

Introduction

This article is a continuation of my [IMU Guide](#), covering additional orientation kinematics topics. I will go through some theory first and then I will present a practical example with code build around an Arduino and a 6DOF IMU sensor (acc_gyro_6dof). The scope of this experiment is to create an algorithm for fusing gyroscope and accelerometer data in order to create an estimation of the device orientation in space. Such an algorithm was already presented in part 3 of my “IMU Guide” and a practical Arduino experiment with code was presented in the “[Using a 5DOF IMU](#)” article and was nicknamed “Simplified Kalman Filter”, providing a simple alternative to the well known Kalman Filter algorithm. In this article we’ll use another approach utilizing the DCM (Direction Cosine Matrix). For the reader that is unfamiliar with MEMS sensors it is recommended to read Part 1 and 2 of the IMU Guide article. Also for following the experiments presented in this text it is recommended to acquire an Arduino board and an [acc_gyro_6dof](#) sensor.

Prerequisites

No really advanced math is necessary. Find a good book on matrix operations, that’s all you might need above school math course. If you would like to refresh your knowledge below are some quick articles:

Cartesian Coordinate System – http://en.wikipedia.org/wiki/Cartesian_coordinate_system

Rotation – http://en.wikipedia.org/wiki/Rotation_%28mathematics%29

Vector scalar product – http://en.wikipedia.org/wiki/Dot_product

Vector cross product – http://en.wikipedia.org/wiki/Cross_product

Matrix Multiplication – http://en.wikipedia.org/wiki/Matrix_multiplication

Block Matrix – http://en.wikipedia.org/wiki/Block_matrix

Transpose Matrix – <http://en.wikipedia.org/wiki/Transpose>

Triple Product – http://en.wikipedia.org/wiki/Triple_product

Notations

Vectors are marked in **bold text** - so for example “**v**” is a vector and “v” is a scalar (if you can’t distinguish the two there’s problem with the text formatting wherever you’re reading this).

Part 1. The DCM Matrix

Generally speaking orientation kinematics deals with calculating the relative orientation of a body relative to a global coordinate system. It is useful to attach a coordinate system to our body frame and call it Oxyz, and another one to our global frame and call it OXYZ. Both the global and the body frames have the same fixed origin O (see *Fig. 1*). Let’s also define **i**, **j**, **k** to be unity vectors co-directional with the body frame’s x, y, and z axes – in other words they are versors of Oxyz and let **I**, **J**, **K** be the versors of global frame OXYZ.

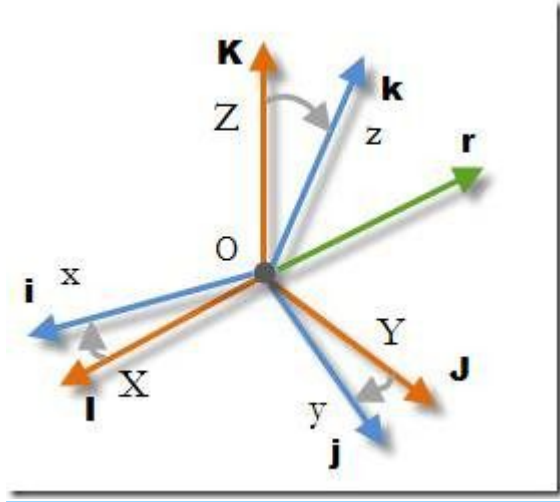


Figure 1

Thus, by definition, expressed **in terms of global coordinates** vectors **I, J, K** can be written as:

$$\mathbf{I}^G = \{1, 0, 0\}^T, \mathbf{J}^G = \{0, 1, 0\}^T, \mathbf{K}^G = \{0, 0, 1\}^T$$

Note: we use $\{\dots\}^T$ notation to denote a column vector, in other words a column vector is a translated row vector. The orientation of vectors (row/column) will become relevant once we start multiplying them by a matrix later on in this text.

And similarly, **in terms of body coordinates** vectors **i, j, k** can be written as:

$$\mathbf{i}^B = \{1, 0, 0\}^T, \mathbf{j}^B = \{0, 1, 0\}^T, \mathbf{k}^B = \{0, 0, 1\}^T$$

Now let's see if we can write vectors **i, j, k** in terms of global coordinates. Let's take vector **i** as an example and write its global coordinates:

$$\mathbf{i}^G = \{i_x^G, i_y^G, i_z^G\}^T$$

Again, by example let's analyze the X coordinate i_x^G , it's calculated as the length of projection of the **i** vector onto the global X axis.

$$i_x^G = |\mathbf{i}| \cos(X, \mathbf{i}) = \cos(\mathbf{I}, \mathbf{i})$$

Where $|\mathbf{i}|$ is the norm (length) of the **i** unit vector and $\cos(\mathbf{I}, \mathbf{i})$ is the cosine of the angle formed by the vectors **I** and **i**. Using the fact that $|\mathbf{I}| = 1$ and $|\mathbf{i}| = 1$ (they are unit vectors by definition). We can write:

$$i_x^G = \cos(\mathbf{I}, \mathbf{i}) = |\mathbf{I}||\mathbf{i}| \cos(\mathbf{I}, \mathbf{i}) = \mathbf{I} \cdot \mathbf{i}$$

Where $\mathbf{I} \cdot \mathbf{i}$ is the scalar (dot) product of vectors **I** and **i**. For the purpose of calculating scalar product $\mathbf{I} \cdot \mathbf{i}$ it doesn't matter in which coordinate system these vectors are measured as long as they are both expressed in the same system, since a rotation does not modify the angle between vectors so: $\mathbf{I} \cdot \mathbf{i} = \mathbf{I}^B \cdot \mathbf{i}^B = \mathbf{I}^G \cdot \mathbf{i}^G = \cos(\mathbf{I}^B \cdot \mathbf{i}^B) = \cos(\mathbf{I}^G \cdot \mathbf{i}^G)$, so for simplicity we'll skip the superscript in scalar products $\mathbf{I} \cdot \mathbf{i}$, $\mathbf{J} \cdot \mathbf{j}$, $\mathbf{K} \cdot \mathbf{k}$ and in cosines $\cos(\mathbf{I}, \mathbf{i})$, $\cos(\mathbf{J}, \mathbf{j})$, $\cos(\mathbf{K}, \mathbf{k})$.

Similarly we can show that:

$i_y^G = \mathbf{J} \cdot \mathbf{i}$, $i_z^G = \mathbf{K} \cdot \mathbf{i}$, so now we can write vector \mathbf{i} in terms of global coordinate system as:

$$\mathbf{i}^G = \{ \mathbf{I} \cdot \mathbf{i}, \mathbf{J} \cdot \mathbf{i}, \mathbf{K} \cdot \mathbf{i} \}^T$$

Furthermore, similarly it can be shown that $\mathbf{j}^G = \{ \mathbf{I} \cdot \mathbf{j}, \mathbf{J} \cdot \mathbf{j}, \mathbf{K} \cdot \mathbf{j} \}^T$, $\mathbf{k}^G = \{ \mathbf{I} \cdot \mathbf{k}, \mathbf{J} \cdot \mathbf{k}, \mathbf{K} \cdot \mathbf{k} \}^T$.

We now have a complete set of global coordinates for our body's versors \mathbf{i} , \mathbf{j} , \mathbf{k} and we can organize these values in a convenient matrix form:

$$[\mathbf{i}^G, \mathbf{j}^G, \mathbf{k}^G] = \begin{bmatrix} \mathbf{I} \cdot \mathbf{i} & \mathbf{I} \cdot \mathbf{j} & \mathbf{I} \cdot \mathbf{k} \\ \mathbf{J} \cdot \mathbf{i} & \mathbf{J} \cdot \mathbf{j} & \mathbf{J} \cdot \mathbf{k} \\ \mathbf{K} \cdot \mathbf{i} & \mathbf{K} \cdot \mathbf{j} & \mathbf{K} \cdot \mathbf{k} \end{bmatrix} = \begin{bmatrix} \cos(\mathbf{I}, \mathbf{i}) & \cos(\mathbf{I}, \mathbf{j}) & \cos(\mathbf{I}, \mathbf{k}) \\ \cos(\mathbf{J}, \mathbf{i}) & \cos(\mathbf{J}, \mathbf{j}) & \cos(\mathbf{J}, \mathbf{k}) \\ \cos(\mathbf{K}, \mathbf{i}) & \cos(\mathbf{K}, \mathbf{j}) & \cos(\mathbf{K}, \mathbf{k}) \end{bmatrix} = \text{DCM}^G$$

(Eq. 1.1)

This matrix is called Direction Cosine Matrix for now obvious reasons – it consists of cosines of angles of all possible combinations of body and global versors.

The task of expressing the global frame versors \mathbf{I}^G , \mathbf{J}^G , \mathbf{K}^G in body frame coordinates is symmetrical in nature and can be achieved by simply swapping the notations \mathbf{I} , \mathbf{J} , \mathbf{K} with \mathbf{i} , \mathbf{j} , \mathbf{k} , the results being:

$$\mathbf{I}^B = \{ \mathbf{I} \cdot \mathbf{i}, \mathbf{I} \cdot \mathbf{j}, \mathbf{I} \cdot \mathbf{k} \}^T, \mathbf{J}^B = \{ \mathbf{J} \cdot \mathbf{i}, \mathbf{J} \cdot \mathbf{j}, \mathbf{J} \cdot \mathbf{k} \}^T, \mathbf{K}^B = \{ \mathbf{K} \cdot \mathbf{i}, \mathbf{K} \cdot \mathbf{j}, \mathbf{K} \cdot \mathbf{k} \}^T$$

and organized in a matrix form:

$$[\mathbf{I}^B, \mathbf{J}^B, \mathbf{K}^B] = \begin{bmatrix} \mathbf{I} \cdot \mathbf{i} & \mathbf{J} \cdot \mathbf{i} & \mathbf{K} \cdot \mathbf{i} \\ \mathbf{I} \cdot \mathbf{j} & \mathbf{J} \cdot \mathbf{j} & \mathbf{K} \cdot \mathbf{j} \\ \mathbf{I} \cdot \mathbf{k} & \mathbf{J} \cdot \mathbf{k} & \mathbf{K} \cdot \mathbf{k} \end{bmatrix} = \begin{bmatrix} \cos(\mathbf{I}, \mathbf{i}) & \cos(\mathbf{J}, \mathbf{i}) & \cos(\mathbf{K}, \mathbf{i}) \\ \cos(\mathbf{I}, \mathbf{j}) & \cos(\mathbf{J}, \mathbf{j}) & \cos(\mathbf{K}, \mathbf{j}) \\ \cos(\mathbf{I}, \mathbf{k}) & \cos(\mathbf{J}, \mathbf{k}) & \cos(\mathbf{K}, \mathbf{k}) \end{bmatrix} = \text{DCM}^B$$

(Eq. 1.2)

It is now easy to notice that $\text{DCM}^B = (\text{DCM}^G)^T$ or $\text{DCM}^G = (\text{DCM}^B)^T$, in other words the two matrices are transposes of each other, we'll use this important property later on.

Also notice that $\text{DCM}^B \cdot \text{DCM}^G = (\text{DCM}^G)^T \cdot \text{DCM}^G = \text{DCM}^B \cdot (\text{DCM}^B)^T = \mathbf{I}_3$, where \mathbf{I}_3 is the 3×3 identity matrix. In other words the DCM matrices are orthogonal.

This can be proven by simply expanding the matrix multiplication in block matrix form:

$$\text{DCM}^B \text{DCM}^G = \text{DCM}^{GT} \text{DCM}^G = \begin{bmatrix} \mathbf{i}^{GT} \\ \mathbf{j}^{GT} \\ \mathbf{k}^{GT} \end{bmatrix} [\mathbf{i}^G, \mathbf{j}^G, \mathbf{k}^G] =$$

$$\begin{bmatrix} \mathbf{i}^{GT} \cdot \mathbf{i}^G & \mathbf{i}^{GT} \cdot \mathbf{j}^G & \mathbf{i}^{GT} \cdot \mathbf{k}^G \\ \mathbf{j}^{GT} \cdot \mathbf{i}^G & \mathbf{j}^{GT} \cdot \mathbf{j}^G & \mathbf{j}^{GT} \cdot \mathbf{k}^G \\ \mathbf{k}^{GT} \cdot \mathbf{i}^G & \mathbf{k}^{GT} \cdot \mathbf{j}^G & \mathbf{k}^{GT} \cdot \mathbf{k}^G \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_3$$

(Eq. 1.3)

To prove this we use such properties as for example: $\mathbf{i}^{GT} \cdot \mathbf{i}^G = |\mathbf{i}^G| |\mathbf{i}^G| \cos(0) = 1$ and $\mathbf{i}^{GT} \cdot \mathbf{j}^G = 0$ because (\mathbf{i} and \mathbf{j} are orthogonal) and so forth.

The DCM matrix (also often called the rotation matrix) has a great importance in orientation kinematics since it defines the rotation of one frame relative to another. It can also be used to determine the global coordinates of an arbitrary vector if we know its coordinates in the body frame (and vice versa).

Let's consider such a vector with body coordinates:

$\mathbf{r}^B = \{r_x^B, r_y^B, r_z^B\}^T$ and let's try to determine its coordinates in the global frame, by using a known rotation matrix DCM^G .

We start by doing following notation:

$$\mathbf{r}^G = \{r_x^G, r_y^G, r_z^G\}^T.$$

Now let's tackle the first coordinate r_x^G :

$r_x^G = |\mathbf{r}^G| \cos(\mathbf{I}^G, \mathbf{r}^G)$, because r_x^G is the projection of \mathbf{r}^G onto X axis that is co-directional with \mathbf{I}^G .

Next let's note that by definition a rotation is such a transformation that does not change the scale of a vector and does not change the angle between two vectors that are subject to the same rotation, so if we express some vectors in a different rotated coordinate system the norm and angle between vectors will not change:

$|\mathbf{r}^G| = |\mathbf{r}^B|$, $|\mathbf{I}^G| = |\mathbf{I}^B| = 1$ and $\cos(\mathbf{I}^G, \mathbf{r}^G) = \cos(\mathbf{I}^B, \mathbf{r}^B)$, so we can use this property to write

$r_x^G = |\mathbf{r}^G| \cos(\mathbf{I}^G, \mathbf{r}^G) = |\mathbf{I}^B| |\mathbf{r}^B| \cos(\mathbf{I}^B, \mathbf{r}^B) = \mathbf{I}^B \cdot \mathbf{r}^B = \mathbf{I}^B \cdot \{r_x^B, r_y^B, r_z^B\}^T$, by using one the two definition of the scalar product.

Now recall that $\mathbf{I}^B = \{\mathbf{I.i}, \mathbf{I.j}, \mathbf{I.k}\}^T$ and by using the other definition of scalar product:

$$r_x^G = \mathbf{I}^B \cdot \mathbf{r}^B = \{\mathbf{I.i}, \mathbf{I.j}, \mathbf{I.k}\}^T \cdot \{r_x^B, r_y^B, r_z^B\}^T = r_x^B \mathbf{I.i} + r_y^B \mathbf{I.j} + r_z^B \mathbf{I.k}$$

In same fashion it can be shown that:

$$\begin{aligned} r_y^G &= r_x^B \mathbf{J.i} + r_y^B \mathbf{J.j} + r_z^B \mathbf{J.k} \\ r_z^G &= r_x^B \mathbf{K.i} + r_y^B \mathbf{K.j} + r_z^B \mathbf{K.k} \end{aligned}$$

Finally let's write this in a more compact matrix form:

$$\mathbf{r}^G = \begin{bmatrix} r_x^G \\ r_y^G \\ r_z^G \end{bmatrix} = \begin{bmatrix} \mathbf{I.i} & \mathbf{I.j} & \mathbf{I.k} \\ \mathbf{J.i} & \mathbf{J.j} & \mathbf{J.k} \\ \mathbf{K.i} & \mathbf{K.j} & \mathbf{K.k} \end{bmatrix} \begin{bmatrix} r_x^B \\ r_y^B \\ r_z^B \end{bmatrix} = DCM^G \mathbf{r}^B \quad (Eq. 1.4)$$

Thus the DCM matrix can be used to covert an arbitrary vector \mathbf{r}^B expressed in one coordinate system B, to a rotated coordinate system G.

We can use similar logic to prove the reverse process:

$$\mathbf{r}^B = \text{DCM}^B \mathbf{r}^G \quad (\text{Eq. 1.5})$$

Or we can arrive at the same conclusion by multiplying both parts in (Eq. 1.4) by DCM^B which equals to DCM^{GT} , and using the property that $\text{DCM}^{GT} \cdot \text{DCM}^G = I_3$, see (Eq. 1.3):

$$\text{DCM}^B \mathbf{r}^G = \text{DCM}^B \text{DCM}^G \mathbf{r}^B = \text{DCM}^{GT} \text{DCM}^G \mathbf{r}^B = I_3 \mathbf{r}^B = \mathbf{r}^B$$

Part 2. Angular Velocity

So far we have a way to characterize the orientation of one frame relative to another rotated frame, it is the DCM matrix and it allows us to easily convert the global and body coordinates back and forth using (Eq. 1.4) and (Eq. 1.5). In this section we'll analyze the rotation as a function of time that will help us establish the rules of updating the DCM matrix based on a characteristic called angular velocity. Let's consider an arbitrary rotating vector \mathbf{r} and define its coordinates at time t to be $\mathbf{r}(t)$. Now let's consider a small time interval dt and make the following notations: $\mathbf{r} = \mathbf{r}(t)$, $\mathbf{r}' = \mathbf{r}(t+dt)$ and $d\mathbf{r} = \mathbf{r}' - \mathbf{r}$:

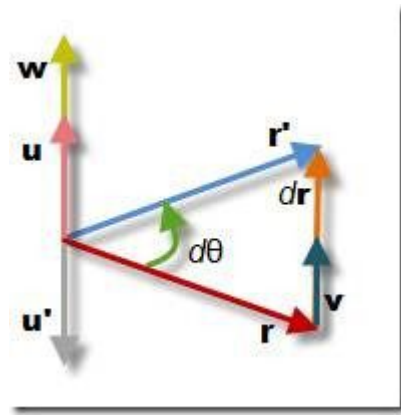


Figure 2

Let's say that during a very small time interval $dt \rightarrow 0$ the vector \mathbf{r} has rotated about an axis co-directional with a unity vector \mathbf{u} by an angle $d\theta$ and ended up in the position \mathbf{r}' . Since \mathbf{u} is our axis of rotation it is perpendicular to the plane in which the rotation took place (the plane formed by \mathbf{r} and \mathbf{r}') so \mathbf{u} is orthogonal to both \mathbf{r} and \mathbf{r}' . There are two unity vectors that are orthogonal to the plane formed by \mathbf{r} and \mathbf{r}' , they are shown on the picture as \mathbf{u} and \mathbf{u}' since we're still defining things we'll choose the one that is co-directional with the cross product $\mathbf{r} \times \mathbf{r}'$, following the rule of [right-handed coordinate system](#). Thus because \mathbf{u} is a unity vector $|\mathbf{u}| = 1$ and is co-directional with $\mathbf{r} \times \mathbf{r}'$ we can deduct it as follows:

$$\mathbf{u} = (\mathbf{r} \times \mathbf{r}') / |\mathbf{r} \times \mathbf{r}'| = (\mathbf{r} \times \mathbf{r}') / (|\mathbf{r}| |\mathbf{r}'| \sin(d\theta)) = (\mathbf{r} \times \mathbf{r}') / (|\mathbf{r}|^2 \sin(d\theta)) \quad (\text{Eq. 2.1})$$

Since a rotation does not alter the length of a vector we used the property that $|\mathbf{r}'| = |\mathbf{r}|$.

The linear velocity of the vector \mathbf{r} can be defined as the vector:

$$\mathbf{v} = d\mathbf{r} / dt = (\mathbf{r}' - \mathbf{r}) / dt \text{ (Eq. 2.2)}$$

Please note that since our dt approaches 0 so does $d\theta \rightarrow 0$, hence the angle between vectors \mathbf{r} and $d\mathbf{r}$ (let's call it α) can be found from the isosceles triangle contoured by \mathbf{r} , \mathbf{r}' and $d\mathbf{r}$:

$$\alpha = (\pi - d\theta) / 2 \text{ and because } d\theta \rightarrow 0, \text{ then } \alpha \rightarrow \pi/2$$

What this tells us is that \mathbf{r} is perpendicular to $d\mathbf{r}$ when $dt \rightarrow 0$ and hence \mathbf{r} is perpendicular to \mathbf{v} since \mathbf{v} and $d\mathbf{r}$ are co-directional from (Eq. 2.2):

$$\mathbf{v} \perp \mathbf{r} \text{ (Eq. 2.21)}$$

We are now ready to define the angular velocity vector. Ideally such a vector should define the rate of change of the angle θ and the axis of the rotation, so we define it as follows:

$$\mathbf{w} = (d\theta/dt) \mathbf{u} \text{ (Eq. 2.3)}$$

Indeed the norm of the \mathbf{w} is $|\mathbf{w}| = d\theta/dt$ and the direction of \mathbf{w} coincides with the axis of rotation \mathbf{u} . Let's expand (Eq. 2.3) and try to establish a relationship with the linear velocity \mathbf{v} :

Using (Eq. 2.3) and (Eq. 2.1):

$$\mathbf{w} = (d\theta/dt) \mathbf{u} = (d\theta/dt) (\mathbf{r} \times \mathbf{r}') / (|\mathbf{r}|^2 \sin(d\theta))$$

Now note that when $dt \rightarrow 0$, so does $d\theta \rightarrow 0$ and hence for small $d\theta$, $\sin(d\theta) \approx d\theta$, we end up with:

$$\mathbf{w} = (\mathbf{r} \times \mathbf{r}') / (|\mathbf{r}|^2 dt) \text{ (Eq. 2.4)}$$

Now because $\mathbf{r}' = \mathbf{r} + d\mathbf{r}$, $d\mathbf{r}/dt = \mathbf{v}$, $\mathbf{r} \times \mathbf{r} = \mathbf{0}$ and using the distributive property of cross product over addition:

$$\mathbf{w} = (\mathbf{r} \times (\mathbf{r} + d\mathbf{r})) / (|\mathbf{r}|^2 dt) = (\mathbf{r} \times \mathbf{r} + \mathbf{r} \times d\mathbf{r}) / (|\mathbf{r}|^2 dt) = \mathbf{r} \times (d\mathbf{r}/dt) / |\mathbf{r}|^2$$

And finally:

$$\mathbf{w} = \mathbf{r} \times \mathbf{v} / |\mathbf{r}|^2 \text{ (Eq. 2.5)}$$

This equation establishes a way to calculate angular velocity from a known linear velocity \mathbf{v} .

We can easily prove the reverse equation that lets us deduct linear velocity from angular velocity:

$$\mathbf{v} = \mathbf{w} \times \mathbf{r} \text{ (Eq. 2.6)}$$

This can be proven simply by expanding \mathbf{w} from (Eq. 2.5) and using vector [triple product](#) rule $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{b} \cdot \mathbf{c})\mathbf{a}$. Also we'll use the fact that \mathbf{v} and \mathbf{r} are perpendicular (Eq. 2.21) and thus $\mathbf{v} \cdot \mathbf{r} = 0$

$$\mathbf{w} \times \mathbf{r} = (\mathbf{r} \times \mathbf{v} / |\mathbf{r}|^2) \times \mathbf{r} = (\mathbf{r} \times \mathbf{v}) \times \mathbf{r} / |\mathbf{r}|^2 = ((\mathbf{r} \cdot \mathbf{r})\mathbf{v} + (\mathbf{v} \cdot \mathbf{r})\mathbf{r}) / |\mathbf{r}|^2 = (|\mathbf{r}|^2 \mathbf{v} + 0) / |\mathbf{r}|^2 = \mathbf{v}$$

So we just proved that (Eq. 2.6) is true. Just to check (Eq. 2.6) intuitively – from Figure 2 indeed \mathbf{v} has the direction of $\mathbf{w} \times \mathbf{r}$ using the right hand rule and indeed $\mathbf{v} \perp \mathbf{r}$ and $\mathbf{v} \perp \mathbf{w}$ because it is in the same plane with \mathbf{r} and \mathbf{r}' .

Part 3. Gyroscopes and angular velocity vector

A 3-axis MEMS gyroscope is a device that senses rotation about 3 axes attached to the device itself (body frame). If we adopt the device's coordinate system (body's frame), and analyze some vectors attached to the earth (global frame), for example vector \mathbf{K} pointing to the zenith or vector \mathbf{I} pointing North – then it would appear to an observer inside the device that these vector rotate about the device center. Let w_x, w_y, w_z be the outputs of a gyroscope expressed in rad/s – the measured rotation about axes x, y, z respectively. Converting from the raw output of the gyroscope to physical values is discussed for example here: http://www.starlino.com/imu_guide.html . If we query the gyroscope at regular, small time intervals dt , then what gyroscope output tells us is that during this time interval the earth rotated about gyroscope's x axis by an angle of $d\theta_x = w_x dt$, about y axis by an angle of $d\theta_y = w_y dt$ and about z axis by an angle of $d\theta_z = w_z dt$. These rotations can be characterized by the angular velocity vectors: $\mathbf{w}_x = w_x \mathbf{i} = \{w_x, 0, 0\}^T$, $\mathbf{w}_y = w_y \mathbf{j} = \{0, w_y, 0\}^T$, $\mathbf{w}_z = w_z \mathbf{k} = \{0, 0, w_z\}^T$, where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are versors of the local coordinate frame (they are co-directional with body's axes x, y, z respectively). Each of these three rotations will cause a linear displacement which can be expressed by using (Eq. 2.6):

$$d\mathbf{r}_1 = dt \mathbf{v}_1 = dt (\mathbf{w}_x \times \mathbf{r}) ; d\mathbf{r}_2 = dt \mathbf{v}_2 = dt (\mathbf{w}_y \times \mathbf{r}) ; d\mathbf{r}_3 = dt \mathbf{v}_3 = dt (\mathbf{w}_z \times \mathbf{r}) .$$

The combined effect of these three displacements will be:

$$d\mathbf{r} = d\mathbf{r}_1 + d\mathbf{r}_2 + d\mathbf{r}_3 = dt (\mathbf{w}_x \times \mathbf{r} + \mathbf{w}_y \times \mathbf{r} + \mathbf{w}_z \times \mathbf{r}) = dt (\mathbf{w}_x + \mathbf{w}_y + \mathbf{w}_z) \times \mathbf{r} \text{ (cross product is distributive over addition)}$$

Thus the equivalent linear velocity resulting from these 3 transformations can be expressed as:

$$\mathbf{v} = d\mathbf{r}/dt = (\mathbf{w}_x + \mathbf{w}_y + \mathbf{w}_z) \times \mathbf{r} = \mathbf{w} \times \mathbf{r} , \text{ where we introduce } \mathbf{w} = \mathbf{w}_x + \mathbf{w}_y + \mathbf{w}_z = \{w_x, w_y, w_z\}$$

Which looks exactly like (Eq. 2.6) and suggests that the combination of three **small** rotations about axes x, y, z characterized by angular rotation vectors $\mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z$ is equivalent to one **small** rotation characterized by angular rotation vector $\mathbf{w} = \mathbf{w}_x + \mathbf{w}_y + \mathbf{w}_z = \{w_x, w_y, w_z\}$. Please note that we're stressing out that these are **small** rotations, since in general when you combine large rotations the order in which rotations are performed become important and you cannot simply sum them up. Our main assumption that let us go from a linear displacement to a rotation by using (Eq. 2.6) was that dt is really small, and thus the rotations $d\theta$ and linear displacement $d\mathbf{r}$ are small as well. In practice this means that the larger the dt interval between gyro queries the larger will be our accumulated error, we'll deal with this error later on. Now, since w_x, w_y, w_z are the output of the gyroscope, then we arrive at the conclusion that in fact a 3 axis

gyroscope measures the instantaneous angular velocity of the world rotating about the device's center.

Part 4. DCM complimentary filter algorithm using 6DOF or 9DOF IMU sensors

In the context of this text a 6DOF device is an IMU device consisting of a 3 axis gyroscope and a 3 axis accelerometer. A 9DOF device is an IMU device of a 3 axis gyroscope, a 3 axis accelerometer and a 3 axis magnetometer. Let's attach a global right-handed coordinate system to the Earth's frame such that the **I** versor points North, **K** versor points to the Zenith and thus, with these two versors fixed, the **J** versor will be constrained to point West.

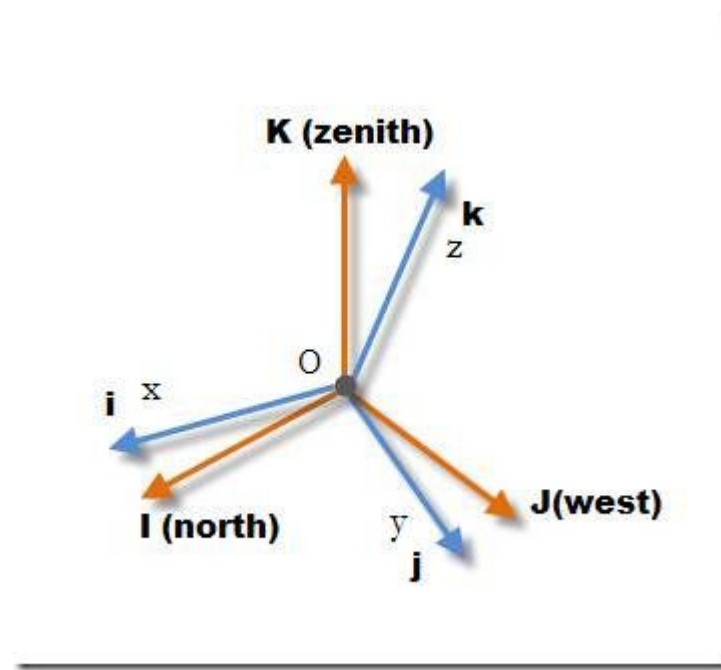


Figure 3

Also let's consider the body coordinate system to be attached to our IMU device (acc_gyro used as an example),

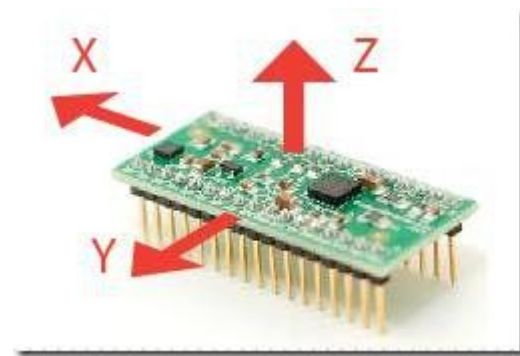


Figure 4

We already established the fact that gyroscopes can measure the angular velocity vector. Let's see how accelerometer and magnetometer measurements will fall into our model.

Accelerometers are devices that can sense gravitation. Gravitation vector is pointing towards the center of the earth and is opposite to the vector pointing to Zenith \mathbf{K}^B . If the 3 axis accelerometer output is $\mathbf{A} = \{A_x, A_y, A_z\}$ and we assume that there are no external accelerations or we have corrected them then we can estimate that $\mathbf{K}^B = -\mathbf{A}$. (See this IMU Guide for more clarifications http://www.starlino.com/imu_guide.html).

Magnetometers are devices that are really similar to accelerometers, except that instead of gravitation they can sense the Earth's magnetic North. Just like accelerometers they are not perfect and often need corrections and initial calibration. If the corrected 3-axis magnetometer output is $\mathbf{M} = \{M_x, M_y, M_z\}$, then according to our model \mathbf{I}^B is pointing North, thus $\mathbf{I}^B = \mathbf{M}$.

Knowing \mathbf{I}^B and \mathbf{K}^B allows us calculate $\mathbf{J}^B = \mathbf{K}^B \times \mathbf{I}^B$.

Thus an accelerometer and a magnetometer alone can give us the DCM matrix, expressed either as DCM^B or DCM^G

$$\text{DCM}^G = \text{DCM}^{BT} = [\mathbf{I}^B, \mathbf{J}^B, \mathbf{K}^B]^T$$

The DCM matrix can be used to convert any vector from body's(devices) coordinate system to the global coordinate system. Thus for example if we know that the nose of the plane has some fixed coordinates expressed in body's coordinate system as $\mathbf{r}^B = \{1, 0, 0\}$, then we can find where the device is heading in other words the coordinates of the nose in global coordinate systems using (Eq. 1.4):

$$\mathbf{r}^G = \text{DCM}^G \mathbf{r}^B$$

So far you're asking yourself if an accelerometer and a magnetometer gives us the DCM matrix at any point in time, why do we need the gyroscope? The gyroscope is actually a more precise device than the accelerometer and magnetometer are, it is used to "fine-tune" the DCM matrix returned by the accelerometer and magnetometer.

Gyroscopes have no sense of absolute orientation of the device, i.e. they don't know where north is and where zenith is (things that we can find out using the accelerometer and magnetometer), instead if we know the orientation of the device at time t , expressed as a DCM matrix $\text{DCM}(t)$, we can find a more precise orientation $\text{DCM}(t+dt)$ using the gyroscope, then the one estimated directly from the accelerometer and magnetometer direct readings which are subject to a lot of noise in form of external (non-gravitational) inertial forces (i.e. acceleration) or magnetically forces that are not caused by the earth's magnetic field.

These facts call for an algorithm that would combine the readings from all three devices (accelerometer, magnetometer and gyroscope) in order to create our best guess or estimate regarding the device orientation in space (or space's orientation in device's coordinate systems), the two orientations are related since they are simply expressed using two DCM matrices that are transpose of one another ($\text{DCM}^G = \text{DCM}^{BT}$).

We'll now go ahead and introduce such an algorithm.

We'll work with the DCM matrix that consists of the versors of the global (earth's) coordinate system aligned on each row:

$$\begin{aligned} \text{DCM}^G &= [\mathbf{i}^G, \mathbf{j}^G, \mathbf{k}^G] = \begin{bmatrix} \mathbf{I.i} & \mathbf{I.j} & \mathbf{I.k} \\ \mathbf{J.i} & \mathbf{J.j} & \mathbf{J.k} \\ \mathbf{K.i} & \mathbf{K.j} & \mathbf{K.k} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\mathbf{I,i}) & \cos(\mathbf{I,j}) & \cos(\mathbf{I,k}) \\ \cos(\mathbf{J,i}) & \cos(\mathbf{J,j}) & \cos(\mathbf{J,k}) \\ \cos(\mathbf{K,i}) & \cos(\mathbf{K,j}) & \cos(\mathbf{K,k}) \end{bmatrix} = \begin{bmatrix} \mathbf{I}^{BT} \\ \mathbf{J}^{BT} \\ \mathbf{K}^{BT} \end{bmatrix} \end{aligned}$$

If we read the rows of DCM^G we get the vectors \mathbf{I}^B , \mathbf{J}^B , \mathbf{K}^B . We'll work mostly with vectors \mathbf{K}^B (that can be directly estimated by accelerometer) and vector \mathbf{I}^B (that can be directly estimated by the magnetometer). The vector \mathbf{J}^B is simply calculated as $\mathbf{J}^B = \mathbf{K}^B \times \mathbf{I}^B$, since it's orthogonal to the other two vectors (remember versors are unity vectors with same direction as coordinate axes).

Let's say we know the zenith vector expressed in body frame coordinates at time t_0 and we note it as \mathbf{K}^{B_0} . Also let's say we measured our gyro output and we have determined that our angular velocity is $\mathbf{w} = \{w_x, w_y, w_z\}$. Using our gyro we want to know the position of our zenith vector after a small period of time dt has passed we'll note it as $\mathbf{K}^{B_{1G}}$. And we find it using (Eq. 2.6):

$$\mathbf{K}^{B_{1G}} \approx \mathbf{K}^{B_0} + dt \mathbf{v} = \mathbf{K}^{B_0} + dt (\mathbf{w}_g \times \mathbf{K}^{B_0}) = \mathbf{K}^{B_0} + (d\theta_g \times \mathbf{K}^{B_0})$$

Where we noted $d\theta_g = dt \mathbf{w}_g$. Because \mathbf{w}_g is angular velocity as measured by the gyroscope. We'll call $d\theta_g$ angular displacement. In other words it tells us by what **small** angle (given for all 3 axis in form of a vector) has the orientation of a vector \mathbf{K}^B changed during this **small** period of time dt .

Obviously, another way to estimate \mathbf{K}^B is by making another reading from accelerometer so we can get a reading that we note as $\mathbf{K}^{B_{1A}}$.

In practice the values $\mathbf{K}^{B_{1G}}$ will be different from $\mathbf{K}^{B_{1A}}$. One was estimated using our gyroscope and the other was estimated using our accelerometer.

Now it turns out we can go the reverse way and estimate the angular velocity \mathbf{w}_a or angular displacement $d\theta_a = dt \mathbf{w}_a$, from the new accelerometer reading $\mathbf{K}^{B_{1A}}$, we'll use (Eq. 2.5):

$$\mathbf{w}_a = \mathbf{K}^{B_0} \times \mathbf{v}_a / |\mathbf{K}^{B_0}|^2$$

Now $\mathbf{v}_a = (\mathbf{K}^{B_{1A}} - \mathbf{K}^{B_0}) / dt$, and is basically the linear velocity of the vector \mathbf{K}^{B_0} . And $|\mathbf{K}^{B_0}|^2 = 1$, since \mathbf{K}^{B_0} is a unity vector. So we can calculate:

$$d\theta_a = dt \mathbf{w}_a = \mathbf{K}^{B_0} \times (\mathbf{K}^{B_{1A}} - \mathbf{K}^{B_0})$$

The idea of calculating a new estimate \mathbf{K}^{B_1} that combines both $\mathbf{K}^{B_{1A}}$ and $\mathbf{K}^{B_{1G}}$ is to first estimate $d\theta$ as a weighted average of $d\theta_a$ and $d\theta_g$:

$d\theta = (s_a d\theta_a + s_g d\theta_g) / (s_a + s_g)$, we'll discuss about the weights later on , but shortly they are determined and tuned experimentally in order to achieve a desired response rate and noise rejection.

And then \mathbf{K}^{B_1} is calculated similar to how we calculated $\mathbf{K}^{B_{1G}}$:

$$\mathbf{K}^{B_1} \approx \mathbf{K}^{B_0} + (d\theta \times \mathbf{K}^{B_0})$$

Why we went all the way to calculate $d\theta$ and did not apply the weighted average formula directly to $\mathbf{K}^{B_{1A}}$ and $\mathbf{K}^{B_{1G}}$? Because $d\theta$ can be used to calculate the other elements of our DCM matrix in the same way:

$$\mathbf{I}^{B_1} \approx \mathbf{I}^{B_0} + (d\theta \times \mathbf{I}^{B_0})$$

$$\mathbf{J}^{B_1} \approx \mathbf{J}^{B_0} + (d\theta \times \mathbf{J}^{B_0})$$

The idea is that all three versors \mathbf{I}^B , \mathbf{J}^B , \mathbf{K}^B are attached to each other and will follow the same angular displacement $d\theta$ during our small interval dt . So in a nutshell this is the algorithm that allows us to calculate the DCM_1 matrix at time t_1 from our previous estimated DCM_0 matrix at time t_0 . It is applied recursively at regular small time intervals dt and gives us an updated DCM matrix at any point in time. The matrix will not drift too much because it is fixed to the absolute position dictated by the accelerometer and will not be too noisy from external accelerations because we also use the gyroscope data to update it.

So far we didn't mention a word about our magnetometer. One reasons being that it is not available on all IMU units (6DOF) and we can go away without using it, but our resulting orientation will then have a drifting heading (i.e. it will not show if we're heading north, south, west or east), or we can introduce a virtual magnetometer that is always pointing North, to introduce stability in our model. This situation is demonstrated in the accompanying source code that used a 6DOF IMU.

Now we'll show how to integrate magnetometer readings into our algorithm. As it turns out it is really simple since magnetometer is really similar to accelerometer (they even use similar calibration algorithms), the only difference being that instead of estimating the Zenith vector \mathbf{K}^B vector it estimates the vector pointing North \mathbf{I}^B . Following the same logic as we did for our accelerometer we can determine the angular displacement according to the updated magnetometer reading as being:

$$d\theta_m = dt \mathbf{w}_m = \mathbf{I}^{B_0} \times (\mathbf{I}^{B_{1M}} - \mathbf{I}^{B_0})$$

Now let's incorporate it into our weighted average:

$$d\theta = (s_a d\theta_a + s_g d\theta_g + s_m d\theta_m) / (s_a + s_g + s_m)$$

From here we go the same path to calculate the updated DCM_1

$$\mathbf{I}^{B_1} \approx \mathbf{I}^{B_0} + (d\theta \times \mathbf{I}^{B_0}), \mathbf{K}^{B_1} \approx \mathbf{K}^{B_0} + (d\theta \times \mathbf{K}^{B_0}) \text{ and } \mathbf{J}^{B_1} \approx \mathbf{J}^{B_0} + (d\theta \times \mathbf{J}^{B_0}),$$

In practice we'll calculate $\mathbf{J}^{B_1} = \mathbf{K}^{B_1} \times \mathbf{I}^{B_1}$, after correcting \mathbf{K}^{B_1} and \mathbf{I}^{B_1} to be perpendicular unity vectors again , note that all our logic is approximated and dependent on dt being small, the larger the dt the larger the error we'll accumulate.

So if vectors \mathbf{I}^{B_0} , \mathbf{J}^{B_0} , \mathbf{K}^{B_0} form a valid DCM matrix, in other words they are orthogonal to each other and are unity vectors, then we can't say the same about \mathbf{I}^{B_1} , \mathbf{J}^{B_1} , \mathbf{K}^{B_1} , the formulas used for calculating them does not guarantee the orthogonality or length of the vector to be preserved, however we will not get a big error if dt is small, all we need to do is to correct them after each iteration.

First let's see how we can ensure that two vectors are orthogonal again. Let's consider two unity vectors \mathbf{a} and \mathbf{b} that are "almost orthogonal" in other words the angle between these two vectors is close to 90° , but not exactly 90° . We're looking to find a vector \mathbf{b}' that is orthogonal to \mathbf{a} and that is in the same plane formed by the vectors \mathbf{a} and \mathbf{b} . Such a vector is easy to find as shown in Figure 5. First we find vector $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ that by the rules of cross product is orthogonal to both \mathbf{a} and \mathbf{b} and thus is perpendicular to the plane formed by \mathbf{a} and \mathbf{b} . Next the vector $\mathbf{b}' = \mathbf{c} \times \mathbf{a}$ is calculated as the cross product of \mathbf{c} and \mathbf{a} . From the definition of cross product \mathbf{b}' is orthogonal to \mathbf{a} and because it is also orthogonal to \mathbf{c} – it ends up in the plane orthogonal to \mathbf{c} , which is the plane formed by \mathbf{a} and \mathbf{b} . Thus \mathbf{b}' is the corrected vector we're seeking that is orthogonal to \mathbf{a} and belongs to the plane formed by \mathbf{a} and \mathbf{b} .

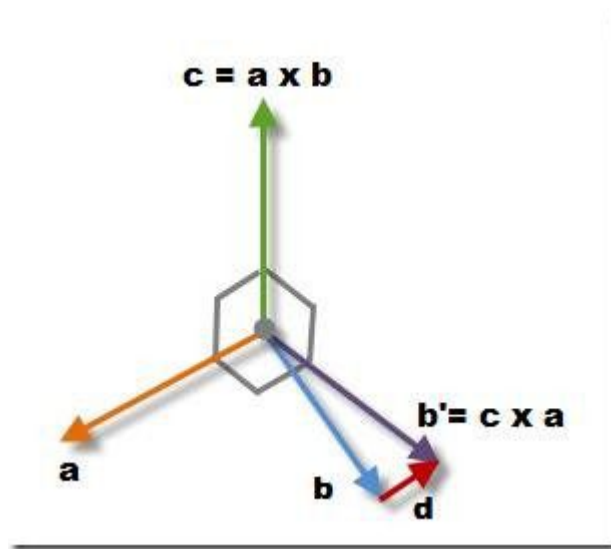


Figure 5

We can extend the equation using [the triple product rule](#) and the fact that $\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2 = 1$:

$\mathbf{b}' = \mathbf{c} \times \mathbf{a} = (\mathbf{a} \times \mathbf{b}) \times \mathbf{a} = -\mathbf{a} (\mathbf{a} \cdot \mathbf{b}) + \mathbf{b} (\mathbf{a} \cdot \mathbf{a}) = \mathbf{b} - \mathbf{a} (\mathbf{a} \cdot \mathbf{b}) = \mathbf{b} + \mathbf{d}$, where $\mathbf{d} = -\mathbf{a} (\mathbf{a} \cdot \mathbf{b})$ (Scenario 1, \mathbf{a} is fixed \mathbf{b} is corrected)

You can reflect a little bit on the results ... So we obtain corrected vector \mathbf{b}' from vector \mathbf{b} by adding a "correction" vector $\mathbf{d} = -\mathbf{a} (\mathbf{a} \cdot \mathbf{b})$. Notice that \mathbf{d} is parallel to \mathbf{a} . Its direction is dependent upon the angle between \mathbf{a} and \mathbf{b} , for example in Figure 5 $\mathbf{a} \cdot \mathbf{b} = \cos(\mathbf{a}, \mathbf{b}) > 0$, because angle between \mathbf{a} and \mathbf{b} is less than 90° thus \mathbf{d} has opposite direction from \mathbf{a} and a magnitude of $\cos(\mathbf{a}, \mathbf{b}) = \sin(\mathbf{b}, \mathbf{b}')$.

In the scenario above we considered that vector **a** is fixed and we found a corrected vector **b'** that is orthogonal to **a**. We can consider the symmetric problem – we fix **b** and find the corrected vector **a'**:

$\mathbf{a}' = \mathbf{a} - \mathbf{b} (\mathbf{b} \cdot \mathbf{a}) = \mathbf{a} - \mathbf{b} (\mathbf{a} \cdot \mathbf{b}) = \mathbf{a} + \mathbf{e}$, where $\mathbf{e} = -\mathbf{b} (\mathbf{a} \cdot \mathbf{b})$ (Scenario 2, **b** is fixed **a** is corrected)

Finally in the third scenario we want both vectors to move towards their corrected state, we consider them both “equally wrong”, so intuitively we apply half correction to both vectors from scenario 1 and 2:

$\mathbf{a}' = \mathbf{a} - \mathbf{b} (\mathbf{a} \cdot \mathbf{b}) / 2$ (Scenario 3, both **a** and **b** are corrected)
 $\mathbf{b}' = \mathbf{b} - \mathbf{a} (\mathbf{a} \cdot \mathbf{b}) / 2$

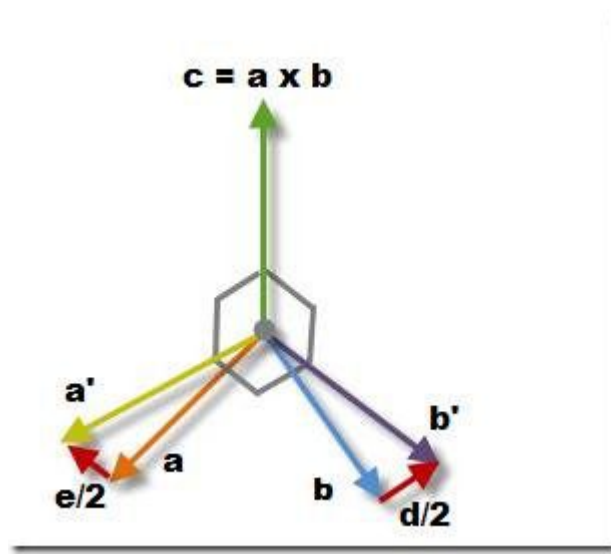


Figure 6

This is an relatively easy formula to calculate on a microprocessor since we can pre-compute $\text{Err} = (\mathbf{a} \cdot \mathbf{b})/2$ and then use it to correct both vectors:

$\mathbf{a}' = \mathbf{a} - \text{Err} * \mathbf{b}$
 $\mathbf{b}' = \mathbf{b} - \text{Err} * \mathbf{a}$

Please note that we're not proving that **a'** and **b'** are orthogonal in Scenario 3, but we presented the intuitive reasoning why the angle between **a'** and **b'** will get closer to 90° if we apply the above corrective transformations.

Now going back to our updated DCM matrix that consists of three vectors \mathbf{I}^{B_1} , \mathbf{J}^{B_1} , we apply the following corrective actions before reintroducing the DCM matrix into the next loop:

$$\text{Err} = (\mathbf{I}^{B_1} \cdot \mathbf{J}^{B_1}) / 2$$

$\mathbf{I}^{B_1'} = \mathbf{I}^{B_1} - \text{Err} * \mathbf{J}^{B_1}$
 $\mathbf{J}^{B_1'} = \mathbf{J}^{B_1} - \text{Err} * \mathbf{I}^{B_1}$
 $\mathbf{I}^{B_1''} = \text{Normalize}[\mathbf{I}^{B_1'}]$

$$\mathbf{J}^{B_1''} = \text{Normalize}[\mathbf{J}^{B_1'}]$$

$$\mathbf{K}^{B_1''} = \mathbf{I}^{B_1''} \times \mathbf{J}^{B_1''}$$

Where $\text{Normalize}[\mathbf{a}] = \mathbf{a} / |\mathbf{a}|$, is the formula calculating the unit vector co-directional with \mathbf{a} .

So finally our corrected DCM_1 matrix can be recomposed from vectors $\mathbf{I}^{B_1''}$, $\mathbf{J}^{B_1''}$, $\mathbf{K}^{B_1''}$ that have been ortho-normalized (each vector constitutes a row of the updated and corrected DCM matrix).

We repeat the loop to find DCM_2 , DCM_3 , or in general DCM_n , at any time interval n .

References

1. [Theory of Applied Robotics: Kinematics, Dynamics, and Control \(Reza N. Jazar\)](#)
2. [Linear Algebra and Its Applications \(David C. Lay\)](#)
3. [Fundamentals of Matrix Computations \(David S. Watkins\)](#)
4. [Direction Cosine Matrix IMU: Theory \(W Premerlani\)](#)

Additional Notes

For the implementation of the algorithm for now see my quadcopter project in particular releases 6/7 have a nice [Processing](#) program for visual display of the DCM matrix and a model plane. The entire code is on SVN repository:

<https://code.google.com/p/picquadcontroller/source/browse/trunk/>

The code is in imu.h file:

<https://code.google.com/p/picquadcontroller/source/browse/trunk/imu.h>

A PDF Version of this article is available here

[DCM Tutorial – An Introduction to Orientation Kinematics by Starlino \(PDF, Rev 0.1 Draft\)](#)

Please mention and link to the source when using information in this article:

http://www.starlino.com/dcm_tutorial.html

Starlino Electronics // Spring , 2011

[Comments \(99\)](#)