

1 Introduction

The purpose of this code is to normalize a matrix that should represent a rotation matrix for the AeroQuad. This matrix should be normalized and each vector should be orthogonal.

2 Original DCM code

The original DCM code defines a matrix $\overline{\mathbf{M}}$ as group of row vector \vec{x}, \vec{y} , and \vec{z} . The correction begins by calculating an error ϵ as

$$\epsilon = \frac{1}{2} \vec{x} \cdot \vec{y}, \quad (1)$$

then calculates a correction to \vec{x} and \vec{y} as

$$\vec{x} = \vec{x} + \epsilon \vec{y}, \quad (2)$$

$$\vec{y} = \vec{y} + \epsilon \vec{x}. \quad (3)$$

Following this correction, the \vec{x} and \vec{y} should be orthogonal. The \vec{z} vector is then defined by the cross product

$$\vec{z} = \vec{x} \times \vec{y}. \quad (4)$$

Then each vector is normalized with the assumption that the vectors are close to unity, allowing a Taylor expansion to be used.

$$\vec{v} = 0.5 * (3 - \vec{v} \cdot \vec{v}) \vec{v}, \quad (5)$$

where \vec{v} is x, y or z .

3 Square root normalization

The Taylor expansion above fails if the matrix is not close to a valid rotation matrix (if the magnitude of each component vector is not close to unity and the components are not orthogonal). This happens when the rotation rate is large and results in the rotation matrix failing during fast turns of the AeroQuad. When this happens, the rotation matrix output from the DCM normalization is neither normalized nor a valid rotation matrix. Typically, the \vec{x} and \vec{y} are not orthogonal, and the resulting \vec{z} is not orthogonal to the xy plane, nor close to unit length. As a result, successive iterations of the normalize lead to instability as the DCM diverges *unless* the rotation was transient and the AeroQuad returns to a relatively stable state (allowing the DCM Integral term to correct the error).

An easy correction to this problem is to normalize the component vectors without approximation, using the true magnitude calculation

$$|v| = \sqrt{\sum_i v_i^2}, \quad (6)$$

$$\hat{v} = \frac{\vec{v}}{\sqrt{\sum_i v_i^2}}. \quad (7)$$

If this calculation is performed on each vector after \vec{z} is calculated, the DCM is always normalized and much more stable under fast rotation.

In principle, \vec{x} and \vec{y} should be orthogonal after the error correction stage, which means that one square-root can be saved by normalizing \vec{x} and \vec{y} before calculating \vec{z} , which is then normalized by definition

$$|\vec{z}| = |\vec{x} \times \vec{y}| = |\vec{x}| |\vec{y}| \sin \theta, \quad (8)$$

if \vec{x} and \vec{y} are orthogonal and normalized then $\sin \theta = 1$, $|\vec{x}| = 1$, $|\vec{y}| = 1$. However, \vec{x} and \vec{y} are not guaranteed to be orthogonal after the error correction, so the “two square root” method is not as stable as the “three square root” method above.

4 Alternate normalizing method

The desired output of the normalization (and the DCM) is a valid rotation matrix $\bar{\mathbf{M}}$ that represents the orientation of the AeroQuad in space, constructed from the error-prone input matrix. To construct this rotation matrix we determine the rotation required to put the xy plane into the ij plane (the true xy plane of inertial space), and the rotation around the \hat{k} vector required to rotate the \vec{x} vector onto the \hat{i} vector.

First we calculate the vector normal to the xy plane by calculating the cross product and normalizing the result.

$$\hat{z} = \widehat{\vec{x} \times \vec{y}}. \quad (9)$$

Then we calculate the angle of the \hat{z} vector from \hat{k} by taking the dot product (both are normalized, so the dot product is cosine of the angle)

$$\cos \theta = \hat{z} \cdot \hat{k}, \quad (10)$$

$$\sin \theta = \sqrt{1 - \cos^2 \theta}. \quad (11)$$

θ is the angle we need to rotate the \hat{z} vector around the vector $\hat{z} \times \hat{k}$ so that \hat{z} lines up with \hat{k} axis. Given the angle of rotation and the axis $\vec{u} = (u_x, u_y, u_z)$ the rotation matrix R is

$$\begin{pmatrix} \cos \theta + u_x^2(1 - \cos \theta) & u_x u_y(1 - \cos \theta) - u_z \sin \theta & u_x u_z(1 - \cos \theta) + u_y \sin \theta \\ u_y u_x(1 - \cos \theta) + u_z \sin \theta & \cos \theta + u_y^2(1 - \cos \theta) & u_y u_z(1 - \cos \theta) - u_x \sin \theta \\ u_z u_x(1 - \cos \theta) - u_y \sin \theta & u_z u_y(1 - \cos \theta) + u_x \sin \theta & \cos \theta + u_z^2(1 - \cos \theta) \end{pmatrix}. \quad (12)$$

For the particular rotation matrix $\overline{\mathbf{R}}_\theta$ of an angle θ around axis u given by

$$\hat{u} = \hat{z} \times \hat{k}, \quad (13)$$

$$= (z_y, -z_x, 0), \quad (14)$$

$$(15)$$

the rotation matrix becomes

$$\overline{\mathbf{R}}_\theta = \begin{pmatrix} \cos \theta + z_y^2(1 - \cos \theta) & -z_y z_x(1 - \cos \theta) & -z_x \sin \theta \\ -z_x z_y(1 - \cos \theta) & \cos \theta + z_x^2(1 - \cos \theta) & -z_y \sin \theta \\ z_x \sin \theta & z_y \sin \theta & \cos \theta \end{pmatrix}. \quad (16)$$

Defining $c_\theta = \cos \theta$, $d_\theta = 1 - \cos \theta$, and $s_\theta = \sin \theta$

$$\overline{\mathbf{R}}_\theta = \begin{pmatrix} c_\theta + z_y^2 d_\theta & -z_y z_x d_\theta & -z_x s_\theta \\ -z_x z_y d_\theta & c_\theta + z_x^2 d_\theta & -z_y s_\theta \\ z_x s_\theta & z_y s_\theta & c_\theta \end{pmatrix}. \quad (17)$$

Then we find the vector mean of \vec{x} and \vec{y} , defined as \hat{v} ,

$$\vec{v} = \frac{1}{2}(\vec{x} + \vec{y}). \quad (18)$$

This vector lies in the current xy plane and bisects the angle between the \vec{x} and \vec{y} vectors. If the input matrix as orthogonal the vector \vec{v} would be 45 degrees from both \vec{x} and \vec{y} vectors. We wish to distribute the deviation from 45 degrees to each axis so that we re-orthogonalize the xy plane properly. We do this by rotating \vec{v} to the ij plane, then measuring the angle ψ from the \hat{i} axis. Knowing this should be 45 degrees we rotate the new vector \vec{v}' by -45 degrees and then by the measured angle ψ .

First rotate \vec{v} into the xy plane, and normalize the resulting vector so that we can get rotation angles.

$$\hat{v}' = \widehat{R_\theta \vec{v}} = R_\theta \cdot (\vec{x} + \vec{y}). \quad (19)$$

\hat{v}' is now a unit vector with the orientation of the vector mean of \vec{x} and \vec{y} projected into the ij plane. This vector is the orientation vector that in an orthogonal system would be rotated 45 degrees from the \hat{i} axis. In the imperfect input DCM this angle is not 45 degrees, but an arbitrary angle ψ . We calculate the cosine and sine of this angle to create the rotation matrix that will rotate this vector onto the \hat{i} axis. \hat{v}' and \hat{i} are again normalized, meaning their dot product is simply the cosine of the angle separating them,

$$\cos \psi = \hat{v}' \cdot \hat{i}, \quad (20)$$

$$\sin \psi = \hat{v}' \cdot \hat{j}, \quad (21)$$

where $\sin \psi$ can be calculated knowing that \hat{v}' lies in the ij plane. These operations are straightforward and the dot product is not actually needed

$$\cos \psi = v'_x, \quad (22)$$

$$\sin \psi = v'_y. \quad (23)$$

The rotation matrix $\overline{\mathbf{R}}_\psi$ can be constructed as $\overline{\mathbf{R}}_\theta$ was, with the rotation axis $\hat{u} = \hat{k}$

$$\overline{\mathbf{R}}_\psi = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (24)$$

We (could) apply this rotation to \hat{v}' , but that is unnecessary at this stage. The final rotation is the -45 degree rotation to move the (idealized 45 degree) \vec{v}' back to the \hat{i} axis. With $\phi = -45^\circ$,

$$\overline{\mathbf{R}}_\phi = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (25)$$

The final rotation matrix that gives the orientation of the vehicle is then the product of these three rotation matrices in reverse order

$$\overline{\mathbf{M}} = \overline{\mathbf{R}}_\theta^T \overline{\mathbf{R}}_\psi \overline{\mathbf{R}}_\phi \quad (26)$$

Expanding these matrices

$$\overline{\mathbf{M}} = \begin{pmatrix} c_\theta + z_y^2 d_\theta & -z_y z_x d_\theta & -z_x s_\theta \\ -z_x z_y d_\theta & c_\theta + z_x^2 d_\theta & -z_y s_\theta \\ z_x s_\theta & z_y s_\theta & c_\theta \end{pmatrix}^T \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (27)$$

$$= \begin{pmatrix} c_\theta + z_y^2 d_\theta & -z_y z_x d_\theta & -z_x s_\theta \\ -z_x z_y d_\theta & c_\theta + z_x^2 d_\theta & -z_y s_\theta \\ z_x s_\theta & z_y s_\theta & c_\theta \end{pmatrix}^T \begin{pmatrix} c_\psi c_\phi - s_\psi s_\phi & -s_\psi c_\phi - c_\psi s_\phi & 0 \\ +s_\psi c_\phi + c_\psi s_\phi & c_\psi c_\phi - s_\psi s_\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (28)$$

Given $\cos \phi = -\sin \phi = \frac{1}{\sqrt{2}}$

$$\overline{\mathbf{M}} = \begin{pmatrix} c_\theta + z_y^2 d_\theta & -z_y z_x d_\theta & -z_x s_\theta \\ -z_x z_y d_\theta & c_\theta + z_x^2 d_\theta & -z_y s_\theta \\ z_x s_\theta & z_y s_\theta & c_\theta \end{pmatrix}^T \begin{pmatrix} c_\psi \frac{1}{\sqrt{2}} + s_\psi \frac{1}{\sqrt{2}} & -s_\psi \frac{1}{\sqrt{2}} + c_\psi \frac{1}{\sqrt{2}} & 0 \\ +s_\psi \frac{1}{\sqrt{2}} - c_\psi \frac{1}{\sqrt{2}} & c_\psi \frac{1}{\sqrt{2}} + s_\psi \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (29)$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} c_\theta + z_y^2 d_\theta & -z_y z_x d_\theta & -z_x s_\theta \\ -z_x z_y d_\theta & c_\theta + z_x^2 d_\theta & -z_y s_\theta \\ z_x s_\theta & z_y s_\theta & c_\theta \end{pmatrix}^T \begin{pmatrix} c_\psi + s_\psi & -s_\psi + c_\psi & 0 \\ +s_\psi - c_\psi & c_\psi + s_\psi & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix} \quad (30)$$

5 Alternate Alternate normalizing method

Similar to the last method, but without rotating \vec{z} onto \vec{k} . \vec{x} and \vec{y} define a plane, $\vec{z}' = \widehat{\vec{x} \times \vec{y}}$ is normal to that plane. The mean of \vec{x} and \vec{y} is in the plane and 45 degrees from the corrected vectors \vec{x}' and \vec{y}' .

The rotation matrix above can also be written as :

$$\overline{\mathbf{R}}_\theta = \cos \theta \overline{\mathbf{I}} + \sin \theta [\vec{u}]_\times + (1 - \cos \theta) \vec{u} \otimes \vec{u} \quad (31)$$

Where \vec{u} is a column vector, $[\vec{u}]_\times$ is the cross product matrix, and $\vec{u} \otimes \vec{u}$ is the tensor or outer product matrix:

$$[\vec{u}]_\times = \begin{pmatrix} 0 & -u_z & -u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}, \vec{u} \otimes \vec{u} = \vec{u} \vec{u}^T = \begin{pmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{pmatrix} \quad (32)$$

The cosine terms are symmetric and the sin term is antisymmetric. So the transpose :

$$\overline{\mathbf{R}_\theta^T} = \cos \theta \bar{\mathbf{I}} - \sin \theta [\vec{u}]_\times + (1 - \cos \theta) \vec{u} \otimes \vec{u} \quad (33)$$

Substituting \vec{z}' for \vec{u} and using 45 degrees :

$$\cos(\frac{\pi}{4}) = \sin(\frac{\pi}{4}) = \frac{1}{\sqrt{2}} \quad (34)$$

$$\overline{\mathbf{R}_{\frac{\pi}{4}}} = \frac{1}{\sqrt{2}} \bar{\mathbf{I}} + \frac{1}{\sqrt{2}} [\vec{z}']_\times + (1 - \frac{1}{\sqrt{2}}) (\vec{z}' \otimes \vec{z}') \quad (35)$$

and applying it to $\vec{v} = \widehat{(\vec{x} + \vec{y})}$ to get :

$$\vec{x}' = \frac{1}{\sqrt{2}} \vec{v} + \frac{1}{\sqrt{2}} [\vec{z}']_\times \vec{v} + (1 - \frac{1}{\sqrt{2}}) (\vec{z}' \otimes \vec{z}') \vec{v} \quad (36)$$

$$= \frac{1}{\sqrt{2}} \vec{v} + \frac{1}{\sqrt{2}} (\vec{z}' \times \vec{v}) + \vec{z}' \vec{z}'^T (\vec{v} - \frac{1}{\sqrt{2}} \vec{v}) \quad (37)$$

defining $\vec{v}' = \frac{1}{\sqrt{2}} \vec{v}$, then the equations for \vec{x}' and \vec{y}' become :

$$\vec{x}' = \vec{v}' + \vec{z}' \vec{z}'^T (\vec{v} - \vec{v}') + \vec{z}' \times \vec{v}' \quad (38)$$

$$\vec{y}' = \vec{v}' + \vec{z}' \vec{z}'^T (\vec{v} - \vec{v}') - \vec{z}' \times \vec{v}' \quad (39)$$

writing \vec{x} out in full gives :

$$x'_0 = v'_0 + z'_0 (z'_0 u_0 + z'_1 u_1 + z'_2 u_2) + z'_1 v'_2 - z'_2 v'_1 \quad (40)$$

$$x'_1 = v'_1 + z'_1 (z'_0 u_0 + z'_1 u_1 + z'_2 u_2) - z'_0 v'_2 + z'_2 v'_0 \quad (41)$$

$$x'_2 = v'_2 + z'_2 (z'_0 u_0 + z'_1 u_1 + z'_2 u_2) + z'_0 v'_1 - z'_1 v'_0 \quad (42)$$

The pseudocode for this :

$$\vec{z}' = \text{normalize}(\vec{x} \times \vec{y}) \quad (43)$$

$$\vec{v} = \text{normalize}(\vec{x} + \vec{y}) \quad (44)$$

$$\vec{v}' = \frac{1}{\sqrt{2}} \vec{v} \quad (45)$$

$$c = \vec{z}' \bullet (\vec{v} - \vec{v}') \quad (46)$$

$$\vec{a} = \vec{z}' \times \vec{v}' \quad (47)$$

$$\vec{b} = c \vec{z}' + \vec{v}' \quad (48)$$

$$\vec{x}' = \vec{b} - \vec{a} \quad (49)$$

$$\vec{y}' = \vec{b} + \vec{a} \quad (50)$$

Computing values for \vec{v} , \vec{v}' and c should be interleaved.