AN5021

Calculation of Orientation Matrices from Sensor Data

Rev. 2.0 — 21 June 2016 **Application note**

Document information

Info	Content
Abstract	This application note introduces the rotation or orientation matrix and documents the functions in file *orientation.c* used by the NXP Sensor Fusion Library to compute an orientation matrix from different combinations of sensor measurements.

Revision history

Document ID	Release date	Supercedes
AN5021 v2.0	20160621	AN5021 v1.0
Modifications:	• Minor changes • The format of this document has been redesigned to comply with the new identity guidelines of NXP Semiconductors. Legal texts have been adapted to the new company name where appropriate.	
AN5021 v1.0	2015 October	—

Contact information

For more information, please visit: http://www.nxp.com

1. Introduction

1.1 Summary

This application note introduces the rotation or orientation matrix and documents the functions in file *orientation.c* used by the NXP Sensor Fusion Library to compute an orientation matrix from different combinations of sensor measurements.

1.2 Terminology

Symbol	Definition
B	Geomagnetic field strength
${}^{S}\boldsymbol{B}_{c}$	Calibrated magnetometer measurements in the sensor frame
${}^{S}B_{cx},\ {}^{S}B_{cy},\ {}^{S}B_{cz}$	*x*, *y* and *z* components of the calibrated magnetometer reading in the sensor frame
ENU	*x*=East, *y*=North, *z*=Up coordinate system (Android and Windows 8)
${}^{S}\boldsymbol{G}_{c}$	Calibrated accelerometer measurement in the sensor frame
${}^{S}G_{cx},\ {}^{S}G_{cy},\ {}^{S}G_{cz}$	*x*, *y* and *z* components of the accelerometer reading in the sensor frame
$\hat{\boldsymbol{n}}$	Normalized rotation axis
$\hat{n}_x, \hat{n}_y, \hat{n}_z$	Components of normalized rotation axis $\hat{\boldsymbol{n}}$
NED	*x*=North, *y*=East, *z*=Down coordinate system (Aerospace)
$\boldsymbol{R}$	General orientation matrix
$\boldsymbol{R}_{Android}$	Orientation matrix in the Android coordinate system
$\boldsymbol{R}_{NED}$	Orientation matrix in the NED/Aerospace coordinate system
$\boldsymbol{R}_{Win8}$	Orientation matrix in the Windows 8 coordinate system
R_{xy}	Element in row *x* and column *y* of orientation matrix $\boldsymbol{R}$
$tr(\boldsymbol{R})$	Trace of orientation matrix $\boldsymbol{R}$
δ	Geomagnetic inclination angle
η	Rotation angle
θ	Pitch angle
ϕ	Roll angle
ψ	Yaw angle

1.3 Software Functions

Functions	Description	Reference

`void f3DOFTiltNED (float fR[][3], float fGc[]);`
 	
`void f3DOFTiltAndroid (float fR[][3], float fGc[]);`
 	
`void f3DOFTiltWin8 (float fR[][3], float fGc[]);`
	Computes the tilt orientation matrix from accelerometer measurements in the Aerospace/NED, Android and Windows 8 coordinate systems.	4

`void f3DOFMagnetometerMatrixNED (float fR[][3], float fBc[]);`
 	
`void f3DOFMagnetometerMatrixAndroid (float fR[][3], float fBc[]);`
 	
`void f3DOFMagnetometerMatrixWin8 (float fR[][3], float fBc[]);`
	Computes the 2D eCompass yaw orientation matrix from calibrated magnetometer measurements in the Aerospace/NED, Android and Windows 8 coordinate systems.	5

`void feCompassNED (float fR[][3], float *pfDelta, float *pfsinDelta, float *pfcosDelta, float fBc[], float fGc[], float *pfmodBc, float *pfmodGc)`
 	
`void feCompassAndroid (float fR[][3], float *pfDelta, float *pfsinDelta, float *pfcosDelta, float fBc[], float fGc[], float *pfmodBc, float *pfmodGc)`
 	
`void feCompassWin8 (float fR[][3], float *pfDelta, float *pfsinDelta, float *pfcosDelta, float fBc[], float fGc[], float *pfmodBc, float *pfmodGc)`
	Computes the full 3D orientation matrix from accelerometer and calibrated magnetometer measurements in the Aerospace/NED, Android and Windows 8 coordinate systems.
	Also returns the geomagnetic inclination angle and the moduli of the accelerometer and magnetometer measurements.
		6

2. General Rotation Matrix

2.1 Introduction

The general rotation matrix $\boldsymbol{R}$ which transforms a vector as a result of a rotation of the coordinate system around the normalized axis $\hat{\boldsymbol{n}}$ by angle η is:

$$\boldsymbol{R} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} = \begin{pmatrix} \hat{n}_x{}^2 + (1-\hat{n}_x{}^2)cos\eta & \hat{n}_x\hat{n}_y(1-cos\eta) + \hat{n}_z sin\eta & \hat{n}_x\hat{n}_z(1-cos\eta) - \hat{n}_y sin\eta \\ \hat{n}_x\hat{n}_y(1-cos\eta) - \hat{n}_z sin\eta & \hat{n}_y{}^2 + (1-\hat{n}_y{}^2)cos\eta & \hat{n}_y\hat{n}_z(1-cos\eta) + \hat{n}_x sin\eta \\ \hat{n}_x\hat{n}_z(1-cos\eta) + \hat{n}_y sin\eta & \hat{n}_y\hat{n}_z(1-cos\eta) - \hat{n}_x sin\eta & \hat{n}_z{}^2 + (1-\hat{n}_z{}^2)cos\eta \end{pmatrix} \quad \textbf{(1)}$$

Equation (1) is known as the Rodrigues rotation matrix. The product of the rotation axis $\hat{\boldsymbol{n}}$ and rotation angle η is termed the rotation vector $\hat{\boldsymbol{n}}\eta$.

If the coordinate system rotates by angle η about the z axis, then a vector rotates by angle $-\eta$ about the z axis, relative to the newly rotated coordinate system. When NXP documents refer to a rotation vector defining a rotation matrix, it always refers to a coordinate system rotation.

In addition, when a rotation matrix $\boldsymbol{R}$ models the orientation of the sensors relative to the earth (global) reference frame, the NXP Sensor Fusion Library standard is that the product $\boldsymbol{Rv}$ transforms the vector $\boldsymbol{v}$ from the global to the sensor frame. The inverse rotation $\boldsymbol{R}^{-1}\boldsymbol{v} = \boldsymbol{R}^T\boldsymbol{v}$ transforms the vector $\boldsymbol{v}$ from the sensor to the global frame. The orientation matrix is always defined relative to an initial orientation in the global frame with the product flat and pointed to magnetic north giving zero roll, pitch and yaw angles.

Two special cases of equation (1) occur for rotation angles of i) 0º and ii) 180º about any axis. Substituting into equation (1) gives the 0º rotation matrix $\boldsymbol{R}(0)$ and 180º rotation matrix $\boldsymbol{R}(\pi)$ as:

$$\boldsymbol{R}(0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \textbf{(2)}$$

$$\boldsymbol{R}(\pi) = \begin{pmatrix} 2\hat{n}_x{}^2 - 1 & 2\hat{n}_x\hat{n}_y & 2\hat{n}_x\hat{n}_z \\ 2\hat{n}_x\hat{n}_y & 2\hat{n}_y{}^2 - 1 & 2\hat{n}_y\hat{n}_z \\ 2\hat{n}_x\hat{n}_z & 2\hat{n}_y\hat{n}_z & 2\hat{n}_z{}^2 - 1 \end{pmatrix} \quad \textbf{(3)}$$

In both cases the rotation matrix is symmetric.

2.2 Eigenvector and Eigenvalue

The normalized rotation axis $\hat{\boldsymbol{n}}$ is the eigenvector of the rotation matrix product $\boldsymbol{R}$ corresponding to eigenvalue 1. Geometrically, this means that any vector parallel to the rotation axis is unchanged by the rotation.

$$\boldsymbol{R}\hat{\boldsymbol{n}} = \begin{pmatrix} \hat{n}_x{}^2 + (1 - \hat{n}_x{}^2)cos\eta & \hat{n}_x\hat{n}_y(1 - cos\eta) + \hat{n}_z sin\eta & \hat{n}_x\hat{n}_z(1 - cos\eta) - \hat{n}_y sin\eta \\ \hat{n}_x\hat{n}_y(1 - cos\eta) - \hat{n}_z sin\eta & \hat{n}_y{}^2 + (1 - \hat{n}_y{}^2)cos\eta & \hat{n}_y\hat{n}_z(1 - cos\eta) + \hat{n}_x sin\eta \\ \hat{n}_x\hat{n}_z(1 - cos\eta) + \hat{n}_y sin\eta & \hat{n}_y\hat{n}_z(1 - cos\eta) - \hat{n}_x sin\eta & \hat{n}_z{}^2 + (1 - \hat{n}_z{}^2)cos\eta \end{pmatrix} \begin{pmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{pmatrix} \quad \textbf{(4)}$$

$$= \begin{pmatrix} \hat{n}_x(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2) - \hat{n}_x cos\eta(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2 - 1) \\ \hat{n}_y(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2) - \hat{n}_y cos\eta(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2 - 1) \\ \hat{n}_z(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2) - \hat{n}_z cos\eta(\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2 - 1) \end{pmatrix} = \begin{pmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{pmatrix} \quad \textbf{(5)}$$

3. Converting Between Rotation Matrix and Rotation Vector

3.1 Rotation Vector from Rotation Matrix

Equation (1) allows the rotation angle η to be determined from the trace of the rotation matrix $\boldsymbol{R}$:

$$tr(\boldsymbol{R}) = R_{xx} + R_{yy} + R_{zz} = \hat{n}_x{}^2 + (1 - \hat{n}_x{}^2)cos\eta + \hat{n}_y{}^2 + (1 - \hat{n}_y{}^2)cos\eta + \hat{n}_z{}^2 + (1 - \hat{n}_z{}^2)cos\eta \quad (6)$$

$$= (\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2) + 3cos\eta - (\hat{n}_x{}^2 + \hat{n}_y{}^2 + \hat{n}_z{}^2)cos\eta \quad (7)$$

$$\Rightarrow tr(\boldsymbol{R}) = 1 + 2cos\eta \quad (8)$$

$$\Rightarrow \eta = cos^{-1}\left(\frac{tr(\boldsymbol{R}) - 1}{2}\right) \quad (9)$$

The trace of the matrix varies between –1 and +3. A trace of –1 corresponds to a rotation of 180° about any axis and a trace of +3 corresponds to a rotation of 0°.

There is no angle ambiguity in equation (9) because the inverse cosine returns an angle between 0° and 180° meaning that the rotation angle η also varies between 0° and 180°. Rotation angles between –180° and 0° are equivalent to rotation by the negated angle $-\eta$ about the negated rotation axis $-\hat{\boldsymbol{n}}$ and the rotation vector $\eta\hat{\boldsymbol{n}}$ is unaltered by a double sign change.

Differencing elements across the leading diagonal of $\boldsymbol{R}$ gives:

$$2\hat{\boldsymbol{n}}sin\eta = \begin{pmatrix} R_{yz} - R_{zy} \\ R_{zx} - R_{xz} \\ R_{xy} - R_{yx} \end{pmatrix} \quad (10)$$

In the general case where $sin\eta \neq 0$ and the rotation matrix is not symmetric, the rotation axis $\hat{\boldsymbol{n}}$ equals:

$$\hat{\boldsymbol{n}} = \frac{1}{\sqrt{(R_{yz} - R_{zy})^2 + (R_{zx} - R_{xz})^2 + (R_{xy} - R_{yx})^2}} \begin{pmatrix} R_{yz} - R_{zy} \\ R_{zx} - R_{xz} \\ R_{xy} - R_{yx} \end{pmatrix} \quad (11)$$

As the rotation angle η approaches zero, equation (11) becomes undefined mathematically corresponding to the physical situation that all rotation axes are equivalent at zero rotation. The rotation vector $\hat{\boldsymbol{n}}\eta$ is still, however, defined as:

$$\hat{\boldsymbol{n}}\eta = \frac{1}{2}\begin{pmatrix} R_{yz} - R_{zy} \\ R_{zx} - R_{xz} \\ R_{xy} - R_{yx} \end{pmatrix} \quad (12)$$

For the case of the rotation angle η near 180°, the rotation matrix $\boldsymbol{R}$ approximates the form of equation (3) with a trace of –1. The elements on the leading diagonal allow the rotation axis $\hat{\boldsymbol{n}}$ to be given by:

$$\hat{\boldsymbol{n}} = \begin{pmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{pmatrix} = \begin{pmatrix} \pm\sqrt{\frac{R_{xx}+1}{2}} \\ \pm\sqrt{\frac{R_{yy}+1}{2}} \\ \pm\sqrt{\frac{R_{zz}+1}{2}} \end{pmatrix} \tag{13}$$

The sign ambiguity can be resolved by differencing across the leading diagonal and using the result that $sin\eta$ is nonnegative.

$$R_{yz} - R_{zy} = 2\hat{n}_x sin\eta \Rightarrow sign(\hat{n}_x) = sign(R_{yz} - R_{zy}) \tag{14}$$

$$R_{zx} - R_{xz} = 2\hat{n}_y sin\eta \Rightarrow sign(\hat{n}_y) = sign(R_{zx} - R_{xz}) \tag{15}$$

$$R_{xy} - R_{yx} = 2\hat{n}_z sin\eta \Rightarrow sign(\hat{n}_z) = sign(R_{xy} - R_{yx}) \tag{16}$$

Any rotation can be represented by its normalized rotation axis $\hat{\boldsymbol{n}}$ and the rotation angle η about that axis, giving a total of three degrees of freedom. The rotation vector $\eta\hat{\boldsymbol{n}}$ is the most efficient encoding of a rotation but has no useful mathematical properties. The rotation quaternion (four components) and rotation matrix (9 components) are, therefore, used in preference.

The output of the gyroscope sensor (in degrees/second) multiplied by the sampling interval is a rotation vector and is therefore immediately converted in software to a rotation quaternion or matrix for orientation integration.

3.2 Rotation Matrix from Rotation Vector

Equation (1) defines the Rodrigues rotation matrix directly in terms of the rotation axis $\hat{\boldsymbol{n}}$ and rotation angle η. These can easily be determined for the case of non-zero rotation vector $\eta\hat{\boldsymbol{n}}$ using:

$$\eta = |\eta\hat{\boldsymbol{n}}| \tag{17}$$

$$\hat{\boldsymbol{n}} = \frac{\eta\hat{\boldsymbol{n}}}{|\eta\hat{\boldsymbol{n}}|} \tag{18}$$

If the rotation vector is zero then the rotation matrix is the identity matrix.

4. Accelerometer Tilt Orientation Matrix

4.1 Introduction

Accelerometer sensors are sensitive to both linear acceleration and gravity and are insensitive to rotation about the vertical gravity axis. An accelerometer operating without other sensors can therefore detect roll, pitch and tilt angles from horizontal provided there is no linear acceleration but cannot detect compass heading.

The mathematics below assumes that the yaw or compass angle is always zero degrees. One consequence of this assumption is that the orientation undergoes a twist of 180° at ±90° pitch (Aerospace / NED coordinate system) and at ±90° roll (in Android and Windows 8 coordinate systems) to ensure that the orientation remains pointing northward.

In the Aerospace / NED coordinate system, another way of describing the same phenomenon is to apply a 180° roll (resulting in the circuit board pointed northward but inverted) followed by a 180° yaw rotation (which has no effect because the accelerometer is insensitive to rotation about the vertical gravity axis). The same orientation can also be reached by a 180° pitch rotation. Without the 180° orientation twist at 90° pitch, the same accelerometer reading would lead to two different orientations depending upon the path taken.

4.2 Aerospace / NED

This section documents the mathematics underlying the function `f3DOFTiltNED` in file *orientation.c*.

The Aerospace / NED orientation matrix $\boldsymbol{R}_{NED}$ for zero yaw angle ψ after rotation from the horizontal by pitch angle θ followed by roll angle ϕ is:

$$\boldsymbol{R}_{NED}(\psi=0)=\begin{pmatrix}\cos\theta & 0 & -\sin\theta\\ \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi\\ \cos\phi\sin\theta & -\sin\phi & \cos\theta\cos\phi\end{pmatrix} \quad \textbf{(19)}$$

The Aerospace / NED accelerometer reading ${}^{S}\boldsymbol{G}_c$ in the sensor frame at this orientation can be computed by multiplying the accelerometer measurement in the initial flat orientation by $\boldsymbol{R}_{NED}(\psi=0)$:

$$\begin{pmatrix}{}^{S}G_{cx}\\ {}^{S}G_{cy}\\ {}^{S}G_{cz}\end{pmatrix}=\begin{pmatrix}\cos\theta & 0 & -\sin\theta\\ \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi\\ \cos\phi\sin\theta & -\sin\phi & \cos\theta\cos\phi\end{pmatrix}\left|{}^{S}\boldsymbol{G}_c\right|\begin{pmatrix}0\\0\\1\end{pmatrix}=\left|{}^{S}\boldsymbol{G}_c\right|\begin{pmatrix}-\sin\theta\\ \sin\phi\cos\theta\\ \cos\phi\cos\theta\end{pmatrix} \quad \textbf{(20)}$$

$$=\left|{}^{S}\boldsymbol{G}_c\right|\begin{pmatrix}-\sin\theta\\ \sin\phi\sqrt{1-\left(\dfrac{{}^{S}G_{cx}}{\left|{}^{S}\boldsymbol{G}_c\right|}\right)^2}\\ \cos\phi\sqrt{1-\left(\dfrac{{}^{S}G_{cx}}{\left|{}^{S}\boldsymbol{G}_c\right|}\right)^2}\end{pmatrix} \quad \textbf{(21)}$$

$$sin\theta = \frac{-\,{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|} \tag{22}$$

$$cos\theta = \sqrt{1 - \left(\frac{{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|}\right)^2} \tag{23}$$

$$sin\phi = \frac{{}^{S}G_{cy}}{|\,{}^{S}\boldsymbol{G}_c|\sqrt{1 - \left(\frac{{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|}\right)^2}} \tag{24}$$

$$cos\phi = \frac{{}^{S}G_{cz}}{|\,{}^{S}\boldsymbol{G}_c|\sqrt{1 - \left(\frac{{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|}\right)^2}} \tag{25}$$

The positive square root for $cos\theta$ can always be taken in equation (23) because the pitch angle θ in the Aerospace / NED coordinate system varies between –90° and +90° giving a nonnegative value of $cos\theta$.

Substituting back into equation (19) gives the Aerospace / NED 3DOF orientation matrix directly in terms of the sensor frame accelerometer reading ${}^{S}\boldsymbol{G}_c$ as:

$$\boldsymbol{R}_{NED}(\psi = 0) = \begin{pmatrix} \frac{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}}{|\,{}^{S}\boldsymbol{G}_c|} & 0 & \frac{{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|} \\ \frac{-\,{}^{S}G_{cx}\,{}^{S}G_{cy}}{|\,{}^{S}\boldsymbol{G}_c|\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cz}}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cy}}{|\,{}^{S}\boldsymbol{G}_c|} \\ \frac{-\,{}^{S}G_{cx}\,{}^{S}G_{cz}}{|\,{}^{S}\boldsymbol{G}_c|\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{-\,{}^{S}G_{cy}}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cz}}{|\,{}^{S}\boldsymbol{G}_c|} \end{pmatrix} \tag{26}$$

$$= \begin{pmatrix} \frac{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}}{|\,{}^{S}\boldsymbol{G}_c|} & 0 & \frac{{}^{S}G_{cx}}{|\,{}^{S}\boldsymbol{G}_c|} \\ \frac{-R_{xz}R_{yz}|\,{}^{S}\boldsymbol{G}_c|}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{R_{zz}|\,{}^{S}\boldsymbol{G}_c|}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cy}}{|\,{}^{S}\boldsymbol{G}_c|} \\ \frac{-R_{xz}R_{zz}|\,{}^{S}\boldsymbol{G}_c|}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{-R_{yz}|\,{}^{S}\boldsymbol{G}_c|}{\sqrt{{}^{S}G_{cy}{}^2 + {}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cz}}{|\,{}^{S}\boldsymbol{G}_c|} \end{pmatrix} \tag{27}$$

At the gimbal lock orientations with ±90° pitch θ rotation, where ${}^S G_{cy} = {}^S G_{cz} = 0$ and ${}^S G_{cx} = |{}^S\boldsymbol{G}_c|$, equations (24) to (27) show that the roll angle ϕ and the rotation matrix $\boldsymbol{R}_{NED}$ are undefined. Since the roll angle ϕ is now about the same rotation axis as the yaw rotation angle ψ which is assumed to be zero, the simplest solution is to also set the roll angle to zero giving:

$$\boldsymbol{R}_{NED}(\phi = \psi = 0) = \begin{pmatrix} 0 & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & sign({}^S G_{cx}) \\ 0 & 1 & 0 \\ -sign({}^S G_{cx}) & 0 & 0 \end{pmatrix} \tag{28}$$

4.3 Android

This section documents the mathematics underlying the function `f3DOFTiltAndroid` in file *orientation.c*.

The Android rotation matrix $\boldsymbol{R}_{Android}$ for zero yaw angle ψ after rotation from the horizontal by roll angle ϕ followed by pitch angle θ is:

$$\boldsymbol{R}_{Android}(\psi = 0) = \begin{pmatrix} \cos\phi & 0 & sin\phi \\ sin\phi sin\theta & \cos\theta & -\cos\phi\, sin\theta \\ -\cos\theta\, sin\phi & sin\theta & \cos\phi\cos\theta \end{pmatrix} \tag{29}$$

The Android accelerometer reading ${}^S\boldsymbol{G}_c$ in the sensor frame at this orientation can be computed by multiplying the accelerometer measurement in the initial flat orientation by $\boldsymbol{R}_{Android}(\psi = 0)$:

$$\begin{pmatrix} {}^S G_{cx} \\ {}^S G_{cy} \\ {}^S G_{cz} \end{pmatrix} = \begin{pmatrix} \cos\phi & 0 & sin\phi \\ sin\phi sin\theta & \cos\theta & -\cos\phi\, sin\theta \\ -\cos\theta\, sin\phi & sin\theta & \cos\phi\cos\theta \end{pmatrix} |{}^S\boldsymbol{G}_c| \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = |{}^S\boldsymbol{G}_c| \begin{pmatrix} sin\phi \\ -cos\phi\sin\theta \\ cos\phi\cos\theta \end{pmatrix} \tag{30}$$

$$= |{}^S\boldsymbol{G}_c| \begin{pmatrix} sin\phi \\ -\sin\theta\sqrt{1 - \left(\frac{{}^S G_{cx}}{|{}^S\boldsymbol{G}_c|}\right)^2} \\ \cos\theta\sqrt{1 - \left(\frac{{}^S G_{cx}}{|{}^S\boldsymbol{G}_c|}\right)^2} \end{pmatrix} \tag{31}$$

$$sin\phi = \frac{{}^S G_{cx}}{|{}^S\boldsymbol{G}_c|} \tag{32}$$

$$cos\phi = \sqrt{1 - \left(\frac{{}^S G_{cx}}{|{}^S\boldsymbol{G}_c|}\right)^2} \tag{33}$$

$$\sin\theta = \frac{-\,{}^{S}G_{cy}}{\left|{}^{S}\boldsymbol{G}_c\right|\sqrt{1-\left(\frac{{}^{S}G_{cx}}{\left|{}^{S}\boldsymbol{G}_c\right|}\right)^2}} \tag{34}$$

$$\cos\theta = \frac{{}^{S}G_{cz}}{\left|{}^{S}\boldsymbol{G}_c\right|\sqrt{1-\left(\frac{{}^{S}G_{cx}}{\left|{}^{S}\boldsymbol{G}_c\right|}\right)^2}} \tag{35}$$

The positive square root for $\cos\phi$ can always be taken in equation (33) because the roll angle ϕ in the Android coordinate system varies between –90° and +90° giving a non-negative value of $\cos\phi$.

Substituting into equation (29) gives the 3DOF Android orientation matrix directly in terms of the accelerometer reading as:

$$\boldsymbol{R}_{Android}(\psi=0) = \frac{1}{\left|{}^{S}\boldsymbol{G}_c\right|}\begin{pmatrix} \sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2} & 0 & {}^{S}G_{cx} \\ \frac{-\,{}^{S}G_{cx}\,{}^{S}G_{cy}}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cz}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & {}^{S}G_{cy} \\ \frac{-\,{}^{S}G_{cx}\,{}^{S}G_{cz}}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{-\,{}^{S}G_{cy}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & {}^{S}G_{cz} \end{pmatrix} \tag{36}$$

$$= \begin{pmatrix} \frac{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}}{\left|{}^{S}\boldsymbol{G}_c\right|} & 0 & \frac{{}^{S}G_{cx}}{\left|{}^{S}\boldsymbol{G}_c\right|} \\ \frac{-R_{xz}R_{yz}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{R_{zz}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cy}}{\left|{}^{S}\boldsymbol{G}_c\right|} \\ \frac{-R_{xz}R_{zz}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{-R_{yz}\left|{}^{S}\boldsymbol{G}_c\right|}{\sqrt{{}^{S}G_{cy}{}^2+{}^{S}G_{cz}{}^2}} & \frac{{}^{S}G_{cz}}{\left|{}^{S}\boldsymbol{G}_c\right|} \end{pmatrix} \tag{37}$$

In the case of gimbal lock after ±90° roll rotation, when ${}^{S}G_{cy} = {}^{S}G_{cz} = 0$, the pitch angle θ is undefined and can be set to zero:

$$\boldsymbol{R}_{Android}(\theta=\psi=0) = \begin{pmatrix} 0 & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & sign({}^{S}G_{cx}) \\ 0 & 1 & 0 \\ -sign({}^{S}G_{cx}) & 0 & 0 \end{pmatrix} \tag{38}$$

4.4 Windows 8

This section documents the mathematics underlying the function `f3DOFTiltWin8` in file *orientation.c*.

The Windows 8 rotation matrix $\boldsymbol{R}_{Win8}$ for zero yaw angle ψ, after rotation from the horizontal by pitch angle θ followed by roll angle ϕ, is:

$$\boldsymbol{R}_{Win8}(\psi = 0) = \begin{pmatrix} \cos\phi & \sin\phi\sin\theta & -\sin\phi\cos\theta \\ 0 & \cos\theta & \sin\theta \\ \sin\phi & -\cos\phi\sin\theta & \cos\phi\cos\theta \end{pmatrix} \quad \textbf{(39)}$$

The Windows 8 accelerometer reading ${}^{S}\boldsymbol{G}_c$ in the sensor frame at this orientation can be computed by multiplying the accelerometer measurement in the initial flat orientation by $\boldsymbol{R}_{Win8}(\psi = 0)$:

$$\begin{pmatrix} {}^{S}G_{cx} \\ {}^{S}G_{cy} \\ {}^{S}G_{cz} \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi\sin\theta & -\sin\phi\cos\theta \\ 0 & \cos\theta & \sin\theta \\ \sin\phi & -\cos\phi\sin\theta & \cos\phi\cos\theta \end{pmatrix} |{}^{S}\boldsymbol{G}_c| \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = |{}^{S}\boldsymbol{G}_c| \begin{pmatrix} \sin\phi\cos\theta \\ -\sin\theta \\ -\cos\phi\cos\theta \end{pmatrix} \quad \textbf{(40)}$$

$$\tan\phi = \frac{-{}^{S}G_{cx}}{{}^{S}G_{cz}} \Rightarrow \cos\phi = \frac{1}{\sqrt{1 + \left(\frac{{}^{S}G_{cx}}{{}^{S}G_{cz}}\right)^2}} \quad \textbf{(41)}$$

$$\Rightarrow \sin\phi = \tan\phi\cos\phi = \frac{-{}^{S}G_{cx}}{{}^{S}G_{cz}\sqrt{1 + \left(\frac{{}^{S}G_{cx}}{{}^{S}G_{cz}}\right)^2}} \quad \textbf{(42)}$$

$$\sin\theta = \frac{-{}^{S}G_{cy}}{|{}^{S}\boldsymbol{G}_c|} \quad \textbf{(43)}$$

$$\cos\theta = \frac{-{}^{S}G_{cz}}{|{}^{S}\boldsymbol{G}_c|\cos\phi} = \frac{-{}^{S}G_{cz}}{|{}^{S}\boldsymbol{G}_c|}\sqrt{1 + \left(\frac{{}^{S}G_{cx}}{{}^{S}G_{cz}}\right)^2} \quad \textbf{(44)}$$

The positive square root for $\cos\phi$ can always be taken in equation (41) because the roll angle ϕ in the Windows 8 coordinate system varies between –90° and +90° resulting in a nonnegative value of $\cos\phi$.

Substituting into equation (39) gives the Windows 8 3DOF orientation matrix directly in terms of the accelerometer reading as:

$$\boldsymbol{R}_{Win8}(\psi = 0) = \frac{1}{\left| {}^{S}\boldsymbol{G}_c \right|} \begin{pmatrix} \dfrac{\left| {}^{S}\boldsymbol{G}_c \right|}{\sqrt{1 + \left(\dfrac{{}^{S}G_{cx}}{{}^{S}G_{cz}} \right)^2}} & \dfrac{{}^{S}G_{cx}\ {}^{S}G_{cy}}{{}^{S}G_{cz} \sqrt{1 + \left(\dfrac{{}^{S}G_{cx}}{{}^{S}G_{cz}} \right)^2}} & -{}^{S}G_{cx} \\ 0 & -{}^{S}G_{cz} \sqrt{1 + \left(\dfrac{{}^{S}G_{cx}}{{}^{S}G_{cz}} \right)^2} & -{}^{S}G_{cy} \\ \dfrac{-{}^{S}G_{cx} \left| {}^{S}\boldsymbol{G}_c \right|}{{}^{S}G_{cz} \sqrt{1 + \left(\dfrac{{}^{S}G_{cx}}{{}^{S}G_{cz}} \right)^2}} & \dfrac{{}^{S}G_{cy}}{\sqrt{1 + \left(\dfrac{{}^{S}G_{cx}}{{}^{S}G_{cz}} \right)^2}} & -{}^{S}G_{cz} \end{pmatrix} \tag{45}$$

$$= \begin{pmatrix} \dfrac{{}^{S}G_{cz}}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cx} R_{yz}}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cx}}{\left| {}^{S}\boldsymbol{G}_c \right|} \\ 0 & \dfrac{-sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}}{\left| {}^{S}\boldsymbol{G}_c \right|} & \dfrac{-{}^{S}G_{cy}}{\left| {}^{S}\boldsymbol{G}_c \right|} \\ \dfrac{-{}^{S}G_{cx}}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cy} R_{zz}}{sign(G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cz}}{\left| {}^{S}\boldsymbol{G}_c \right|} \end{pmatrix} \tag{46}$$

$$= \begin{pmatrix} \dfrac{-R_{zz} \left| {}^{S}\boldsymbol{G}_c \right|}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{R_{xz} R_{yz} \left| {}^{S}\boldsymbol{G}_c \right|}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cx}}{\left| {}^{S}\boldsymbol{G}_c \right|} \\ 0 & \dfrac{-1}{\left\{ \dfrac{\left| {}^{S}\boldsymbol{G}_c \right|}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} \right\}} & \dfrac{-{}^{S}G_{cy}}{\left| {}^{S}\boldsymbol{G}_c \right|} \\ \dfrac{R_{xz} \left| {}^{S}\boldsymbol{G}_c \right|}{sign(G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{R_{yz} R_{zz} \left| {}^{S}\boldsymbol{G}_c \right|}{sign({}^{S}G_{cz}) \sqrt{{}^{S}G_{cx}{}^2 + {}^{S}G_{cz}{}^2}} & \dfrac{-{}^{S}G_{cz}}{\left| {}^{S}\boldsymbol{G}_c \right|} \end{pmatrix} \tag{47}$$

In the case of gimbal lock after ±90° pitch rotation, when ${}^{S}G_{cx} = {}^{S}G_{cz} = 0$, the roll angle ϕ is undefined and can be set to zero:

$$\boldsymbol{R}_{Win8}(\phi = \psi = 0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & sin\theta \\ 0 & -sin\theta & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -sign({}^{S}G_{cy}) \\ 0 & sign({}^{S}G_{cy}) & 0 \end{pmatrix} \tag{48}$$

5. Magnetometer 2D Compass Orientation Matrix

5.1 Introduction

This section documents the 2D, magnetometer-only, eCompass functions that calculate the orientation matrix from calibrated magnetometer measurements with the assumption that the magnetometer is always flat and has zero tilt away from the horizontal. The market application for this arrangement is an automotive compass where the normal accelerometer plus magnetometer tilt-compensated eCompass functions cannot be used because of vehicle acceleration, braking and cornering forces.

Use these functions with caution. In the automotive application, the 2D eCompass will be robust against the effects of acceleration, braking and cornering forces because the magnetometer sensor is unaffected by acceleration. However, the compass heading will change when the vehicle tilt angle, whether roll or pitch, deviates from horizontal, violating the assumption that the magnetometer sensor is always flat.

5.2 Aerospace / NED

This section documents the mathematics underlying the function `f3DOFMagnetometerMatrixNED` in file *orientation.c*.

The Aerospace / NED rotation matrix for rotation in yaw angle ψ only while remaining horizontal is:

$$\boldsymbol{R}_{NED}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{49}$$

The Aerospace / NED calibrated magnetometer reading at this orientation is:

$$\begin{pmatrix} {}^SB_{cx} \\ {}^SB_{cy} \\ {}^SB_{cz} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} B \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} = B \begin{pmatrix} \cos\psi\cos\delta \\ -\sin\psi\cos\delta \\ \sin\delta \end{pmatrix} \tag{50}$$

The horizontal component of the geomagnetic field has the value:

$$B\cos\delta = \sqrt{{}^SB_{cx}{}^2 + {}^SB_{cy}{}^2} \tag{51}$$

The yaw ψ (and compass heading angle) is given by:

$$\tan\psi = \frac{\sin\psi}{\cos\psi} = \frac{-{}^SB_{cy}}{{}^SB_{cx}} \tag{52}$$

$$\sin\psi = \frac{-\,{}^{S}B_{cy}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{53}$$

$$\cos\psi = \frac{{}^{S}B_{cx}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{54}$$

The orientation matrix is independent of the unknown geomagnetic inclination angle δ and has the value:

$$\boldsymbol{R}_{NED}(\psi) = \frac{1}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}}\begin{pmatrix} {}^{S}B_{cx} & -\,{}^{S}B_{cy} & 0 \\ {}^{S}B_{cy} & {}^{S}B_{cx} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{55}$$

5.3 Android

This section documents the mathematics underlying the function `f3DOFMagnetometerMatrixAndroid` in file *orientation.c*.

The Android rotation matrix for rotation in yaw angle ψ only while remaining horizontal is:

$$\boldsymbol{R}_{Android}(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{56}$$

The Android calibrated magnetometer reading at this orientation is:

$$\begin{pmatrix} {}^{S}B_{cx} \\ {}^{S}B_{cy} \\ {}^{S}B_{cz} \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} B \begin{pmatrix} 0 \\ \cos\delta \\ -\sin\delta \end{pmatrix} = B \begin{pmatrix} -\sin\psi\cos\delta \\ \cos\psi\cos\delta \\ -\sin\delta \end{pmatrix} \tag{57}$$

The horizontal component of the geomagnetic field has the value:

$$B\cos\delta = \sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}} \tag{58}$$

The yaw ψ and compass heading angles are given by:

$$\tan\psi = \frac{\sin\psi}{\cos\psi} = \frac{-\,{}^{S}B_{cx}}{{}^{S}B_{cy}} \tag{59}$$

$$sin\psi = \frac{-\,{}^{S}B_{cx}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{60}$$

$$cos\psi = \frac{{}^{S}B_{cy}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{61}$$

The orientation matrix is independent of the unknown geomagnetic inclination angle δ and has the value:

$$\boldsymbol{R}_{Android}(\psi) = \frac{1}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}}\begin{pmatrix} {}^{S}B_{cy} & {}^{S}B_{cx} & 0 \\ -\,{}^{S}B_{cx} & {}^{S}B_{cy} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{62}$$

5.4 Windows 8

This section documents the mathematics underlying the function `f3DOFMagnetometerMatrixWin8` in file *orientation.c.*

The Windows 8 rotation matrix for rotation in yaw angle ψ only while remaining horizontal is:

$$\boldsymbol{R}_{Win8}(\psi) = \begin{pmatrix} \cos\psi & sin\psi & 0 \\ -\,sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{63}$$

The Windows 8 calibrated magnetometer reading at this orientation is:

$$\begin{pmatrix} {}^{S}B_{cx} \\ {}^{S}B_{cy} \\ {}^{S}B_{cz} \end{pmatrix} = \begin{pmatrix} \cos\psi & sin\psi & 0 \\ -\,sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} B \begin{pmatrix} 0 \\ cos\delta \\ -sin\delta \end{pmatrix} = B \begin{pmatrix} sin\psi cos\delta \\ cos\psi cos\delta \\ -sin\delta \end{pmatrix} \tag{64}$$

The horizontal component of the geomagnetic field has the value:

$$\boldsymbol{Bcos\delta} = \sqrt{{}^{S}\boldsymbol{B}_{cx}{}^{2} + {}^{S}\boldsymbol{B}_{cy}{}^{2}} \tag{65}$$

The yaw ψ angle, which is equal to minus the compass heading in the Windows 8 coordinate system, is given by:

$$tan\psi = \frac{sin\psi}{cos\psi} = \frac{{}^{S}B_{cx}}{{}^{S}B_{cy}} \tag{66}$$

$$sin\psi = \frac{{}^{S}B_{cx}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{67}$$

$$cos\psi = \frac{{}^{S}B_{cy}}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \tag{68}$$

The orientation matrix is independent of the unknown geomagnetic inclination angle δ and has the value:

$$\boldsymbol{R}_{Win8}(\psi) = \frac{1}{\sqrt{{}^{S}B_{cx}{}^{2} + {}^{S}B_{cy}{}^{2}}} \begin{pmatrix} {}^{S}B_{cy} & {}^{S}B_{cx} & 0 \\ -{}^{S}B_{cx} & {}^{S}B_{cy} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{69}$$

6. Accelerometer and Magnetometer 3D eCompass Orientation Matrix

6.1 Introduction

This section documents the calculation of the orientation matrix from accelerometer and magnetometer measurements. This is the classic tilt-compensated eCompass which provides a complete determination of orientation. The assumption is that there is no linear acceleration so that the accelerometer reading is a function of orientation in the earth's gravitational field only.

The orientation matrix can be calculated with an algorithm which is most easily understood from the trivial mathematical identity below:

$$\begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{70}$$

The three columns of the identity matrix on the right hand side are the three unit vectors in the *x*, *y* and *z* directions aligned with the initial orientation in the global reference frame. In the Aerospace/NED coordinate system, the *x* axis points North, the *y* axis points East and the *z* axis points Down. In the Android and Windows 8 coordinate systems, the *x* axis points East, the *y* axis points North and the *z* axis points Up

Separating equation (70) into column vectors clearly shows that the *x*, *y*, and *z* components of the orientation matrix are the original *x*, *y*, and *z* unit vectors rotated from the global reference frame to the sensor frame:

$$\begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{71}$$

$$\begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \textbf{(72)}$$

$$\begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \textbf{(73)}$$

In the absence of acceleration, the accelerometer measures the gravity vector measured in the sensor frame and, therefore, provides the right-hand column of the orientation matrix. The geomagnetic vector in the global frame points northward and downward (upward) in the northern (southern) hemisphere. The vector product between the rotated gravity and rotated geomagnetic vector is, by definition, at right angles to both and is, therefore, the rotated easterly-pointing vector from the global frame which forms the *y* (Aerospace / NED coordinate system) or *x* (Android and Windows 8 coordinate systems) column vector in the orientation matrix. Because the orientation matrix is orthonormal, a final vector product between these two column vectors gives the remaining column vector of the orientation matrix.

6.2 Aerospace / NED

This section documents the mathematics underlying the function `feCompassNED` in file *orientation.c*.

The *z* axis in the initial NED orientation is aligned downwards and parallel to gravity. The *z* axis column vector of the orientation matrix is the rotated *z* axis vector which, on the assumption of zero acceleration, is the normalized accelerometer measurement ${}^{S}\boldsymbol{G}_c$:

$$\begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} = \frac{{}^{S}\boldsymbol{G}_c}{\left| {}^{S}\boldsymbol{G}_c \right|} \quad \textbf{(74)}$$

The geomagnetic vector, estimated from the calibrated magnetometer measurement, points northwards in the initial orientation (in the *x* direction for the NED coordinate system) and downward (upward) in the northern (southern) hemisphere. The normalized vector product of the accelerometer and magnetometer measurements gives a normalized vector orthogonal to both which is, therefore, aligned along the east pointing *y* axis and equal to the *y* column of the rotation matrix:

$$\begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} = \frac{{}^{S}\boldsymbol{G}_c \times {}^{S}\boldsymbol{B}_c}{\left| {}^{S}\boldsymbol{G}_c \times {}^{S}\boldsymbol{B}_c \right|} \quad \textbf{(75)}$$

Finally, since rotation matrices are column (and row) orthonormal, the vector product of the *y* and *z* columns of the rotation matrix columns equals the remaining *x* column

$$\begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} = \begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} \times \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \quad \textbf{(76)}$$

The geomagnetic inclination angle δ is the angle by which the geomagnetic field dips below horizontal in the global frame. Because the scalar product of two vectors is invariant under rotation, the inclination angle can be computed from the scalar product of the normalized accelerometer and calibrated magnetic measurements in the sensor frame giving:

$${}^S\boldsymbol{G}_c \cdot {}^S\boldsymbol{B}_c = |{}^S\boldsymbol{G}_c||{}^S\boldsymbol{B}_c| sin\delta \Rightarrow sin\delta = \frac{{}^SG_{cx}\,{}^SB_{cx} + {}^SG_{cy}\,{}^SB_{cy} + {}^SG_{cz}\,{}^SB_{cz}}{|{}^S\boldsymbol{G}_c||{}^S\boldsymbol{B}_c|} \quad \textbf{(77)}$$

The orientation matrix resulting from applying equations (74) to (76) can be shown to equal the Aerospace / NED orientation matrix derived in equation (5) of AN5017 “*Aerospace, Android and Windows Coordinate Systems*”:

$$\boldsymbol{R}_{NED} = \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\, sin\psi & -sin\theta \\ \cos\psi\, sin\theta sin\phi - \cos\phi\, sin\psi & \cos\phi\cos\psi + sin\theta sin\phi sin\psi & \cos\theta\, sin\phi \\ cos\phi\cos\psi\, sin\theta + sin\phi sin\psi & \cos\phi\, sin\theta sin\psi - \cos\psi\, sin\phi & \cos\theta\, cos\phi \end{pmatrix} \quad \textbf{(78)}$$

The proof is simplified by using the standard result that the vector product of any two rotated vectors equals their rotated vector product:

$$\boldsymbol{R}(\boldsymbol{a} \times \boldsymbol{b}) = \boldsymbol{Ra} \times \boldsymbol{Rb} \quad \textbf{(79)}$$

Equation (74) evaluates to:

$$\frac{{}^S\boldsymbol{G}_c}{|{}^S\boldsymbol{G}_c|} = \boldsymbol{R}_{NED}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -sin\theta \\ \cos\theta\, sin\phi \\ \cos\theta\, cos\phi \end{pmatrix} = \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \quad \textbf{(80)}$$

Equation (75) evaluates to:

$$\frac{{}^S\boldsymbol{G}_c \times {}^S\boldsymbol{B}_c}{|{}^S\boldsymbol{G}_c \times {}^S\boldsymbol{B}_c|} = \frac{\boldsymbol{R}_{NED}\left\{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} cos\delta \\ 0 \\ sin\delta \end{pmatrix}\right\}}{\left|\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} cos\delta \\ 0 \\ sin\delta \end{pmatrix}\right|} = \boldsymbol{R}_{NED}\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\theta\, sin\psi \\ \cos\phi\cos\psi + sin\theta sin\phi sin\psi \\ \cos\phi\, sin\theta sin\psi - \cos\psi\, sin\phi \end{pmatrix} = \begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} \quad \textbf{(81)}$$

Equation (76) evaluates to:

$$\begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} \times \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} = \begin{pmatrix} \cos\theta\, sin\psi \\ \cos\phi\cos\psi + sin\theta sin\phi sin\psi \\ \cos\phi\, sin\theta sin\psi - \cos\psi\, sin\phi \end{pmatrix} \times \begin{pmatrix} -sin\theta \\ \cos\theta\, sin\phi \\ \cos\theta\, cos\phi \end{pmatrix} \tag{82}$$

$$= \begin{pmatrix} (\cos\phi\cos\psi + sin\theta sin\phi sin\psi)\cos\theta\, cos\phi - (\cos\phi\, sin\theta sin\psi - \cos\psi\, sin\phi)\cos\theta\, sin\phi \\ -(\cos\phi\, sin\theta sin\psi - \cos\psi\, sin\phi) sin\theta - \cos^2\theta\, sin\psi cos\phi \\ sin\psi\cos^2\theta\, sin\phi + (\cos\phi\cos\psi + sin\theta sin\phi sin\psi) sin\theta \end{pmatrix} \tag{83}$$

$$= \begin{pmatrix} \cos\theta\cos\psi \\ \cos\psi\, sin\theta sin\phi - \cos\phi\, sin\psi \\ cos\phi\cos\psi\, sin\theta + sin\phi sin\psi \end{pmatrix} = \begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} \tag{84}$$

6.3 Android

This section documents the mathematics underlying the function `feCompassAndroid` in file *orientation.c*.

The *z* axis in the initial Android orientation is aligned upwards and anti-parallel to gravity. The Android standard specifies that the accelerometer is acceleration positive and not gravity positive. The *z* axis column vector of the orientation matrix is therefore the rotated z axis vector which, on the assumption of zero acceleration, is the normalized accelerometer measurement $\boldsymbol{G}_s$:

$$\begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} = \frac{{}^{S}\boldsymbol{G}_c}{|{}^{S}\boldsymbol{G}_c|} \tag{85}$$

The geomagnetic vector, estimated from the calibrated magnetometer measurement, points northwards in the initial orientation (in the *y* direction for the Android coordinate system) and downward (upward) in the northern (southern) hemisphere. The normalized vector product of the magnetometer and accelerometer measurements gives a normalized vector orthogonal to both which is, therefore, aligned along the east pointing *x* axis:

$$\begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} = \frac{{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c}{|{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c|} \tag{86}$$

Finally, since rotation matrices are orthonormal, the vector product of the rotation matrix *z* and *x* axes gives the remaining *y* component of the orientation matrix:

$$\begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} = \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \times \begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} \tag{87}$$

The inclination angle δ is computed from the scalar product of the accelerometer and calibrated magnetic measurements to give:

$$ {}^{S}\boldsymbol{G}_c \cdot {}^{S}\boldsymbol{B}_c = -|{}^{S}\boldsymbol{G}_c||{}^{S}\boldsymbol{B}_c| sin\delta \Rightarrow sin\delta = \frac{-({}^{S}G_{cx}\,{}^{S}B_{cx} + {}^{S}G_{cy}\,{}^{S}B_{cy} + {}^{S}G_{cz}\,{}^{S}B_{cz})}{|{}^{S}\boldsymbol{G}_c||{}^{S}\boldsymbol{B}_c|} \quad (88)$$

The orientation matrix resulting from applying equations (85) to (87) can be shown to equal the Android orientation matrix derived in equation (30) of AN5017 "*Aerospace, Android and Windows Coordinate Systems*":

$$\boldsymbol{R}_{Android} = \begin{pmatrix} \cos\phi\cos\psi & -cos\phi sin\psi & sin\phi \\ \cos\theta\, sin\psi + \cos\psi\, sin\phi sin\theta & \cos\psi\cos\theta - sin\phi sin\psi sin\theta & -\cos\phi\, sin\theta \\ -\cos\psi\cos\theta\, sin\phi + sin\psi sin\theta & \cos\theta\, sin\phi sin\psi + \cos\psi\, sin\theta & \cos\phi\cos\theta \end{pmatrix} \quad (89)$$

Equation (85) evaluates to:

$$\frac{{}^{S}\boldsymbol{G}_c}{|{}^{S}\boldsymbol{G}_c|} = \boldsymbol{R}_{Android}\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix} sin\phi \\ -\cos\phi\, sin\theta \\ \cos\phi\cos\theta \end{pmatrix} = \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \quad (90)$$

Equation (86) evaluates to:

$$\frac{{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c}{|{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c|} = \frac{\boldsymbol{R}_{Android}\left\{\begin{pmatrix}0\\cos\delta\\-sin\delta\end{pmatrix} \times \begin{pmatrix}0\\0\\1\end{pmatrix}\right\}}{\left|\begin{pmatrix}0\\cos\delta\\-sin\delta\end{pmatrix} \times \begin{pmatrix}0\\0\\1\end{pmatrix}\right|} = \boldsymbol{R}_{Android}\begin{pmatrix}1\\0\\0\end{pmatrix} \quad (91)$$

$$= \begin{pmatrix} \cos\phi\cos\psi \\ \cos\theta\, sin\psi + \cos\psi\, sin\phi sin\theta \\ -\cos\psi\cos\theta\, sin\phi + sin\psi sin\theta \end{pmatrix} = \begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} \quad (92)$$

Equation (87) evaluates to:

$$\begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \times \begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} = \begin{pmatrix} sin\phi \\ -\cos\phi\, sin\theta \\ \cos\phi\cos\theta \end{pmatrix} \times \begin{pmatrix} \cos\phi\cos\psi \\ \cos\theta\, sin\psi + \cos\psi\, sin\phi sin\theta \\ -\cos\psi\cos\theta\, sin\phi + sin\psi sin\theta \end{pmatrix} \quad (93)$$

$$= \begin{pmatrix} -\cos\phi\, sin\theta(-\cos\psi\cos\theta\, sin\phi + sin\psi sin\theta) - \cos\phi\cos\theta\,(\cos\theta\, sin\psi + \cos\psi\, sin\phi sin\theta) \\ \cos\phi\cos\theta\cos\phi\cos\psi - sin\phi(-\cos\psi\cos\theta\, sin\phi + sin\psi sin\theta) \\ sin\phi(\cos\theta\, sin\psi + \cos\psi\, sin\phi sin\theta) + \cos\phi\, sin\theta\cos\phi\cos\psi \end{pmatrix} \quad (94)$$

$$= \begin{pmatrix} -cos\phi sin\psi \\ \cos\psi\cos\theta - sin\phi sin\psi sin\theta \\ \cos\theta\, sin\phi sin\psi + \cos\psi\, sin\theta \end{pmatrix} = \begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} \quad (95)$$

6.4 Windows 8

This section documents the mathematics underlying the function `feCompassWin8` in file *orientation.c.*

The accelerometer reading in the Windows 8 coordinate system in the initial orientation is anti-parallel to the *z* axis because the Windows 8 coordinate system is ENU with the *z* axis upward and is gravity (not acceleration) positive. The *z* axis column vector of the orientation matrix is therefore the rotated z axis vector which, on the assumption of zero acceleration, is the normalized accelerometer measurement $\boldsymbol{G}_s$:

$$\begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} = \frac{-\,{}^{S}\boldsymbol{G}_c}{|\,{}^{S}\boldsymbol{G}_c|} \tag{96}$$

The geomagnetic vector, estimated from the calibrated magnetometer measurement, points northwards in the initial orientation (in the *y* direction for the Windows 8 coordinate system) and downward (upward) in the northern (southern) hemisphere. The normalized vector product of the magnetometer and negated accelerometer measurements gives a normalized vector orthogonal to both which is, therefore, aligned along the east pointing *x* axis:

$$\begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} = \frac{-\,{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c}{|\,{}^{S}\boldsymbol{B}_c \times {}^{S}\boldsymbol{G}_c|} \tag{97}$$

Finally, since rotation matrices are orthonormal, the vector product of the rotation matrix *z* and *x* axes gives the remaining *y* component of the orientation matrix:

$$\begin{pmatrix} R_{xy} \\ R_{yy} \\ R_{zy} \end{pmatrix} = \begin{pmatrix} R_{xz} \\ R_{yz} \\ R_{zz} \end{pmatrix} \times \begin{pmatrix} R_{xx} \\ R_{yx} \\ R_{zx} \end{pmatrix} \tag{98}$$

The inclination angle δ is computed from the scalar product of the accelerometer and calibrated magnetic measurements giving:

$${}^{S}\boldsymbol{G}_c \cdot {}^{S}\boldsymbol{B}_c = |\,{}^{S}\boldsymbol{G}_c||\,{}^{S}\boldsymbol{B}_c| sin\delta \Rightarrow sin\delta = \frac{{}^{S}G_{cx}\,{}^{S}B_{cx} + {}^{S}G_{cy}\,{}^{S}B_{cy} + {}^{S}G_{cz}\,{}^{S}B_{cz}}{|\,{}^{S}\boldsymbol{G}_c||\,{}^{S}\boldsymbol{B}_c|} \tag{99}$$

The orientation matrix resulting from applying equations (96) to (98) can be shown to equal the Windows 8 orientation matrix derived in equation (55) of AN5017 "*Aerospace, Android and Windows Coordinate Systems*":

$$\boldsymbol{R}_{Win8} = \begin{pmatrix} \cos\phi\cos\psi - sin\phi sin\theta sin\psi & cos\phi sin\psi + \cos\psi\, sin\phi sin\theta & -sin\phi\cos\theta \\ -\cos\theta\, sin\psi & \cos\theta\cos\psi & sin\theta \\ \cos\psi\, sin\phi + \cos\phi\, sin\psi sin\theta & sin\phi sin\psi - \cos\phi\cos\psi\, sin\theta & \cos\phi\cos\theta \end{pmatrix} \tag{100}$$

Equation (96) evaluates to:

$$\frac{-\,{}^{S}\boldsymbol{G}_{c}}{\left|\,{}^{S}\boldsymbol{G}_{c}\right|} = -\boldsymbol{R}_{Win8}\begin{pmatrix}0\\0\\-1\end{pmatrix} = \begin{pmatrix}-sin\phi\cos\theta\\ sin\theta\\ \cos\phi\cos\theta\end{pmatrix} = \begin{pmatrix}R_{xz}\\R_{yz}\\R_{zz}\end{pmatrix} \tag{101}$$

Equation (97) evaluates to:

$$\frac{-\,{}^{S}\boldsymbol{B}_{c}\times{}^{S}\boldsymbol{G}_{c}}{\left|\,{}^{S}\boldsymbol{B}_{c}\times{}^{S}\boldsymbol{G}_{c}\right|} = \frac{-\boldsymbol{R}_{Win8}\left\{\begin{pmatrix}0\\cos\delta\\-sin\delta\end{pmatrix}\times\begin{pmatrix}0\\0\\-1\end{pmatrix}\right\}}{\left|\begin{pmatrix}0\\cos\delta\\-sin\delta\end{pmatrix}\times\begin{pmatrix}0\\0\\-1\end{pmatrix}\right|} = \boldsymbol{R}_{Win8}\begin{pmatrix}1\\0\\0\end{pmatrix} \tag{102}$$

$$= \begin{pmatrix}\cos\phi\cos\psi - sin\phi sin\theta sin\psi\\ -\cos\theta\, sin\psi\\ \cos\psi\, sin\phi + \cos\phi\, sin\psi sin\theta\end{pmatrix} = \begin{pmatrix}R_{xx}\\R_{yx}\\R_{zx}\end{pmatrix} \tag{103}$$

Equation (98) evaluates to:

$$\begin{pmatrix}R_{xz}\\R_{yz}\\R_{zz}\end{pmatrix}\times\begin{pmatrix}R_{xx}\\R_{yx}\\R_{zx}\end{pmatrix} = \begin{pmatrix}-sin\phi\cos\theta\\ sin\theta\\ \cos\phi\cos\theta\end{pmatrix}\times\begin{pmatrix}\cos\phi\cos\psi - sin\phi sin\theta sin\psi\\ -\cos\theta\, sin\psi\\ \cos\psi\, sin\phi + \cos\phi\, sin\psi sin\theta\end{pmatrix} \tag{104}$$

$$= \left(\begin{matrix}sin\theta(\cos\psi\, sin\phi + \cos\phi\, sin\psi sin\theta) + \cos\phi\cos\theta\cos\theta\, sin\psi\\ \cos\phi\cos\theta\,(\cos\phi\cos\psi - sin\phi sin\theta sin\psi) + sin\phi\cos\theta\,(\cos\psi\, sin\phi + \cos\phi\, sin\psi\\ sin\phi\cos\theta\cos\theta\, sin\psi - sin\theta(\cos\phi\cos\psi - sin\phi sin\theta sin\psi)\end{matrix}\right. \tag{105}$$

$$= \begin{pmatrix}cos\phi sin\psi + \cos\psi\, sin\phi sin\theta\\ \cos\theta\cos\psi\\ sin\phi sin\psi - \cos\phi\cos\psi\, sin\theta\end{pmatrix} = \begin{pmatrix}R_{xy}\\R_{yy}\\R_{zy}\end{pmatrix} \tag{106}$$

7. Legal information

7.1 Definitions

7.2 Disclaimers

7.3 Trademarks

8. Contents

1. Introduction 3
1.1 Summary 3
1.2 Terminology 3
1.3 Software Functions 4
2. General Rotation Matrix 4
2.1 Introduction 4
2.2 Eigenvector and Eigenvalue 5
3. Converting Between Rotation Matrix and Rotation Vector 5
3.1 Rotation Vector from Rotation Matrix 5
3.2 Rotation Matrix from Rotation Vector 7
4. Accelerometer Tilt Orientation Matrix 8
4.1 Introduction 8
4.2 Aerospace / NED 8
4.3 Android 10
4.4 Windows 8 12
5. Magnetometer 2D Compass Orientation Matrix 14
5.1 Introduction 14
5.2 Aerospace / NED 14
5.3 Android 15
5.4 Windows 8 16
6. Accelerometer and Magnetometer 3D eCompass Orientation Matrix 17
6.1 Introduction 17
6.2 Aerospace / NED 18
6.3 Android 20
6.4 Windows 8 22
7. Legal information 24
7.1 Definitions 24
7.2 Disclaimers 24
7.3 Trademarks 24
8. Contents 25

Date of release: 21 June 2016
Document identifier: AN5021