatrix} \Delta X(1,4) \\ \Delta X(2,4) \\ \Delta X(3,4) \end{bmatrix} \tag{6.54}$$

then, using Eqs. (6.39) and (6.40), where $m = 4$, yields

$$P_{zz} = \begin{bmatrix} P(13,13) & P(13,14) & P(13,15) \\ P(14,13) & P(14,14) & P(14,15) \\ P(15,13) & P(15,14) & P(15,15) \end{bmatrix} \tag{6.55}$$

Thus, one can evaluate the covariance matrix, P_{ee}, by, first, computing $P_{k/k}$ from Eq. (6.38), then, extracting the submatrix P_{zz} (Eq. 6.55), and finally, using Eqs. (6.52) and (6.53). Notice that the matrix N in Eq. (6.53) depends only on the chosen reference directions; thus, if the reference frame is inertial, N is computed once at the start of the algorithm; otherwise N is propagated using the dynamics of the reference frame, which is assumed to be accurately known.

6.6 Constrained Estimation

In this section we employ Pseudo-Measurement (PM) techniques in order to constrain the estimation process. This technique was applied in a quaternion EKF [28] to normalize the quaternion estimate. We apply it here to enforce the properties of symmetry and zero-trace on the K-matrix estimate. Using a PM to constrain the estimation process is similar to introducing “soft constraints” in a standard weighted least squares optimization problem. The substance of this technique is explained as follows. Suppose the true state variable, X, satisfies the following constraint

$$f(X) = 0 \tag{6.56}$$

where $f(X)$ may be scalar or vector, and suppose we have an imaginary device that measures $f(X)$ with some small error. The associated PM is thus formulated as

$$f(X) = v \tag{6.57}$$

where v is the associated PM noise with appropriate dimension. This noise is typically chosen as a zero-mean white noise with a given covariance matrix R_v. This covariance matrix is used as a tuning parameter in the filter to "strengthen" or "soften" the constraint. Note that when $f(\cdot)$ is non-linear, following the standard procedure, the PM is linearized and the linearized model is implemented in an EKF (see [28]). In this work we propose two different PM models to enforce symmetry, and a PM for enforcing the zero-trace constraint on the estimated K-matrix. These linear constraints yield linear PMs which are implementable in a linear matrix Kalman filter.

6.6.1 Symmetry Constraint

We know that the true matrix state variable is a symmetric matrix. This is expressed as

$$X^T - X = 0 \tag{6.58}$$

or, equivalently, as

$$X^T = X \tag{6.59}$$

or, also, as

$$X = \frac{1}{2}(X + X^T) \tag{6.60}$$

We use Eqs. (6.59) and (6.60) to develop two different PM models.

First Type of Symmetry Pseudo-Measurement

It is assumed that an imaginary device is measuring the 4×4 state matrix, X_{k+1}, with some small zero-mean white noise, denoted by V_{k+1}^{sym}, and is giving as output the transpose of the best available estimate, $\widehat{X}_{k+1/k+1}$, where $\widehat{X}_{k+1/k+1}$ is obtained from Eq. (6.36). Thus, the symmetry pseudo-measurement equation is

$$\widehat{X}_{k+1/k+1}^T = X_{k+1} + V_{k+1}^{sym} \tag{6.61}$$

Note that if the PM, $\widehat{X}_{k+1/k+1}$, is error-free and if we drop the matrix noise, V^{sym}_{k+1} from Eq. (6.61), we obtain $X^T = X$ which is the desired basic property as given in Eq (6.59). Since, as is evident from Eq. (6.15), Eq. (6.61) has the standard structure of a linear matrix measurement, it can be incorporated in the system mathematical model as follows. Let R^{sym}_{k+1} denote the 16×16 covariance matrix of V^{sym}_{k+1}. Using Eq. (6.61) and R^{sym}_{k+1}, the symmetry-measurement update stage is formulated as

$$\widetilde{Y}_{k+1} = \widehat{X}^T_{k+1/k+1} - \widehat{X}_{k+1/k+1} \tag{6.62a}$$

$$S_{k+1} = P_{k+1/k+1} + R^{sym}_{k+1} \tag{6.62b}$$

$$\mathcal{K}_{k+1} = P_{k+1/k+1}\, S^{-1}_{k+1} \tag{6.62c}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k+1} + \sum_{j=1}^{4}\sum_{l=1}^{4} \mathcal{K}^{jl}_{k+1}\, \widetilde{Y}_{k+1}\, E^{lj} \tag{6.62d}$$

$$P_{k+1/k+1} = \left(I_{16} - \mathcal{K}_{k+1}\right) P_{k+1/k+1} \left(I_{16} - \mathcal{K}_{k+1}\right)^T + \mathcal{K}_{k+1} R^{sym}_{k+1} \mathcal{K}^T_{k+1} \tag{6.62e}$$

where the 4×4 matrices $\mathcal{K}^{jl}_{k+1}$ in Eq. (6.62d) are submatrices of the 16×16 matrix $\mathcal{K}_{k+1}$; they are defined according to the partition described in Eq. (6.37). The covariance matrix R^{sym}_{k+1} is a filter tuning matrix parameter according to which one can weigh the influence of the symmetry pseudo-measurement, $\widehat{X}^T_{k+1/k+1}$. As an example, in view of Eq. (6.61), it may be intuitive to choose R^{sym}_{k+1} of the order of $P_{k+1/k+1}$, say

$$R^{sym}_{k+1} = P_{k+1/k+1} \tag{6.63}$$

Using Eq. (6.63) in Eq. (6.62b), and computing the gain matrix using Eq. (6.62c) yields

$$\mathcal{K}_{k+1} = \frac{1}{2}\, I_{16} \tag{6.64}$$

Thus, inserting Eq. (6.64) into Eq. (6.62d) yields, after some manipulations,

$$\widehat{X}_{k+1/k+1} = \frac{1}{2}\left(\widehat{X}_{k+1/k+1} + \widehat{X}^T_{k+1/k+1}\right) \tag{6.65}$$

In this example, the symmetry update stage, Eq. (6.65), is intuitively appealing; it produces the symmetric matrix closest (in the Frobenius norm [57]) to $\widehat{X}_{k+1/k+1}$.

Second Type of Symmetry Pseudo-Measurement

The second type of symmetry PM is based on the symmetry property as given in Eq. (6.60).

$$\widehat{X}_{k+1/k+1} = \frac{1}{2}\left(X + X^T\right) + V_{k+1}^{sym} \tag{6.66}$$

Before developing the expression for the measurement matrix of the vectorized form of Eq. (6.66), we use a result from [58, p. 259] to formulate the following theorem.

Theorem 1. *Let vecX and vecX^T, respectively, denote the vectorized transform of X and X^T, then*

$$vecX^T = Per_{4,4}\, vecX \tag{6.67}$$

where $Per_{4,4}$ is the 16×16 *permutation matrix defined as*

$$Per_{4,4} \overset{\triangle}{=} \left[E^{ji}\right]_{ij} \qquad i, j = 1, \ldots, 4 \tag{6.68}$$

and each 4×4 *matrix E^{ji} has* 1 *at position (ji) and all other entries are zero.*

Proof. See [58, p. 259]. □

Applying the vec operator (see [59, p. 272]) to Equation (6.66) and using the last theorem yields

$$\begin{aligned} \mathrm{vec}\,\widehat{X}_{k+1/k+1} &= \frac{1}{2}\left(\mathrm{vec}X + \mathrm{vec}X^T\right) + \mathrm{vec}V_{k+1}^{sym} \\ &= \frac{1}{2}\left(\mathrm{vec}X + Per_{4,4}\,\mathrm{vec}X\right) + \mathrm{vec}V_{k+1}^{sym} \\ &= \left(I_{16} + Per_{4,4}\right)\,\mathrm{vec}X + \mathrm{vec}V_{k+1}^{sym} \end{aligned} \tag{6.69}$$

The first equation stems from the linearity property of the vec operator, and the second equation is obtained using the theorem stated above. Let $\mathcal{H}^{sym}$ denotes the 16×16 measurement matrix of the vectorized system. Equation (6.69) leads thus to the following definition of $\mathcal{H}^{sym}$,

$$\mathcal{H}^{sym} \overset{\triangle}{=} \frac{1}{2}\left(I_{16} + Per_{4,4}\right) \tag{6.70}$$

Using Eqs. (6.66) and (6.70), the symmetry measurement update stage is summarized as follows.

$$\widetilde{Y}_{k+1} = \frac{1}{2}\left(\widehat{X}_{k+1/k+1} - \widehat{X}^T_{k+1/k+1}\right) \tag{6.71a}$$

$$S_{k+1} = \mathcal{H}^{sym}\, P_{k+1/k+1}\, \mathcal{H}^{symT} + R^{sym}_{k+1} \tag{6.71b}$$

$$\mathcal{K}_{k+1} = P_{k+1/k+1}\, \mathcal{H}^{symT}\, S^{-1}_{k+1} \tag{6.71c}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k+1} + \sum_{j=1}^{4}\sum_{l=1}^{4} \mathcal{K}^{jl}_{k+1}\, \widetilde{Y}_{k+1}\, E^{lj} \tag{6.71d}$$

$$P_{k+1/k+1} = \left(I_{16} - \mathcal{K}_{k+1}\, \mathcal{H}^{sym}\right) P_{k+1/k+1} \left(I_{16} - \mathcal{K}_{k+1}\, \mathcal{H}^{sym}\right)^T + \mathcal{K}_{k+1} R^{sym}_{k+1} \mathcal{K}^T_{k+1} \tag{6.71e}$$

where $\mathcal{K}^{jl}_{k+1}$ is a 4 × 4 submatrix at position (jl) of the 16 × 16 gain matrix $\mathcal{K}_{k+1}$, and I_{16} is the 16 × 16 identity matrix. It can be seen from Eqs. (6.62a) and (6.71a) that the innovations sequence, $\widetilde{Y}_{k+1}$, is almost identical in both types of PMs (they differ only by a constant scalar multiplication); however, the covariance computation is very different. This stems from the difference in the model equations (Eqs. 6.61 and 6.66). As mentioned before the measurement covariance, R^{sym}_{k+1}, is used for filter tuning. Using a high covariance, the estimate, $\widehat{X}_{k+1/k+1}$, remains almost unchanged, while a low covariance eventually yields a gain close to I_{16}. This leads to the following intuitive update of the estimate

$$\widehat{X}_{k+1/k+1} = \frac{1}{2}\left(\widehat{X}_{k+1/k+1} + \widehat{X}^T_{k+1/k+1}\right) \tag{6.72}$$

6.6.2 Trace Constraint

The trace constraint is handled using the following pseudo-measurement model.

$$0 = tr X_{k+1} + v^{tr}_{k+1} \tag{6.73}$$

where "*tr*" denotes the trace operator, v^{tr}_{k+1} is a scalar zero-mean white noise with covariance r^{tr}_{k+1}, and the value of the pseudo-measurement is 0. Using the definition of the trace operator yields

$$\begin{aligned} tr X_{k+1} &= \sum_{i=1}^{4} X_{k+1}(i,i) \\ &= \sum_{i=1}^{4} \mathbf{e}_i{}^T X_{k+1}\, \mathbf{e}_i \end{aligned} \tag{6.74}$$

where $\mathbf{e}_i$, $i = 1, 2, 3, 4$, are the standard unit vectors in $\mathbb{R}^4$. Equation (6.74) has the standard form of a linear matrix measurement model. Using Eq. (6.74), the trace update stage is formulated as follows

$$\mathbf{h}^{tr} = \sum_{i=1}^{4} (\, \mathbf{e}_i \otimes \mathbf{e}_i) \tag{6.75a}$$

$$s_{k+1} = \mathbf{h}^{trT} P_{k+1/k+1} \mathbf{h}^{tr} + r^{tr}_{k+1} \tag{6.75b}$$

$$\mathbf{k}_{k+1} = P_{k+1/k+1} \mathbf{h}^{tr} / s_{k+1} \tag{6.75c}$$

$$\overline{K}_{k+1} = \sum_{j=1}^{4} \mathbf{k}^j_{k+1} \, \mathbf{e}_j{}^T \tag{6.75d}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k+1} - \left(tr \widehat{X}_{k+1/k+1} \right) \overline{K}_{k+1} \tag{6.75e}$$

$$P_{k+1/k+1} = \left(I_{16} - \mathbf{k}_{k+1} \mathbf{h}^{trT} \right) P_{k+1/k+1} \left(I_{16} - \mathbf{k}_{k+1} \mathbf{h}^{trT} \right)^T + r^{tr}_{k+1} \mathbf{k}_{k+1} \mathbf{k}^T_{k+1} \tag{6.75f}$$

where each vector $\mathbf{k}^j_{k+1}$ in the computation of $\overline{K}_{k+1}$ (Eq. 6.75d) is the 4×1 column-vector at position (j), $j = 1, 2, 3, 4$, in the 16×1 gain column-vector, $\mathbf{k}_{k+1}$. Equation (6.75e) is proved as follows:

$$\begin{aligned}
\widehat{X}_{k+1/k+1} &= \widehat{X}_{k+1/k+1} + \sum_{j=1}^{4} \left[\left(-\, tr \widehat{X}_{k+1/k+1} \right) \left(\mathbf{k}^j_{k+1} \, \mathbf{e}_j{}^T \right) \right] \\
&= \widehat{X}_{k+1/k+1} - \left(tr \widehat{X}_{k+1/k+1} \right) \left(\sum_{j=1}^{4} \mathbf{k}^j_{k+1} \, \mathbf{e}_j{}^T \right) \\
&= \widehat{X}_{k+1/k+1} - \left(tr \widehat{X}_{k+1/k+1} \right) \overline{K}_{k+1}
\end{aligned}$$

The first equation is the general formulation of the state measurement update stage in a matrix KF. The second equation comes from the fact that $\left(tr \widehat{X}_{k+1/k+1} \right)$ is independent of j, and the last equation is obtained by using Eq. (6.75d). Similarly to the symmetry constraint case, the covariance r^{tr}_{k+1} is used as a tuning parameter in order to optimally enforce the zero-trace property along the estimation process. In the extreme case of $r^{tr}_{k+1} = 0$, straightforward computations yields

$$\overline{K}_{k+1} = \frac{1}{4} I_4 \tag{6.76}$$

Using Eq. (6.76) into Eq. (6.75e) yields

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k+1} - \frac{1}{4} \left(tr \widehat{X}_{k+1/k+1} \right) I_4 \tag{6.77}$$

where $\widehat{X}_{k+1/k+1}$ denotes the updated estimate. Thus computing the trace of $\widehat{X}_{k+1/k+1}$, and using the fact that $tr I_4 = 4$, yields $tr \widehat{X}_{k+1/k+1} = 0$; that is, $\widehat{X}_{k+1/k+1}$ satisfies exactly the zero-trace constraint.

The constrained matrix Kalman filter consists of the algorithm described earlier, Eqs. (6.29) to Eq. (6.38), to which the symmetry and trace update stages are added. Thus, each update stage operates on the preceding state estimate and estimation-error covariance. Like in an iterative Kalman filter, the three stages are performed sequentially at the same epoch time.

6.7 Numerical Study

In this section the unconstrained filter, the constrained filter, and the filter Optimal-REQUEST are tested and compared by means of extensive Monte-Carlo simulations.

In the present case study we consider a spacecraft whose kinematics is similar to that of the Microwave Anisotropy Probe (MAP) satellite, which was launched on June 30, 2001 [62]. The attitude measurement devices simulated here are composed of a digital sun sensor (DSS), an autonomous star-tracker (AST), and a triad of rate gyroscopes. Two Cartesian coordinate frames are considered; namely, the Sun frame, which is assumed to be inertial, and the body frame. The rotation of the body frame with respect to the Sun frame is composed of a spin rotation and of a nutation; the spin and the nutation rates are 0.464 rpm and 1 rph, respectively; the constant nutation angle, which is defined between the spacecraft spinning axis and the anti-Sun line of sight, is equal to 22.5 deg.

It is assumed here that the AST observes the same star during the whole simulation. Therefore, two identical inertial directions are observed at each sampling time; namely, the Sun direction, and the direction to that star. These directions are represented in the Sun frame by the unit vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively. The Sun frame that is considered here has its third axis coinciding with the line of sight between the spacecraft and the Sun, thus $\mathbf{r}_1 = [0\,0\,1]^T$. For observability enhancement, it is assumed that the AST can find a star along a direction perpendicular to $\mathbf{r}_1$, for instance, $\mathbf{r}_2 = [1\,0\,0]^T$. The unit vector measurements, $\mathbf{b}_i$, $i = 1, 2$, are simulated by adding a small zero-mean white Gaussian noise to the ideal observed directions, and by normalizing the result; that is

$$\mathbf{b}_i = \frac{A\,\mathbf{r}_i + \boldsymbol{\delta b}_i}{\|A\,\mathbf{r}_i + \boldsymbol{\delta b}_i\|} \tag{6.78}$$

where A is the correct transformation matrix from the Sun to the body coordinates and

$$\boldsymbol{\delta b}_i \sim \mathcal{N}\{\,\mathbf{0},\,\sigma_i^2\,I_3\} \tag{6.79}$$

for $i = 1, 2$. Here, σ_1 equals 1 arc-mn ($\simeq 17\,$mdeg), and σ_2 equals 10 arc-sec ($\simeq 2.8\,$mdeg). The vector measurement sampling period is 10 seconds. The output of the triad of gyroscopes is contaminated by a zero-mean Gaussian white noise with a covariance matrix $\sigma_\epsilon^2\,I_3$, where $\sigma_\epsilon = 100\,$mdeg/hr. The gyros sampling period is 0.5 second. It is assumed that there is no initial knowledge of the attitude.

6.7.1 Comparison between Optimal-REQUEST and the MKF algorithm

The results of the Monte-Carlo simulation (100 runs) are summarized in Figs. 6.1 to 6.9. Figures 6.1 to 6.2 present plots of some elements of the matrix ΔX where ΔX is the error between the true K-matrix and its estimate; that is, $\Delta X_{k/k} = K_k - K_{k/k}$. In each figure the solid lines are used for the unconstrained matrix Kalman filter (MKF)[2] variables, and the dotted lines are used for Optimal-REQUEST (OPREQ) variables. Let ΔX_{11}, ΔX_{13}, and ΔX_{14} denote the elements $(1,1)$, $(1,3)$, and $(1,4)$, respectively, of the updated estimation error matrix, $\Delta X_{k/k}$. Extensive simulations showed that any element of the matrix $\Delta X_{k/k}$ behaves like one of these three representative elements. In the ensuing analysis, we use superscripts "OPREQ" and "MKF", respectively, on any variable that belongs to OPREQ and MKF. Figure 6.1 shows the time history of the mean and of the $\pm 1\sigma$-envelope of ΔX_{11}. We see that the plots of ΔX_{11}^{OPREQ} and ΔX_{11}^{MKF} are practically undistinguishable. The means oscillate around zero with two time periods; namely, a short period of about two minutes, which corresponds to the spin rotation of the spacecraft, and a long period of one hour, which is due to the spacecraft nutation. The order of magnitude of the standard deviations is $5 \cdot 10^{-6}$. Figure 6.2 shows the variations of the mean and $\pm 1\sigma$-envelope of ΔX_{13}. Similarly to ΔX_{11}, the filters OPREQ and MKF yield very close variations in ΔX_{13}. However, the oscillations in ΔX_{13} are much more random than in

[2] We apply the *full* covariance filter as described in Eqs. (6.29) to (6.38)

ΔX_{11}. Figure 6.3 depicts the mean and ±1σ-envelope of ΔX_{14}. The oscillations in the mean of ΔX_{14}^{OPREQ} are much less damped than the oscillations in the mean of ΔX_{14}^{MKF}. The ratio between the amplitudes of those oscillations reaches 8. The ±1σ-envelope of ΔX_{14}^{OPREQ} is an upper-bound for that of ΔX_{14}^{MKF}. The dc-level of the oscillations in the standard deviation is about $4 \cdot 10^{-7}$ for the MKF, and twice as large ($8 \cdot 10^{-7}$) for OPREQ. Thus, there are clear differences between the performance of the two filters. The filter OPREQ produces an estimate of X_{13} that is much less damped and twice less accurate than that of the MKF. Note that the magnitude of ΔX_{14} is ten times less than that of ΔX_{11} and ΔX_{13}.

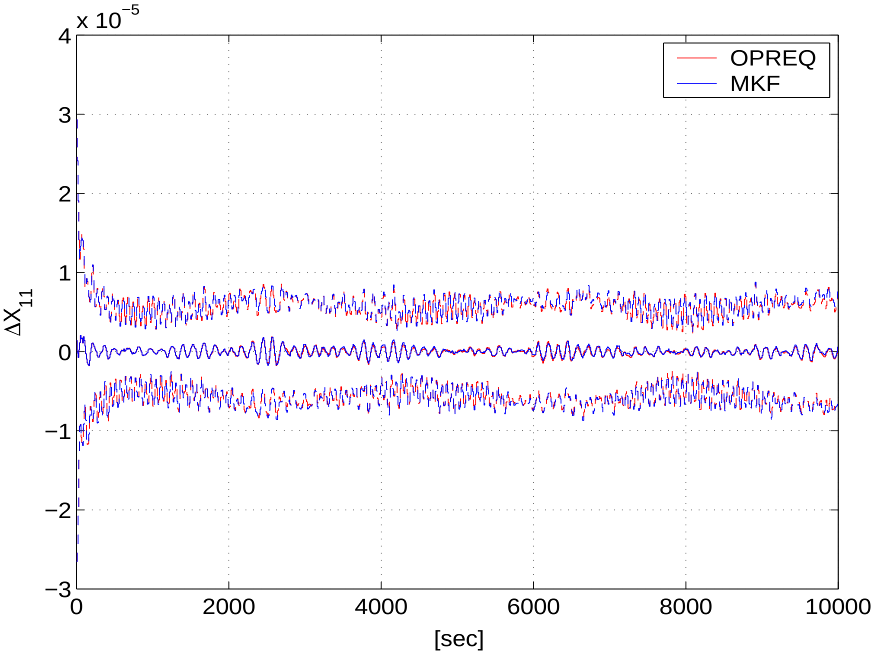


Fig. 6.1: Monte-Carlo mean and ±1σ envelope of ΔX_{11}.

Next we compare the gains of the two algorithms. Since the gain of OPREQ is a scalar, which is here denoted by ρ_{OPREQ}, and the gain of the MKF is a 16 × 16 matrix denoted in Eq.(6.35) by $\mathcal{K}_{k+1}$. We define the scalar ρ_{MKF} as the Euclidean norm of $\mathcal{K}_{k+1}$, and compare it to ρ_{OPREQ}. Also, we check the structure of $\mathcal{K}_{k+1}$ along the estimation process and see how close it is to be a multiple of the identity matrix. Figure 6.4 shows the variations of the means of ρ_{MKF} (solid line) and ρ_{OPREQ} (dotted-line). During the

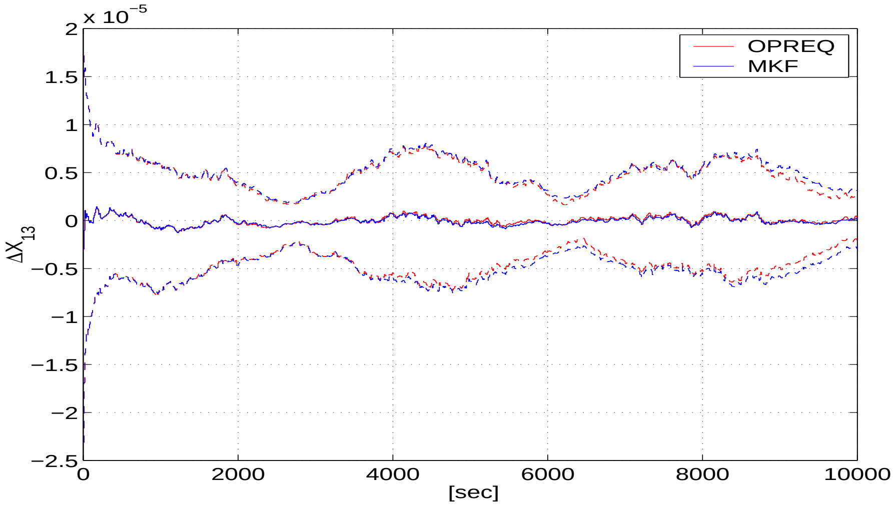


Fig. 6.2: Monte-Carlo mean and $\pm 1\sigma$ envelope of ΔX_{13}.

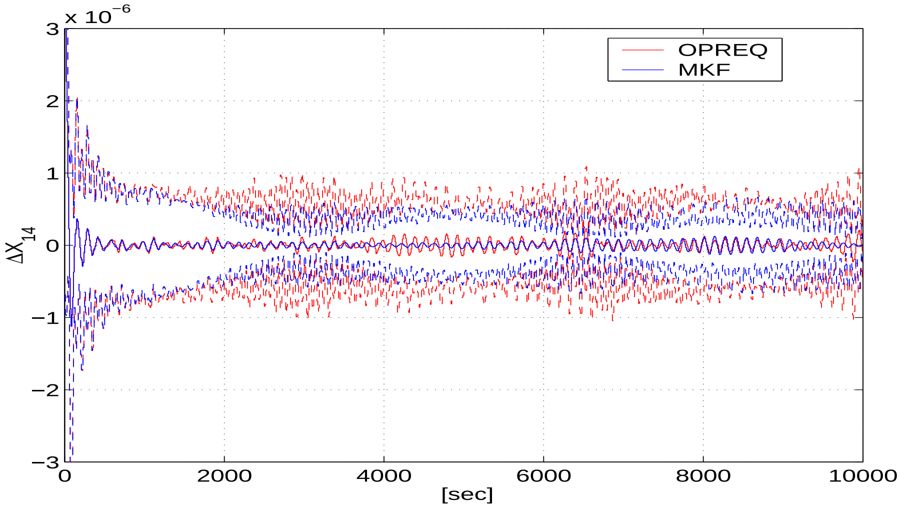


Fig. 6.3: Monte-Carlo mean and $\pm 1\sigma$ envelope of ΔX_{14}.

transition phase, the first 1500 seconds, the two quantities are very close one another. Then ρ_{MKF} reaches a steady-state value around 0.015, while ρ_{OPREQ} oscillates at the spin and nutation frequencies. The maxima of ρ_{OPREQ} are about 0.025 and the dc-level of the oscillations is around 0.020. This result gives some insight into the result described earlier in Fig. 6.3; because of the higher gain, the filter OPREQ weighs more heavily the new incoming observations than the filter MKF; therefore, the update estimate in OPREQ

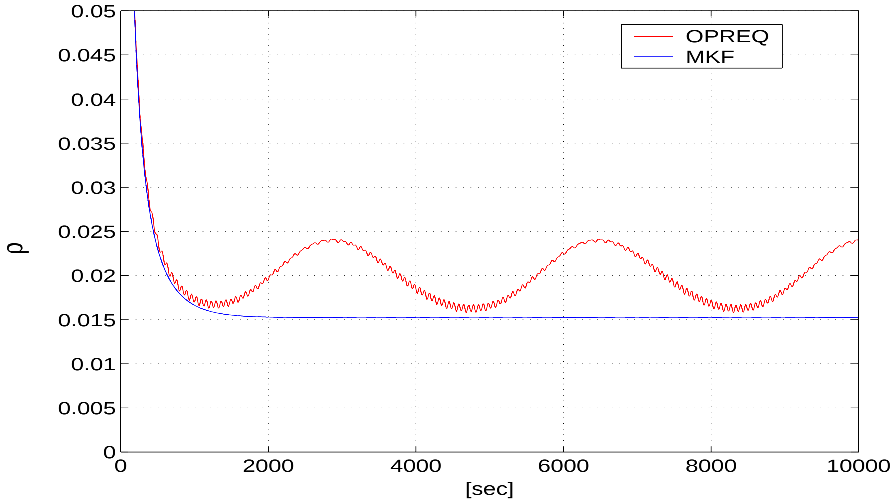


Fig. 6.4: Monte-Carlo mean of ρ_{OPREQ} and ρ_{MKF}.

is noisier than that in the MKF. The simulation also shows that the gain matrix in the MKF is close to be diagonal at the beginning, but converges to a steady-state structure which is far from being diagonal. We omit the plots related to the latter result for the sake of brevity.

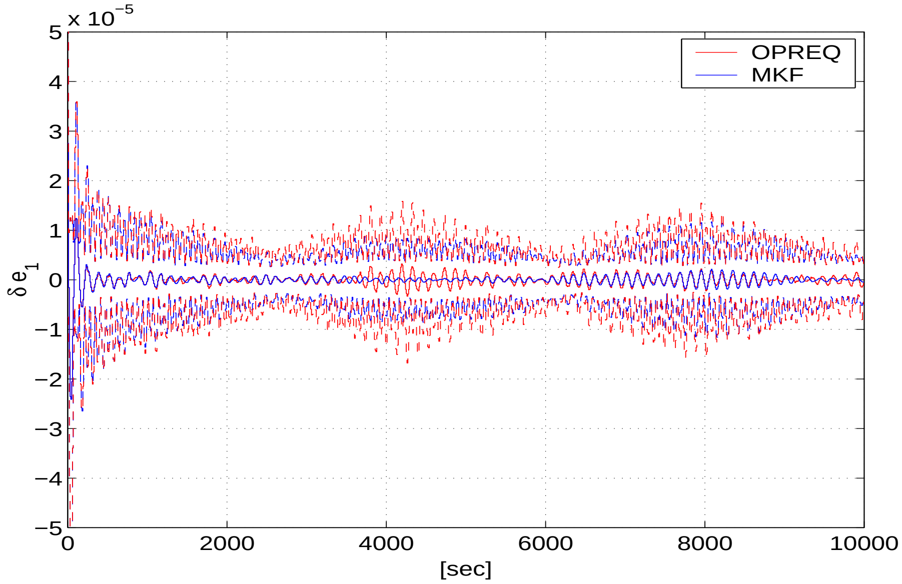


Fig. 6.5: Monte-Carlo mean and ±1σ envelope of δe_1.

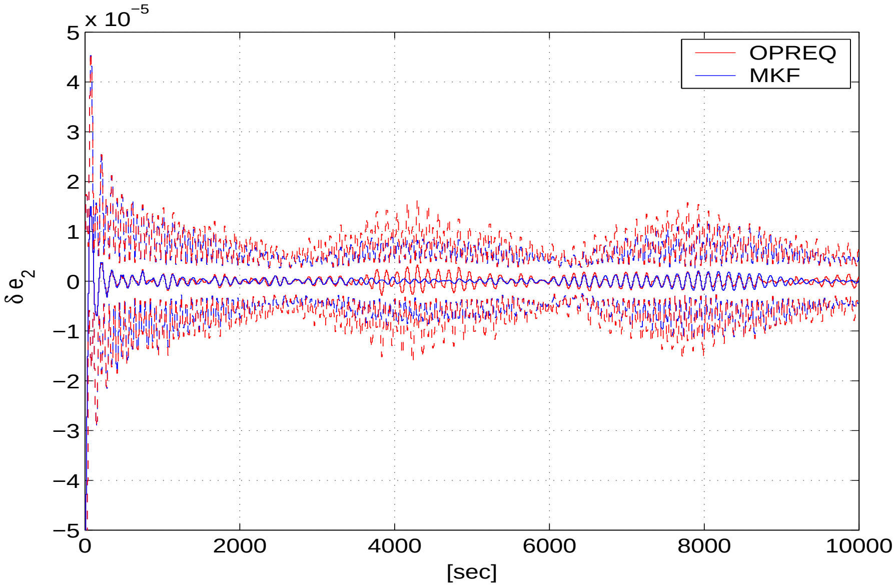


Fig. 6.6: Monte-Carlo mean and $\pm 1\sigma$ envelope of δe_2.

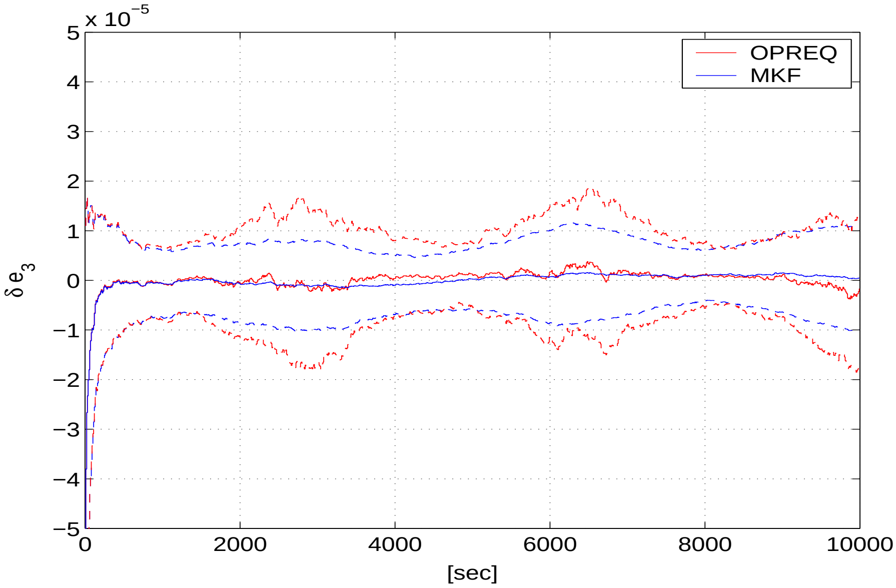


Fig. 6.7: Monte-Carlo mean and $\pm 1\sigma$ envelope of δe_3.

As mentioned earlier, the quaternion estimation error, denoted by $\boldsymbol{\delta q}$, is defined as the quaternion of the small rotation that brings the estimated body frame onto the true body frame (Eq. 6.51. It has four components, which are denoted by δe_1, δe_2, δe_3, and δq. The

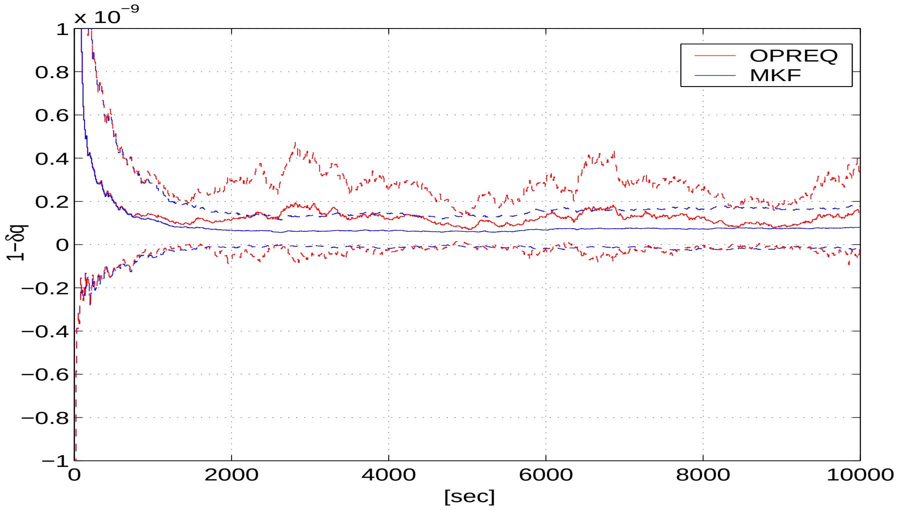


Fig. 6.8: Monte-Carlo mean and $\pm 1\sigma$ envelope of $1 - \delta q$.

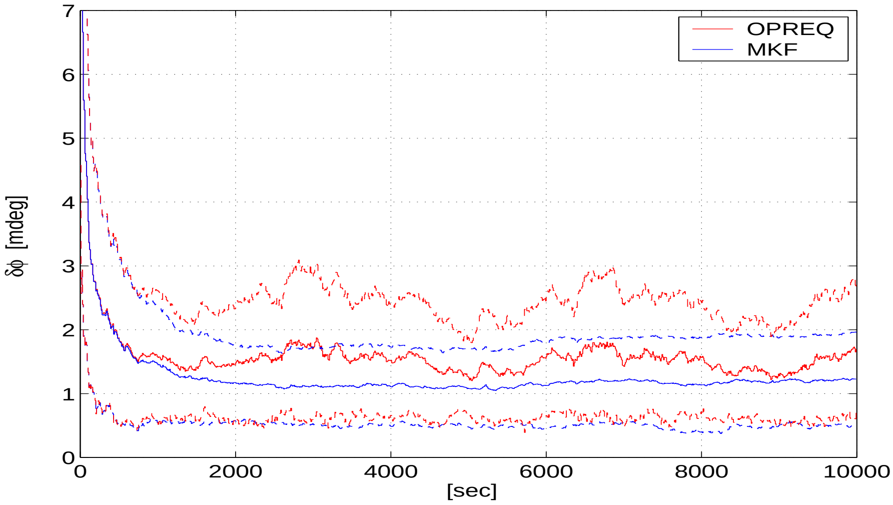


Fig. 6.9: Monte-Carlo mean and $\pm 1\sigma$ envelope of $\delta\phi$.

variations of the means and $\pm 1\sigma$-envelopes of the four components of $\boldsymbol{\delta q}$ are depicted in Figures 6.5 to 6.8, respectively. As can be seen from Figures 6.5 and 6.6, the errors δe_1 and δe_2 have very similar variations. The oscillations in the means are less damped in OPREQ than in the MKF; the ratio between the oscillation picks reaches 8. The dc-level of the $\pm 1\sigma$-envelopes in OPREQ is twice that of the MKF $(1.2 \cdot 10^{-5}$ against $6 \cdot 10^{-6})$. The same analysis, as that done for Figs. 6.5 and 6.6, can be done for δe_3 in Fig. 6.7, except for the fact that the variations of δe_3^{OPREQ} are much more random. As opposed to δe_3^{OPREQ}, the variations of δe_3^{MKF} have a regular oscillating pattern, modulated by the nutation and the spin frequencies. Instead of plotting the variations of δq, we plot those of $(1 - \delta q)$; indeed, this quantity is the one that becomes small when the quaternion estimation error becomes small (a quaternion expressing a zero-rotation is equal to $[0\,0\,0\,1]^T$). After a transition phase of 1500 seconds, the plots of OPREQ and the MKF clearly separate. The variations of $1 - \delta q^{MKF}$ are smooth, with a mean of 10^{-10} and a standard deviation of 10^{-10}; on the other hand, the mean of $1 - \delta q^{MKF}$ oscillates above 10^{-10}, and the standard deviation is of the order of $2 \cdot 10^{-10}$. Let $\delta\phi$ denote the angle of the small rotation that is represented by $\boldsymbol{\delta q}$. This angle is extracted from δq using the known relation, $\delta q = \cos(\delta\phi/2)$. Figure 6.9 presents the time histories of the mean and of the $\pm 1\sigma$-envelope of $\delta\phi$. The mean of the MKF is stabilized at 1.2 mdeg, while the mean of OPREQ oscillates above it, around a dc-level of 1.7 mdeg. The standard deviation in the MKF is about 0.8 mdeg; it is smaller than that in OPREQ, which is 1.2 mdeg.

Overall, we see that the MKF performs better than OPREQ. We also deduce from Fig. 6.8 that there is a small bias in the estimated quaternion. This bias may be the result of applying the **q**-method to the estimated state matrix; that is, the eigenvector normalization that is performed by the eigenvector solver induces a bias in the estimate. Considering the order of magnitude in $\delta\phi$, both algorithms perform well since the measurement angular error, which is about 3 mdeg in the most accurate measurements, and about 17 mdeg in the less accurate ones, is filtered and brought down to a level of 1.2 mdeg (in MKF) and of 1.7 mdeg (in OPREQ).

6.7.2 Estimate Properties in the First Stage of the MKF

Here we investigate the discrepancies in the symmetry and in the zero-trace properties of the estimated matrix along the estimation process. We measure these discrepancies by computing the following performance indices

$$J_{sym}(k) = \| \widehat{X}_{k/k} - \widehat{X}^T_{k/k} \| \tag{6.80}$$

$$J_{tra}(k) = tr\left(\widehat{X}_{k/k} \right) \tag{6.81}$$

where $\|[\cdot]\|$ denote the Euclidean norm of $[\cdot] \in \mathbb{R}^{m\times n}$. Extensive simulations were run, starting with a symmetric, zero-trace initial estimate, and processing the measurements according to the MKF algorithm. These simulations showed that J_{sym} and J_{tra} started and remained on a level of 10^{-15} during the whole estimation process. We conclude that the measurement update stage of the MKF (Eq. 6.62) does not destroy the symmetry and zero-trace properties of the initial estimate.

Next we analyze the estimate properties in view of the covariance computation for specific estimation errors. We only consider four elements in the estimation error matrix, $\Delta X_{k/k}$, which is enough to illustrate the present analysis; namely, we choose the 4×1 vector $[\Delta X_{11}\ \Delta X_{21}\ \Delta X_{12}\ \Delta X_{22}]^T$. Let D denote the 4×4 covariance matrix of this vector. Using the relations described in Eqs. (6.39) and Eq. (6.40), the symmetric matrix D is directly extracted from the 16×16 covariance matrix $P_{k/k}$ as

$$D = \begin{bmatrix} P_{11} & P_{12} & P_{15} & P_{16} \\ & P_{22} & P_{25} & P_{26} \\ & & P_{55} & P_{56} \\ & & & P_{66} \end{bmatrix} \tag{6.82}$$

Figure 6.10 shows the time variations of the elements of the matrix D. The location of the subplots in Fig. 6.10 corresponds to the location of the elements in D, Eq. (6.82). All the elements of D show oscillations combining the spin and the nutation frequencies. This latter fact is a direct consequence of the estimate-dependence of the process covariance matrix, Q_k (Eq. 6.23). Beneath these oscillations we observe that the diagonal elements, which are the covariances, are bounded by $8 \cdot 10^{-11}$ in steady-state. Looking at the off-diagonal elements, which are the cross-covariances, we distinguish two types of behaviors; the first one consists of oscillations around zero, this is the case of D_{12} and D_{13}, for instance; the second type shows a steady-state with an upper(lower) bound of $6 \cdot 10^{-11}$ ($-6 \cdot 10^{-11}$); this is the case of D_{23} (D_{14}). We deduce from the second type that the estimation errors ΔX_{21} and ΔX_{12} are highly correlated; similarly ΔX_{11} and ΔX_{22} are highly correlated, but with a negative sign. Indeed, as shown by extensive simulations, once the estimated state has converged to the true state, the *linear* properties of symmetry and zero-trace are satisfied in the estimation error, too; the symmetry property induces, for instance, $\Delta X_{21} \simeq \Delta X_{12}$; this explains why these estimation errors are highly correlated.

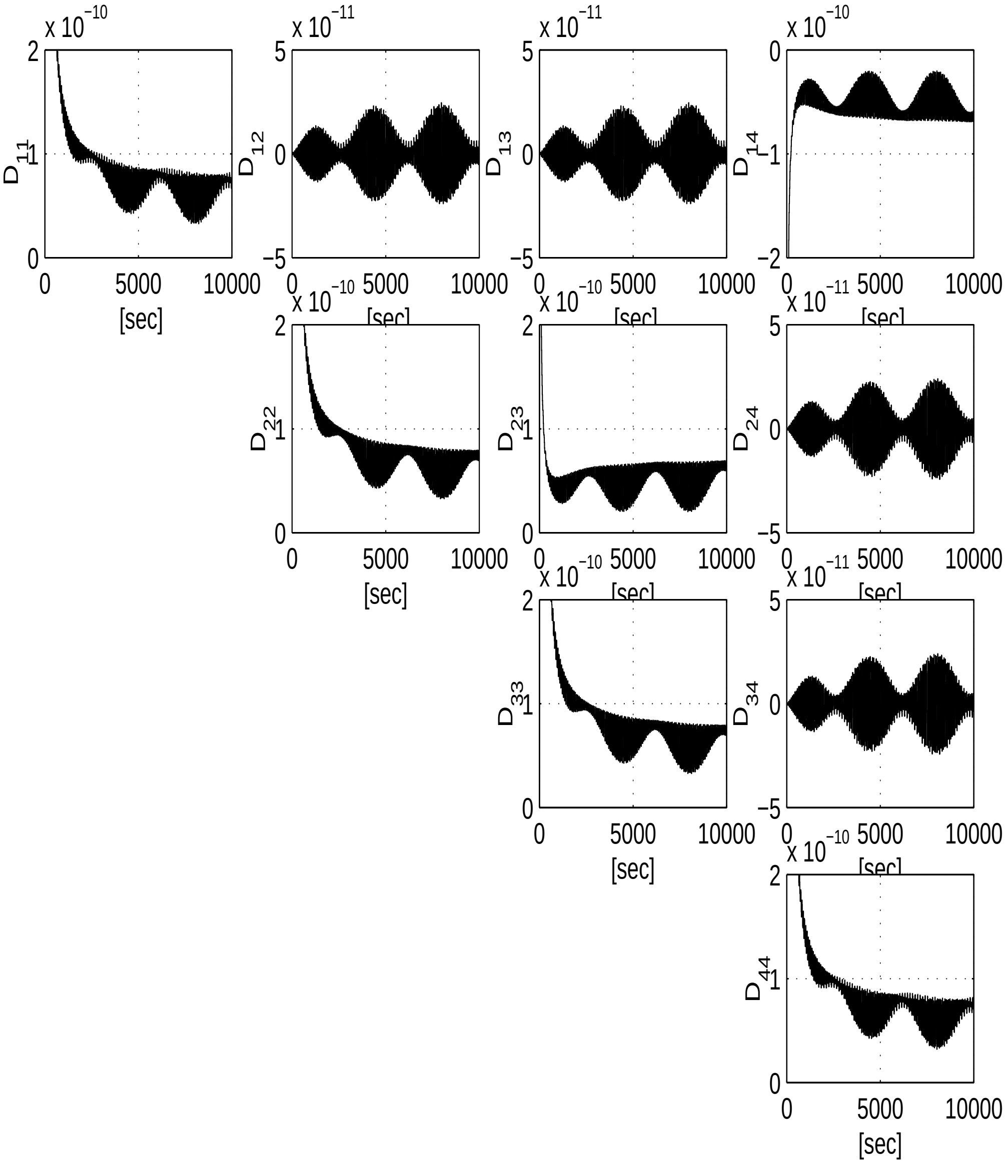


Fig. 6.10: Covariance matrix of $[\Delta X_{11}\ \Delta X_{21}\ \Delta X_{12}\ \Delta X_{22}]^T$.

The zero-trace property induces $tr(\Delta X) = \Delta X_{11} + \Delta X_{22} + \Delta X_{33} + \Delta X_{44} \simeq 0$, which shows why ΔX_{11} and ΔX_{22} are highly correlated. Thus, without any constraint other than the requirement to be optimal the MKF recreates the inherent properties of the true state, and yields an estimation error covariance computation that is consistent with these properties.

6.7.3 Constrained Matrix Kalman Filter

We test here the performance of the constrained matrix Kalman filter (CMKF), which is the MKF that embeds the symmetry and trace update stages. The symmetry update is performed via the second type of pseudo-measurement, which is described in the preceding section. We compare the response of the CMKF and the MKF to an initial perturbation in the symmetry and trace properties of the estimate matrix. The initial perturbations in the elements of the initial estimate are zero-mean uniformly distributed random variables with standard deviation $\sigma_{dst} = 0.05$. The covariance matrix of the symmetry pseudo-measurement, R^{sym}_{k+1}, is chosen as $R^{sym}_{k+1} = (\sigma_{dst})^2 \, I_{16}$, and the covariance of the trace pseudo-measurement is $r^{tr}_{k+1} = \sigma^2_{dst}$. All the other simulation conditions are identical to those of the preceding simulation. The results of a 100-runs Monte-Carlo simulation are presented in Figs. 6.11 to 6.15. The figures contain the plots of δe_1, δe_2, δe_3, $1 - \delta q$, and $\delta\phi$, respectively. Each figure shows the time-histories of the means that were obtained in MKF and in CMKF. An inspection of all the plots reveals that CMKF performs better than MKF. Adding to this, it appears that the initial symmetry and trace perturbation yields an estimation performance degradation in MKF by a factor of $\sim$ 50.

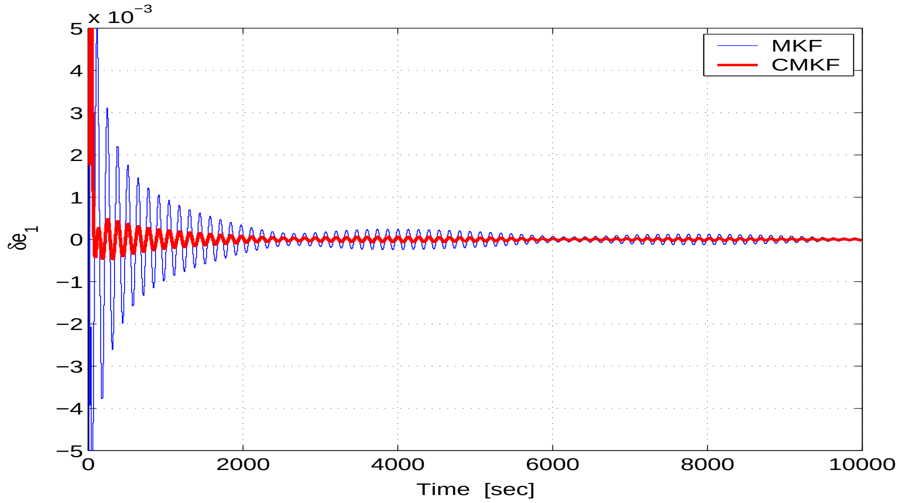


Fig. 6.11: Constrained Estimation. Monte-Carlo mean of δe_1.

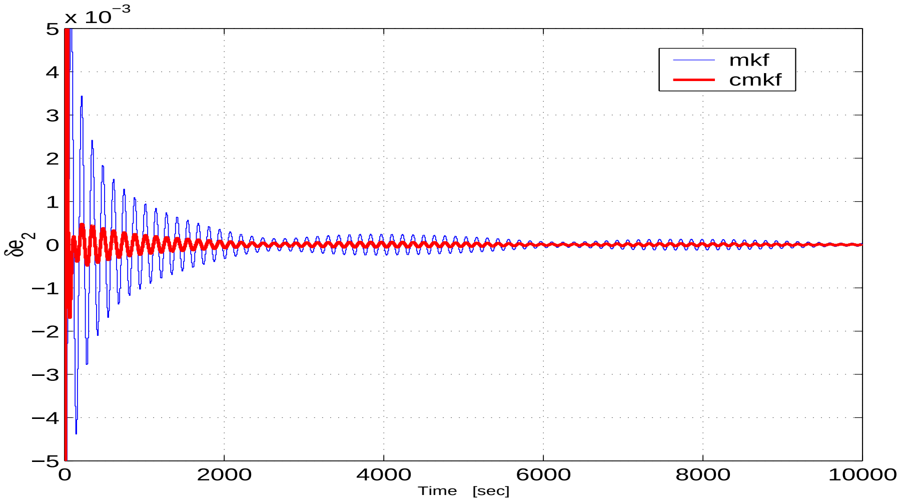


Fig. 6.12: Constrained Estimation. Monte-Carlo mean of δe_2.

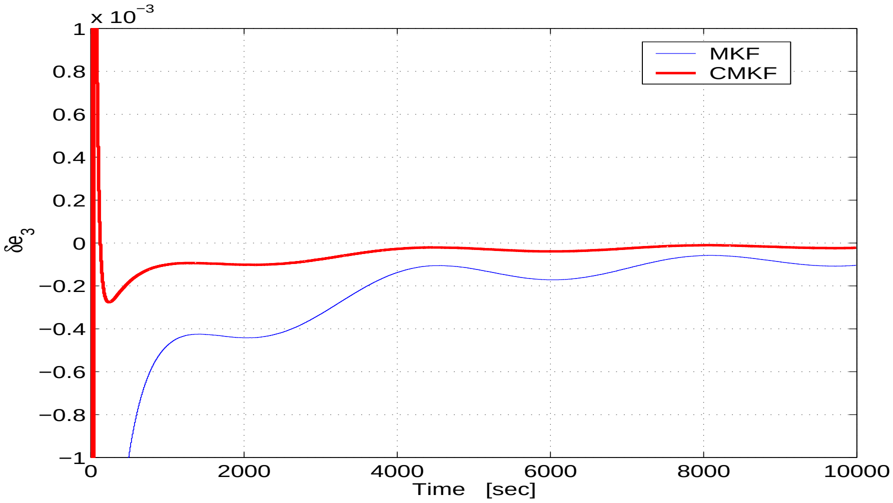


Fig. 6.13: Constrained Estimation. Monte-Carlo mean of δe_3.

6.8 Discussion

An overall comparison between the performance of the MKF algorithm and that of Optimal-REQUEST states an improvement by a factor 1.5. As mentioned in the beginning of this work this qualitative improvement was expected. Here, we analyze its causes. We begin by showing under what conditions the MKF algorithm becomes the Optimal-REQUEST algorithm.

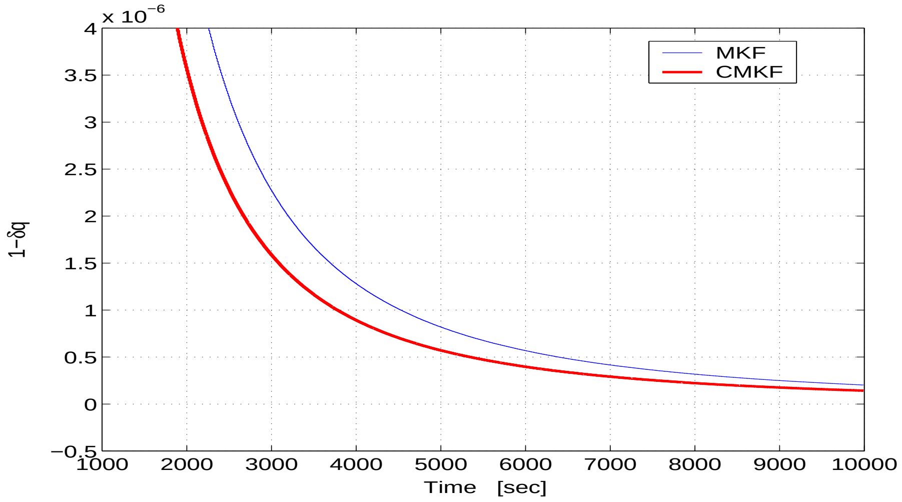


Fig. 6.14: Constrained Estimation. Monte-Carlo mean of $1-\delta q$.

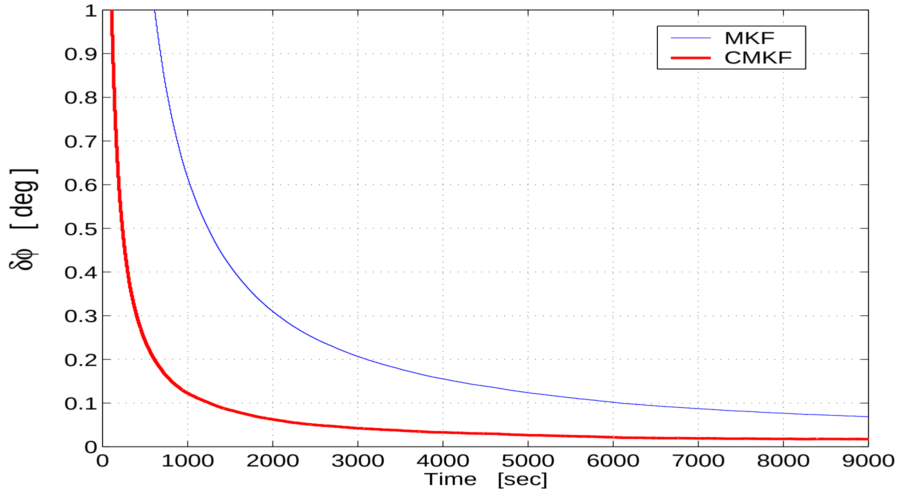


Fig. 6.15: Constrained Estimation. Monte-Carlo mean of $\delta\phi$.

Assume that W_k, V_k, and $P_{0/0}$, respectively, have independent identically distributed rows with 4×4 covariance matrices, $\overline{Q}_k$, $\overline{R}_k$, and $\overline{P}_{0/0}$[3]. In addition, assume that the gain matrix in Eq. (6.35), $\mathcal{K}_{k+1}$, is a scalar matrix; that is

$$\mathcal{K}_{k+1} = \rho_{k+1} I_{16} \tag{6.83}$$

[3] These assumptions lead to the *reduced* covariance MKF as given in Eqs. (6.42) to (6.49)

then the filter update equations (6.35) and (6.36) become

$$\widehat{X}_{k+1/k+1} = \left(1 - \rho_{k+1}\right) \widehat{X}_{k+1/k} + \rho_{k+1} Y_{k+1} \tag{6.84a}$$

$$\overline{P}_{k+1/k+1} = \left(1 - \rho_{k+1}\right)^2 \overline{P}_{k+1/k} + \rho_{k+1}^2 \overline{R}_{k+1} \tag{6.84b}$$

where $\widehat{X}$ denotes the estimate of the K-matrix, Y_{k+1} is the matrix measurement constructed using the vector measurement acquired at t_{k+1}, and $\overline{P}$ denotes the 4×4 estimation error covariance matrix for each row of the matrix state. Equations (6.84) are obtained by using Eq. (6.83) in Eqs. (6.48) and (6.49). We recall now that the update equations in Optimal-REQUEST are given by[4] (see Chapter 3)

$$K_{k+1/k+1} = \left(1 - \rho_{k+1}^*\right) K_{k+1/k} + \rho_{k+1}^* \, \delta K_{k+1} \tag{6.85a}$$

$$P_{k+1/k+1} = \left(1 - \rho_{k+1}^*\right)^2 P_{k+1/k} + \rho_{k+1}^*{}^2 \mathcal{R}_{k+1} \tag{6.85b}$$

where $K_{k+1/k+1}$ denotes the updated estimate of the K-matrix, δK_{k+1} is the matrix measurement at t_{k+1}, ρ_{k+1}^* is the optimized fading memory factor, and $P_{k+1/k+1}$ and $\mathcal{R}_{k+1}$, respectively, denote the uncertainty matrices in the updated estimation error and in the new measurement. Comparing Eqs. (6.84) and (6.85) we realize that they have similar structures. They differ, however, because the matrices $\overline{P}_{k+1/k+1}$ and $P_{k+1/k+1}$ are different, and so are the matrices $\overline{R}_{k+1}$ and $\mathcal{R}_{k+1}$. In fact, as will be shown next, these matrices are related as follows

$$\mathcal{R}_{k+1} = 4\overline{R}_{k+1} \qquad P_{k+1/k+1} = 4\overline{P}_{k+1/k+1} \tag{6.86}$$

Recalling the definition of $\mathcal{R}_{k+1}$ from Chapter 3 yields

$$\begin{aligned} \mathcal{R}_{k+1} &= E\left[V_{k+1} V_{k+1}^T\right] \\ &= E\left[\sum_{i=1}^{4} \mathbf{v}_i^c \mathbf{v}_i^{cT}\right] \\ &= \sum_{i=1}^{4} E\left[\mathbf{v}_i^c \mathbf{v}_i^{cT}\right] \\ &= 4\overline{R}_{k+1} \end{aligned}$$

[4] The weights m_k, m_{k+1}, δm_{k+1}, are not rewritten in Eqs. (6.85) for the sake of clarity in the comparison. They actually enter the computation of the matrices $K_{k+1/k}$, δK_{k+1}, $P_{k+1/k+1}$, and $\mathcal{R}_{k+1}$.

where $\mathbf{v}_i^c$, $i = 1,2,3,4$, denote the 4×1 column-vectors of the matrix V_{k+1}. Note that these vectors are identical to the rows of V_{k+1} since V_{k+1} is a symmetric matrix (see Eqs. 6.10 and 6.11). The third equation is due to the linearity of the expectation operator. The last equation comes from the assumption that the rows of V_{k+1} (and thus also its columns) are independently identically distributed with covariance matrix $\overline{R}_{k+1}$. The same argument is readily used for $P_{k+1/k+1}$. As a result, the covariance update equations of the MKF and of Optimal-REQUEST are identical, up to a multiplication by a constant.

While finding the conditions under which the general MKF algorithm reduces to the Optimal-REQUEST algorithm, we have quantified the difference between the measurement noise levels in the filters, and found that it is four times greater in Optimal-REQUEST than in the matrix filter. It is believed that this is the principal cause of the discrepancy between the filters' performance. The latter is illustrated by a simple example. Consider the following scalar state-space equations

$$x_{k+1} = x_k + w_k \tag{6.87}$$

$$y_{k+1} = x_{k+1} + v_{k+1} \tag{6.88}$$

where w_k and v_{k+1} are the process and measurement noise sequences, respectively, which satisfy the usual stochastic assumptions of the basic state-space model. Let q and r, denote, respectively, the covariances of w_k and v_k. The scalar algebraic Riccati equation is readily formulated as follows.

$$p_\infty^2 - qp_\infty + qr = 0 \tag{6.89}$$

where p_∞ denotes the steady-state estimation error covariance. Solving for p_∞ in Eq. (6.89) and using the assumption that $q/r << 1^5$ yields the following good approximation to p_∞.

$$p_\infty \simeq \sqrt{qr} \tag{6.90}$$

It is clear from Eq. (6.90) that multiplying the measurement noise covariance, r, by 4 will deteriorate the estimation error standard deviation by factor $\sqrt{2} \simeq 1.4$, which reminds us of the factor 1.5 observed in the Monte-Carlo simulations.

[5] This assumption is fully justified in the application case where the process noise comes from gyro outputs and the measurement noise comes from vector measurement sensing devices. In this case, $q \simeq 10^{-14}\,[\mathrm{rad}^2/\sec^2]$ and $r \simeq 10^{-10}$ so that $q/r \simeq 10^{-4}$.

6.9 Concluding Remarks

This chapter presented a new recursive estimator of the quaternion-of-rotation from sequential vector observations. The estimation algorithm that was developed here is an enhanced Optimal-REQUEST filter. The approach was to develop a linear matrix Kalman filter for the K-matrix. This was made possible by the linear dynamics and the linear measurement models. The tool which was used to tackle the matrix estimation problem was the matrix Kalman filter (MKF). Optimally constraining the symmetry and zero-trace properties in the matrix estimate was easily achieved in the framework of the MKF. Extensive Monte-Carlo (MC) simulations were run to test the MKF against Optimal-REQUEST, and against the constrained MKF.

The results showed that, in general, Optimal-REQUEST and the MKF have similar transient phases, but that the MKF clearly outperforms Optimal-REQUEST in steady-state. The MC-means of all the estimation errors in the MKF are much more damped, and the MC standard deviations of the MKF are approximately 1.5 times smaller than those of Optimal-REQUEST. The relation between both algorithms was analytically investigated. The analysis showed that the difference in the filters performance is due, to a large extent, to the conservative optimality cost used in Optimal-REQUEST. That analysis also showed that special assumptions on the noise stochastic models may substantially lower (16 times) the filter computational burden.

The MC simulations showed that the MKF K-matrix estimate acquired and preserved the desired symmetry and zero-trace properties, without any aiding algorithm. The MC results were corroborated by analyzing the estimation error covariance on a sample run. The constrained MKF achieved a better accuracy in the case of an initial perturbation in the desired estimate properties.

In conclusion:

- It is recommended to use Optimal-REQUEST in the initial phase, and to switch to the MKF only if finer and smoother attitude is needed.
- It is also recommended that the constrained MKF be used to mitigate the effect of an unpredicted perturbation in the symmetry and zero-trace properties of the estimated K-matrix.

Chapter 7

Matrix Kalman Filtering of the Direction Cosine Matrix

This work will be presented as Paper AIAA 2003-5482 at the 43th AIAA Guidance, Navigation, and Control Conference, Austin, TX, Aug. 2003.

7.1 Introduction

This chapter is concerned with the development of a matrix Kalman filter (MKF) for estimating the Direction Cosine Matrix (DCM), also called attitude matrix, from vector observations. It is well known that three independent parameters are necessary and sufficient to fully represent three-axis attitude. The DCM is a 3×3 rotation matrix [1, p. 411]; that is, it is a nine-elements rotation parameter. In spite of this redundancy, the attitude matrix, denoted here by D, is a convenient attitude representation. This is so because the equations that describe the vector measurement model and the spacecraft kinematical model are linear in D [1, pp. 411,512]. The latter fact enables the use of efficient algorithms for determining D in the framework of linear least squares estimation theory.

There is a long history of algorithms that estimate D using vector observations. Notice that two non-collinear vector observations are necessary and sufficient to fully determine D [2, 17, 24]. Optimal estimators of D, which handle more than two vector observations, typically fall into two categories. The first one has its origin in Wahba's problem [5], which was described in Chapter 1. As mentioned already, the solutions of Wahba's problem are closed-form solutions, and need no a priori estimate of D [6–9, 12–14]. Furthermore, the orthogonality constraint on D is inherently formulated in that problem.

The second approach; namely, the Kalman filtering approach, provides a convenient framework for developing minimum-variance attitude matrix estimators for time-varying systems. Moreover, the linear nature of the attitude matrix state-space model equations enables the use of a linear Kalman filter (KF) [34]. A feature of the KF developed in [34] is that the state matrix, D, is transformed into its vectorized representation, by stacking its columns one underneath the other. This is done in order to comply with the state-vector formulation of the conventional state-space model. Following that approach all the system equations are vectorized. As a result, the physical insight in the plant equations is lost, and the analysis of the related KF becomes uneasy.

As mentioned in Chapter 5, the issue of developing optimization algorithms for estimation and control, which operate on matrix systems and preserve the original system formulation, has been given much attention. The various solutions of Wahba's problem are such examples: their development and analysis, which involve standard matrix decompositions, would have been cumbersome, if not impossible, without the matrix notation.

In this work, we present a matrix Kalman filter (MKF) of the DCM. Due to the linear structure of the DCM state-space equations, the MKF is naturally fitted to operate on the DCM plant. A traditional issue when estimating the DCM is the orthogonalization of the updated matrix. Orthogonality is, indeed, critical when the DCM estimate is needed for other tasks, like axis rotation. It was shown in [34] that applying an iterative orthogonalization algorithm [60] after each measurement update stage enhanced the estimator performance. In the present work, besides that iterative procedure, three other orthogonalization algorithms are presented. One of them is a known closed-form algorithm. The other two are novel applications of the pseudo-measurement (PM) technique, as presented in [28]. According to that technique, a state constraint equation (here - the orthogonality constraint) is modelled as a virtual measurement equation. This way, the constraint is naturally embedded in the framework of the MKF. The covariance associated with the PM model is a tuning parameter that can strengthen or soften the state constraint along the estimation process. The term pseudo-measurement is used in the literature because the constraint is interpreted as a virtual measurement model. That technique was successfully applied in a quaternion Extended Kalman filter in order to normlize the updated quaternion estimate [28].

The special case of gyro white noises is considered in order to focus on the attitude estimation problem itself. Nevertheless, there are conventional techniques, like the state vector augmentation technique, that handle the case of biased gyros in a Kalman filter. Here we utilize the state augmentation procedure in order to cope with the matrix formulation of the DCM plant equations. We develop the mathematical model of an augmented 3×4 state matrix, which includes the DCM and a set of constant gyro biases.

The remaining of the chapter is organized as follows. Section 7.2 presents the mathematical model of the attitude matrix system. This model is known in the literature, excepted the explicit expression for the process noise covariance matrix, which is given here. Two matrix Kalman filters of the attitude matrix are developed in section 7.3. The first filter includes a full (9×9) covariance computation algorithm. The second one includes a reduced (3×3) covariance computation. Both filters do not include any orthogonalization procedure. Four orthogonalization procedures are presented in section 7.4. Section 7.5 presents the development of the augmented state matrix mathematical model. The performance of the different filters of the DCM are demonstrated and compared in section 7.6 via Monte-Carlo simulations. Finally, a summary and conclusions are presented in the last section.

7.2 Mathematical Model

In the following we write the state-space matrix equations that describe the discrete time dynamical model and the measurement model of the system.

7.2.1 Process Model

Let the spacecraft body-frame and the reference frame be denoted, respectively, by $\mathcal{B}$ and $\mathcal{R}$. Assuming that $\mathcal{B}$ is rotating with respect to $\mathcal{R}$, we denote by $\boldsymbol{\omega}^o$ the angular velocity vector expressed in $\mathcal{B}$. It is well known that the dynamics of the attitude matrix,

D, is governed by the following matrix differential equation [1, p. 512]

$$\frac{d}{dt}D = -\left[\boldsymbol{\omega}^o\times\right]D \tag{7.1}$$

The matrix $\left[\boldsymbol{\omega}^o\times\right]$ in Eq. (7.1) is a 3×3 skew-symmetric matrix and is defined according to the identity $\left[\boldsymbol{\omega}^o\times\right]\mathbf{x} = \boldsymbol{\omega}^o \times \mathbf{x}$, where × denotes the cross-product and $\mathbf{x}$ is any 3×1 column-matrix. The discrete time version of Eq. (7.1) is the difference equation given by

$$D_{k+1} = \Phi^o_k D_k \tag{7.2}$$

where Φ^o_k is the transition matrix from time t_k to time t_{k+1} that corresponds to $-\left[\boldsymbol{\omega}^o\times\right]$. Taking the incremental time, $\Delta t = t_{k+1} - t_k$, small enough we assume that $\boldsymbol{\omega}^o$ is piecewise constant in the intervals of time length Δt, so that Φ^o_k in Eq. (7.2) can be approximated as

$$\Phi^o_k \simeq e^{-\left[\boldsymbol{\omega}^o_k\times\right]\Delta t} \tag{7.3}$$

The true angular velocity, $\boldsymbol{\omega}^o_k$, is unknown; we assume here that a triad of body-mounted gyroscopes measures $\boldsymbol{\omega}^o_k$ and that the output of the gyros, $\boldsymbol{\omega}_k$, is corrupted by an additive error, $\boldsymbol{\epsilon}_k$, thus

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}^o_k + \boldsymbol{\epsilon}_k \tag{7.4}$$

Then we substitute $\boldsymbol{\omega}_k$ for $\boldsymbol{\omega}^o_k$ in Eq. (7.3) and use the measured transition matrix, denoted by Φ_k, in Eq. (7.2). In order to compensate for the modelling errors induced by our approximations, an additive process noise matrix, denoted by W_k, is introduced in Eq. (7.2), which becomes now

$$D_{k+1} = \Phi_k D_k + W_k \tag{7.5}$$

where $\Phi_k = \exp^{-\left[\boldsymbol{\omega}_k\times\right]\Delta t}$. As shown in Appendix F.1, W_k can be written as

$$W_k = \left[\boldsymbol{\epsilon}_k\times\right]D_k\,\Delta t + h.o.t. \tag{7.6}$$

where $h.o.t.$ denotes terms of order Δt^2 and higher. Notice that the process noise matrix, W_k, is state dependent, and that the first-order term is linear in the components of $\boldsymbol{\epsilon}_k$. The gyro noise, $\boldsymbol{\epsilon}_k$, is assumed here to be a zero-mean white noise sequence with a known covariance matrix Q^ϵ_k. There are several techniques, however, that can handle the case of gyro bias.

The process model for the attitude matrix is described by Eqs. (7.5) and (7.6). These equations show that the natural form of the process model equations features a state matrix, D_k and a process noise matrix, W_k. The matrix Kalman filter, as developed in Chapter 5, is designed to handle such matrix plant.

7.2.2 Measurement Model

In the present work we use the classical vector measurement model equation, which is detailed next. We assume that a physical vector quantity is observed at each epoch time t_k; namely, we simultaneously know its decomposition in $\mathcal{R}$, $\mathbf{r}_k$, and its measured decomposition in $\mathcal{B}$, $\mathbf{b}_k$. Moreover, without loss of generality, we assume that the observed physical vector is of unit length. In general, $\mathbf{r}_k$ is accurately known from tables or almanacs while $\mathbf{b}_k$ is a noisy measurement acquired on-board by sensing devices. Modelling the measurement noise as an additive error term, $\mathbf{v}_k$, yields the classical vector measurement equation

$$\mathbf{b}_k = D_k \mathbf{r}_k + \mathbf{v}_k \tag{7.7}$$

where $\mathbf{v}_k$ is assumed to be a zero-mean white noise sequence with known covariance matrix R_k. Furthermore, we assume that $\boldsymbol{\epsilon}_k$ and $\mathbf{v}_k$ are uncorrelated with one another and with the initial attitude state matrix D_0.

The measurement equation (7.7) is a linear equation in the state matrix D_k, with the column-matrix $\mathbf{r}_k$ as a measurement matrix, and the column-matrix $\mathbf{b}_k$ as a measurement. This structure fits naturally the linear model for which the Matrix Kalman filter was developed in Chapter 5.

7.3 Filter Design

In this section we present two matrix Kalman filters (MKF) for the estimation of the attitude matrix. Both of them are operating on the state-space system described in the previous section and are written according to the general algorithm developed in Chapter 5. Both filters differ by the complexity of the covariance computation. The first algorithm performs a *full covariance computation*, where the covariance computation involves 9 × 9 matrix equations. The second algorithm performs a *reduced covariance computation*, in the sense that it needs to solve 3×3 matrix equations only. For the sake of clear notation all the matrices that are involved in the full covariance computation are denoted by calligraphic letters, like $\mathcal{P}$. We use normal fonts, like P, to denote the matrices involved in the reduced covariance computation.

7.3.1 Full Covariance Filter

This filter is developed by applying the MKF algorithm to the plant described by equations (7.5) and (7.7). In order to obtain an approximate expression for the 9×9 covariance matrix[1] of W_k, denoted by $\mathcal{Q}_k$, we neglect the high order terms ($h.o.t$) and substitute the best available estimate,$\widehat{D}_{k/k}$, for D_k in Eq.(7.6). As shown in Appendix F.2, straightforward computations yield

$$\mathcal{Q}_k = \left(\widehat{D}^T_{k/k} \otimes I_3\right) \mathcal{L} Q^{\epsilon}_k \mathcal{L}^T \left(\widehat{D}_{k/k} \otimes I_3\right) \ \Delta t^2 \tag{7.8}$$

where $\otimes$ denotes the Kronecker product (see [15, p. 227]), I_3 is the three-dimensional identity matrix, $\mathcal{L}$ is a 9×3 matrix defined as

$$\mathcal{L}^T \triangleq \begin{bmatrix} [\mathbf{e}_1 \times] & [\mathbf{e}_2 \times] & [\mathbf{e}_3 \times] \end{bmatrix} \tag{7.9}$$

and $\mathbf{e}_j$, $j = 1, 2, 3$, is the j^{th} column of I_3. The full covariance filter computes $\mathcal{Q}_k$ from Eqs. (7.8) and (7.9). The filter measurement covariance matrix is equal to R_k. The full covariance MKF is summarized as follows.

1. Initialization:

   The matrices $\widehat{D}_{0/0}$ and $\mathcal{P}_{0/0}$ are given.

2. Time Update equations:

$$\widehat{D}_{k+1/k} = \Phi_k \widehat{D}_{k/k} \tag{7.10}$$

$$\Psi_k = I_3 \otimes \Phi_k \tag{7.11}$$

$$\mathcal{P}_{k+1/k} = \Psi_k \mathcal{P}_{k/k} \Psi^T_k + \mathcal{Q}_k \tag{7.12}$$

[1] The covariance matrix of a random matrix is defined as the covariance matrix of its vectorized transform. Appendix D.1.4 presents a detailed definition.

3. Measurement Update equations:

$$\mathcal{H}_{k+1} = \mathbf{r}_{k+1}^T \otimes I_3 \tag{7.13}$$

$$\mathcal{S}_{k+1} = \mathcal{H}_{k+1}\mathcal{P}_{k+1/k}\,\mathcal{H}_{k+1}^T + R_{k+1} \tag{7.14}$$

$$\mathcal{K}_{k+1} = \mathcal{P}_{k+1/k}\,\mathcal{H}_{k+1}^T\mathcal{S}_{k+1}^{-1} \tag{7.15}$$

$$\widehat{D}_{k+1/k+1} = \widehat{D}_{k+1/k} + \sum_{j=1}^{3}\mathcal{K}_{k+1}^j(\,\mathbf{b}_{k+1} - \widehat{D}_{k+1/k}\,\mathbf{r}_{k+1})\,\mathbf{e}_j^T \tag{7.16}$$

$$\mathcal{P}_{k+1/k+1} = \left(I_9 - \mathcal{K}_{k+1}\,\mathcal{H}_{k+1}\right)\mathcal{P}_{k+1/k}\left(I_9 - \mathcal{K}_{k+1}\,\mathcal{H}_{k+1}\right)^T + \mathcal{K}_{k+1}\,R_{k+1}\mathcal{K}_{k+1}^T \tag{7.17}$$

where $\mathcal{K}_{k+1}^j$ in Eq. (7.16), $j = 1, 2, 3$, are 3×3 submatrices of the 9×3 Kalman gain matrix, $\mathcal{K}_{k+1}$, such that $\mathcal{K}^T = [(\mathcal{K}^1)^T\ (\mathcal{K}^2)^T\ (\mathcal{K}^3)^T]$. Note from Eqs .(7.10) and (7.16) that the full covariance filter, described by Eqs.(7.10) to (7.17), produces a Kalman filter estimate of the state attitude matrix, $\widehat{D}_{k/k}$, using the original matrices of the given plant.

7.3.2 Reduced Covariance Filter

The matrix compact notation of the preceding filter facilitates the analysis of the algorithm. This enables an easy manipulation and the reduction of the algorithm structure under special model assumptions. We assume here that the rows of the process noise matrix, W_k, are independently identically distributed with covariance matrix $Q_k = Q_k^\epsilon(\,\Delta t)^2$. This is equivalent to $\mathcal{Q}_k = Q_k \otimes I_3$. Similarly, we assume that the elements of the measurement noise column-matrix, $\mathbf{v}_k$, are independently identically distributed with scalar covariance μ_k, that is, $R_{k+1} = \mu_{k+1}\,I_3$. We also choose $\mathcal{P}_{0/0} = P_{0/0} \otimes I_3$, where $P_{0/0}$ is given. Then, as shown in Appendix F.3, straightforward computations, based on basic properties of the Kronecker product, show that the nine-dimensional covariance computation can be reduced to a three-dimensional covariance computation. The computational load of the covariance algorithm is thus reduced by a factor of 27 (3^3). This computational gain is achieved using rather strong stochastic model assumptions. Nevertheless, as will be shown by Monte-Carlo simulations, the simplified estimator performs well. The reduced covariance MKF is summarized as follows.

1. Initialization:

   The initial estimate, $\widehat{D}_{0/0}$, and the 3×3 matrix $P_{0/0}$ are given.

2. Time Update equations:

$$\widehat{D}_{k+1/k} = \Phi_k \widehat{D}_{k/k} \tag{7.18}$$

$$P_{k+1/k} = P_{k/k} + Q_k \tag{7.19}$$

3. Measurement Update equations:

$$s_{k+1} = \mathbf{r}_{k+1}^T \, P_{k+1/k} \, \mathbf{r}_{k+1} + \mu_{k+1} \tag{7.20}$$

$$\boldsymbol{g}_{k+1} = P_{k+1} \, \mathbf{r}_{k+1} / s_{k+1} \tag{7.21}$$

$$\widehat{D}_{k+1/k+1} = \widehat{D}_{k+1/k} + (\, \mathbf{b}_{k+1} - \widehat{D}_{k+1/k} \, \mathbf{r}_{k+1}) \, \boldsymbol{g}_{k+1}^T \tag{7.22}$$

$$P_{k+1/k+1} = \left(I_3 - \boldsymbol{g}_{k+1} \, \mathbf{r}_{k+1}^T \right) P_{k+1/k} \left(I_3 - \boldsymbol{g}_{k+1} \, \mathbf{r}_{k+1}^T \right)^T + \mu_{k+1} \, \boldsymbol{g}_{k+1} \, \boldsymbol{g}_{k+1}^T \tag{7.23}$$

Notice that the dynamics matrix, Φ_k, does not appear in the covariance time propagation (Eq. 7.19). As shown in Appendix F.3, the orthogonality property of Φ_k is the cause of this cancellation. The consequence of the stochastic assumptions is that the three rows of the state matrix are estimated using *the same* covariance computation algorithm of dimension 3 × 3. Indeed, all the necessary information is conveniently represented and processed by the 3 × 3 algorithm equations. If the *full-scale* matrices are needed, they are readily computed from the reduced associated matrices as follows (see Appendix F.3).

$$\mathcal{P}_{k+1/k} = P_{k+1/k} \otimes I_3 \tag{7.24}$$

$$\mathcal{P}_{k+1/k+1} = P_{k+1/k+1} \otimes I_3 \tag{7.25}$$

$$\mathcal{S}_{k+1} = s_{k+1} \, I_3 \tag{7.26}$$

$$\mathcal{K}_{k+1} = \boldsymbol{g}_{k+1} \otimes I_3 \tag{7.27}$$

7.4 Orthogonalization

It is well known that the attitude matrix, D, is an orthogonal matrix. This is so because D is a rotation matrix. The orthogonality property of D is crucial when the matrix is used to perform vector transformation at a high computation rate as it is in inertial navigation. Therefore, a good estimate of D is one that is nearly orthogonal.

The Kalman filter is not designed to preserve any relationship among the components of the estimated state matrix, $\widehat{D}_{k/k}$. Even if the a priori estimate, $\widehat{D}_{k+1/k}$, were orthogonal, the measurement update equation (7.22) would note preserve the orthogonality.

In this section, four different orthogonalization algorithms are presented. The first two algorithms use known results while the other two are novel applications of the concept of pseudo-measurement (PM) for constrained Kalman filtering. Recall that a square matrix, $M \in \mathbb{R}^{n\times n}$, is orthogonal if it satisfies the identities $M^T M = MM^T = I_n$, where $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix in $\mathbb{R}^{n\times n}$. This property will be used in the ensuing. For the sake of brevity the a posteriori estimate $\widehat{D}_{k/k}$ will be denoted by $\widehat{D}$ in the following.

7.4.1 Optimal Brute-Force Orthogonalization (OBF)

Following this method, once the Kalman filter has computed the a posteriori estimate, $\widehat{D}$, the following optimization problem is solved

$$\begin{aligned} &\min_D \|D - \widehat{D}\|_F^2 \\ &\textit{subject to} \;\; D^T D = I_3 \end{aligned} \tag{7.28}$$

where $\|M\|_F$ denotes the Frobenius norm of the matrix M, that is, $\|M\|_F = \sqrt{tr(MM^T)}$, and I_3 is the 3×3 identity matrix. This optimization problem, known as the orthogonal Procrustes problem [61], has the following closed form solution [13]

$$D^* = \left(\widehat{D}\widehat{D}^T\right)^{-\frac{1}{2}} \widehat{D} \tag{7.29}$$

or, equivalently [61]

$$D^* = \widehat{D}\left(\widehat{D}^T\widehat{D}\right)^{-\frac{1}{2}} \tag{7.30}$$

The orthogonal estimate, D^*, is then substituted for $\widehat{D}$ in the time propagation equation (7.18). This technique is called "brute-force" orthogonalization because the orthogonalization procedure is performed outside the filter.

7.4.2 Iterative Brute-Force Orthogonalization (IBF)

It is a procedure where orthogonalization is enforced by iterations of the following algorithm [34]

$$D^* = \widehat{D}\left(1.5\, I_3 - 0.5\widehat{D}^T\widehat{D}\right) \tag{7.31}$$

This algorithm is suboptimal, but it is much simpler than the optimal ones of Eqs. (7.29) and (7.30). When applied recursively, this algorithm produces a sequence of estimates that converges to the optimal solution of Eqs. (7.29) and (7.30) [60]. Moreover, as evidenced by simulations, orthogonality is normally reached, for all practical purpose, after one or two iterations of Eq. (7.31).

7.4.3 First Type of Orthogonality Pseudo-Measurement (OPM1)

The PM technique is a smooth way of incorporating constraints in the estimation process. Here we apply this technique in order to orthogonalize the attitude matrix estimate. Consider the orthogonality state constraint equation.

$$I_3 = D_k^T D_k \tag{7.32}$$

The matrix D_k is invertible since it is orthogonal. Thus, left-multiplication of Eq. (7.32) by the inverse of D_k^T yields

$$D_k^{-T} = D_k \tag{7.33}$$

Then we substitute $\widehat{D}$ for D_k in the left-hand side of Eq. (7.33), and we add a term, V^{ort}, in the right-hand side in order to compensate for the modelling error. This yields

$$\widehat{D}^{-T} = D_k + V_k^{ort} \tag{7.34}$$

Equation (7.34) is the orthogonality PM equation (OPM1), where $\widehat{D}^{-T}$ is the measurement and V^{ort} is the measurement noise matrix. We choose the covariance matrix of V_k^{ort} of the form $\mu^{ort} I_9$, where μ^{ort} is a tuning parameter. We choose μ^{ort} according to the weight given to the linear constraint expressed in Eq. (7.34). Specifically, if we pick a high value for μ^{ort} the filter will not significantly update $\widehat{D}$, and the new estimate, D^*, will be close to $\widehat{D}$. On the other hand, picking a low value for μ^{ort} will yield a value for D^* that is close to the pseudo-measurement, namely to $\widehat{D}^{-T}$. Extensive Monte-Carlo simulations are used to properly tuning the value of μ^{ort}.

The PM equation, Eq. (7.34), is a linear matrix equation in D_k on which the Matrix Kalman filter can operate in a natural way. The measurement update stage, which is designed on OPM1, is written next. Since the covariance matrix of V_k^{ort} is $\mu^{ort} I_9$, the reduced covariance form is utilized.

$$S_{k+1} = P_{k+1/k} + \mu^{ort} I_3 \tag{7.35}$$

$$K_{k+1} = P_{k+1/k} S_{k+1}^{-1} \tag{7.36}$$

$$D^*_{k+1/k+1} = \widehat{D}_{k+1/k+1} + \left(\widehat{D}^{-T}_{k+1/k+1} - \widehat{D}_{k+1/k+1}\right) K^T_{k+1} \tag{7.37}$$

$$P^*_{k+1/k+1} = \left(I_3 - K_{k+1}\right) P_{k+1/k+1} \left(I_3 - K_{k+1}\right)^T + \mu^{ort} K_{k+1} K^T_{k+1} \tag{7.38}$$

7.4.4 Second Type of Orthogonality Pseudo-Measurement (OPM2)

In this section we present an alternate model for an orthogonality pseudo-measurement, and develop the associated measurement update equations. Starting with the orthogonality property as given in Eq. (7.32)

$$I_3 = D_k^T D_k \tag{7.39}$$

and left-multiplying Eq. (7.39) by D_k yields

$$D_k = D_k \left(D_k^T D_k\right) \tag{7.40}$$

In the following, the time subscript on D will be dropped for the sake of brevity. Denoting by $\widehat{D}$ an available estimate of D, we substitute it for D in the left-hand side and in the parentheses of Eq. (7.40), and add a term, V^{ort}, on the right-hand side to compensate for the modelling error. This yields

$$\widehat{D} = D \left(\widehat{D}^T \widehat{D}\right) + V^{ort} \tag{7.41}$$

which is a linear-in-D orthogonality PM equation (OPM2). The measurement update stage, which is designed using Eq. (7.41), is readily formulated using the MKF algorithm. We choose V^{ort} as a zero-mean white sequence with 9 × 9 covariance matrix $\mu^{ort} I_9$, and write the reduced covariance filter equations as follows.

$$S_{k+1} = \left(\widehat{D}^T \widehat{D}\right) P_{k+1/k} \left(\widehat{D}^T \widehat{D}\right) + \mu^{ort} I_9 \tag{7.42}$$

$$K_{k+1} = P_{k+1/k} \left(\widehat{D}^T \widehat{D}\right) S_{k+1}^{-1} \tag{7.43}$$

$$D^*_{k+1/k+1} = \widehat{D} \left[I_3 + \left(I_3 - \widehat{D}^T \widehat{D}\right) K^T_{k+1}\right] \tag{7.44}$$

$$P^*_{k+1/k+1} = \left[I_3 - K_{k+1} \left(\widehat{D}^T \widehat{D}\right)\right] P_{k+1/k+1} \left[I_3 - K_{k+1} \left(\widehat{D}^T \widehat{D}\right)\right]^T + \mu^{ort} K_{k+1} K^T_{k+1} \tag{7.45}$$

In Eq. (7.42) to Eq. (7.45), $\widehat{D}$ denotes the best available estimate of D_k at time t_k, that is $\widehat{D}_{k/k}$. The state estimate update equation (7.44) results from the following manipulation on the original update equation, as shown next.

$$\begin{aligned} D^*_{k+1/k+1} &= \widehat{D} + \left[\widehat{D} - \widehat{D}\left(\widehat{D}^T\widehat{D}\right)\right] K^T_{k+1} \\ &= \widehat{D} + \widehat{D}\left(I_3 - \widehat{D}^T\widehat{D}\right) K^T_{k+1} \\ &= \widehat{D}\left[I_3 + \left(I_3 - \widehat{D}^T\widehat{D}\right) K^T_{k+1}\right] \end{aligned}$$

In the case of a very high μ^{ort}, the gain K_{k+1} is close to zero, which yields an almost unchanged estimate in Eq. (7.44). On the other hand using a very low μ^{ort} in Eqs. (7.42) to (7.44) yields the following estimate, $D^*_{k+1/k+1} = \widehat{D}^{-T}_{k+1/k+1}$. Note that the iterative brute-force PM (IBF) appears to be a special case of OPM2 where the gain is chosen as constant as shown next. Inserting $K_{k+1} = 0.5\, I_3$ into Eq. (7.44) yields

$$\begin{aligned} D^*_{k+1/k+1} &= \widehat{D}\left[I_3 + \left(I_3 - \widehat{D}^T\widehat{D}\right) 0.5\, I_3\right] \\ &= \widehat{D}\left(1.5\, I_3 - 0.5\widehat{D}^T\widehat{D}\right) \end{aligned}$$

which is identical to Eq. (7.31).

7.5 Augmented State-Matrix Mathematical Model

In this section we show how to handle the case of biased gyro outputs in the framework of a matrix Kalman filter. It is assumed that the gyro output is described by

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}^o_k - \mathbf{c}_k - \boldsymbol{\epsilon}_k \tag{7.46}$$

where $\boldsymbol{\omega}^o_k$ is the true angular velocity vector, $\mathbf{c}_k$ is the 3×1 vector of the gyro biases, and $\boldsymbol{\epsilon}_k$ is assumed to be a zero-mean white noise sequence with covariance matrix Q^ϵ_k. We moreover assume that the bias vector, $\mathbf{c}_k$, is time invariant. As usual, in order to estimate a constant vector via a Kalman filter we model its dynamics as a random process driven by a white noise as follows

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \boldsymbol{\eta}_k \tag{7.47}$$

where $\boldsymbol{\eta}_k$ is assumed to be a zero-mean white noise sequence with covariance matrix Q^η_k. (The addition of η_k is needed for filter stability). The process equation of the DCM was derived in a preceding section under the assumption of gyro white noises. To develop the DCM process equation here we proceed in a similar manner. In fact we just have to substitute the expression $\mathbf{c}_k + \boldsymbol{\epsilon}_k$ for $\boldsymbol{\epsilon}_k$ in Eqs. (7.5) and (7.6). This yields

$$D_{k+1} = \Phi_k D_k + [\mathbf{c}_k \times]\, D_k\, \Delta t + \mathcal{E}_k \tag{7.48}$$

where $\mathcal{E}_k \stackrel{\triangle}{=} [\boldsymbol{\epsilon}_k \times]\, D_k\, \Delta t$. To complete the state-space model of the system we recall the vector measurement equation (7.7)

$$\mathbf{b}_{k+1} = D_{k+1}\, \mathbf{r}_{k+1} + \mathbf{v}_{k+1} \tag{7.49}$$

where $\mathbf{v}_{k+1}$ is zero-mean white noise sequence with covariance matrix R_{k+1}. It is assumed that the sequences $\boldsymbol{\epsilon}_k$, η_k, and $\mathbf{v}_k$ are uncorrelated with one another and with the initial state variables D_0, and $\mathbf{c}_0$.

We wish to embed the estimation of the DCM and the constant bias vector in the framework of the MKF. To meet this end, we generalize the known technique of *state augmentation* to the given matrix state-space model (Eqs. 7.47, 7.48, and 7.49). We begin by defining the following 3×4 augmented state-matrix, X_k.

$$X_k \stackrel{\triangle}{=} \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} \tag{7.50}$$

Then we state the main results through the two following propositions, which are proved, respectively, in Appendix F.4 and Appendix F.5.

Proposition 7.5.1. *The dynamics model equations of D_k and $\mathbf{c}_k$ (Eqs. 7.47 and 7.48) are equivalent to the following augmented 3×4 matrix difference equation*

$$X_{k+1} = \sum_{r=1}^{5} \Theta_k^r\, X_k\, \Psi_k^r + W_k \tag{7.51}$$

where the noise matrix, W_k, is defined as

$$\mathcal{E}_k \stackrel{\triangle}{=} [\boldsymbol{\epsilon}_k \times]\, D_k\, \Delta t \tag{7.52a}$$

$$W_k \stackrel{\triangle}{=} \begin{bmatrix} \mathcal{E}_k & \boldsymbol{\eta}_k \end{bmatrix} \tag{7.52b}$$

and $\boldsymbol{\epsilon}_k$, η_k *are zero-mean white noise sequences with covariance matrices* Q_k^{ϵ} *and* Q_k^{η}. *The dynamics matrices,* Θ_k^r, Ψ_k^r, *in Eq.* (7.51) *are defined as*

$$\begin{aligned}
\Theta_k^1 &\triangleq \Phi_k & \Psi_k^1 &\triangleq E^{11} + E^{22} + E^{33} \\
\Theta_k^2 &\triangleq I_3 & \Psi_k^2 &\triangleq E^{44} \\
\Theta_k^3 &\triangleq -\left[\mathbf{d}_{k,1}\times\right] & \Psi_k^3 &\triangleq E^{41}\,\Delta t \\
\Theta_k^4 &\triangleq -\left[\mathbf{d}_{k,2}\times\right] & \Psi_k^4 &\triangleq E^{42}\,\Delta t \\
\Theta_k^5 &\triangleq -\left[\mathbf{d}_{k,3}\times\right] & \Psi_k^5 &\triangleq E^{43}\,\Delta t
\end{aligned} \tag{7.53}$$

where $\mathbf{d}_{k,j}$, $j = 1, 2, 3$, *are the three columns of the state matrix* D_k, *and each* E^{ij} *denotes a* 4×4 *matrix with* 1 *at position* (ij) *and* 0 *elsewhere.*

Proposition 7.5.2. *The measurement model equation for* D_{k+1} *(Eq. 7.49) is equivalent to the following measurement equation for the augmented state matrix* X_{k+1}.

$$\mathbf{b}_{k+1} = X_{k+1}\,\mathbf{h}_{k+1} + \mathbf{v}_{k+1} \tag{7.54}$$

where the 4×1 *vector* $\mathbf{h}_{k+1}$ *is defined as*

$$\mathbf{h}_{k+1} \triangleq \begin{bmatrix} \mathbf{r}_{k+1} \\ 0 \end{bmatrix} \tag{7.55}$$

Because the sequences $\mathcal{E}_k$ and $\boldsymbol{\eta}_k$ are uncorrelated, the covariance matrix of the augmented noise matrix, W_k, is readily expressed by the following 12×12 diagonal block matrix.

$$\operatorname{cov}[W_k] = \begin{bmatrix} \operatorname{cov}[\mathcal{E}_k] & O_{3\times 3} \\ O_{3\times 9} & Q_k^{\eta} \end{bmatrix} \tag{7.56}$$

The expression for $\operatorname{cov}[\mathcal{E}_k]$ is given in Eqs. (7.8) and (7.9).

The matrix state-space model of the augmented system, which features a state-matrix including the DCM and the gyro biases, is described in Propositions 7.5.1 and 7.5.2. The process equation (7.54) is not linear because the matrix coefficients Θ_k^r, $r = 3, 4, 5$, are functions of elements of D_k. One way to overcome this difficulty is to linearize the model and to implement the linearized model in an Extended Kalman filter. Another technique, the one that is adopted here, is to substitute $\widehat{D}_{k/k}$ for D_k in the process equation. This substitution yields a pseudo-linear process equation. Since the measurement equation is linear in the state, the matrix Kalman filter can operate on this matrix plant. The formulation of the MKF algorithm is straightforward and is omitted here for the sake of brevity.

7.6 Simulation Study

For this simulation study we considered a spacecraft rotating with respect to an inertial reference frame with the following angular velocity expressed in body coordinates:

$$\boldsymbol{\omega}(t) = 0.2\sin(2\pi t/150)[1, -1, 1]^T \quad [\text{rad/sec}] \tag{7.57}$$

It was assumed that a triad of body-mounted gyros measured this $\boldsymbol{\omega}(t)$ at a sampling rate of 10 Hz with a zero-mean Gaussian white noise having a standard deviation of 0.2 deg/hr, which is a large value compared with commonly used gyros on-board spacecrafts. The vector observation rate was chosen to be 10 Hz too. The measurement noise was a zero-mean Gaussian white noise. Its equivalent angular standard deviation was $\sigma_b = 100\,\text{arcsec}$ (typical of star tracker sensors accuracy). The tuning parameters for the first-type Orthogonality Pseudo-Measurement and for the second-type Orthogonality Pseudo-Measurement, respectively, (OPM1) and (OPM2), were chosen as $6\,\sigma_b^2$. Using these data we ran 100 Monte-Carlo simulation runs, where each simulation run lasted 20 sec.

Five algorithms were implemented. The first algorithm is the *nominal* algorithm; namely, the reduced covariance filter without orthogonalization. The four other algorithms consist of the nominal algorithm to which are added, respectively, the Optimal Brute-Force estimator (OBF), the Iterative Brute-Force estimator (IBF), the OPM1 estimator, and the OPM2 estimator. The performance of the five estimators were compared using two figures of merit, which are defined by

$$J_c(k) \triangleq \|D_k - D^*_{k/k}\|_F \tag{7.58}$$

$$J_o(k) \triangleq \|\, I_3 - (D^*_{k/k})^T D^*_{k/k}\|_F \tag{7.59}$$

where D_k is the true state matrix and $D^*_{k/k}$ is the estimated matrix after orthogonalization. (When normalization was not performed $D^*_{k/k}$ was just $\widehat{D}_{k/k}$.) Clearly, $J_c(k)$ is an attitude determination convergence criterion, while $J_o(k)$ is an orthogonality convergence criterion.

Table 7.1 shows the Monte-Carlo means of $J_c.10^5$ for the five algorithms at the final time. Figures 7.1-a and 7.1-b present the time histories of the Monte-Carlo means of J_c for all five cases. The case "OPM" stands for both algorithms OPM1 and OPM2 since the simulation results were almost identical for the two algorithms. Figure 7.1-a shows the transient phase, and Figure 7.1-b shows the steady-state phase. After the transients which lasted approximately two seconds, the four plots reach steady-state behaviors with small decay rates. As evidenced in Figure 7.1-a, during the transient phase, OPM achieves a better performance than all other algorithms. In particular, OBF and IBF show relatively high picks during the first samples. These picks illustrate the non-smooth character of the orthogonalization procedures. On the contrary, the OPM estimators, which were designed to be embedded in the Matrix Kalman filter, produce monotonously decreasing and smoother estimates. From Figure 7.1-b, we see that OBF yields the lower bound of all four solutions while the nominal algorithm yields the upper bound. As expected, IBF converges asymptotically to OBF. The performance of OPM remains on an intermediary steady-state level.

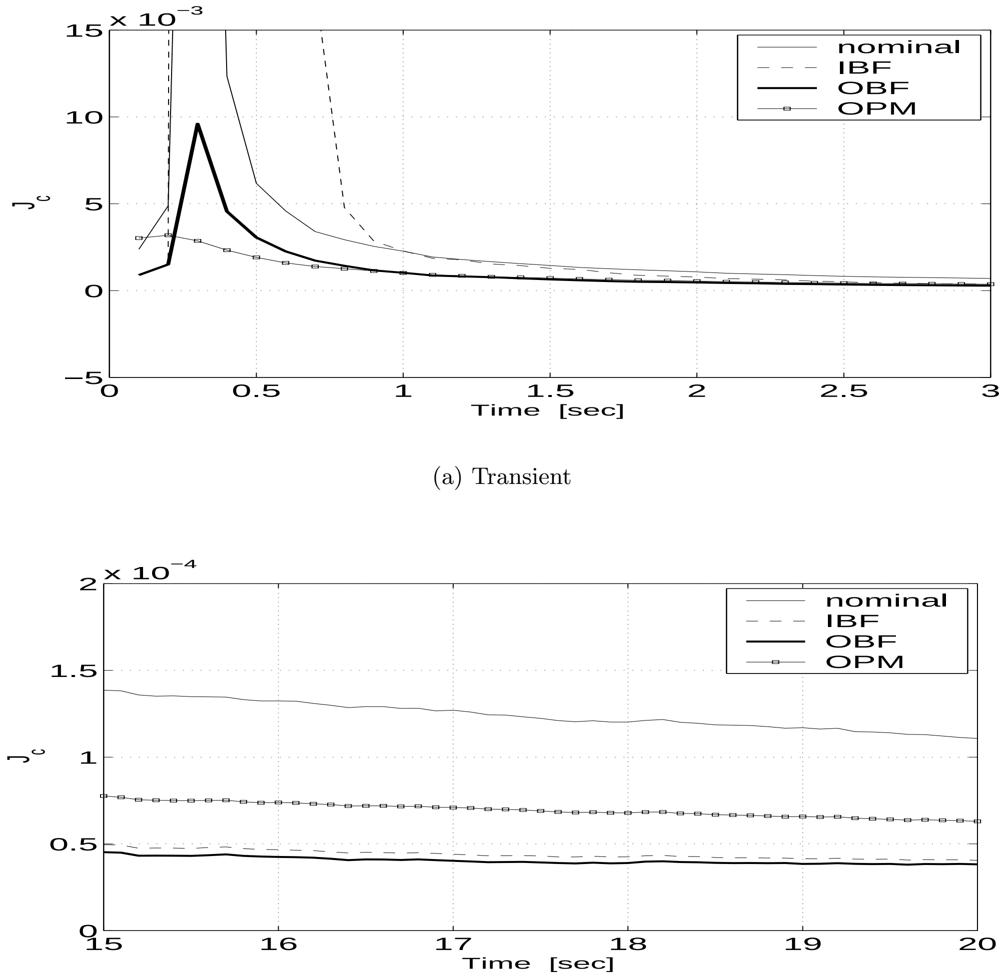
Table 7.1: Monte-Carlo means of $J_c \cdot 10^5$ at the final time

nominal	OBF	IBF	OPM
10	3	3	6

Table 7.2: Monte-Carlo means of J_o at the final time

nominal	OBF	IBF	OPM
5.10^{-4}	10^{-30}	10^{-15}	10^{-4}

Table 7.2 shows the Monte-Carlo means of J_o at the final time. Figure 7.2 depicts the time variations of the Monte-Carlo means of J_o during the simulation time span. Clearly, OBF achieves orthogonality with a numerical error accuracy. The IBF algorithm yields an estimate that is orthogonal for all practical purpose. The nominal algorithm, as expected, yields an estimate which becomes more and more orthogonal, as time increases. The



(a) Transient

(b) Steady-State

Fig. 7.1: Monte-Carlo means of the estimation convergence index J_c.

OPM filters produce better orthogonal estimates than the nominal filter, which in return explains the improvement in the convergence performance observed in Figure 7.1.

7.7 Summary and Concluding Remarks

Several matrix Kalman filters estimating the attitude matrix from vector observations were presented. The highlights of these filters are that they preserve the natural matrix notation of the state-space system model on which they operate. The special case of gyro

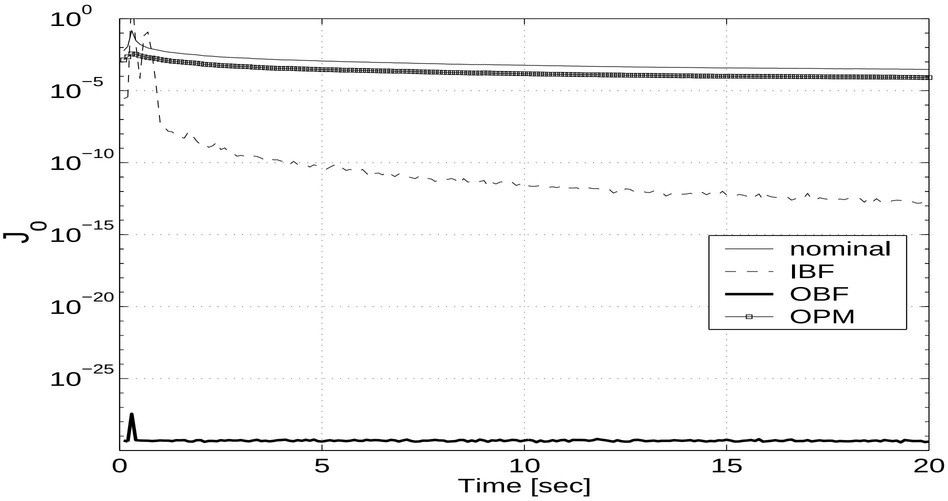


Fig. 7.2: Monte-Carlo means of the orthogonality index J_o.

white noise was considered. It was shown how to handle the case of biased gyros by a state-matrix augmentation technique. Two algorithms for the covariance computation were proposed: a full (9 × 9) covariance computation and a reduced (3 × 3) covariance computation. The latter was based on special but important assumptions on the noises stochastic models. The reduced covariance filter lowers significantly (9 times) the computation load and still yields satisfying performance as shown by extensive Monte-Carlo simulations.

Four different procedures were added to the reduced covariance filter in order to enforce on-line orthogonalization of the state matrix estimate. The first two procedures, named OBF and IBF, use known results. The other two, named OPM1 an OPM2, feature novel applications of the pseudo-measurement technique.

The design of the filters and the simulation results pointed at some advantages and disadvantages of the various orthogonalization procedures as summarized in Table 7.3. The iterative method (IBF) is the best trade-off between steady-state performance and low computation load. The pseudo-measurement techniques produce smooth transients with lower values than IBF, and enable a rigorous filter covariance computation.

As a conclusion, the matrix Kalman filter is a convenient tool for developing and analyzing Kalman filters of the attitude matrix. It is recommended to use the reduced

Table 7.3: Qualitative Comparison of Orthogonalization Techniques

	Advantages	Disadvantages
OBF	optimal closed-form	high computation load bad transient biased estim. error
IBF	low computation load asymptotically optimal	bad transient biased estim. error
OPM (1 and 2)	smooth transient unbiased estim. error covariance display	suboptimal needs tuning

covariance filter together with an orthogonalization procedure. If ensuring a low estimation error transient is not critical, it is recommended to use the iterative method (IBF). If, however, a smooth and low value transient is needed, a good practice is to start with a PM procedure until the filter reaches its steady-state and then switch to IBF.

Chapter 8

Discussion

In this chapter we compare the various quaternion optimal estimators; first, on a qualitative basis, then on a quantitative one, supported by the results of MC-simulations. We then show the direct link between the novel quaternion Kalman filter developed in this work and the classical **q**-method. Finally, a discussion of the relative merits of the matrix Kalman filters used here for attitude estimation concludes this chapter.

8.1 Qualitative Comparison of the Quaternion Optimal Estimators

Four algorithms for optimal estimation of the rotation quaternion were developed in this work in the framework of the general linear least squares estimation theory; namely, a novel quaternion Kalman filter (QKF), an optimized REQUEST algorithm (OPREQ), a quadratically-constrained batch-recursive least-squares estimator (BRLS), and a matrix Kalman filter of the K-matrix (KMKF). The highlights of the various filters are summarized in the following.

As any Kalman filter, the QKF algorithm has a probabilistic foundation. It filters optimally the measurement and the propagation noises according to their covariance matrices, which are analytically expressed in the previous chapters. The inclusion of parameters other than attitude in the filter is straightforward using the known state-augmentation technique. Even a single new vector observation at a sampling time point contributes to the updating of the quaternion estimate. The upgrade of the QKF algorithm to an adaptive filter is relatively easy.

The OPREQ algorithm was developed by optimizing the fading memory factor of REQUEST according to a special probabilistic estimation performance index. It uses analytical expressions for the noises uncertainty matrices, and optimally filters measurement as well as propagation noises. The unity constraint on the norm of the quaternion estimate is optimally satisfied. The update of the K-matrix may be formulated using a single vector observation. Since OPREQ is based on the **q**-method, it needs no a priori quaternion estimate. Separate techniques were devised for measurement and process noise adaptive estimation.

The filter KMKF has all the qualities of OPREQ except the ability of operating with a single new vector observation. The KMKF implements a matrix gain coefficient in the measurement update stage, and performs rigorous computation of the estimation error covariance in order to compute the gain matrix.

The BRLS estimator is optimal when filtering measurement as well as propagation noises. It handles the case of biased gyros through a smoothing procedure, which assumes a priori knowledge of the constant biases. The normalization stage of the quaternion estimate is optimally formulated in that filter. The BRLS filter has the virtue of being very computationally efficient.

The above analysis is summarized in Table 8.1. Each column refers to a specific property: 1-Probabilistic basis[1], 2-Optimal filtering of measurement and propagation noises, 3-Ease of estimation of parameters other than attitude, 4-Optimal quaternion estimate normalization, 5-Single new vector observation, 6-Low computation load.

When considering the four quaternion estimators in their nominal version; that is, without the adaptive or smoothing procedures, we notice that they all operate in a two-stage mode. In the first stage of OPREQ and KMKF the vector observations are prefiltered via the linear optimal filtering of the K-matrix. Then, the estimated K-matrix is processed in a **q**-method-based algorithm to yield the optimal quaternion. The BRLS algorithm operates differently. It first processes the single-frame vector observations to estimate the

[1] By *probabilistic basis* we mean that the estimator is optimal according to the probabilistic expectation of a given cost function.

Table 8.1: Properties of the Quaternion Optimal Filters

	1	2	3	4	5	6
QKF	X	X	X		X	
OPREQ	X	X		X	X	
KMKF	X	X	X	X		
BRLS		X		X		X

corresponding single-frame quaternion and, in the second stage, operates recursively on these estimates to yield a time-varying normalized quaternion. In QKF the first stage only processes the vector observations to compute the pseudo-measurement matrix. In the second stage, a Kalman filter operates on the pseudo-measurement model. The effort of unifying the two quaternion estimation approaches, that is, the **q**-method based approach and the Kalman filtering approach, led to the design of two-stages filters, where each stage features one of the two approaches.

8.2 Quantitative Comparison of the Attitude Optimal Estimators

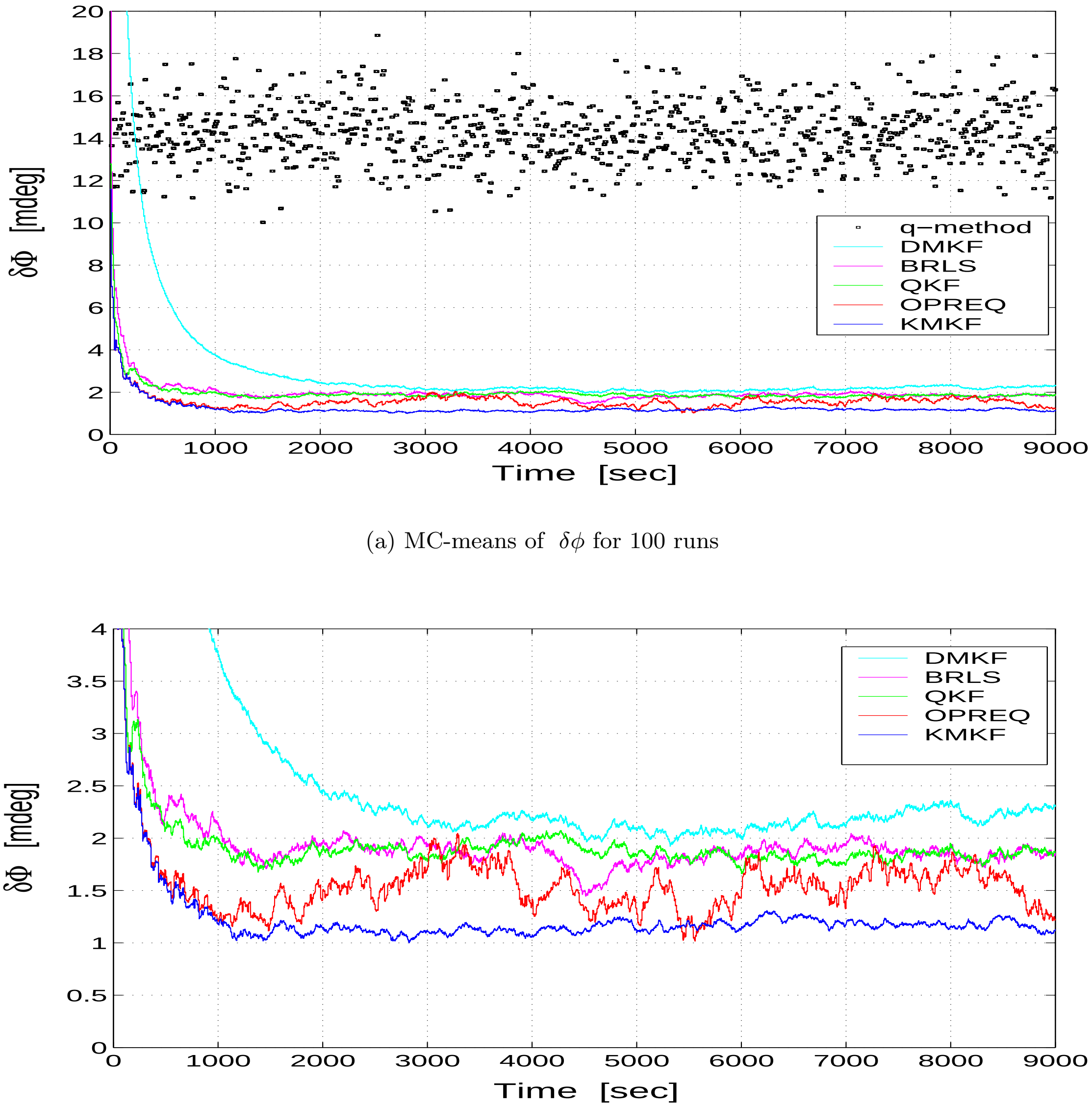
A quantitative comparison between OPREQ and KMKF was performed in Chapter 6 on the basis of extensive Monte-Carlo simulation results. It was shown that the steady-state performance of KMKF was superior to that of OPREQ. This confirmed the assertion that the special measure of uncertainty that was used in OPREQ was more conservative than the classical estimation error covariance. The analysis of KMKF, which was easily done thanks to the compact notation of the state matrix Kalman filter, showed under what conditions the KMKF algorithm becomes the OPREQ algorithm. That analysis also yielded a quantitative explanation for the discrepancy between the performance of both filters. Namely, if the gain of KMKF is chosen as a scalar, if the rows of the measurement noise matrix are independently identically distributed, and if their covariance matrix is multiplied by 4, then both algorithms are identical.

In the following we extend the quantitative comparison to the other attitude optimal estimators. In order to do so, the various quaternion optimal estimators; namely, QKF, OPREQ, KMKF, and BRLS, were run with the same simulation scenario. The estimator of the DCM; namely, the DMKF, was also implemented using the same simulation data. We used the nominal versions of the filters; that is, we did not implement any adaptive algorithm, or bias estimation procedure in the simulation. In addition, the classical single-frame **q**-method was implemented and compared with these attitude sequential filters. The simulation scenario was the same as that of section 6.7 and is described next for the sake of convenience. The simulated spacecraft kinematics was similar to that of the Maximum Anisotropy Probe (MAP) [62]. Two Cartesian coordinate frames were considered; namely, the moving body frame, $\mathcal{B}$, and the inertial Sun frame, $\mathcal{S}$. The rotation of $\mathcal{B}$ with respect to $\mathcal{S}$ consisted of a spin rotation and a nutation; the spin and the nutation rates were 0.464 rpm and 1 rph, respectively; the constant nutation angle, which is defined between the spacecraft spinning axis and the anti-Sun line of sight, was equal to 22.5 deg. The attitude measurement devices simulated here were composed of a digital sun sensor (DSS), an autonomous star-tracker (AST), and a triad of rate gyroscopes. We considered here the case where the AST observed the same star during the whole simulation. Therefore, two identical inertial directions were observed at each sampling time; namely, the Sun direction, and the direction to that star. These directions were represented in $\mathcal{S}$ by the unit vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively, where $\mathbf{r}_1 = [0\,0\,1]^T$ and $\mathbf{r}_2 = [1\,0\,0]^T$. The unit vector measurements, $\mathbf{b}_i$, $i = 1, 2$, were simulated by adding a small zero-mean white Gaussian noise to the ideal observed directions, and by normalizing the result; that is $\mathbf{b}_i = (A\,\mathbf{r}_i + \boldsymbol{\delta b}_i)/\|A\,\mathbf{r}_i + \boldsymbol{\delta b}_i\|$ where A is the correct transformation matrix from the Sun to the body coordinates. The standard deviations of the DSS and of the AST were, respectively, $\sigma_1 = 1$ arcmn, ($\simeq 17\,$mdeg), and $\sigma_2 = 10$ arcsec, ($\simeq 3\,$mdeg). The vector measurement sampling period was 10 seconds. The output of the triad of gyroscopes was contaminated by a zero-mean Gaussian white noise with a covariance matrix $\sigma_\epsilon^2\,I_3$, where $\sigma_\epsilon = 100\,$mdeg/hr. The gyros sampling period was 0.5 second. These features are summarized in Table 8.2. In the algorithms OPREQ and KMKF, the measured K-matrices, δK_k, $k = 0, 1, \ldots$, were computed using the weights $\delta m_k = 1/\sigma_1^2 + 1/\sigma_2^2$, where $\sigma_1 = 1$ arcmn and $\sigma_2 = 10$ arcsec. In the algorithm BRLS, the measurement

Table 8.2: Sampling Times and Noises Stochastic Models

	T_s	σ	Covariance Matrix Expression
Gyro	0.5 sec	100 mdeg/hr	$Q^\epsilon = \sigma_\epsilon^2 \, I_3$
DSS	10 sec	1 arcmn	$R_1 = \sigma_1^2 \, I_3$
AST	10 sec	10 arcsec	$R_2 = \sigma_2^2 \, I_3$

weighting coefficients, α_k were chosen to be equal to $1/\,\sigma_1^2{+}1/\,\sigma_2^2$, and the process weighting coefficients were chosen equal to $1/(\,\sigma_\epsilon\, T_s^{GYRO})^2$, where T_s^{GYRO} is the gyro sampling time (0.5 sec). The implemented DMKF consists of the reduced covariance algorithm and includes the first type of orthogonality pseudo-measurement (see section 7.4.3), where the pseudo-measurement variance was chosen as $\mu_{1/2} = 8\,\sigma_{1/2}^2$. The initial true attitude was represented by the quaternion $\mathbf{q}(0) = [1\ -1\,0\,1]^T/\sqrt{3}$, or by the corresponding DCM. In QKF and DMKF the filters were initialized as follows: $\widehat{\mathbf{q}}(0) = [\mathbf{q}(0) + \boldsymbol{\Delta q}(0)]\,/\|\mathbf{q}(0) + \boldsymbol{\Delta q}(0)\|$, where $\|\boldsymbol{\Delta q}(0)\| \simeq 10^{-2}$. This initial estimation error was chosen for the sake of a fair comparison with the filters that do not require an initial estimate; namely, OPREQ, KMKF, and the BRLS filter. The implemented single-frame **q**-method, used for reference, consisted of extracting the optimal quaternion from the measured K-matrices at each sampling time t_k, *without blending it with the a priori estimate at* t_k. Each simulation run lasted 9000 sec. We ran 100 simulations for each filter and computed the Monte-Carlo means (MC-mean) and Monte-Carlo standard deviations (MC-sigma) of the angular error, $\delta\phi$. The graphs of the MC-means of $\delta\phi$ for the various filters are showed on Fig. 8.1. It is clearly seen that the KMKF provides the lowest mean value of $\delta\phi$, about 1.2 mdeg in the steady-state. As already seen in Chapter 6, the performance of the OPREQ algorithm in steady-state is almost always worse than that of the KMKF. Nevertheless the OPREQ filter outperforms the filters QKF and BRLS. These two filters show similar performance. The latter can be explains by two facts. First, the noise covariance matrices are scalar matrices (see Tab. 8.2), which justifies to some extent the choice of scalar weights in the BRLS algorithm. Second, the BRLS algorithm includes *two* filtering stages in contrast to the QKF that includes only one. When considering only the performance of the attitude filters, the DMKF provides the worst results; namely, it has the slowest transient and



(a) MC-means of $\delta\phi$ for 100 runs

(b) Zoom on the MC-means of $\delta\phi$ for the sequential filters

Fig. 8.1: Comparison of the optimal attitude estimators.

the highest steady-state ($\simeq$ 2.2 mdeg). Notice that the performance of the DMKF with iterative orthogonalization (which is not shown here for brevity), is not significantly better. Nevertheless, the performance of any of the proposed time-varying filters is still better than that of the **q**-method. As seen in Fig. 8.1-a the steady-state of $\delta\phi$ in the **q**-method is about 14 mdeg. This was a foreseen result since the **q**-method is a single-frame algorithm, where the past information is forgotten rather than being utilized in order to improve the current estimate. The MC-means and MC-sigmas results are summarized in Table 8.3. The featured values are time averages of the steady-state Monte-Carlo values. Figure 8.1-b shows a zoom on the MC-means of the error $\delta\phi$ for the sequential filters. As seen from Fig. 8.1-b and from Table 8.3, these means are by an order of magnitude less than that of the single-frame **q**-method. The values of the MC-sigmas are consistent with those of the MC-means; namely, the KMKF is the most accurate and the **q**-method is the far less accurate algorithm (around its respective mean). Notice that the ratio between the standard deviations of the KMKF and the OPREQ algorithm is almost equal to 1.44. This agrees with the scalar analysis in section 6.8 that predicts a factor of $\sqrt{2}$. The standard deviations in the OPREQ and the QKF algorithms are similar (0.85 mdeg and 0.88 mdeg, respectively) although, as mentioned, the OPREQ includes two filtering stages which should yield a better accuracy. Actually, the QKF compensates for the single filtering stage by utilizing the variance cost function, which is more efficient than that used in the OPREQ filter. The BRLS filter yields a lower standard deviation (0.82 mdeg) than the OPREQ filter. This indicates that the BRLS algorithm is well suited to the implemented noise stochastic models. The standard deviation in the DMKF is the highest among the filters, which shows, in this example, that the kinematics model using the DCM is less observable than that using the quaternion. A further investigation of this issue would need the use of other kinds of reference vector ($\mathbf{r}_i$) geometries.

Table 8.3: Steady-state Monte-Carlo Means and Standard Deviations of $\delta\phi$ for 100 runs

	$\overline{\delta\phi}$ [mdeg]	$\overline{\sigma}$ [mdeg]
KMKF	1.2	0.59
OPREQ	1.5	0.85
QKF	1.9	0.88
BRLS	1.9	0.82
DMKF	2.2	1.00
q-method	14.2	9.10

8.3 The QKF Algorithm and the q-Method

The QKF algorithm, although derived in the framework of Kalman filtering, can be seen as a mechanization of the **q**-method. Consider the H-matrix, as given in Eqs. (2.33a) to (2.33c), which is computed using a single pair (**b**, **r**), and let **q** denote an eigenvector of H that belongs to its non-empty Null-space; that is,

$$H\mathbf{q} = \mathbf{0} \tag{8.1}$$

Equation (8.1) is the normal equation of the following constrained optimization problem.

$$\min_{\mathbf{q}^T\mathbf{q}=1} \left\{ f\left(\mathbf{q}\right) = \mathbf{q}^T H^T H \mathbf{q} \right\} \tag{8.2}$$

As shown in Appendix G.1, the H-matrix and the K-matrix of the **q**-method, as given in Eqs. (1.5) and (1.6), are related by

$$H^T H = \frac{1}{2}\left(I_4 - K \right) \tag{8.3}$$

Substituting Equation (8.3) into Eq.(8.2), and developing, yields after some development

$$\begin{aligned} \min_{\mathbf{q}^T\mathbf{q}=1} \left\{ f\left(\mathbf{q}\right) \right\} &= \min_{\mathbf{q}^T\mathbf{q}=1} \left\{ \mathbf{q}^T \left(I_4 - K \right) \mathbf{q} \right\} \\ &= \min_{\mathbf{q}^T\mathbf{q}=1} \left\{ 1 - \mathbf{q}^T K \mathbf{q} \right\} \end{aligned} \tag{8.4}$$

From Eq. (8.4), we realize that minimizing $f\left(\mathbf{q}\right)$ over the unit sphere in $\mathbb{R}^4$ is equivalent to minimizing Davenport's cost function as given in Eq. (1.7). Since the constrained QKF operates on the pseudo-measurement model, which is derived from Eq. (8.1), we deduce that it solves the optimization problem in Eq. (8.2), and, thus, the problem of Eq. (8.4). Therefore, the constrained QKF appears as a recursive form of the **q**-method with additional qualities, like the ability of implementing matrix weights instead of scalar weights, and the property of filtering optimally the propagation noises according to their covariance matrix.

8.4 State Matrix Kalman Filter

The state matrix Kalman filter (MKF) presented in Chapter 5 is a general Kalman filter that handles naturally linear plants which are described by a matrix difference equation and a matrix measurement equation. The development of the algorithm tried to maintain the equilibrium between generalization of the filter purview and concise notation for aiding intuition and mathematical manipulation. It was shown that the MKF is a natural extension of the conventional Kalman filter. It was also proved that the MKF includes as special cases other matrix filters that were developed in the past.

The implementation of the pseudo-measurement technique for constrained estimation of a matrix process is straightforward in the framework of the MKF. This is so because the MKF can handle any linear matrix constraint equation as a matrix pseudo-measurement equation.

The MKF is a general filter for linear matrix discrete-time plants. It can operate on non-linear models, but becomes then a pseudo-linear MKF. An extension of the MKF to non-linear plants through linearization techniques would involve matrix differentiation tools, and is beyond the scope of the present work.

Two applications of the MKF were presented in Chapters 6 and 7; namely, the KMKF, for estimation of the K-matrix, and the DMKF, for estimation of the DCM. The general notation of the state matrix Kalman filter handles the case of matrix measurements (KMKF) as easily as it handles the standard case of vector measurements (DMKF). The constraints of symmetry, zero-trace, or orthogonality of the state-matrices are easily incorporated in the filters as pseudo-measurement models. Special important cases for the noise stochastic models were considered. This led to the simplification of the filters covariance computation via a straightforward manipulation of the MKF equations.

The filters KMKF and DMKF are both attitude estimation algorithms. The filter DMKF, however, directly estimates the attitude matrix, while, as mentioned before, the KMKF works in a two-stage mode to estimate the quaternion. KMKF needs no a priori estimate of the quaternion, while DMKF needs an initial DCM estimate. DMKF can process a single new vector measurement to update the attitude. On the other hand, the same set of at least two non-collinear vector observations must be processed in KMKF. The KMKF presents linear constraints on the state-matrix estimate. In DMKF, the constraint is non-linear. Extensive simulations showed that, under nominal conditions, KMKF is relatively insensitive to the constrained estimation procedure. On the contrary, the performance of DMKF can be greatly enhanced through constrained estimation. The DMKF algorithm is computationally more efficient than the KMKF since it implements 9×9 matrix equations instead of the 16×16 equations of KMKF. As shown in the simulation described before the K-matrix state space model is more observable than the DCM state-space model. The latter is due to the specific relative orientation of the vector observations, as well as to the angular velocity of the rotating body. These geometric considerations should be addressed when choosing a particular matrix estimator of the attitude. While there is no general reason to prefer one algorithm over the other, the algorithms' user should be aware of their differences.

Chapter 9

Conclusions

The fusion of the two conventional quaternion estimation approaches, namely, the **q**-method based approach and the Kalman filtering approach, was achieved. This led to novel quaternion optimal filters that inherit the complementary qualities of both approaches.

It was shown that linear unconstrained Kalman filtering of the quaternion is possible.

The QKF algorithm can be interpreted as a recursive stochastic generalized **q**-method. Instead of the independent identically distributed measurement noises which are implicitly assumed in the classical **q**-method, one may considered arbitrary measurement noise covariance matrices in the filtering process.

Among the quaternion estimators, the novel Quaternion Kalman filter and the Optimal-REQUEST algorithm appear to exhibit satisfying trade-offs between performance and computation load.

The state-matrix Kalman filter is the appropriate algorithm for handling estimation of matrix plants. Its compact notation allows easy mathematical manipulation of the matrix equations.

Special, but important assumptions, on the filter noise stochastic models can yield simplified algorithm equations and substantially lower computation loads, while maintaining a satisfying level of performance.

Adaptive noise estimation can be a successful alternative to state augmentation in order to cope with unmodeled parameters.

Combining the pseudo-measurement technique with ad-hoc iterative procedures seems to be a promising technique for constrained Kalman filtering.

The issue of constrained Kalman filtering should, however, be considered with some care. It is indeed tempting to enforce the estimation algorithm to comply with some relationships when we know that these relationships exists among the true parameters. One must however keep in mind that forcing this additional feature has its cost, like a big overshoot in the error transient or an increase in the computational burden, and is not always efficient. This was clearly seen in the symmetrization of the estimated K-matrix in the KMKF algorithm. This was also the case in the novel quaternion Kalman filter, whose nominal version does not require the normalization of the quaternion estimate after each update. As mentioned earlier in this work, normalization of the quaternion still raises debates and controversies that are nourished by arguments on the error covariance properties. At least, most of the reported works seem to agree upon the necessity of normalizing the quaternion estimate. Using the novel quaternion KF, we conclude that quaternion normalization is *not* always necessary, and should anyway be considered in the context of the implemented model and the desired filter performance.

In the ensuing we summarize in a table the essential features of each proposed algorithm. The property of adaptive filtering is not included because it was not developed for all the algorithms. As an example, if the same set of at least two vector observations is acquired at each sampling time, then, as shown in the simulation of Chapter 8, the KMKF will be the best filter. If only a single vector observation is acquired at each sampling time, which is the case of a magnetometer, and if the noises are independently identically distributed, then OPREQ will be the adequate method. However, if the stochastic models are more complex, then the QKF will be appropriate. The purpose of that table is to remind the user of the basics of each filter, and guiding the user in choosing the most adequate filter.

Algorithm	Attitude Parameter	Noise Covariance Matrices		Gain	Performance Index	Initial Estimate	Covariance Computation	Type of Vector Measurement
		Q	R					
KMKF	**q**	Full	Full	matrix	Error variance + Weighted least-squares	no	Full: 16 x 16 Reduced: 4 x 4	At least Two, Same set
OPREQ	**q**	$\sigma_{\varepsilon}^{2}\ I_3$	$\sigma_{b}^{2}\ I_3$	scalar	Error variance + Weighted least-squares	no	4 x 4	Single, Different
QKF	**q**	Full	Full	matrix	Error variance	yes	Full: 16 x 16 Reduced: 4 x 4	Single, Different
BRLS	**q**	$\sigma_{\varepsilon}^{2}\ I_3$	$\sigma_{b}^{2}\ I_3$	scalar	Scalar Weighted least-squares	no	Least-squares costs: scalar	At least Two, Different
DMKF	**DCM**	Full	Full	matrix	Error variance	yes	Full: 9 x 9 Reduced: 3 x 3	Single, Different

Chapter 10

Recommendations For Further Work

A possible application of the state-matrix Kalman filter is in the field of non-linear optimization. In that field, Newton's method and the quasi-Newton methods play a central role. The quasi-Newton algorithms typically feature recursive deterministic estimators of the inverse Hessian matrix of the optimized cost function. Sometimes the data that is gathered to an optimization algorithm is corrupted by random errors, which may cause divergence of the method. It is thought that, under these conditions, the state-matrix Kalman filter is an adequate algorithm to be implemented in a quasi-Newton-like optimization algorithm, because it is a matrix estimator and it is, by design, an optimal filter of random noises.

We also recommend to further investigate the normalization issue in the QKF algorithm. Borrowing principles of constrained estimation from the Deterministic Parameter Estimation Theory, might be a better way to perform normalization, and to do convergence analysis.

Appendix A

A.1 Property of the Ξ-matrix

The property of the Ξ-matrix that is presented here has been recognized and used in previous works [26, 27]. (The symbol Ξ is inspired by the first paper.)

Given a 4×1 column-matrix $\mathbf{q}$, with vector part $\mathbf{e}$ and scalar part q, and given a 3×1 column-matrix $\mathbf{x}$, define the general 4×4 matrix X, and the 4×3 matrix Ξ as follows:

$$X \triangleq \begin{bmatrix} -[\mathbf{x}\times] & \mathbf{x} \\ -\mathbf{x}^T & 0 \end{bmatrix} \qquad \Xi \triangleq \begin{bmatrix} [\mathbf{e}\times] + qI_3 \\ -\mathbf{e}^T \end{bmatrix} \tag{A.1.1}$$

then the following identity can be proven by direct computation,

$$X\mathbf{q} = \Xi\mathbf{x} \tag{A.1.2}$$

A.2 Derivation of Eq. (2.71)

Given are three square matrices of the same dimension A, C, D, with $D \neq 0$ (not all the elements of D are zero), and a matrix function $B(\eta)$ of the real scalar η defined by $B(\eta) = C + \eta D$. Consider the following unconstrained minimization problem:

$$\min_{\eta} \left\{ J(\eta) = \|A - B(\eta)\|^2 \right\} \tag{A.2.1}$$

where $\|\cdot\|$ denotes the Frobenius norm; that is $\|\cdot\|^2 = tr(\cdot \cdot^T)$. We show that the solution of this problem, described in Eq. (A.2.1), is

$$\widehat{\eta} = \frac{tr\left[(A - C)\, D^T\right]}{tr(DD^T)} \tag{A.2.2}$$

We start by writing

$$\begin{aligned}
J(\eta) =& \|A - C - \eta D\|^2 \\
=& tr\left[(A - C - \eta D)\,(A - C - \eta D)^T\right] \\
=& \eta^2\, tr(DD^T) - \eta\, tr\left[(A-C)D^T + D(A-C)^T\right] \\
&+ tr\left[(A - C)\,(A - C)^T\right] \\
=& \eta^2\, tr(DD^T) - 2\eta\, tr\left[(A - C)\,D^T\right] \\
&+ tr\left[(A - C)\,(A - C)^T\right]
\end{aligned}$$

The first derivative of the cost function $J(\eta)$ is then

$$\frac{dJ}{d\eta} = 2\, tr(DD^T)\eta - 2\, tr\left[(A - C)\,D^T\right] \tag{A.2.3}$$

The necessary condition for $\widehat{\eta}$ to yield a minimum of J is $dJ/d\eta = 0$. This gives

$$\widehat{\eta} = \frac{tr\left[(A - C)\,D^T\right]}{tr\,(DD^T)} \tag{A.2.4}$$

Note that $D \neq 0$ implies that $tr(DD^T) \neq 0$, and, thus, Eq. (A.2.4) always has a solution. The second derivative of $J(\eta)$ is

$$\frac{d^2 J}{d\eta^2} = 2\, tr(DD^T) \tag{A.2.5}$$

It is well known that for any square matrix D the matrix DD^T is positive semi-definite, therefore its trace is non-negative. Since the matrix $D \neq 0$ then $DD^T \neq 0$, so that its trace is positive. Therefore, the sufficient condition for $\widehat{\eta}$ to be a minimum is satisfied.

Appendix B

B.1 Proof of Eq. (3.20)

Proof. The time subscripts are omitted for the sake of clarity. The present derivation is based on Taylor series development of the exponential matrices Φ and Φ^o to the second order in Δt [15, p. 165]

$$\Phi = I_4 + \Omega\,\Delta t + \frac{1}{2}\,\Omega^2\,\Delta t^2 + \mathcal{O}\left(\Delta t^3\right) \tag{B.1.1}$$

$$\Phi^o = I_4 + (\Omega - \mathcal{E})\;\Delta t + \frac{1}{2}\;(\Omega - \mathcal{E})^2\;\Delta t^2 + \mathcal{O}\left(\Delta t^3\right) \tag{B.1.2}$$

where

$$\mathcal{E} = \frac{1}{2}\begin{bmatrix} -\,[\epsilon\times] & \boldsymbol{\epsilon} \\ -\,\boldsymbol{\epsilon}^T & 0 \end{bmatrix} \tag{B.1.3}$$

Then, by definition of the error matrix $\Delta\Phi$, one obtains

$$\Delta\Phi \;=\; \mathcal{E}\,\Delta t \;+\; \frac{1}{2}\;(\Omega\,\mathcal{E} \;+\; \mathcal{E}\,\Omega)\;\Delta t^2 \;-\; \frac{1}{2}\;\mathcal{E}^2\,\Delta t^2 \;+\; \mathcal{O}\left(\Delta t^3\right) \tag{B.1.4}$$

In order to develop expressions for the two matrices $(\Omega\,\mathcal{E} \;+\; \mathcal{E}\,\Omega)$ and $\mathcal{E}^2$, some intermediary results are presented here.

Intermediary results:

$$\Omega\,\mathcal{E} \;=\; \frac{1}{4}\begin{bmatrix} [\boldsymbol{\omega}\times]\;[\epsilon\times] - \boldsymbol{\omega}\,\boldsymbol{\epsilon}^T & -\,[\boldsymbol{\omega}\times]\;\boldsymbol{\epsilon} \\ \boldsymbol{\epsilon}^T\;[\boldsymbol{\omega}\times] & \boldsymbol{\omega}^T\,\boldsymbol{\epsilon} \end{bmatrix} \tag{B.1.5}$$

and, using the skew-symmetric property of Ω and $\mathcal{E}$, it comes,

$$\mathcal{E}\,\Omega \;=\; (\Omega\,\mathcal{E})^T \tag{B.1.6}$$

Note that the matrices Ω and $\mathcal{E}$ are not commutative, so that, unlike the scalar case, the exponential of a sum of matrices is different from the product of single exponential matrices :

$$\exp\{(\Omega - \mathcal{E})\,\Delta t\} \neq \exp\{\Omega\,\Delta t\}\ \exp\{\mathcal{E}\}\ \Delta t \tag{B.1.7}$$

The following identities are used in the development,

$$\begin{aligned} [\boldsymbol{\omega}\times]\,[\epsilon\times] &\equiv \boldsymbol{\epsilon}\,\boldsymbol{\omega}^T - \boldsymbol{\epsilon}^T\,\boldsymbol{\omega}\,I_3 \\ [\epsilon\times]\,[\boldsymbol{\omega}\times] &\equiv \boldsymbol{\omega}\,\boldsymbol{\epsilon}^T - \boldsymbol{\omega}^T\,\boldsymbol{\epsilon}\,I_3 \\ [\epsilon\times]\,[\epsilon\times] &\equiv \boldsymbol{\epsilon}\,\boldsymbol{\epsilon}^T - \boldsymbol{\epsilon}^T\,\boldsymbol{\epsilon}\,I_3 \end{aligned} \tag{B.1.8}$$

Using the above results and identities yields

$$\begin{aligned} \Omega\,\mathcal{E} + \mathcal{E}\,\Omega &= -\tfrac{1}{2}\left(\boldsymbol{\omega}^T\boldsymbol{\epsilon}\right) I_4 \\ \mathcal{E}^2 &= -\tfrac{1}{4}\left(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right) I_4 \end{aligned} \tag{B.1.9}$$

Finally, to second order in Δt and in $\boldsymbol{\epsilon}$, the approximation to $\Delta\Phi$ is obtained.

$$\Delta\Phi \simeq \mathcal{E}\,\Delta t + \frac{1}{8}\left[-2\left(\boldsymbol{\omega}^T\boldsymbol{\epsilon}\right) + \left(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right)\right]\Delta t^2\, I_4 \tag{B.1.10}$$

Keeping the first order terms in Δt in Eq. (B.1.10) and in Eq. (B.1.1), that is

$$\Phi_k \simeq I_4 + \Omega_k\,\Delta t \tag{B.1.11}$$

$$\Delta\Phi_k \simeq \mathcal{E}_k\,\Delta t \tag{B.1.12}$$

and using Eqs.(B.1.11) and (B.1.12) in Eq.(3.17), which is rewritten here

$$W^o_k \triangleq -\Phi_k\,K^o_{k/k}\,\Delta\Phi^T_k - \Delta\Phi^T_k\,K^o_{k/k}\,\Phi^T_k + \Delta\Phi_k\,K^o_{k/k}\,\Delta\Phi^T_k \tag{B.1.13}$$

yields the first order approximation to W^o_k in Δt

$$W_k = \left(K_{k/k}\,\mathcal{E}_k - \mathcal{E}_k\,K_{k/k}\right)\Delta t \tag{B.1.14}$$

where $K_{k/k}$ was substituted for $K^o_{k/k}$ since in practice $K^o_{k/k}$ is not known. Recalling that the matrices $K_{k/k}$ and $\mathcal{E}_k$ are given by

$$K = \begin{bmatrix} S - \sigma\,I_3 & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix} \qquad \mathcal{E} = \frac{1}{2}\begin{bmatrix} -[\epsilon\times] & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & 0 \end{bmatrix} \tag{B.1.15}$$

(for clarity the subscripts were omitted in Eqs. B.1.15), then, another expression for W_k can be derived as shown next. Defining the following quantities,

$$\begin{aligned} B_\epsilon &\triangleq [\epsilon\times]\,B \\ S_\epsilon &\triangleq B_\epsilon + B_\epsilon^{\,T} \\ (\mathbf{z}_\epsilon\times) &\triangleq B_\epsilon - B_\epsilon \\ \sigma_\epsilon &\triangleq tr(B_\epsilon) \end{aligned} \qquad \text{(B.1.16)}$$

and using the identities

$$[\mathbf{u}\times]\,[\mathbf{v}\times] \equiv \mathbf{v}\mathbf{u}^T - \mathbf{v}^T\mathbf{u}\,I_3 \qquad \text{(B.1.17)}$$

and

$$\begin{aligned} \mathbf{u}^T\mathbf{v} &\equiv tr([\mathbf{u}\times]\,M) \\ [\mathbf{v}\times] &= M^T - M \end{aligned} \qquad \text{(B.1.18)}$$

then, straightforward computations yield

$$\begin{bmatrix} S_\epsilon - \sigma_\epsilon I_3 & \mathbf{z}_\epsilon \\ \mathbf{z}_\epsilon^T & \sigma_\epsilon \end{bmatrix} = K\,\mathcal{E} - \mathcal{E}K \qquad \text{(B.1.19)}$$

Thus, multiplying Eq. (B.1.19) by Δt, and using Eq. (B.1.14) leads to

$$W_k = \begin{bmatrix} S_\epsilon - \sigma_\epsilon I_3 & \mathbf{z}_\epsilon \\ \mathbf{z}_\epsilon^T & \sigma_\epsilon \end{bmatrix} \Delta t \qquad \text{(B.1.20)}$$

□

B.2 Computation of the Matrices $\mathcal{Q}_k$ and $\mathcal{R}_k$

In the following the best available estimate of a given quantity, $\mathbf{x}$ will be denoted by $\widehat{\mathbf{x}}$. The time subscripts will be generally omitted for the sake of clarity. In order to compute the matrices $\mathcal{Q}_k$ and $\mathcal{R}_k$ the following identities are necessary:

$$[\mathbf{u}\times]\,[\mathbf{v}\times] \equiv \mathbf{v}\mathbf{u}^T - \mathbf{u}^T\mathbf{v}\,I_3 \qquad \text{(B.2.1a)}$$

$$\mathbf{u}^T\,[\mathbf{v}\times] \equiv -\mathbf{v}^T\,[\mathbf{u}\times] \qquad \text{(B.2.1b)}$$

$$[\mathbf{u}\times]\,\mathbf{u} \equiv 0 \qquad \text{(B.2.1c)}$$

B.2.1 Computation of $\mathcal{Q}_k$

Recall that (see Eq. 3.38)

$$\mathcal{Q}_k \stackrel{\triangle}{=} E\left[W_k W_k^{\,T}\right] \tag{B.2.1.1}$$

Using Eq. (3.20) and considering the following partition of the 4×4 matrix $\mathcal{Q}_k$

$$\mathcal{Q}_k = \begin{bmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{21} & \mathcal{Q}_{22} \end{bmatrix} \Delta t^2 \tag{B.2.1.2}$$

where $\mathcal{Q}_{11}$ is a 3×3 submatrix, yields the following expressions for the submatrices $\mathcal{Q}_{11}$, $\mathcal{Q}_{12}$, $\mathcal{Q}_{21}$, and $\mathcal{Q}_{22}$.

$$\mathcal{Q}_{11} = E\left[(S_\epsilon - \sigma_\epsilon I)^2 + \mathbf{z}_\epsilon \mathbf{z}_\epsilon^T\right] \tag{B.2.1.3a}$$

$$\mathcal{Q}_{12} = E\left[S_\epsilon \mathbf{z}_\epsilon\right] \tag{B.2.1.3b}$$

$$\mathcal{Q}_{21} = \mathcal{Q}_{12}^{\,T} \tag{B.2.1.3c}$$

$$\mathcal{Q}_{22} = E\left[\mathbf{z}_\epsilon^T \mathbf{z}_\epsilon + \sigma_\epsilon^2\right] \tag{B.2.1.3d}$$

After some computation, Eqs. (B.2.1.3) yield

$$\mathcal{Q}_{11} = \eta_k \left\{\left[\widehat{\mathbf{z}}^T \widehat{\mathbf{z}} + \widehat{\sigma}^2 - tr\left(\widehat{B}\widehat{B}^T\right)\right] I_3 + 2\left[\widehat{B}^T\widehat{B} - \widehat{B}^2 - (\widehat{B}^T)^2\right]\right\} \tag{B.2.1.4a}$$

$$\mathcal{Q}_{12} = -\eta_k \left(\mathbf{y} + \widehat{B}^T \mathbf{z}\right) \tag{B.2.1.4b}$$

$$\mathcal{Q}_{21} = \mathcal{Q}_{12}^{\,T} \tag{B.2.1.4c}$$

$$\mathcal{Q}_{22} = \eta_k \left[tr\left(\widehat{B}\widehat{B}^T\right) + \widehat{\sigma}^2 + \widehat{\mathbf{z}}^T \widehat{\mathbf{z}}\right] \tag{B.2.1.4d}$$

where the 3×1 column vector $\mathbf{y}$ is defined as

$$\begin{aligned} M &\stackrel{\triangle}{=} \widehat{B}\left(\widehat{B} - \widehat{\sigma}\, I_3\right) \\ (\mathbf{y}\times) &\stackrel{\triangle}{=} M^T - M \end{aligned} \tag{B.2.1.5}$$

The derivation of the element $\mathcal{Q}_{22}$ is detailed next. The derivation of the other submatrices of $\mathcal{Q}_k$ follow a similar computation.

Computation of $\mathcal{Q}_{22}$

From Eq. (B.2.1.3d) we have

$$\mathcal{Q}_{22} = E\left[\mathbf{z}_\epsilon^T \mathbf{z}_\epsilon + \sigma_\epsilon^2\right] \tag{B.2.1.6}$$

Computation of $E\left[\mathbf{z}_\epsilon^T\mathbf{z}_\epsilon\right]$ It can be shown, using the identities Eqs. (B.2.1), that the vector $\mathbf{z}_\epsilon$ is expressed by

$$\mathbf{z}_\epsilon \;=\; (B - \sigma\, I_3)\; \boldsymbol{\epsilon}_k \tag{B.2.1.7}$$

so that, the dyad matrix $\mathbf{z}_\epsilon\mathbf{z}_\epsilon^T$ is given by

$$\mathbf{z}_\epsilon\mathbf{z}_\epsilon^T \;=\; (B - \sigma\, I_3)\; \boldsymbol{\epsilon}_k\, \boldsymbol{\epsilon}_k^T\, (B - \sigma\, I_3) \tag{B.2.1.8}$$

Then, taking the expectations of both sides, and using Eq. (3.29), that is $E[\,\boldsymbol{\epsilon}_k\, \boldsymbol{\epsilon}_k^T] = \eta_k\, I_3$ yields

$$E\left[\mathbf{z}_\epsilon\mathbf{z}_\epsilon^T\right] \;=\; \eta_k E\left[B - \sigma\, I_3\right]\, E\left[B - \sigma\, I_3\right]^T \tag{B.2.1.9}$$

Replacing, in Eq. (B.2.1.9), the unknown quantities, B and σ, by their best available estimates, $\widehat{B}$ and $\widehat{\sigma}$, respectively, we have

$$E\left[\mathbf{z}_\epsilon\mathbf{z}_\epsilon^T\right] \;=\; \eta_k\, \left(\widehat{B} - \widehat{\sigma}\, I_3\right)\, \left(\widehat{B} - \widehat{\sigma}\, I_3\right)^T \tag{B.2.1.10}$$

Using the linear property of the trace and expectation operators, and Eq. (B.2.1.10) yields

$$\begin{aligned} E\left[\mathbf{z}_\epsilon^T\mathbf{z}_\epsilon\right] &= tr\left(E\left[\mathbf{z}_\epsilon\mathbf{z}_\epsilon^T\right]\right) \\ &= \eta_k\, \left[\, tr\left(BB^T\right) + \sigma^2\right] \end{aligned} \tag{B.2.1.11}$$

Computation of $E\left[\,\sigma_\epsilon^2\right]$ Using the definitions of B and B_ϵ,

$$B \triangleq \sum_{i=1}^{n} a_i\, \mathbf{b}_i\, \mathbf{r}_i^T \tag{B.2.1.12}$$

$$B_\epsilon \triangleq\; [\epsilon\times]\, B \tag{B.2.1.13}$$

in the definition of σ_ϵ, $\sigma_\epsilon = tr(\, B_\epsilon)$, one gets

$$\begin{aligned} \sigma_\epsilon &= \left[\sum_{i=1}^{n} a_i\, [\mathbf{b}_i\times]\; \mathbf{r}_i\right]^T\; \boldsymbol{\epsilon}_k \\ &= \mathbf{z}^T\, \boldsymbol{\epsilon}_k \end{aligned} \tag{B.2.1.14}$$

Then, squaring Eq. (B.2.1.14), taking the expectations of both sides, using again the assumption that $E[\,\boldsymbol{\epsilon}_k\, \boldsymbol{\epsilon}_k^T] = \eta_k\, I_3$ yields.

$$E\left[\,\sigma_\epsilon^2\right] \;=\; \eta_k \mathbf{z}^T\mathbf{z} \tag{B.2.1.15}$$

The computation will involve, as usual, the estimate, $\widehat{\mathbf{z}}$, rather than $\mathbf{z}$ itself.

Finally, both results from Eqs. (B.2.1.14) and (B.2.1.11) are added to give $\mathcal{Q}_{22}$.

$$\mathcal{Q}_{22} \;=\; \eta_k \left(tr\left(\widehat{B}(\widehat{B})^T\right) + (\widehat{\sigma})^2 + \widehat{\mathbf{z}}^T \widehat{\mathbf{z}} \right) \tag{B.2.1.16}$$

B.2.2 Computation of $\mathcal{R}_k$

Recall that $\mathcal{R}_k$ is defined as

$$\mathcal{R}_k \overset{\triangle}{=} E\left[V_k\, V_k^{\;T}\right] \tag{B.2.2.1}$$

Using the expression for V_{k+1}, (Eq. 3.12), and partitioning the matrix 4×4 matrix $\mathcal{R}_k$

$$\mathcal{R}_k = \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \mathcal{R}_{21} & \mathcal{R}_{22} \end{bmatrix} \tag{B.2.2.2}$$

where $\mathcal{R}_{11}$ is a 3×3 submatrix, yields the following expressions for $\mathcal{R}_{11}$, $\mathcal{R}_{12}$, $\mathcal{R}_{21}$, and $\mathcal{R}_{22}$.

$$\mathcal{R}_{11} = E\left[(\mathcal{S}_b - \sigma_b I)^2 \;+\; \mathbf{z}_b \mathbf{z}_b^T\right] \tag{B.2.2.3a}$$

$$\mathcal{R}_{12} = E\left[\mathcal{S}_b \mathbf{z}_b\right] \tag{B.2.2.3b}$$

$$\mathcal{R}_{21} = \mathcal{R}_{12}^{\;T} \tag{B.2.2.3c}$$

$$\mathcal{R}_{22} = E\left[\mathbf{z}_b^T \mathbf{z}_b \;+\; \sigma_b^2\right] \tag{B.2.2.3d}$$

After some computation, Eqs. (B.2.2.3) become

$$\begin{aligned} \mathcal{R}_{11} \;=\; \mu_k/n_k \;\Bigg\{ \sum_{i=1}^{n_k} \left[3 - (\mathbf{r}_i^T\, \mathbf{b}_i)^2\right] I_3 + \left(\mathbf{b}_i^T\, \mathbf{r}_i\right)\left(\mathbf{b}_i\, \mathbf{r}_i^T + \mathbf{r}_i\, \mathbf{b}_i^T\right) \\ + (\mathbf{r}_i \times)\left(\mathbf{b}_i\, \mathbf{b}_i^T\right)(\mathbf{r}_i \times)^T \Bigg\} \end{aligned} \tag{B.2.2.4a}$$

$$\mathcal{R}_{12} \;=\; \mathbf{0} \tag{B.2.2.4b}$$

$$\mathcal{R}_{21} \;=\; \mathbf{0}^T \tag{B.2.2.4c}$$

$$\mathcal{R}_{22} = 2\,\mu_k/n_k \tag{B.2.2.4d}$$

The details of the computation of $\mathcal{R}_{22}$ are given next. The other elements of $\mathcal{R}_k$ are obtained by similar computations. From Eq. (B.2.2.3d),

$$\mathcal{R}_{22} \overset{\triangle}{=} E\left[\mathbf{z}_b^T \mathbf{z}_b \;+\; \sigma_b^2\right] \tag{B.2.2.5}$$

The assumptions in the following derivation are that n_k simultaneous observations are performed at time t_k, that they all have the same variance μ_k, and that the weights $a_i$add to one; that is $\sum_{i=1}^{n_k} a_i = 1$.

Computation of $E\left[\mathbf{z}_b^T \mathbf{z}_b\right]$ The vector $\mathbf{z}_b$ is defined by (Eq. 3.11)

$$\mathbf{z}_b \triangleq \sum_{i=1}^{n_k} a_i \left[\delta \mathbf{b}_i \times\right] \mathbf{r}_i \tag{B.2.2.6}$$

Its squared norm is given by

$$\mathbf{z}_b^T \mathbf{z}_b \;=\; \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} a_i\, a_j\, \mathbf{r}_i^T \left[\delta \mathbf{b}_i \times\right]^T \left[\delta \mathbf{b}_j \times\right] \mathbf{r}_j \tag{B.2.2.7}$$

Then, taking the expectation of both sides, and assuming, that $\mathbf{r}_i$ are deterministic entities,

$$E\left[\mathbf{z}_b^T \mathbf{z}_b\right] \;=\; \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} a_i\, a_j\, \mathbf{r}_i^T E\left[\left[\delta \mathbf{b}_i \times\right]^T \left[\delta \mathbf{b}_j \times\right]\right] \mathbf{r}_j \tag{B.2.2.8}$$

The above expression is further developed under the assumption, that $\boldsymbol{\delta b}_i$ is a zero-mean white noise in time sequence and that the sensors output are independent. The covariance matrix of the unit norm measurement error model is assumed to be

$$E\left[\,\boldsymbol{\delta b}_i\, \boldsymbol{\delta b}_i^T\right] \;=\; \mu_i^2 \left(I_3 - \mathbf{b}_i\, \mathbf{b}_i^T\right) \tag{B.2.2.9}$$

Then, using the identity,

$$\left[\delta \mathbf{b}_i \times\right] \left[\delta \mathbf{b}_i \times\right] \equiv \boldsymbol{\delta b}_i\, \boldsymbol{\delta b}_i^T - \boldsymbol{\delta b}_i^T\, \boldsymbol{\delta b}_i\, I_3 \tag{B.2.2.10}$$

and using the fact that the weights a_i are normalized, one gets

$$E\left[\mathbf{z}_b^T \mathbf{z}_b\right] \;=\; \mu_k \frac{1}{n_k} \sum_{i=1}^{n_k} \left[1 + \left(\mathbf{r}_i^T\, \mathbf{b}_i\right)^2\right] \tag{B.2.2.11}$$

Computation of $E\left[\,\sigma_b^2\right]$ By definition (Eq. 3.11) σ_b is given by

$$\sigma_b \triangleq \sum_{i=1}^{n_k} a_i\, \boldsymbol{\delta b}_i^T\, \mathbf{r}_i \tag{B.2.2.12}$$

Taking the expectation of the square

$$E\left[\,\sigma_b^2\right] \;=\; \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} a_i\, a_j E\left[\,\mathbf{b}_i^T\, \boldsymbol{\delta b}_i\, \boldsymbol{\delta b}_j^T\, \mathbf{r}_j\right] \tag{B.2.2.13}$$

Then, under the special assumptions on $\boldsymbol{\delta b}_i$ and on a_i, we obtain

$$E\left[\,\sigma_b^2\right] = \mu_k / n_k \sum_{i=1}^{n_k} \left[1 - \left(\mathbf{r}_i^T\, \mathbf{b}_i\right)^2\right] \tag{B.2.2.14}$$

Adding the results obtained in Eqs. (B.2.2.11) and (B.2.2.14) yields the expression for $\mathcal{R}_{22}$,

$$\mathcal{R}_{22} = 2\,\mu_k/n_k \tag{B.2.2.15}$$

B.3 Proof of Eq. (3.75)

Proof. We rewrite the definition of the residual r^{λ}_{k+1} (Eq. 3.69)

$$r^{\lambda}_{k+1} \overset{\triangle}{=} 1 - \widehat{\mathbf{q}}^T_{k+1/k}\left(I_4 - \delta K_{k+1}\right)\widehat{\mathbf{q}}_{k+1/k} \tag{B.3.1}$$

The predicted estimation error $\boldsymbol{\Delta q}_{k+1/k}$ is define as

$$\boldsymbol{\Delta q}_{k+1/k} \overset{\triangle}{=} \mathbf{q}_{k+1} - \widehat{\mathbf{q}}_{k+1/k} \tag{B.3.2}$$

Substituting Eq.(B.3.2) into Eq.(B.3.1) and developing yields

$$r^{\lambda}_{k+1} = \mathbf{q}^T_{k+1}\left(I_4 - \delta K_{k+1}\right)\mathbf{q}_{k+1} - 2\,\boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K_{k+1}\right)\mathbf{q}_{k+1} + \boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K_{k+1}\right)\boldsymbol{\Delta q}_{k+1/k} \tag{B.3.3}$$

Then we substitute the following equation

$$\delta K_{k+1} = \delta K^o_{k+1} + V_{k+1} \tag{B.3.4}$$

into Eq.(B.3.3) and we look individually at each of the three terms in the right hand side of Eq.(B.3.3). This gives

$$\begin{aligned}
\mathbf{q}^T_{k+1}\left(I_4 - \delta K_{k+1}\right)\mathbf{q}_{k+1} &= \mathbf{q}^T_{k+1}\left(I_4 - \delta K^o_{k+1}\right)\mathbf{q}_{k+1} &- \mathbf{q}^T_{k+1}V_{k+1}\mathbf{q}_{k+1}\\
\boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K_{k+1}\right)\mathbf{q}_{k+1} &= \boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K^o_{k+1}\right)\mathbf{q}_{k+1} &- \boldsymbol{\Delta q}^T_{k+1/k}V_{k+1}\mathbf{q}_{k+1}\\
\boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K_{k+1}\right)\boldsymbol{\Delta q}_{k+1/k} &= \boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K^o_{k+1}\right)\boldsymbol{\Delta q}_{k+1/k} &- \boldsymbol{\Delta q}^T_{k+1/k}V_{k+1}\boldsymbol{\Delta q}_{k+1/k}
\end{aligned} \tag{B.3.5}$$

From the **q**-method, we know that

$$\left(I_4 - \delta K^o_{k+1}\right)\mathbf{q}_{k+1} = \mathbf{0} \tag{B.3.6}$$

Therefore, the exact value of the residual is given by

$$\begin{aligned}
r^{\lambda}_{k+1} = &-\mathbf{q}^T_{k+1}V_{k+1}\mathbf{q}_{k+1}\\
&-\boldsymbol{\Delta q}^T_{k+1/k}V_{k+1}\mathbf{q}_{k+1} + \boldsymbol{\Delta q}^T_{k+1/k}\left(I_4 - \delta K^o_{k+1}\right)\boldsymbol{\Delta q}_{k+1/k}\\
&-\boldsymbol{\Delta q}^T_{k+1/k}V_{k+1}\boldsymbol{\Delta q}_{k+1/k}
\end{aligned} \tag{B.3.7}$$

and if we retain only the first order approximation in V_{k+1} and $\boldsymbol{\Delta q}_{k+1/k}$, we have

$$r^{\lambda}_{k+1} \;=\; -\mathbf{q}^{T}_{k+1}\, V_{k+1}\mathbf{q}_{k+1} \tag{B.3.8}$$

Taking the expectation of both sides in Eq. (B.3.8) and remembering that $\mathbf{q}_{k+1}$ is a deterministic variable, yields

$$E\left[r^{\lambda}_{k+1}\right] \;=\; \mathbf{q}^{T}_{k+1} E\left[V_{k+1}\right]\mathbf{q}_{k+1} \tag{B.3.9}$$

Since the measurement noise V_{k+1} is zero-mean, we conclude that the residual r^{λ}_{k+1} is also zero-mean in a first order approximation.

Using Eq. (B.3.8) the expectation of the squared residual is

$$\begin{aligned}\left(r^{\lambda}_{k+1}\right)^2 &= \left(\mathbf{q}^{T}_{k+1}\, V_{k+1}\mathbf{q}_{k+1}\right)\left(\mathbf{q}^{T}_{k+1}\, V_{k+1}\mathbf{q}_{k+1}\right)\\ &= \mathbf{q}^{T}_{k+1}\left(V_{k+1}\mathbf{q}_{k+1}\mathbf{q}^{T}_{k+1}\, V^{T}_{k+1}\right)\mathbf{q}_{k+1}\\ &= \mathbf{q}^{T}_{k+1}\left(V_{k+1}\mathbf{q}_{k+1}\right)\left(V_{k+1}\mathbf{q}_{k+1}\right)^{T}\mathbf{q}_{k+1}\end{aligned} \tag{B.3.10}$$

Considering the quantity $V_{k+1}\mathbf{q}_{k+1}$, assuming that there is one single new observation, with $\mathbf{r}_{k+1}$ as the reference unit vector, defining the following 4×3 real matrix Θ_{k+1}

$$\Theta_{k+1} \triangleq \begin{bmatrix} \mathbf{r}^{T}_{k+1}\,\mathbf{e}_{k+1}\, I_3 + \mathbf{r}_{k+1}\,\mathbf{e}^{T}_{k+1} - \mathbf{e}_{k+1}\,\mathbf{r}^{T}_{k+1} - q_{k+1}\left[\mathbf{r}_{k+1}\times\right] \\ \mathbf{e}^{T}_{k+1}\left[\mathbf{r}_{k+1}\times\right] + q_{k+1}\,\mathbf{r}^{T}_{k+1} \end{bmatrix} \tag{B.3.11}$$

and using the expression for V_{k+1} given in Eqs. (3.11), (3.12), we can write the following identity

$$V_{k+1}\mathbf{q}_{k+1} \equiv \Theta_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{B.3.12}$$

where $\boldsymbol{\delta b}_{k+1}$ is the measurement error in the unit vector $\mathbf{b}_{k+1}$. Equation (B.3.12) can be proved by direct computation. Using Eq. (B.3.12) in Eq. (B.3.10) yields

$$\left(r^{\lambda}_{k+1}\right)^2 \;=\; \mathbf{q}^{T}_{k+1}\,\Theta_{k+1}\,\boldsymbol{\delta b}_{k+1}\,\boldsymbol{\delta b}^{T}_{k+1}\,\Theta^{T}_{k+1}\mathbf{q}_{k+1} \tag{B.3.13}$$

Taking the expectation of both sides in Eq. (B.3.13) and recalling the expression for the vector measurement covariance matrix

$$E\left[\boldsymbol{\delta b}_{k+1}\,\boldsymbol{\delta b}^{T}_{k+1}\right] \;=\; \mu_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^{T}_{k+1}\right) \tag{B.3.14}$$

yields the following approximate expression for the predicted residual variance

$$E\left[\left(r^{\lambda}_{k+1}\right)^2\right] = \mu_{k+1}\mathbf{q}^T_{k+1}\,\Theta_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^T_{k+1}\right)\Theta^T_{k+1}\mathbf{q}_{k+1} \tag{B.3.15}$$

Substituting $\widehat{\mathbf{q}}_{k+1/k}$ for $\mathbf{q}_{k+1}$ in Eq. (B.3.15), since in practice we know $\widehat{\mathbf{q}}_{k+1/k}$, yields

$$E\left[\left(r^{\lambda}_{k+1}\right)^2\right] = \mu_{k+1}\,\widehat{\mathbf{q}}^T_{k+1/k}\widehat{\Theta}_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^T_{k+1}\right)\widehat{\Theta}^T_{k+1}\,\widehat{\mathbf{q}}_{k+1/k} \tag{B.3.16}$$

where $\widehat{\Theta}_{k+1}$ is defined by using $\widehat{\mathbf{q}}_{k+1/k}$ in Eq. (B.3.11). Finally, we solve

$$E\left[\left(r^{\lambda}_{k+1}\right)^2\right] = \left(r^{\lambda}_{k+1}\right)^2 \tag{B.3.17}$$

for μ_{k+1}. This yields the estimate of the measurement noise, μ^*_{k+1},

$$\mu^*_{k+1} = \frac{\left(r^{\lambda}_{k+1}\right)^2}{\widehat{\mathbf{q}}^T_{k+1/k}\widehat{\Theta}_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^T_{k+1}\right)\widehat{\Theta}^T_{k+1}\,\widehat{\mathbf{q}}_{k+1/k}} \tag{B.3.18}$$

If we want to compute the predicted variance of the residual to higher orders in the measurement and estimation errors, we have to keep all the terms in Eq. (B.3.7). Computing the residual squared and taking the expectation yields after some developments

$$\begin{aligned} E\left[\left(r^{\lambda}_{k+1}\right)^2\right] = {} & \mu_k\left(\widehat{\mathbf{q}}^T_{k+1/k}\widehat{\Theta}_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^T_{k+1}\right)\widehat{\Theta}^T_{k+1}\,\widehat{\mathbf{q}}_{k+1/k}\right) \\ & +\widehat{\mathbf{q}}_{k+1/k}E\left[\left(V_{k+1}\boldsymbol{\Delta q}_{k+1/k}\right)\left(V_{k+1}\boldsymbol{\Delta q}_{k+1/k}\right)^T\right]\widehat{\mathbf{q}}_{k+1/k} \\ & +E\left[\left(\boldsymbol{\Delta q}^T_{k+1/k}\left(I_3 - \delta K^o_{k+1}\right)\boldsymbol{\Delta q}_{k+1/k}\right)^2\right] \end{aligned} \tag{B.3.19}$$

We see from Eq. (B.3.19) that the higher orders terms are two orders of magnitude smaller than the first term. This justifies the first order approximation made along the proof.

□

B.4 Proof of Eqs. (3.78)

Proof. Let $\widehat{\mathbf{q}}_{k/k}$ and $\widehat{\mathbf{q}}_{k+1/k}$ denote the a posteriori and a priori quaternion estimates, respectively, which are computed from the associated K-matrices, $K_{k/k}$ and $K_{k+1/k}$. Let $\boldsymbol{\Delta q}_{k/k}$ and $\boldsymbol{\Delta q}_{k+1/k}$ denote the respective additive estimation errors, and let $P^{\mathbf{q}}_{k/k}$ and $P^{\mathbf{q}}_{k+1/k}$ denote the associated covariance matrices. Moreover, assume that $P^{\mathbf{q}}_{k/k}$ is a known matrix, and recall that the Time Update equation of Optimal-REQUEST is (see Eq. 3.62)

$$K_{k+1/k} = \Phi_k\,K_{k/k}\,\Phi^T_k \tag{B.4.1}$$

As shown in Ref. [24] the Time Update equation (B.4.1) for the K-matrix is associated with the following Time Update equation for the quaternion estimate

$$\widehat{\mathbf{q}}_{k+1/k} = \Phi_k \, \widehat{\mathbf{q}}_{k/k} \tag{B.4.2}$$

Consider the following state process model equation for the true quaternion

$$\mathbf{q}_{k+1} = \Phi_k \mathbf{q}_k + \mathbf{w}_k \tag{B.4.3}$$

It was shown in Chapter 2 that a first order approximation in $\boldsymbol{\epsilon}_k$ and Δt to the process noise in Eq. (B.4.3) is given by

$$\mathbf{w}_k = -\Xi_k \left(\frac{\Delta t}{2} \right) \boldsymbol{\epsilon}_k \tag{B.4.4}$$

$$\Xi_k = \begin{bmatrix} [\,\mathbf{e}_k \times] + q_k I_3 \\ \mathbf{e}_k^T \end{bmatrix} \tag{B.4.5}$$

where $\boldsymbol{\epsilon}_k$ is the gyro outputs noise, which is modelled as a zero-mean white sequence with known covariance matrix Q_k^{ϵ}. The matrix Ξ_k is a 4×3 matrix function of the quaternion, $\mathbf{q}_k$, where $\mathbf{q}_k^T = [\,\mathbf{e}_k^T \;\; q_k]$. Using Eqs. (B.4.3) to (B.4.5), the Time Update equation of the estimation error covariance is shown to be (see Chapter 2)

$$P^{\mathbf{q}}_{k+1/k} = \Phi_k \, P^{\mathbf{q}}_{k/k} \, \Phi_k^T + Q_k^q \tag{B.4.6}$$

$$Q_k^q = \widehat{\Xi}_k Q_k^{\epsilon} \, \widehat{\Xi}_k^T \left(\frac{\Delta t}{2} \right)^2 \tag{B.4.7}$$

where $\widehat{\Xi}_k$ is computed by substituting $\widehat{\mathbf{q}}_{k/k}$ for $\mathbf{q}_k$ in Eq. (B.4.5).

□

B.5 Proof of Eq. (3.79)

Proof. The residual $\mathbf{r}^q_{k+1}$ was defined as (Eq. 3.76)

$$\mathbf{r}^q_{k+1} \stackrel{\triangle}{=} \left(I_4 - \delta K_{k+1} \right) \widehat{\mathbf{q}}_{k+1/k} \tag{B.5.1}$$

Using the following relations

$$\begin{aligned} \boldsymbol{\Delta q}_{k+1/k} &\stackrel{\triangle}{=} \mathbf{q}_{k+1} - \widehat{\mathbf{q}}_{k+1/k} \\ V_{k+1} &\stackrel{\triangle}{=} \delta K^o_{k+1} - \delta K_{k+1} \end{aligned} \tag{B.5.2}$$

in Eq. (B.5.1) and developing yields

$$\mathbf{r}^q_{k+1} \;=\; -\left(I_4 - \delta K^o_{k+1}\right) \boldsymbol{\Delta q}_{k+1/k} + V_{k+1}\mathbf{q}_{k+1} - V_{k+1}\boldsymbol{\Delta q}_{k+1/k} \tag{B.5.3}$$

Taking the expectation of Eq. (B.5.3) and using the facts that the variables V_{k+1} and $\boldsymbol{\Delta q}_{k+1/k}$ are uncorrelated and that V_{k+1} and $\boldsymbol{\Delta q}_{k+1/k}$ are zero-mean yields

$$E\left[\mathbf{r}^q_{k+1}\right] \;=\; \mathbf{0} \tag{B.5.4}$$

That is, the residual $\mathbf{r}^q_{k+1}$ is unbiased. Using the identity Eq. (B.3.12) in Eq. (B.5.3), the residual can be expressed by

$$\mathbf{r}^q_{k+1} \;=\; -\left(I_4 - \delta K^o_{k+1}\right) \boldsymbol{\Delta q}_{k+1/k} + \widehat{\Theta}_{k+1}\,\boldsymbol{\delta b}_{k+1} \tag{B.5.5}$$

Computing the squared residual and taking the expectation, one gets,

$$P_{rr} = E\left[\mathbf{r}^q_{k+1}(\mathbf{r}^q_{k+1})^T\right] \tag{B.5.6}$$

$$= \left(I_4 - \delta K^o_{k+1}\right) P^{\mathbf{q}}_{k+1/k} \left(I_4 - \delta K^o_{k+1}\right)^T \tag{B.5.7}$$

$$+\; \mu_{k+1}\widehat{\Theta}_{k+1}\left(I_3 - \mathbf{b}_{k+1}\,\mathbf{b}^T_{k+1}\right)\widehat{\Theta}^T_{k+1} \tag{B.5.8}$$

In practice, one will compute Eq. (B.5.6) by substituting δK_{k+1} for δK^o_{k+1}, which yields Eq. (3.79). □

Appendix C

C.1 Factorization of type A

Given two real $n \times 1$ column-vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, we wish to factorize the following scalar function, $f(\mathbf{x})$, with respect to $\mathbf{x}$

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \|\mathbf{x} - \mathbf{x}_1\|^2 \;+\; \frac{1}{2}\, a_2 \|\mathbf{x} - \mathbf{x}_2\|^2 \tag{C.1.1}$$

where a_1 and a_2 are given real positive scalars. The steps of the factorization procedure are as follows. The expression in Eq. (C.1.1) is developed using the definition of the Euclidean norm, that is $\|\mathbf{x}\|^2 \stackrel{\triangle}{=} \mathbf{x}^T\mathbf{x}$.

$$f(\mathbf{x}) = \frac{1}{2}\left[(a_1 + a_2)\, \|\mathbf{x}\|^2 - 2\,(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)^T\,\mathbf{x}\right] + \frac{1}{2}\left[a_1\,\|\mathbf{x}_1\|^2 + a_2\,\|\mathbf{x}_2\|^2\right] \tag{C.1.2}$$

Then, factorizing by $(a_1 + a_2)$ yields

$$f(\mathbf{x}) = \frac{1}{2}\left\{(a_1 + a_2)\left[\|\mathbf{x}\|^2 - 2\frac{(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)^T}{(a_1 + a_2)}\mathbf{x}\right]\right\} + \frac{1}{2}\left(a_1\,\|\mathbf{x}_1\|^2 + a_2\,\|\mathbf{x}_2\|^2\right) \tag{C.1.3}$$

The expression in the square brackets can be completed to complete square form, which yields, after some manipulations, the following expression

$$f(\mathbf{x}) = \frac{1}{2}\;(a_1 + a_2)\,\|\mathbf{x} - \frac{(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)}{(a_1 + a_2)}\|^2 + \frac{1}{2}\,\frac{a_1\,a_2}{(a_1 + a_2)}\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \tag{C.1.4}$$

which is the desired expression for $f(\mathbf{x})$.

C.2 Factorization of type B

Given a real $n \times 1$ column-vector, $\mathbf{x}_1$, and the following function $f(\mathbf{x})$

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \|\mathbf{x} - \mathbf{x}_1\|^2 \tag{C.2.1}$$

where a_1 is a given positive real scalar, we wish to factorize $f(\mathbf{x})$ with respect to $\mathbf{x}$, *where* $\mathbf{x}$ *satisfies the constraint* $\|\mathbf{x}\|^2 = 1$. For this purpose we will use the following identity

$$\|\mathbf{x} - \mathbf{y}\|^2 \simeq 2 - 2\mathbf{y}^T\mathbf{x} \tag{C.2.2}$$

which is true for any unit-norm $\mathbf{x}$ and $\mathbf{y}$. We begin by expanding the expression of $f(\mathbf{x})$, using the definition of the Euclidean norm and inserting the unitary property of $\mathbf{x}$. This yields

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \left(-2\mathbf{x}_1^T\mathbf{x}\right) + \frac{1}{2}\, a_1 \left(1 + \|\mathbf{x}_1\|^2\right) \tag{C.2.3}$$

Then, dividing and multiplying the first term in the right-hand side of Eq. (C.2.3) by $\|\mathbf{x}_1\|$ yields

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \|\mathbf{x}_1\| \left(-2\frac{\mathbf{x}_1^T}{\|\mathbf{x}_1\|}\mathbf{x}\right) + \frac{1}{2}\, a_1 \left(1 + \|\mathbf{x}_1\|^2\right) \tag{C.2.4}$$

By adding and subtracting 2 to the expression that is lying in the parenthesis of the same term gives, after expanding,

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \|\mathbf{x}_1\| \left(2 - 2\frac{\mathbf{x}_1^T}{\|\mathbf{x}_1\|}\mathbf{x}\right) + \frac{1}{2}\, a_1 \left(1 - 2\|\mathbf{x}_1\| + \|\mathbf{x}_1\|^2\right) \tag{C.2.5}$$

Using the identity of Eq. (C.2.2), we see that the first term in the right hand side of Eq. (C.2.5) can be expressed as a complete square. The second term of the sum can be also factorized as a complete square. Using these two facts we obtain the desired expression for $f(\mathbf{x})$, that is

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \|\mathbf{x}_1\|\|\mathbf{x} - \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}\|^2 + \frac{1}{2}\, a_1 \left(1 - \|\mathbf{x}_1\|\right)^2 \tag{C.2.6}$$

C.3 Factorization of type C

Given two real $n \times 1$ unit-norm column-vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, and a function $f(\mathbf{x})$ of the form

$$f(\mathbf{x}) = \frac{1}{2}\, a_1\|\mathbf{x} - \mathbf{x}_1\|^2 + \frac{1}{2}\, a_2\|\mathbf{x} - \mathbf{x}_2\|^2 \tag{C.3.1}$$

where a_1 and a_2 are two given positive real scalars, we wish to factorize $f(\mathbf{x})$ with respect to $\mathbf{x}$, *where* $\mathbf{x}$ *satisfies the constraint* $\|\mathbf{x}\|^2 = 1$. Using Eq. (C.2.2) in Eq. (C.3.1) gives

$$f(\mathbf{x}) = \frac{1}{2}\, a_1 \left(2 - 2\mathbf{x}_1^T\mathbf{x}\right) + \frac{1}{2}\, a_2 \left(2 - 2\mathbf{x}_2^T\mathbf{x}\right) \tag{C.3.2}$$

Rearranging Eq. (C.3.2) in terms that depends on $\mathbf{x}$ and terms that are constants with respect to $\mathbf{x}$ yields

$$f(\mathbf{x}) = \frac{1}{2}\left(-2\,(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)^T\,\mathbf{x}\right) + (a_1 + a_2) \tag{C.3.3}$$

Dividing and multiplying the first term in the right-hand side of Eq. (C.3.3) by $\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|$ we get

$$f(\mathbf{x}) = \frac{1}{2}\,\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|\left(-2\frac{(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)^T}{\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|}\mathbf{x}\right) + (a_1 + a_2) \tag{C.3.4}$$

Then, adding and subtracting 2 to the expression that is lying inside of the parenthesis in Eq. (C.3.4) yields, after expansion,

$$\begin{aligned} f(\mathbf{x}) = \frac{1}{2}\,\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|\left(2 - 2\frac{(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)^T}{\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|}\mathbf{x}\right) \\ + (a_1 + a_2 - \|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|) \end{aligned} \tag{C.3.5}$$

Using Eq. (C.2.2) in the parenthesis in the first line of Eq. (C.3.5) yields the desired expression for $f(\mathbf{x})$

$$f(\mathbf{x}) = \frac{1}{2}\,\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|\|\mathbf{x} - \frac{(a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2)}{\|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|}\|^2 + (a_1 + a_2 - \|\,a_1\,\mathbf{x}_1 \;+\; a_2\,\mathbf{x}_2\|) \tag{C.3.6}$$

Appendix D

D.1 Definitions

Definition D.1.1. *The* canonical decomposition *of a matrix $Z \in \mathbb{R}^{m\times n}$ is defined by*

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} z_{_{ij}}\, E^{ij} \tag{D.1.1}$$

where $z_{_{ij}}$ is the (ij) element of Z and $E^{ij} \in \mathbb{R}^{m\times n}$ is a matrix with 1 at position (ij) and zeros elsewhere.

Definition D.1.2. *The* vec *operator stacks the columns of a rectangular matrix one underneath the other to form a single column-matrix [59, p. 272]. The vec operator executed on $X \in \mathbb{R}^{m\times n}$ is defined by*

$$\mathbf{x} = \mathit{vec}\, X \tag{D.1.2}$$

where

$$X[i,j] = x_{_{(j-1)m+i}} \qquad i = 1\ldots m\ ,\ j = 1\ldots n \tag{D.1.3}$$

Note that $\mathbf{x}$ is a mn-dimensional vector.

Definition D.1.3. *The* Kronecker product, direct product, *or* tensor product *of the real matrices $A = [\,a_{_{ij}}] \in \mathbb{R}^{m\times n}$ and $B = [\,b_{_{ij}}] \in \mathbb{R}^{p\times q}$ is denoted by $\otimes$ and is defined to be the block-matrix [58, p. 243]*

$$A \otimes B = \begin{pmatrix} a_{_{11}}B & \cdots & a_{_{1n}}B \\ \vdots & \ddots & \vdots \\ a_{_{m1}}B & \cdots & a_{_{mn}}B \end{pmatrix} \in \mathbb{R}^{mp\times nq} \tag{D.1.4}$$

Definition D.1.4. *A random matrix $W \in \mathbb{R}^{m\times n}$ is a matrix whose elements w_{ij} are random variables. The expected value of W is a $m \times n$ matrix defined as*

$$E\{W\} \overset{\triangle}{=} E\{w_{ij}\} \tag{D.1.5}$$

where

$$W = [w_{ij}] \tag{D.1.6}$$

The covariance matrix of W is a $mn \times mn$ matrix defined as

$$cov(W) \overset{\triangle}{=} cov(\mathbf{w}) \tag{D.1.7}$$

where

$$\mathbf{w} = vecW \tag{D.1.8}$$

D.2 Propositions

Proposition D.2.1. *Any matrix E^{ij} can be expressed as*

$$E^{ij} = \mathbf{e}_i(\mathbf{e}_j)^T \tag{D.2.1}$$

where $\mathbf{e}_k \in \mathbb{R}^n$ $(k = i, j)$ is a column-matrix with 1 at position k and zeros elsewhere.

Proof. Equation (D.2.1) is proved by direct computation using Definition D.1.1. □

Proposition D.2.2. *Given a matrix $X \in \mathbb{R}^{i_1\times i_2}$, the general expression for a matrix $Z \in \mathbb{R}^{i_3\times i_4}$, whose elements z_{ij} are linear combinations of* all *the elements x_{kl} of X, is*

$$Z = \sum_{i=1}^{i_3}\sum_{j=1}^{i_4}\sum_{k=1}^{i_1}\sum_{l=1}^{i_2} \alpha_{ijkl}\, E^{ik} X\, E^{lj} \tag{D.2.2}$$

where α_{ijkl} is the coefficient of x_{kl} in the linear expression for z_{ij}.

Proof. By assumption, z_{ij}, for $i = 1 \ldots i_3$, $j = 1 \ldots i_4$, is a linear combination of the x_{kl}'s, for $k = 1 \ldots i_1$, $l = 1 \ldots i_2$. Therefore,

$$z_{ij} = \sum_{k=1}^{i_1}\sum_{l=1}^{i_2} \alpha_{ijkl}\, x_{kl} \tag{D.2.3}$$

Since

$$x_{kl} = (\mathbf{e}_k)^T X \, \mathbf{e}_l \tag{D.2.4}$$

then, inserting Eq. (D.2.4) into Eq. (D.2.3) yields

$$z_{ij} = \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \alpha_{ijkl} \left[(\mathbf{e}_k)^T X \, \mathbf{e}_l \right] \tag{D.2.5}$$

Writing the canonical decomposition of Z $\in \mathbb{R}^{i_3 \times i_4}$ (see Definition D.1.1), and using Proposition D.2.1, we have the following expression for Z

$$\begin{aligned} Z &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} z_{ij} \, E^{ij} \\ &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} z_{ij} [\mathbf{e}_i (\mathbf{e}_j)^T] \\ &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \mathbf{e}_i \, z_{ij} (\mathbf{e}_j)^T \end{aligned} \tag{D.2.6}$$

The expression for Z can be further developed by inserting Eq. (D.2.5) into Eq. (D.2.6); this yields

$$\begin{aligned} Z &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \mathbf{e}_i \left\{ \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \alpha_{ijkl} \left[(\mathbf{e}_k)^T X \, \mathbf{e}_l \right] \right\} (\mathbf{e}_j)^T \\ &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \alpha_{ijkl} \left[\mathbf{e}_i (\mathbf{e}_k)^T \right] X \left[\mathbf{e}_l (\mathbf{e}_j)^T \right] \end{aligned} \tag{D.2.7}$$

Using Proposition D.2.1, one recognizes the matrices E^{ik} and E^{lj} in the left and right squared parentheses, respectively, in the right-hand side of Eq. (D.2.7). Thus

$$Z = \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \alpha_{ijkl} \, E^{ik} X \, E^{lj} \tag{D.2.8}$$

□

Proposition D.2.3. *Given a matrix $X \in \mathbb{R}^{i_1 \times i_2}$, the general expression for a matrix Z $\in \mathbb{R}^{i_3 \times i_4}$, whose elements z_{ij} are linear combinations of* all *the elements x_{kl} of X, is*

$$Z = \sum_{u=1}^{v} A^u X \, B^u \tag{D.2.9}$$

where $A^u \in \mathbb{R}^{i_3 \times i_1}$, $B^u \in \mathbb{R}^{i_2 \times i_4}$, and $v = i_1 \cdot i_2 \cdot i_3 \cdot i_4$.

Proof. The proof consists of showing the equivalence between Eq. (D.2.9) and Eq. (D.2.2).

- Eq. (D.2.2) ⇒ Eq. (D.2.9)

  There are $v = i_1 \cdot i_2 \cdot i_3 \cdot i_4$ terms in Eq. (D.2.2). Let us order the quadruples (i, j, k, l) in a lexicographic order, and denote by u the numbering index $(u = 1, 2, \ldots v)$. For instance, if $i_1 = i_2 = i_3 = 2$ and $i_4 = 1$, we can construct a correspondence table for each value of u and the quadruple (i, j, k, l) as described in Tab. D.1. The matrices A^u and B^u are defined as follows:

  Table D.1: Correspondence between u and (i,j,k,l)

u	i,j,k,l
1	1,1,1,1
2	1,1,1,2
3	1,1,2,1
4	1,1,2,2
5	2,1,1,1
6	2,1,1,2
7	2,1,2,1
8	2,1,2,2

  $$A^u = \alpha_{ijkl} E^{ik} \tag{D.2.10}$$

  $$B^u = E^{lj} \tag{D.2.11}$$

  where, in the general case, $i = 1 \ldots i_3$, $j = 1 \ldots i_4$, $k = 1 \ldots i_1$, $l = 1 \ldots i_2$, and therefore, $u = 1 \ldots v$. Using Eqs. (D.2.10) and (D.2.11), Eq. (D.2.2) is rewritten as

  $$Z = \sum_{u=1}^{v} A^u X B^u \tag{D.2.12}$$

  which is identical to Eq. (D.2.9).

- Eq. (D.2.9) ⇒ Eq. (D.2.2)

  For every $u = 1 \ldots v$, the matrices A^u and B^u of equation (D.2.9) are expressed using Definition D.1.1 as

$$A^u = \sum_{i=1}^{i_3} \sum_{k=1}^{i_1} a^u_{ik} \, E^{ik} \tag{D.2.13}$$

$$B^u = \sum_{l=1}^{i_2} \sum_{j=1}^{i_4} b^u_{lj} \, E^{lj} \tag{D.2.14}$$

Substituting Eqs. (D.2.13) and (D.2.14) into Eq. (D.2.9), and manipulating the resulted equation yields

$$\begin{aligned} Z &= \sum_{u=1}^{v} \left[\left(\sum_{i=1}^{i_3} \sum_{k=1}^{i_1} a^u_{ik} \, E^{ik} \right) X \left(\sum_{l=1}^{i_2} \sum_{j=1}^{i_4} b^u_{lj} \, E^{lj} \right) \right] \\ &= \sum_{u=1}^{v} \left[\sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} a^u_{ik} \, b^u_{lj} \left(E^{ik} X \, E^{lj} \right) \right] \\ &= \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \left[\left(\sum_{u=1}^{v} a^u_{ik} \, b^u_{lj} \right) \left(E^{ik} X \, E^{lj} \right) \right] \end{aligned} \tag{D.2.15}$$

Then, defining α_{ijkl} as

$$\alpha_{ijkl} \triangleq \sum_{u=1}^{v} a^u_{ik} \, b^u_{lj} \tag{D.2.16}$$

and inserting Eq. (D.2.16) into Eq. (D.2.15) yields

$$Z = \sum_{i=1}^{i_3} \sum_{j=1}^{i_4} \sum_{k=1}^{i_1} \sum_{l=1}^{i_2} \alpha_{ijkl} \, E^{ik} X \, E^{lj} \tag{D.2.17}$$

which is identical to Eq. (D.2.2).

□

Proposition D.2.4. *Let $A \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times p}$, and $B \in \mathbb{R}^{p \times q}$ then*

$$vec(AXB) = \left(B^T \otimes A \right) \, vecX \tag{D.2.18}$$

Proof. See [58, p. 255]. □

Proposition D.2.5. *Let $A^u \in \mathbb{R}^{i_3 \times i_1}$, $B^u \in \mathbb{R}^{i_2 \times i_4}$, and $Y \in \mathbb{R}^{i_3 \times i_4}$ be given matrices and let $X \in \mathbb{R}^{i_1 \times i_2}$ be unknown. Then, the matrix equation*

$$\sum_{u=1}^{v} A^u X \, B^u = Y \tag{D.2.19}$$

is equivalent to a system of $i_3 i_4$ *equations in* $i_1 i_2$ *unknowns described by*

$$\left\{\sum_{u=1}^{v}\left[(\,B^u)^T\otimes\,A^u\right]\right\}\,vecX\,=\,vecY \tag{D.2.20}$$

where $v = i_1 \cdot i_2 \cdot i_3 \cdot i_4$.

Proof. Applying the Vec-operator on Eq. (D.2.19) yields

$$\mathrm{vec}\left(\sum_{u=1}^{v} A^u X\, B^u\right)\,=\,\mathrm{vec}Y \tag{D.2.21}$$

Since the Vec-operator is linear, Eq. (D.2.20) leads to

$$\sum_{u=1}^{v}\mathrm{vec}\,(\,A^u X\,B^u)\,=\,\mathrm{vec}Y \tag{D.2.22}$$

Then, applying Proposition D.2.4 on each term $A^u X\, B^u$ in Eq. (D.2.22) yields

$$\mathrm{vec}(\,A^u X\,B^u) = \left[(\,B^u)^T\otimes\,A^u\right]\,\mathrm{vec}X \tag{D.2.23}$$

Using Eq. (D.2.23) in Eq. (D.2.22) and factorizing the resulted equation with respect to vec$X$ yields

$$\left\{\sum_{u=1}^{v}\left[(\,B^u)^T\otimes\,A^u\right]\right\}\,\mathrm{vec}X\,=\,\mathrm{vec}Y \tag{D.2.24}$$

which completes the proof. □

Proposition D.2.6. *Let* $X \in \mathbb{R}^{m\times n}$, $Z \in \mathbb{R}^{p\times q}$, *and* $A \in \mathbb{R}^{pq\times mn}$ *be given. Let* $\mathbf{x} \in \mathbb{R}^{mn}$ *and* $\mathbf{z} \in \mathbb{R}^{pq}$ *denote the Vec-transforms of* X *and* Z *respectively; that is* $\mathbf{x} =\, vecX$ *and* $\mathbf{z} =\, vecZ$, *then the vector equation*

$$\mathbf{z} = A\mathbf{x} \tag{D.2.25}$$

is equivalent to the matrix equation

$$Z = \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{m}\sum_{l=1}^{n} a_{ijkl}\,E^{ik}X\,E^{lj} \tag{D.2.26}$$

where

$$a_{ijkl} \overset{\triangle}{=} A[(j-1)p+i,(l-1)m+k] \tag{D.2.27}$$

for $i = 1\ldots p$, $j = 1\ldots q$, $k = 1\ldots m$ *and* $l = 1\ldots n$.

Proof. Applying Eq. (D.1.3) to X and Z yields

$$Z[i,j] = z_{(j-1)p+i} \qquad i = 1 \ldots p\ ,\ j = 1 \ldots q \tag{D.2.28}$$

$$X[k,l] = x_{(l-1)m+k} \qquad k = 1 \ldots m\ ,\ l = 1 \ldots n \tag{D.2.29}$$

Using Eq. (D.2.25) the component α of $\mathbf{z}$ is

$$z_\alpha = \sum_{\beta=1}^{mn} A[\alpha,\beta]\, x_\beta \qquad \alpha = 1 \ldots pq \tag{D.2.30}$$

where the integers α and β can be expressed as

$$\alpha = (j-1)p + i \qquad i = 1 \ldots p\ ,\ j = 1 \ldots q \tag{D.2.31}$$

$$\beta = (l-1)m + k \qquad k = 1 \ldots m\ ,\ l = 1 \ldots n \tag{D.2.32}$$

Equation (D.2.30) is equivalently written as

$$z_{(j-1)p+i} = \left\{ \sum_{k=1}^{m} \sum_{l=1}^{n} A\left[(j-1)p+i, (l-1)m+k\right] \right\} x_{(l-1)m+k} \tag{D.2.33}$$

Using Eqs. (D.2.28) and (D.2.29), equation (D.2.33) has the form

$$Z[i,j] = \left\{ \sum_{k=1}^{m} \sum_{l=1}^{n} A\left[(j-1)p+i, (l-1)m+k\right] \right\} X[k,l] \tag{D.2.34}$$

Recall (see Eq. D.2.4) that

$$X[k,l] = (\mathbf{e}_k)^T X\, \mathbf{e}_l \tag{D.2.35}$$

and that [see Eq. (D.2.6) with $i_3 = p$ and $i_4 = q$]

$$Z = \sum_{i=1}^{p} \sum_{j=1}^{q} \mathbf{e}_i Z[i,j] (\mathbf{e}_j)^T \tag{D.2.36}$$

Then, inserting Eq. (D.2.35) into Eq. (D.2.34) and using the resulted equation in Eq. (D.2.36) yields

$$\begin{aligned} Z &= \sum_{i=1}^{p} \sum_{j=1}^{q} \mathbf{e}_i \left\{ \sum_{k=1}^{m} \sum_{l=1}^{n} A[(j-1)p+i, (l-1)m+k] \left[(\mathbf{e}_k)^T X\, \mathbf{e}_l \right] \right\} \left(\mathbf{e}_j \right)^T \\ &= \sum_{i=1}^{p} \sum_{j=1}^{q} \sum_{k=1}^{m} \sum_{l=1}^{n} A\left[(j-1)p+i, (l-1)m+k\right] \left[\mathbf{e}_i \left(\mathbf{e}_k \right)^T \right] X \left[\mathbf{e}_l \left(\mathbf{e}_j \right)^T \right] \end{aligned} \tag{D.2.37}$$

Applying Proposition D.2.1 to the factors in the squared parentheses of Eq. (D.2.37) and recognizing the expression a_{ijkl} (Eq. D.2.27) leads to the following expression for Z

$$Z = \sum_{i=1}^{p} \sum_{j=1}^{q} \sum_{k=1}^{m} \sum_{l=1}^{n} a_{ijkl} \, E^{ik} X \, E^{lj} \tag{D.2.38}$$

which is identical to Eq. (D.2.26). □

Proposition D.2.7. *Let $X \in \mathbb{R}^{i_3 \times i_4}$, $Z \in \mathbb{R}^{i_1 \times i_2}$, and $A \in \mathbb{R}^{i_1 i_2 \times i_3 i_4}$ be given. Let $\mathbf{x} \in \mathbb{R}^{i_3 i_4}$ and $\mathbf{z} \in \mathbb{R}^{i_1 i_2}$ denote the Vec-transforms of X and Z respectively; that is $\mathbf{x} =$ vec$X$ and $\mathbf{z} =$ vec$Z$. Let the matrix A be defined by*

$$A = \begin{bmatrix} A^{11} & A^{12} & \cdots & A^{1l} & \cdots & A^{1i_4} \\ A^{21} & A^{22} & \cdots & A^{2l} & \cdots & A^{2i_4} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A^{j1} & A^{j2} & \cdots & A^{jl} & \cdots & A^{ji_4} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A^{i_2 1} & A^{i_2 2} & \cdots & A^{i_2 l} & \cdots & A^{i_2 i_4} \end{bmatrix} \tag{D.2.39}$$

where $A^{jl} \in \mathbb{R}^{i_1 i_3}$, then the vector equation

$$\mathbf{z} = A\mathbf{x} \tag{D.2.40}$$

is equivalent to the matrix equation

$$Z = \sum_{j=1}^{i_2} \sum_{l=1}^{i_4} A^{jl} X \, E^{lj} \tag{D.2.41}$$

Proof. Applying Proposition D.2.6 to Eq. (D.2.40), with $m = i_3$, $n = i_4$, $p = i_1$, and $q = i_2$, yields

$$\begin{aligned} Z &= \sum_{i=1}^{i_1} \sum_{j=1}^{i_2} \sum_{k=1}^{i_3} \sum_{l=1}^{i_4} a_{ijkl} \, E^{ik} X \, E^{lj} \\ &= \sum_{j=1}^{i_2} \sum_{l=1}^{i_4} \left(\sum_{i=1}^{i_1} \sum_{k=1}^{i_3} a_{ijkl} \, E^{ik} \right) X \, E^{lj} \end{aligned} \tag{D.2.42}$$

where (Proposition D.2.6, Eq. D.2.27)

$$a_{ijkl} = A[(j-1)p + i, (l-1)m + k] \tag{D.2.43}$$

Moreover, the partition of A, which is described in Eq (D.2.39), leads to the following relation

$$A[(j-1)p+i,(l-1)m+k] = A^{jl}[i,k] \tag{D.2.44}$$

for $i = 1 \dots i_1$, $j = 1 \dots i_2$, $k = 1 \dots i_3$, and $l = 1 \dots i_4$. Using Eqs. (D.2.43) and (D.2.44), we can write Eq. (D.2.42) as

$$Z = \sum_{j=1}^{i_2} \sum_{l=1}^{i_4} \left(\sum_{i=1}^{i_1} \sum_{k=1}^{i_3} A^{jl}[i,k]\, E^{ik} \right) X\, E^{lj} \tag{D.2.45}$$

Looking at the expression in parenthesis in Eq. (D.2.45) we recognize the decomposition of A^{jl} in the canonical basis (E^{ik}) of $\mathbb{R}^{i_1 i_3}$, (see Definition D.1.1). Therefore

$$A^{jl} = \sum_{i=1}^{i_1} \sum_{k=1}^{i_3} A^{jl}[i,k]\, E^{ik} \tag{D.2.46}$$

Thus, using Eq. (D.2.46) in Eq. (D.2.45) yields

$$Z = \sum_{j=1}^{i_2} \sum_{l=1}^{i_4} A^{jl} X\, E^{lj} \tag{D.2.47}$$

which is identical to Eq. (D.2.41). □

Proposition D.2.8. *Let X denote a $m \times n$ random matrix with covariance matrix P (see Definition D.1.4). Then the variances and the covariances of $X[i,j]$ are*

$$var\{X[i,j]\} = P\,[(j-1)m+i,(j-1)m+i] \tag{D.2.48}$$

$$cov\{X[i,j], X[k,l]\} = P[(j-1)m+i,(l-1)m+k] \tag{D.2.49}$$

where $i, k = 1 \dots m$ and $j, l = 1 \dots n$.

Proof. Equation (D.2.48) is proved using the definition of the Vec-operator (Eq. D.1.3) and the definition of a matrix random variable (Eq. D.1.7). □

D.3 Information Form of the MKF algorithm

The matrix KF, as formulated along Eqs. (5.7) to (5.17) is in the so-called Covariance Form. The purpose of this appendix is to provide the Information Form of the matrix

Kalman filter. The Information Form algorithm is readily formulated as follows:

$$\widetilde{Y}_{k+1} = Y_{k+1} - \sum_{s=1}^{S} H^s_{k+1} \widehat{X}_{k+1/k} G^s_{k+1} \tag{D.3.50}$$

$$\mathcal{H}_{k+1} = \sum_{s=1}^{S} \Big[(G^s_{k+1})^T \otimes H^s_{k+1} \Big] \tag{D.3.51}$$

$$P^{-1}_{k+1/k+1} = P^{-1}_{k+1/k} + \mathcal{H}^T_{k+1} R^{-1}_{k+1} \mathcal{H}_{k+1} \tag{D.3.52}$$

$$K_{k+1} = P_{k+1/k+1} \mathcal{H}^T_{k+1} R^{-1}_{k+1} \tag{D.3.53}$$

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \sum_{j=1}^{q} \sum_{l=1}^{n} K^{jl}_{k+1} \widetilde{Y}_{k+1} E^{lj} \tag{D.3.54}$$

where K^{jl}_{k+1} denotes the $p \times m$ submatrix at position (jl) in the $mn \times pq$ block-matrix K_{k+1}, and E^{lj} denotes the $q \times n$ matrix with 1 at position (lj) and 0 elsewhere.

D.4 Proof of Prop. 5.2.1

Proof. Using Eqs. (5.20) and (5.26) in Eq. (D.3.50) yields the following innovations matrix

$$\widetilde{Y} = Y - H \widehat{X}_0 = Y \tag{D.4.55}$$

Using Eqs. (5.27) and (5.25) in Eq. (D.3.52) yields

$$\begin{aligned} P^{-1} &= P_0{}^{-1} + \mathcal{H}^T (\operatorname{cov} V)^{-1} \mathcal{H} \\ &= (I_n \otimes H)^T \big(R \otimes I_p\big)^{-1} (I_n \otimes H) \\ &= R^{-1} \otimes H^T H \end{aligned} \tag{D.4.56}$$

where the third equality is obtained using the mixed-product property of the Kronecker product (see Lemma 4.2.10 of [58, p. 244]). Inverting both sides of Eq. (D.4.56) yields

$$\begin{aligned} P &= \big(R^{-1} \otimes H^T H\big)^{-1} \\ &= R \otimes \big(H^T H\big)^{-1} \end{aligned} \tag{D.4.57}$$

where the second equality is obtained using Lemma 4.2.5 of [58, p. 243]. Using Eqs. (D.4.57), (5.24), and (5.25) in Eq. (D.3.53) yields the $mn \times pq$ Kalman gain matrix

$$\begin{aligned} K &= P\mathcal{H}^T\left(\mathrm{cov}V\right)^{-1} \\ &= \left[R \otimes \left(H^T H\right)^{-1}\right]\left(I_n \otimes H\right)^T \left(R \otimes I_p\right)^{-1} \\ &= \left[RR^{-1} \otimes \left(H^T H\right)^{-1} H^T\right] \\ &= \left[I_n \otimes \left(H^T H\right)^{-1} H^T\right] \end{aligned} \tag{D.4.58}$$

The expression for K as given in Eq. (D.4.58) shows that K is a diagonal block-matrix such that each sub-matrix at position (jl), K^{jl}, is expressed as

$$K^{jl} = \delta_{jl}\left(H^T H\right)^{-1} H^T \tag{D.4.59}$$

for $j, l = 1, 2, \ldots, n$, where δ_{jl} denotes the delta of Kronecker. Using Eqs. (5.26), (D.4.59), and (D.4.55) in Eq. (D.3.54) yields the updated matrix estimate

$$\begin{aligned} \widehat{X} &= \sum_{j=1}^{n}\sum_{l=1}^{n} K^{jl}\,\widetilde{Y}\,E^{lj} \\ &= \sum_{j=1}^{n}\left(H^T H\right)^{-1} H^T Y\,E^{jj} \\ &= \left(H^T H\right)^{-1} H^T Y\left(\sum_{j=1}^{n} E^{jj}\right) \\ &= \left(H^T H\right)^{-1} H^T Y \end{aligned} \tag{D.4.60}$$

where the last equality is due to the following identity

$$I_n = \sum_{j=1}^{n} E^{jj} \tag{D.4.61}$$

Comparing Eqs. (D.4.61) and (5.23) we realize that

$$\widehat{X} = \overline{X} \tag{D.4.62}$$

Therefore the MKF algorithm contains the matrix BLUE of [43] as a special case under the conditions specified in Eqs. (5.24) to (5.27). □

D.5 Proof of Prop. 5.2.2

Proof. We begin with the formulation of the measurement update stage according to the MKF algorithm (Eqs. 5.11 to 5.17). Assuming that at time t_k we have

$$P_k = I_n \otimes \overline{P}_k \tag{D.5.63}$$

where $\overline{P}_k$ denotes the $n \times n$ covariance matrix of the estimation error in each *column* of the state-matrix, X_k, then the innovations vector is given as

$$\widetilde{\mathbf{y}}_{k+1}^T = \mathbf{y}_{k+1}^T - \mathbf{h}_{k+1}^T \widehat{X}_k \tag{D.5.64}$$

and the innovations covariance is computed as

$$\mathcal{H}_{k+1} = I_n \otimes \mathbf{h}_{k+1}^T \tag{D.5.65}$$

$$\begin{aligned} S_{k+1} &= \mathcal{H}_{k+1} P_k \mathcal{H}_{k+1}^T + R_{k+1} \\ &= \left(I_n \otimes \mathbf{h}_{k+1}^T\right)\left(I_n \otimes \overline{P}_k\right)\left(I_n \otimes \mathbf{h}_{k+1}^T\right)^T + I_n \\ &= \left(1 + \mathbf{h}_{k+1}^T \overline{P}_k \mathbf{h}_{k+1}\right) I_n \\ &= s_{k+1} I_n \end{aligned} \tag{D.5.66}$$

where $\otimes$ denotes the Kronecker product. The second equality is obtained using the mixed-product property of the Kronecker product, and Lemma 4.2.5 in [58, pp. 243,244]. The third equality stems from the following definition

$$s_{k+1} \triangleq 1 + \mathbf{h}_{k+1}^T \overline{P}_k \mathbf{h}_{k+1} \tag{D.5.67}$$

Using Eqs. (D.5.63), (D.5.65), and (D.5.67), the $mn \times n$ Kalman gain matrix is computed as

$$\begin{aligned} K_{k+1} &= P_k \mathcal{H}_{k+1}^T S_{k+1}^{-1} \\ &= \left(I_n \otimes \overline{P}_k\right)\left(I_n \otimes \mathbf{h}_{k+1}^T\right)^T \left(s_{k+1} I_n\right)^{-1} \\ &= I_n \otimes \left(\overline{P}_k \mathbf{h}_{k+1}/s_{k+1}\right) \\ &= I_n \otimes \mathbf{k}_{k+1} \end{aligned} \tag{D.5.68}$$

where

$$\mathbf{k}_{k+1} \triangleq \overline{P}_k \mathbf{h}_{k+1}/s_{k+1} \tag{D.5.69}$$

Denoting by K^{jl}_{k+1} the $m \times 1$ submatrix at position (j, l) in the block-matrix K_{k+1}, and using Equation (D.5.68), yields

$$K^{jl}_{k+1} = \delta_{jl} \mathbf{k}_{k+1} \tag{D.5.70}$$

for $j, l = 1, 2, \ldots, n$, where δ_{jl} is the delta of Kronecker. Using Eq. (D.5.70) gives the following state matrix update

$$\begin{aligned}
\widehat{X}_{k+1} &= \widehat{X}_k + \sum_{j=1}^{n} \sum_{l=1}^{n} K^{jl}_{k+1} \widetilde{\mathbf{y}}^T_{k+1} E^{lj} \\
&= \widehat{X}_k + \sum_{j=1}^{n} \mathbf{k}_{k+1} \widetilde{\mathbf{y}}^T_{k+1} E^{jj} \\
&= \widehat{X}_k + \mathbf{k}_{k+1} \widetilde{\mathbf{y}}^T_{k+1} \left(\sum_{j=1}^{n} E^{jj} \right) \\
&= \widehat{X}_k + \mathbf{k}_{k+1} \widetilde{\mathbf{y}}^T_{k+1}
\end{aligned} \tag{D.5.71}$$

where the identity given in Eq. (D.4.61) is used to obtain the last equality. Using Eqs. (D.5.67), (D.5.68), and (D.5.69) one obtains the following expanded expression for $\widehat{X}_{k+1}$,

$$\widehat{X}_{k+1} = \widehat{X}_k + \overline{P}_k \mathbf{h}_{k+1} \left(\widetilde{\mathbf{y}}^T_{k+1} - \mathbf{h}^T_{k+1} \widehat{X}_k \right) / \left(1 + \mathbf{h}^T_{k+1} \overline{P}_k \mathbf{h}_{k+1} \right) \tag{D.5.72}$$

Comparing Eqs. (D.5.72) with Eq. (5.31) we realize that the state update stages in the SMLS and the MKF algorithms are identical. Based on Eq. (5.17), the non-symmetric covariance update stage of the MKF is formulated as follows

$$\begin{aligned}
P_{k+1} &= P_k - K_{k+1} \mathcal{H}_{k+1} P_k \\
&= \left(I_n \otimes \overline{P}_k \right) - \left(I_n \otimes \mathbf{k}_{k+1} \right) \left(I_n \otimes \mathbf{h}^T_{k+1} \right) \left(I_n \otimes \overline{P}_k \right) \\
&= \left[I_n \otimes \left(\overline{P}_k - \mathbf{k}_{k+1} \mathbf{h}^T_{k+1} \overline{P}_k \right) \right] \\
&= I_n \otimes \overline{P}_{k+1}
\end{aligned} \tag{D.5.73}$$

where

$$\overline{P}_{k+1} = \overline{P}_k - \mathbf{k}_{k+1} \mathbf{h}^T_{k+1} \overline{P}_k \tag{D.5.74}$$

Using Eqs. (D.5.67) and (D.5.69) in Eq. (D.5.74) yields

$$\overline{P}_{k+1} = \overline{P}_k - \overline{P}_k \mathbf{h}_{k+1} \mathbf{h}^T_{k+1} \overline{P}_k / \left(1 + \mathbf{h}^T_{k+1} \overline{P}_k \mathbf{h}_{k+1} \right) \tag{D.5.75}$$

An inspection of Eqs. (D.5.75) and (5.32) reveals that they are identical. As a conclusion, we have proved that the SMLS algorithm derived in [44, p. 96] can be derived as a special case of the MKF. □

Appendix E

E.1 Derivation of Eq. (6.17)

To derive Eq. (6.17) we begin by presenting some known matrix identities. For any vector $\mathbf{u}$, $\mathbf{v}$ in $\mathbb{R}^3$ and any general matrix M in $\mathbb{R}^{3\times3}$ the following identities are satisfied.

$$[\mathbf{u}\times]\,\mathbf{v} = -\;[\mathbf{v}\times]\,\mathbf{u} \tag{E.1.1a}$$

$$[\mathbf{u}\times]\;[\mathbf{v}\times] = \mathbf{v}\mathbf{u}^T - \mathbf{v}^T\mathbf{u}\,I_3 \tag{E.1.1b}$$

$$[(\mathbf{u}\times\mathbf{v})\times] = \mathbf{v}\mathbf{u}^T - \mathbf{u}\mathbf{v}^T \tag{E.1.1c}$$

$$\mathbf{v} = [\,tr(M)\;I_3 - M]\,\mathbf{u} \;\;\Rightarrow\;\; [\mathbf{v}\times] = M^T\;[\mathbf{u}\times] + [\mathbf{u}\times]\,M \tag{E.1.1d}$$

$$[\mathbf{u}\times] = M^T - M \;\;\Rightarrow\;\; \mathbf{u}^T\mathbf{v} = \;tr(\,[\mathbf{v}\times]\,M) \tag{E.1.1e}$$

where $[\,\mathbf{a}\times]$ is the cross-product matrix of the general vector $\mathbf{a}$ (Eq. 1.3). All these results arise from the definition of the cross-product matrix and can be easily established by direct computation. Equations (E.1.1a) to (E.1.1d) correspond to Eqs. (A15) to (A18) in [20].

Proposition E.1.1. *Given a K-matrix that is computed from n vector observations [Eqs.* (3.3) *and* (3.4)*), then the vector* $\mathbf{z}$ *and the matrix* B*, defined in Eq.* (3.3)*, are related by the following identity.*

$$[\mathbf{z}\times] = B^T - B \tag{E.1.2}$$

Proof. We prove Eq. (E.1.2) by direct computation. Using Eq. (3.3),

$$
\begin{aligned}
[\mathbf{z}\times] &= \left[\left(\sum_{i=1}^{n} a_{_i}\, \mathbf{b}_{_i} \times \mathbf{r}_{_i}\right)\times\right] \\
&= \sum_{i=1}^{n} a_{_i}\, [(\, \mathbf{b}_{_i} \times \mathbf{r}_{_i})\times] \\
&= \sum_{i=1}^{n} a_{_i}\left(\mathbf{r}_{_i}\, \mathbf{b}_{_i}^T - \mathbf{b}_{_i}\, \mathbf{r}_{_i}^T\right) \\
&= \sum_{i=1}^{n} a_{_i}\, \mathbf{r}_{_i}\, \mathbf{b}_{_i}^T - \sum_{i=1}^{n} a_{_i}\, \mathbf{b}_{_i}\, \mathbf{r}_{_i}^T \\
&= B^T - B
\end{aligned}
\tag{E.1.3}
$$

where the second equation comes from the linearity of the cross-product, the third equation is obtained using Eq. (E.1.1c) with $\mathbf{u} = \mathbf{b}_{_i}$ and $\mathbf{v} = \mathbf{r}_{_i}$, and the last equation results from the definition of the B-matrix in Eq. (3.3). □

Next we derive Eq. (6.17). The time subscripts will be omitted for the sake of clarity. Thus $X_{_k}$ and $\mathcal{E}_{_k}$ are denoted by X and $\mathcal{E}$. Since X is symmetric and $\mathcal{E}$ is skew-symmetric,

$$
X\,\mathcal{E} - \mathcal{E}X = X\,\mathcal{E} + (X\,\mathcal{E})^T \tag{E.1.4}
$$

Using Eqs. (3.4) and (6.18), we compute the first term on the right-hand-side of Eq. (E.1.4)

$$
\begin{aligned}
X\,\mathcal{E} &= \frac{1}{2}\begin{bmatrix} S - \sigma\, I_{_3} & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix}\begin{bmatrix} -\,[\epsilon\times] & \boldsymbol{\epsilon} \\ -\,\boldsymbol{\epsilon}^T & 0 \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} -S\,[\epsilon\times] + \sigma\,[\epsilon\times] - \mathbf{z}\,\boldsymbol{\epsilon}^T & S\,\boldsymbol{\epsilon} - \sigma\,\boldsymbol{\epsilon} \\ -\mathbf{z}^T\,[\epsilon\times] - \sigma\,\boldsymbol{\epsilon}^T & \mathbf{z}^T\,\boldsymbol{\epsilon} \end{bmatrix}
\end{aligned}
\tag{E.1.5}
$$

Using Eq. (E.1.5) in the right-hand side of Eq. (E.1.4) yields

$$
\begin{aligned}
X\,\mathcal{E} - \mathcal{E}X &= \frac{1}{2}\begin{bmatrix} [\epsilon\times]\,S - S\,[\epsilon\times] - \left(\mathbf{z}\,\boldsymbol{\epsilon}^T + \boldsymbol{\epsilon}\mathbf{z}^T\right) & S\,\boldsymbol{\epsilon} + [\epsilon\times]\,\mathbf{z} - 2\sigma\,\boldsymbol{\epsilon} \\ \boldsymbol{\epsilon}^T S - \mathbf{z}^T\,[\epsilon\times] - 2\sigma\,\boldsymbol{\epsilon}^T & 2\,\mathbf{z}^T\,\boldsymbol{\epsilon} \end{bmatrix} \\
&= \begin{bmatrix} M_{11} & \mathbf{m}_{12} \\ \mathbf{m}_{12}^T & m_{22} \end{bmatrix}
\end{aligned}
\tag{E.1.6}
$$

The 3×3 sub-matrix M_{11}, the 3×1 vector $\mathbf{m}_{12}$, and the scalar m_{22} are obviously defined from Eq. (E.1.6). Using Eq. (E.1.1b), with $\mathbf{u} = \mathbf{z}$ and $\mathbf{v} = \boldsymbol{\epsilon}$, yields

$$
[\mathbf{z}\times]\;[\epsilon\times] = \boldsymbol{\epsilon}\mathbf{z}^T - \mathbf{z}^T\,\boldsymbol{\epsilon}\, I_{_3} \tag{E.1.7}
$$

where we use the fact that $\boldsymbol{\epsilon}^T\mathbf{z} = \mathbf{z}^T\,\boldsymbol{\epsilon}$. Substituting Eq. (E.1.2) into the matrix $[\mathbf{z}\times]$ in the left-hand-side of Eq. (E.1.7), we obtain

$$\boldsymbol{\epsilon}\mathbf{z}^T = \left(B^T - B\right)\,[\epsilon\times] + \mathbf{z}^T\,\boldsymbol{\epsilon}\,I_3 \tag{E.1.8}$$

Summing Eq. (E.1.8) with its transpose, and utilizing the skew-symmetry of $[\epsilon\times]$, yields

$$\boldsymbol{\epsilon}\mathbf{z}^T + \mathbf{z}\,\boldsymbol{\epsilon}^T = \left(B^T - B\right)\,[\epsilon\times] - \,[\epsilon\times]\left(B - B^T\right) + 2\,\mathbf{z}^T\,\boldsymbol{\epsilon}\,I_3 \tag{E.1.9}$$

Inserting Eq. (E.1.9) into the expression for M_{11}, using $S = B + B^T$, we get after some development

$$\begin{aligned} M_{11} &= \frac{1}{2}\,\left\{\,[\epsilon\times]\left(B + B^T\right) - \left(B + B^T\right)\,[\epsilon\times] - \left(B^T - B\right)\,[\epsilon\times] + \,[\epsilon\times]\left(B - B^T\right) - 2\,\mathbf{z}^T\,\boldsymbol{\epsilon}\,I_3\right\} \\ &= \left(\,[\epsilon\times]\,B - B^T\,\,[\epsilon\times]\right) - \mathbf{z}^T\,\boldsymbol{\epsilon}\,I_3 \end{aligned} \tag{E.1.10}$$

Let the 3×3 B_ϵ matrix be defined by

$$B_\epsilon \overset{\triangle}{=} \,[\epsilon\times]\,B \tag{E.1.11}$$

Using Eq. (E.1.1e) with $\mathbf{u} = \mathbf{z}$, $M = B$, and $\mathbf{v} = \,\boldsymbol{\epsilon}$, results in

$$\mathbf{z}^T\,\boldsymbol{\epsilon} = \,tr(\,[\epsilon\times]\,B) \tag{E.1.12}$$

Define the matrix S_ϵ and the scalar σ_ϵ by

$$S_\epsilon \overset{\triangle}{=} \,B_\epsilon + \,B_\epsilon{}^T \qquad\quad \sigma_\epsilon \overset{\triangle}{=} \,tr(\,B_\epsilon) \tag{E.1.13}$$

Using Eqs. (E.1.12) and (E.1.13) in Eq. (E.1.10) yields

$$M_{11} = \,S_\epsilon - \,\sigma_\epsilon\,I_3 \tag{E.1.14}$$

The computation of m_{22} is immediate. From Eqs. (E.1.12) and (E.1.13), we write

$$m_{22} = \,\sigma_\epsilon \tag{E.1.15}$$

To compute $\mathbf{m}_{12}$ we start from its expression given in Eq. (E.1.6).

$$\begin{aligned} \mathbf{m}_{12} &= \frac{1}{2}\,\left(S\,\boldsymbol{\epsilon} + \,[\epsilon\times]\,\mathbf{z} - 2\sigma\,\boldsymbol{\epsilon}\right) \\ &= \frac{1}{2}\,\left(S\,\boldsymbol{\epsilon} - \,[\mathbf{z}\times]\,\,\boldsymbol{\epsilon} - 2\sigma\,\boldsymbol{\epsilon}\right) \\ &= \frac{1}{2}\,\left(S - \,[\mathbf{z}\times] - 2\sigma\,I_3\right)\,\boldsymbol{\epsilon} \\ &= \frac{1}{2}\,\left[\left(B + B^T\right) - \left(B^T - B\right) - 2\sigma\,I_3\right]\,\boldsymbol{\epsilon} \\ &= \left(B - \sigma\,I_3\right)\,\boldsymbol{\epsilon} \end{aligned} \tag{E.1.16}$$

where the second equation is obtained from Eq. (E.1.1a) (with $\mathbf{u} = \boldsymbol{\epsilon}$ and $\mathbf{v} = \mathbf{z}$), and the fourth equation is obtained using $S = B + B^T$ and Eq. (E.1.2). In order to compute the cross-product matrix $[\,\mathbf{m}_{12}\times]$, we use the fact that $\sigma = tr(B)$ (see Eq. 3.3), and we apply the proposition (E.1.1d), where $\mathbf{v} = \mathbf{m}_{12}$, $M = -B$, and $\mathbf{u} = \boldsymbol{\epsilon}$; thus

$$\begin{aligned}[\,\mathbf{m}_{12}\times] &= -B^T\,[\epsilon\times] + [\epsilon\times]\,(-B) \\ &= (\,[\epsilon\times]\,B)^T - [\epsilon\times]\,B \\ &= B_\epsilon^{\;T} - B_\epsilon \end{aligned} \tag{E.1.17}$$

where the third equation comes from Eq. (E.1.11). Therefore, denoting the vector $\mathbf{m}_{12}$ by $\mathbf{z}_\epsilon$, we can write

$$[\mathbf{z}_\epsilon\times] = B_\epsilon^{\;T} - B_\epsilon \tag{E.1.18}$$

To conclude, using Eqs. (E.1.14), (E.1.15), and (E.1.18) in Eq. (E.1.6) yields

$$X\,\mathcal{E} - \mathcal{E}X = \begin{bmatrix} S_\epsilon - \sigma_\epsilon\,I_3 & \mathbf{z}_\epsilon \\ \mathbf{z}_\epsilon^T & \sigma_\epsilon \end{bmatrix} \tag{E.1.19}$$

where

$$\begin{array}{ll} B_\epsilon \triangleq [\epsilon\times]\,B & S_\epsilon \triangleq B_\epsilon + B_\epsilon^{\;T} \\ [\mathbf{z}_\epsilon\times] \triangleq B_\epsilon^{\;T} - B_\epsilon & \sigma_\epsilon \triangleq tr(\,B_\epsilon) \end{array} \tag{E.1.20}$$

E.2 Process Noise Covariance Matrix Q_k

We recall that the process equation (6.16) is

$$X_{k+1} = \Phi_k\,X_k\,\Phi_k^T + W_k \tag{E.2.1}$$

The expression for the process noise matrix, W_k, is written here again

$$W_k = (\,X_k\,\mathcal{E}_k - \mathcal{E}_k\,X_k)\,\Delta t \tag{E.2.2}$$

where $\mathcal{E}_k$ is the 4×4 skew-symmetric matrix defined as

$$\mathcal{E}_k = \frac{1}{2}\begin{bmatrix} -\,[\boldsymbol{\epsilon}_k\times] & \boldsymbol{\epsilon}_k \\ -\,\boldsymbol{\epsilon}_k^T & 0 \end{bmatrix} \tag{E.2.3}$$

and $\boldsymbol{\epsilon}_k$ denotes the additive noise error in the gyro outputs. As mentioned earlier, the covariance matrix of W_k, denoted by Q_k, is defined as the covariance of $\operatorname{vec} W_k$, where

$\mathrm{vec}\, W_k$ is the 16×1 vector obtained by applying the vec-operator on W_k. The column-vector $\mathrm{vec}\, W_k$ will be denoted by $\mathbf{w}_k$. Thus,

$$Q_k = \mathrm{cov}\{ W_k \} \stackrel{\triangle}{=} \mathrm{cov}\{ \mathbf{w}_k \} \tag{E.2.4}$$

In order to derive the expression for Q_k given in Eq. (6.23), we apply the vec-operator to Eq. (E.2.2), we obtain a linear relation between $\mathbf{w}_k$ and $\boldsymbol{\epsilon}_k$, and we compute the covariance of $\mathbf{w}_k$ as a function of the covariance of $\boldsymbol{\epsilon}_k$, Q_k^{ϵ}. It is assumed that the matrix Q_k^{ϵ} is known. Applying the vec-operator on Eq. (E.2.2) yields

$$\begin{aligned}
\mathbf{w}_k &= \Delta t\,[\mathrm{vec}\,(X_k\, \mathcal{E}_k - \mathcal{E}_k\, X_k)] \\
&= \Delta t\,[\mathrm{vec}\,(X_k\, \mathcal{E}_k) - \mathrm{vec}\,(\mathcal{E}_k\, X_k)] \\
&= \Delta t\,\left[(I_4 \otimes X_k)\ \mathrm{vec}\, \mathcal{E}_k - \left({X_k}^T \otimes I_4\right)\ \mathrm{vec}\, \mathcal{E}_k\right] \\
&= \Delta t\,\left[(I_4 \otimes X_k) - \left({X_k}^T \otimes I_4\right)\right]\ \mathrm{vec}\, \mathcal{E}_k
\end{aligned} \tag{E.2.5}$$

where $\otimes$ denotes the Kronecker product, the second equation is obtained using the linearity property of the vec-operator, the third equation is derived using a basic property of the Kronecker product (see [58, p. 255]), and the last equation is the result of the factorization with respect to $\mathrm{vec}\, \mathcal{E}_k$. Equation (E.2.5) shows that $\mathbf{w}_k$ is a linear state-dependent function of $\mathrm{vec}\, \mathcal{E}_k$. In order to derive a practical expression for $\mathbf{w}_k$, we approximate the state matrix by its best available estimate. In this case the measurement update estimate, $\widehat{X}_{k/k}$, is substituted for X_k in Eq. (E.2.5), which yields

$$\mathbf{w}_k = \Delta t \left[\left(I_4 \otimes \widehat{X}_{k/k} \right) - \left(\widehat{X}^T_{k/k} \otimes I_4 \right) \right] \ \mathrm{vec}\, \mathcal{E}_k \tag{E.2.6}$$

Applying the vec-operator to Eq. (E.2.3) yields

$$\begin{aligned}
(\operatorname{vec}\mathcal{E}_k)^T &= \left\{ \operatorname{vec}\left(\frac{1}{2} \begin{bmatrix} -[\boldsymbol{\epsilon}_k \times] & \boldsymbol{\epsilon}_k \\ -\boldsymbol{\epsilon}_k^T & 0 \end{bmatrix} \right) \right\}^T \\
&= \frac{1}{2} \left\{ \operatorname{vec}\left(\begin{bmatrix} 0 & \epsilon_3 & -\epsilon_2 & \epsilon_1 \\ -\epsilon_3 & 0 & \epsilon_1 & \epsilon_2 \\ \epsilon_2 & -\epsilon_1 & 0 & \epsilon_3 \\ -\epsilon_1 & -\epsilon_2 & -\epsilon_3 & 0 \end{bmatrix} \right) \right\}^T \\
&= \frac{1}{2} \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 & -\epsilon_1 & \epsilon_3 & 0 & -\epsilon_1 & \epsilon_2 & -\epsilon_2 & \epsilon_1 & 0 & -\epsilon_3 & \epsilon_1 & \epsilon_2 & \epsilon_3 & 0 \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} \\
&= \frac{1}{2} (M\,\boldsymbol{\epsilon}_k)^T
\end{aligned} \tag{E.2.7}$$

where $\boldsymbol{\epsilon}_k = [\ \epsilon_1 \quad \epsilon_2 \quad \epsilon_3\ ]^T$, and the 16×3 matrix M is defined in Eq. (E.2.7). Thus we write

$$\operatorname{vec}\mathcal{E}_k = \frac{1}{2} M\,\boldsymbol{\epsilon}_k \tag{E.2.8}$$

Using Eq. (E.2.8) in Eq. (E.2.5) yields

$$\mathbf{w}_k = \frac{\Delta t}{2} \left[\left(I_4 \otimes \widehat{X}_{k/k} \right) - \left(\widehat{X}_{k/k}^T \otimes I_4 \right) \right] M\,\boldsymbol{\epsilon}_k \tag{E.2.9}$$

Defining the 16×3 matrix Γ_k as

$$\Gamma_k \triangleq \frac{\Delta t}{2} \left[\left(I_4 \otimes \widehat{X}_{k/k} \right) - \left(\widehat{X}_{k/k}^T \otimes I_4 \right) \right] M \tag{E.2.10}$$

and using Eq. (E.2.10) in Eq. (E.2.9) yields

$$\mathbf{w}_k = \Gamma_k\,\boldsymbol{\epsilon}_k \tag{E.2.11}$$

The expression for Q_k, the covariance matrix of $\mathbf{w}_k$, stems from Eq. (E.2.11):

$$Q_k = \Gamma_k Q_k^{\epsilon}\,\Gamma_k^T \tag{E.2.12}$$

E.3 Measurement Noise Covariance Matrix R_{k+1}

In the case of m simultaneous vector observations at time t_{k+1}, the measurement noise term in Eq. (6.24), V_{k+1}, is given by Eq. (6.25), which is written here

$$V_{k+1} = \sum_{i=1}^{m} \alpha_i \, V_{k+1}^i \tag{E.3.1}$$

where

$$V_{k+1}^i = \begin{bmatrix} \mathcal{S}_b{}^i - \sigma_b^i \, I_3 & \mathbf{z}_b^i \\ \mathbf{z}_b^{i\,T} & \sigma_b^i \end{bmatrix} \tag{E.3.2a}$$

$$B_{b_i} = \boldsymbol{\delta b}_i \, \mathbf{r}_i^T \qquad \mathcal{S}_b{}^i = B_{b_i}{}^T + B_{b_i} \tag{E.3.2b}$$

$$\mathbf{z}_b^i = \boldsymbol{\delta b}_i \times \mathbf{r}_i \qquad \sigma_b^i = tr\Big(B_{b_i}\Big) \tag{E.3.2c}$$

$$\alpha_i = a_i / \sum_{i=1}^{m} a_i \tag{E.3.2d}$$

and a_i, $i = 1, 2 \ldots, m$, are positive weights generally chosen as the inverse of the variance of the i^{th} observation error. Let R_{k+1} and R_{k+1}^i denote, respectively, the covariance matrices of V_{k+1} and $\boldsymbol{\delta b}_i(t_{k+1})$, $i = 1, 2 \ldots, m$. We will express V_{k+1} as a linear function of $\boldsymbol{\delta b}_i(t_{k+1})$, $i = 1, 2 \ldots, m$, and we will compute R_{k+1} as a function of the matrices R_{k+1}^i. Let $\mathbf{v}_{k+1}$ and $\mathbf{v}_i(t_{k+1})$ denote the vec-transforms of V_{k+1} and V_{k+1}^i, $i = 1, 2 \ldots, m$, respectively. Applying the vec-operator to Eq. (E.3.1) yields

$$\begin{aligned} \mathbf{v}_{k+1} &= \text{vec}\left(\sum_{i=1}^{m} \alpha_i \, V_{k+1}^i \right) \\ &= \sum_{i=1}^{m} \alpha_i \mathbf{v}_i(t_{k+1}) \end{aligned} \tag{E.3.3}$$

Proposition E.3.1. *The matrix V_{k+1}^i given in Eq.* (E.3.2) *can be factorized as follows*

$$V_{k+1}^i = \mathcal{R}_i^T \, \delta\mathcal{B}_i \tag{E.3.4}$$

where

$$\mathcal{R}_i = \begin{bmatrix} [\mathbf{r}_i \times] & \mathbf{r}_i \\ -\mathbf{r}_i^T & 0 \end{bmatrix} \tag{E.3.5a}$$

$$\delta\mathcal{B}_i = \begin{bmatrix} -[\delta\mathbf{b}_i \times] & \boldsymbol{\delta b}_i \\ -\boldsymbol{\delta b}_i^T & 0 \end{bmatrix} \tag{E.3.5b}$$

Proof. Proposition E.3.1 is proven by direct computation, and using Eq. (E.1.1b), with $\mathbf{u} = \mathbf{r}$, and $\mathbf{v} = \boldsymbol{\delta b}_i$. □

Applying the vec-operator on Eq. (E.3.4) yields

$$\begin{aligned} \mathbf{v}_i(t_{k+1}) &= \operatorname{vec}\left(\mathcal{R}_i^T\,\delta\mathcal{B}_i\right) \\ &= \left(I_4 \otimes \mathcal{R}_i^T\right)\operatorname{vec}(\delta\mathcal{B}_i) \\ &= -\left(I_4 \otimes \mathcal{R}_i\right)\operatorname{vec}(\delta\mathcal{B}_i) \end{aligned} \tag{E.3.6}$$

where the second equation is obtained using a basic property of the Kronecker product (see [58, p. 255]), and the third equation is due to the skew-symmetry of $\mathcal{R}_i$ (see Eq. E.3.5a). The 16×1 vector $\operatorname{vec}(\delta\mathcal{B}_i)$ is expressed as a function of the 3×1 vector $\boldsymbol{\delta b}_i$ as follows.

$$\begin{aligned} [\operatorname{vec}(\delta\mathcal{B}_i)]^T &= \left\{ \operatorname{vec}\left(\begin{bmatrix} 0 & \delta b_3 & -\delta b_2 & \delta b_1 \\ -\delta b_3 & 0 & \delta b_1 & \delta b_2 \\ \delta b_2 & -\delta b_1 & 0 & \delta b_3 \\ -\delta b_1 & -\delta b_2 & -\delta b_3 & 0 \end{bmatrix} \right) \right\}^T \\ &= \begin{bmatrix} \delta b_1 & \delta b_2 & \delta b_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} \\ &= (M\,\boldsymbol{\delta b}_i)^T \end{aligned} \tag{E.3.7}$$

where $\delta b_1, \delta b_2$, and δb_3 are the three components of $\boldsymbol{\delta b}$, and the 16×3 matrix M is defined from the third equation in Eq. (E.3.7). Consequently, we write

$$\operatorname{vec}(\delta\mathcal{B}_i) = M\,\boldsymbol{\delta b}_i \tag{E.3.8}$$

Using Eq. (E.3.8) in Eq. (E.3.6) yields

$$\mathbf{v}_i(t_{k+1}) = -\left(I_4 \otimes \mathcal{R}_i\right) M\,\boldsymbol{\delta b}_i \tag{E.3.9}$$

Defining the 16×16 matrix Λ^i as

$$\Lambda^i \triangleq \left(I_4 \otimes \mathcal{R}_i\right) M \tag{E.3.10}$$

and using Eq. (E.3.10) in Eq. (E.3.9) yields

$$\mathbf{v}_i(t_{k+1}) = -\Lambda^i \boldsymbol{\delta b}_i \tag{E.3.11}$$

Substituting Eq. (E.3.11) for $\mathbf{v}_i(t_{k+1})$ in Eq. (E.3.3) one obtains

$$\mathbf{v}_{k+1} = -\sum_{i=1}^{m} \alpha_i \Lambda^i \boldsymbol{\delta b}_i \tag{E.3.12}$$

Since $\mathbf{v}_{k+1}$ is a linear combination of zero-mean white-noise processes (see Eq. 6.26), it is a zero-mean process itself, and its covariance, R_{k+1}, is readily computed as

$$R_{k+1} = \sum_{i=1}^{m} \alpha_i^2 \Lambda^i R_{k+1}^i \Lambda^{iT} \tag{E.3.13}$$

where R_{k+1}^i is the covariance of the measurement error in $\mathbf{b}_i$, $i = 1, 2 \ldots, m$.

E.4 Proof of the Reduced Covariance Filter

The proof is straightforward and relies on basic properties of the Kronecker product, and on the orthogonal nature of the transition matrix Φ_k. We begin with the time update stage.

Time Update

Proof. Assume that $P_{k/k} = \overline{P}_{k/k} \otimes I_4$ and $Q_k = \overline{Q}_k \otimes I_4$, where $\otimes$ denotes the Kronecker product, then the covariance time update is formulated as (see Eqs. 6.31 and 6.32)

$$\begin{aligned}
P_{k+1/k} &= \mathcal{F}_k P_{k/k} \mathcal{F}_k^T + Q_k \\
&= (\Phi_k \otimes \Phi_k)\left(\overline{P}_{k/k} \otimes I_4\right)(\Phi_k \otimes \Phi_k)^T + \left(\overline{Q}_k \otimes I_4\right) \\
&= \left(\Phi_k \overline{P}_{k/k} \Phi_k^T \otimes \Phi_k \Phi_k^T\right) + \left(\overline{Q}_k \otimes I_4\right) \\
&= \left(\Phi_k \overline{P}_{k/k} \Phi_k^T \otimes I_4\right) + \left(\overline{Q}_k \otimes I_4\right) \\
&= \left(\Phi_k \overline{P}_{k/k} \Phi_k^T + \overline{Q}_k\right) \otimes I_4
\end{aligned}$$

The second equation is obtained using Eqs. (6.31) and the above assumptions. The third equation is obtained using the *mixed product property* of the Kronecker product (Lemma 4.2.10 in [58, p. 243]) and Lemma 4.2.4 of [58, p. 244]. The forth equation results from the orthogonal nature of Φ_k. The last equation is obtained using Lemma 4.2.7 of [58, p. 243]. Looking at the last equation we define the 4×4 matrix $\overline{P}_{k+1/k}$ as

$$\overline{P}_{k+1/k} \triangleq \Phi_k \overline{P}_{k+1/k} \Phi_k^T + \overline{Q}_k$$

which yields

$$P_{k+1/k} = \overline{P}_{k+1/k} \otimes I_4$$

□

Measurement Update

Proof. Assume that $P_{k+1/k} = \overline{P}_{k+1/k} \otimes I_4$ and $R_{k+1} = \overline{R}_{k+1} \otimes I_4$, then the innovations covariance is computed as (see Eq. 6.34)

$$\begin{aligned} S_{k+1} &= P_{k+1/k} + R_{k+1} \\ &= \left(\overline{P}_{k+1/k} \otimes I_4)\right) + (\overline{R}_{k+1} \otimes I_4) \\ &= \left(\overline{P}_{k+1/k} + R_{k+1}\right) \otimes I_4 \end{aligned}$$

The third equation is obtained using Lemma 4.2.7 of [58, p. 243]. Defining the 4×4 matrix $\overline{S}_{k+1}$ as

$$\overline{S}_{k+1} \triangleq \overline{P}_{k+1/k} + R_{k+1}$$

yields

$$S_{k+1} = \overline{S}_{k+1} \otimes I_4$$

The 16×16 Kalman gain matrix, $\mathcal{K}_{k+1}$ is computed from (Eq. 6.35)

$$\begin{aligned} \mathcal{K}_{k+1} &= P_{k+1/k} S_{k+1}^{-1} \\ &= \left(\overline{P}_{k+1/k} \otimes I_4\right) \left(\overline{S}_{k+1} \otimes I_4\right)^{-1} \\ &= \left(\overline{P}_{k+1/k} \otimes I_4\right) \left((\overline{S}_{k+1})^{-1} \otimes I_4\right) \\ &= \left(\overline{P}_{k+1/k} (\overline{S}_{k+1})^{-1}\right) \otimes I_4 \end{aligned}$$

Lemma 4.2.5 and Lemma 4.2.10, respectively, are used to obtain the third and forth equations. Then defining the 4×4 matrix $\overline{K}_{k+1}$ as

$$\overline{K}_{k+1} \triangleq \overline{P}_{k+1/k} (\overline{S}_{k+1})^{-1}$$

yields

$$\mathcal{K}_{k+1} = \overline{K}_{k+1} \otimes I_4 \tag{E.4.1}$$

The estimation error covariance measurement update is formulated as (Eq. 6.38)

$$\begin{aligned}
P_{k+1/k+1} &= \left(I_{16} - \mathcal{K}_{k+1}\right) P_{k+1/k} \left(I_{16} - \mathcal{K}_{k+1}\right)^T + \mathcal{K}_{k+1} R_{k+1} \mathcal{K}_{k+1}^T \\
&= \left[\left(I_4 \otimes I_4\right) - \left(\overline{K}_{k+1} \otimes I_4\right)\right] \left(\overline{P}_{k+1/k} \otimes I_4\right) \left[\left(I_4 \otimes I_4\right) - \left(\overline{K}_{k+1} \otimes I_4\right)\right]^T \\
&+ \left(\overline{K}_{k+1} \otimes I_4\right) \left(\overline{R}_{k+1} \otimes I_4\right) \left(\overline{K}_{k+1} \otimes I_4\right)^T \\
&\left[\left(I_4 - \overline{K}_{k+1}\right) \overline{P}_{k+1/k} \left(I_4 - \overline{K}_{k+1}\right)^T + \overline{K}_{k+1} \overline{R}_{k+1} \overline{K}_{k+1}^T\right] \otimes I_4
\end{aligned}$$

Thus, defining the 4×4 matrix $\overline{P}_{k+1/k+1}$ as

$$\overline{P}_{k+1/k+1} \overset{\triangle}{=} \left(I_4 - \overline{K}_{k+1}\right) \overline{P}_{k+1/k} \left(I_4 - \overline{K}_{k+1}\right)^T + \overline{K}_{k+1} \overline{R}_{k+1} \overline{K}_{k+1}^T$$

yields

$$P_{k+1/k+1} = \overline{P}_{k+1/k+1} \otimes I_4$$

The estimate measurement update, as given in Eq. (6.36), is rewritten here

$$\widehat{X}_{k+1/k+1} = \widehat{X}_{k+1/k} + \sum_{j=1}^{4} \sum_{l=1}^{4} \mathcal{K}_{k+1}^{jl} \, \widetilde{Y}_{k+1} \, E^{lj} \tag{E.4.2}$$

where each 4×4 block-matrix $\mathcal{K}_{k+1}^{jl}$ is, in view of Eq. (E.4.1), a scalar matrix of the form

$$\mathcal{K}_{k+1}^{jl} = \overline{K}_{k+1}[j, l] \, I_4 \tag{E.4.3}$$

and $\overline{K}_{k+1}[j, l]$ denotes the element $[j, l]$ in $\overline{K}_{k+1}$. Using Eq. (E.4.3) in Eq. (E.4.2) yields

$$\begin{aligned}
\widehat{X}_{k+1/k+1} &= \widehat{X}_{k+1/k} + \sum_{j=1}^{4} \sum_{l=1}^{4} \widetilde{Y}_{k+1} \overline{K}_{k+1}[j, l] \, E^{lj} \\
&= \widehat{X}_{k+1/k} + \widetilde{Y}_{k+1} \left(\sum_{j=1}^{4} \sum_{l=1}^{4} \overline{K}_{k+1}[j, l] \, E^{lj} \right) \\
&= \widehat{X}_{k+1/k} + \widetilde{Y}_{k+1} \overline{K}_{k+1}^T
\end{aligned}$$

The first equation is obtained using the fact that $\overline{K}_{k+1}[j, l]$ is a scalar. In the second equation we use the fact that $\widetilde{Y}_{k+1}$ is independent from the summing indices j and l. The third equation comes from the canonical decomposition of a matrix in $\mathbb{R}^{4 \times 4}$. □

Appendix F

F.1 Proof of Eq. (7.6)

Proof. Consider the difference equation (7.2), which is rewritten here

$$D_{k+1} = \Phi^o_k D_k \tag{F.1.1}$$

For a small enough time increment, Δt, we use a first-order approximation in Δt of the Taylor series expression of Φ^o_k. As given in [1, p. 512] this approximate expression is

$$\Phi^o_k = I_3 - \left[\boldsymbol{\omega}^o_k \times\right] \Delta t + h.o.t. \tag{F.1.2}$$

Using Eq. (7.4) in Eq. (F.1.2) yields

$$\begin{aligned}\Phi^o_k &= I_3 - [\boldsymbol{\omega}_k \times] \Delta t + [\boldsymbol{\epsilon}_k \times] \Delta t + h.o.t.\\ &= \Phi_k + [\boldsymbol{\epsilon}_k \times] \Delta t + h.o.t.\end{aligned} \tag{F.1.3}$$

where the second equality is obtained using $\boldsymbol{\omega}_k$ in Eq. (F.1.2). Inserting Eq. (F.1.3) into Eq. (F.1.1) yields

$$\begin{aligned}D_{k+1} &= (\,\Phi_k + [\boldsymbol{\epsilon}_k \times]\ \Delta t)\, D_k + h.o.t.\\ &= \Phi_k D_k + [\boldsymbol{\epsilon}_k \times] D_k\, \Delta t + h.o.t.\\ &= \Phi_k D_k + W_k\end{aligned} \tag{F.1.4}$$

where W_k is defined from the second equality. □

F.2 Proof of Eq. (7.8)

Proof. The approximated expression for the process noise matrix, W_k, as given in Eq. (7.6) is

$$W_k = [\boldsymbol{\epsilon}_k \times] D_k \Delta t \tag{F.2.1}$$

By definition, the covariance matrix of the 3×3 matrix W_k is the 9×9 covariance matrix of the vectorized form of W_k, $\operatorname{vec} W_k$ (see Appendix D.1.4); that is

$$\operatorname{cov}[W_k] \triangleq \operatorname{cov}\left[\operatorname{vec} W_k \, (\operatorname{vec} W_k)^T\right] \tag{F.2.2}$$

The 9×1 vector $\operatorname{vec} W_k$ is expressed as

$$\begin{aligned} \operatorname{vec} W_k &= \operatorname{vec}([\boldsymbol{\epsilon}_k \times] D_k \Delta t) \\ &= \operatorname{vec}([\boldsymbol{\epsilon}_k \times] D_k) \Delta t \\ &= \left(D_k^T \otimes I_3\right)(\operatorname{vec}[\boldsymbol{\epsilon}_k \times]) \Delta t \end{aligned} \tag{F.2.3}$$

where, as before, ⊗ denotes the Kronecker product. The second equality is due to the fact that Δt is scalar and the vec-operator is linear. The third equality is obtained using Lemma 4.3.1 of [58, p. 254]. As shown next, the vec-transform of the cross-product matrix $[\boldsymbol{\epsilon}_k \times]$ is readily expressed as a linear mapping of the 3×1 vector $\boldsymbol{\epsilon}_k$. Recalling that

$$[\boldsymbol{\epsilon}_k \times] = \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix} \tag{F.2.4}$$

and applying the vec-operator to both sides of Eq. (F.2.4) yields, after transposing,

$$\begin{aligned} (\operatorname{vec}[\boldsymbol{\epsilon}_k \times])^T &= \boldsymbol{\epsilon}_k^T \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \boldsymbol{\epsilon}_k^T \left[[\mathbf{e}_1 \times] \quad [\mathbf{e}_2 \times] \quad [\mathbf{e}_3 \times] \right] \\ &= \boldsymbol{\epsilon}_k^T \mathcal{L}^T \end{aligned} \tag{F.2.5}$$

where $\mathbf{e}_j$, $j = 1, 2, 3$, denote the 3×1 vectors with 1 at position j and 0 elsewhere, and the 9×3 matrix $\mathcal{L}$ is defined from the second equality. Inserting the transpose of Eq. (F.2.5) into Eq. (F.2.3) yields

$$\operatorname{vec} W_k = \left(D_k^T \otimes I_3\right) \mathcal{L}\, \boldsymbol{\epsilon}_k\, \Delta t \tag{F.2.6}$$

Subsequently, the covariance matrix of $\operatorname{vec} W_k$, denoted by $\mathcal{Q}_k$, is readily expressed as

$$\begin{aligned}\mathcal{Q}_k &= \operatorname{cov}(W_k)\\ &= \left(D_k^T \otimes I_3\right) \mathcal{L}\left(\operatorname{cov}[\boldsymbol{\epsilon}_k]\right) \mathcal{L}^T \left(D_k^T \otimes I_3\right)^T \ \Delta t^2\\ &= \left(D_k^T \otimes I_3\right) \mathcal{L} Q_k^{\epsilon} \mathcal{L}^T \left(D_k \otimes I_3\right) \ \Delta t^2\end{aligned} \tag{F.2.7}$$

where the third equality is obtained using Lemma 4.2.5 of [58, p. 249]. Finally, since we do not know the true state matrix, D_k, we replace it, in Eq. (F.2.7), by its best available estimate, $\widehat{D}_{k/k}$, which yields the desired result. □

F.3 Derivation of the DCM Reduced Covariance Filter

The assumptions on the stochastic model are:

$$\mathcal{Q}_k = Q_k \otimes I_3 \tag{F.3.1a}$$

$$\mathcal{R}_k = \mu_k I_3 \tag{F.3.1b}$$

$$\mathcal{P}_{0/0} = P_{0/0} \otimes I_3 \tag{F.3.1c}$$

where $\otimes$ denotes the Kronecker product.

Time Update

Assuming that $\mathcal{P}_{k/k} = P_{k/k} \otimes I_3$, the 9×9 covariance time-update equation is formulated (see Eq. 7.34)

$$\begin{aligned}\mathcal{P}_{k+1/k} &= \Psi_k \mathcal{P}_{k/k} \Psi_k^T + \mathcal{Q}_k\\ &= (I_3 \otimes \Phi_k) \left(P_{k/k} \otimes I_3\right) (I_3 \otimes \Phi_k)^T + \mathcal{Q}_k\\ &= \left(P_{k/k} \otimes \Phi_k \Phi_k^T\right) + \mathcal{Q}_k\\ &= \left(P_{k/k} \otimes I_3\right) + (Q_k \otimes I_3)\\ &= \left(P_{k/k} + Q_k\right) \otimes I_3\end{aligned} \tag{F.3.2}$$

The third equality in Eq. (F.3.2) results from the *mixed-product* property of the Kronecker product [58, p. 244]. The forth equality is due to the orthogonality of Φ_k. The last equality is obtained using Lemma 4.2.7 of [58, p. 213]. Thus, defining

$$P_{k+1/k} \overset{\triangle}{=} P_{k/k} + Q_k \tag{F.3.3}$$

yields

$$\mathcal{P}_{k+1/k} = P_{k+1/k} \otimes I_3 \tag{F.3.4}$$

Measurement Update

It is assumed that $\mathcal{P}_{k+1/k} = P_{k+1/k} \otimes I_3$. The innovations covariance matrix, $\mathcal{S}_{k+1}$, is expressed as given in Eq. (7.14)

$$\begin{aligned}\mathcal{S}_{k+1} &= \mathcal{H}_{k+1}\mathcal{P}_{k+1/k}\mathcal{H}^T_{k+1} + R_{k+1}\\ &= \left(\mathbf{r}^T_{k+1} \otimes I_3\right)\left(P_{k+1/k} \otimes I_3\right)\left(\mathbf{r}^T_{k+1} \otimes I_3\right)^T + R_{k+1}\\ &= \left(\mathbf{r}^T_{k+1}\, P_{k+1/k}\, \mathbf{r}_{k+1}\right) I_3 + \mu_{k+1} I_3\\ &= \left(\mathbf{r}^T_{k+1}\, P_{k+1/k}\, \mathbf{r}_{k+1} + \mu_{k+1}\right) I_3\end{aligned} \tag{F.3.5}$$

In Eq. (F.3.5), similarly to Eq. (F.3.4), the third equality results from the *mixed-product* property of the Kronecker product, and the forth equality is obtained using Lemma 4.2.7 of [58, p. 213]. Defining the scalar

$$s_{k+1} \overset{\triangle}{=} \mathbf{r}^T_{k+1}\, P_{k+1/k}\, \mathbf{r}_{k+1} + \mu_{k+1} \tag{F.3.6}$$

S_{k+1} can be written as

$$\mathcal{S}_{k+1} = s_{k+1}\, I_3 \tag{F.3.7}$$

The Kalman gain matrix, $\mathcal{K}_{k+1}$, is computed by

$$\begin{aligned}\mathcal{K}_{k+1} &= \mathcal{P}_{k+1/k}\,\mathcal{H}^T_{k+1}\mathcal{S}^{-1}_{k+1}\\ &= \left(P_{k+1/k} \otimes I_3\right)\left(\mathbf{r}^T_{k+1} \otimes I_3\right)^T \left(s_{k+1}\, I_3\right)^{-1}\\ &= \left(P_{k+1/k}\,\mathbf{r}_{k+1}/s_{k+1}\right) \otimes I_3\\ &= \boldsymbol{g}_{k+1} \otimes I_3\end{aligned} \tag{F.3.8}$$

where obviously $\boldsymbol{g}_{k+1}$ is defined using the third equality of Eq. (F.3.8). Before formulating the covariance measurement-update stage, we express each of its components, as given in Eq. (7.17).

$$\begin{aligned}
\left(I_9 - \mathcal{K}_{k+1}\mathcal{H}_{k+1}\right) &= I_9 - \left(\boldsymbol{g}_{k+1} \otimes I_3\right)\left(\mathbf{r}_{k+1}^T \otimes I_3\right) \\
&= \left(I_3 \otimes I_3\right) - \left(\boldsymbol{g}_{k+1} \otimes I_3\right)\left(\mathbf{r}_{k+1}^T \otimes I_3\right) \\
&= \left(I_3 - \boldsymbol{g}_{k+1}\mathbf{r}_{k+1}^T\right) \otimes I_3 \qquad \text{(F.3.9)} \\
\mathcal{K}_{k+1}\mathcal{R}_{k+1}\mathcal{K}_{k+1}^T &= \left(\boldsymbol{g}_{k+1} \otimes I_3\right)\left(\mu_{k+1} I_3\right)\left(\boldsymbol{g}_{k+1} \otimes I_3\right)^T \\
&= \left(\mu_{k+1}\boldsymbol{g}_{k+1}\boldsymbol{g}_{k+1}^T\right) \otimes I_3 \qquad \text{(F.3.10)}
\end{aligned}$$

Inserting Eqs. (F.3.9) and (F.3.10) into Eq. (7.17) yields

$$\begin{aligned}
\mathcal{P}_{k+1/k+1} &= \left[\left(I_3 - \boldsymbol{g}_{k+1}\mathbf{r}_{k+1}^T\right) \otimes I_3\right]\left(P_{k+1/k} \otimes I_3\right)\left[\left(I_3 - \boldsymbol{g}_{k+1}\mathbf{r}_{k+1}^T\right) \otimes I_3\right]^T + \left(\mu_{k+1}\boldsymbol{g}_{k+1}\boldsymbol{g}_{k+1}^T \otimes I_3\right) \\
&= \left[\left(I_3 - \boldsymbol{g}_{k+1}\mathbf{r}_{k+1}^T\right) P_{k+1/k}\left(I_3 - \boldsymbol{g}_{k+1}\mathbf{r}_{k+1}^T\right)^T + \mu_{k+1}\boldsymbol{g}_{k+1}\boldsymbol{g}_{k+1}^T\right] \otimes I_3 \\
&= P_{k+1/k+1} \otimes I_3 \qquad \text{(F.3.11)}
\end{aligned}$$

The measurement update equation (7.16) is simplified as follows. Substituting Eq. (F.3.8) into Eq. (7.16) yields

$$\begin{aligned}
\widehat{D}_{k+1/k+1} &= \widehat{D}_{k+1/k} + \sum_{j=1}^{3} \boldsymbol{g}_{k+1}[j]\left(\mathbf{b}_{k+1} - \widehat{D}_{k+1/k}\mathbf{r}_{k+1}\right)\mathbf{e}_j^T \\
&= \widehat{D}_{k+1/k} + \left(\mathbf{b}_{k+1} - \widehat{D}_{k+1/k}\mathbf{r}_{k+1}\right)\left(\sum_{j=1}^{3} \boldsymbol{g}_{k+1}[j]\,\mathbf{e}_j^T\right) \\
&= \widehat{D}_{k+1/k} + \left(\mathbf{b}_{k+1} - \widehat{D}_{k+1/k}\mathbf{r}_{k+1}\right)\boldsymbol{g}_{k+1}^T \qquad \text{(F.3.12)}
\end{aligned}$$

where $\boldsymbol{g}_{k+1}[j]$, $j = 1, 2, 3$, denotes the j^{th} component of the 3×1 vector $\boldsymbol{g}_{k+1}$. The second equality in Eq. (F.3.12) results from the fact that $\boldsymbol{g}_{k+1}[j]$ is a scalar, and the vector $\mathbf{b}_{k+1} - \widehat{D}_{k+1/k}\mathbf{r}_{k+1}$ is independent of the index j. The third equality is obtained by recognizing the decomposing of $\boldsymbol{g}_{k+1}$ in the standard basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ of $\mathbb{R}^{3\times 3}$.

F.4 Proof of Proposition 7.5.1

Proof. The proof consists in inserting Eqs. (7.53) into Eq. (7.51) and computing term by term the sum in Eq. (7.51). In the development we use the symbol $O_{i\times j}$ to denote the

$i \times j$ matrix of zeros.

$$\begin{aligned}
\Theta^1_k X_k \Psi^1_k &= \Phi_k \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} \left(E^{11} + E^{22} + E^{33} \right) \\
&= \Phi_k \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} \begin{bmatrix} I_3 & O_{3\times 1} \\ O_{1\times 3} & 0 \end{bmatrix} \\
&= \Phi_k \begin{bmatrix} D_k & O_{3\times 1} \end{bmatrix} \\
&= \begin{bmatrix} (\Phi_k D_k) & O_{3\times 1} \end{bmatrix} \qquad \text{(F.4.1)} \\
\Theta^2_k X_k \Psi^2_k &= I_3 \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} E^{44} \\
&= \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} \begin{bmatrix} O_{3\times 3} & O_{3\times 1} \\ O_{1\times 3} & 1 \end{bmatrix} \\
&= \begin{bmatrix} O_{3\times 3} & \mathbf{c}_k \end{bmatrix} \qquad \text{(F.4.2)}
\end{aligned}$$

Summing Eqs. (F.4.1) and (F.4.2) yields

$$\sum_{r=1}^{2} \Theta^r_k X_k \Psi^r_k = \begin{bmatrix} (\Phi_k D_k) & \mathbf{c}_k \end{bmatrix} \tag{F.4.3}$$

The third term in the sum of Eq. (7.51) is expressed as follows.

$$\begin{aligned}
\Theta^3_k X_k \Psi^3_k &= - \left[\mathbf{d}_{k,1} \times \right] X_k E^{41} \Delta t \\
&= - \left[\mathbf{d}_{k,1} \times \right] \begin{bmatrix} D_k & \mathbf{c}_k \end{bmatrix} \begin{bmatrix} O_{3\times 3} & O_{3\times 1} \\ \Delta t & O_{1\times 3} \end{bmatrix} \\
&= - \left[\mathbf{d}_{k,1} \times \right] \begin{bmatrix} \mathbf{c}_k & O_{4\times 3} \end{bmatrix} \Delta t \\
&= \begin{bmatrix} \left(- \left[\mathbf{d}_{k,1} \times \right] \mathbf{c}_k \right) & O_{3\times 3} \end{bmatrix} \Delta t \\
&= \begin{bmatrix} \left([\mathbf{c}_k \times] \, \mathbf{d}_{k,1} \right) & O_{3\times 3} \end{bmatrix} \Delta t \qquad \text{(F.4.4)}
\end{aligned}$$

where the last equality in Eq. (F.4.4) is the result of the cross-product identity $\mathbf{x} \times \mathbf{y} = -\mathbf{y} \times \mathbf{x}$, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$. Similar computations to those of Eq. (F.4.4) yield

$$\Theta^4_k X_k \Psi^4_k = \begin{bmatrix} O_{3\times 1} & \left([\mathbf{c}_k \times] \, \mathbf{d}_{k,2} \right) & O_{3\times 2} \end{bmatrix} \Delta t \tag{F.4.5}$$

$$\Theta^5_k X_k \Psi^5_k = \begin{bmatrix} O_{3\times 2} & \left([\mathbf{c}_k \times] \, \mathbf{d}_{k,3} \right) & O_{3\times 1} \end{bmatrix} \Delta t \tag{F.4.6}$$

Summing Eqs. (F.4.4) to (F.4.6) yield

$$\sum_{r=3}^{5} \Theta_k^r \, X_k \, \Psi_k^r =$$

$$\begin{aligned}
&\left\{ \begin{bmatrix} ([\mathbf{c}_k \times] \, \mathbf{d}_{k,1}) & O_{3\times3} \end{bmatrix} + \begin{bmatrix} O_{3\times1} & ([\mathbf{c}_k \times] \, \mathbf{d}_{k,2}) & O_{3\times2} \end{bmatrix} + \begin{bmatrix} O_{3\times2} & ([\mathbf{c}_k \times] \, \mathbf{d}_{k,3}) & O_{3\times1} \end{bmatrix} \right\} \Delta t \\
&= [\mathbf{c}_k \times] \begin{bmatrix} \mathbf{d}_{k,1} & \mathbf{d}_{k,2} & \mathbf{d}_{k,3} & O_{3\times1} \end{bmatrix} \Delta t \\
&= [\mathbf{c}_k \times] \begin{bmatrix} D_k & O_{3\times1} \end{bmatrix} \Delta t
\end{aligned} \tag{F.4.7}$$

Summing Eqs. (F.4.4) and (F.4.7) yields

$$\sum_{r=1}^{5} \Theta_k^r \, X_k \, \Psi_k^r = \begin{bmatrix} (\, \Phi_k D_k + [\mathbf{c}_k \times] \, D_k \, \Delta t) & \mathbf{c}_k \end{bmatrix} \tag{F.4.8}$$

Using Eq. (F.4.8) and the expression for W_k, as given in Eqs. (7.52), in Eq. (7.51), we obtain

$$\begin{bmatrix} D_{k+1} & \mathbf{c}_{k+1} \end{bmatrix} = \begin{bmatrix} (\, \Phi_k D_k + [\mathbf{c}_k \times] \, D_k \, \Delta t + \mathcal{E}_k) & \mathbf{c}_k + \boldsymbol{\eta}_k \end{bmatrix} \tag{F.4.9}$$

□

F.5 Proof of Proposition 7.5.2

Proof. Using Eq. (7.55) in Eq. (7.54) yields

$$\begin{aligned}
\mathbf{b}_{k+1} &= X_{k+1} \, \mathbf{r}_{k+1} + \mathbf{v}_{k+1} \\
&= \begin{bmatrix} D_{k+1} & \mathbf{c}_{k+1} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{k+1} \\ 0 \end{bmatrix} + \mathbf{v}_{k+1} \\
&= D_{k+1} \, \mathbf{r}_{k+1} + \mathbf{v}_{k+1}
\end{aligned} \tag{F.5.1}$$

□

Appendix G

G.1 Proof of Eq. (8.3)

Proof. Using the definition of the H-matrix, as given in Eqs. (2.33a) to (2.33c), yields

$$
\begin{aligned}
C &\overset{\triangle}{=} H^T H \\
&= H H^T \\
&= \tfrac{1}{4}\begin{bmatrix} -[(\mathbf{b}+\mathbf{r})\times] & (\mathbf{b}-\mathbf{r}) \\ -(\mathbf{b}-\mathbf{r})^T & 0 \end{bmatrix}\begin{bmatrix} -[(\mathbf{b}+\mathbf{r})\times] & (\mathbf{b}-\mathbf{r}) \\ -(\mathbf{b}-\mathbf{r})^T & 0 \end{bmatrix}^T \\
&= \tfrac{1}{4}\begin{bmatrix} -[(\mathbf{b}+\mathbf{r})\times] & (\mathbf{b}-\mathbf{r}) \\ -(\mathbf{b}-\mathbf{r})^T & 0 \end{bmatrix}\begin{bmatrix} [(\mathbf{b}+\mathbf{r})\times] & -(\mathbf{b}-\mathbf{r}) \\ (\mathbf{b}-\mathbf{r})^T & 0 \end{bmatrix} \\
&= \tfrac{1}{4}\begin{bmatrix} [(\mathbf{b}+\mathbf{r})\times]^2 + (\mathbf{b}-\mathbf{r})\,(\mathbf{b}-\mathbf{r})^T & [(\mathbf{b}+\mathbf{r})\times]\,(\mathbf{b}-\mathbf{r}) \\ \{-[(\mathbf{b}+\mathbf{r})\times]\,(\mathbf{b}-\mathbf{r})\}^T & (\mathbf{b}-\mathbf{r})^T\,(\mathbf{b}-\mathbf{r}) \end{bmatrix} \\
&= \begin{bmatrix} C_{11} & \mathbf{c}_{12} \\ \mathbf{c}_{12}^T & c_{22} \end{bmatrix}
\end{aligned}
\tag{G.1.1}
$$

The 3×3 submatrix C_{11} is computed as follows:

$$
\begin{aligned}
C_{11} &= \tfrac{1}{4}\,[(\mathbf{b}+\mathbf{r})\times]^2 + (\mathbf{b}-\mathbf{r})\,(\mathbf{b}-\mathbf{r})^T \\
&= \tfrac{1}{4}\left[(\mathbf{b}+\mathbf{r})\,(\mathbf{b}+\mathbf{r})^T\, I_3 - (\mathbf{b}+\mathbf{r})\,(\mathbf{b}+\mathbf{r})^T\right] \\
&= \tfrac{1}{4}\left(2\, I_3 + 2\, \mathbf{b}^T\mathbf{r}\, I_3 - 2\, \mathbf{b}\,\mathbf{r}^T - 2\, \mathbf{r}\,\mathbf{b}^T\right) \\
&= \tfrac{1}{2}\, I_3 - \left(\mathbf{b}\,\mathbf{r}^T + \mathbf{r}\,\mathbf{b}^T - \mathbf{b}^T\mathbf{r}\, I_3\right)
\end{aligned}
\tag{G.1.2}
$$

where the second equality is obtained using the identity

$$
[\mathbf{u}\times]\ [\mathbf{v}\times] = \mathbf{v}\mathbf{u}^T - \mathbf{u}^T\mathbf{v}\, I_3 \tag{G.1.3}
$$

for all $\mathbf{u}$, $\mathbf{v}$ in $\mathbb{R}^3$. The 3×1 column-matrix $\mathbf{c}_{12}$ is computed as follows.

$$\begin{aligned}
\mathbf{c}_{12} &= \tfrac{1}{4}\,[(\mathbf{b}+\mathbf{r})\times]\,(\mathbf{b}-\mathbf{r}) \\
&= \tfrac{1}{4}\,(-\,[\mathbf{b}\times]\,\mathbf{r}+[\mathbf{r}\times]\,\mathbf{b}) \\
&= -\tfrac{1}{2}\,[\mathbf{b}\times]\,\mathbf{r}
\end{aligned} \tag{G.1.4}$$

where we used the skew symmetric property of the cross-product. Finally, the scalar c_{22} is expressed by

$$\begin{aligned}
c_{22} &= \tfrac{1}{4}\,(\mathbf{b}-\mathbf{r})^T\,(\mathbf{b}-\mathbf{r}) \\
&= \tfrac{1}{4}\,\left(\mathbf{b}^T\mathbf{b}+\mathbf{r}^T\mathbf{r}-2\,\mathbf{b}^T\mathbf{r}\right) \\
&= \tfrac{1}{2}\,\left(1-\mathbf{b}^T\mathbf{r}\right)
\end{aligned} \tag{G.1.5}$$

where the third equality is obtained using the fact that $\mathbf{b}$ and $\mathbf{r}$ are unit vectors. Using Eqs. (G.1.2) to (G.1.5) in Eq. (G.1.1) yields

$$\begin{aligned}
C &= \begin{bmatrix} \tfrac{1}{2}\,I_3-\left(\mathbf{b}\,\mathbf{r}^T+\mathbf{r}\,\mathbf{b}^T-\mathbf{b}^T\mathbf{r}\,I_3\right) & -\tfrac{1}{2}\,[\mathbf{b}\times]\,\mathbf{r} \\ \tfrac{1}{2}\,\mathbf{r}^T\,[\mathbf{b}\times] & \tfrac{1}{2}\,\left(1-\mathbf{b}^T\mathbf{r}\right) \end{bmatrix} \\
&= \frac{1}{2}\left(\begin{bmatrix} I_3 & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{b}\,\mathbf{r}^T+\mathbf{r}\,\mathbf{b}^T-\mathbf{b}^T\mathbf{r}\,I_3 & \mathbf{b}\times\mathbf{r} \\ (\mathbf{b}\times\mathbf{r})^T & \mathbf{b}^T\mathbf{r} \end{bmatrix}\right) \\
&= \frac{1}{2}\left(I_4-\begin{bmatrix} B+B^T-\sigma\,I_3 & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix}\right) \\
&= \frac{1}{2}\,(I_4-K)
\end{aligned} \tag{G.1.6}$$

where, in the third equality, B, $\mathbf{z}$, and σ are defined as

$$B \stackrel{\triangle}{=} \mathbf{b}\,\mathbf{r}^T \tag{G.1.7}$$

$$\mathbf{z} \stackrel{\triangle}{=} \mathbf{b}\times\mathbf{r} \tag{G.1.8}$$

$$\sigma \stackrel{\triangle}{=} \mathbf{b}^T\mathbf{r} \tag{G.1.9}$$

□

Bibliography

[1] Wertz, J.R. (ed.), *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, The Netherlands, 1984.

[2] Black, H.D., “A Passive System for Determining the Attitude of a Satellite,” *AIAA Journal*, Vol. 2, No. 7, 1964, pp. 1350,1351.

[3] Keat, J., “Analysis of Least Squares Attitude Determination Routine DOAOP,” Computer Sciences Corp., CSC/TM-77/6034, Silver Spring, MD, Feb. 1977.

[4] Stuelpnagel, J., “On the Parametrization of the Three-Dimensional Rotation Group,” *SIAM Review*, Vol. 6, No. 4, October 1964, pp. 422–430.

[5] Wahba, G., “A Least Squares Estimate of Spacecraft Attitude,” *SIAM Review*, Vol. 7, No. 3, 1965, p. 409.

[6] Farrel, J.L. and Stuelpnagel, “A Least Squares Estimate of Spacecraft Attitude,” *SIAM Review*, Vol. 8, No. 3, 1966, pp. 384–386.

[7] Wessner, R.H., “A Least Squares Estimate of Spacecraft Attitude,” *SIAM Review*, Vol. 8, No. 3, 1966, pp. 384–386.

[8] Velman, J.R., “A Least Squares Estimate of Spacecraft Attitude,” *SIAM Review*, Vol. 8, No. 3, 1966, pp. 384–386.

[9] Brock, J.E., “A Least Squares Estimate of Spacecraft Attitude,” *SIAM Review*, Vol. 8, No. 3, 1966, pp. 384–386.

[10] Brock, J.E., “Optimal Matrices Describing Linear Systems,” *AIAA Journal*, Vol. 6, No. 7, Jul. 1968, pp. 1292–1296.

[11] Markley, F.L., “Attitude Determination Using Vector Observations: a Fast Optimal Matrix Algorithm,” *Journal of the Astronautical Sciences*, Vol. 41, No. 2, Apr.-Jun. 1993, pp. 261–280.

[12] Markley, F.L., “Attitude Determination Using Vector Observations and the Singular Value Decomposition,” *Journal of the Astronautical Sciences*, Vol. 36, No. 3, July-Sept. 1988, pp. 245–258.

[13] Bar-Itzhack, I.Y., “Polar Decomposition for Attitude Determination from Vector Observations,” *Flight Mechanics/Estimation Theory Symposium*, Goddard Space Flight Center, Greenbelt, MD, May 1992, NASA Conference Publication 3186, pp. 243–257.

[14] Bar-Itzhack, I.Y., and Markley, F.L., “Unconstrained Optimal Transformation Matrix,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, Jan. 1998, pp. 338–340.

[15] Bellman, R., *Introduction to Matrix Analysis*, McGraw-Hill, New-York, 1960.

[16] Golub, G.H. and Van Loan, C.F., *Matrix Computations*, the Johns Hopkins University Press, Baltimore, 1983.

[17] Shuster, M.D., and Oh, S.D., “Three-Axis Attitude Determination from Vector Observations,” *Journal of Guidance and Control*, Vol. 4, No. 1, 1981, pp. 70–77.

[18] Mortari, D., “A Closed-Form Solution to the Wahba Problem,” *Journal of the Astronautical Sciences*, Vol. 45, No. 2, April-June 1997, pp. 195–204.

[19] Mortari, D., “Second Estimator of the Optimal Quaternion,” *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, Sept.-Oct. 2000, pp. 885–887.

[20] Markley, F.L., “Attitude Determination Using Vector Observations: Theory,” *Journal of the Astronautical Sciences*, Vol. 37, No. 1, January-March 1989, pp. 41–58.

[21] Psiaki, M.L., “Attitude Determination Filtering via Extended Quaternion Estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, March-April 2000, pp. 210–214.

[22] Shuster, M.D., “A Simple Kalman Filter and Smoother for Spacecraft Attitude,” *Journal of the Astronautical Sciences*, Vol. 37, No. 1, January-March 1989, pp. 89–106.

[23] Gelb, A. (ed.), *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1979.

[24] Bar-Itzhack, I.Y., “REQUEST: a Recursive QUEST Algorithm for Sequential Attitude Determination,” *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, Sept.-Oct. 1996, pp. 1034–1038.

[25] Shuster, M.D., “Maximum Likelihood Estimation of Spacecraft Attitude,” *Journal of the Astronautical Sciences*, Vol. 37, No. 1, January-March 1989, pp. 79–88.

[26] Lefferts, E.J., Markley, F.L., and Shuster, M.D., “Kalman Filtering for Spacecraft Attitude Estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 5, Sept.-Oct. 1982, pp. 417–429.

[27] Bar-Itzhack, I.Y., and Oshman, Y., “Attitude Determination from Vector Observations: Quaternion Estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol-AES-21, Jan. 1985, pp. 128–136.

[28] Bar-Itzhack, I.Y., Deutschmann, J., and Markley, F.L., “Quaternion Normalization in Additive EKF for Spacecraft Attitude Determination,” AIAA Paper 91-2706, *Guidance, Navigation, and Control Conference*, New Orleans, LA, August 1991.

[29] Bar-Itzhack, I.Y., “Optimum Normalization of a Computed Quaternion of Rotation,” *IEEE Transactions on Aerospace and Electronic Systems*, AES-7, Mar. 1971, pp. 401,402.

[30] Kalman, R.E., “A New Approach to Linear Filtering and Prediction Problems,” *Trans. ASME, J. Basic Eng.*, Ser. D, Vol. 82, March 1960, pp. 35–45.

[31] Kalman, R.E., and Bucy, R.S., “New Results in Linear Filtering and Prediction Theory,” *Trans. ASME, J. Basic Eng.*, Ser. D, Vol. 83, March 1961, pp. 95–108.

[32] Baruch M., and Bar-Itzhack, I. Y., “Optimal Weighted Orthogonalization of Measured Modes,” *AIAA Journal*, Vol. 16, No.4, pp. 346-351.

[33] Curtain, R.F., *An Introduction to Infinite-Dimensional Linear Systems Theory*, Springer-Verlag New York Inc., July 1995, 697 pages.

[34] Bar-Itzhack, I.Y. and Reiner, J., "Recursive Attitude Determination from Vector Observations: DCM Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 7, Jan.-Feb. 1984, pp. 51–56.

[35] Choukroun, D., Weiss, H., Bar-Itzhack I.Y., Oshman, Y., "Quaternion Estimation Using Kalman Filtering of the Vectorized K-matrix," TAE 916, Technical Report, Faculty of Aerospace Engineering, Technion, Israel Inst. of Techn., March 2003.

[36] Athans, M., and Tse, E., "A Direct Derivation of the Optimal Linear Filter Using the Maximum Principle," *IEEE Transactions on Automatic Control*, Vol. AC-12, No. 6, 1967, pp. 690–698.

[37] Athans, M., "The Matrix Maximum Principle," *Information and Control*, Vol. 11, Nos. 5-6, 1967, pp. 592–606.

[38] Tracy, D.S., and Dwyer, P.S., "Multivariate Maxima and Minima with Matrix Derivatives," *J. Amer. Statist. Assoc.*, Vol. 64, 1969, pp. 1576–1594.

[39] Van Huffel, S., and Vandewalle, J., *The Total Least Squares Problem: Computational Aspects and Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[40] Mendel, J.M., *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, Prentice Hall PTR, Englewood Cliffs, N.J., 1995.

[41] Neudecker, H., "Some Theorems on Matrix Differentiation with special References to Kronecker Products," *J. Amer. Statist. Assoc.*, Vol. 64, 1969, pp. 953–963.

[42] MacRae, E.C., "Matrix Derivatives with an Application to an Adaptive Linear Decision Problem," *The Annals of Statistics*, Vol. 2, No.2 1974, pp. 337–346.

[43] Nissen, D.H., "A note on the Variance of a Matrix," *Econometrica*, Vol. 36, 1968, pp. 603–604.

[44] Goodwin, G.C., and Sin, K.S., *Adaptive Filtering Prediction and Control*, Englewood Cliffs, N.J., Prentice Hall, 1984.

[45] Choukroun, D., Oshman, Y., and Bar-Itzhack, I.Y., "A Novel Quaternion Kalman Filter," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Monterey, CA, Aug., 2002.

[46] Jazwinski, A.H., *Stochastic Processes and Filtering Theory*, New York: Academic, 1970.

[47] Cohen, C.E., "Attitude Determination Using GPS," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, Dec. 1992.

[48] Lightsey, E.G., Cohen, C.E., Feess, W.A., and Parkinson, B.M., "Analysis of Spacecraft Attitude Measurements Using Onboard GPS," *Advances in Astronautical Sciences*, Vol. 86, pp. 521-532; also AAS Paper 94-063.

[49] Axelrad, P., and Behre, C.P., "Attitude Estimation Algorithms for Spinning Satellites Using Global Positioning System Phase Data," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 164–169.

[50] Oshman, Y., and Markley, F.L., "Spacecraft Attitude/Rate Estimation Using Vector-Aided GPS Observations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 3, July 1999, pp. 1019–1027.

[51] Conway, A., Montgomery, P., Rock, S., Cannon, R., and Parkinson, B., "A New Motion-Based Algorithm for GPS Attitude Integer Resolution," *Navigation*, Vol. 43, No. 2, 1996, pp. 179–190.

[52] Garrick,J., "Investigation of Models and Estimation Techniques for GPS Attitude Determination," *1996 NASA-GSFC Flight Dynamics Division Flight Mechnaics/Estimation Theory Symposium*, NASA CP-3333, NASA Goddard Space Flight Center, greenbelt, MD, may 1996, pp. 89–98.

[53] Hwang, P.Y. C., "Kinematics GPS for Differential Position: Resolving Integer Ambiguities on the Fly," *Navigation*, Vol. 38, No. 1, 1991, pp. 1–15.

[54] Choukroun, D., Oshman, Y., and Bar-Itzhack, I.Y., "Optimal-REQUEST Algorithm for Attitude Determination," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Montreal, Quebec, Aug., 2001.

[55] Cox, H., "On the Estimation of State Variables and Parameters for Noisy Dynamic Systems," *IEEE Transactions on Automatic Control*, Vol. AC-9, 1964, pp. 5–12.

[56] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, New York: Academic, 1977.

[57] Bar-Itzhack, I.Y., "On Matrix Symmetrization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No.1, Jan.-Feb. 1998, pp. 178–179.

[58] Horn, R.A., and Johnson, C.R., *Topics in Matrix Analysis*, Cambridge University press, Cambridge, 1991.

[59] Henderson, H.V., and Searle, S.R., "The Vec-Permutation Matrix, The Vec-Operator and Kronecker Products: A review," *Linear Multilinear Algebra*, Vol. 9, 1981, pp. 271–288.

[60] Bar-Itzhack, I.Y., and Meyer, J., "On the Convergence of Iterative Orthogonalization Processes," *IEEE Transactions on Aerospace and Electronic Systems*, Vol-AES-12, Jan. 1976, pp. 146–151.

[61] Markley, F.L., Mortari, D., "How to Compute Attitude from Vector Observations," AIAA/AAS Paper 99-427, *Astrodynamics Specialist Conference*, Girdwood, AK, August 15-19, 1999.

[62] Bar-Itzhack, I.Y., and Harman, R.R., "The MAP Spacecraft Angular State Estimation After Sensor Failure," *43rd Israel Annual Conference on Aerospace Sciences*, Tel Aviv, Israel, February 19–20, 2003.