Xtrinsic Sensor Fusion Kalman Filter (fusion.c)

Technical Note

Document XXX
Version: Draft
Authors: Mark Pedley, Michael Stanley and Zbigniew Baranski
Date: xx Aug 2014

Table of Contents

1 Introduction 6
- 1.1 Summary 6
- 1.2 References 6
- 1.3 Functions 6

2 Mathematical Background 7
- 2.1 Vector Product 7
- 2.2 Incremental Rotations 7
- 2.3 Product of Small Rotation Matrices 8

3 Accelerometer, Magnetometer, Gyroscope Kalman Filter 9
- 3.1 Model of the Error Process 9
- 3.2 Model of the Measurement Error Process 9
- 3.3 Kalman Filter Equations 10

4 Sensor Models 12
- 4.1 Gyroscope Sensor Model 12
- 4.2 Accelerometer Sensor Model 12
- 4.3 Magnetometer Sensor Model 13

5 Orientation Estimates 14
- 5.1 A Priori Orientation Estimate 14
- 5.2 A Posteriori Orientation Estimate 14

6 Acceleration Estimates 15
- 6.1 A Priori Acceleration Estimate 15
- 6.2 A Posteriori Acceleration Estimate 15

7 Measurement Error and Matrix 16
- 7.1 Introduction 16
- 7.2 Calculation of the Measurement Error Vector 16
- 7.3 Basic Model of the Measurement Matrix 17
- 7.4 Extended Model of the Measurement Matrix 18

8 A Posteriori Error Vector Calculation 19
- 8.1 Magnetic Jamming Flag 19
- 8.2 Normal Case 19
- 8.3 Magnetic Jamming Case 19

9 Geomagnetic Inclination Angle Tracking 20
- 9.1 Introduction 20
- 9.2 Algorithm 20
- 9.3 Geometric Implementation 21

10 Covariance Matrix Updates 22
- 10.1 Process Error Qw Matrix Terms 22
- 10.2 Measurement Error Covariance Qv Matrix 24

Glossary

Superscript - denotes an *a priori* estimate.

Superscript + denotes an *a posteriori* estimate.

Subscript ε denotes an error component.

Subscript k denotes a measurement at iteration k.

Subscript A denotes an accelerometer measurement or estimate.

Subscript G denotes a gyro measurement or estimate.

Subscript M denotes a magnetometer measurement or estimate.

${}^{G}\boldsymbol{a}_k, {}^{S}\boldsymbol{a}_k$	The linear acceleration in global and sensor frames (units g)
${}^{G}\boldsymbol{a}_{\varepsilon,k}, {}^{S}\boldsymbol{a}_{\varepsilon,k}$	The linear acceleration error vector in global and sensor frames (units g)
$\boldsymbol{A}_k$	The linear prediction or state matrix for the error process $\boldsymbol{x}_{\varepsilon,k}$
$\boldsymbol{b}_k$	The gyro offset vector (units deg/s)
$\boldsymbol{b}_{\varepsilon,k}$	The gyro offset error vector (units deg/s)
B	The local geomagnetic field strength (units uT)
c_a	Filter decay coefficient for linear acceleration Markov process
c_d	Filter decay coefficient for magnetic disturbance Markov process
$\boldsymbol{C}_k$	The measurement matrix relating the measured process $\boldsymbol{z}_\varepsilon$ to the underlying process $\boldsymbol{x}_\varepsilon$
${}^{G}\boldsymbol{d}_k, {}^{S}\boldsymbol{d}_k$	The magnetic disturbance in global and sensor frames (units uT)
${}^{G}\boldsymbol{d}_{\varepsilon,k}, {}^{S}\boldsymbol{d}_{\varepsilon,k}$	The magnetic disturbance error vector in global and sensor frames (units uT)
${}^{G}\boldsymbol{g}_k$	The constant downwards pointing gravity vector in the global frame (units g).
${}^{G}\boldsymbol{g}_{A,k}, {}^{S}\boldsymbol{g}_{A,k}$	Gravity vector estimate from accelerometer in global and sensor frames (units g)
${}^{G}\boldsymbol{g}_{G,k}, {}^{S}\boldsymbol{g}_{G,k}$	Gravity vector estimate from gyro in global and sensor frames (units g)
${}^{G}\boldsymbol{g}_{\varepsilon,AG,k}$	Gravity error vector between accelerometer and gyro estimates in global frame (units g) ${}^{G}\boldsymbol{g}_{\varepsilon,AG,k} = {}^{G}\boldsymbol{g}_{A,k} - {}^{G}\boldsymbol{g}_{G,k}$
${}^{S}\boldsymbol{g}_{\varepsilon,AG,k}$	Gravity error vector between accelerometer and gyro estimates in sensor frame (units g) ${}^{S}\boldsymbol{g}_{\varepsilon,AG,k} = {}^{S}\boldsymbol{g}_{A,k} - {}^{S}\boldsymbol{g}_{G,k}$
$\boldsymbol{I}_n$	n by n identity matrix
$\boldsymbol{K}_k$	The Kalman filter gain at iteration k
${}^{G}\boldsymbol{m}_k$	The constant geomagnetic vector in the global frame (units uT).
${}^{G}\boldsymbol{m}_{M,k}, {}^{S}\boldsymbol{m}_{M,k}$	Geomagnetic vector estimate from magnetometer in global and sensor frames (units uT)
${}^{G}\boldsymbol{m}_{G,k}, {}^{S}\boldsymbol{m}_{G,k}$	Geomagnetic vector estimate from gyro in global and sensor frames (units uT)
${}^{G}\boldsymbol{m}_{\varepsilon,MG,k}$	Geomagnetic error vector between magnetometer and gyro in global frame (units uT) ${}^{G}\boldsymbol{m}_{\varepsilon,MG,k} = {}^{G}\boldsymbol{m}_{M,k} - {}^{G}\boldsymbol{m}_{G,k}$
${}^{S}\boldsymbol{m}_{\varepsilon,MG,k}$	Geomagnetic error vector between magnetometer and gyro in sensor frame (units uT) ${}^{S}\boldsymbol{m}_{\varepsilon,MG,k} = {}^{S}\boldsymbol{m}_{M,k} - {}^{S}\boldsymbol{m}_{G,k}$
$\boldsymbol{P}_k^-$	The covariance matrix of the errors in the *a priori* error estimates

	$\boldsymbol{P}_k^- = E\left[\left(\widehat{\boldsymbol{x}}_{\varepsilon,k}^- - \boldsymbol{x}_{\varepsilon,k}\right)\left(\widehat{\boldsymbol{x}}_{\varepsilon,k}^- - \boldsymbol{x}_{\varepsilon,k}\right)^T\right] = E\left[\widehat{\boldsymbol{x}}_{\varepsilon,k}^- {\widehat{\boldsymbol{x}}_{\varepsilon,k}^-}^T\right]$
$\boldsymbol{P}_k^+$	The covariance matrix of the errors in the *a posteriori* error estimates $\boldsymbol{P}_k^+ = E\left[\left(\widehat{\boldsymbol{x}}_{\varepsilon,k}^+ - \boldsymbol{x}_{\varepsilon,k}\right)\left(\widehat{\boldsymbol{x}}_{\varepsilon,k}^+ - \boldsymbol{x}_{\varepsilon,k}\right)^T\right] = E\left[\widehat{\boldsymbol{x}}_{\varepsilon,k}^+ {\widehat{\boldsymbol{x}}_{\varepsilon,k}^+}^T\right]$
$\boldsymbol{Q}_{w,k}$	Covariance matrix of noise process $\boldsymbol{w}_k$ in the underlying process $\boldsymbol{x}$ $\boldsymbol{Q}_{w,k} = cov\{\boldsymbol{w}_k, \boldsymbol{w}_k\} = E[\boldsymbol{w}_k {\boldsymbol{w}_k}^T]$
$\boldsymbol{Q}_{v,k}$	Covariance matrix of measurement noise process $\boldsymbol{v}_k$ in the measured process $\boldsymbol{z}$ $\boldsymbol{Q}_{v,k} = cov\{\boldsymbol{v}_k, \boldsymbol{v}_k\} = E[\boldsymbol{v}_k {\boldsymbol{v}_k}^T]$
q_k	True orientation quaternion transforming from global to sensor frame at iteration k
$\hat{q}_k^-$	*A priori* estimate of the true orientation quaternion q_k
$\hat{q}_k^+$	*A posteriori* estimate of the true orientation quaternion q_k
$\boldsymbol{R}_k$	True orientation matrix transforming from global to sensor frame at iteration k
$\widehat{\boldsymbol{R}}_k^-$	*A priori* estimate of the true orientation matrix $\boldsymbol{R}_k$
$\widehat{\boldsymbol{R}}_k^+$	*A posteriori* estimate of the true orientation matrix $\boldsymbol{R}_k$
$\boldsymbol{v}_k$	Additive noise in measurement error process $\boldsymbol{z}_{\varepsilon,k}$: $\boldsymbol{z}_{\varepsilon,k} = \boldsymbol{C}_k \boldsymbol{x}_{\varepsilon,k} + \boldsymbol{v}_k$
$\boldsymbol{v}_{A,k}$	Accelerometer sensor additive noise (units g)
$\boldsymbol{v}_{G,k}$	Gyro sensor additive noise (units deg/s)
$\boldsymbol{v}_{M,k}$	Magnetometer sensor instantaneous additive noise (units uT)
$\boldsymbol{w}_k$	Additive noise in error of underlying process $\boldsymbol{x}_{\varepsilon,k}$: $\boldsymbol{x}_{\varepsilon,k} = \boldsymbol{A}_k \boldsymbol{x}_{\varepsilon,k-1} + \boldsymbol{w}_k$
$\boldsymbol{w}_{a,k}$	Driving noise for linear acceleration low pass Markov process (units g)
$\boldsymbol{w}_{b,k}$	Driving noise for gyro offset random walk (units deg/s)
$\boldsymbol{w}_{d,k}$	Driving noise for magnetic disturbance low pass Markov process (units uT)
$\boldsymbol{x}_{\varepsilon,k}$	The underlying error process
$\widehat{\boldsymbol{x}}_{\varepsilon,k}^-$	The *a priori* estimate of error process $\boldsymbol{x}_{\varepsilon,k}$
$\widehat{\boldsymbol{x}}_{\varepsilon,k}^+$	The *a posteriori* estimate of error process $\boldsymbol{x}_{\varepsilon,k}$
$\boldsymbol{y}_{A,k}$	Accelerometer reading (g) which by definition is in the sensor frame
$\boldsymbol{y}_{G,k}$	Gyro reading (units deg/s) which by definition is in the sensor frame
$\boldsymbol{y}_{M,k}$	Magnetometer reading (units uT) which by definition is in the sensor frame
${}^G\boldsymbol{z}_{\varepsilon,k}, {}^S\boldsymbol{z}_{\varepsilon,k}$	The measurement error vector in global and sensor frames
δt	Sampling interval or small time interval (units s)
$\boldsymbol{\theta}_{\varepsilon,k}$	True orientation error vector (units deg)
$\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-$	The *a priori* estimate of the orientation error vector $\boldsymbol{\theta}_{\varepsilon,k}$
$\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+$	The *a posteriori* estimate of the orientation error vector $\boldsymbol{\theta}_{\varepsilon,k}$
$\boldsymbol{\omega}_k$	True angular velocity (deg/s)

$\boldsymbol{\omega}_k^-$	The *a priori* estimate of the angular velocity $\boldsymbol{\omega}_k$ (deg/s)
$(\boldsymbol{\omega}\times)$	The matrix constructed from the vector $\boldsymbol{\omega}$ to be equivalent to vector product operation

1 Introduction

1.1 Summary

This document derives the mathematics of the complementary indirect Kalman filter (function fRun_9DOF_GBY_KALMAN) used to fuse the accelerometer, magnetometer and gyroscope data (termed 9DOF or nine degree of freedom sensor fusion).

The 6DOF Kalman filter for six degree of freedom fusion of accelerometer and gyroscope data (function fRun_6DOF_GY_KALMAN) is an obvious simplification of the 9DOF algorithm and is not derived separately.

The term complementary indicates that the Kalman filter balances orientation estimates coming from i) the accelerometer and magnetometer and ii) from the gyroscope sensor. The term indirect indicates that the Kalman filter operates on the error vector rather than the state vector itself.

The introductory document "Basic Kalman Filter Theory" should be read and understood before tackling this document.

1.2 References

[1] "Compensation of Magnetic Disturbances Improves Inertial and Magnetic Sensing of Human Body Segment Orientation", Daniel Roetenberg, Henk J. Luinge, Chris T. M. Baten, and Peter H. Veltink, IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 13, No. 3, September 2004

1.3 Functions

void fInit_9DOF_GBY_KALMAN(struct SV_9DOF_GBY_KALMAN *pthisSV, int16 ithisCoordSystem, int16 iSensorFS, int16 iOverSampleRatio) Initialization of accelerometer, magnetometer and gyroscope sensor fusion algorithm.
void fRun_9DOF_GBY_KALMAN(struct SV_9DOF_GBY_KALMAN *pthisSV, struct AccelSensor *pthisAccel, struct MagSensor *pthisMag, struct GyroSensor *pthisGyro, struct MagCalibration *pthisMagCal, int16 ithisCoordSystem, int16 iOverSampleRatio); Executes one pass of the accelerometer, magnetometer and gyroscope sensor fusion algorithm.
void fInit_6DOF_GY_KALMAN(struct SV_6DOF_GY_KALMAN *pthisSV, int16 iSensorFS, int16 iOverSampleRatio) Initialization of accelerometer plus gyroscope sensor fusion algorithm.
void fRun_6DOF_GY_KALMAN(struct SV_6DOF_GY_KALMAN *pthisSV, struct AccelSensor *pthisAccel, struct GyroSensor *pthisGyro, int16 ithisCoordSystem, int16 iOverSampleRatio); Executes one pass of the accelerometer plus gyroscope sensor fusion algorithm.

2 Mathematical Background

2.1 Vector Product

The vector product of the vectors $\boldsymbol{\omega}$ and $\boldsymbol{a}$ is written as $\boldsymbol{\omega} \times \boldsymbol{a}$ and defined to be:

$$\boldsymbol{\omega} \times \boldsymbol{a} = \begin{vmatrix} \hat{\boldsymbol{i}} & \hat{\boldsymbol{j}} & \hat{\boldsymbol{k}} \\ \omega_x & \omega_y & \omega_z \\ a_x & a_y & a_z \end{vmatrix} = \begin{pmatrix} \omega_y a_z - \omega_z a_y \\ \omega_z a_x - \omega_x a_z \\ \omega_x a_y - \omega_y a_x \end{pmatrix} \qquad \text{Eq 2.1.1}$$

By inspection:

$$\boldsymbol{\omega} \times \boldsymbol{a} = -\boldsymbol{a} \times \boldsymbol{\omega} \qquad \text{Eq 2.1.2}$$

Equation 2.1.1 can be written in terms of the equivalent matrix multiplication as:

$$\begin{pmatrix} \omega_y a_z - \omega_z a_y \\ \omega_z a_x - \omega_x a_z \\ \omega_x a_y - \omega_y a_x \end{pmatrix} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = (\boldsymbol{\omega} \times) \boldsymbol{a} \qquad \text{Eq 2.1.3}$$

where the matrix $(\boldsymbol{\omega} \times)$ is defined as:

$$(\boldsymbol{\omega} \times) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \qquad \text{Eq 2.1.4}$$

The terminology in equation 2.1.3 emphasis the equivalence between the matrix $(\boldsymbol{\omega} \times)$ multiplying the vector $\boldsymbol{a}$ and the vector product between the two vectors written as $\boldsymbol{\omega} \times \boldsymbol{a}$:

$$(\boldsymbol{\omega} \times)\boldsymbol{a} = \boldsymbol{\omega} \times \boldsymbol{a} \qquad \text{Eq 2.1.5}$$

A consequence of equations 2.1.2 and 2.1.5 is that:

$$(\boldsymbol{\omega} \times)\boldsymbol{a} = -(\boldsymbol{a} \times)\boldsymbol{\omega} \qquad \text{Eq 2.1.6}$$

2.2 Incremental Rotations

The Rodriguez rotation matrix which transforms vectors as a result of the rotation of the coordinate frame about normalized axis $\hat{\boldsymbol{n}}$ by angle η is:

$$\boldsymbol{R}(\hat{\boldsymbol{n}}\eta) = \begin{pmatrix} \hat{n}_x{}^2 + (1 - \hat{n}_x{}^2)cos\eta & \hat{n}_x \hat{n}_y (1 - cos\eta) + \hat{n}_z sin\eta & \hat{n}_x \hat{n}_z (1 - cos\eta) - \hat{n}_y sin\eta \\ \hat{n}_x \hat{n}_y (1 - cos\eta) - \hat{n}_z sin\eta & \hat{n}_y{}^2 + (1 - \hat{n}_y{}^2)cos\eta & \hat{n}_y \hat{n}_z (1 - cos\eta) + \hat{n}_x sin\eta \\ \hat{n}_x \hat{n}_z (1 - cos\eta) + \hat{n}_y sin\eta & \hat{n}_y \hat{n}_z (1 - cos\eta) - \hat{n}_x sin\eta & \hat{n}_z{}^2 + (1 - \hat{n}_z{}^2)cos\eta \end{pmatrix} \qquad \text{Eq 2.2.1}$$

The rotation matrix which rotates a vector (not the coordinate frame) about the normalized axis $\hat{\boldsymbol{n}}$ by angle η is the transpose of equation 2.2.1.

To first order accuracy for small rotation angles η measured in radians:

$$sin\eta = \eta \qquad \text{Eq 2.2.2}$$

$$cos\eta = 1 \qquad \text{Eq 2.2.3}$$

The first order approximation to equation 2.2.1 for coordinate frame rotation around axis $\hat{\boldsymbol{n}}$ by the small angle $\delta\eta$ radians is the rotation matrix $\delta\boldsymbol{R}$:

$$\delta\boldsymbol{R}(\hat{\boldsymbol{n}}\delta\eta) \approx \begin{pmatrix} 1 & \hat{n}_z \delta\eta & -\hat{n}_y \delta\eta \\ -\hat{n}_z \delta\eta & 1 & \hat{n}_x \delta\eta \\ \hat{n}_y \delta\eta & -\hat{n}_x \delta\eta & 1 \end{pmatrix} for\ \delta\eta\ in\ rad \qquad \text{Eq 2.2.4}$$

If the angular velocity of the coordinate frame is $\boldsymbol{\omega}$, then the angle rotated $\delta\eta$ rotated during time interval δt is:

$$\boldsymbol{\omega}\delta t = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \delta t = \begin{pmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{pmatrix} \delta\eta \qquad \text{Eq 2.2.5}$$

Substituting equation 2.2.5 into equation 2.2.4 gives the incremental rotation matrix $\delta\boldsymbol{R}(\boldsymbol{\omega}\delta t)$ which models rotation of the coordinate frame by angular velocity $\boldsymbol{\omega}$ over time δt:

$$\delta\boldsymbol{R}(\boldsymbol{\omega}\delta t) \approx \begin{pmatrix} 1 & \omega_z\delta t & -\omega_y\delta t \\ -\omega_z\delta t & 1 & \omega_x\delta t \\ \omega_y\delta t & -\omega_x\delta t & 1 \end{pmatrix} for\ \boldsymbol{\omega}\ in\ rad\ per\ sec \qquad \text{Eq 2.2.6}$$

$$\delta\boldsymbol{R}(\boldsymbol{\omega}\delta t) \approx \begin{pmatrix} 1 & \alpha\omega_z\delta t & -\alpha\omega_y\delta t \\ -\alpha\omega_z\delta t & 1 & \alpha\omega_x\delta t \\ \alpha\omega_y\delta t & -\alpha\omega_x\delta t & 1 \end{pmatrix} for\ \boldsymbol{\omega}\ in\ deg\ per\ sec \qquad \text{Eq 2.2.7}$$

where α is the scaling factor from degrees to radians:

$$\alpha = \left(\frac{\pi}{180}\right) \qquad \text{Eq 2.2.8}$$

The orientation matrix $\boldsymbol{R}(t)$ at time t is then related to the matrix $\boldsymbol{R}(t-\delta t)$ at time $t-\delta t$ by:

$$\boldsymbol{R}(t) = \delta\boldsymbol{R}(\boldsymbol{\omega}\delta t)\boldsymbol{R}(t-\delta t) \approx \begin{pmatrix} 1 & \omega_z\delta t & -\omega_y\delta t \\ -\omega_z\delta t & 1 & \omega_x\delta t \\ \omega_y\delta t & -\omega_x\delta t & 1 \end{pmatrix} \boldsymbol{R}(t-\delta t)\ for\ \boldsymbol{\omega}\ in\ rad\ per\ sec \qquad \text{Eq 2.2.9}$$

$$= \{\boldsymbol{I} - (\boldsymbol{\omega}\delta t\ \times)\}\ \boldsymbol{R}(t-\delta t) = \boldsymbol{R}(t-\delta t) - (\boldsymbol{\omega}\delta t\ \times)\boldsymbol{R}(t-\delta t)\ for\ \boldsymbol{\omega}\ in\ rad\ per\ sec$$

$$\boldsymbol{R}(t) = \delta\boldsymbol{R}(\boldsymbol{\omega}\delta t)\boldsymbol{R}(t-\delta t) \approx \begin{pmatrix} 1 & \alpha\omega_z\delta t & -\alpha\omega_y\delta t \\ -\alpha\omega_z\delta t & 1 & \alpha\omega_x\delta t \\ \alpha\omega_y\delta t & -\alpha\omega_x\delta t & 1 \end{pmatrix} \boldsymbol{R}(t-\delta t)\ for\ \boldsymbol{\omega}\ in\ deg\ per\ sec \qquad \text{Eq 2.2.10}$$

$$= \{\boldsymbol{I} - (\alpha\boldsymbol{\omega}\delta t\ \times)\}\boldsymbol{R}(t-\delta t) = \boldsymbol{R}(t-\delta t) - (\alpha\boldsymbol{\omega}\delta t\ \times)\boldsymbol{R}(t-\delta t)\ for\ \boldsymbol{\omega}\ in\ deg\ per\ sec \qquad \text{Eq 2.2.11}$$

2.3 Product of Small Rotation Matrices

Small angle rotation matrices commute to first order. For small angles $\delta\varepsilon$ and $\delta\eta$ (in radians) to first order:

$$\delta\boldsymbol{R}(\widehat{\boldsymbol{m}}\delta\varepsilon)\delta\boldsymbol{R}(\widehat{\boldsymbol{n}}\delta\eta) = \begin{pmatrix} 1 & \widehat{m}_z\delta\varepsilon & -\widehat{m}_y\delta\varepsilon \\ -\widehat{m}_z\delta\varepsilon & 1 & \widehat{m}_x\delta\varepsilon \\ \widehat{m}_y\delta\varepsilon & -\widehat{m}_x\delta\varepsilon & 1 \end{pmatrix} \begin{pmatrix} 1 & \hat{n}_z\delta\eta & -\hat{n}_y\delta\eta \\ -\hat{n}_z\delta\eta & 1 & \hat{n}_x\delta\eta \\ \hat{n}_y\delta\eta & -\hat{n}_x\delta\eta & 1 \end{pmatrix} \qquad \text{Eq 2.3.1}$$

$$= \begin{pmatrix} 1 & (\widehat{m}_z\delta\varepsilon + \hat{n}_z\delta\eta) & -(\widehat{m}_y\delta\varepsilon + \hat{n}_y\delta\eta) \\ -(\widehat{m}_z\delta\varepsilon + \hat{n}_z\delta\eta) & 1 & (\widehat{m}_x\delta\varepsilon + \hat{n}_x\delta\eta) \\ (\widehat{m}_y\delta\varepsilon + \hat{n}_y\delta\eta) & -(\widehat{m}_x\delta\varepsilon + \hat{n}_x\delta\eta) & 1 \end{pmatrix} = \delta\boldsymbol{R}(\widehat{\boldsymbol{n}}\delta\eta)\delta\boldsymbol{R}(\widehat{\boldsymbol{m}}\delta\varepsilon) \qquad \text{Eq 2.3.2}$$

The product of a small angle rotation matrix and a large angle rotation matrix or the product of two large angle rotation matrixes do not, in contrast, commute.

3 Accelerometer, Magnetometer, Gyroscope Kalman Filter

3.1 Model of the Error Process

The indirect Kalman filter used for sensor fusion models the error process $\boldsymbol{x}_{\varepsilon,k}$ with the recursive update:

$$\boldsymbol{x}_{\varepsilon,k} = \boldsymbol{A}_k \boldsymbol{x}_{\varepsilon,k-1} + \boldsymbol{w}_k \qquad \text{Eq 3.1.1}$$

$$\Rightarrow \boldsymbol{x}_{\varepsilon,k} = \begin{pmatrix} \boldsymbol{\theta}_{\varepsilon,k} \\ \boldsymbol{b}_{\varepsilon,k} \\ {}^S\boldsymbol{a}_{\varepsilon,k} \\ {}^S\boldsymbol{d}_{\varepsilon,k} \end{pmatrix} = \boldsymbol{A}_k \begin{pmatrix} \boldsymbol{\theta}_{\varepsilon,k-1} \\ \boldsymbol{b}_{\varepsilon,k-1} \\ {}^S\boldsymbol{a}_{\varepsilon,k-1} \\ {}^S\boldsymbol{d}_{\varepsilon,k-1} \end{pmatrix} + \boldsymbol{w}_k \qquad \text{Eq 3.1.2}$$

where $\boldsymbol{x}_{\varepsilon,k}$ is defined as the 12x1 vector comprising:

- the 3x1 orientation error vector $\boldsymbol{\theta}_{\varepsilon,k}$ (with units deg) at time k
- the 3x1 gyroscope zero rate offset vector $\boldsymbol{b}_{\varepsilon,k}$ (with units deg/s) at time k
- the 3x1 acceleration error vector ${}^S\boldsymbol{a}_{\varepsilon,k}$ measured in the sensor frame (with units g) at time k
- the 3x1 magnetic disturbance error vector ${}^S\boldsymbol{d}_{\varepsilon,k}$ measured in the sensor frame (with units uT) at time k

and where $\boldsymbol{w}_k$ is a 12x1 additive noise vector.

The orientation error vector $\boldsymbol{\theta}_{\varepsilon,k}$ defines the rotation vector by which the estimated orientation exceeds the true orientation.

The gyroscope zero rate offset vector $\boldsymbol{b}_{\varepsilon,k}$ models the output of the gyroscope sensor when the sensor is not rotating. This is typically less than 5 deg/s or less than 1% of the gyroscope sensor's full range. It is a slowly varying additive offset which must be subtracted from every gyroscope measurement.

The acceleration error vector ${}^S\boldsymbol{a}_{\varepsilon,k}$ is the error in the estimation of the linear acceleration of the sensor board measured in the sensor frame of reference. Accelerometer sensors are sensitive to both gravity and linear acceleration and the acceleration component affects the accelerometer's estimate of the gravity vector.

Hard and soft iron magnetic interference are defined in the coordinate frame of the sensor board and rotate with the sensor board. The magnetic disturbance term models magnetic interference external to the sensor board (such as a magnet brought close to the sensor board or a steel filing cabinet) which cannot be modeled as hard or soft iron interference which is approximately constant in the sensor coordinate frame. The magnetic disturbance error vector ${}^S\boldsymbol{d}_{\varepsilon,k}$ is the error in the estimate of the external magnetic disturbance in the sensor frame of reference.

3.2 Model of the Measurement Error Process

The measurement process ${}^S\boldsymbol{z}_{\varepsilon,k}$ is defined as the 6x1 vector of differences (in the sensor coordinate frame) between i) the accelerometer and gyroscope estimates of the gravity vector and ii) the magnetometer and gyroscope estimates of the geomagnetic vector.

$${}^S\boldsymbol{z}_{\varepsilon,k} = \begin{pmatrix} {}^S\boldsymbol{g}_{A,k} - {}^S\boldsymbol{g}_{G,k} \\ {}^S\boldsymbol{m}_{M,k} - {}^S\boldsymbol{m}_{G,k} \end{pmatrix} \qquad \text{Eq 3.2.1}$$

${}^S\boldsymbol{g}_{A,k}$ and ${}^S\boldsymbol{g}_{G,k}$ are the gravity vectors computed in the sensor frame from the accelerometer and gyro sensors respectively. ${}^S\boldsymbol{m}_{M,k}$ and ${}^S\boldsymbol{m}_{G,k}$ are the geomagnetic vectors computed in the sensor frame from the magnetometer and gyro sensors respectively.

The standard Kalman filter formalism linearly relates the measurement process to the underlying process via the Kalman measurement matrix $\boldsymbol{C}_k$ and 6x1 additive noise vector $\boldsymbol{v}_k$:

$$ {}^{S}\boldsymbol{z}_{\varepsilon,k} = \begin{pmatrix} {}^{S}\boldsymbol{g}_{A,k} - {}^{S}\boldsymbol{g}_{G,k} \\ {}^{S}\boldsymbol{m}_{M,k} - {}^{S}\boldsymbol{m}_{G,k} \end{pmatrix} = \boldsymbol{C}_k {}^{S}\boldsymbol{x}_{\varepsilon,k} + \boldsymbol{v}_k = \boldsymbol{C}_k \begin{pmatrix} \boldsymbol{\theta}_{\varepsilon,k} \\ \boldsymbol{b}_{\varepsilon,k} \\ {}^{S}\boldsymbol{a}_{\varepsilon,k} \\ {}^{S}\boldsymbol{d}_{\varepsilon,k} \end{pmatrix} + \boldsymbol{v}_k \quad \text{Eq 3.2.2} $$

The covariance matrices of the noise processes are defined in the usual manner as:

$$ \boldsymbol{Q}_{w,k} = E[\boldsymbol{w}_k \boldsymbol{w}_k{}^T] \quad \text{Eq 3.2.3} $$

$$ \boldsymbol{Q}_{v,k} = E[\boldsymbol{v}_k \boldsymbol{v}_k{}^T] \quad \text{Eq 3.2.4} $$

where $\boldsymbol{Q}_{w,k}$ is a symmetric 12x12 matrix and $\boldsymbol{Q}_{v,k}$ is a symmetric 6x6 matrix.

3.3 Kalman Filter Equations

The measurement error ${}^{S}\boldsymbol{z}_{\varepsilon,k}$ and underlying error process $\boldsymbol{x}_{\varepsilon,k}$ have the same form as in the standard Kalman filter despite being defined in terms of the error process. The standard Kalman filter equations can therefore be reused with minor modifications.

<u>Kalman equation 1</u>

Kalman equation B (see below) corrects the components of the vector $\widehat{\boldsymbol{x}}_k^+$ by applying the *a posteriori* error vector $\widehat{\boldsymbol{x}}_{\varepsilon,k}^+$. The *a priori* error estimate $\widehat{\boldsymbol{x}}_{\varepsilon,k}^-$ is therefore zero and the linear prediction matrix $\boldsymbol{A}_k$ is zero:

$$ \widehat{\boldsymbol{x}}_{\varepsilon,k}^- = \begin{pmatrix} \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^- \\ \widehat{\boldsymbol{b}}_{\varepsilon,k}^- \\ {}^{S}\widehat{\boldsymbol{a}}_{\varepsilon,k}^- \\ {}^{S}\widehat{\boldsymbol{d}}_{\varepsilon,k}^- \end{pmatrix} = \boldsymbol{0} \quad \text{Eq 3.3.1} $$

$$ \Rightarrow \boldsymbol{A}_k = \boldsymbol{0} \quad \text{Eq 3.3.2} $$

<u>Kalman equation 2</u>

The *a priori* (linear prediction) 12x12 error covariance matrix $\boldsymbol{P}_k^-$ is updated using the linear prediction matrix $\boldsymbol{A}_k$ and the noise matrix $\boldsymbol{Q}_{w,k}$.

$$ \boldsymbol{P}_k^- = \boldsymbol{A}_k \boldsymbol{P}_{k-1}^+ \boldsymbol{A}_k{}^T + \boldsymbol{Q}_{w,k} \quad \text{Eq 3.3.3} $$

Substituting equation 3.3.2 into 3.3.3 gives:

$$ \boldsymbol{P}_k^- = \boldsymbol{Q}_{w,k} \quad \text{Eq 3.3.4} $$

<u>Kalman equation A</u>

The measurement matrix $\boldsymbol{C}_k$ is computed each iteration as a function of the *a priori* estimates of the gravitational and geomagnetic vectors in the sensor frame.

$$ \boldsymbol{C}_k = function\left({}^{S}\widehat{\boldsymbol{g}}_{G,k}^-, {}^{S}\widehat{\boldsymbol{m}}_{G,k}^-\right) \quad \text{Eq 3.3.5} $$

<u>Kalman equation 3</u>

The Kalman filter gain matrix $\boldsymbol{K}_k$ is updated:

$$ \boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{C}_k{}^T \left(\boldsymbol{C}_k \boldsymbol{P}_k^- \boldsymbol{C}_k{}^T + \boldsymbol{Q}_{v,k}\right)^{-1} \quad \text{Eq 3.3.6} $$

Substituting equation 3.3.4 into 3.3.6 gives:

$$\boldsymbol{K}_k = \boldsymbol{Q}_{w,k}\boldsymbol{C}_k{}^T\left(\boldsymbol{C}_k\boldsymbol{Q}_{w,k}\boldsymbol{C}_k{}^T + \boldsymbol{Q}_{v,k}\right)^{-1} \qquad \text{Eq 3.3.7}$$

Kalman equation 4

The Kalman filter (*a posteriori*) estimate $\widehat{\boldsymbol{x}}^{+}_{\varepsilon,k}$ is computed from the *a priori* error vector estimate $\widehat{\boldsymbol{x}}^{-}_{\varepsilon,k}$ and from the measurement error vector ${}^{S}\boldsymbol{z}_{\varepsilon,k}$:

$$\widehat{\boldsymbol{x}}^{+}_{\varepsilon,k} = \widehat{\boldsymbol{x}}^{-}_{\varepsilon,k} + \boldsymbol{K}_k\left({}^{S}\boldsymbol{z}_{\varepsilon,k} - \boldsymbol{C}_k\widehat{\boldsymbol{x}}^{-}_{\varepsilon,k}\right) \qquad \text{Eq 3.3.8}$$

Substituting equation 3.3.1 into 3.3.8 gives:

$$\widehat{\boldsymbol{x}}^{+}_{\varepsilon,k} = \begin{pmatrix} \widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k} \\ \widehat{\boldsymbol{b}}^{+}_{\varepsilon,k} \\ {}^{S}\widehat{\boldsymbol{a}}^{+}_{\varepsilon,k} \\ {}^{S}\widehat{\boldsymbol{d}}^{+}_{\varepsilon,k} \end{pmatrix} = \boldsymbol{K}_k\,{}^{S}\boldsymbol{z}_{\varepsilon,k} = \boldsymbol{K}_k\begin{pmatrix} {}^{S}\widehat{\boldsymbol{g}}^{-}_{A,k} - {}^{S}\widehat{\boldsymbol{g}}^{-}_{G,k} \\ {}^{S}\widehat{\boldsymbol{m}}^{-}_{M,k} - {}^{S}\widehat{\boldsymbol{m}}^{-}_{G,k} \end{pmatrix} \qquad \text{Eq 3.3.9}$$

Kalman equation 5

The *a posteriori* Kalman error covariance matrix $\boldsymbol{P}^{+}_{k}$ is updated for the next iteration:

$$\boldsymbol{P}^{+}_{k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k)\boldsymbol{P}^{-}_{k} \qquad \text{Eq 3.3.10}$$

Substituting equation 3.3.4 into 3.3.10 gives:

$$\boldsymbol{P}^{+}_{k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k)\boldsymbol{Q}_{w,k} \qquad \text{Eq 3.3.11}$$

Kalman equation B

The components of the *a posteriori* vector $\widehat{\boldsymbol{x}}^{+}_{k}$ are corrected by the *a posteriori* error vector $\widehat{\boldsymbol{x}}^{+}_{\varepsilon,k}$.

The *a posteriori* orientation matrix is computed by multiplying the *a priori* orientation matrix by a rotation matrix constructed from the negative of the *a posteriori* rotation vector corrections:

$$\widehat{\boldsymbol{R}}^{+}_{k} = \Delta\boldsymbol{R}\left(-\widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k}\right)\widehat{\boldsymbol{R}}^{-}_{k} \qquad \text{Eq 3.3.12}$$

The *a posteriori* estimate of the remaining terms can be computed by simple vector subtraction:

$$\begin{pmatrix} \widehat{\boldsymbol{b}}^{+}_{k} \\ {}^{S}\widehat{\boldsymbol{a}}^{+}_{k} \\ {}^{S}\widehat{\boldsymbol{d}}^{+}_{k} \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{b}}^{-}_{k} \\ {}^{S}\widehat{\boldsymbol{a}}^{-}_{k} \\ {}^{S}\widehat{\boldsymbol{d}}^{-}_{k} \end{pmatrix} - \begin{pmatrix} \widehat{\boldsymbol{b}}^{+}_{\varepsilon,k} \\ {}^{S}\widehat{\boldsymbol{a}}^{+}_{\varepsilon,k} \\ {}^{S}\widehat{\boldsymbol{d}}^{+}_{\varepsilon,k} \end{pmatrix} \qquad \text{Eq 3.3.13}$$

A consequence of Kalman equation B is that the *a priori* estimate $\widehat{\boldsymbol{x}}^{-}_{\varepsilon,k+1}$ of the error vector for the next iteration $k+1$ is pre-corrected to zero (equation 3.3.1) and the prediction matrix $\boldsymbol{A}_k$ is zero.

Kalman equation C

The noise covariance matrix $\boldsymbol{Q}_{w,k+1}$ is updated for the next iteration as a function of the *a posteriori* covariance matrix $\boldsymbol{P}^{+}_{k}$.

$$\boldsymbol{Q}_{w,k+1} = \boldsymbol{Q}(\boldsymbol{P}^{+}_{k}) \qquad \text{Eq 3.3.14}$$

4 Sensor Models

4.1 Gyroscope Sensor Model

The gyroscope sensor measurement $\boldsymbol{y}_{G,k}$ (deg/s) is modeled as:

$$\boldsymbol{y}_{G,k} = \boldsymbol{\omega}_k + \boldsymbol{b}_k + \boldsymbol{v}_{G,k} \qquad \text{Eq 4.1.1}$$

$\boldsymbol{\omega}_k$ is the true angular velocity in deg/s, $\boldsymbol{b}_k$ is the zero rate gyro offset (deg/s) and $\boldsymbol{v}_{G,k}$ is the additive gyro noise (deg/s). $\boldsymbol{y}_{G,k}$ is obviously measured in the sensor frame of reference.

The gyro offset $\boldsymbol{b}_k$ (deg/s) is modeled as a first order random walk:

$$\boldsymbol{b}_k = \boldsymbol{b}_{k-1} + \boldsymbol{w}_{b,k} \qquad \text{Eq 4.1.2}$$

where $\boldsymbol{w}_{b,k}$ is a zero mean white Gaussian noise process (with units deg/s).

The *a priori* estimate of the gyro offset is simply the *a posteriori* estimate from the previous iteration since $\boldsymbol{w}_{b,k}$ is zero mean and white:

$$\hat{\boldsymbol{b}}_k^- = \hat{\boldsymbol{b}}_{k-1}^+ \qquad \text{Eq 4.1.3}$$

Simple algebra gives $\hat{\boldsymbol{b}}_{\varepsilon,k}^-$ as a function of $\hat{\boldsymbol{b}}_{\varepsilon,k-1}^+$:

$$\hat{\boldsymbol{b}}_{\varepsilon,k}^- = \hat{\boldsymbol{b}}_k^- - \boldsymbol{b}_k = \hat{\boldsymbol{b}}_{k-1}^+ - \boldsymbol{b}_k = \hat{\boldsymbol{b}}_{k-1}^+ - \left(\boldsymbol{b}_{k-1} + \boldsymbol{w}_{b,k}\right) = \left(\hat{\boldsymbol{b}}_{k-1}^+ - \boldsymbol{b}_{k-1}\right) - \boldsymbol{w}_{b,k} \qquad \text{Eq 4.1.4}$$

$$\Rightarrow \hat{\boldsymbol{b}}_{\varepsilon,k}^- = \hat{\boldsymbol{b}}_{\varepsilon,k-1}^+ - \boldsymbol{w}_{b,k} \qquad \text{Eq 4.1.5}$$

The *a priori estimate* $\hat{\boldsymbol{\omega}}_k^-$ of the true angular velocity $\boldsymbol{\omega}_k$ is determined from the gyro reading $\boldsymbol{y}_{G,k}$ by:

$$\hat{\boldsymbol{\omega}}_k^- = \boldsymbol{y}_{G,k} - \hat{\boldsymbol{b}}_k^- = \boldsymbol{y}_{G,k} - \hat{\boldsymbol{b}}_{k-1}^+ \qquad \text{Eq 4.1.6}$$

Substituting for $\boldsymbol{y}_{G,k}$ in equation 4.1.6 relates the *a priori* error in the angular velocity as a function of the *a posteriori* error in the gyroscope offset:

$$\hat{\boldsymbol{\omega}}_k^- = \boldsymbol{\omega}_k + \boldsymbol{b}_k + \boldsymbol{v}_{G,k} - \hat{\boldsymbol{b}}_{k-1}^+ = \boldsymbol{\omega}_k + \boldsymbol{b}_{k-1} + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k} - \hat{\boldsymbol{b}}_{k-1}^+ \qquad \text{Eq 4.1.7}$$

$$\Rightarrow \hat{\boldsymbol{\omega}}_k^- = \boldsymbol{\omega}_k - \hat{\boldsymbol{b}}_{\varepsilon,k-1}^+ + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k} \qquad \text{Eq 4.1.8}$$

$$\Rightarrow \hat{\boldsymbol{\omega}}_{\varepsilon,k}^- = \hat{\boldsymbol{\omega}}_k^- - \boldsymbol{\omega}_k = -\hat{\boldsymbol{b}}_{\varepsilon,k-1}^+ + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k} \qquad \text{Eq 4.1.9}$$

4.2 Accelerometer Sensor Model

The Aerospace and Windows 8 coordinate systems are 'gravity positive' standards whereas Android is an 'acceleration positive' standard. The accelerometer sensor models are then:

$$\boldsymbol{y}_{A,k} = -{}^S\boldsymbol{a}_k + {}^S\boldsymbol{g}_k - \boldsymbol{v}_{A,k} \qquad \text{Aerospace, Windows 8} \qquad \text{Eq 4.2.1}$$

$$\boldsymbol{y}_{A,k} = {}^S\boldsymbol{a}_k - {}^S\boldsymbol{g}_k + \boldsymbol{v}_{A,k} \qquad \text{Android} \qquad \text{Eq 4.2.2}$$

The linear acceleration is modeled as a low pass filtered white noise process in the global frame.

$${}^G\boldsymbol{a}_k = c_a\,{}^G\boldsymbol{a}_{k-1} + \boldsymbol{w}_{a,k} \qquad \text{Eq 4.2.3}$$

where $\boldsymbol{w}_{a,k}$ is a zero mean white Gaussian noise process with units of g.

The *a priori* estimate of the linear acceleration (in the global frame) is the *a posteriori* estimate (in the global frame) for the previous iteration decayed by the low pass filter constant c_a:

$${}^G\hat{\boldsymbol{a}}_k^- = c_a\,{}^G\hat{\boldsymbol{a}}_{k-1}^+ \qquad \text{Eq 4.2.4}$$

Simple algebra gives ${}^G\hat{\boldsymbol{a}}_{\varepsilon,k}^-$ as a function of ${}^G\hat{\boldsymbol{a}}_{\varepsilon,k-1}^+$:

$${}^G\hat{\boldsymbol{a}}_{\varepsilon,k}^- = {}^G\hat{\boldsymbol{a}}_k^- - {}^G\boldsymbol{a}_k = c_a\,{}^G\hat{\boldsymbol{a}}_{k-1}^+ - {}^G\boldsymbol{a}_k = c_a\,{}^G\hat{\boldsymbol{a}}_{k-1}^+ - \left(c_a\,{}^G\boldsymbol{a}_{k-1} + \boldsymbol{w}_{a,k}\right) \qquad \text{Eq 4.2.5}$$

$$ {}^{G}\hat{\boldsymbol{a}}_{\varepsilon,k}^{-} = c_a\, {}^{G}\hat{\boldsymbol{a}}_{\varepsilon,k-1}^{+} - \boldsymbol{w}_{a,k} \quad \text{Eq 4.2.6} $$

The a priori estimate of the gravity vector in the sensor frame using the accelerometer measurement is:

$$ {}^{S}\hat{\boldsymbol{g}}_{A,k}^{-} = \boldsymbol{y}_{A,k} + {}^{S}\hat{\boldsymbol{a}}_{k}^{-} = {}^{S}\boldsymbol{g}_{k} - {}^{S}\boldsymbol{a}_{k} - \boldsymbol{v}_{A,k} + {}^{S}\hat{\boldsymbol{a}}_{k}^{-} \quad \text{Aerospace, Windows 8} \quad \text{Eq 4.2.7} $$

$$ {}^{S}\hat{\boldsymbol{g}}_{A,k}^{-} = -\boldsymbol{y}_{A,k} + {}^{S}\hat{\boldsymbol{a}}_{k}^{-} = {}^{S}\boldsymbol{g}_{k} - {}^{S}\boldsymbol{a}_{k} - \boldsymbol{v}_{A,k} + {}^{S}\hat{\boldsymbol{a}}_{k}^{-} \quad \text{Android} \quad \text{Eq 4.2.8} $$

Conveniently the result is the same for all operating system standards, giving:

$$ \Rightarrow {}^{S}\hat{\boldsymbol{g}}_{A,k}^{-} = {}^{S}\boldsymbol{g}_{k} + {}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^{-} - \boldsymbol{v}_{A,k} \quad \text{Aerospace, Windows 8, Android} \quad \text{Eq 4.2.9} $$

4.3 Magnetometer Sensor Model

The magnetometer reading model is in the sensor frame:

$$ \boldsymbol{y}_{M,k} = {}^{S}\boldsymbol{m}_{k} + {}^{S}\boldsymbol{d}_{k} + \boldsymbol{v}_{M,k} \quad \text{Eq 4.3.1} $$

The magnetic disturbance model in the global frame is another low pass filtered white noise process.

$$ {}^{G}\boldsymbol{d}_{k} = c_d\, {}^{G}\boldsymbol{d}_{k-1} + \boldsymbol{w}_{d,k} \quad \text{Eq 4.3.2} $$

$\boldsymbol{w}_{d,k}$ is a zero mean white Gaussian noise process with units uT.

The *a priori* estimate of the magnetic disturbance (in the global frame) is therefore the *a posteriori* estimate (in the global frame) for the previous iteration decayed by the low pass filter constant c_d:

$$ {}^{G}\hat{\boldsymbol{d}}_{k}^{-} = c_d\, {}^{G}\hat{\boldsymbol{d}}_{k-1}^{+} \quad \text{Eq 4.3.3} $$

Simple algebra again gives ${}^{G}\hat{\boldsymbol{d}}_{\varepsilon,k}^{-}$ as a function of ${}^{G}\hat{\boldsymbol{d}}_{\varepsilon,k-1}^{+}$:

$$ {}^{G}\hat{\boldsymbol{d}}_{\varepsilon,k}^{-} = {}^{G}\hat{\boldsymbol{d}}_{k}^{-} - {}^{G}\boldsymbol{d}_{k} = c_d\, {}^{G}\hat{\boldsymbol{d}}_{k-1}^{+} - {}^{G}\boldsymbol{d}_{k} = c_d\, {}^{G}\hat{\boldsymbol{d}}_{k-1}^{+} - \left(c_d\, {}^{G}\boldsymbol{d}_{k-1} + \boldsymbol{w}_{d,k}\right) = c_d\, {}^{G}\hat{\boldsymbol{d}}_{\varepsilon,k-1}^{+} - \boldsymbol{w}_{d,k} \quad \text{Eq 4.3.4} $$

A useful expression can be derived for the *a priori* geomagnetic vector (units of uT):

$$ {}^{S}\hat{\boldsymbol{m}}_{M,k}^{-} = \boldsymbol{y}_{M,k} - {}^{S}\hat{\boldsymbol{d}}_{k}^{-} = {}^{S}\boldsymbol{m}_{k} + {}^{S}\boldsymbol{d}_{k} - {}^{S}\hat{\boldsymbol{d}}_{k}^{-} + \boldsymbol{v}_{M,k} \quad \text{Eq 4.3.5} $$

$$ \Rightarrow {}^{S}\hat{\boldsymbol{m}}_{M,k}^{-} = {}^{S}\boldsymbol{m}_{k} - {}^{S}\hat{\boldsymbol{d}}_{\varepsilon,k}^{-} + \boldsymbol{v}_{M,k} \quad \text{Eq 4.3.6} $$

5 Orientation Estimates

5.1 A Priori Orientation Estimate

The *a priori* estimate of the orientation matrix $\widehat{\boldsymbol{R}}_k^-$ is computed as a linear prediction from the *a posteriori* orientation matrix $\widehat{\boldsymbol{R}}_{k-1}^+$ from the previous iteration rotated by the *a priori* angular velocity rotation vector:

$$\widehat{\boldsymbol{R}}_k^- = \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^- \delta t)\widehat{\boldsymbol{R}}_{k-1}^+ \quad \text{Eq 5.1.1}$$

The *a priori* angular velocity estimate $\widehat{\boldsymbol{\omega}}_k^-$ is given in equation 4.1.6 as the gyroscope output $\boldsymbol{y}_{G,k}$ corrected for the *a priori* estimate of the gyroscope offset $\widehat{\boldsymbol{b}}_k^-$:

$$\widehat{\boldsymbol{\omega}}_k^- = \boldsymbol{y}_{G,k} - \widehat{\boldsymbol{b}}_k^- = \boldsymbol{y}_{G,k} - \widehat{\boldsymbol{b}}_{k-1}^+ \quad \text{Eq 5.1.2}$$

In practice equation 5.1.1 is more efficiently computed using the incremental quaternion $\Delta q(\widehat{\boldsymbol{\omega}}_k^- \delta t)$ rather than the incremental rotation matrix:

$$\hat{q}_k^- = \hat{q}_{k-1}^+ \Delta q(\widehat{\boldsymbol{\omega}}_k^- \delta t) \quad \text{Eq 5.1.3}$$

The sensor fusion software uses both rotation matrix and quaternion algebras and selects the most efficient representation for each operation. Note that successive product rotation matrices accumulate to the left and successive product quaternions accumulate to the right.

The sensor fusion software samples the gyroscope sensor, by default, at 200Hz and executes the computationally expensive Kalman filter at 25Hz. At the start of each pass of the 25Hz Kalman filter a total of eight buffered gyroscope measurements from the previous 40ms are integrated as:

$$\Delta q(\widehat{\boldsymbol{\omega}}_k^- \delta t) = \Delta q(\widehat{\boldsymbol{\omega}}_{k,0}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,1}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,2}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,3}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,4}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,5}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,6}^- \delta t)\Delta q(\widehat{\boldsymbol{\omega}}_{k,7}^- \delta t) \quad \text{Eq 5.1.4}$$

where, with obvious notation, $\widehat{\boldsymbol{\omega}}_{k,j}^-$ is the jth buffered gyroscope measurement at the start of Kalman filter iteration k.

5.2 A Posteriori Orientation Estimate

The *a posteriori* orientation estimate $\widehat{\boldsymbol{R}}_k^+$ is estimated by applying the *a posteriori* orientation error vector $\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+$ to the *a priori* orientation estimate computed as described in section 5.1.

$$\widehat{\boldsymbol{R}}_k^+ = \Delta\boldsymbol{R}(-\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+)\widehat{\boldsymbol{R}}_k^- \quad \text{Eq 5.2.1}$$

It is again more efficient to compute equation 5.2.1 using quaternion algebra where $\Delta q(-\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+)$ is the quaternion correction computed from the *a posteriori error* vector $\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+$:

$$\hat{q}_k^+ = \hat{q}_k^- \Delta q(-\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^+) \quad \text{Eq 5.2.2}$$

6 Acceleration Estimates

6.1 A Priori Acceleration Estimate

The Kalman filter requires the *a priori* linear acceleration estimate in the sensor frame rather than in the global frame. This can easily be computed by rotating the *a priori* estimate from the global to the sensor frame using the *a priori* orientation matrix:

$$ {}^{S}\hat{\boldsymbol{a}}_k^- = \hat{\boldsymbol{R}}_k^- \, {}^{G}\hat{\boldsymbol{a}}_k^- \qquad \text{Eq 6.1.1}$$

Substituting equation 4.2.4 for ${}^{G}\hat{\boldsymbol{a}}_k^-$ and equation 5.1.1 for $\hat{\boldsymbol{R}}_k^-$ gives:

$$ {}^{S}\hat{\boldsymbol{a}}_k^- = c_a \Delta\boldsymbol{R}(\hat{\boldsymbol{\omega}}_k^- \delta t)\hat{\boldsymbol{R}}_{k-1}^+ \, {}^{G}\hat{\boldsymbol{a}}_{k-1}^+ \qquad \text{Eq 6.1.2}$$

$$ = c_a \Delta\boldsymbol{R}(\hat{\boldsymbol{\omega}}_k^- \delta t)\, {}^{S}\hat{\boldsymbol{a}}_{k-1}^+ \approx c_a\{\boldsymbol{I} - (\hat{\boldsymbol{\omega}}_k^- \delta t \times)\}\, {}^{S}\hat{\boldsymbol{a}}_{k-1}^+ \qquad \text{Eq 6.1.3}$$

Equation 6.1.3 simply states that the a priori linear acceleration estimate in the sensor frame for Kalman filter iteration k is the decayed and incrementally rotated estimate from the previous iteration.

In practice the incremental rotation is ignored giving:

$$ {}^{S}\hat{\boldsymbol{a}}_k^- = c_a \, {}^{S}\hat{\boldsymbol{a}}_{k-1}^+ \qquad \text{Eq 6.1.4}$$

6.2 A Posteriori Acceleration Estimate

The a posteriori error vector components $\boldsymbol{x}_{\varepsilon,k}$ are defined in the sensor frame (see equation 3.1.2) making it very convenient to compute the *a posteriori* acceleration estimate in the sensor frame ${}^{S}\hat{\boldsymbol{a}}_k^+$ as:

$$ {}^{S}\hat{\boldsymbol{a}}_k^+ = {}^{S}\hat{\boldsymbol{a}}_k^- - {}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^+ \qquad \text{Eq 6.2.1}$$

This is a component of the Kalman filter equation B in equation 3.3.13.

7 Measurement Error and Matrix

7.1 Introduction

The measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ is defined as the vector difference between the a priori estimates of the gravity and geomagnetic vectors determined from i) the accelerometer and gyroscope sensors and ii) the magnetometer and gyroscope sensors. It is related to the underlying error process $\boldsymbol{x}_{\varepsilon,k}$ through the measurement matrix $\boldsymbol{C}_k$ as:

$$\boldsymbol{z}_{\varepsilon,k} = \begin{pmatrix} {}^S\hat{\boldsymbol{g}}_{A,k}^- - {}^S\hat{\boldsymbol{g}}_{G,k}^- \\ {}^S\hat{\boldsymbol{m}}_{M,k}^- - {}^S\hat{\boldsymbol{m}}_{G,k}^- \end{pmatrix} = \boldsymbol{C}_k \hat{\boldsymbol{x}}_{\varepsilon,k}^- + \boldsymbol{v}_k = \boldsymbol{C}_k \begin{pmatrix} \hat{\boldsymbol{\theta}}_{\varepsilon,k}^- \\ \hat{\boldsymbol{b}}_{\varepsilon,k}^- \\ {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- \\ {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- \end{pmatrix} + \boldsymbol{v}_k \qquad \text{Eq 7.1.1}$$

Section 7.2 documents the straightforward problem of computing the measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ from the sensor measurements. Sections 7.3 and 7.4 derive the mathematical form of the time varying measurement matrix $\boldsymbol{C}_k$.

7.2 Calculation of the Measurement Error Vector

The four components of the *a priori* measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ are easily computed from the sensor measurements and the Kalman filter state vector. The measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ is then used in Kalman equation 4 to computed the a posteriori error state vector.

<u>Gravity Vector Estimate From Accelerometer</u>

The *a priori* estimate of the gravity vector ${}^S\hat{\boldsymbol{g}}_{A,k}^-$ in the sensor frame computed from the accelerometer measurement ${}^S\boldsymbol{y}_{A,k}$ is given by equations 4.2.1 and 4.2.2 as:

$${}^S\hat{\boldsymbol{g}}_{A,k}^- = {}^S\boldsymbol{y}_{A,k} + {}^S\hat{\boldsymbol{a}}_k^- \qquad \text{Aerospace, Windows 8} \qquad \text{Eq 7.2.1}$$

$${}^S\hat{\boldsymbol{g}}_{A,k}^- = -{}^S\boldsymbol{y}_{A,k} + {}^S\hat{\boldsymbol{a}}_k^- \qquad \text{Android} \qquad \text{Eq 7.2.2}$$

<u>Geomagnetic Vector Estimate from Magnetometer</u>

The *a priori* estimate of the geomagnetic vector ${}^S\hat{\boldsymbol{m}}_{M,k}^-$ in the sensor frame computed from the magnetometer measurement ${}^S\boldsymbol{y}_{M,k}$ is given by equation 4.3.1 as:

$${}^S\hat{\boldsymbol{m}}_{M,k}^- = {}^S\boldsymbol{y}_{M,k} - {}^S\hat{\boldsymbol{d}}_k^- \qquad \text{Eq 7.2.3}$$

<u>Gravity Vector Estimate from Gyroscope</u>

The *a priori* estimate of the gravity vector in the sensor frame computed from the gyroscope sensor ${}^S\hat{\boldsymbol{g}}_{G,k}^-$ is the fixed downwards pointing gravity vector in the global frame rotated by the *a priori* orientation matrix into the sensor frame:

$${}^S\hat{\boldsymbol{g}}_{G,k}^- = \hat{\boldsymbol{R}}_k^- \, {}^G\boldsymbol{g}_k \qquad \text{Eq 7.2.4}$$

The *a priori* orientation matrix is the a posteriori orientation matrix rotated by the incremental orientation matrix computed from the *a priori* angular velocity estimate from the gyroscope sensor and *a priori* estimate of the zero rate gyroscope offset:

$$\hat{\boldsymbol{R}}_k^- = \Delta\boldsymbol{R}(\hat{\boldsymbol{\omega}}_k^- \delta t)\hat{\boldsymbol{R}}_{k-1}^+ \qquad \text{Eq 7.2.5}$$

Equation 7.2.4 is easy to compute since ${}^G\boldsymbol{g}_k$ is defined in the global frame and has the fixed value of 1g directed downwards. The multiplication in equation 7.2.4 simply gives the right hand column of the a priori orientation matrix $\hat{\boldsymbol{R}}_k^-$ scaled by +1 or -1 depending on the coordinate system in use.

Geomagnetic Vector Estimate from Gyroscope

The *a priori* estimate of the geomagnetic vector in the sensor frame computed from the gyroscope sensor ${}^S\hat{\boldsymbol{m}}_{G,k}^-$ is the fixed northwards (and downwards) pointing geomagnetic vector in the global frame rotated by the *a priori* orientation matrix:

$$ {}^S\hat{\boldsymbol{m}}_{G,k}^- = \hat{\boldsymbol{R}}_k^- {}^G\boldsymbol{m}_k \qquad \text{Eq 7.2.6} $$

The magnitude B of the geomagnetic vector is determined by the hard and soft iron calibration algorithms. If the geomagnetic inclination angle δ is also known then ${}^G\boldsymbol{m}_k$ is known. The inclination angle delta varies over the surface of the earth and also has significant variation inside steel-framed buildings. It must therefore be determined within the sensor fusion algorithms and this is documented in section 9.

7.3 Basic Model of the Measurement Matrix

Kalman equations 3 and 5 require knowledge of the measurement matrix $\boldsymbol{C}_k$ which theoretically relates the measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ to the process error vector $\boldsymbol{x}_{\varepsilon,k}$.

Equation 4.2.9 simply states that the a priori estimate of the gravity vector in the sensor frame differs from the true gravity vector as a result of error in the estimate of linear acceleration and from accelerometer noise.

$$ {}^S\hat{\boldsymbol{g}}_{A,k}^- - {}^S\boldsymbol{g}_k = {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} \qquad \text{Aerospace, Windows 8, Android} \qquad \text{Eq 7.3.1} $$

By definition, the *a priori* gyroscope estimate of the gravity vector in the sensor frame is in error by the *a priori* orientation error vector:

$$ {}^S\hat{\boldsymbol{g}}_{G,k}^- = \Delta\boldsymbol{R}\left(\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right){}^S\boldsymbol{g}_k \qquad \text{Eq 7.3.2} $$

Combining equations 7.3.1 and 7.3.2 gives:

$$ {}^S\hat{\boldsymbol{g}}_{A,k}^- - {}^S\hat{\boldsymbol{g}}_{G,k}^- = {}^S\boldsymbol{g}_k + {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} - {}^S\hat{\boldsymbol{g}}_{G,k}^- = \Delta\boldsymbol{R}\left(-\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right){}^S\hat{\boldsymbol{g}}_{G,k}^- + {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} - {}^S\hat{\boldsymbol{g}}_{G,k}^- \qquad \text{Eq 7.3.3} $$

$$ = \left\{\left(\boldsymbol{I} + \alpha\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)\times\right\}{}^S\hat{\boldsymbol{g}}_{G,k}^- + {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} - {}^S\hat{\boldsymbol{g}}_{G,k}^- = \left(\alpha\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\times\right){}^S\hat{\boldsymbol{g}}_{G,k}^- + {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} \qquad \text{Eq 7.3.4} $$

$$ = -\left(\alpha{}^S\hat{\boldsymbol{g}}_{G,k}^-\times\right)\hat{\boldsymbol{\theta}}_{\varepsilon,k}^- + {}^S\hat{\boldsymbol{a}}_{\varepsilon,k}^- - \boldsymbol{v}_{A,k} \qquad \text{Eq 7.3.5} $$

Similarly equation 4.3.6 gives:

$$ {}^S\hat{\boldsymbol{m}}_{M,k}^- - {}^S\boldsymbol{m}_k = -{}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} \qquad \text{Eq 7.3.6} $$

By the definition of the *a priori* orientation error, the gyroscope's estimate of the geomagnetic vector in the sensor frame is related to the true geomagnetic vector by:

$$ {}^S\hat{\boldsymbol{m}}_{G,k}^- = \Delta\boldsymbol{R}\left(\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right){}^S\boldsymbol{m}_k \qquad \text{Eq 7.3.7} $$

Combining equations 7.3.6 and 7.3.7 gives:

$$ {}^S\hat{\boldsymbol{m}}_{M,k}^- - {}^S\hat{\boldsymbol{m}}_{G,k}^- = {}^S\boldsymbol{m}_k - {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} - {}^S\hat{\boldsymbol{m}}_{G,k}^- = \Delta\boldsymbol{R}\left(-\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right){}^S\hat{\boldsymbol{m}}_{G,k}^- - {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} - {}^S\hat{\boldsymbol{m}}_{G,k}^- \qquad \text{Eq 7.3.8} $$

$$ = \left\{\left(\boldsymbol{I} + \alpha\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)\times\right\}{}^S\hat{\boldsymbol{m}}_{G,k}^- - {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} - {}^S\hat{\boldsymbol{m}}_{G,k}^- = \left(\alpha\hat{\boldsymbol{\theta}}_{\varepsilon,k}^-\times\right){}^S\hat{\boldsymbol{m}}_{G,k}^- - {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} \qquad \text{Eq 7.3.9} $$

$$ = -\left(\alpha{}^S\hat{\boldsymbol{m}}_{G,k}^-\times\right)\hat{\boldsymbol{\theta}}_{\varepsilon,k}^- - {}^S\hat{\boldsymbol{d}}_{\varepsilon,k}^- + \boldsymbol{v}_{M,k} \qquad \text{Eq 7.3.10} $$

The measurement matrix $\boldsymbol{C}_k$ can then be written as:

$$ \boldsymbol{C}_k = \begin{pmatrix} -\alpha\left({}^S\hat{\boldsymbol{g}}_{G,k}^-\times\right) & \boldsymbol{0_3} & \boldsymbol{I}_3 & \boldsymbol{0_3} \\ -\alpha\left({}^S\hat{\boldsymbol{m}}_{G,k}^-\times\right) & \boldsymbol{0_3} & \boldsymbol{0_3} & -\boldsymbol{I}_3 \end{pmatrix} \qquad \text{Eq 7.3.11} $$

Equations 7.3.10 and 7.3.11 are not entirely satisfactory since, as written, they are independent of the gyro zero rate offset error vector $\widehat{\boldsymbol{b}}^{-}_{\varepsilon,k}$. The derivation in the next section expands the orientation error component $\widehat{\boldsymbol{\theta}}^{-}_{\varepsilon,k}$ to include the gyroscope offset error vector $\widehat{\boldsymbol{b}}^{-}_{\varepsilon,k}$.

7.4 Extended Model of the Measurement Matrix

Substituting equation 4.1.8 into equation 5.1.1 gives:

$$\widehat{\boldsymbol{R}}^{-}_{k} = \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}^{-}_{k}\delta t)\widehat{\boldsymbol{R}}^{+}_{k-1} = \Delta\boldsymbol{R}\left\{\left(\boldsymbol{\omega}_k - \widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1} + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k}\right)\delta t\right\}\widehat{\boldsymbol{R}}^{+}_{k-1} \qquad \text{Eq 7.4.1}$$

$$= \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)\Delta\boldsymbol{R}\left(-\widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{w}_{b,k}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{v}_{G,k}\delta t\right)\widehat{\boldsymbol{R}}^{+}_{k-1} \qquad \text{Eq 7.4.2}$$

The four small angle rotation matrices on the right hand side of equation 7.4.2 all mutually commute and can be rearranged as needed.

By definition:

$$\widehat{\boldsymbol{R}}^{-}_{k} = \Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{-}_{\varepsilon,k}\right)\boldsymbol{R}_k \qquad \text{Eq 7.4.3}$$

$$\widehat{\boldsymbol{R}}^{+}_{k} = \Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k}\right)\boldsymbol{R}_k \qquad \text{Eq 7.4.4}$$

$$\boldsymbol{R}_k = \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)\boldsymbol{R}_{k-1} \qquad \text{Eq 7.4.5}$$

Substituting equations 7.4.3 through 7.4.5 into equation 7.4.2 and rearranging gives:

$$\Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{-}_{\varepsilon,k}\right)\boldsymbol{R}_k = \Delta\boldsymbol{R}\left(-\widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{w}_{b,k}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{v}_{G,k}\delta t\right)\Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k-1}\right)\Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)\boldsymbol{R}_{k-1} \qquad \text{Eq 7.4.6}$$

$$= \Delta\boldsymbol{R}\left(-\widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{w}_{b,k}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{v}_{G,k}\delta t\right)\Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k-1}\right)\boldsymbol{R}_k \qquad \text{Eq 7.4.7}$$

$$\Rightarrow \Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{-}_{\varepsilon,k}\right) = \Delta\boldsymbol{R}\left(-\widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{w}_{b,k}\delta t\right)\Delta\boldsymbol{R}\left(\boldsymbol{v}_{G,k}\delta t\right)\Delta\boldsymbol{R}\left(\widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k-1}\right) \qquad \text{Eq 7.4.8}$$

$$\Rightarrow \widehat{\boldsymbol{\theta}}^{-}_{\varepsilon,k} = \widehat{\boldsymbol{\theta}}^{+}_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^{+}_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t \qquad \text{Eq 7.4.9}$$

Equation 7.4.9 allows the measurement matrix to be written in the form:

$$\boldsymbol{C}_k = \begin{pmatrix} -\alpha\left({}^{S}\widehat{\boldsymbol{g}}^{-}_{G,k}\times\right) & \alpha\delta t\left({}^{S}\widehat{\boldsymbol{g}}^{-}_{G,k}\times\right) & \boldsymbol{I}_3 & \boldsymbol{0_3} \\ -\alpha\left({}^{S}\widehat{\boldsymbol{m}}^{-}_{G,k}\times\right) & \alpha\delta t\left({}^{S}\widehat{\boldsymbol{m}}^{-}_{G,k}\times\right) & \boldsymbol{0_3} & -\boldsymbol{I}_3 \end{pmatrix} \qquad \text{Eq 7.4.10}$$

Expanding the 3x3 blocks in equation 7.4.10 gives:

$$\boldsymbol{C}_k = \begin{pmatrix} 0 & \alpha^S\hat{g}^{-}_{Gz,k} & -\alpha^S\hat{g}^{-}_{Gy,k} & 0 & -\alpha\delta t^S\hat{g}^{-}_{Gz,k}0 & \alpha\delta t^S\hat{g}^{-}_{Gy,k} & 1 & 0 & 0 & 0 & 0 & 0 \\ -\alpha^S\hat{g}^{-}_{Gz,k} & 0 & \alpha^S\hat{g}^{-}_{Gx,k} & \alpha\delta t^S\hat{g}^{-}_{Gz,k} & 0 & -\alpha\delta t^S\hat{g}^{-}_{Gx,k} & 0 & 1 & 0 & 0 & 0 & 0 \\ \alpha^S\hat{g}^{-}_{Gy,k} & -\alpha^S\hat{g}^{-}_{Gx,k} & 0 & -\alpha\delta t^S\hat{g}^{-}_{Gy,k} & \alpha\delta t^S\hat{g}^{-}_{Gx,k} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \alpha^S\widehat{m}^{-}_{Gz,k} & -\alpha^S\widehat{m}^{-}_{Gy,k} & 0 & -\alpha\delta t^S\widehat{m}^{-}_{Gz,k} & \alpha\delta t^S\widehat{m}^{-}_{Gy,k} & 0 & 0 & 0 & -1 & 0 & 0 \\ -\alpha^S\widehat{m}^{-}_{Gz,k} & 0 & \alpha^S\widehat{m}^{-}_{Gx,k} & \alpha\delta t^S\widehat{m}^{-}_{Gz,k} & 0 & -\alpha\delta t^S\widehat{m}^{-}_{Gx,k} & 0 & 0 & 0 & 0 & -1 & 0 \\ \alpha^S\widehat{m}^{-}_{Gy,k} & -\alpha^S\widehat{m}^{-}_{Gx,k} & 0 & -\alpha\delta t^S\widehat{m}^{-}_{Gy,k} & \alpha\delta t^S\widehat{m}^{-}_{Gx,k} & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Eq 7.4.11

The component terms in equation 7.4.11 are computed from the gyroscope estimate of the orientation according to equations 7.2.4 and 7.2.6.

8 A Posteriori Error Vector Calculation

8.1 Magnetic Jamming Flag

Equation 3.3.9 is the standard Kalman equation for determining the a posteriori error vector $\hat{\boldsymbol{x}}_{\varepsilon,k}^{+}$ in terms of the Kalman gain matrix $\boldsymbol{K}_k$ and the measurement error vector $\boldsymbol{z}_{\varepsilon,k}$ as:

$$\hat{\boldsymbol{x}}_{\varepsilon,k}^{+} = \begin{pmatrix} \hat{\boldsymbol{\theta}}_{\varepsilon,k}^{+} \\ \hat{\boldsymbol{b}}_{\varepsilon,k}^{+} \\ {}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^{+} \\ {}^{S}\hat{\boldsymbol{d}}_{\varepsilon,k}^{+} \end{pmatrix} = \boldsymbol{K}_k {}^{S}\boldsymbol{z}_{\varepsilon,k} = \boldsymbol{K}_k \begin{pmatrix} {}^{S}\hat{\boldsymbol{g}}_{A,k}^{-} - {}^{S}\hat{\boldsymbol{g}}_{G,k}^{-} \\ {}^{S}\hat{\boldsymbol{m}}_{M,k}^{-} - {}^{S}\hat{\boldsymbol{m}}_{G,k}^{-} \end{pmatrix} \quad \text{Eq 8.1.1}$$

The *a posteriori* magnetic disturbance error vector ${}^{S}\hat{\boldsymbol{d}}_{\varepsilon,k}^{+}$ is always computed according to equation 8.1.1.

The presence of magnetic jamming is detected by the test:

$$\left| {}^{S}\hat{\boldsymbol{d}}_{\varepsilon,k}^{+} \right|^2 > 4B^2 \quad \text{Eq 8.1.2}$$

Equation 8.1.2 determines whether equation 8.11 or a simplified form is used to update the remaining components $\hat{\boldsymbol{\theta}}_{\varepsilon,k}^{+}$, $\hat{\boldsymbol{b}}_{\varepsilon,k}^{+}$ and ${}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^{+}$ of the *a posteriori* error vector $\hat{\boldsymbol{x}}_{\varepsilon,k}^{+}$.

8.2 Normal Case (No Magnetic Jamming)

In the normal case, where equation 8.1.2 is not satisfied, then the components $\hat{\boldsymbol{\theta}}_{\varepsilon,k}^{+}$, $\hat{\boldsymbol{b}}_{\varepsilon,k}^{+}$ and ${}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^{+}$

are computed according to equation 8.1.1.

8.3 Magnetic Jamming Case

When equation 8.1.2 is satisfied, the geomagnetic vector estimate is deemed inaccurate and the gravity vector component only is used to update the *a posteriori* error components:

$$\begin{pmatrix} \hat{\boldsymbol{\theta}}_{\varepsilon,k}^{+} \\ \hat{\boldsymbol{b}}_{\varepsilon,k}^{+} \\ {}^{S}\hat{\boldsymbol{a}}_{\varepsilon,k}^{+} \end{pmatrix} = \boldsymbol{K}_k[0-8,0-2]\left({}^{S}\hat{\boldsymbol{g}}_{A,k}^{-} - {}^{S}\hat{\boldsymbol{g}}_{G,k}^{-}\right) \quad \text{Eq 8.3.1}$$

where the terminology $\boldsymbol{K}_k[0-8,0-2]$ refers to the first 9 rows and first 3 columns from the upper left of the full 12 row by six column Kalman gain matrix $\boldsymbol{K}_k$. Equation 8.3.1 essentially reverts the accelerometer, magnetic and gyroscope Kalman filter to the simplified accelerometer plus gyroscope Kalman filter. The orientation is still provided primarily by the gyroscope sensor but is now only stabilized against the gravity vector. The orientation will still be accurate and stabilized in roll and pitch but the loss of yaw stabilization against the geomagnetic vector will result in a slow drift about the vertical gravity axis.

Once the external magnetic interference is removed, the Kalman filter will re-stabilize the gyroscope zero rate offset about the vertical gravity axis and will correct the compass heading relative to the geomagnetic vector.

9 Geomagnetic Inclination Angle Tracking

9.1 Introduction

The hard and soft iron algorithms compute the geomagnetic field strength but not the inclination angle δ which defines the angle by which the geomagnetic vector dips below horizontal. In the northern hemisphere δ is positive, rising from near zero at the equator to 90^o at the north magnetic pole where the geomagnetic field points vertically downwards. In the southern hemisphere δ is typically negative, falling from near zero at the equator to -90 degrees at the south magnetic pole where the geomagnetic field points vertically upwards. A further complication is that the inclination angle can be strongly influenced by nearby ferromagnetic materials such as steel building reinforcements. The inclination angle is required by the sensor fusion algorithms and must be determined in real time. The approach taken is to use the magnetic disturbance error vector ${}^S\hat{\boldsymbol{d}}^+_{\varepsilon,k}$ to update the inclination angle without modifying the geomagnetic field strength from the value determined by magnetic calibration algorithms.

9.2 Algorithm

Each iteration of the Kalman filter computes the *a posteriori* magnetic disturbance error vector in the sensor frame ${}^S\hat{\boldsymbol{d}}^+_{\varepsilon,k}$:

$${}^S\hat{\boldsymbol{d}}^+_{\varepsilon,k} = \begin{pmatrix} {}^S\hat{d}^+_{\varepsilon,x,k} \\ {}^S\hat{d}^+_{\varepsilon,y,k} \\ {}^S\hat{d}^+_{\varepsilon,x,k} \end{pmatrix} \quad \text{Eq 9.2.1}$$

Equation 9.2.1 can be rotated into the global reference frame using the inverse (transpose) of the *a posteriori* orientation matrix to give the magnetic disturbance in the global frame ${}^G\hat{\boldsymbol{d}}^+_{\varepsilon,k}$:

$${}^G\hat{\boldsymbol{d}}^+_{\varepsilon,k} = \left(\hat{\boldsymbol{R}}^+_k\right)^T {}^S\hat{\boldsymbol{d}}^+_{\varepsilon,k} \quad \text{Eq 9.2.2}$$

The horizontal and vertical components of ${}^G\hat{\boldsymbol{d}}^+_{\varepsilon,k}$ are interpreted as resulting from an error in the estimated inclination angle and used to update the inclination angle δ.

For the Aerospace / NED coordinate system the *x* axis points northwards and the *z* axis downwards. The new inclination angle estimate δ_k for iteration k is then given by simple trigonometry as:

$$\delta_k = tan^{-1}\left(\frac{{}^Gm_{z,k-1} - {}^G\hat{d}^+_{\varepsilon z,k}}{{}^Gm_{x,k-1} - {}^G\hat{d}^+_{\varepsilon x,k}}\right)$$

Aerospace / NED Eq 9.2.3

where ${}^Gm_{x,k-1}$ and ${}^Gm_{z,k-1}$ are the *x* and *y* components of the geomagnetic vector in the global frame ${}^G\boldsymbol{m}_k$.

The geomagnetic vector ${}^G\boldsymbol{m}_k$ is not, however, updated by simply subtracting the relevant components of ${}^G\hat{\boldsymbol{d}}^+_{\varepsilon,k}$ since the magnitude of the new vector would no longer equal the geomagnetic field strength B computed by the calibration algorithms. Instead, simple trigonometry gives:

$${}^G\boldsymbol{m}_k = \begin{pmatrix} {}^Gm_{x,k} \\ {}^Gm_{y,k} \\ {}^Gm_{z,k} \end{pmatrix} = \begin{pmatrix} Bcos(\delta_k) \\ 0 \\ Bsin(\delta_k) \end{pmatrix}$$

Aerospace / NED Eq 9.2.4

For the Android and Windows 8 coordinate systems, the *y* axis points northwards and the *z* axis points upwards and the inclination angle update equations become:

$$\delta_k = tan^{-1}\left(\frac{-\left({}^Gm_{z,k-1} - {}^G\hat{d}^+_{\varepsilon z,k}\right)}{{}^Gm_{y,k-1} - {}^G\hat{d}^+_{\varepsilon y,k}}\right)$$

Android, Windows 8 Eq 9.2.5

$$^{G}\boldsymbol{m}_k = \begin{pmatrix} ^{G}m_{x,k} \\ ^{G}m_{y,k} \\ ^{G}m_{z,k} \end{pmatrix} = \begin{pmatrix} 0 \\ Bcos(\delta_k) \\ -Bsin(\delta_k) \end{pmatrix}$$ Android, Windows 8 Eq 9.2.6

For all coordinate systems, the geomagnetic field strength is unaltered:

$$|^{G}\boldsymbol{m}_k| = B$$ Eq 9.2.7

9.3 Geometric Implementation

Calculation of sines and cosines is particularly expensive in terms of clock ticks on microcontrollers lacking a hardware floating point unit. A more efficient geometrical algorithm is used instead.

For the Aerospace / NED coordinate system:

$$sin(\delta_k) = \frac{^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}}{\sqrt{\left(^{G}m_{x,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon x,k}\right)^2 + \left(^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}\right)^2}}$$

Aerospace / NED Eq 9.3.1

$$cos(\delta_k) = \frac{^{G}m_{x,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon x,k}}{\sqrt{\left(^{G}m_{x,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon x,k}\right)^2 + \left(^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}\right)^2}}$$

Aerospace / NED Eq 9.3.2

For the Android and Windows 8 coordinate systems:

$$sin(\delta_k) = \frac{-\left(^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}\right)}{\sqrt{\left(^{G}m_{y,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon y,k}\right)^2 + \left(^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}\right)^2}}$$

Android, Windows 8 Eq 9.3.3

$$cos(\delta_k) = \frac{^{G}m_{y,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon y,k}}{\sqrt{\left(^{G}m_{y,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon y,k}\right)^2 + \left(^{G}m_{z,k-1} - {}^{G}\hat{d}^{+}_{\varepsilon z,k}\right)^2}}$$

Android, Windows 8 Eq 9.3.4

Equations 9.2.4 and 9.2.6 remain unchanged.

10 Covariance Matrix Updates

10.1 Process Error Qw Matrix Terms

The covariance matrix $\boldsymbol{Q}_w$ is required in Kalman equations 2, 3 and 4 (see section 3). Kalman equation C (equation 3.3.14) updates $\boldsymbol{Q}_w$ as a function of the a posteriori covariance matrix $\boldsymbol{P}_k^+$. This section derives the explicit form of equation 3.3.14.

Setting $\boldsymbol{A}_k = \boldsymbol{0}$ into equation 3.1.2 and taking the covariance gives :

$$E\left[\widehat{\boldsymbol{x}}_{\varepsilon,k}^-{\widehat{\boldsymbol{x}}_{\varepsilon,k}^-}^T\right] = E\left[\begin{pmatrix} \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^- \\ \widehat{\boldsymbol{b}}_{\varepsilon,k}^- \\ {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^- \\ {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^- \end{pmatrix}\begin{pmatrix} \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^- \\ \widehat{\boldsymbol{b}}_{\varepsilon,k}^- \\ {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^- \\ {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^- \end{pmatrix}^T\right] = E\left[\boldsymbol{w}_k {\boldsymbol{w}_k}^T\right] = \boldsymbol{Q}_w \quad \text{Eq 10.1.1}$$

Expanding equation 10.1.1 into 3x3 blocks gives:

$$\boldsymbol{Q}_w = E\begin{pmatrix} \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{b}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\right)^T \\ \widehat{\boldsymbol{b}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{b}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{b}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{b}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\right)^T & \widehat{\boldsymbol{b}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\right)^T \\ {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{b}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\right)^T \\ {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\left(\widehat{\boldsymbol{b}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-\right)^T & {}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\left({}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-\right)^T \end{pmatrix} \quad \text{Eq 10.1.2}$$

By inspection, the matrix $\boldsymbol{Q}_w$ is symmetric and only the on-diagonal and above-diagonal terms need to be considered.

Equations 7.4.9 and 4.1.5, reproduced below, allow the determination of some of the required terms:

$$\widehat{\boldsymbol{\theta}}_{\varepsilon,k}^- = \widehat{\boldsymbol{\theta}}_{\varepsilon,k-1}^+ - \widehat{\boldsymbol{b}}_{\varepsilon,k-1}^+\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t \quad \text{Eq 10.1.3}$$

$$\widehat{\boldsymbol{b}}_{\varepsilon,k}^- = \widehat{\boldsymbol{b}}_{\varepsilon,k-1}^+ - \boldsymbol{w}_{b,k} \quad \text{Eq 10.1.4}$$

The remaining terms require the derivation of expressions for ${}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^-$ and ${}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^-$.

By definition:

$${}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^- = {}^S\widehat{\boldsymbol{a}}_k^- - {}^S\boldsymbol{a}_k \quad \text{Eq 10.1.5}$$

Substituting equation 6.1.3 gives:

$${}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^- = \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{k-1}^+ - {}^S\boldsymbol{a}_k = \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{k-1}^+ - \boldsymbol{R}_k\left(c_a{}^G\boldsymbol{a}_{k-1} + \boldsymbol{w}_{a,k}\right) \quad \text{Eq 10.1.6}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{k-1}^+ - \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)\boldsymbol{R}_{k-1}c_a{}^G\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.7}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{k-1}^+ - \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)c_a{}^S\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.8}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{k-1}^+ - \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\boldsymbol{a}_{k-1} + \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\boldsymbol{a}_{k-1} - \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)c_a{}^S\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.9}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a({}^S\widehat{\boldsymbol{a}}_{k-1}^+ - {}^S\boldsymbol{a}_{k-1}) + \{\Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t) - \Delta\boldsymbol{R}(\boldsymbol{\omega}_k\delta t)\}c_a{}^S\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.10}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^+ + \{\alpha\delta t(\boldsymbol{\omega}_k - \widehat{\boldsymbol{\omega}}_k^-)\times\}c_a{}^S\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.11}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^+ - \left\{\alpha\delta t\widehat{\boldsymbol{\omega}}_{\varepsilon,k}^-\times\right\}c_a{}^S\boldsymbol{a}_{k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.12}$$

$$= \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t)c_a{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^+ + \{\alpha\delta t{}^S\boldsymbol{a}_{k-1}\times\}c_a\widehat{\boldsymbol{\omega}}_{\varepsilon,k}^- - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.13}$$

Substituting for $\widehat{\boldsymbol{\omega}}_{\varepsilon,k}^-$ from equation 4.1.9 gives:

$${}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^- = c_a\Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}_k^-\delta t){}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^+ + c_a\{\alpha\delta t{}^S\boldsymbol{a}_{k-1}\times\}\left(-\widehat{\boldsymbol{b}}_{\varepsilon,k-1}^+ + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k}\right) - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \approx c_a{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^+ - \boldsymbol{R}_k\boldsymbol{w}_{a,k} \quad \text{Eq 10.1.14}$$

By analogy with equation 10.1.14

$$ {}^S\widehat{\boldsymbol{d}}^-_{\varepsilon,k} \approx c_d \Delta\boldsymbol{R}(\widehat{\boldsymbol{\omega}}^-_k \delta t)\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} + c_d\{\alpha\delta t\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k} \times\}\left(-\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} + \boldsymbol{w}_{b,k} + \boldsymbol{v}_{G,k}\right) - \boldsymbol{R}_k \boldsymbol{w}_{d,k} \approx c_d\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k \boldsymbol{w}_{d,k} \quad \text{Eq 10.1.15} $$

Equations 10.1.3, 10.1.4, 10.1.14 and 10.1.15 provide the needed terms for evaluating the covariance matrix. In the following derivations all cross-correlations terms are ignored relative to auto-correlation terms.

$$ E\left[\widehat{\boldsymbol{\theta}}^-_{\varepsilon,k} {\widehat{\boldsymbol{\theta}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t\right)\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t\right)^T\right] \quad \text{Eq 10.1.16} $$

$$ = \boldsymbol{Q}^+_{\theta\varepsilon\theta\varepsilon,k-1} + \delta t^2\left\{\boldsymbol{Q}^+_{b\varepsilon b\varepsilon,k-1} + (Q_{wb} + Q_{vG})\boldsymbol{I}\right\} \quad \text{Eq 10.1.17} $$

where:

$$ \boldsymbol{Q}^+_{\theta\varepsilon\theta\varepsilon,k-1} = E\left[\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1}\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1}\right)^T\right] = \boldsymbol{P}^+_{k-1}[0,2:0,2] \quad \text{Eq 10.1.18} $$

$$ \boldsymbol{Q}^+_{b\varepsilon b\varepsilon,k-1} = E\left[\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\right)^T\right] = \boldsymbol{P}^+_{k-1}[3,5:3,5] \quad \text{Eq 10.1.19} $$

$$ Q_{wb}\boldsymbol{I} = E\left[\boldsymbol{w}_{b,k}\left(\boldsymbol{w}_{b,k}\right)^T\right] \quad \text{Eq 10.1.20} $$

$$ Q_{vG}\boldsymbol{I} = E\left[\boldsymbol{v}_{G,k}\left(\boldsymbol{v}_{G,k}\right)^T\right] \quad \text{Eq 10.1.21} $$

and where $\boldsymbol{P}^+_{k-1}[i,j:i,j]$ means the sub-matrix of $\boldsymbol{P}^+_{k-1}$ located in rows i to j and columns i to j.

$$ E\left[\widehat{\boldsymbol{\theta}}^-_{\varepsilon,k} {\widehat{\boldsymbol{b}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t\right)\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} - \boldsymbol{w}_{b,k}\right)^T\right] \quad \text{Eq 10.1.22} $$

$$ = -\delta t\left(\boldsymbol{Q}^+_{b\varepsilon b\varepsilon,k-1} + Q_{wb}\boldsymbol{I}\right) = -\delta t(\boldsymbol{P}^+_{k-1}[0,2:3,5] + Q_{wb}\boldsymbol{I}) \quad \text{Eq 10.1.23} $$

$$ E\left[\widehat{\boldsymbol{b}}^-_{\varepsilon,k} {\widehat{\boldsymbol{b}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} - \boldsymbol{w}_{b,k}\right)\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} - \boldsymbol{w}_{b,k}\right)^T\right] = \boldsymbol{Q}^+_{b\varepsilon b\varepsilon,k-1} + Q_{wb}\boldsymbol{I} = \boldsymbol{P}^+_{k-1}[3,5:3,5] + Q_{wb}\boldsymbol{I} \quad \text{Eq 10.1.24} $$

$$ E\left[\widehat{\boldsymbol{\theta}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{a}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t\right)\left(c_a\, {}^S\widehat{\boldsymbol{a}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k}\right)^T\right] = 0 \quad \text{Eq 10.1.25} $$

$$ E\left[\widehat{\boldsymbol{\theta}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{d}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{\theta}}^+_{\varepsilon,k-1} - \widehat{\boldsymbol{b}}^+_{\varepsilon,k-1}\delta t + \boldsymbol{w}_{b,k}\delta t + \boldsymbol{v}_{G,k}\delta t\right)\left(c_d\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{d,k}\right)^T\right] = 0 \quad \text{Eq 10.1.26} $$

$$ E\left[\widehat{\boldsymbol{b}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{a}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} - \boldsymbol{w}_{b,k}\right)\left(c_a\, {}^S\widehat{\boldsymbol{a}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k}\right)^T\right] = 0 $$

Eq 10.1.27

$$ E\left[\widehat{\boldsymbol{b}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{d}}^-_{\varepsilon,k}}^T\right] = E\left[\left(\widehat{\boldsymbol{b}}^+_{\varepsilon,k-1} - \boldsymbol{w}_{b,k}\right)\left(c_d\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{d,k}\right)^T\right] = 0 $$

Eq 10.1.28

$$ E\left[{}^S\widehat{\boldsymbol{a}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{a}}^-_{\varepsilon,k}}^T\right] = E\left[\left(c_a\, {}^S\widehat{\boldsymbol{a}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k}\right)\left(c_a\, {}^S\widehat{\boldsymbol{a}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{a,k}\right)^T\right] = {c_a}^2\boldsymbol{Q}^+_{a\varepsilon a\varepsilon,k-1} + Q_{wa}\boldsymbol{I} \quad \text{Eq 10.1.29} $$

$$ = {c_a}^2\boldsymbol{P}^+_{k-1}[6,8:6,8] + Q_{wa}\boldsymbol{I} \quad \text{Eq 10.1.30} $$

$$ E\left[{}^S\widehat{\boldsymbol{d}}^-_{\varepsilon,k} {{}^S\widehat{\boldsymbol{d}}^-_{\varepsilon,k}}^T\right] = E\left[\left(c_d\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{d,k}\right)\left(c_d\, {}^S\widehat{\boldsymbol{d}}^+_{\varepsilon,k-1} - \boldsymbol{R}_k\boldsymbol{w}_{d,k}\right)^T\right] = {c_d}^2\boldsymbol{Q}^+_{d\varepsilon d\varepsilon,k-1} + Q_{wd}\boldsymbol{I} \quad \text{Eq 10.1.31} $$

$$= c_d{}^2 \boldsymbol{P}_{k-1}^{+}[9{,}11{:}\,9{,}11] + Q_{wd}\boldsymbol{I}$$ Eq 10.1.32

$$E\left[{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k}^{-}\,{}^S\widehat{\boldsymbol{d}}_{\varepsilon,k}^{-}{}^T\right] = E\left[\left(c_a{}^S\widehat{\boldsymbol{a}}_{\varepsilon,k-1}^{+} - \boldsymbol{R}_k\boldsymbol{w}_{a,k}\right)\left(c_d{}^S\widehat{\boldsymbol{d}}_{\varepsilon,k-1}^{+} - \boldsymbol{R}_k\boldsymbol{w}_{d,k}\right)^T\right] = 0$$ Eq 10.1.33

10.2 Measurement Error Covariance Qv Matrix

The covariance matrix $\boldsymbol{Q}_v$ is the covariance matrix of the noise term v_k defined in equation 3.2.2. By inspection, it evaluates to the sums of the covariances of the three sensors defining the measurement ${}^S\boldsymbol{z}_{\varepsilon,k}$:

$$\boldsymbol{Q}_v = E[\boldsymbol{v}_k\boldsymbol{v}_k{}^T] = \begin{pmatrix} \left[Q_{vA} + Q_{wa} + \alpha^2\delta t^2\left(Q_{wb} + Q_{vG}\right)\right]\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \left[Q_{vM} + Q_{wd} + \alpha^2\delta t^2 B^2\left(Q_{wb} + Q_{vG}\right)\right]\boldsymbol{I}_3 \end{pmatrix}$$ Eq 10.2.1