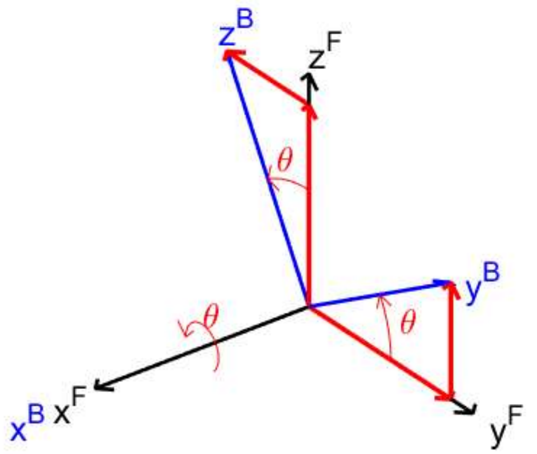
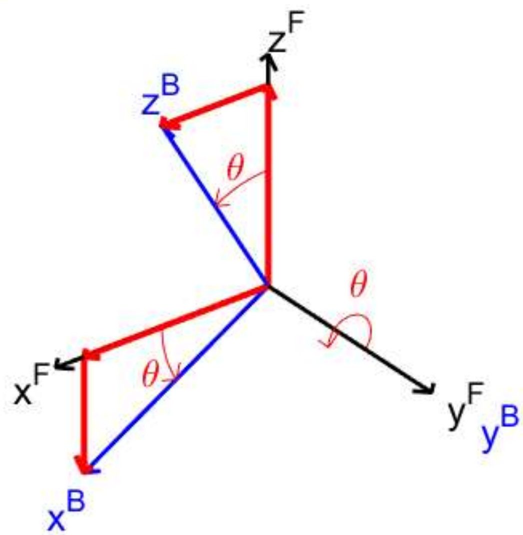
Euler Angles

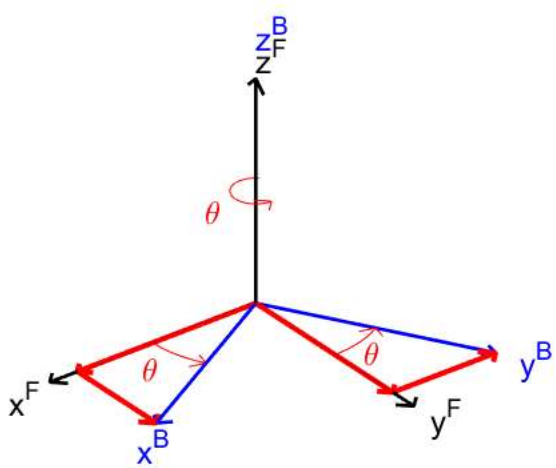
Monday, December 2, 2019 8:07 AM



$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}}_{R_1(\theta)} \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$



$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}}_{R_2(\theta)} \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$



$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_3(\theta)} \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$

Direction Cosine Matrix (DCM)

Monday, December 2, 2019 8:39 AM

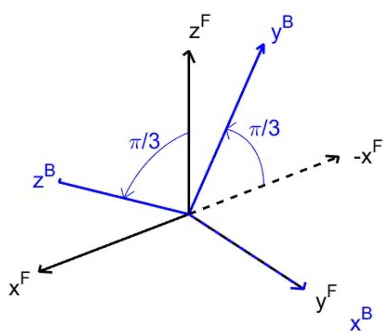
Find the components of the fixed axes along the body axes. The component of x^F along x^B, use the dot product. Similarly for all other axes.

$$
\begin{aligned}
\hat{x}^F \cdot \hat{x}^B &= 1 \cdot 1 \cdot \cos(\theta_{x^F x^B}) \\
\hat{y}^F \cdot \hat{x}^B &= 1 \cdot 1 \cdot \cos(\theta_{y^F x^B}) \\
\hat{z}^F \cdot \hat{x}^B &= 1 \cdot 1 \cdot \cos(\theta_{z^F x^B}) \\
\hat{x}^F \cdot \hat{y}^B &= 1 \cdot 1 \cdot \cos(\theta_{x^F y^B}) \\
&\vdots \\
\hat{z}^F \cdot \hat{z}^B &= 1 \cdot 1 \cdot \cos(\theta_{z^F z^B})
\end{aligned}
$$

$$
\begin{bmatrix} \hat{x}^B \\ \hat{y}^B \\ \hat{z}^B \end{bmatrix} = \begin{bmatrix} \cos(\theta_{x^F x^B}) & \cos(\theta_{y^F x^B}) & \cos(\theta_{z^F x^B}) \\ \cos(\theta_{x^F y^B}) & \cos(\theta_{y^F y^B}) & \cos(\theta_{z^F y^B}) \\ \cos(\theta_{x^F z^B}) & \cos(\theta_{y^F z^B}) & \cos(\theta_{z^F z^B}) \end{bmatrix} \begin{bmatrix} \hat{x}^F \\ \hat{y}^F \\ \hat{z}^F \end{bmatrix}
$$

Direction Cosine Matrix (DCM) is a 3x3 matrix of cosines

Example:



$$
\begin{bmatrix} \hat{x}^B \\ \hat{y}^B \\ \hat{z}^B \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) & \cos(0) & \cos(\pi/2) \\ \cos(2\pi/3) & \cos(\pi/2) & \cos(\pi/6) \\ \cos(\pi/6) & \cos(\pi/2) & \cos(\pi/3) \end{bmatrix} \begin{bmatrix} \hat{x}^F \\ \hat{y}^F \\ \hat{z}^F \end{bmatrix}
$$

$$
\begin{bmatrix} \hat{x}^B \\ \hat{y}^B \\ \hat{z}^B \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1/2 & 0 & \sqrt{3}/2 \\ \sqrt{3}/2 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} \hat{x}^F \\ \hat{y}^F \\ \hat{z}^F \end{bmatrix}
$$

Properties of a valid DCM:

All rows/columns are orthogonal:
Dot product between any 2 rows or any 2 columns must be 0. The DCM is "Orthogonal"

The Determinate of the DCM is 1.
The DCM is "Normal"
The DCM is "Ortho-normal"

DCM*e = e
(DCM-I)*e = 0
1 eigenvalue of the DCM must be 1 such that the vector e exists.

DCM to Euler Angles:

$$DCM = R_{313} = R_3(\theta_3)R_1(\theta_2)R_3(\theta_1)$$

$$DCM = \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_2) & \sin(\theta_2) \\ 0 & -\sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM = \begin{bmatrix} \cos(\theta_3) & \cos(\theta_2)\sin(\theta_3) & \sin(\theta_2)\sin(\theta_3) \\ -\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) & \sin(\theta_2)\cos(\theta_3) \\ 0 & -\sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM = \begin{bmatrix} \cos(\theta_1)\cos(\theta_3) - \sin(\theta_1)\cos(\theta_2)\sin(\theta_3) & \sin(\theta_1)\cos(\theta_3) + \cos(\theta_1)\cos(\theta_2)\sin(\theta_3) & \sin(\theta_2)\sin(\theta_3) \\ -\cos(\theta_1)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_2)\cos(\theta_3) & -\sin(\theta_1)\sin(\theta_3) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_3) & \sin(\theta_2)\cos(\theta_3) \\ \sin(\theta_1)\sin(\theta_2) & -\cos(\theta_1)\sin(\theta_2) & \cos(\theta_2) \end{bmatrix}$$

In general:

$$\begin{aligned}
&1\,\text{term}: \pm\sin(\theta_2) \quad \text{or} \quad +\cos(\theta_2) \\
&2\,\text{terms}: \pm\sin(\theta_1)\cos(\theta_2) \quad \text{or} \quad \pm\sin(\theta_1)\sin(\theta_2) \\
&\qquad\qquad \pm\cos(\theta_1)\cos(\theta_2) \quad \text{or} \quad \pm\cos(\theta_1)\sin(\theta_2) \\
&2\,\text{terms}: \pm\sin(\theta_3)\cos(\theta_2) \quad \text{or} \quad \pm\sin(\theta_3)\sin(\theta_2) \\
&\qquad\qquad \pm\cos(\theta_3)\cos(\theta_2) \quad \text{or} \quad \pm\cos(\theta_3)\sin(\theta_2) \\
&4\,\text{terms}: \pm(\theta_1)(\theta_3) \pm (\theta_1)(\theta_2)(\theta_3)
\end{aligned}$$

Only need these 4 terms for the DCM to Euler Angles problem if you have a singularity, e.g., $\sin(\theta_2)$ or $\cos(\theta_2) = 0$

For this sequence: 3-1-3

$$\cos(\theta_2) = DCM(3,3)$$

$$\theta_2 = \cos^{-1} DCM(3,3)$$

$$\tan(\theta_1) = \frac{\sin(\theta_1)}{\cos(\theta_1)} = \frac{\sin(\theta_1)\sin(\theta_2)}{\cos(\theta_1)\sin(\theta_2)} = \frac{DCM(3,1)}{-DCM(3,2)}$$

$$\tan(\theta_3) = \frac{\sin(\theta_3)}{\cos(\theta_3)} = \frac{\sin(\theta_3)\sin(\theta_2)}{\cos(\theta_3)\sin(\theta_2)} = \frac{DCM(1,3)}{DCM(2,3)}$$

as long as $\sin(\theta_2) \neq 0$ which is the singularity that you would need the other 4 terms for.

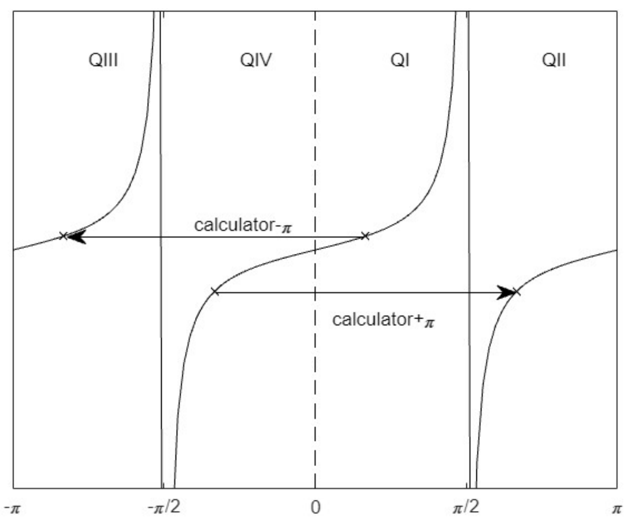
Use inverse tangents for θ_1 and θ_3 with quadrant checks. Use atan2 in MATLAB to do the quadrant check for you. If doing the inverse tangent on your calculator then:

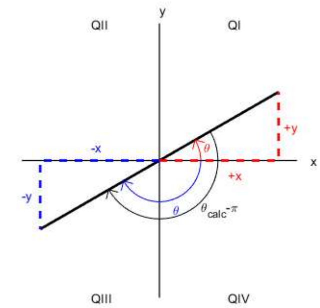
$$\tan^{-1}\left(\frac{+y}{+x}\right) = \text{QI (calculator answer)}$$

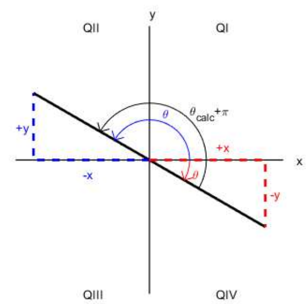
$$\tan^{-1}\left(\frac{+y}{-x}\right) = \text{QII (calculator answer} + \pi)$$

$$\tan^{-1}\left(\frac{-y}{-x}\right) = \text{QIII (calculator answer} - \pi)$$

$$\tan^{-1}\left(\frac{-y}{+x}\right) = \text{QIV (calculator answer)}$$

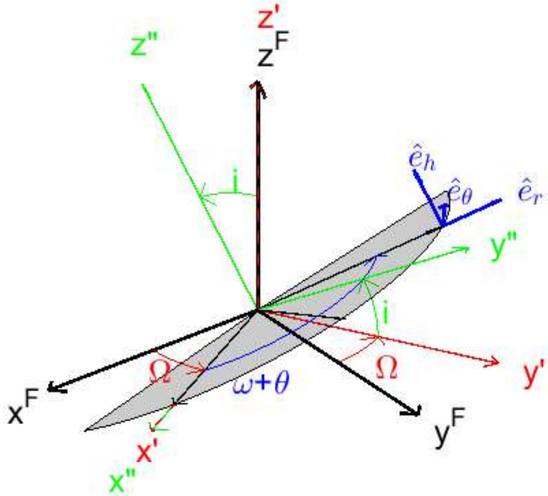






Euler stated that any new orientation can be achieved in no more than 3 base rotations.

Example: Space Mechanics



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_3(\Omega) \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = R_1(i) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} \hat{e}_r \\ \hat{e}_\theta \\ \hat{e}_h \end{bmatrix} = R_3(\omega+\theta) \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix}$$

Rotate from the fixed frame, F, to the prime frame by a rotation about z^F through angle Right Ascension, Ω.

Then rotate about x' axis by inclination, i, such that z' moves to then angular momentum vector and y' moves up from the x-y plane to the orbit plane. This new frame the double-prime frame.

Finally, rotate about z" by the angle argument of periapse plus true anomaly, also known as the argument of latitude, until x" lines up with the radial direction and y" lines up with the transverse direction.

Now, using back substitution we have: $\begin{bmatrix} \hat{e}_r \\ \hat{e}_\theta \\ \hat{e}_h \end{bmatrix} = R_3(\omega+\theta)R_1(i)R_3(\Omega) \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix} = R_{313}(\bar{\theta}) \begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$

Where the Euler Angles are $\bar{\theta} = \begin{bmatrix} \Omega \\ i \\ \omega+\theta \end{bmatrix}$ in this example

There are 12 possible sequences:

1-2-3	1-2-1
1-3-2	1-3-1
2-1-3	2-1-2
2-3-1	2-3-2
3-1-2	3-1-3
3-2-1	3-2-3

$$R_{123}(\bar{\theta}) = R_3(\theta_3)R_2(\theta_2)R_1(\theta_1)$$
$$R_{121}(\bar{\theta}) = R_1(\theta_3)R_2(\theta_2)R_1(\theta_1)$$
$$R_{132}(\bar{\theta}) = R_2(\theta_3)R_3(\theta_2)R_1(\theta_1)$$
$$R_{131}(\bar{\theta}) = R_1(\theta_3)R_3(\theta_2)R_1(\theta_1)$$
$$\vdots$$
$$R_{321}(\bar{\theta}) = R_1(\theta_3)R_2(\theta_2)R_3(\theta_1)$$
$$R_{323}(\bar{\theta}) = R_3(\theta_3)R_2(\theta_2)R_3(\theta_1)$$

Rotation matrix from the Fixed Frame to the Body Frame known as the Direction Cosine Matrix, DCM. Multiply 3 base rotation matrices together to get the DCM. Each multiplication of a 3x3 matrix with a 3x3 matrix requires 27 multiplications and 18 additions.

To return the Euler Angles from a DCM:

$$R_{313}(\bar{\theta}) = R_3(\theta_3)R_1(\theta_2)R_3(\theta_1) = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_2 & \sin\theta_2 \\ 0 & -\sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{313}(\bar{\theta}) = \begin{bmatrix} \cos\theta_3 & \cos\theta_2\sin\theta_3 & \sin\theta_2\sin\theta_3 \\ -\sin\theta_3 & \cos\theta_2\cos\theta_3 & \sin\theta_2\cos\theta_3 \\ 0 & -\sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{313}(\bar{\theta}) = DCM = \begin{bmatrix} \cos\theta_1\cos\theta_3 - \sin\theta_1\cos\theta_2\sin\theta_3 & \sin\theta_1\cos\theta_3 + \cos\theta_1\cos\theta_2\sin\theta_3 & \sin\theta_2\sin\theta_3 \\ -\cos\theta_1\sin\theta_3 - \sin\theta_1\cos\theta_2\cos\theta_3 & -\sin\theta_1\sin\theta_3 + \cos\theta_1\cos\theta_2\cos\theta_3 & \sin\theta_2\cos\theta_3 \\ \sin\theta_1\sin\theta_2 & -\cos\theta_1\sin\theta_2 & \cos\theta_2 \end{bmatrix}$$

$$\cos\theta_2 = DCM(3,3)$$

$$\theta_2 = \cos^{-1}(DCM(3,3))$$

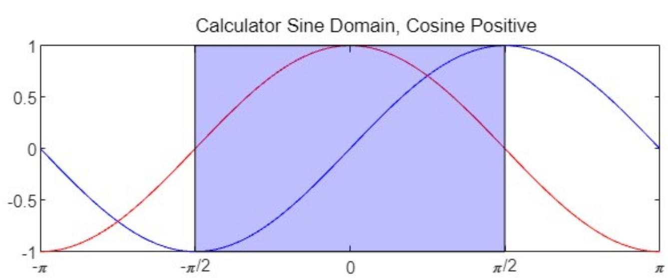
$$\tan\theta_1 = \frac{\sin\theta_1}{\cos\theta_1} = \frac{\sin\theta_1\sin\theta_2}{\cos\theta_1\sin\theta_2} = \frac{DCM(3,1)}{-DCM(3,2)}$$

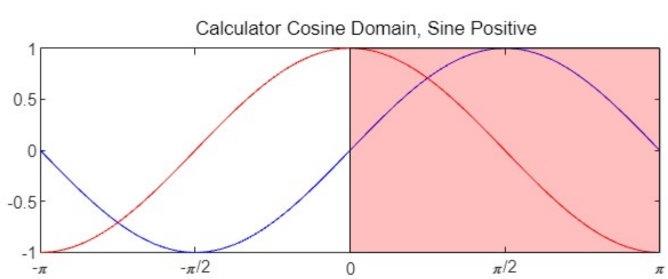
$$\theta_1 = \tan^{-1}\left(\frac{DCM(3,1)}{-DCM(3,2)}\right)$$

$$\tan\theta_3 = \frac{\sin\theta_3}{\cos\theta_3} = \frac{\sin\theta_3\sin\theta_2}{\cos\theta_3\sin\theta_2} = \frac{DCM(1,3)}{DCM(2,3)}$$

$$\theta_3 = \tan^{-1}\left(\frac{DCM(1,3)}{DCM(2,3)}\right)$$

12 sequences so 12 different sets of formulas. Always inverse sine or cosine for θ_2. Always calculator answer such that opposite trig function, cosine or sine, is positive:





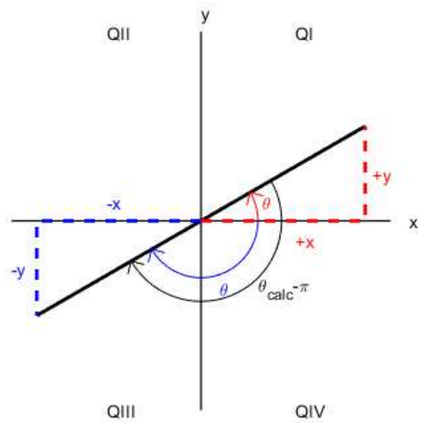
Use inverse tangents for θ_1 and θ_3 with quadrant checks. Use atan2 in MATLAB to do the quadrant check for you. If doing the inverse tangent on your calculator then:

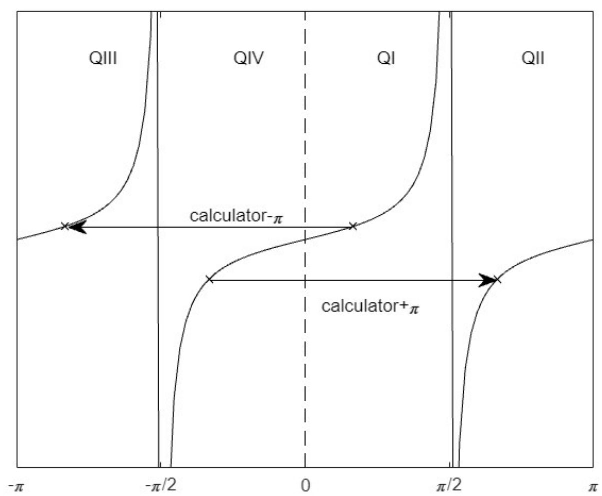
$$\tan^{-1}\left(\frac{+y}{+x}\right) = \text{QI (calculator answer)}$$

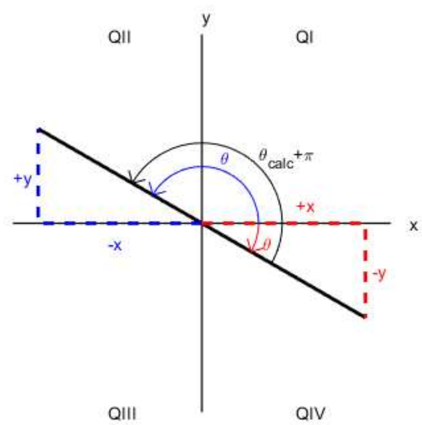
$$\tan^{-1}\left(\frac{+y}{-x}\right) = \text{QII (calculator answer}+\pi)$$

$$\tan^{-1}\left(\frac{-y}{-x}\right) = \text{QIII (calculator answer}-\pi)$$

$$\tan^{-1}\left(\frac{-y}{+x}\right) = \text{QIV (calculator answer)}$$







Euler Parameters (e,ϕ)

Monday, December 2, 2019 8:48 AM

Euler reasoned that any orientation could be visualized as a single rotation about an arbitrary axis, e (the Euler Axis), through an angle, ϕ (the Euler Angle).
Thus:

$$DCM\overline{e} = \overline{e}$$

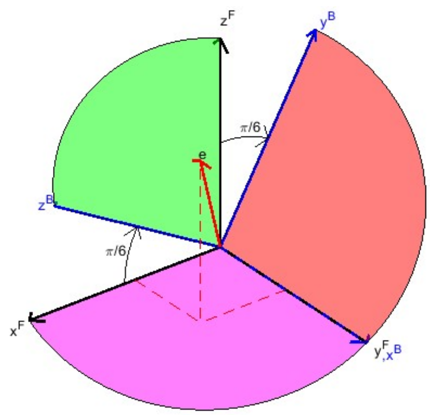
$$\left(DCM - I_{3x3}\right)\overline{e} = \overline{0}$$

General eigenvector/eigenvalue problem:

$$[A]\overline{v} = \lambda\overline{v}$$

This means that it is an eigenvector/eigenvalue problem where the eigenvalue is 1.

Example:



$$\left(\begin{bmatrix} 0 & 1 & 0 \\ -1/2 & 0 & \sqrt{3}/2 \\ \sqrt{3}/2 & 0 & 1/2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\overline{e} = \overline{0}$$

$$\begin{bmatrix} -1 & 1 & 0 \\ -1/2 & -1 & \sqrt{3}/2 \\ \sqrt{3}/2 & 0 & -1/2 \end{bmatrix}\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Let e_1=1 and solve for e_2 and e_3:
Row 1:

$$e_2 = e_1 = 1$$

Row 3:

$$\left(\sqrt{3}/2\right)e_1 - (1/2)e_3 = 0$$

$$e_3 = \sqrt{3}$$

Check the row we didn't use, row 2 for this example:

$$(-1/2)e_1 - e_2 + \left(\sqrt{3}/2\right)e_3 = 0$$

$$(-1/2) - 1 + \left(\sqrt{3}/2\right)\sqrt{3} = 0 = 0$$

It checks out which means we did the algebra correctly to find $\bar{e}$.
Now, put the 3 components together in a vector and normalize so that it is a unit vector:

$$\overline{e} = \frac{1}{\|\;\|}\begin{bmatrix} 1 \\ 1 \\ \sqrt{3} \end{bmatrix} = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 1 \\ \sqrt{3} \end{bmatrix}$$

Next, we need to find the Euler Angle, ϕ.
It turns out, as we will prove later, that the sum of the main diagonal components of the DCM, also known as the "trace", is:

$$\text{trace}(DCM) = 1 + 2\cos\phi$$

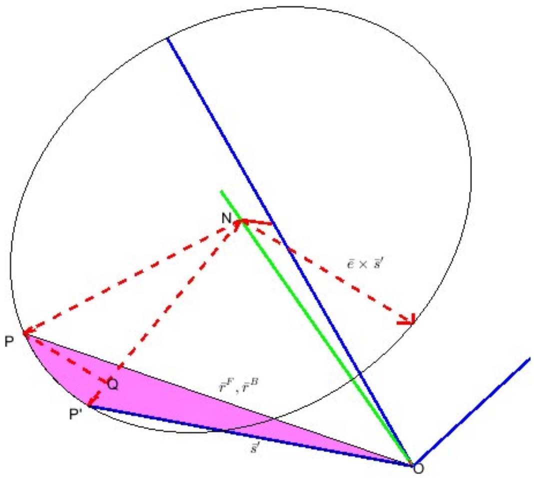
(This is also true of the 3 base rotation matrices)
For this example we have:

$$1/2 = 1 + 2\cos\phi$$

$$\cos\phi = -1/4$$

$$\phi = \pm 1.823\text{rad}$$

Now, how do we determine which sign is correct?
First, we need to derive the DCM in terms of the Euler Parameters:



The fixed frame rotates to a body frame denoted by the vector $\bar{s}'$. We need to determine the position vector relative to the body frame:

$$\bar{r}^B = DCM\bar{r}^F$$

$$\bar{r}^B = \bar{r}^B_{//} + \bar{r}^B_{\perp}$$

$$\bar{r}^B = \overline{ON} + \overline{NQ} + \overline{QP}$$

$$\bar{r}^B = \left(\bar{e}\cdot\bar{r}^F\right)\bar{e} + \left(\cos\phi\bar{r}^F_{\perp} - \sin\phi\left(\bar{e}\times\bar{r}^F_{\perp}\right)\right)$$

$$\bar{r}^F_{\perp} = \bar{r}^F - \bar{r}^F_{//} = \bar{r}^F - \left(\bar{e}\cdot\bar{r}^F\right)\bar{e}$$

$$\bar{r}^F_{\perp} = \bar{r}^F - \left(x^F e_1 + y^F e_2 + z^F e_3\right)\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \bar{r}^F - \begin{bmatrix} e_1^2 & e_1e_2 & e_1e_3 \\ e_1e_2 & e_2^2 & e_2e_3 \\ e_1e_3 & e_2e_3 & e_3^2 \end{bmatrix}\bar{r}^F = \begin{bmatrix} 1-e_1^2 & -e_1e_2 & -e_1e_3 \\ -e_1e_2 & 1-e_2^2 & -e_2e_3 \\ -e_1e_3 & -e_2e_3 & 1-e_3^2 \end{bmatrix}\bar{r}^F$$

$$\left(\bar{e}\times\bar{r}^F_{\perp}\right) = \begin{bmatrix} x^F & y^F & z^F \\ e_1 & e_2 & e_3 \\ \left(1-e_1^2\right)x^F - e_1e_2y^F - e_1e_3z^F & -e_1e_2x^F + \left(1-e_2^2\right)y^F - e_2e_3z^F & -e_1e_3x^F - e_2e_3y^F + \left(1-e_3^2\right)z^F \end{bmatrix}$$

$$\left(\bar{e}\times\bar{r}^F_{\perp}\right) = \begin{bmatrix} -e_1e_2e_3x^F - e_2^2e_3y^F + e_2\left(1-e_3^2\right)z^F + e_1e_2e_3x^F - e_3\left(1-e_2^2\right)y^F + e_2e_3^2z^F \\ e_3\left(1-e_1^2\right)x^F - e_1e_2e_3y^F - e_1e_3^2z^F + e_1^2e_3x^F + e_1e_2e_3z^F - e_1\left(1-e_3^2\right)z^F \\ -e_1^2e_2x^F + e_1\left(1-e_2^2\right)y^F - e_1e_2e_3z^F - e_2\left(1-e_1^2\right)x^F + e_1e_2^2y^F + e_1e_2e_3z^F \end{bmatrix}$$

$$\left(\bar{e}\times\bar{r}^F_{\perp}\right) = \begin{bmatrix} \cancel{-e_1e_2e_3x^F}\ \cancel{-e_2^2e_3y^F} + e_2\left(1\cancel{-e_3^2}\right)z^F\ \cancel{+e_1e_2e_3x^F} - e_3\left(1\cancel{-e_2^2}\right)y^F\ \cancel{+e_2e_3^2z^F} \\ e_3\left(1\cancel{-e_1^2}\right)x^F\ \cancel{-e_1e_2e_3y^F}\ \cancel{-e_1e_3^2z^F}\ \cancel{+e_1^2e_3x^F}\ \cancel{+e_1e_2e_3y^F} - e_1\left(1\cancel{-e_3^2}\right)z^F \\ \cancel{-e_1^2e_2x^F} + e_1\left(1\cancel{-e_2^2}\right)y^F\ \cancel{-e_1e_2e_3z^F} - e_2\left(1\cancel{-e_1^2}\right)x^F\ \cancel{+e_1e_2^2y^F}\ \cancel{+e_1e_2e_3z^F} \end{bmatrix}$$

$$\left(\bar{e}\times\bar{r}^F_{\perp}\right) = \begin{bmatrix} e_2z^F - e_3y^F \\ e_3x^F - e_1z^F \\ e_1y^F - e_2x^F \end{bmatrix} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}\bar{r}^F$$

$$\bar{r}^B = \left(\bar{e}\cdot\bar{r}^F\right)\bar{e} + \left(\cos\phi\bar{r}^F_{\perp} - \sin\phi\left(\bar{e}\times\bar{r}^F_{\perp}\right)\right)$$

$$\bar{r}^B = \begin{bmatrix} e_1^2 & e_1e_2 & e_1e_3 \\ e_1e_2 & e_2^2 & e_2e_3 \\ e_1e_3 & e_2e_3 & e_3^2 \end{bmatrix}\bar{r}^F + \cos\phi\begin{bmatrix} 1-e_1^2 & -e_1e_2 & -e_1e_3 \\ -e_1e_2 & 1-e_2^2 & -e_2e_3 \\ -e_1e_3 & -e_2e_3 & 1-e_3^2 \end{bmatrix}\bar{r}^F - \sin\phi\begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}\bar{r}^F$$

$$\bar{r}^B = \begin{bmatrix} \cos\phi + e_1^2(1-\cos\phi) & e_1e_2(1-\cos\phi)+e_3\sin\phi & e_1e_3(1-\cos\phi)-e_2\sin\phi \\ e_1e_2(1-\cos\phi)-e_3\sin\phi & \cos\phi+e_2^2(1-\cos\phi) & e_2e_3(1-\cos\phi)+e_1\sin\phi \\ e_1e_3(1-\cos\phi)+e_2\sin\phi & e_2e_3(1-\cos\phi)-e_1\sin\phi & \cos\phi+e_3^2(1-\cos\phi) \end{bmatrix} \bar{r}^F$$

$$DCM(\bar{e},\phi) = \begin{bmatrix} \cos\phi + e_1^2(1-\cos\phi) & e_1e_2(1-\cos\phi)+e_3\sin\phi & e_1e_3(1-\cos\phi)-e_2\sin\phi \\ e_1e_2(1-\cos\phi)-e_3\sin\phi & \cos\phi+e_2^2(1-\cos\phi) & e_2e_3(1-\cos\phi)+e_1\sin\phi \\ e_1e_3(1-\cos\phi)+e_2\sin\phi & e_2e_3(1-\cos\phi)-e_1\sin\phi & \cos\phi+e_3^2(1-\cos\phi) \end{bmatrix}$$

$$DCM(\bar{e},\phi) = \cos\phi I_{3x3} + (1-\cos\phi)\bar{e}\bar{e}^T + \sin\phi \begin{bmatrix} 0 & e_3 & -e_2 \\ -e_3 & 0 & e_1 \\ e_2 & -e_1 & 0 \end{bmatrix}$$

Notice that the trace of the DCM is:

$$\text{trace}(DCM) = 3\cos\phi + \left(e_1^2+e_2^2+e_3^2\right)(1-\cos\phi) = 3\cos\phi + 1(1-\cos\phi) = 1+2\cos\phi$$

Back to the example problem; Which ϕ do we use?

Since we used an inverse cosine to determine ϕ, then we need to use a sine to determine the correct quadrant. This means we have to use an off-diagonal component of the DCM to determine which ϕ to use. Let's use DCM(1,2) = 1.

$$DCM(1,2) = 1 = e_1e_2(1-\cos\phi) + e_3\sin\phi$$

$$DCM(1,2) = 1 = \frac{1}{\sqrt{5}}\frac{1}{\sqrt{5}}\left(1-\left(-\frac{1}{4}\right)\right) + \frac{\sqrt{3}}{\sqrt{5}}\sin(\pm 1.823\text{rad})$$

$$DCM(1,2) = 1 = \frac{1}{4} + \frac{\sqrt{3}}{\sqrt{5}}\left(\pm\frac{\sqrt{15}}{4}\right)$$

$$DCM(1,2) = 1 = \frac{1}{4} \pm \frac{3}{4} = 1 \quad \text{or} \quad -\frac{1}{2}$$

$$\phi = +1.823\text{rad}$$

$$\bar{e} = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 1 \\ \sqrt{3} \end{bmatrix}$$

Euler Parameters without solving the eigenvalue/eigenvector problem:

First solve for ϕ, using the trace of the DCM as normal.

$$\text{trace}(DCM) = 1+2\cos\phi$$

Now manipulate the off-diagonal terms of *DCM(e,ϕ)* to determine the values of e based on the ϕ you have already determined (just use the calculator answer for ϕ for convenience.

$$DCM(2,3) - DCM(3,2) = 2e_1\sin(\phi)$$

$$e_1 = \frac{1}{2\sin\phi}\left(DCM(2,3) - DCM(3,2)\right)$$

$$e_2 = \frac{1}{2\sin\phi}\left(DCM(3,1) - DCM(1,3)\right)$$

$$e_3 = \frac{1}{2\sin\phi}\left(DCM(1,2) - DCM(2,1)\right)$$

Quaternion

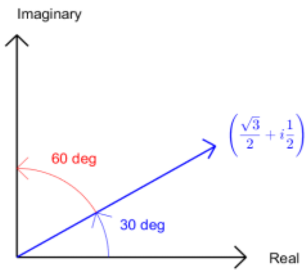
Friday, December 20, 2019 12:31 PM

The complex plane can be used to complete 2D rotations. Euler, and other mathematicians knew this.
Example:

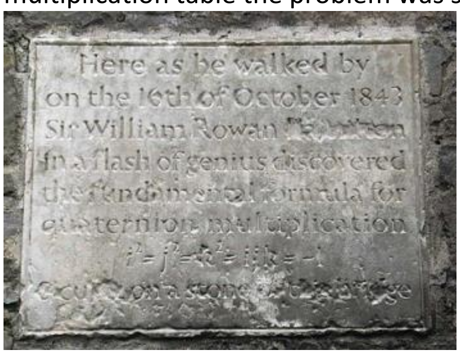
$$30^\circ = \frac{\sqrt{3}}{2} + i\frac{1}{2}$$

$$60^\circ = \frac{1}{2} + i\frac{\sqrt{3}}{2}$$

$$\left(\frac{\sqrt{3}}{2} + i\frac{1}{2}\right)\left(\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) = \frac{\sqrt{3}}{4} + i\frac{3}{4} + i\frac{1}{4} - \frac{\sqrt{3}}{4} = i = 90^\circ$$



The question then became, how can this mathematics be extended to rotations in 3D space?
Many mathematicians were trying to figure out this problem by adding another real or imaginary axis to the complex plane. However, on October 16, 1843, while on a walk with his wife, Sir William Rowan Hamilton had a "flash of genius" by realizing that if he added two additional imaginary axes with a new multiplication table the problem was solved. This is the origin story of quaternions.



Side notes on Hamilton:
Irish mathematician, 1805-1865, that developed Hamiltonian Mechanics, the Lagrangian and Lagrange's Equations, the term "vector", the vector cross product and dot product, and the quaternion.

Quaternion multiplication is NOT commutative, e.g., $a * b \neq b * a$ which was ground-breaking at the time and really opened up the field of mathematics. The quaternion contains the "vector" part that includes 3 distinct imaginary values: i, j, and k, and the "scalar", real, part. Most mathematicians, and math software including MATLAB, have the scalar first, followed by the vector part:

$$\bar{q} = q_0 + iq_1 + jq_2 + kq_3$$

However, in the field of spacecraft attitude we use the scalar last, as q_4:

$$\bar{q} = iq_1 + jq_2 + kq_3 + q_4$$

When we multiply two quaternions, to perform a 3D rotation, we follow the quaternion multiplication table based on Hamilton's famous carving in the bridge:

$\otimes$	i	j	k	1
i	-1	k	-j	i
j	-k	-1	i	j
k	j	-i	-1	k
1	i	j	k	1

So, in general, if we have an initial attitude $\bar{q}$ and a rotation $\bar{q}'$, then the resulting attitude is:

$$\bar{q}''=\bar{q}\otimes\bar{q}'=\left(iq_1+jq_2+kq_3+q_4\right)\otimes\left(iq'_1+jq'_2+kq'_3+q'_4\right)$$

$$\bar{q}''=\begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

This matrix equation eliminates the imaginary components, i, j, and k, and displays the rotation as regular multiplication requiring only 16 multiplications for a 3D rotation. When using Euler Angles / DCM it requires the multiplication of a 3x3 matrix times another 3x3 matrix that requires 27 multiplications. This efficiency is why quaternions are so widely used, not just in spacecraft attitude, but in any field that requires 3d rotations like gaming.

Now we need to know how we get the quaternion from other parameterizations and vis versa. We start with the Euler Parameters. By definition we have:

$$\bar{q}=\begin{bmatrix} \hat{q} \\ q_4 \end{bmatrix}=\begin{bmatrix} \bar{e}\sin\left(\dfrac{\phi}{2}\right) \\ \cos\left(\dfrac{\phi}{2}\right) \end{bmatrix}$$

It is significant that it uses half-angles, $\phi/2$, since now the quaternion is unique for 2 full revolutions, not 1. It is also significant in that the **quaternion is a unit vector**. Then the Euler Parameters given the quaternion:

$$\phi=2\cos^{-1}q_4$$

$$\bar{e}=\frac{\hat{q}}{\sin\left(\dfrac{\phi}{2}\right)}$$

Next, quaternion to DCM:

The derivation starts with the DCM in terms of the Euler Parameters:

$$DCM\left(\bar{e},\phi\right)=\cos\phi I_{3x3}+\left(1-\cos\phi\right)\bar{e}\bar{e}^T+\sin\phi\begin{bmatrix} 0 & e_3 & -e_2 \\ -e_3 & 0 & e_1 \\ e_2 & -e_1 & 0 \end{bmatrix}$$

We also will need double-angle formulas:

$$\sin\phi=2\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\phi}{2}\right)$$

$$\cos\phi=\cos^2\left(\frac{\phi}{2}\right)-\sin^2\left(\frac{\phi}{2}\right)$$

Starting with the first term, since the magnitude of the Euler Axis is 1, the magnitude of $\hat{q}$ is $\sin\frac{\phi}{2}$ thus:

$$\cos\phi=q_4^2-\hat{q}^T\hat{q}$$

The second term:

$$(1-\cos\phi)\bar{e}\bar{e}^T = \underbrace{\hat{q}\hat{q}^T}_{\bar{e}\bar{e}^T\sin^2\left(\frac{\phi}{2}\right)=\bar{e}\bar{e}^T\left(1-\cos^2\left(\frac{\phi}{2}\right)\right)} + \underbrace{\hat{q}\hat{q}^T}_{\bar{e}\bar{e}^T\sin^2\left(\frac{\phi}{2}\right)}$$

$$\left(1-\cos^2\left(\frac{\phi}{2}\right)+\sin^2\left(\frac{\phi}{2}\right)\right)\bar{e}\bar{e}^T = 2\hat{q}\hat{q}^T$$

Finally, the last term:

$$\sin\phi\begin{bmatrix} 0 & e_3 & -e_2 \\ -e_3 & 0 & e_1 \\ e_2 & -e_1 & 0 \end{bmatrix} = 2\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\phi}{2}\right)\begin{bmatrix} 0 & e_3 & -e_2 \\ -e_3 & 0 & e_1 \\ e_2 & -e_1 & 0 \end{bmatrix}$$

$$= 2\cos\left(\frac{\phi}{2}\right)\begin{bmatrix} 0 & e_3\sin\left(\frac{\phi}{2}\right) & -e_2\sin\left(\frac{\phi}{2}\right) \\ -e_3\sin\left(\frac{\phi}{2}\right) & 0 & e_1\sin\left(\frac{\phi}{2}\right) \\ e_2\sin\left(\frac{\phi}{2}\right) & -e_1\sin\left(\frac{\phi}{2}\right) & 0 \end{bmatrix}$$

$$= 2q_4\begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix}$$

Thus the full DCM is:

$$DCM(\bar{q}) = \left(q_4^2 - \hat{q}^T\hat{q}\right)I_{3x3} + 2\hat{q}\hat{q}^T + 2q_4\begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix}$$

$$DCM(\bar{q}) = \begin{bmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right) \\ 2\left(q_1q_2-q_3q_4\right) & q_4^2-q_1^2+q_2^2-q_3^2 & 2\left(q_2q_3+q_1q_4\right) \\ 2\left(q_1q_3+q_2q_4\right) & 2\left(q_2q_3-q_1q_4\right) & q_4^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

If you need to find the quaternion directly from the DCM, not going through the Euler Parameters, then you can manipulate the on-diagonal and off-diagonal terms very similarly to finding the Euler Parameters from the DCM not through the eigenvalue/eigenvector problem.
Starting with the trace of the DCM:

$$\text{trace}(DCM(\bar{q})) = 3q_4^2 - q_1^2 - q_2^2 - q_3^2 = 3q_4^2 - \left(1-q_4^2\right) = 4q_4^2 - 1$$

$$q_4 = \pm\frac{1}{2}\sqrt{\text{trace}(DCM(\bar{q}))+1}$$

$$q_4 = +\frac{1}{2}\sqrt{\text{trace}(DCM(\bar{q}))+1}$$

We choose to use the positive square root answer, however, this means we need to use q_4 to calculate $\hat{q}$ so that it has the correct sign for the sign we have assumed for q_4. Manipulating the off-diagonal terms:

$$DCM(2,3) - DCM(3,2) = 4q_1q_4$$

$$q_1 = \frac{1}{4q_4}\left(DCM(2,3) - DCM(3,2)\right)$$

$$q_2 = \frac{1}{4q_4}\left(DCM(3,1) - DCM(1,3)\right)$$

$$q_3 = \frac{1}{4q_4}\left(DCM(1,2) - DCM(2,1)\right)$$

This set of formulas doesn't work if $q_4=0$.

Then you have to manipulate the on-diagonal terms differently than just a sum to solve for a different component first. For example:

$$DCM(1,1) - DCM(2,2) - DCM(3,3) = 3q_1^2 - \left(q_2^2 + q_3^2 + q_4^2\right) = 3q_1^2 - \left(1 - q_1^2\right) = 4q_1^2 - 1$$

$$q_1 = +\frac{1}{2}\sqrt{DCM(1,1) - DCM(2,2) - DCM(3,3) + 1}$$

And then manipulate the off-diagonal terms to get the terms that multiply q_1:

$$DCM(1,2) + DCM(2,1) = 4q_1q_2$$

$$q_2 = \frac{1}{4q_1}\left(DCM(1,2) + DCM(2,1)\right)$$

$$q_3 = \frac{1}{4q_1}\left(DCM(3,1) + DCM(1,3)\right)$$

$$q_4 = \frac{1}{4q_1}\left(DCM(2,3) - DCM(3,2)\right)$$

There are 4 sets of these formulas, one solving for each of the 4 quaternion values first; I have only presented 2 of them. In practice you would use the set with the largest denominator so the answer is the most accurate.

YPR to quaternion

Tuesday, October 29, 2019 4:24 PM

Start with quaternion multiplication in the order of your sequence, left to right for quaternions (opposite of Euler Angle's right to left):

$$\bar{q}_{YPR} = \bar{q}_Y \otimes \bar{q}_P \otimes \bar{q}_R$$

Then write the quaternions out and start multiplying using the 4x4 quaternion rotation matrix from the Quaternion lecture:

$$= \begin{bmatrix} 0 \\ 0 \\ \sin\frac{Y}{2} \\ \cos\frac{Y}{2} \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \sin\frac{P}{2} \\ 0 \\ \cos\frac{P}{2} \end{bmatrix} \otimes \begin{bmatrix} \sin\frac{R}{2} \\ 0 \\ 0 \\ \cos\frac{R}{2} \end{bmatrix}$$

$$= \left(\begin{bmatrix} \cos\frac{P}{2} & 0 & -\sin\frac{P}{2} & 0 \\ 0 & \cos\frac{P}{2} & 0 & \sin\frac{P}{2} \\ \sin\frac{P}{2} & 0 & \cos\frac{P}{2} & 0 \\ 0 & -\sin\frac{P}{2} & 0 & \cos\frac{P}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \sin\frac{Y}{2} \\ \cos\frac{Y}{2} \end{bmatrix} \right) \otimes \begin{bmatrix} \sin\frac{R}{2} \\ 0 \\ 0 \\ \cos\frac{R}{2} \end{bmatrix}$$

$$= \begin{bmatrix} -\sin\frac{P}{2}\sin\frac{Y}{2} \\ \sin\frac{P}{2}\cos\frac{Y}{2} \\ \cos\frac{P}{2}\sin\frac{Y}{2} \\ \cos\frac{P}{2}\cos\frac{Y}{2} \end{bmatrix} \otimes \begin{bmatrix} \sin\frac{R}{2} \\ 0 \\ 0 \\ \cos\frac{R}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{R}{2} & 0 & 0 & \sin\frac{R}{2} \\ 0 & \cos\frac{R}{2} & \sin\frac{R}{2} & 0 \\ 0 & -\sin\frac{R}{2} & \cos\frac{R}{2} & 0 \\ -\sin\frac{R}{2} & 0 & 0 & \cos\frac{R}{2} \end{bmatrix} \begin{bmatrix} -\sin\frac{P}{2}\sin\frac{Y}{2} \\ \sin\frac{P}{2}\cos\frac{Y}{2} \\ \cos\frac{P}{2}\sin\frac{Y}{2} \\ \cos\frac{P}{2}\cos\frac{Y}{2} \end{bmatrix}$$

$$\bar{q}_{YPR} = \begin{bmatrix} \sin\frac{R}{2}\cos\frac{P}{2}\cos\frac{Y}{2} - \cos\frac{R}{2}\sin\frac{P}{2}\sin\frac{Y}{2} \\ \sin\frac{R}{2}\cos\frac{P}{2}\sin\frac{Y}{2} + \cos\frac{R}{2}\sin\frac{P}{2}\cos\frac{Y}{2} \\ \cos\frac{R}{2}\cos\frac{P}{2}\sin\frac{Y}{2} - \sin\frac{R}{2}\sin\frac{P}{2}\cos\frac{Y}{2} \\ \cos\frac{R}{2}\cos\frac{P}{2}\cos\frac{Y}{2} + \sin\frac{R}{2}\sin\frac{P}{2}\sin\frac{Y}{2} \end{bmatrix}$$

This is only valid for the YPR (3-2-1) sequence. It would be different for the other 11 Euler Angle sequences.

Quaternion between 2 vectors

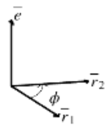
Thursday, September 15, 2022 8:13 AM

We start with the definition of the quaternion in terms of the Euler Parameters:

$$\bar{q} = \begin{bmatrix} \bar{e}\sin\frac{\phi}{2} \\ \cos\frac{\phi}{2} \end{bmatrix}$$

Then, $\bar{r}_1$ and $\bar{r}_2$ make a plane and the Euler Axis is perpendicular to this plane, thus:

$$\bar{e} = \frac{\bar{r}_1 \times \bar{r}_2}{\left\| \bar{r}_1 \times \bar{r}_2 \right\|}$$



The magnitude of a cross product is the product of their magnitudes times the sine of the angle between them thus:

$$\bar{e} = \frac{\bar{r}_1 \times \bar{r}_2}{\left\| \bar{r}_1 \times \bar{r}_2 \right\|} = \frac{\bar{r}_1 \times \bar{r}_2}{r_1 r_2 \sin\phi}$$

Next, we need the double-angle formulas:

$$\sin\phi = 2\sin\frac{\phi}{2}\cos\frac{\phi}{2}$$

$$\cos\phi = \cos^2\frac{\phi}{2} - \sin^2\frac{\phi}{2}$$

Replacing $\sin\phi$ we get:

$$\bar{e} = \frac{\bar{r}_1 \times \bar{r}_2}{\left\| \bar{r}_1 \times \bar{r}_2 \right\|} = \frac{\bar{r}_1 \times \bar{r}_2}{r_1 r_2 \sin\phi} = \frac{\bar{r}_1 \times \bar{r}_2}{r_1 r_2 \left(2\sin\frac{\phi}{2}\cos\frac{\phi}{2} \right)}$$

Now, substitute this into the quaternion definition and simplify:

$$\bar{q} = \begin{bmatrix} \left(\dfrac{\bar{r}_1 \times \bar{r}_2}{r_1 r_2 \left(2\sin\frac{\phi}{2}\cos\frac{\phi}{2} \right)} \right) \sin\frac{\phi}{2} \\ \cos\frac{\phi}{2} \end{bmatrix} = \frac{1}{2 r_1 r_2 \cos\frac{\phi}{2}} \begin{bmatrix} \bar{r}_1 \times \bar{r}_2 \\ 2 r_1 r_2 \cos^2\frac{\phi}{2} \end{bmatrix}$$

The scalar part now has a cosine squared term that we can replace using a trigonometric identity:

$$\cos\phi = \cos^2\frac{\phi}{2} - \sin^2\frac{\phi}{2}$$

$$1 = \cos^2\frac{\phi}{2} + \sin^2\frac{\phi}{2}$$

$$1 + \cos\phi = 2\cos^2\frac{\phi}{2}$$

Thus we get:

$$\bar{q} = \frac{1}{2 r_1 r_2 \cos\frac{\phi}{2}} \begin{bmatrix} \bar{r}_1 \times \bar{r}_2 \\ r_1 r_2 \left(1 + \cos\phi \right) \end{bmatrix}$$

Now, we use the fact that the magnitude of a dot product is the product of the magnitudes times the cosine of the angle between them:

$\bar{r}_1 \cdot \bar{r}_2 = r_1 r_2 \cos\phi$

Thus we have:

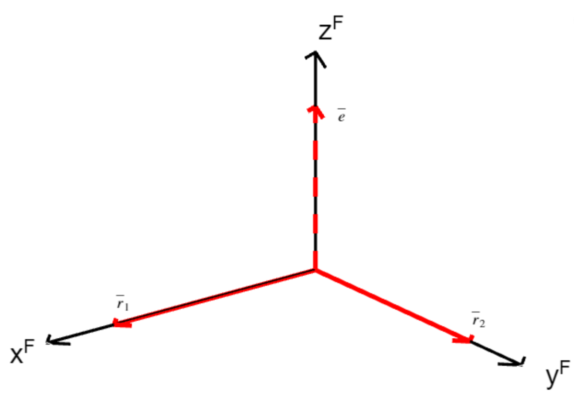
$$\bar{q} = \frac{1}{2 r_1 r_2 \cos\frac{\phi}{2}} \begin{bmatrix} \bar{r}_1 \times \bar{r}_2 \\ r_1 r_2 + \bar{r}_1 \cdot \bar{r}_2 \end{bmatrix}$$

Finally, since the quaternion is a unit vector, the term out front of the vector must simply be the magnitude of the vector, thus we can write this as simply:

$$\bar{q} = \frac{1}{\|\text{mag}\|} \begin{bmatrix} \bar{r}_1 \times \bar{r}_2 \\ r_1 r_2 + \bar{r}_1 \cdot \bar{r}_2 \end{bmatrix}$$

Example:

Let $\bar{r}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $\bar{r}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$



Then:

$$\bar{r}_1 x \bar{r}_2 = \begin{bmatrix} x & y & z \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

And

$r_1 r_2 + \bar{r}_1 \cdot \bar{r}_2 = (1)(1) + (0 + 0 + 0) = 1$

Then,

$$\bar{q} = \frac{1}{\text{mag}} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \text{mag} = \sqrt{2}$$

Thus:

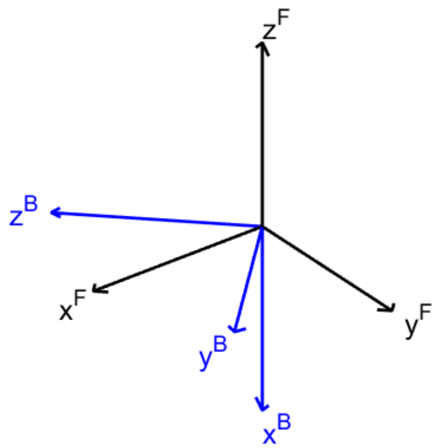
$$\bar{q} = \begin{bmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

This is the same as the quaternion for a rotation of $\pi/2$ about the z axis so it checks out

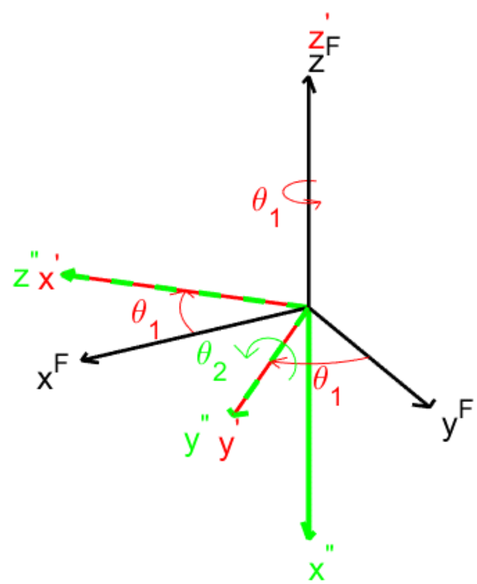
Parameterization Example Problem

Thursday, January 28, 2021 9:49 AM

Given:



Break up into rotations about base vectors using the 3-2-1 (given) sequence:



Euler Angles: Sequence 3-2-1

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_3\left(-\frac{\pi}{4}\right)\begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = R_2\left(\frac{\pi}{2}\right)\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = R_1(0)\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix}$$

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = R_1(0)R_2\left(\frac{\pi}{2}\right)R_3\left(-\frac{\pi}{4}\right)\begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix} = R_{321}(\bar{\theta})\begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$

Thus, the Euler Angles are, in order:

$$\bar{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{4} \\ \frac{\pi}{2} \\ 0 \end{bmatrix}$$

Plugging these angles into the base rotation matrices, we can determine the Direction Cosine Matrix (DCM):

$$DCM = R_1(0)R_2\left(\frac{\pi}{2}\right)R_3\left(-\frac{\pi}{4}\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM = \begin{bmatrix} 0 & 0 & -1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

The DCM can also be determined directly using the definition of the DCM in terms of the 9 cosines:

$$DCM = \begin{bmatrix} \cos\left(\theta_{x^F x^B}\right) & \cos\left(\theta_{y^F x^B}\right) & \cos\left(\theta_{z^F x^B}\right) \\ \cos\left(\theta_{x^F y^B}\right) & \cos\left(\theta_{y^F y^B}\right) & \cos\left(\theta_{z^F y^B}\right) \\ \cos\left(\theta_{x^F z^B}\right) & \cos\left(\theta_{y^F z^B}\right) & \cos\left(\theta_{z^F z^B}\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) & \cos(\pi) \\ \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{2}\right) \\ \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{3\pi}{4}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix}$$

$$DCM = \begin{bmatrix} 0 & 0 & -1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

This results in the same matrix as with the Euler Angles confirming both solutions.

Euler Parameters via the eigenvalue/eigenvector problem:

$$DCM\overline{e} = \overline{e}$$

$$\left(DCM - I_{3x3}\right)\overline{e} = \overline{0}$$

$$\begin{bmatrix} -1 & 0 & -1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} - 1 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

By definition, this 3x3 matrix is singular, meaning it has no inverse; its determinant is zero. This also means that the rows (and columns) of this matrix are dependent not independent. Thus, to solve, we assume a value for one of the components of the Euler Axis and then solve for the other two values. Finally, since I can multiply both sides of this equation by a constant and the equation will still hold, the magnitude of the eigenvector can be anything. In this problem, however, it is desired to be a unit vector. That is what a numerical eigenvalue/eigenvector algorithm will give you anyway. Most of the time, unless there is a clear mathematical advantage to do something different, we assume $e_1 = 1$, and solve for e_2 and e_3. Row 1 then becomes:

$$-1 - e_3 = 0, \quad \therefore e_3 = -1$$

Using Row 2 yields:

$$\frac{1}{\sqrt{2}} + \left(\frac{1}{\sqrt{2}} - 1\right)e_2 = 0$$

$$\left(\frac{1-\sqrt{2}}{\sqrt{2}}\right)e_2 = -\frac{1}{\sqrt{2}}$$

$$e_2 = \frac{1}{\sqrt{2}-1} = \frac{1}{\sqrt{2}-1}\frac{\sqrt{2}+1}{\sqrt{2}+1} = \sqrt{2}+1$$

Thus, the Euler Axis should point in the direction of the vector: $\begin{bmatrix} 1 \\ \sqrt{2}+1 \\ -1 \end{bmatrix}$

Before we make it a unit vector, I recommend checking the row that we didn't use yet in solving for the other e values, in this case Row 3, to make sure we did the algebra correctly. So Row 3:

$$\frac{1}{\sqrt{2}}(1)-\frac{1}{\sqrt{2}}\left(\sqrt{2}+1\right)-1(-1)=0?$$

$$\frac{1}{\sqrt{2}}-1-\frac{1}{\sqrt{2}}+1=0=0$$

Since Row 3 checks out, let's divide the vector by its' magnitude to make it a unit vector:

$$\bar{e}=\frac{1}{\sqrt{1^2+\left(\sqrt{2}+1\right)^2+(-1)^2}}\begin{bmatrix}1\\ \sqrt{2}+1\\ -1\end{bmatrix}=\frac{1}{\sqrt{5+2\sqrt{2}}}\begin{bmatrix}1\\ \sqrt{2}+1\\ -1\end{bmatrix}\approx\begin{bmatrix}0.3574\\ 0.8629\\ -0.3574\end{bmatrix}$$

Now we can use the trace of the DCM to determine the Euler Angle:

$$\text{trace}(DCM)=1+2\cos\phi$$

$$0+\frac{1}{\sqrt{2}}+0=1+2\cos\phi$$

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

$$\phi=\pm 1.7178\text{rad}$$

To determine which sign of the Euler Angle goes with the Euler Axis that we have, with the assumed positive e_1 value, we need to plug this ϕ into any off-diagonal value of the DCM to check the value. Here I will check DCM(2,3) which, in this case, needs to be 0:

$$DCM(2,3)=e_2e_3(1-\cos\phi)+e_1\sin\phi=0$$

$$DCM(2,3)=(0.8629)(-0.3574)\left(1-\frac{1-\sqrt{2}}{2\sqrt{2}}\right)+(0.3574)\sin\left(\pm 1.7178\right)=0?$$

$$DCM(2,3)=-0.3536\pm 0.3536=0?$$

Clearly then, the sign must be positive which corresponds to the positive Euler Angle, thus, a solution is:

$$\bar{e}=\begin{bmatrix}0.3574\\ 0.8629\\ -0.3574\end{bmatrix},\quad \phi=1.7178\text{rad}$$

The opposite Euler Angle can be used with the opposite Euler Axis, the same way rotating $\pi/2$ about the z axis is the same as rotating $-\pi/2$ about the negative z axis:

$$\bar{e}=\begin{bmatrix}-0.3574\\ -0.8629\\ 0.3574\end{bmatrix},\quad \phi=-1.7178\text{rad}$$

In any assignment or exam, you only have to provide 1 answer, not both.
Finally, we can use either Euler Parameters answer to determine the quaternion using the definition of the quaternion in terms of the Euler Parameters.

3D Dynamics

Sunday, January 12, 2020 1:57 PM

In Dynamics you learned that motion is always relative to a fixed frame and that the sum of force and moment equations are:

$$\sum \bar{F}^F = \frac{d^F}{dt}\left(m\bar{v}_{cm}^F\right)$$

$$\sum \bar{M}_{cm}^F = \frac{d^F}{dt}\left([I]^F \bar{\omega}^F\right) = \bar{T}^F$$

Where the mass Moment Of Inertia (MOI) matrix was defined in Chapter 8 of Statics in terms of "mass moments of inertia" and "product mass moments of inertia" respectively as:

$$[I] = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{bmatrix}$$

$$I_x, I_y, I_z = \int s^2 dm$$

$$I_x = \int y^2 + z^2 dm, I_y = \int x^2 + z^2 dm, I_z = \int x^2 + y^2 dm$$

$$I_{xy} = -\int xy\, dm, I_{xz} = -\int xz\, dm, I_{yz} = -\int yz\, dm$$

When we talk about the dynamics of a spacecraft, it is not convenient to find the forces and moments on the spacecraft in the Fixed frame when the forces and torques we use to control the position and attitude of the spacecraft are on the spacecraft, in the Body frame. The Body frame can be visualized as the frame used in a CAD drawing package like Catia; when you rotate the object to get a different view, the frame moves with the object. All of the x,y,z coordinates of each part that makes up the object is constant. This means that if the mass is constant, so is the mass moment of inertia.

In this class, for analytical analysis, we will choose the Body frame axes to be colinear with the Principle Axes of the spacecraft. These axes are such that the product mass MOI are exactly zero making the mass MOI diagonal. See the subpage, "Mass Moment of Inertia", for more details.

Let's write the sum of moments equation in the Body frame:

$$[I]^B = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

$$[I]^B \bar{\omega}_{B/F}^B = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} I_1\omega_1 \\ I_2\omega_2 \\ I_3\omega_3 \end{bmatrix}$$

$$\frac{d^B}{dt}\left([I]^B \bar{\omega}_{B/F}^B\right) = \begin{bmatrix} I_1\dot{\omega}_1 \\ I_2\dot{\omega}_2 \\ I_3\dot{\omega}_3 \end{bmatrix}$$

$$\bar{\omega}_{B/F}^B \times [I]^B \bar{\omega}_{B/F}^B = \begin{bmatrix} e_1 & e_2 & e_3 \\ \omega_1 & \omega_2 & \omega_3 \\ I_1\omega_1 & I_2\omega_2 & I_3\omega_3 \end{bmatrix} = \begin{bmatrix} (I_3 - I_2)\omega_2\omega_3 \\ (I_1 - I_3)\omega_1\omega_3 \\ (I_2 - I_1)\omega_1\omega_2 \end{bmatrix}$$

$$\begin{bmatrix} I_1\dot{\omega}_1 \\ I_2\dot{\omega}_2 \\ I_3\dot{\omega}_3 \end{bmatrix} + \begin{bmatrix} (I_3 - I_2)\omega_2\omega_3 \\ (I_1 - I_3)\omega_1\omega_3 \\ (I_2 - I_1)\omega_1\omega_2 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

These ODEs are known as Euler's Equations for Rigid-Body Motion
They are 1st order, nonlinear, coupled odes so there is no general analytical solution, however, there are solutions and approximations that exist for specific cases. Let's look at one of those cases now.

Solve for the approximate motion of a spacecraft spinning mostly about a single axis in a torque-free environment.
Let the initial angular velocity of the spacecraft be:

$$\bar{\omega}(0) = \begin{bmatrix} \tilde{\omega}_1 \\ \tilde{\omega}_2 \\ \tilde{\omega}_3 + N \end{bmatrix}$$

Where each $\tilde{\omega}$ is small such that any product of $\tilde{\omega}$'s is negligable, and N is the nominal/desired spin rate. So, the spacecraft is mostly spinning about the 3rd axis with small errors/perturbations in all 3 axes. We want to know for what conditions do the $\tilde{\omega}$'s stay small such that we have a stable spacecraft. First, we assume that ω_1, ω_2 stay small then the third ODE simplifys:

$$I_3\dot{\omega}_3 + (I_2 - I_1)\cancelto{\approx 0}{\omega_1\omega_2} = \cancelto{0}{T_3}$$

$$I_3\dot{\omega}_3 = 0$$

$$\omega_3(t) = \text{constant} = \tilde{\omega}_3 + N$$

Thus $\tilde{\omega}_3$ also stays small. Plug $\omega_3(t)$ into the first 2 ODEs and simplify. Since $\omega_1\tilde{\omega}_3$ and $\omega_2\tilde{\omega}_3$ are negligable and torque is zero:

$$I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2 N = 0$$

$$I_2\dot{\omega}_2 + (I_1 - I_3)\omega_1 N = 0$$

These are 1st order, coupled, linear ODEs that we can solve:
Let:

$$\bar{\omega}(t) = \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix} = \bar{c}e^{rt} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} e^{rt}$$

$$\dot{\bar{\omega}}(t) = r\bar{c}e^{rt}$$

Now, plug this into the ODEs and simplify by realizing the term, e^{rt}, can never be equal to zero:

$$I_1 r c_1 \cancel{e^{rt}} + (I_3 - I_2)c_2 \cancel{e^{rt}} N = 0$$

$$I_2 r c_2 \cancel{e^{rt}} + (I_1 - I_3)c_1 \cancel{e^{rt}} N = 0$$

These equations still have 3 unknowns: r, c_1, and c_2, but are linear in terms of c_1 and c_2. Write as a matrix equation in terms of c_1 and c_2:

$$\begin{bmatrix} I_1 r & (I_3 - I_2)N \\ (I_1 - I_3)N & I_2 r \end{bmatrix} \underbrace{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}_{\neq 0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since c_1 and c_2 cannot be zero, since that would not satisfy the boundary condition that ω_1 and ω_2 start small but not zero. That means that the matrix must be singular, it has a determinate that is zero. This leads to:

$$\begin{aligned} \det &= I_1 I_2 r^2 - (I_3 - I_2)(I_1 - I_3)N^2 = 0 \\ &= I_1 I_2 r^2 + (I_3 - I_2)(I_3 - I_1)N^2 = 0 \\ &= r^2 + (I_3 - I_2)(I_3 - I_1)\frac{N^2}{I_1 I_2} = 0 \\ &= r^2 + a = 0 \end{aligned}$$

If a < 0, then r has 1 positive and 1 negative real root. This leads to $\omega_1(t)$ and $\omega_2(t)$ to be exponentially increasing functions which does not satisfy our assumption that ω_1 and ω_2 stay small. This case is unstable. For a < 0 this means that the mass MOI for the axis we are spinning about, axis 3, must be the "intermediate" mass MOI, i.e.,
$I_1 > I_3 > I_2$ or $I_2 > I_3 > I_1$. Dancing T-handle in zero-g, HD

If a > 0 then r is pure imaginary. This leads to $\omega_1(t)$ and $\omega_2(t)$ to be pure sinusoids which means that since they start small, they will stay small. This case is stable. For a > 0 this means that I_3 must be the maximum or minimum mass MOI axis, i.e., $I_3 > I_{1,2}$ or $I_{1,2} > I_3$.

If a = 0, which is a theoretic possibility since this requires an exactly axi-symmetric spacecraft (this can't really happen because of the internal components of the spacecraft), then r is also zero. This means that $\omega_1(t)$ and $\omega_2(t)$ are constants; they must stay at their small initial conditions of $\tilde{\omega}_1$ and $\tilde{\omega}_2$. This is also stable.

Mass moment of inertia

Sunday, November 24, 2019 3:38 PM

$$[I]=\begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{bmatrix} \rightarrow \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

The goal is to find axes such that a spacecraft with angular momentum:

$$\bar{H}=[I]\bar{\omega}$$

Can have angular momentum:

$$\bar{H}=I_i\bar{\omega}, \quad i=1,2,3$$

Thus:

$$[I]\bar{\omega}=I_i\bar{\omega}$$

This is an eigenvalue/eigenvector problem where the eigenvalue is I_i and the eigenvector is $\bar{\omega}$. Let the eigenvectors be the basis

$$\langle \hat{e}_1, \hat{e}_2, \hat{e}_3 \rangle$$

Since the mass MOI matrix is size 3x3, there will be 3 solutions, if we add these equations we get:

$$[I]\hat{e}_1+[I]\hat{e}_2+[I]\hat{e}_3=I_1\hat{e}_1+I_2\hat{e}_2+I_3\hat{e}_3$$

$$[I]\begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix}=\begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix}\begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

The matrix with our basis vectors are orthogonal unit vectors, thus the inverse of this matrix is simply the transpose. So, to transform a full mass MOI matrix into a diagonal matrix along the principle axes, the basis vectors, we have:

$$\begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}=\begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix}^T [I]\begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix}$$

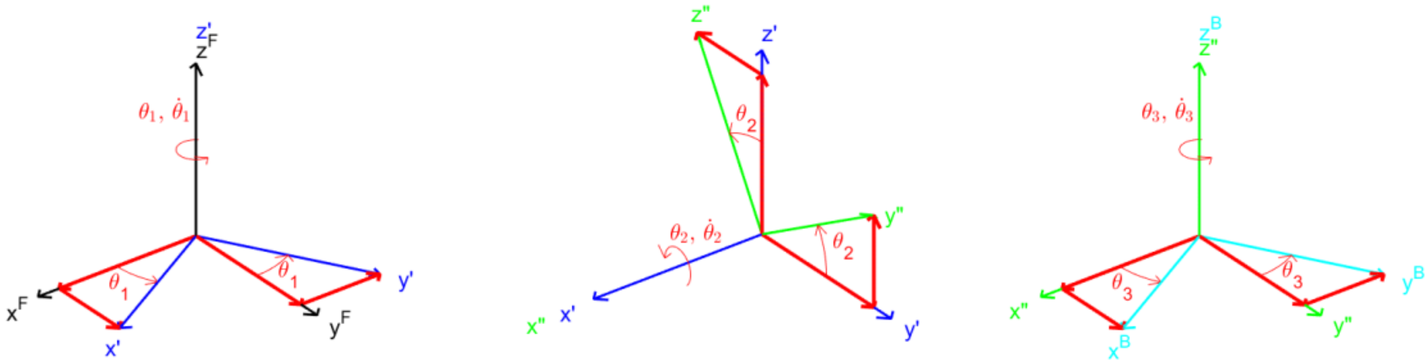
Euler Angle EOM

Friday, December 20, 2019 12:32 PM

Now that we know how to find Euler Angles, we also want to know how they will change in time as my spacecraft rotates. This is sequence dependent, so, there will be 12 answers, one for each of the 12 possible Euler Angle sequences. Here we will continue using the 3-1-3 example that you used in Space Mechanics. First, start with writing the rotation matrix out for your sequence:

$$R_{313} = R_3(\theta_3)R_1(\theta_1)R_3(\theta_3)$$

Recall, with the first rotation we rotate from the Fixed Frame to the Prime Frame, with the second rotation we rotate from the Prime Frame to the Double-Prime Frame, and with the third rotation we rotate from the Double-Prime Frame to the Body Frame.



The concept of this problem is that the sum of the time rates of change of the 3 Euler Angles have to add to get the total angular velocity of the spacecraft, i.e.:

$$\bar{\omega}^B = \sum_{i=1}^{3} \dot{\bar{\theta}}_i^B$$

For this sequence, the first Euler Angle, θ_1, rotates about the 3rd axis, z, so the time rate of change of the first Euler Angle, $\dot{\theta}_1$, will also be about the 3rd axis, z. Then we need to rotate this from the Fixed Frame to the Body Frame, the frame of the angular velocity of the spacecraft. Since the z axis is also the z' axis then rotating this into the body frame requires only 2 rotations:

$$R_3(\theta_3)R_1(\theta_1)\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}$$

We repeat this process for the other 2 Euler Angles, θ_2 and θ_3. θ_2 rotates about the 1st axis, x', so the time rate of change of the 2nd Euler Angle, $\dot{\theta}_2$, will also be about the 1st axis, x'. Then we rotate this from the Fixed Frame to the Body Frame. Since the x' axis is also the x" axis then we can write the time rate of change of the second Euler Angle, $\dot{\theta}_2$, in the Body Frame as:

$$R_3(\theta_3)\begin{bmatrix} \dot{\theta}_2 \\ 0 \\ 0 \end{bmatrix}$$

Finally, for the time rate of change of the 3rd Euler Angle, $\dot{\theta}_3$, This occurs about the 3rd axis, z" which is also z^B, thus it can be written simply as:

$$\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

Now we simply add these three time rates of change together to get the total angular velocity of the spacecraft:

$$\bar{\omega}^B = \sum_{i=1}^{3} \dot{\bar{\theta}}_i^B = R_3(\theta_3)R_1(\theta_1)\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + R_3(\theta_3)\begin{bmatrix} \dot{\theta}_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

Now, we must simply complete the necessary algebra:

$$\bar{\omega}^B = \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_2) & \sin(\theta_2) \\ 0 & -\sin(\theta_2) & \cos(\theta_2) \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{\theta}_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

$$\bar{\omega}^B = \begin{bmatrix} \cos(\theta_3) & \cos(\theta_2)\sin(\theta_3) & \sin(\theta_2)\sin(\theta_3) \\ -\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) & \sin(\theta_2)\cos(\theta_3) \\ 0 & -\sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} \cos(\theta_3) \\ -\sin(\theta_3) \\ 0 \end{bmatrix} \dot{\theta}_2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_3$$

$$\bar{\omega}^B = \begin{bmatrix} \sin(\theta_2)\sin(\theta_3) \\ \sin(\theta_2)\cos(\theta_3) \\ \cos(\theta_2) \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} \cos(\theta_3) \\ -\sin(\theta_3) \\ 0 \end{bmatrix} \dot{\theta}_2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_3$$

$$\bar{\omega}^B = \begin{bmatrix} \sin(\theta_2)\sin(\theta_3) & \cos(\theta_3) & 0 \\ \sin(\theta_2)\cos(\theta_3) & -\sin(\theta_3) & 0 \\ \cos(\theta_2) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \left[S(\bar{\theta}) \right] \dot{\bar{\theta}}$$

Then, the Euler Angle EOM is found by solving explicitly for $\dot{\bar{\theta}}$ by multiplying both sides by the inverse of the matrix *S*:

$$\dot{\bar{\theta}} = \left[S(\bar{\theta}) \right]^{-1} \bar{\omega}^B$$

The inverse of a matrix is 1/determinate times the adjugate matrix:
Starting with the determinate:

$$\det\left(\left[S(\bar{\theta}) \right]\right) = \sin(\theta_2)\sin(\theta_3)\left(-\sin(\theta_3)\right) - \cos(\theta_3)\left(\sin(\theta_2)\cos(\theta_3)\right)$$

$$\det\left(\left[S(\bar{\theta}) \right]\right) = -\sin(\theta_2)\left(\sin^2(\theta_3) + \cos^2(\theta_3)\right) = -\sin(\theta_2)$$

It turns out, for all 12 possible sequences, the determinate is either $\pm \sin \theta_2$ or $\pm \cos \theta_2$
If the determinate is zero, that means the matrix is singular and the inverse does not exist. For this case that is where:

$$-\sin(\theta_2) = 0$$

$$\theta_2 = n\pi, \quad n = \mathbb{Z}$$

where $\mathbb{Z}$ represents all integers: $\{\cdots -2, -1, 0, 1, 2, \cdots\}$
If your sequence has a determinate that is a cosine then the answer would be:

$$\cos(\theta_2) = 0$$

$$\theta_2 = \frac{\pi}{2}(2n+1)\pi, \quad n = \mathbb{Z}$$

Calculating the rest of the inverse using the minor, cofactor, adjugate method (not covered here):

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{-\sin(\theta_2)} \begin{bmatrix} -\sin(\theta_3) & -\cos(\theta_3) & 0 \\ -\sin(\theta_2)\cos(\theta_3) & \sin(\theta_2)\sin(\theta_3) & 0 \\ \cos(\theta_2)\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) & -\sin(\theta_2) \end{bmatrix} \bar{\omega}^B$$

$$\dot{\bar{\theta}} = \begin{bmatrix} \csc(\theta_2)\sin(\theta_3) & \csc(\theta_2)\cos(\theta_3) & 0 \\ \cos(\theta_3) & -\sin(\theta_3) & 0 \\ -\cot(\theta_2)\sin(\theta_3) & -\cot(\theta_2)\cos(\theta_3) & 1 \end{bmatrix} \bar{\omega}^B$$

Quaternion EOM

Sunday, January 12, 2020 2:22 PM

Let the attitude of the spacecraft be $\bar{q}$, the change in attitude during a Δt be $\bar{q}$', and the new attitude at time $t + \Delta t$ be $\bar{q}$". Thus we can write the following using quaternion multiplication:

$\bar{q}" = \bar{q} \otimes \bar{q}'$

Now, using the definition of the quaternion in terms of the Euler Parameters we can rewrite the quaternion,

$$\bar{q}' = \begin{bmatrix} \bar{e}\sin\left(\frac{\Delta\phi}{2}\right) \\ \cos\left(\frac{\Delta\phi}{2}\right) \end{bmatrix}$$

If $\Delta\phi$ small during the Δt then the angular velocity of the spacecraft can be written as:

$$\omega = \frac{\Delta\phi}{\Delta t}, \text{and} \quad \Delta\phi = \omega\Delta t$$

Also, by definition, the Euler Axis, $\bar{e}$, is the axis that the spacecraft is rotating about thus this is also the axis the angular velocity of the spacecraft points:

$$\bar{e} = \frac{\bar{\omega}}{\omega}$$

Substituting these 2 expressions into $\bar{q}$':

$$\bar{q}' = \begin{bmatrix} \frac{\bar{\omega}}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) \\ \cos\left(\frac{\omega\Delta t}{2}\right) \end{bmatrix}$$

To turn this into a differential equation we will have to take the limit as $\Delta t \to 0$ thus we can use small angle approximation to yield:

$$\bar{q}' = \begin{bmatrix} \frac{\bar{\omega}}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) \\ \cos\left(\frac{\omega\Delta t}{2}\right) \end{bmatrix} \approx \begin{bmatrix} \frac{\Delta t}{2}\bar{\omega} \\ 1 \end{bmatrix}$$

Then, this gets plugged into the quaternion multiplication equation:

$$\bar{q}" = \begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

Which yields:

$$\bar{q}"=\left(I_{4x4}+\frac{\Delta t}{2}\begin{bmatrix}0 & \omega_3 & -\omega_2 & \omega_1\\ -\omega_3 & 0 & \omega_1 & \omega_2\\ \omega_2 & -\omega_1 & 0 & \omega_3\\ -\omega_1 & -\omega_2 & -\omega_3 & 0\end{bmatrix}\right)\bar{q}$$

Now, by definition of a time rate of change:

$$\dot{\bar{q}}=\lim_{\Delta t\to 0}\frac{\bar{q}"-\bar{q}}{\Delta t}=\frac{1}{2}\begin{bmatrix}0 & \omega_3 & -\omega_2 & \omega_1\\ -\omega_3 & 0 & \omega_1 & \omega_2\\ \omega_2 & -\omega_1 & 0 & \omega_3\\ -\omega_1 & -\omega_2 & -\omega_3 & 0\end{bmatrix}\bar{q}=\frac{1}{2}[\Omega]\bar{q}$$

The quaternion must maintain unit magnitude, however, traditional integrators, e.g. Euler, Runge-Kutta, etc., use summation to approximate an integral. Summation does not exist in quaternion space, on the surface of a 4D unit hypersphere (meaning the quaternion won't stay unit magnitude after you add something to it). Require a rotational operator to rotate quaternion from t to $t + \Delta t$ if you want/need to maintain unit magnitude.

Integrate the Quaternion

If we integrate the quaternion, it will not maintain unit magnitude so you must normalize the quaternion after every integration step. To maintain the accuracy of the result, you must keep the step size "small" as outlined in the Quaternion Integration PDF on Canvas. In this class, as recommended in the paper, we will use 0.01 sec. To ensure this step size, we will use a fixed-step-size Runge-Kutta algorithm, ode4.

Propagate the Quaternion

If we propagate the quaternion, we use the original quaternion multiplication with no small angle approximation:

$\bar{q}" = \bar{q} \otimes \bar{q}'$

Where:

$$\bar{q}'=\begin{bmatrix}\frac{\bar{\omega}}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right)\\ \cos\left(\frac{\omega\Delta t}{2}\right)\end{bmatrix}$$

Written out completely, this yields the following:

$$\bar{q}\,'' = \bar{q}(t+\Delta t) = \begin{bmatrix} \cos\left(\frac{\omega\Delta t}{2}\right) & \frac{\omega_3}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & -\frac{\omega_2}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & \frac{\omega_1}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) \\ -\frac{\omega_3}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & \cos\left(\frac{\omega\Delta t}{2}\right) & \frac{\omega_1}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & \frac{\omega_2}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) \\ \frac{\omega_2}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & -\frac{\omega_1}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & \cos\left(\frac{\omega\Delta t}{2}\right) & \frac{\omega_3}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) \\ -\frac{\omega_1}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & -\frac{\omega_2}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & -\frac{\omega_3}{\omega}\sin\left(\frac{\omega\Delta t}{2}\right) & \cos\left(\frac{\omega\Delta t}{2}\right) \end{bmatrix} \bar{q}(t)$$

$$\bar{q}(t+\Delta t) = \Phi\left(t, t+\Delta t\right)\bar{q}(t)$$

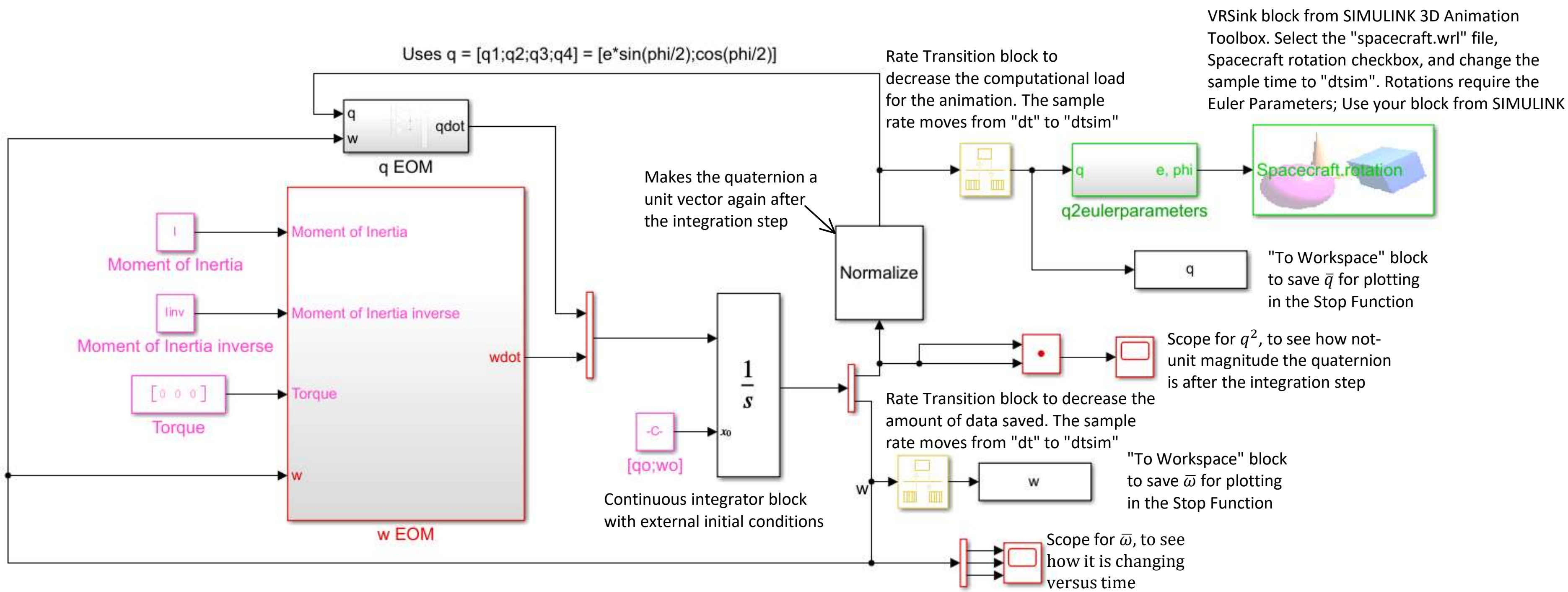
Where Φ is known as a "State-Transition Matrix"

This is similar to using the F and G functions in Space Mechanics to propagate, not integrate, an orbit:

$$\begin{bmatrix} \bar{r}(t) \\ \bar{v}(t) \end{bmatrix} \begin{bmatrix} F & G \\ \dot{F} & \dot{G} \end{bmatrix} \begin{bmatrix} \bar{r}(0) \\ \bar{v}(0) \end{bmatrix}$$

SIMULINK 2

Tuesday, February 6, 2024 3:36 PM



"w EOM": Contains the sum of moments equation:

$$\sum \bar{M}_{cm}^{B} = \frac{d^{B}}{dt}\left([I]^{B}\bar{\omega}_{B/F}^{B}\right) + \bar{\omega}_{B/F}^{B} \times [I]^{B}\bar{\omega}_{B/F}^{B} = \bar{T}^{B}$$

$$[I]^{B}\dot{\bar{\omega}}_{B/F}^{B} = \bar{T}^{B} - \bar{\omega}_{B/F}^{B} \times [I]^{B}\bar{\omega}_{B/F}^{B}$$

$$\dot{\bar{\omega}}_{B/F}^{B} = \left([I]^{B}\right)^{-1}\left(\bar{T}^{B} - \bar{\omega}_{B/F}^{B} \times [I]^{B}\bar{\omega}_{B/F}^{B}\right)$$

"q EOM": Contains the quaternion EOM:

$$\dot{\bar{q}} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}\bar{q} = \frac{1}{2}[\Omega]\bar{q}$$

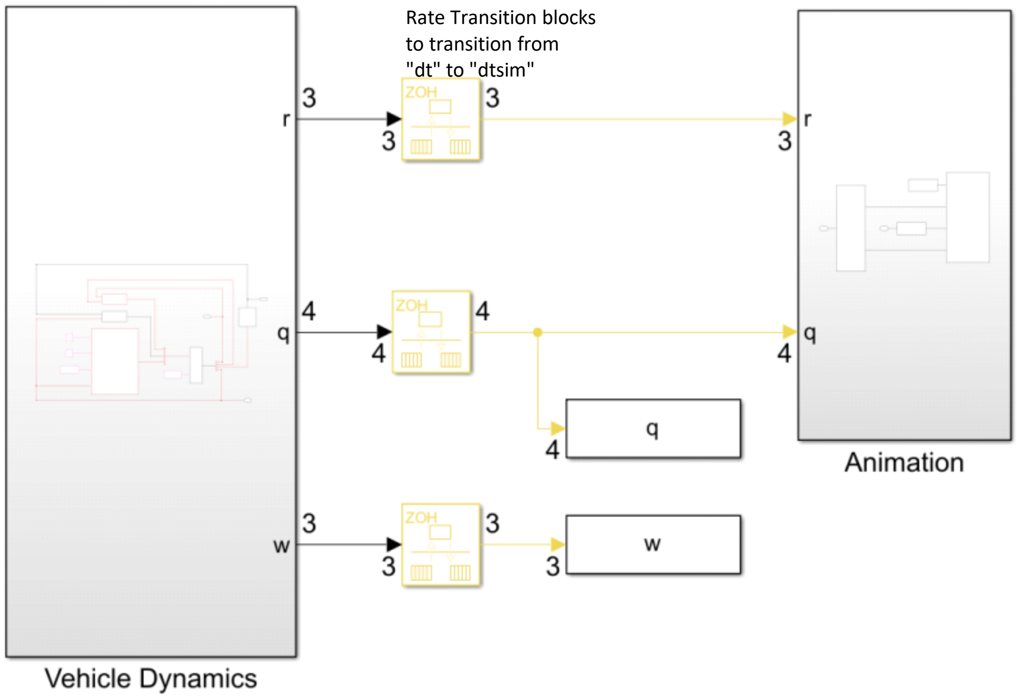
Define all constants in the InitFcn in the Model Properties menu

In Model Settings: change the solver to "Fixed-Step", "ode4 (Runge-Kutta)", and change the fundamental sample time to "dt".

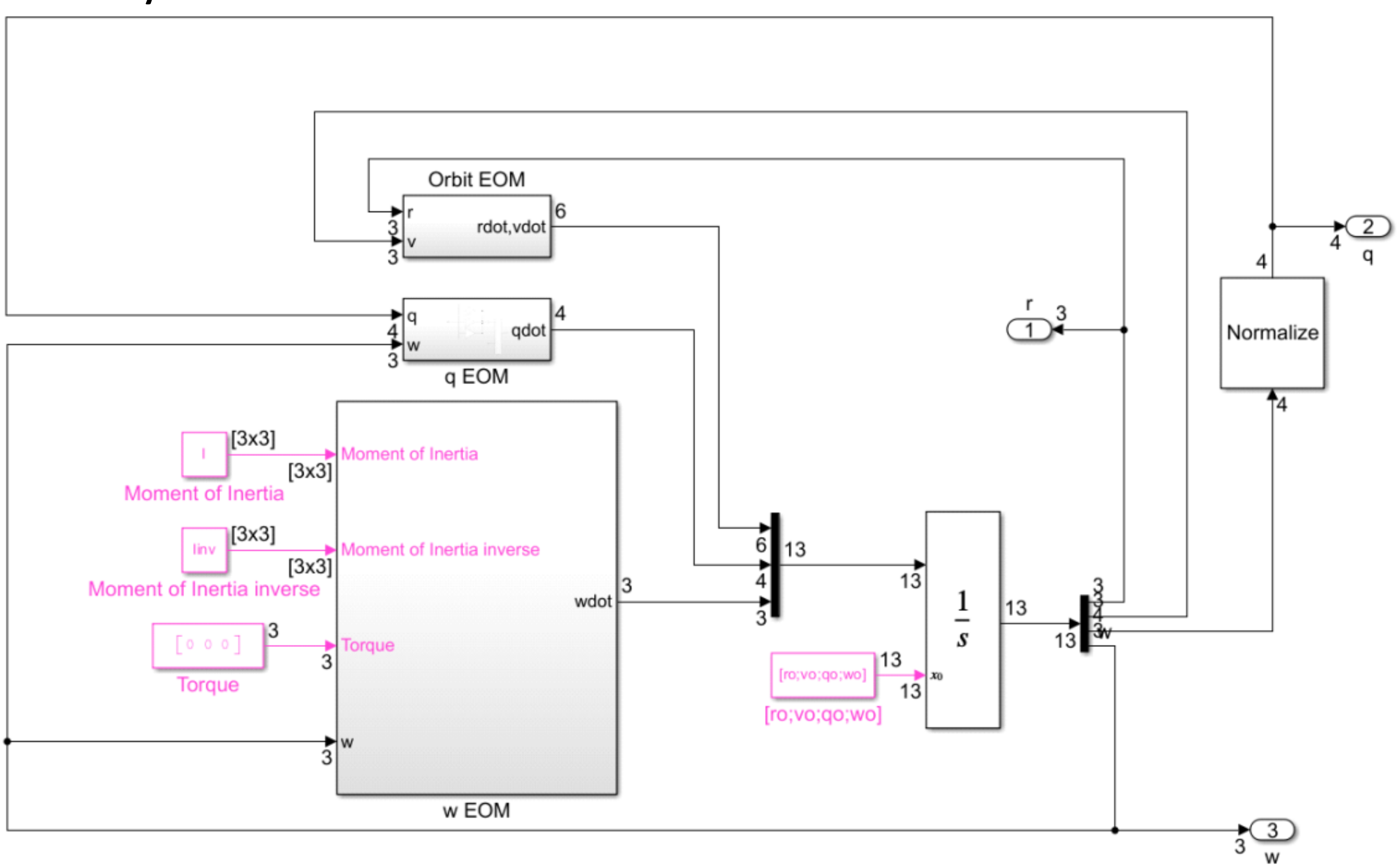
SIMULINK 3

Thursday, September 17, 2020 10:06 AM

Overview



Vehicle Dynamics

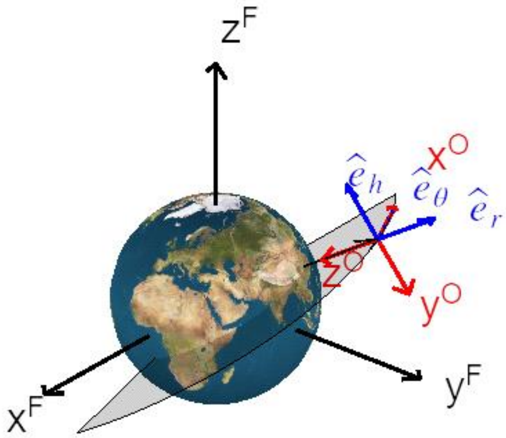


Include the translational motion of the spacecraft in a low-Earth orbit:

SIMULINK 4 Overview

Thursday, March 12, 2020 5:11 PM

In Simulink 4 we will add a 3rd frame, the Orbit Frame, which will lie between the Fixed Frame and the Body Frame. This frame is the same as that used for aircraft, specifically, the x-axis points out the nose, the y-axis points out the right wing, and the z-axis points down. As it relates to the frame used in Space Mechanics, the x-axis points in the transverse direction, the y-axis points in the negative angular momentum direction, and the z-axis points in the negative radial direction, i.e., down to the center of the Earth.



Our Body Frame, attached to our spacecraft, is now relative to this Orbit Frame, not the Fixed Frame. Thus, our dynamics will have the inclusion of the moving orbit frame:

$$\sum \bar{M}^B_{cm} = \frac{d^B}{dt}\bar{H}^B + \bar{\omega}^B_{B/F} \times \bar{H}^B = \bar{T}^B$$

$$\bar{H}^B = [I]^B \bar{\omega}^B_{B/F} + \bar{H}_{bias}$$

$$\bar{\omega}^B_{B/F} = \bar{\omega}^B_{O/F} + \bar{\omega}^B_{B/O}$$

$$\dot{\bar{\omega}}^B_{B/F} = \dot{\bar{\omega}}^B_{O/F} + \dot{\bar{\omega}}^B_{B/O}$$

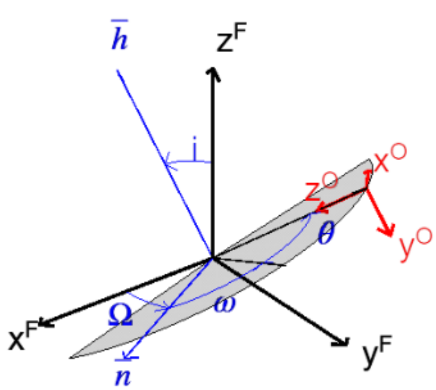
We will also include the possibility of angular momentum bias, that may be Control Moment Gyros (CMGs), Reaction Wheels, or any other spinning object internal to your spacecraft. In our model, we will assume this is a constant bias therefore it will not have any derivative term. We will now integrate $\dot{\bar{\omega}}^B_{B/O}$ to yield $\bar{\omega}^B_{B/O}$ by algebraically manipulating the sum of moments equation:

$$[I]^B \dot{\bar{\omega}}^B_{B/F} = \bar{T}^B - \bar{\omega}^B_{B/F} \times \left([I]^B \bar{\omega}^B_{B/F} + \bar{H}_{bias}\right)$$

$$\dot{\bar{\omega}}^B_{B/O} = \left([I]^B\right)^{-1}\left(\bar{T}^B - \bar{\omega}^B_{B/F} \times \left([I]^B \bar{\omega}^B_{B/F} + \bar{H}_{bias}\right)\right) - \dot{\bar{\omega}}^B_{O/F}$$

In order to do this, we must determine both $\dot{\bar{\omega}}^B_{O/F}$ and $\bar{\omega}^B_{O/F}$. Since these describe the motion of the spacecraft in the orbit about the Earth, it makes sense they are functions of the orbit, specifically $\bar{r}^F$ and $\bar{v}^F$.

Angular Velocity of Orbit Frame Relative to Fixed Frame



Starting with the angular velocity of the Orbit Frame relative to the Fixed Frame, assuming we are in a constant orbit, the Orbit Frame rotates about the angular momentum vector, negative y-orbit, by angle θ, thus angular velocity $\dot{\theta}$.

$$\bar{\omega}_{O/F}^{O} = \begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix}$$

Since this term is in the angular momentum direction, it makes sense to determine the value of $\dot{\theta}$ using the angular momentum vector. In the Orbit Frame we have:

$$\bar{r} = r\hat{e}_r = -r\,z^O$$

$$\bar{v} = \dot{r}\hat{e}_r + r\dot{\theta}\hat{e}_\theta = r\dot{\theta}\,x^O - \dot{r}\,z^O$$

$$\bar{h} = \bar{r} \times \bar{v} = \begin{bmatrix} x^O & y^O & z^O \\ 0 & 0 & -r \\ r\dot{\theta} & 0 & -\dot{r} \end{bmatrix} = \begin{bmatrix} 0 \\ -r^2\dot{\theta} \\ 0 \end{bmatrix}$$

Then, we can determine $\dot{\theta}$ by using our Fixed Frame vectors since the magnitude of the angular momentum is the same in all frames:

$$\dot{\theta} = \frac{h}{r^2} = \frac{\left\| \bar{r}^F \times \bar{v}^F \right\|}{\left(r^F\right)^2}$$

Finally, this term needs to be rotated from the Orbit Frame to the Body Frame. Since we will be using $\bar{\omega}_{B/O}^{B}$ to create the quaternion EOM, the quaternion that we will have will also be Body Frame relative to Orbit Frame, $\bar{q}_{B/O}$. We can use this in a quaternion to Direction Cosince Matrix conversion block, from SIMULINK 1, to make the rotation matrix we need to rotate the Orbit Frame vector into the Body Frame:

$$\bar{\omega}_{O/F}^{B} = DCM(\bar{q}_{B/O}) \begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix}$$

Angular Acceleration of Orbit Frame Relative to Fixed Frame

To determine the angular acceleration of the Orbit Frame relative to the Fixed Frame we must take the derivative of the angular velocity. Using the product rule:

$$\dot{\bar{\omega}}_{O/F}^{B} = \frac{d}{dt}\left(DCM(\bar{q}_{B/O}) \begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix} \right) = DCM(\bar{q}_{B/O}) \begin{bmatrix} 0 \\ -\ddot{\theta} \\ 0 \end{bmatrix} + \frac{d}{dt}\left(DCM(\bar{q}_{B/O})\right) \begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix}$$

We can determine the $\ddot{\theta}$ term by using the sum of forces in the transverse, or x-orbit, direction. Since we assume no forces, only torques, in this class we have:

$$\sum F_\theta = \sum F_{x^O} = m\left(r\ddot{\theta} + 2\dot{r}\dot{\theta}\right) = \cancel{F}_{\theta}^{0} = 0$$

$$\ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r}$$

To determine the radial velocity, $\dot{r}$, we need the velocity component along the radial direction which we can get using a dot product. This can be clearly seen using Orbit Frame vectors but it is true in any frame:

$$\bar{r}^O \cdot \bar{v}^O = \left(-r\,z^O\right) \cdot \left(r\dot{\theta}\,x^O - \dot{r}\,z^O\right) = r\dot{r}$$

$$\bar{r}^F \cdot \bar{v}^F = r\dot{r}$$

Thus, the $\ddot{\theta}$ term is:

$$\ddot{\theta} = -\frac{2\left(\bar{r}^F \cdot \bar{v}^F\right)\dot{\theta}}{r^2}$$

using the $\dot{\theta}$ term from the previous section.

Next, we need to determine the derivative of the Direction Cosine Matrix. The full derivation of this term is in the sub-page document with the final result used below::

$$\frac{d}{dt}\left(DCM(\overline{q}_{B/O})\right)\begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_{B/O_3} & -\omega_{B/O_2} \\ -\omega_{B/O_3} & 0 & \omega_{B/O_1} \\ \omega_{B/O_2} & -\omega_{B/O_1} & 0 \end{bmatrix} DCM(\overline{q}_{B/O})\begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix}$$

where

$$\overline{\omega}_{O/F}^{B} = DCM(\overline{q}_{B/O})\begin{bmatrix} 0 \\ -\dot{\theta} \\ 0 \end{bmatrix}$$

derived above. Now putting this term with the first term we have:

$$\dot{\overline{\omega}}_{O/F}^{B} = DCM(\overline{q}_{B/O})\begin{bmatrix} 0 \\ -\ddot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & \omega_{B/O_3} & -\omega_{B/O_2} \\ -\omega_{B/O_3} & 0 & \omega_{B/O_1} \\ \omega_{B/O_2} & -\omega_{B/O_1} & 0 \end{bmatrix} \overline{\omega}_{O/F}^{B}$$

$$\sum \bar{F}^{F} = \frac{d^{F}}{dt}\left(m\bar{v}^{F}\right)$$

$$\ddot{\bar{r}} = -\frac{\mu}{r^{3}}\bar{r} + \frac{\bar{F}}{m}$$

$$\dot{\bar{r}} = \bar{v}$$

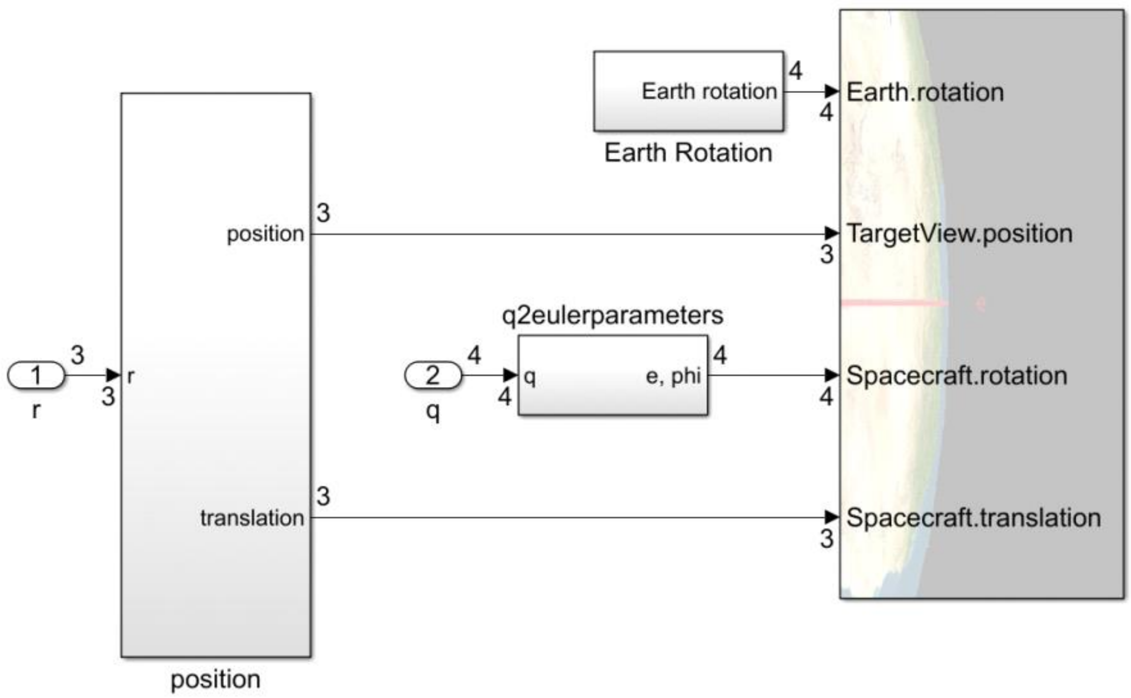
$$\dot{\bar{v}} = - -\frac{\mu}{r^{3}}\bar{r} + \frac{\cancel{\bar{F}}^{0}}{m}$$

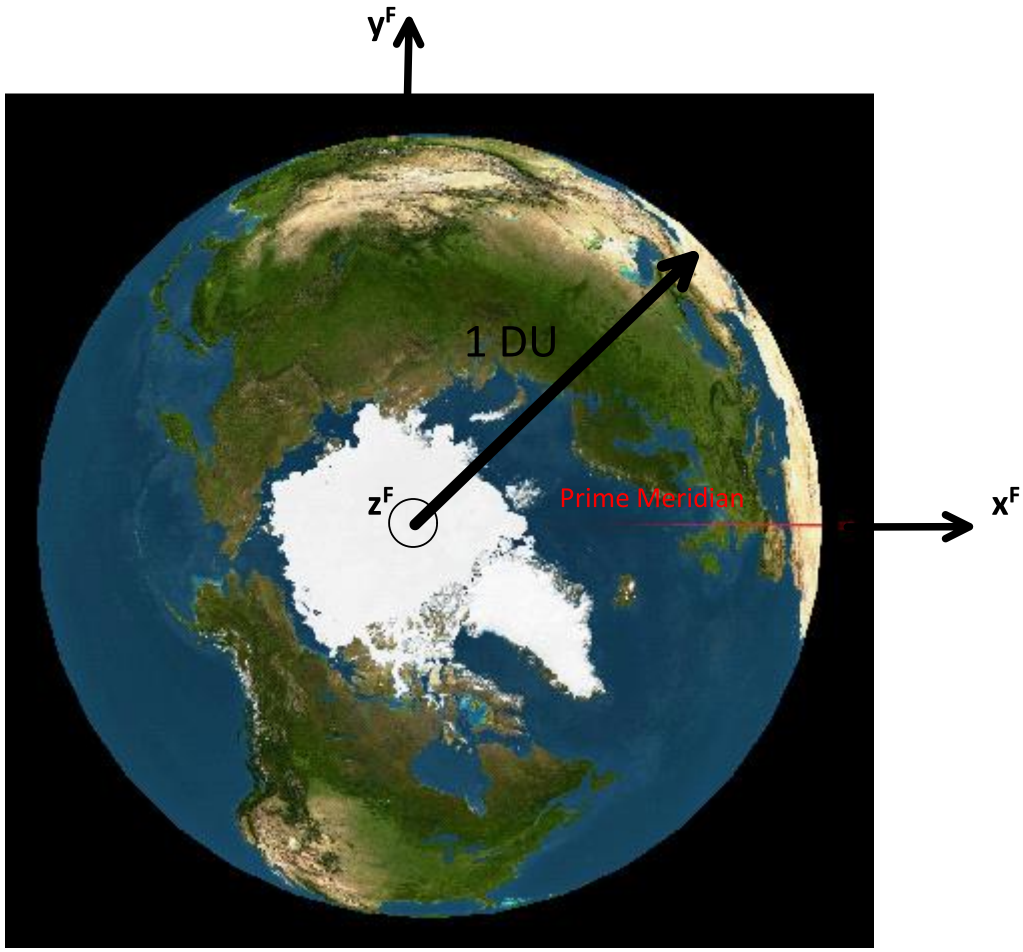
In this class we will assume no translational forces, i.e., no thrusting to change our orbit.
Here are some initial conditions you might try:

$$\mu = 398600\,\mathrm{km^3/s^2}$$

$$\bar{r}(0) = \begin{bmatrix} 6678 \\ 0 \\ 0 \end{bmatrix}\mathrm{km}, \quad \bar{v}(0) = \sqrt{\frac{\mu}{r}}\begin{bmatrix} 0 \\ 0.8 \\ 0.7 \end{bmatrix}\mathrm{km/s}$$

Animation

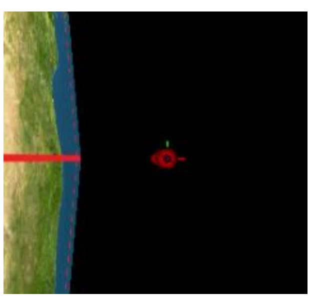




1 Distance Units (DU) = 6378 km, also known as Canonical Units

$$\text{"spacecraft.translation"} = \frac{\bar{r}}{6378}\,\text{DU}$$

$$\text{"TargetView.position"} = \frac{\bar{r}}{6378} + \begin{bmatrix} 0 \\ 0 \\ 1.25 \end{bmatrix}\text{DU}$$



This gives a "bird's-eye view" of the spacecraft as it moves about the Earth in its' orbit.

"Earth.rotation", being a rotation, requires the Euler Parameters. The Earth rotates about the z axis and rotates with a velocity equivalent to 1 rotation every 24 hours:

$$\omega_{\oplus} = \frac{2\pi}{24 \cdot 3600}\,\text{rad/s}$$

The angle of rotation then is simply this angular velocity times the current simulation time. This time is obtained from a "digital clock" block with the correct sample rate of "dtsim"

$$\bar{q} = \begin{bmatrix} \hat{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \bar{e}\sin\left(\dfrac{\phi}{2}\right) \\ \cos\left(\dfrac{\phi}{2}\right) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0.3574 \\ 0.8629 \\ -0.3574 \end{bmatrix} \sin\left(\dfrac{1.7178}{2}\right) \\ \cos\left(\dfrac{1.7178}{2}\right) \end{bmatrix} = \begin{bmatrix} 0.2076 \\ 0.6533 \\ -0.2076 \\ 0.6533 \end{bmatrix}$$

The Euler Parameters and the quaternion solutions can be verified using the formulas from the class equation sheet:

$$\mathrm{trace}(DCM) = 1 + 2\cos\phi$$

$$\bar{e} = \begin{bmatrix} DCM(2,3) - DCM(3,2) \\ DCM(3,1) - DCM(1,3) \\ DCM(1,2) - DCM(2,1) \end{bmatrix}$$

$$q_4 = +\frac{1}{2}\sqrt{\mathrm{trace}(DCM(\bar{q})) + 1}$$

$$\hat{q} = \frac{1}{4q_4}\begin{bmatrix} DCM(2,3) - DCM(3,2) \\ DCM(3,1) - DCM(1,3) \\ DCM(1,2) - DCM(2,1) \end{bmatrix}$$

DCM EOM

Saturday, February 10, 2024 5:26 PM

Similar to the quaternion EOM we can start with a DCM at time t, DCM(t), a DCM during a time Δt, DCM(Δt), and a DCM at time t + Δt, DCM(t+Δt), such that:

$$DCM(t+\Delta t) = DCM(\Delta t)DCM(t)$$

Then, let the DCM(Δt) be in terms of the Euler Parameters, where the angle swept out during the Δt is Δϕ:

$$DCM(\Delta t) = \begin{bmatrix} \cos\Delta\phi + e_1^2(1-\cos\Delta\phi) & e_1e_2(1-\cos\Delta\phi)+e_3\sin\Delta\phi & e_1e_3(1-\cos\Delta\phi)-e_2\sin\Delta\phi \\ e_1e_2(1-\cos\Delta\phi)-e_3\sin\Delta\phi & \cos\Delta\phi + e_2^2(1-\cos\Delta\phi) & e_2e_3(1-\cos\Delta\phi)+e_1\sin\Delta\phi \\ e_1e_3(1-\cos\Delta\phi)+e_2\sin\Delta\phi & e_2e_3(1-\cos\Delta\phi)-e_1\sin\Delta\phi & \cos\Delta\phi + e_3^2(1-\cos\Delta\phi) \end{bmatrix}$$

The Δϕ will be small as the Δt approaches 0, as will be required to make this an EOM, so next we use small angle approximation such that all $\cos\Delta\phi$ terms approach 1 and all $\sin\Delta\phi$ terms approach $\Delta\phi$:

$$DCM(\Delta t) = \begin{bmatrix} 1 & e_3\Delta\phi & -e_2\Delta\phi \\ -e_3\Delta\phi & 1 & e_1\Delta\phi \\ e_2\Delta\phi & -e_1\Delta\phi & 1 \end{bmatrix}$$

Now replace $\Delta\phi$ with $\omega\Delta t$ and since

$$\bar{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \omega \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

we get the following and can plug this back into our first equation:

$$DCM(\Delta t) = I_{3x3} + \Delta t \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}$$

$$DCM(t+\Delta t) = DCM(\Delta t)DCM(t) = \left(I_{3x3} + \Delta t \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \right) DCM(t)$$

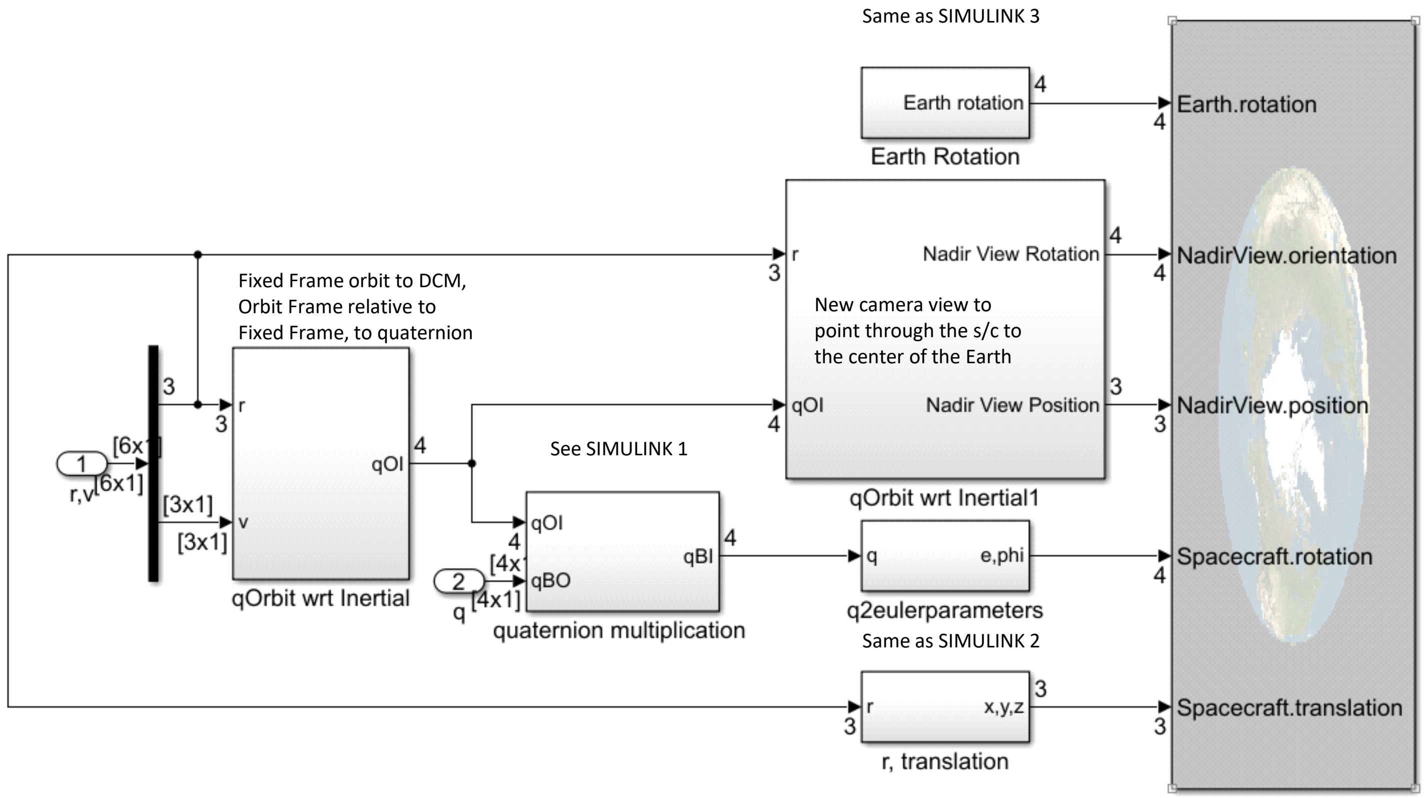
Finally, to make it a differential equation we use the following limit and get our final answer:

$$\frac{d}{dt}DCM = D\dot{C}M = \lim_{\Delta t \to 0} \frac{DCM(t+\Delta t) - DCM(t)}{\Delta t}$$

$$\frac{d}{dt}DCM = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} DCM(t)$$

SIMULINK 4 Animation

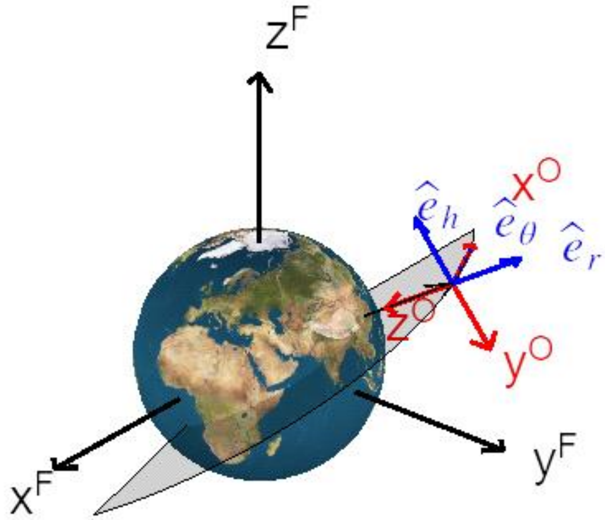
Thursday, August 13, 2020 10:59 AM



By definition of the Direction Cosine Matrix we have:

$$\begin{bmatrix} \hat{x}^O \\ \hat{y}^O \\ \hat{z}^O \end{bmatrix} = \begin{bmatrix} \cos\left(\theta_{x^F x^O}\right) & \cos\left(\theta_{y^F x^O}\right) & \cos\left(\theta_{z^F x^O}\right) \\ \cos\left(\theta_{x^F y^O}\right) & \cos\left(\theta_{y^F y^O}\right) & \cos\left(\theta_{z^F y^O}\right) \\ \cos\left(\theta_{x^F z^O}\right) & \cos\left(\theta_{y^F z^O}\right) & \cos\left(\theta_{z^F z^O}\right) \end{bmatrix} \begin{bmatrix} \hat{x}^F \\ \hat{y}^F \\ \hat{z}^F \end{bmatrix}$$

Physically, this means the first row of the DCM is simply the x-Orbit Frame axis as components in the x, y, and z Fixed Frame axes. It follows that the second row is the y-Orbit Frame axis and the third row is the z-Orbit Frame axis also as components in the x, y, and z Fixed Frame axes. Since we have the orbit in terms of fixed frame vectors, $\bar{r}^F$ and $\bar{v}^F$, then we simply have to define the Orbit Frame directions using these Fixed Frame vectors.



The easiest is the z-Orbit axis which is simply in the negative radial direction. Hence,

$$\hat{z}^O = -\frac{\overline{r}^F}{r^F}$$

Then the y-Orbit axis is negative of the specific angular momentum vector:

$$\hat{y}^O = -\frac{\overline{h}}{h} = \frac{\overline{v}^F \times \overline{r}^F}{\left\|\overline{v}^F \times \overline{r}^F\right\|}$$

Finally, with y-Orbit and z-Orbit defined, a cross product can be used to determine the x-Orbit, orthogonal, vector:

$$\hat{x}^O = \hat{y}^O \times \hat{z}^O$$

These 3 vectors can then be concatenated to make the DCM Orbit Frame relative to Fixed Frame, or from the Fixed Frame to the Orbit Frame.

Although there are 4 different formulas to determine the quaternion from the DCM, 2 of which are derived in the quaternion page, we can use this one:

$$q_4 = +\frac{1}{2}\sqrt{\text{trace}(DCM(\overline{q}))+1}$$

$$q_1 = \frac{1}{4q_4}\left(DCM(2,3) - DCM(3,2)\right)$$

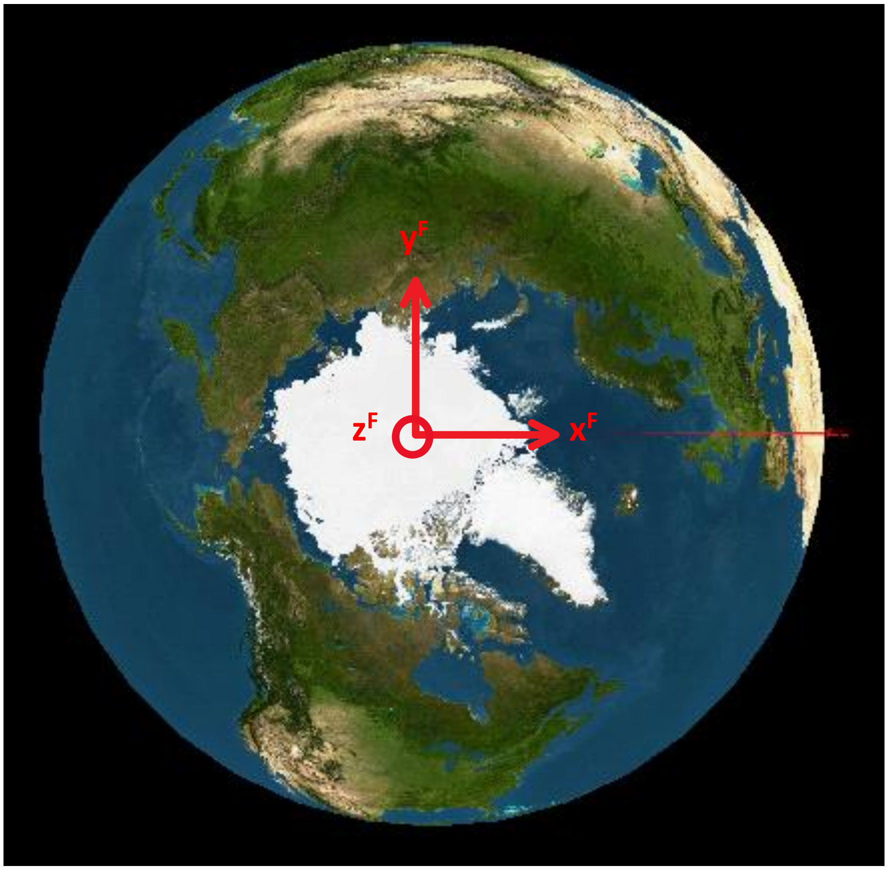
$$q_2 = \frac{1}{4q_4}\left(DCM(3,1) - DCM(1,3)\right)$$

$$q_3 = \frac{1}{4q_4}\left(DCM(1,2) - DCM(2,1)\right)$$

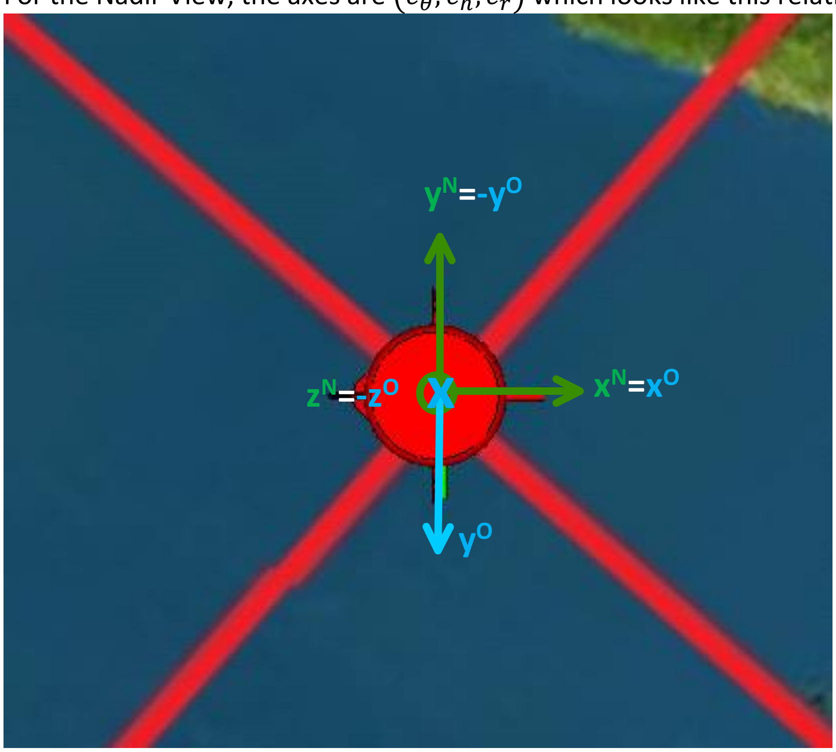
The only location where this would not work is if the Euler Axis, ϕ, is *exactly* π, although it is not computationally accurate as ϕ approaches π. We will ignore this possibility in the effort of ease of use and brevity.

VR Nadir View

The VR figure windows all use a standard x, y, z coordinate frame, e.g., for the Fixed Frame:



For the Nadir View, the axes are $(\hat{e}_\theta, \hat{e}_h, \hat{e}_r)$ which looks like this relative to the Orbit Frame we have defined:



Thus, the difference between the Nadir View and the Orbit Frame is simply a rotation of π radians about the x axis. In terms of a quaternion this is:

$$\overline{q}_{N/O}=\begin{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\sin\left(\frac{\pi}{2}\right)\\ \cos\left(\frac{\pi}{2}\right)\end{bmatrix}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

Since we already have the quaternion, Orbit Frame relative to Fixed Frame, $\overline{q}_{O/F}$, above, we can use quaternion multiplication to determine the quaternion, Nadir View relative to Fixed Frame:

$$\overline{q}_{N/F}=\overline{q}_{O/F}\otimes\overline{q}_{N/O}=\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&-1&0&0\\-1&0&0&0\end{bmatrix}\overline{q}_{O/F}$$

Then just convert the quaternion into the Euler Parameters needed by the VRSink block for NadirView.orientation.

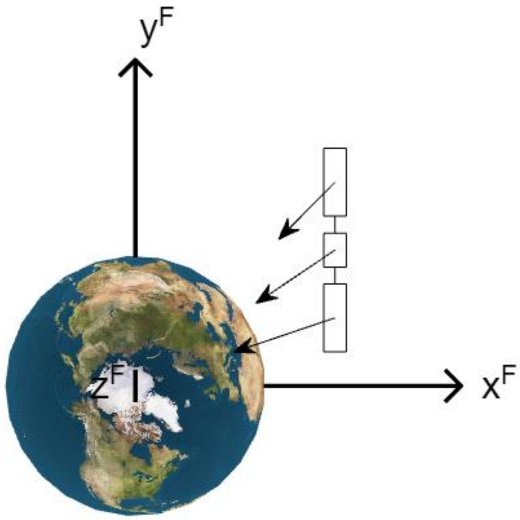
The position of the Nadir View camera has to be along the radial direction, since the camera will be pointed negative radial to the center of the Earth. It must be a little past the spacecraft such that the spacecraft is between the camera and the Earth. Thus, we can take the translation of the spacecraft from SIMULINK 3 and add a constant distance in the radial direction to it as the position of the Nadir View camera.

$$\frac{\overline{r}^F}{6378}+c\frac{\overline{r}^F}{r^F}$$

Gravity-Gradient Torque

Wednesday, January 15, 2020 9:13 PM

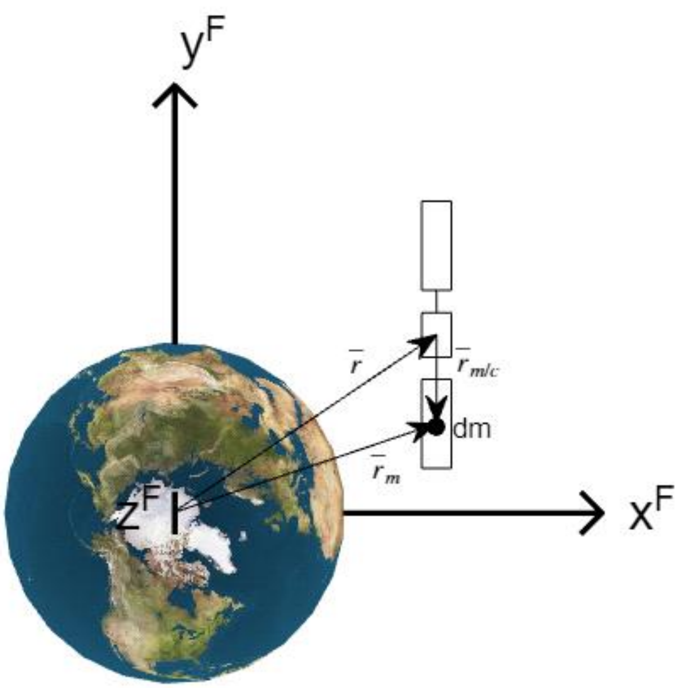
What is gravity-gradient torque?



Since spacecrafts are not particles, point-masses, some of the mass of a spacecraft is closer to the Earth than other parts. The closer parts of a spacecraft experience more gravity than parts that are farther away. This gradient of gravity forces acting along a spacecraft cause a torque about the center of mass of the spacecraft.

How do we determine this torque?

Start by determining the force on the spacecraft due to gravity.



Force due to gravity is proportional to the product of the masses and inversely proportional to the distance between the masses squared. Using the above diagram for notation we have:

$$\bar{F}_g = \int_{m_{s/c}} G\frac{m_\oplus}{r_m^2}\left(-\frac{\bar{r}_m}{r_m}\right)dm = \int_{m_{s/c}} -\frac{\mu_\oplus}{r_m^3}\bar{r}_m dm$$

where G is the gravitational constant and μ is the gravitational parameter of Earth. Next, write it in terms of the position of the center of mass of the spacecraft and the position of the differential mass, dm, relative to the center of mass:

$$\bar{F}_g = -\mu_\oplus \int_{m_{s/c}} \frac{\bar{r}+\bar{r}_{m/c}}{\left\|\bar{r}+\bar{r}_{m/c}\right\|^3}dm = -\mu_\oplus \int_{m_{s/c}} \frac{\bar{r}+\bar{r}_{m/c}}{\left[\left(\bar{r}+\bar{r}_{m/c}\right)\cdot\left(\bar{r}+\bar{r}_{m/c}\right)\right]^{3/2}}dm$$

Now, expand the denominator with the knowledge that the position of the center of mass, $\bar{r}$, does not depend on the differential mass so it can be pulled out of the integral:

$$\bar{F}_g = -\mu_\oplus \int_{m_{s/c}} \frac{\bar{r}+\bar{r}_{m/c}}{\left(r^2+2\bar{r}\cdot\bar{r}_{m/c}+\bar{r}_{m/c}\cdot\bar{r}_{m/c}\right)^{3/2}}dm = -\frac{\mu_\oplus}{r^3}\int_{m_{s/c}} \frac{\bar{r}+\bar{r}_{m/c}}{\left(1+\frac{2\bar{r}\cdot\bar{r}_{m/c}}{r^2}+\frac{\bar{r}_{m/c}\cdot\bar{r}_{m/c}}{r^2}\right)^{3/2}}dm$$

Next, we simplify the integrand denominator term by using a McLaurin's Series (Taylor Series about zero):

$$f(x) \approx f(0)+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+\cdots$$

$$f(\bar{r}_{m/c}) = \frac{1}{\left(1+\frac{2\bar{r}\cdot\bar{r}_{m/c}}{r^2}+\frac{\bar{r}_{m/c}\cdot\bar{r}_{m/c}}{r^2}\right)^{3/2}} \approx 1+\underbrace{\frac{-\frac{3}{2}\left(2\frac{\bar{r}}{r^2}+\frac{\bar{r}_{m/c}}{r^2}\right)}{\left(1+\frac{2\bar{r}\cdot\bar{r}_{m/c}}{r^2}+\frac{\bar{r}_{m/c}\cdot\bar{r}_{m/c}}{r^2}\right)^{5/2}}}_{\text{eval at } \bar{r}_{m/c}=\bar{0}}\cdot\bar{r}_{m/c}+\cdots$$

$$\frac{1}{\left(1+\frac{2\bar{r}\cdot\bar{r}_{m/c}}{r^2}+\frac{\bar{r}_{m/c}\cdot\bar{r}_{m/c}}{r^2}\right)^{3/2}} \approx 1-3\frac{\bar{r}\cdot\bar{r}_{m/c}}{r^2}+\cdots$$

Finally, plug this approximation back into the expression for the force due to gravity on the spacecraft:

$$\bar{F}_g \approx -\frac{\mu_\oplus}{r^3}\int_{m_{s/c}}\left(\bar{r}+\bar{r}_{m/c}\right)\left(1-3\frac{\bar{r}\cdot\bar{r}_{m/c}}{r^2}\right)dm$$

To turn this expression into torque we simply use the moment equation: $\bar{T} = \bar{r}\times\bar{F}$

$$\bar{T}_g = \bar{r}_{m/c}\times\bar{F}_g = -\frac{\mu_\oplus}{r^3}\int_{m_{s/c}}\bar{r}_{m/c}\times\left(\bar{r}+\bar{r}_{m/c}\right)\left(1-3\frac{\bar{r}\cdot\bar{r}_{m/c}}{r^2}\right)dm$$

Distributing, the cross product to the two vector terms we get $\bar{r}_{m/c}$ crossed with itself which, by definition of the dot product, is zero. The other cross product I will reverse the order of the terms, and change the signs since $a\times b = -b\times a$, so that the $\bar{r}$ term can be pulled out of the integral:

$$\bar{T}_g = \frac{\mu_\oplus}{r^3}\bar{r}\times\int_{m_{s/c}}\bar{r}_{m/c}\left(1-3\frac{\bar{r}\cdot\bar{r}_{m/c}}{r^2}\right)dm$$

Expanding the integrand we get two integrals:

$$\bar{T}_g = \frac{\mu_\oplus}{r^3}\bar{r}\times\left[\int_{m_{s/c}}\bar{r}_{m/c}dm-\frac{3}{r^2}\int_{m_{s/c}}\bar{r}_{m/c}\left(\bar{r}\cdot\bar{r}_{m/c}\right)dm\right]$$

The first integral is zero by definition of the center of mass. The second integral needs to be expanded further by assuming that the position of the center of mass has coordinates (x^F, y^F, z^F) and the position of each differential mass, dm, relative to the center of mass has coordinates (x, y, z):

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \int_{m_{s/c}} -\begin{bmatrix} x \\ y \\ z \end{bmatrix}\left(\begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) dm$$

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \int_{m_{s/c}} -\begin{bmatrix} x \\ y \\ z \end{bmatrix}\left(xx^F + yy^F + zz^F\right) dm$$

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \int_{m_{s/c}} -\begin{bmatrix} x\left(xx^F + yy^F + zz^F\right) \\ y\left(xx^F + yy^F + zz^F\right) \\ z\left(xx^F + yy^F + zz^F\right) \end{bmatrix} dm$$

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \left(\int_{m_{s/c}} \begin{bmatrix} -x^2 & -xy & -xz \\ -xy & -y^2 & -yz \\ -xz & -yz & -z^2 \end{bmatrix} dm\right)\begin{bmatrix} x^F \\ y^F \\ z^F \end{bmatrix}$$

Now the integral term is nearly the definition of the mass moment of inertia matrix, however, the main diagonal terms are all wrong by the term: $x^2 + y^2 + z^2$. So, we will add and subtract the same term to yield the mass moment of inertia matrix:

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \left[\left(\int_{m_{s/c}} \begin{bmatrix} -x^2 & -xy & -xz \\ -xy & -y^2 & -yz \\ -xz & -yz & -z^2 \end{bmatrix} dm\right)\overline{r} + \left(\int_{m_{s/c}} x^2 + y^2 + z^2 dm\right)\overline{r} - \left(\int_{m_{s/c}} x^2 + y^2 + z^2 dm\right)\overline{r}\right]$$

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times \left[\left(\int_{m_{s/c}} \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} dm\right)\overline{r} - \left(\int_{m_{s/c}} r_{m/c}^2 dm\right)\overline{r}\right]$$

The second term in the above equation is zero since $\overline{r} \times \overline{r} = \overline{0}$. Thus, with the first integral term equal to the mass moment of inertia, we have:

$$\overline{T}_g = \frac{3\mu_\oplus}{r^5}\overline{r} \times [I]\overline{r}$$

Gravity Gradient Torque Model

Sunday, March 3, 2024 6:59 PM

Here, we will derive the linearized EOM for a spacecraft, in a circular orbit, that is near the Orbit Frame, and with the inclusion of gravity-gradient torque. We start with the sum of moments equation:

$$\sum \bar{M}_{cm}^{B} = \frac{d^{B}}{dt}\left([I]^{B}\bar{\omega}_{B/F}^{B}\right) + \bar{\omega}_{B/F}^{B} \times [I]^{B}\bar{\omega}_{B/F}^{B} = \bar{T}_{g}^{B}$$

Let's begin by stating that we are going to use the Euler Angles and sequence 3-2-1 (YPR) to parameterize the Body Frame relative to the Orbit Frame. These angles are assumed small since the Body Frame is near the Orbit Frame such that:

$$\sin(\theta_i) \approx \theta_i, \cos(\theta_i) \approx 1 \quad \text{for } i = 1,2,3$$

$$\theta_i\theta_j \approx 0 \quad \text{for } i,j = 1,2,3$$

Given this simplification, the Direction Cosine Matrix from the Orbit Frame to the Body Frame can be written as:

$$DCM_O^B = R_{321} = R_1(\theta_3)R_2(\theta_2)R_3(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix}\begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix}\begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM_O^B \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \theta_3 \\ 0 & -\theta_3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -\theta_2 \\ 0 & 1 & 0 \\ \theta_2 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & \theta_1 & 0 \\ -\theta_1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\theta_2 \\ \cancel{\theta_2\theta_3}^{0} & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix}\begin{bmatrix} 1 & \theta_1 & 0 \\ -\theta_1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM_O^B \approx \begin{bmatrix} 1 & \theta_1 & -\theta_2 \\ \theta_1 & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix}$$

The first term we determine is the angular velocity of the Orbit Frame relative to the Fixed Frame. From the SIMULINK 4 lecture, we wrote that the Orbit Frame rotates about the negative y-Orbit axis with magnitude equal to the time-rate-of-change of the true anomaly. Since we are assuming that our spacecraft is in a circular orbit, this is simply the mean motion of the orbit. From Space Mechanics:

$$n = \sqrt{\frac{\mu}{r^3}}$$

Thus:

$$\bar{\omega}_{O/F}^{O} = \begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix}$$

$$\bar{\omega}_{O/F}^{B} = DCM(\bar{q}_{B/O})\begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} \approx \begin{bmatrix} 1 & \theta_1 & -\theta_2 \\ \theta_1 & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} = \begin{bmatrix} -\theta_1 \\ -1 \\ \theta_3 \end{bmatrix} n$$

We need to add this term to the angular velocity of the Body Frame relative to the Orbit Frame to get the angular velocity of the Body Frame relative to the Fixed Frame:

$$\bar{\omega}_{B/F}^{B} = \bar{\omega}_{O/F}^{B} + \bar{\omega}_{B/O}^{B}$$

To determine $\bar{\omega}_{B/O}^{B}$, we use the Euler Angle equation of motion problem:

$$\bar{\omega}_{B/O}^{B}=\sum_{k=1}^{3}\dot{\bar{\theta}}_{k}^{B}=R_1(\theta_3)R_2(\theta_2)\begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix}+R_1(\theta_3)\begin{bmatrix}0\\\dot{\theta}_2\\0\end{bmatrix}+\begin{bmatrix}\dot{\theta}_3\\0\\0\end{bmatrix}$$

$$\bar{\omega}_{B/O}^{B}\approx\begin{bmatrix}1&0&-\theta_2\\0&1&\theta_3\\\theta_2&-\theta_3&1\end{bmatrix}\begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix}+\begin{bmatrix}1&0&0\\0&1&\theta_3\\0&-\theta_3&1\end{bmatrix}\begin{bmatrix}0\\\dot{\theta}_2\\0\end{bmatrix}+\begin{bmatrix}\dot{\theta}_3\\0\\0\end{bmatrix}$$

$$\bar{\omega}_{B/O}^{B}\approx\begin{bmatrix}-\theta_2&0&1\\\theta_3&1&0\\1&-\theta_3&0\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\\dot{\theta}_2\\\dot{\theta}_3\end{bmatrix}$$

Finally, if we assume that the time-rates-of-change of the Euler Angles are also small such that:

$$\theta_i\dot{\theta}_j\approx 0\quad\text{for } i,j=1,2,3$$

Then:

$$\bar{\omega}_{B/O}^{B}\approx\begin{bmatrix}\dot{\theta}_3\\\dot{\theta}_2\\\dot{\theta}_1\end{bmatrix}$$

Now we add these two angular velocity terms together to yield the angular velocity of the Body Frame relative to the Fixed Frame:

$$\bar{\omega}_{B/F}^{B}=\bar{\omega}_{B/O}^{B}+\bar{\omega}_{O/F}^{B}\approx\begin{bmatrix}\dot{\theta}_3\\\dot{\theta}_2\\\dot{\theta}_1\end{bmatrix}+\begin{bmatrix}-\theta_1\\-1\\\theta_3\end{bmatrix}n=\begin{bmatrix}\dot{\theta}_3-\theta_1 n\\\dot{\theta}_2-n\\\dot{\theta}_1+\theta_3 n\end{bmatrix}$$

Now, looking at the first term of the sum of moments equation, we need the derivative of this angular velocity. The derivative yields:

$$\dot{\bar{\omega}}_{B/F}^{B}\approx\begin{bmatrix}\ddot{\theta}_3-\dot{\theta}_1 n\\\ddot{\theta}_2\\\ddot{\theta}_1+\dot{\theta}_3 n\end{bmatrix}$$

Using a Body Frame aligned with the Principle Axes of the spacecraft such that the mass moment of inertial matrix is diagonal, $\begin{bmatrix}I_1&0&0\\0&I_2&0\\0&0&I_3\end{bmatrix}$, the first term in the sum of moments equation is:

$$\frac{d^B}{dt}\left([I]^B\bar{\omega}_{B/F}^{B}\right)\approx\begin{bmatrix}I_1\left(\ddot{\theta}_3-\dot{\theta}_1 n\right)\\I_2\ddot{\theta}_2\\I_3\left(\ddot{\theta}_1+\dot{\theta}_3 n\right)\end{bmatrix}$$

The second term, the cross-product term, can be evaluated using the determinate of a matrix:

$$\bar{\omega}^B_{B/F}\times[I]^B\bar{\omega}^B_{B/F}\approx\begin{bmatrix} x^B & y^B & z^B \\ \dot{\theta}_3-\theta_1 n & \dot{\theta}_2-n & \dot{\theta}_1+\theta_3 n \\ I_1\left(\dot{\theta}_3-\theta_1 n\right) & I_2\left(\dot{\theta}_2-n\right) & I_3\left(\dot{\theta}_1+\theta_3 n\right)\end{bmatrix}=\begin{bmatrix}(I_3-I_2)\left(\dot{\theta}_1+\theta_3 n\right)\left(\dot{\theta}_2-n\right) \\ (I_1-I_3)\left(\dot{\theta}_1+\theta_3 n\right)\left(\dot{\theta}_3-\theta_1 n\right) \\ (I_2-I_1)\left(\dot{\theta}_2-n\right)\left(\dot{\theta}_3-\theta_1 n\right)\end{bmatrix}$$

This can be simplified significantly using the previous simplifying assumptions plus and Euler Angle time-rate-of-change times another Euler Angle time-rate-of-change is also negligible:

$$\dot{\theta}_i\dot{\theta}_j\approx 0$$

This yields:

$$\bar{\omega}^B_{B/F}\times[I]^B\bar{\omega}^B_{B/F}\approx\begin{bmatrix}(I_3-I_2)\left(-n\dot{\theta}_1-\theta_3 n^2\right) \\ 0 \\ (I_2-I_1)\left(-\dot{\theta}_3 n+\theta_1 n^2\right)\end{bmatrix}$$

Last, we have to determine the gravity-gradient torque term on the right-hand side. Using the equation of gravity-gradient torque we derived earlier:

$$\bar{T}_g=\frac{3\mu}{r^3}\hat{r}\times[I]\hat{r}$$

First, we notice that the $\frac{\mu}{r^3}$ term is simply the mean motion squared, n^2. Also, $\hat{r}$, the radial direction, in the Orbit Frame is simply the negative z-Orbit direction. We then rotate this into the Body Frame using the DCM:

$$\hat{r}^O=\begin{bmatrix}0\\0\\-1\end{bmatrix}$$

$$\hat{r}^B=DCM^B_O\begin{bmatrix}0\\0\\-1\end{bmatrix}\approx\begin{bmatrix}1 & \theta_1 & -\theta_2 \\ \theta_1 & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1\end{bmatrix}\begin{bmatrix}0\\0\\-1\end{bmatrix}=\begin{bmatrix}\theta_2\\-\theta_3\\-1\end{bmatrix}$$

Now we evaluate the cross-product term:

$$\hat{r}\times[I]\hat{r}\approx\begin{bmatrix}x^B & y^B & z^B \\ \theta_2 & -\theta_3 & -1 \\ I_1\theta_2 & -I_2\theta_3 & -I_3\end{bmatrix}=\begin{bmatrix}(I_3-I_2)\theta_3 \\ (I_3-I_1)\theta_2 \\ (I_1-I_2)\cancelto{0}{\theta_2\theta_3}\end{bmatrix}=\begin{bmatrix}(I_3-I_2)\theta_3 \\ (I_3-I_1)\theta_2 \\ 0\end{bmatrix}$$

Finally, the gravity-gradient torque term is:

$$\bar{T}^B_g=3n^2\begin{bmatrix}(I_3-I_2)\theta_3 \\ (I_3-I_1)\theta_2 \\ 0\end{bmatrix}$$

The entire sum of moments equation is then:

$$\begin{bmatrix} I_1\left(\ddot{\theta}_3-\dot{\theta}_1 n\right) \\ I_2\ddot{\theta}_2 \\ I_3\left(\ddot{\theta}_1+\dot{\theta}_3 n\right) \end{bmatrix} + \begin{bmatrix} \left(I_3-I_2\right)\left(-n\dot{\theta}_1-\theta_3 n^2\right) \\ 0 \\ \left(I_2-I_1\right)\left(-\dot{\theta}_3 n+\theta_1 n^2\right) \end{bmatrix} = 3n^2 \begin{bmatrix} \left(I_3-I_2\right)\theta_3 \\ \left(I_3-I_1\right)\theta_2 \\ 0 \end{bmatrix}$$

Next, we collect like terms in each equation:

$$\begin{bmatrix} I_1\ddot{\theta}_3+\left(-I_1+I_2-I_3\right)n\dot{\theta}_1+4\left(I_2-I_3\right)n^2\theta_3 \\ I_2\ddot{\theta}_2+3\left(I_1-I_3\right)n^2\theta_2 \\ I_3\ddot{\theta}_1+\left(I_1-I_2+I_3\right)n\dot{\theta}_3+\left(I_2-I_1\right)n^2\theta_1 \end{bmatrix} = \overline{0}$$

We will want to solve these equations for the stability of the spacecraft relative to the Orbit Frame, however, since the mass moment of inertia values can be anything positive, it would be better to use parameters that are confined to a smaller domain. Let us use the following mass moment of inertia parameters:

$$k_1 = \frac{I_2-I_3}{I_1}, k_2 = \frac{I_1-I_3}{I_2}, k_3 = \frac{I_2-I_1}{I_3}$$

These parameters are bounded to be between -1 and 1. At the limits of the domain, the shape is an infinitely long slender rod. At the origin, the shape is a perfect sphere. Substituting these parameters in for the mass moment of inertia values we have:

$$\begin{bmatrix} \ddot{\theta}_3+\left(k_1-1\right)n\dot{\theta}_1+4k_1 n^2\theta_3 \\ \ddot{\theta}_2+3k_2 n^2\theta_2 \\ \ddot{\theta}_1+\left(1-k_3\right)n\dot{\theta}_3+k_3 n^2\theta_1 \end{bmatrix} = \overline{0}$$

We will solve this linear, coupled, 2nd-order ordinary differential equations next.

Stability of Gravity-Gradient EOMs

Sunday, March 3, 2024 8:37 PM

Let's start with the EOMs we ended with previously:

$$\begin{bmatrix} \ddot{\theta}_3 + (k_1 - 1) n\dot{\theta}_1 + 4k_1 n^2 \theta_3 \\ \ddot{\theta}_2 + 3k_2 n^2 \theta_2 \\ \ddot{\theta}_1 + (1 - k_3) n\dot{\theta}_3 + k_3 n^2 \theta_1 \end{bmatrix} = \overline{0}$$

First, we note the y-axis, or Pitch, equation is un-coupled from the x- and z-axis, or Roll and Yaw, equations respectively. We also note that the y-axis is the axis about which we are rotating to always point towards the center of the Earth, nadir. Solving the y-axis equation first, we assume the exponential solution and plug this information into the ODE:

$$\theta_2(t) = c_2 e^{\lambda t}$$

$$\text{then}: \ddot{\theta}_2 = \lambda^2 c_2 e^{\lambda t}$$

$$\underbrace{c_2}_{\neq 0} \underbrace{e^{\lambda t}}_{\neq 0} \underbrace{\left(\lambda^2 + 3k_2 n^2\right)}_{=0} = 0$$

The constant, c_2, can't be zero because, if it was, then θ_2 would be zero forever and it isn't. θ_2 starts small but not zero. Also, the exponential term can never be zero; it can approach zero exponentially but never actually equal zero. Thus, the term in parenthesis must be zero; we call this term the "characteristic equation." Solving for λ we get:

$$\lambda^2 + 3k_2 n^2 = 0$$

$$\lambda = \pm in\sqrt{3k_2}$$

As long as $k_2 > 0$, then λ will be pure imaginary and θ_2 will be a sinusoid. This is stable since θ_2 starts small, it will stay small in a sinusoid. If, however, $k_2 < 0$, then λ will have one positive and one negative real root. The positive real root will make θ_2 an exponentially increasing function, unstable. Thus the stability criteria for the y, Pitch, axis is:

$$k_2 > 0$$

Now, we solve the coupled, x- and z-axis, or Roll and Yaw, equations. Starting with the same step, we let θ_1 and θ_3 be the exponential function:

$$\theta_1(t) = c_1 e^{\kappa t}, \text{and } \theta_3(t) = c_3 e^{\kappa t}$$

$$\text{then}: \dot{\theta}_1(t) = \kappa c_1 e^{\kappa t}, \text{and } \dot{\theta}_3(t) = \kappa c_3 e^{\kappa t}$$

$$\text{and}: \ddot{\theta}_1(t) = \kappa^2 c_1 e^{\kappa t}, \text{and } \ddot{\theta}_3(t) = \kappa^2 c_3 e^{\kappa t}$$

Now, plug these expressions into the x- and z-axis ODEs:

$$e^{\kappa t}\left(\kappa^2 c_3 + (k_1 - 1) n\kappa c_1 + 4k_1 n^2 c_3\right) = 0$$

$$e^{\kappa t}\left(\kappa^2 c_1 + (1 - k_3) n\kappa c_3 + k_3 n^2 c_1\right) = 0$$

Just like the y-axis, the exponential function cannot be zero thus the terms inside of the parenthesis must be zero. These equations have 3 unknowns, $\kappa, c_1, \text{and } c_3$ and are nonlinear in terms of κ but are linear in terms of the constants c_1 and c_3. Let's write the 2 equations in matrix/vector form in terms of the constants:

$$\begin{bmatrix} (k_1-1)n\kappa & \kappa^2+4k_1n^2 \\ \kappa^2+k_3n^2 & (1-k_3)n\kappa \end{bmatrix} \underbrace{\begin{bmatrix} c_1 \\ c_3 \end{bmatrix}}_{\neq\bar{0}} = \bar{0}$$

In these equations, the vector of c_1 and c_3 cannot be zero because that would mean that θ_1 and θ_3 are zero forever and they are not zero at the beginning. Thus, the matrix must be singular, i.e., it must not have an inverse, its' rows are dependent, not independent. A singular matrix has a determinate of zero, so we take the derivative of the matrix and set it equal to zero.

$$(k_1-1)(1-k_3)n^2\kappa^2-(\kappa^2+4k_1n^2)(\kappa^2+k_3n^2)=0$$

Now, simplify, by FOIL-ing and collecting terms, this into its' quartic form with a positive leading polynomial coefficient:

$$\kappa^4+(-k_1+k_1k_3+1-k_3+k_3+4k_1)n^2\kappa^2+4k_1k_3n^4=0$$

$$\kappa^4+(1+3k_1+k_1k_3)n^2\kappa^2+4k_1k_3n^4=0$$

Although this is a quartic equation, since there is no cubic or linear term, it is simply a quadratic in terms of κ^2. We use the quadratic formula to solve:

$$\kappa^2=\frac{-(1+3k_1+k_1k_3)n^2\pm\sqrt{(1+3k_1+k_1k_3)^2n^4-16k_1k_3n^4}}{2}$$

$$\kappa^2=\frac{n^2}{2}\left(-(1+3k_1+k_1k_3)\pm\sqrt{(1+3k_1+k_1k_3)^2-16k_1k_3}\right)$$

For stability, κ must be pure imaginary leading to sinusoidal solutions for the Euler Angles θ_1 and θ_3, thus, κ^2 must have two negative real roots. This means that the leading term in the quadratic formula must be negative.

$$-(1+3k_1+k_1k_3)<0$$

$$(1+3k_1+k_1k_3)>0$$

Then, the square root term must have a smaller magnitude than the leading term so that when the plus sign is used, the number stays negative. Hence:

$$\sqrt{(1+3k_1+k_1k_3)^2-16k_1k_3}<(1+3k_1+k_1k_3)$$

$$-16k_1k_3<0$$

$$k_1k_3>0$$

Finally, the product k_1k_3 must be positive, but not too large. The term under the square root must remain positive so as to avoid having a complex solution for κ^2. Thus:

$$(1+3k_1+k_1k_3)^2-16k_1k_3>0$$

Then, if all three of these conditions are true, κ^2 will have 2 negative real root, making κ have 4 pure imaginary roots such that the Euler Angles θ_1 and θ_3 are pure sinusoids, a stable solution.

$$k_1k_3>0$$

$$(1+3k_1+k_1k_3)>0$$

$$(1+3k_1+k_1k_3)^2-16k_1k_3>0$$

Since these three conditions for stability only use the k_1 and k_3 parameters, it would be nice if the y-axis stability criteria, k_2>0, would also be in terms of k_1 and k_3. First, using the definition of k_2:

$$k_2 = \frac{I_1 - I_3}{I_2} > 0$$

$$I_1 > I_3$$

Then, using the definition of *k1* and *k3*, we solve both of these expressions for I_2, the mass moment of inertia not in the above inequality:

$$k_1 = \frac{I_2 - I_3}{I_1}, k_3 = \frac{I_2 - I_1}{I_3}$$

$$I_2 = I_1 k_1 + I_3 = I_3 k_3 + I_1$$

Now collect terms and solve for I_1 (or I_3):

$$I_1 (1 - k_1) = I_3 (1 - k_3)$$

$$I_1 = I_3 \frac{1 - k_3}{1 - k_1}$$

Substitute this into the I_1>I_3 inequality:

$$I_3 \frac{1 - k_3}{1 - k_1} > I_3$$

$$\frac{1 - k_3}{1 - k_1} > 1$$

Since, each k parameter is bounded between -1 and 1, the denominator is positive and thus doesn't change the direction of the inequality when getting that term on the right-hand side. Then simplify to get:

$$1 - k_3 > 1 - k_1$$

$$-k_3 > -k_1$$

$$k_3 < k_1$$

Thus, for all three axes to be stable, for the spacecraft to stay near the Orbit Frame under to influence of gravity-gradient torque, the following four conditions must be met:

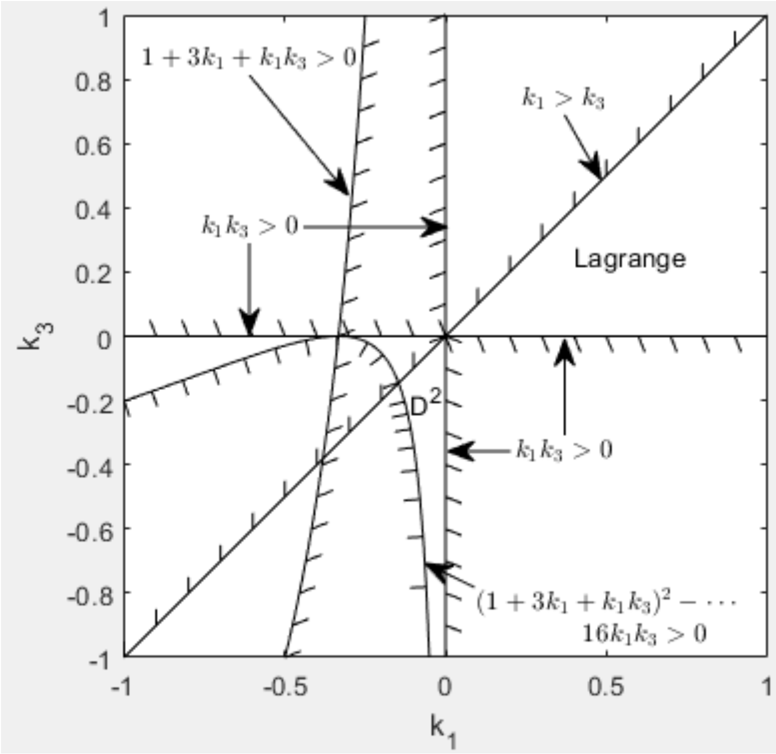
$$k_1 > k_3$$

$$k_1 k_3 > 0$$

$$(1 + 3k_1 + k_1 k_3) > 0$$

$$(1 + 3k_1 + k_1 k_3)^2 - 16 k_1 k_3 > 0$$

Now we can plot the stability space in the k_1, k_3 plane. We start with the y-axis criteria, k_1>k_3. The next criteria, k_1k_3>0 removes the 4th quadrant and the 2nd quadrant (which already was removed by the first criteria).The third criteria removes the left-side of the curve (since the origin satisfies the inequality). Finally, the last of the 4 criteria removes inside the curve (since the origin satisfies the inequality).This leaves only 2 regions of stability: In the 1st quadrant we have the Lagrange Region of stability and in the 3rd quadrant we have the DeBra-Delp (D^2) region, first published in 1961.



SIMULINK 5: Incorporate the gravity-gradient torque model into your completed SIMULINK 4 assignment. Test both stable, one in the Lagrange Region and one in the Debra-Delp Region, and unstable mass MOI matrices to confirm your model. Start at 1 degree Yaw, Pitch, and Roll from the orbit frame. Do this by adding lines in the InitFcn to take in a given Yaw, Pitch, and Roll value and determine the initial quaternion using the 3-2-1 sequence.

$$I=\begin{bmatrix}3&0&0\\0&4&0\\0&0&2\end{bmatrix}, k_1=\frac{2}{3}, k_3=\frac{1}{2}, \text{Lagrange Region, Stable}$$

$$I=\begin{bmatrix}60&0&0\\0&39&0\\0&0&42\end{bmatrix}, k_1=-\frac{1}{20}, k_3=-\frac{1}{2}, \text{DeBra-Delp Region, Stable}$$

$$I=\begin{bmatrix}2&0&0\\0&3&0\\0&0&4\end{bmatrix}, k_1=-\frac{1}{2}, k_3=\frac{1}{4}, \text{Unstable}$$

Introduction

Wednesday, February 19, 2020 1:42 PM

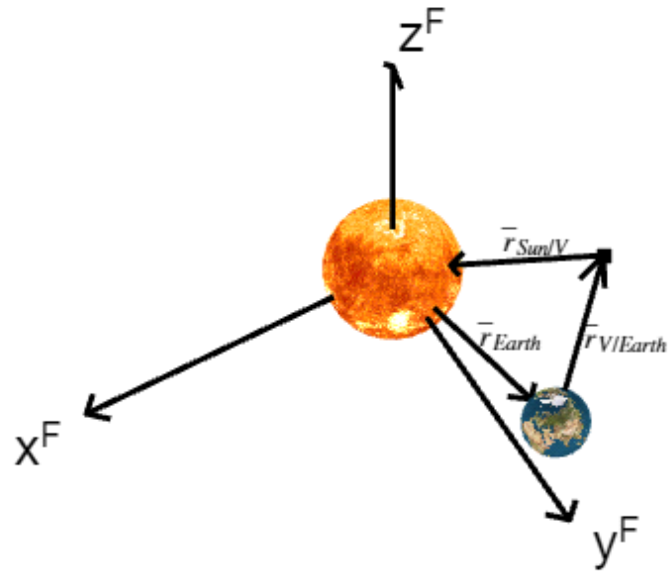
Static Attitude Determination (q):

Sensors are based on the fact that they measure direction, not magnitude, e.g., a sun sensor measures the direction of the sun not how far the sun is from the spacecraft. This yields:

$$\hat{v}_k^B \text{ for } k = 1\ldots N \text{ number of measurements}$$

Then, we can calculate the direction of the vector in the Inertial frame based on a mathematical model, e.g., calculating the position of the sun relative to the sun.

$$\hat{r}_{\odot/v} = -\frac{\overline{r}_{\oplus} + \overline{r}}{\left\|\overline{r}_{\oplus} + \overline{r}\right\|}$$



In general this would be:

$$\overline{v}_k^F \text{ for } k = 1\ldots N$$

Thus:

$$\hat{v}_k^B = DCM_F^B \overline{v}_k^F \text{ for } k = 1\ldots N$$

Since the vectors are unit vectors, they represent only 2 independent equations. The Static Attitude Determination problem has 3 independent values (3 of the 4 values of the unit magnitude quaternion, 3 Euler Angles).

Thus, we require a minimum of 2 sensor measurements resulting in a minimum of 4 independent equations when only 3 are required to solve for the 3 unknowns. Therefore, the Static Attitude Determination problem is over-determined and is really an Attitude Estimation problem.

q-method

Saturday, February 22, 2020 11:51 AM

$$\min J = \frac{1}{2}\sum_{k=1}^{N} w_k \left\| \hat{v}_k^B - DCM_F^B \hat{v}_k^F \right\|^2$$

Cost function, J, is the weighted, least-squared, difference between the sensor measurements in the body frame and the mathematical model of those vectors in the inertial rotated into the body frame. Since the difference is squared, the cost must be positive; this will be important later.

First, calculate the square by taking the vector transpose times itself and then FOIL:

$$\min J = \frac{1}{2}\sum_{k=1}^{N} w_k \left(\hat{v}_k^B - DCM_F^B \hat{v}_k^F \right)^T \left(\hat{v}_k^B - DCM_F^B \hat{v}_k^F \right)$$

$$\min J = \frac{1}{2}\sum_{k=1}^{N} w_k \left(\underbrace{\hat{v}_k^{B^T} \hat{v}_k^B}_{1} - \hat{v}_k^{B^T} DCM_F^B \hat{v}_k^F - \hat{v}_k^{F^T} DCM_F^{B^T} \hat{v}_k^B + \underbrace{\hat{v}_k^{F^T} \underbrace{DCM_F^{B^T} DCM_F^B}_{I_{3x3}} \hat{v}_k^F}_{1} \right)$$

Since the vectors are unit vectors, and a DCM times its' transpose is the identity matrix, the first and last terms equal 1. The 2 middle terms, both scalars, are just the transposes of each other. Since the transpose of a scalar is itself, these two terms are the same, thus:

$$\min J = \frac{1}{2}\sum_{k=1}^{N} w_k \left(2 - 2\hat{v}_k^{B^T} DCM_F^B \hat{v}_k^F \right)$$

Next, cancel the 1/2 and the 2's and distribute the weight term:

$$\min J = \sum_{k=1}^{N} w_k - \sum_{k=1}^{N} w_k \hat{v}_k^{B^T} DCM_F^B \hat{v}_k^F$$

Since, J must be positive, then we rearrange to eliminate the constant, sum of weights term:

$$\min J = \sum_{k=1}^{N} w_k - \sum_{k=1}^{N} w_k \hat{v}_k^{B^T} DCM_F^B \hat{v}_k^F \gtrsim 0$$

$$\min g = -\sum_{k=1}^{N} w_k \hat{v}_k^{B^T} DCM_F^B \hat{v}_k^F \gtrsim -\sum_{k=1}^{N} w_k$$

Remember, we desire to solve for our attitude, Body Frame relative to Fixed Frame, which is currently written as a Direction Cosine Matrix. Next, we write the DCM in terms of the quaternion using the formula:

$$DCM(\bar{q}) = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Entering this into our function we wish to minimize and expanding our Body Frame and Fixed Frame vectors into their 3-components we have:

$$\min g = -\sum_{k=1}^{N} w_k \begin{bmatrix} v_1^B & v_2^B & v_3^B \end{bmatrix}_k \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} v_1^F \\ v_2^F \\ v_3^F \end{bmatrix}_k$$

Here we have the quaternion information in the matrix and the vector information, both Body Frame and Fixed Frame, in the vectors. We wish to flip this so that the quaternion is the vectors and the vector information is the in the matrix, i.e.:

$$\min g = -\bar{q}^T [K] \bar{q}$$

To do this, we multiply everything out, into its scalar form, and then factor out the quaternion vectors.

$$\min g = -\sum_{k=1}^{N} w_k \begin{bmatrix} v_1^B\left(q_4^2+q_1^2-q_2^2-q_3^2\right)+2v_2^B\left(q_1q_2-q_3q_4\right)+2v_3^B\left(q_1q_3+q_2q_4\right) \\ 2v_1^B\left(q_1q_2+q_3q_4\right)+v_2^B\left(q_4^2-q_1^2+q_2^2-q_3^2\right)+2v_3^B\left(q_2q_3-q_1q_4\right) \\ 2v_1^B\left(q_1q_3-q_2q_4\right)+2v_2^B\left(q_2q_3+q_1q_4\right)+v_3^B\left(q_4^2-q_1^2-q_2^2+q_3^2\right) \end{bmatrix}_k^T \begin{bmatrix} v_1^F \\ v_2^F \\ v_3^F \end{bmatrix}_k$$

$$\min g = -\sum_{k=1}^{N} w_k \begin{bmatrix} v_1^F\left\{v_1^B\left(q_4^2+q_1^2-q_2^2-q_3^2\right)+2v_2^B\left(q_1q_2-q_3q_4\right)+2v_3^B\left(q_1q_3+q_2q_4\right)\right\}+\cdots \\ v_2^F\left\{2v_1^B\left(q_1q_2+q_3q_4\right)+v_2^B\left(q_4^2-q_1^2+q_2^2-q_3^2\right)+2v_3^B\left(q_2q_3-q_1q_4\right)\right\}+\cdots \\ v_3^F\left\{2v_1^B\left(q_1q_3-q_2q_4\right)+2v_2^B\left(q_2q_3+q_1q_4\right)+v_3^B\left(q_4^2-q_1^2-q_2^2+q_3^2\right)\right\} \end{bmatrix}_k$$

$$\min g = -\sum_{k=1}^{N} w_k \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} v_1^F\left\{v_1^Bq_1+v_2^Bq_2+v_3^Bq_3\right\}+v_2^F\left\{v_1^Bq_2-v_2^Bq_1-v_3^Bq_4\right\}+v_3^F\left\{v_1^Bq_3+v_2^Bq_4-v_3^Bq_1\right\} \\ v_1^F\left\{-v_1^Bq_2+v_2^Bq_1+v_3^Bq_4\right\}+v_2^F\left\{v_1^Bq_1+v_2^Bq_2+v_3^Bq_3\right\}+v_3^F\left\{-v_1^Bq_4+v_2^Bq_3-v_3^Bq_2\right\} \\ v_1^F\left\{-v_1^Bq_3-v_2^Bq_4+v_3^Bq_1\right\}+v_2^F\left\{v_1^Bq_4-v_2^Bq_3+v_3^Bq_2\right\}+v_3^F\left\{v_1^Bq_1+v_2^Bq_2+v_3^Bq_3\right\} \\ v_1^F\left\{v_1^Bq_4-v_2^Bq_3+v_3^Bq_2\right\}+v_2^F\left\{v_1^Bq_3+v_2^Bq_4-v_3^Bq_1\right\}+v_3^F\left\{-v_1^Bq_2+v_2^Bq_1+v_3^Bq_4\right\} \end{bmatrix}_k$$

$$\min g = -\sum_{k=1}^{N} w_k \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} v_1^Bv_1^F-v_2^Bv_2^F-v_3^Bv_3^F & v_2^Bv_1^F+v_1^Bv_2^F & v_3^Bv_1^F+v_1^Bv_3^F & -v_3^Bv_2^F+v_2^Bv_3^F \\ v_2^Bv_1^F+v_1^Bv_2^F & -v_1^Bv_1^F+v_2^Bv_2^F-v_3^Bv_3^F & v_3^Bv_2^F+v_2^Bv_3^F & v_3^Bv_1^F-v_1^Bv_3^F \\ v_3^Bv_1^F+v_1^Bv_3^F & v_3^Bv_2^F+v_2^Bv_3^F & -v_1^Bv_1^F-v_2^Bv_2^F+v_3^Bv_3^F & -v_2^Bv_1^F+v_1^Bv_2^F \\ -v_3^Bv_2^F+v_2^Bv_3^F & v_3^Bv_1^F-v_1^Bv_3^F & -v_2^Bv_1^F+v_1^Bv_2^F & v_1^Bv_1^F+v_2^Bv_2^F+v_3^Bv_3^F \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

$$\min g = -\bar{q}^T [K] \bar{q}$$

$$[B] = \sum_{k=1}^{N} w_k v_k^B v_k^{F^T} = \sum_{k=1}^{N} w_k \begin{bmatrix} v_1^B v_1^F & v_1^B v_2^F & v_1^B v_3^F \\ v_2^B v_1^F & v_2^B v_2^F & v_2^B v_3^F \\ v_3^B v_1^F & v_3^B v_2^F & v_3^B v_3^F \end{bmatrix}$$

$$[S] = [B] + [B]^T = \sum_{k=1}^{N} w_k \begin{bmatrix} \left(v_1^B v_1^F\right)^2 & v_1^B v_2^F + v_2^B v_1^F & v_1^B v_3^F + v_3^B v_1^F \\ v_1^B v_2^F + v_2^B v_1^F & \left(v_2^B v_2^F\right)^2 & v_2^B v_3^F + v_3^B v_2^F \\ v_1^B v_3^F + v_3^B v_1^F & v_2^B v_3^F + v_3^B v_2^F & \left(v_3^B v_3^F\right)^2 \end{bmatrix}$$

$$\sigma = \text{trace}([B]) = v_1^B v_1^F + v_2^B v_2^F + v_3^B v_3^F$$

$$\bar{Z} = \begin{bmatrix} -v_3^B v_2^F + v_2^B v_3^F \\ v_3^B v_1^F - v_1^B v_3^F \\ -v_2^B v_1^F + v_1^B v_2^F \end{bmatrix} = \begin{bmatrix} B(2,3) - B(3,2) \\ B(3,1) - B(1,3) \\ B(1,2) - B(2,1) \end{bmatrix}$$

$$[K] = \begin{bmatrix} [S] - \sigma[I_{3x3}] & \bar{Z} \\ \bar{Z}^T & \sigma \end{bmatrix}$$

Now we have to minimize this function subject to the constraint that the quaternion must be a unit vector:

$$\min g = -\bar{q}^T [K] \bar{q}$$

$$\bar{q}^T \bar{q} = 1$$

$$\min g = -\bar{q}^T [K] \bar{q} + \lambda \underbrace{\left(\bar{q}^T \bar{q} - 1\right)}_{0}$$

where we append the constraint to the cost function using λ, known as a Lagrange Multiplier. Then, to find the minimum, we take the derivative relative to the free variable, the quaternion. Using the product rule, we get:

$$\frac{\partial g}{\partial \bar{q}} = \bar{0} = -\left([K]\bar{q}\right)^T - \bar{q}^T [K] + \lambda\left(\left(\bar{q}\right)^T + \bar{q}^T\right) = -2\bar{q}^T [K] + 2\lambda \bar{q}^T$$

Rearranging and simplifying we have:

$$\bar{q}^T [K] = \lambda \bar{q}^T$$

$$[K]\bar{q} = \lambda \bar{q}$$

This is the popular eigenvalue/eigenvector problem where the Lagrange Multiplier, λ, is the eigenvector and the quaternion, $\bar{q}$, is the eigenvector. The matrix [K] is size 4x4 so there are 4 eigenvalues and associated eigenvectors. To determine which one we need, we substitute into the cost function using the last equation:

$$\min g = -\bar{q}^T \left([K]\bar{q}\right) = -\bar{q}^T \left(\lambda \bar{q}\right) = -\lambda \underbrace{\bar{q}^T \bar{q}}_{1} = -\lambda$$

Given this equation, we want the largest eigenvalue of matrix [K] to minimize the function g. Recall the original cost function J:

$$\min J = \sum_{k=1}^{N} w_k + g = \sum_{k=1}^{N} w_k - \lambda_{max} \gtrsim 0$$

$$\lambda_{max} \lesssim \sum_{k=1}^{N} w_k$$

Using an eigenvalue/eigenvector solver to solve for the eigenvalue closest to the sum of the weights requires only 2 iterations to obtain the eigenvalue to machine precision. Then, using this eigenvalue, solve for the associated eigenvector, the optimal attitude $\bar{q}$. In MATLAB we use the function "eigs".

QUEST Algorithm

Saturday, February 22, 2020 2:16 PM

The QUEST (quaternion estimate) algorithm is based on the q-method solution but instead of solving the eigenvalue/eigenvector problem exactly, we simply estimate the solution of the eigenvalue as equal to the sum of the weights.
From the q-method:

$$[K]\overline{q} = \lambda\overline{q}$$

$$\min g = -\overline{q}^T\left([K]\overline{q}\right) = -\overline{q}^T\left(\lambda\overline{q}\right) = -\lambda\underbrace{\overline{q}^T\overline{q}}_{1} = -\lambda$$

$$\min J = \sum_{k=1}^{N} w_k + g = \sum_{k=1}^{N} w_k - \lambda_{max} \gtrsim 0$$

$$\lambda_{max} \lesssim \sum_{k=1}^{N} w_k$$

Now, here, we simply state that:

$$\lambda_{max} = \sum_{k=1}^{N} w_k$$

and now estimate the eigenvalue by solving a system of 3 equations and 3 unknowns:

$$[K]\overline{q} = \lambda_{\max}\overline{q}$$

$$\begin{bmatrix} [S]-\left(\sigma+\sum_{k=1}^{N} w_k\right)[I_{3x3}] & \overline{Z} \\ \overline{Z}^T & \sigma-\sum_{k=1}^{N} w_k \end{bmatrix}\begin{bmatrix} \hat{q} \\ q_4 \end{bmatrix} = \overline{0}$$

Recall that the quaternion only has 3 independent values since it is a unit vector, thus, for convenience we use the top 3 rows of K to solve:

$$\left([S]-\left(\sigma+\sum_{k=1}^{N} w_k\right)[I_{3x3}]\right)\hat{q} + \overline{Z}q_4 = \overline{0}$$

$$\left([S]-\left(\sigma+\sum_{k=1}^{N} w_k\right)[I_{3x3}]\right)\hat{q} = -\overline{Z}q_4$$

$$\left([S]-\left(\sigma+\sum_{k=1}^{N} w_k\right)[I_{3x3}]\right)\frac{\hat{q}}{q_4} = -\overline{Z}$$

The term we solve for, $\frac{\hat{q}}{q_4}$, is known as the Gibbs-Rodrigues Parameters and is given:

$$\overline{p} = \frac{\hat{q}}{q_4} = \overline{e}\tan\left(\frac{\phi}{2}\right)$$

The most efficient method of solving this system of linear equations involves using Gaussian Elimination with Backward Substitution. MATLAB's "linsolve" function or the format "A\b" works well.
Once the Gibbs-Rodrigues Parameters are known, then we need to determine the quaternion, or any

other parameterization we may desire. To start, we note that the magnitude of the vector is $\tan\frac{\phi}{2}$ and there is a trigonometric identity:

$$1+\tan^2\left(\frac{\phi}{2}\right)=\sec^2\left(\frac{\phi}{2}\right)$$

Using this, we can relate $\bar{p}$ with $\bar{q}$

$$1+\bar{p}\cdot\bar{p}=\frac{1}{\cos^2\left(\frac{\phi}{2}\right)}=\frac{1}{q_4^2}$$

Now, rearranging we have:

$$q_4=\frac{1}{\pm\sqrt{1+\bar{p}\cdot\bar{p}}}$$

We typically choose the positive square-root solution out of convenience, however, we must then use this solution to determine $\hat{q}$ so that it will have the correct signs for the assumed sign of q_4. Luckily, we can just use the Gibbs-Rodrigues definition:

$$\bar{p}=\frac{\hat{q}}{q_4}$$

$$\hat{q}=q_4\bar{p}$$

Then, putting it all together we have the quaternion in terms of the Gibbs-Rodrigues parameters:

$$\bar{q}=\frac{1}{\sqrt{1+\bar{p}\cdot\bar{p}}}\begin{bmatrix}\bar{p}\\ 1\end{bmatrix}$$

Now that we have the quaternion, we can get to any of the other parameterizations we need using previous equations and techniques.

ESOQ2 Algorithm

Tuesday, January 9, 2024 12:51 PM

This algorithm, developed by Daniele Mortari and published in 2000, also starts with the q-method, but, like the QUEST algorithm, solves for a different parameterization. In this algorithm we solve for the Euler Parameters, $\bar{e}, \phi$. Note: if you solve for the exact eigenvalue, λ_{max}, then this algrotihm will result in the same exact solution as the q-method since you will still be solving for the eigenvector. However, if you use the approximation, $\sum_{k=1}^{N} w_k$, then you will get a slightly different (and more efficient) solution to the QUEST algorithm since the QUEST algorithm only uses the top 3 rows of [K] but this algorithm uses all 4 rows.

Let's start with the eigenvalue/eigenvector problem:

$$\begin{bmatrix} [S]-\sigma[I_{3x3}] & \bar{Z} \\ \bar{Z}^T & \sigma \end{bmatrix} \bar{q} = \lambda_{max}\bar{q}$$

$$\begin{bmatrix} [S]-(\sigma+\lambda_{max})[I_{3x3}] & \bar{Z} \\ \bar{Z}^T & \sigma-\lambda_{max} \end{bmatrix} \begin{bmatrix} \bar{e}\sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} = \bar{0}$$

Now we solve the 4th equation for $\cot\left(\frac{\phi}{2}\right)$:

$$\bar{Z}^T\bar{e}\sin\left(\frac{\phi}{2}\right) + (\sigma-\lambda_{max})\cos\left(\frac{\phi}{2}\right) = 0$$

$$\bar{Z}^T\bar{e}\sin\left(\frac{\phi}{2}\right) = (\lambda_{max}-\sigma)\cos\left(\frac{\phi}{2}\right)$$

$$\cot\left(\frac{\phi}{2}\right) = \frac{\bar{Z}^T}{(\lambda_{max}-\sigma)}\bar{e}$$

Now substitute this expression into the equation obtained from the first 3 rows:

$$\left([S]-(\sigma+\lambda_{max})[I_{3x3}]\right)\bar{e}\sin\left(\frac{\phi}{2}\right) + \bar{Z}\cos\left(\frac{\phi}{2}\right) = \bar{0}$$

$$\left([S]-(\sigma+\lambda_{max})[I_{3x3}]\right)\bar{e} + \bar{Z}\cot\left(\frac{\phi}{2}\right) = \bar{0}, \quad \sin\left(\frac{\phi}{2}\right) \neq 0$$

$$\left([S]-(\sigma+\lambda_{max})[I_{3x3}]\right)\bar{e} + \bar{Z}\frac{\bar{Z}^T}{(\lambda_{max}-\sigma)}\bar{e} = \bar{0}$$

$$\left((\sigma-\lambda_{max})[S]-(\sigma+\lambda_{max})(\sigma-\lambda_{max})[I_{3x3}]-\bar{Z}\bar{Z}^T\right)\bar{e} = \bar{0}$$

$$\left((\sigma-\lambda_{max})[S]-\left(\sigma^2-\lambda_{max}^2\right)[I_{3x3}]-\bar{Z}\bar{Z}^T\right)\bar{e} = [M]\bar{e} = \bar{0}$$

Since, $S = B + B^T$, the matrix M is symmetric, thus I can write the matrix M using 6 unique quantities:

$$M = M^T = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} = \begin{bmatrix} m_a & m_x & m_y \\ m_x & m_b & m_z \\ m_y & m_z & m_c \end{bmatrix}$$

Since, $[M]\bar{e} = \bar{0}$, then $\bar{e}$ must be perpendicular to each row of [M]. Thus, we can determine the Euler Axis by using cross products:

$$\bar{e}_1 = \bar{m}_2 \times \bar{m}_3 = \left[m_b m_c - m_z^2, m_y m_z - m_x m_c, m_x m_z - m_y m_b \right]^T$$

$$\bar{e}_2 = \bar{m}_3 \times \bar{m}_1 = \left[m_y m_z - m_x m_c, m_a m_c - m_y^2, m_x m_y - m_z m_a \right]^T$$

$$\bar{e}_3 = \bar{m}_1 \times \bar{m}_2 = \left[m_x m_z - m_y m_b, m_x m_y - m_z m_a, m_a m_b - m_x^2 \right]^T$$

I color-coded some of the like terms for ease of visibility. The only difference between these 3 expressions is their magnitude. To determine which of the three terms has the largest magnitude, which will provide the most accurate answer, in the most efficient manor we first note these similar terms:

$$\bar{e}_k(i) = \bar{e}_i(k) \text{ for } i,k = 1,2,3$$

Then, noting that each of the 3 vectors are parallel, each of their components have to be proportional:

$$|\bar{e}_k(j)| = \zeta_{ik} |\bar{e}_i(j)| \text{ for } i,j,k = 1,2,3$$

where:

$$\zeta_{ik} = \pm \frac{\|\bar{e}_k(j)\|}{\|\bar{e}_i(j)\|}$$

Then, from the preceding two statements we can write the following:

$$|\bar{e}_k(k)| = \zeta_{ik} |\bar{e}_i(k)| = \zeta_{ik} |\bar{e}_k(i)| = \zeta_{ik}^2 |\bar{e}_i(i)|$$

This means that the farthest-from-zero and closes-to-zero elements are the non-colored terms, the unique terms, in the $\bar{e}$ expressions. Thus, we simply have to evaluate those 3 unique terms and then evaluate the remaining 2 terms for the $\bar{e}$ expression that has the farthest-from-zero, largest magnitude, unique term. Let's call this vector $\bar{e}_k$. Now, using an equation we used previously, from the 4th row of matrix [K]:

$$\cot\left(\frac{\phi}{2}\right) = \frac{\bar{Z}^T}{(\lambda_{max} - \sigma)} \bar{e}$$

We manipulate the sides to yield proportionalities:

$$\cot\left(\frac{\phi}{2}\right) = \frac{\cos\left(\frac{\phi}{2}\right)}{\sin\left(\frac{\phi}{2}\right)} = \frac{\bar{Z}^T}{(\lambda_{max} - \sigma)} \bar{e}$$

$$\cos\left(\frac{\phi}{2}\right) \propto \bar{Z}^T \bar{e}_k$$

$$\sin\left(\frac{\phi}{2}\right) \propto (\lambda_{max} - \sigma)$$

Using these in the definition of the quaternion:

$$\bar{q} = \begin{bmatrix} \bar{e} \sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \propto \begin{bmatrix} \bar{e}_k (\lambda_{max} - \sigma) \\ \bar{Z}^T \bar{e}_k \end{bmatrix}$$

The final step is to divide the vector by its' magnitude to make the quaternion a unit vector:

$$\bar{q} = \frac{1}{\|\square\|} \begin{bmatrix} \bar{e}_k (\lambda_{max} - \sigma) \\ \bar{Z}^T \bar{e}_k \end{bmatrix}$$

TRIAD Algorithm

Saturday, February 22, 2020 2:43 PM

The TRIAD Algorithm can only use 2 measurements but is extremely fast. It starts by determining an orthogonal set of axes in the Body Frame using the sensor measurements in the Body Frame:

$$\hat{x}^B = \hat{v}_1^B$$

$$\hat{y}^B = \left(\hat{v}_1^B \times \hat{v}_2^B\right) / \left\|\hat{v}_1^B \times \hat{v}_2^B\right\|$$

$$\hat{z}^B = \hat{x}^B \times \hat{y}^B$$

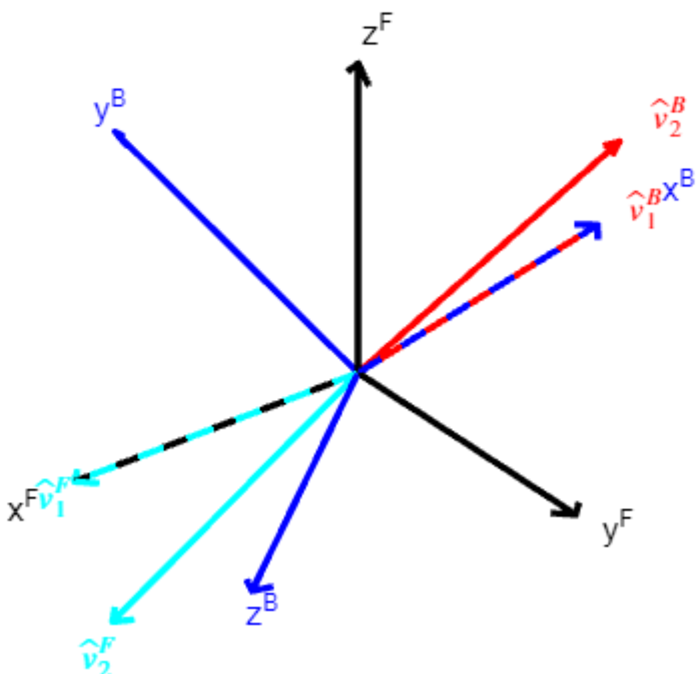
Then, we repeat this process with the mathematical models of these sensor measurement vectors in the Fixed Frame:

$$\hat{x}^F = \hat{v}_1^F$$

$$\hat{y}^F = \left(\hat{v}_1^F \times \hat{v}_2^F\right) / \left\|\hat{v}_1^F \times \hat{v}_2^F\right\|$$

$$\hat{z}^F = \hat{x}^F \times \hat{y}^F$$

Below is a visualization of what this may look like physically:



With the same vectors in both the Body Frame and in the Fixed Frame, we can relate them using a Direction Cosine Matrix, DCM, from the Fixed Frame to the Body Frame:

$$\begin{bmatrix} \hat{x}^B & \hat{y}^B & \hat{z}^B \end{bmatrix} = DCM_F^B \begin{bmatrix} \hat{x}^F & \hat{y}^F & \hat{z}^F \end{bmatrix}$$

Now, the 2 matrices made up of x, y, and z vectors are both orthonormal matrices since their columns are orthogonal and unit vectors. This means that the inverse is simply the transpose, thus solving for the DCM we get:

$$DCM_F^B = \begin{bmatrix} \hat{x}^B & \hat{y}^B & \hat{z}^B \end{bmatrix}\begin{bmatrix} \hat{x}^F & \hat{y}^F & \hat{z}^F \end{bmatrix}^T$$

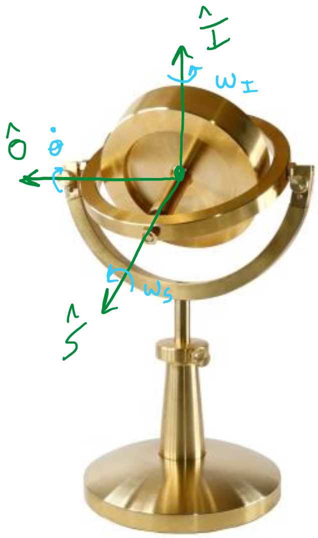
Once the Direction Cosine Matrix is determined we can use previously discussed formulas to calculate any of the other parameterizations.

Dynamic Attitude Determination

Saturday, February 22, 2020 5:15 PM

Because Static Attitude Sensor, e.g. sun sensors, have noise in their signal, if we try and take the derivative of this noisy signal, it will amplify the noise making the output not a good representation of the true derivative. For this reason, we must measure the angular velocity of the spacecraft directly, using gyroscopes, rather than trying to estimate it from a time-history of static attitude information.

There are many types of gyroscopes including MEMS gyroscopes found on microchips in your phone, ring laser gyroscopes, and fiber optic gyroscopes. All of these gyroscopes are based on different physics principles, however, they are all called gyroscopes since they measure angular velocity. Here we will discuss the classical, physical, gyroscope.



$\hat{S}$: Spin Axis. Angular velocity of the rotor, spinning disk, about the Spin Axis is ω_S, a constant.
$\hat{I}$: Input Axis. Angular velocity of the spacecraft about the Input Axis is ω_I
$\hat{O}$: Output (Gimbal) Axis. Angular velocity of the "gimbal" about the Output Axis is $\dot{\theta}$
These 3 orthogonal axes constitute the Gyroscope Frame, G.

Motion about the Output Axis is inhibited by viscous damping and a spring restraint, i.e.:

$$\bar{T}^G = -\left(D\dot{\theta} + K\theta\right)\hat{O}$$

Where

$$K\theta \gg D\dot{\theta}$$

Motion about the Input and Spin Axes are counteracted by reaction torques in the bearings of the gimbal.
Let's start our analysis using the sum of moments equation to determine the dynamics of the gyroscope.

$$\sum \bar{M} = \frac{d^G}{dt}\left(\bar{H}^G_{gyro}\right) + \bar{\omega}^G \times \bar{H}^G_{gyro} = \bar{T}^G$$

where the angular momentum of the gyroscope and the angular velocity of the Gyroscope Frame, G, relative to the Fixed Frame are:

$$\bar{H}^G_{gyro} = I_{rotor}\omega_s\hat{S} + I_O\dot{\theta}\hat{O} = H_{rotor}\hat{S} + I_O\dot{\theta}\hat{O}$$

$$\bar{\omega}^G = \omega_s\hat{S} + \omega_I\hat{I} + \dot{\theta}\hat{O}$$

Since the rotor spins at constant speed, the angular momentum of the rotor about the Spin Axis is constant, thus:

$$\frac{d^G}{dt}\left(\bar{H}^G_{gyro}\right) = I_O\ddot{\theta}\hat{O}$$

The cross-product term is:

$$\bar{\omega}^G \times \bar{H}^G_{gyro} = \begin{bmatrix} \hat{S} & \hat{I} & \hat{O} \\ \omega_s & \omega_I & \dot{\theta} \\ H_{rotor} & 0 & I_O\dot{\theta} \end{bmatrix} = \begin{bmatrix} \omega_I I_O\dot{\theta} \\ H_{rotor}\dot{\theta} - \omega_s I_O\dot{\theta} \\ -H_{rotor}\omega_I \end{bmatrix}$$

Now, we are only interested in the one axis of the gyroscope that can move, the Output Axis, thus we write the sum of moments for just that axis is:

$$I_O\ddot{\theta} - H_{rotor}\omega_I = -\left(D\dot{\theta} + K\theta\right)$$

$$I_O\ddot{\theta} + D\dot{\theta} + K\theta = H_{rotor}\omega_I$$

This is a linear, second order, ordinary differential equation with well-known solutions. If we look at the steady-state case, ignoring the short transient solutions, then:

$$\ddot{\theta}, \dot{\theta} = 0$$

and then we can solve for the angular velocity of the spacecraft about the Input Axis by measuring the angular deflection of the gimbal about the Output Axis:

$$K\theta = H_{rotor}\omega_I$$

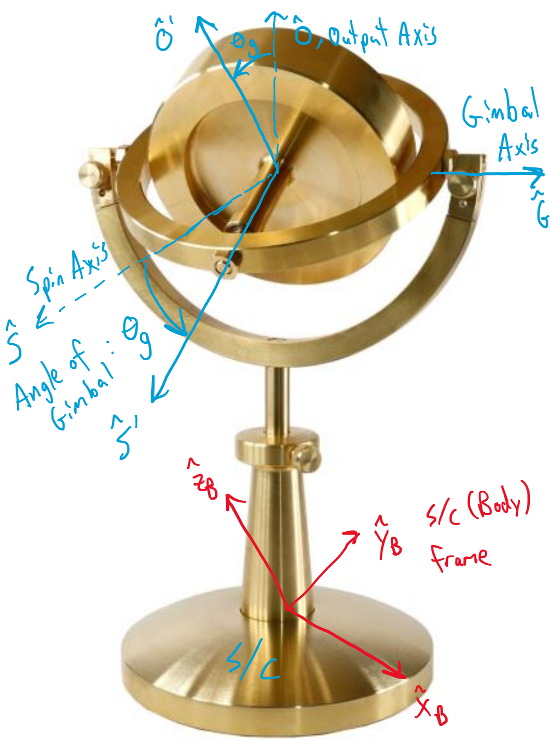
$$\omega_I = \frac{K\theta}{H_{rotor}}$$

With 3 orthogonal gyroscopes, we can measure the entire angular velocity vector of the spacecraft.

A Control Moment Gyro (CMG), is an actuator that uses the same principle but changes the angle of the gimbal to cause a change in the angular velocity of the spacecraft about the Input Axis.

Control Moment Gyro (CMG)

Monday, August 22, 2022 10:49 AM



Let's start in the body frame of the gyro, G', and write the sum of moments equation:

$$\sum \bar{M} = \frac{d^{G'}}{dt}(\bar{H}) + \bar{\varpi} \times \bar{H} = \bar{T}$$

where the angular momentum of the gyro is:

$$\bar{H} = I_G \omega_G \hat{G}' + I_S \omega_S \hat{S}'$$

since the gyro rotates about the Gimbal Axis, G', and the Spin Axis, S'.

$\bar{\varpi}$ is the angular velocity of the G' body frame relative to the G fixed frame. Thus, it is just the angular velocity of the gimbal frame.

The torque vector on the right-hand side has reaction torques about the Gimbal Axis and Spin Axis but not the Output Axis, O'.

Notice the derivative term has no Output Axis information so this will only come from the cross-product term:

$$\bar{\varpi} \times \bar{H} = \begin{bmatrix} \hat{G}' & \hat{O}' & \hat{S}' \\ \omega_G & 0 & 0 \\ I_G \omega_G & 0 & I_S \omega_S \end{bmatrix} = \begin{bmatrix} 0 \\ -I_S \omega_S \omega_G \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -H_S \omega_G \\ 0 \end{bmatrix}$$

where H_S is the angular momentum of the gyro's spin axis, a constant, and ω_G is the angular velocity of the gyro about the gimbal axis. So, if the CMG is rotated about the Gimbal Axis, it produces an apparent torque about the Output Axis. We now need to rotate this torque from the G' frame, to the G frame, and finally to the Body Frame of the spacecraft. Since, in a sense, this is going from a body frame of the CMG to a fixed frame of the spacecraft, the minus signs of the rotation matrix are flipped:

$$\left(\bar{\omega}\times\bar{H}\right)^{G}=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_{G} & -\sin\theta_{G}\\ 0 & \sin\theta_{G} & \cos\theta_{G}\end{bmatrix}\begin{bmatrix}0\\ -H_{S}\omega_{G}\\ 0\end{bmatrix}=-H_{S}\omega_{G}\begin{bmatrix}0\\ \cos\theta_{G}\\ \sin\theta_{G}\end{bmatrix}$$

$$\left(\bar{\omega}\times\bar{H}\right)^{B}=-H_{S}\omega_{G}DCM_{G}^{B}\begin{bmatrix}0\\ \cos\theta_{G}\\ \sin\theta_{G}\end{bmatrix}$$

Where the angular position of the CMG about the Gimbal Axis is θ_G, that changes, and DCM_G^B is the DCM from the CMG frame, G, to the Body Frame of the spacecraft, a constant. This cross-product term is now a part, the internal part, of the cross-product term for the dynamics of the spacecraft in the Body Frame:

$$\sum\bar{M}=\frac{d^{B}}{dt}(\bar{H})+\bar{\omega}^{B}\times\bar{H}=\bar{T}$$

Since we desire to write our term as a torque, we move it to the right-hand side which removes the minus sign in our cross-product term. We also note that a spacecraft must have a minimum of 3 CMGs to control a spacecraft in all 3 axes but almost always has more than 3 for redundancy and more efficiency. For *n* CMGs we have:

$$\bar{T}_{CMG}=\sum_{i=1}^{n}\underbrace{H_{S}DCM_{G_i}^{B}\begin{bmatrix}0\\ \cos\theta_{G_i}\\ \sin\theta_{G_i}\end{bmatrix}}_{\bar{C}_i}\omega_{G_i}=\sum_{i=1}^{n}\bar{C}_i\omega_{G_i}$$

This yields the sum of a vector that changes based on the angular position of the Gimbal Axis times the angular velocity of the respective Gimbal Axis. This can be written in matrix form as:

$$\bar{T}_{CMG}=\bar{C}_1\omega_{G_1}+\bar{C}_2\omega_{G_2}+\cdots+\bar{C}_n\omega_{G_n}=\begin{bmatrix}\bar{C}_1 & \bar{C}_2 & \cdots & \bar{C}_n\end{bmatrix}\begin{bmatrix}\omega_{G_1}\\ \omega_{G_2}\\ \vdots\\ \omega_{G_n}\end{bmatrix}$$

$$\bar{T}_{CMG}=\left[C\left(\bar{\theta}_{G}\right)\right]\bar{\omega}_{G}$$

Where the C matrix is changing and is size 3xn and $\bar{\omega}_G$ is a vector of all n CMG angular velocities about the Gimbal Axes.

Now we need to determine what the angular velocities of the CMG's need to be, the control output, to result in a given torque, the control input. Unfortunately, the matrix C is not square, unless you only have 3 CMGs, so it does not have an inverse. In this case, we have 3 knowns, the torque components, but n (greater than 3) unknowns, the angular velocities of the CMGs. Thus, we need to solve for the angular velocities of the CMGs under some additional constraint. Here we will do so to minimize the overall angular velocities, i.e.:

$$\min\quad\frac{1}{2}\bar{\omega}_{G}^{T}\bar{\omega}_{G}$$

To do this, while satisfying the torque constraint, we append the constraint to the minimizing cost function:

$$J=\frac{1}{2}\bar{\omega}_{G}^{T}\bar{\omega}_{G}+\bar{\lambda}^{T}\underbrace{\left(\bar{T}_{CMG}-\left[C\left(\bar{\theta}_{G}\right)\right]\bar{\omega}_{G}\right)}_{\bar{0}}$$

As you can see, we just added zero to the original cost function by using $\bar{\lambda}$, known as a Lagrange Multiplier in the math world. Now, to minimize J, we take the derivative with respect to the free variable, $\bar{\omega}_G$, and set that equal to zero:

$$\frac{\partial J}{\partial \bar{\varpi}} = 0 = \bar{\varpi}_G^T - \bar{\lambda}^T \left[C\left(\bar{\theta}_G\right) \right]$$

$$\bar{\varpi}_G^T = \bar{\lambda}^T \left[C\left(\bar{\theta}_G\right) \right]$$

$$\bar{\varpi}_G = \left[C\left(\bar{\theta}_G\right) \right]^T \bar{\lambda}$$

This is the expression for $\bar{\omega}_G$ but we still don't know what $\bar{\lambda}$ is. Let's plug this expression into the torque constraint equation.

$$\bar{T}_{CMG} = \left[C\left(\bar{\theta}_G\right) \right] \bar{\varpi}_G = \left(\left[C\left(\bar{\theta}_G\right) \right] \left[C\left(\bar{\theta}_G\right) \right]^T \right) \bar{\lambda}$$

The changing matrix in the parenthesis, CC^T, is size 3x3 and, more importantly, square so it has an inverse, as long as the determinate of this matrix isn't zero.

$$\bar{\lambda} = \left(\left[C\left(\bar{\theta}_G\right) \right] \left[C\left(\bar{\theta}_G\right) \right]^T \right)^{-1} \bar{T}_{CMG}$$

Now that we have $\bar{\lambda}$, plug this back into the equation for $\bar{\omega}_G$.

$$\bar{\varpi}_G = \left[C\left(\bar{\theta}_G\right) \right]^T \bar{\lambda} = \underbrace{\left[C\left(\bar{\theta}_G\right) \right]^T \left(\left[C\left(\bar{\theta}_G\right) \right] \left[C\left(\bar{\theta}_G\right) \right]^T \right)^{-1}}_{\text{pseudo-inverse}} \bar{T}_{CMG}$$

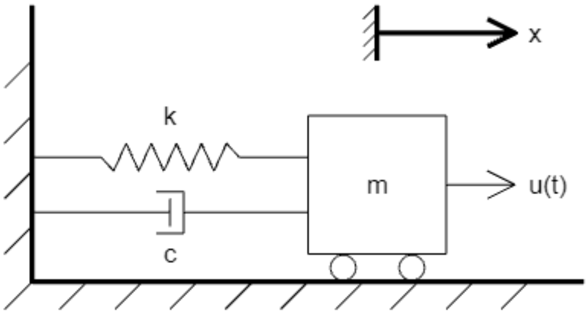
Given a desired torque, this equation will yield the angular velocities of all n CMGs to produce that torque!

As mentioned before, the determinate of the CC^T must not be zero, it must not be a singular matrix, it must don't have dependent rows/columns. If it does, you have what is called "gimbal lock", meaning the CMG's gimbal axes are in a configuration that doesn't allow you to produce any desired torque. For a given initial orientation of the CMG array, you can calculate the singularity space and design a more sophisticated control algorithm to avoid this space. This may involve path planning, designing a maneuver manually to avoid the singularities, or a more sophisticated algorithm than this "simple" pseudo-inverse. These are beyond the scope of this course but great topics for Master's and Ph.D. students.

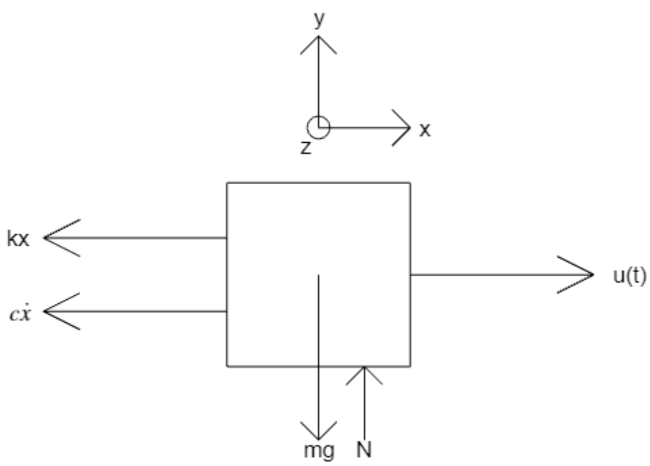
Controls Introduction

Sunday, January 12, 2020 4:32 PM

Basic controls is based on the classic mass-spring-damper system from Dynamics:



We start our analysis with a free-body diagram:



Now, sum forces in the x-direction:

$$\sum F_x = m\ddot{x} = -kx - c\dot{x} + u(t)$$

Re-arranging this equation leads to arguably the most used differential equation in history:

$$m\ddot{x} + c\dot{x} + kx = u(t)$$

Let's solve this linear, 2nd order, ODE for the homogeneous solution, i.e., when u(t)=0. To do this let's assume the exponential solution:

$$m\ddot{x} + c\dot{x} + kx = 0$$

$$x(t) = Ae^{rt}$$

$$\therefore \dot{x} = rAe^{rt}$$

$$\therefore \ddot{x} = r^2 Ae^{rt}$$

Plug these into the ODE and simplify:

$$\underbrace{Ae^{rt}}_{\neq 0}\left(r^2 + cr + k\right) = 0$$

The coefficient *A* cannot be zero because this would lead to the trivial solution where *x(t) = 0* forever. The exponential cannot be zero either; it can exponentially approach zero but it can never equal zero. Thus, the term in the parenthesis, known as the Characteristic Equation in your Differential Equations class, must be zero. This is a quadratic equation for this problem and thus we solve it with the Quadratic Formula:

$$r = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m}$$

The above equation can lead to 4 different unique solution depending on the values of the 3 positive constants:

1. If $c^2 - 4mk < 0$ then the term under the square-root is negative and r is complex with neagtive real. This means that x(t) is an exponentially decaying sinusoid of the form:
   $x(t) = e^{-\frac{c}{2m}t}(A \sin \omega t + B \cos \omega t)$ where $\omega = \frac{\sqrt{4mk - c^2}}{2m}$
   This solution is known as "under-damped".
2. If $c^2 - 4mk = 0$ then the square-root is zero and r has a double-root at $-\frac{c}{m}$. This means that x(t) is a pure exponentially decaying function of the form:
   $x(t) = (A + Bt)e^{-\frac{c}{m}t}$
   This solution is known as "critically damped"
3. If $c^2 - 4mk > 0$ then the square-root is positive but less than the leading term c in magnitude thus r has two, distinct, negative real, roots: r_1 and r_2. This means that x(t) is a pure exponentially decaying function like the critically damped case. However, because of the root using the plus sign, the exponential decay is slower and has the the form:
   $x(t) = Ae^{r_1 t} + Be^{r_2 t}$
   This solution is known as "over-damped"
4. If c = 0, then r simplifies to $r = \pm i\sqrt{\frac{k}{m}}$ thus x(t) must be a pure sinusoid of the form:
   $$x(t) = A \sin\left(\sqrt{\frac{k}{m}}t\right) + B \cos\left(\sqrt{\frac{k}{m}}t\right)$$
   This solution in known as "un-damped". The frequency that the mass oscillates at is known as the "natural frequency" or "un-damped natural frequency"
   $$\omega_n = \sqrt{\frac{k}{m}}$$

One of the goals of this analysis is to parameterize the solution, not in terms of the physical constants of the problem: m, c, and k (because we don't have a mass-spring-damper system, we have a spacecraft attitude system) but in terms of quantities that describe the motion of the object like the natural frequency. The other term we will define is based on the quantity under the square root: $c^2 - 4mk$. With some algebraic manipulation, for the under-damped case, we have:

$$c^2 - 4mk < 0$$

$$c^2 < 4mk$$

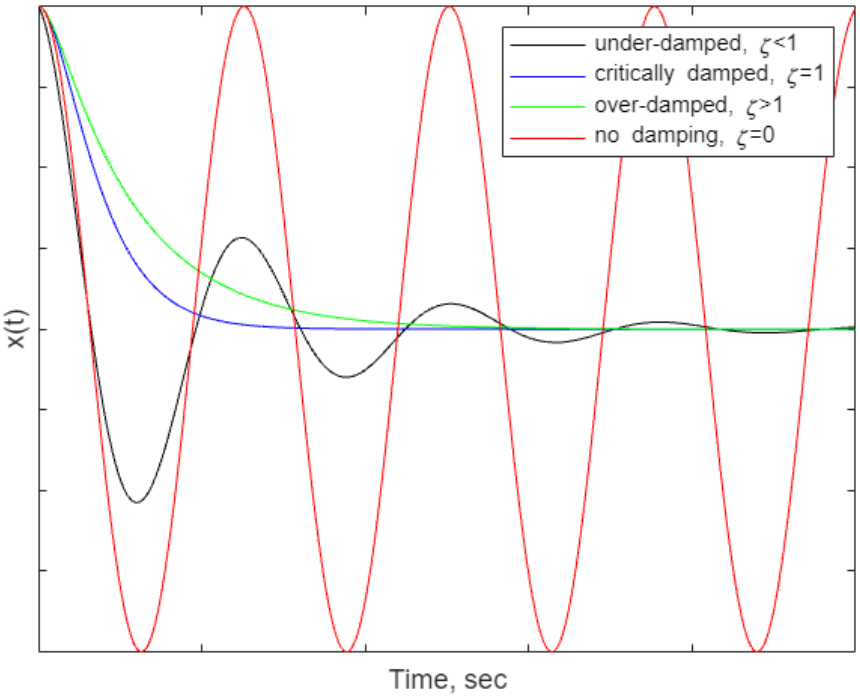
$$\frac{c^2}{4mk} < 1$$

$$\zeta = \frac{c}{2\sqrt{mk}} < 1$$

We call this quantity the "damping ratio" and use the Greek letter zeta as its symbol. Thus, for the 4 different cases we have:

	Under-Damped	Critically Damped	Over-Damped	No Damping
ζ	<1	=1	>1	0

Below we see an example of the 4 different cases of damping ratio:



Now, returning to the beginning and our mass-spring-damper system, we wish to remove the physical constants and replace them with the natural frequency and the damping ratio. Through some algebra we get:

$$m\ddot{x} + c\dot{x} + kx = u(t)$$

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{1}{m}u(t)$$

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \frac{1}{m}u(t)$$

This gives us a characteristic equation to the homogeneous solution of:

$$r^2 + 2\zeta\omega_n r + \omega_n^2 = 0$$

$$r = \frac{-2\zeta\omega_n \pm \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}}{2}$$

$$r = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

Since the most common of the four cases is the under-damped case, the solution is typically written in that form, when r is complex, by factoring out a $\sqrt{-1}$ out of the second term:

$$r = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2} = -\zeta\omega_n \pm i\omega_d$$

Where ω_d is known as the "damped natural frequency". This yields the following *x(t)* solution:

$$x(t) = e^{-\zeta\omega_n t}\left(A\sin\left(\omega_d t\right) + B\cos\left(\omega_d t\right)\right)$$

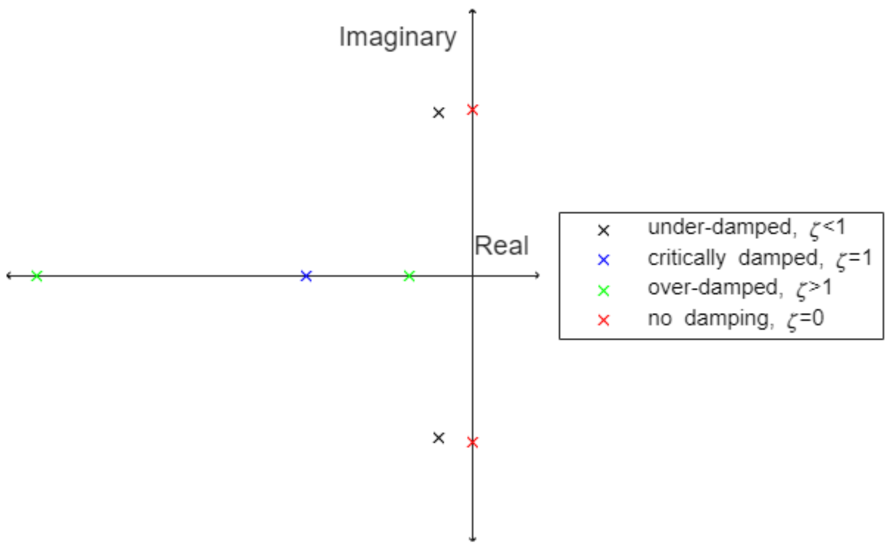
This means that using just algebra, since we have just done all of the differential equations, we can determine the response of the system:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0$$

Root Locus

Sunday, March 10, 2024 3:35 PM

A plot to help visualize the solution to the characteristic equation is called the Root Locus Plot. Below is a visualization of the same 4 solutions, depicted as time-history plots in the Controls Introduction, in a complex-plane Root Locus Plot:



In general, for the under-damped case, the solution to the characteristic equation is:

$$r = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2} = -\zeta\omega_n \pm i\omega_d$$

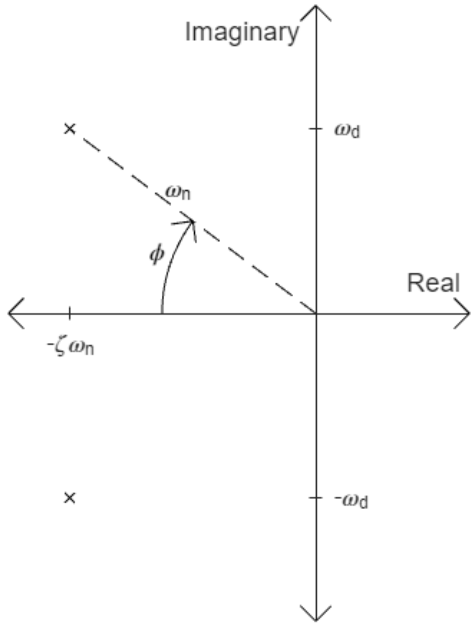
The magnitude of this complex number is simply:

$$\|r\| = \sqrt{(\zeta\omega_n)^2 + \left(\omega_n\sqrt{1-\zeta^2}\right)^2}$$

$$\|r\| = \omega_n\sqrt{\zeta^2 + \left(1-\zeta^2\right)}$$

$$\|r\| = \omega_n$$

Plotting and labeling the root locus for this general case we have:



The angle, ϕ, will be seen again in the next lecture on Step-Response so let's write a few useful equations:

$$\sin\phi = \frac{\omega_d}{\omega_n} = \sqrt{1-\zeta^2}$$

$$\cos\phi = \frac{\zeta\omega_n}{\omega_n} = \zeta$$

$$\tan\phi = \frac{\omega_d}{\zeta\omega_n} = \frac{\sqrt{1-\zeta^2}}{\zeta}$$

Step-Response

Friday, December 20, 2019 12:33 PM

We wish to use nothing but algebra to calculate our controller. Our controller must move our system, our spacecraft, to a desired state with a desired response. To this end, we need to mathematically define what is a "desired response". We start by extending the homogeneous solution analysis to a system with a step-response, i.e. *u(t)* = 1.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \frac{u(t)}{m} = \frac{1}{m}$$

First we solve for the particular, or non-homogeneous, solutions. Since the forcing function is constant, we assume that the particular solution is also constant. Thus the derivative terms, $\dot{x}, \ddot{x}$, are zero and solve for the particular solution, x_p:

$$\omega_n^2 x_p = \frac{1}{m}$$

$$x_p = \frac{1}{m\omega_n^2} = \frac{1}{m\left(\sqrt{\frac{k}{m}}\right)^2} = \frac{1}{k}$$

The homogeneous solution is exactly the same as we derived in the Controls Introduction. Here we assume the under-damped solution:

$$x_h(t) = e^{-\zeta\omega_n t}\left(A\sin\left(\omega_d t\right) + B\cos\left(\omega_d t\right)\right)$$

Adding the homogeneous and particular solutions together gives us the total solution:

$$x(t) = \frac{1}{k} + e^{-\zeta\omega_n t}\left(A\sin\left(\omega_d t\right) + B\cos\left(\omega_d t\right)\right)$$

where the coefficients A and B are used to satisfy boundary conditions. For this analysis we assume the initial position and velocity are both zero: $x(0) = 0, \dot{x}(0) = 0$. Then, using position, we solve for coefficient *B*:

$$x(0) = 0 = \frac{1}{k} + B, \qquad \therefore B = -\frac{1}{k}$$

Next, we take the derivative of the *x(t)* expression resulting in the velocity expression:

$$\dot{x}(t) = -\zeta\omega_n e^{-\zeta\omega_n t}\left(A\sin\left(\omega_d t\right) + B\cos\left(\omega_d t\right)\right) + \omega_d e^{-\zeta\omega_n t}\left(A\cos\left(\omega_d t\right) - B\sin\left(\omega_d t\right)\right)$$

Now, utilizing the initial velocity boundary condition, and the definition of $\omega_d = \omega_n\sqrt{1-\zeta^2}$ we solve for coefficient *A*:

$$\dot{x}(0) = 0 == -\zeta\omega_n B + \omega_d A$$

$$A = \frac{\zeta\omega_n B}{\omega_d} = \frac{\zeta\omega_n\left(-\frac{1}{k}\right)}{\omega_n\sqrt{1-\zeta^2}} = -\frac{1}{k}\frac{\zeta}{\sqrt{1-\zeta^2}}$$

The second fraction of A is simply the cotangent of the ϕ angle introduced in the Root Locus lecture, thus:

$$A = -\frac{\cot\phi}{k}$$

Now, these coefficients can be plugged back into the general *x(t)* equation to yield:

$$x(t)=\frac{1}{k}+e^{-\zeta\omega_n t}\left(A\sin(\omega_d t)+B\cos(\omega_d t)\right)$$

$$x(t)=\frac{1}{k}+e^{-\zeta\omega_n t}\left(\left(-\frac{\cot\phi}{k}\right)\sin(\omega_d t)+\left(-\frac{1}{k}\right)\cos(\omega_d t)\right)$$

$$x(t)=\frac{1}{k}\left(1-e^{-\zeta\omega_n t}\left(\sin(\omega_d t)\cot\phi+\cos(\omega_d t)\right)\right)$$

Now, using some trigonometric manipulation and the expression $\sin\phi=\sqrt{1-\zeta^2}$ from the Root Locus lecture:

$$x(t)=\frac{1}{k}\left(1-\frac{e^{-\zeta\omega_n t}}{\sin\phi}\underbrace{\left(\sin(\omega_d t)\cos\phi+\cos(\omega_d t)\sin\phi\right)}_{\sin(\omega_d t+\phi)}\right)$$

$$x(t)=\frac{1}{k}\left(1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin(\omega_d t+\phi)\right)\text{where}\tan\phi=\frac{\sqrt{1-\zeta^2}}{\zeta}$$

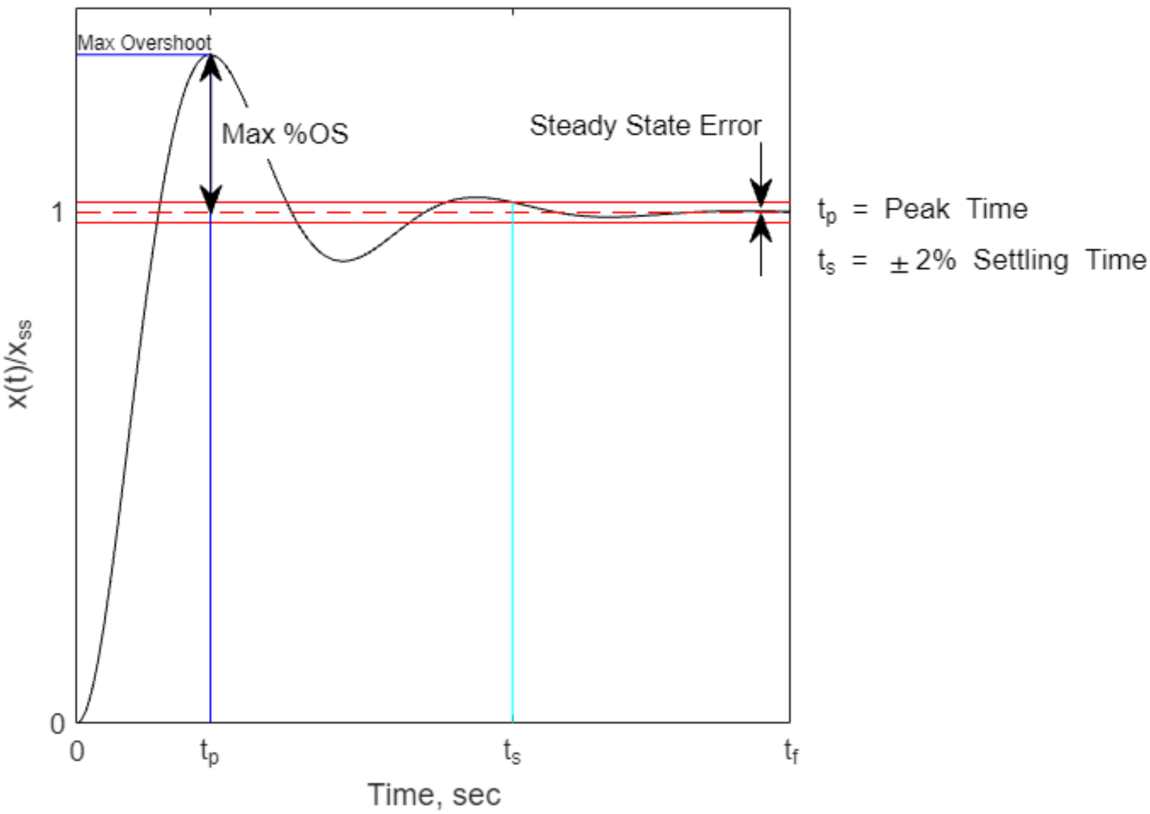
To remove the dependency of this equation to last remaining physical constant, *k*, we note that the steady-state solution is the limit of *x(t)* as *t* approaches infinity:

$$x_{ss}=\lim_{t\to\infty}\frac{1}{k}\left(1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin(\omega_d t+\phi)\right)=\frac{1}{k}$$

Then, by dividing the total *x(t)* equation by the steady-state solution, x_{ss}, we get:

$$\frac{x(t)}{x_{ss}}=1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin(\omega_d t+\phi)$$

Now, let's plot this function and point out some important characteristics of it:



In this class, the desired response will be defined by 3 quantities: maximum percentage overshoot (%OS), $\pm 2\%$ settling time (t_s), and steady state error. Now we can define these terms mathematically.

Max Percent Overshoot (*%OS*)

At max percent overshoot, the slope of x(t) is zero, thus $\dot{x} = 0$. First we have to derive the velocity equation by taking the derivative of the position:

$$\frac{\dot{x}(t)}{x_{ss}} = \zeta\omega_n \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_d t + \phi) - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \overbrace{\omega_d}^{\omega_n\sqrt{1-\zeta^2}} \cos(\omega_d t + \phi)$$

$$\frac{\dot{x}(t)}{x_{ss}} = \frac{\omega_n e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \left(\underbrace{\underbrace{\zeta}_{\cos\phi} \sin(\omega_d t + \phi) - \underbrace{\sqrt{1-\zeta^2}}_{\sin\phi} \cos(\omega_d t + \phi)}_{\sin(\omega_d t + \phi - \phi) = \sin(\omega_d t)} \right)$$

$$\frac{\dot{x}(t)}{x_{ss}} = \frac{\omega_n e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_d t)$$

Since the natural frequency is not zero for an under-damped solution and the exponential function can never be zero, then it is the sine term that is zero at the max percent overshoot:

$$\sin(\omega_d t) = 0$$
$$\omega_d t = n\pi$$
$$t = \frac{n\pi}{\omega_d}, \quad n = 0,1,2,3,\ldots = \mathbb{Z}^+$$
This is the time of every peak and valley. To get the first peak, at max percent overshoot, we let n = 1:
$$t_p = \frac{\pi}{\omega_d}$$
Plug this into the position equation to find the maximum overshoot:
$$\frac{x(t_p)}{x_{ss}} = 1 - \frac{e^{-\zeta\omega_n\left(\frac{\pi}{\omega_d}\right)}}{\sqrt{1-\zeta^2}} \sin\left(\omega_d\left(\frac{\pi}{\omega_d}\right)+\phi\right)$$
$$\frac{x(t_p)}{x_{ss}} = 1 - \frac{e^{-\zeta\omega_n\left(\frac{\pi}{\omega_n\sqrt{1-\zeta^2}}\right)}}{\sqrt{1-\zeta^2}} \underbrace{\sin(\pi+\phi)}_{-\sin\phi=-\sqrt{1-\zeta^2}}$$
$$\frac{x(t_p)}{x_{ss}} = 1 + e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} = 1 + e^{-\frac{\pi}{\tan\phi}}$$
To make this a percentage, we compare this value to the steady state value:
$$\%OS = \frac{x(t_p) - x_{ss}}{x_{ss}} = e^{-\frac{\pi}{\tan\phi}} = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$
We notice that this equation is only a function of the damping ratio, not the natural frequency

$\pm$2% Settling Time (t_s)

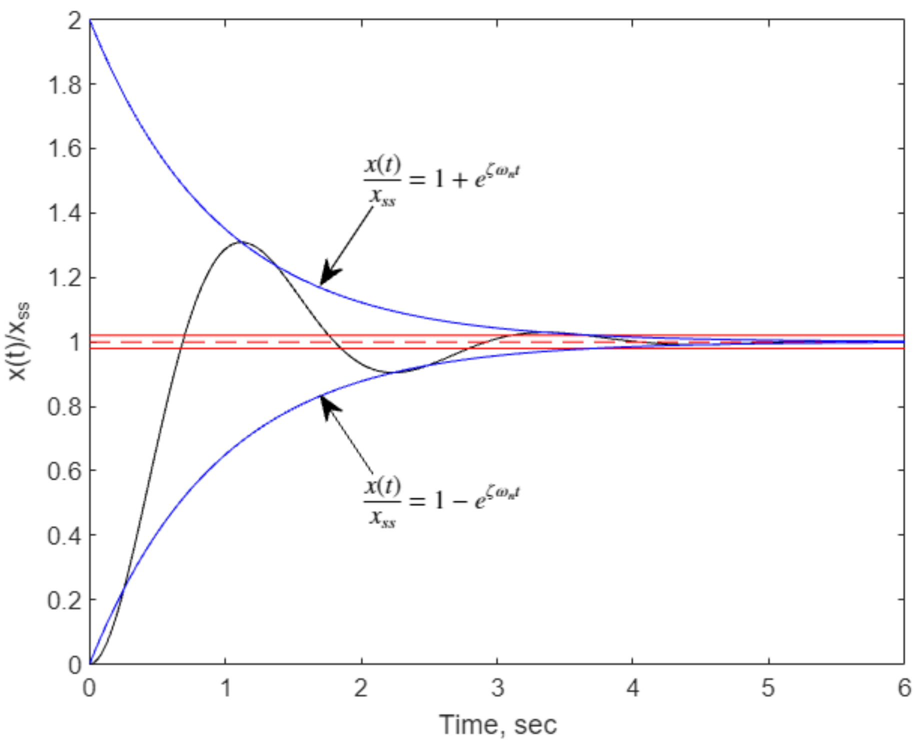
Settling Time is defined as the last time at which the position is more than 2% away from the steady-state solution. In order to guarantee that the signal will never again be more than 2% away from steady-state, we look for the first peak/valley that is within this limit. The times of the peaks and valleys were found in the previous section to be:
$$t = \frac{n\pi}{\omega_d}, \quad n = \mathbb{Z}^+$$
Now, we plug this into the position equation, but only in the sine term and then simplify:

$$\frac{x(t)}{x_{ss}} = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin\left(\omega_d \frac{n\pi}{\omega_d} + \phi\right)$$

$$\frac{x(t)}{x_{ss}} = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \underbrace{\sin(n\pi + \phi)}_{\pm\sin\phi = \pm\sqrt{1-\zeta^2}}$$

$$\frac{x(t)}{x_{ss}} = 1 \pm e^{-\zeta\omega_n t}$$

These two exponential curves go through all of the peaks and valleys of the original position curve as illustrated below:



Of importance here is, note how the blue exponentials go through the peaks/valleys but are not completely outside of the original x(t) function everywhere. Specifically, immediately following a peak/valley the blue exponential is inside of the x(t) function.

Now we make this a percentage by comparing this value to the steady-state value:

$$\frac{x(t) - x_{ss}}{x_{ss}} = \pm e^{-\zeta\omega_n t}$$

This expression has to be within $\pm 2\%$ of steady state. Now that we are comparing this expression to $\pm 2\%$, the time is now the Settling Time, t_s:

$$\pm e^{-\zeta\omega_n t_s} < \pm 2\% = \pm 0.02$$

$$-2\% = -0.02 < -e^{-\zeta\omega_n t_s}, e^{-\zeta\omega_n t_s} < 0.02 = 2\%$$

$$e^{-\zeta\omega_n t_s} < 0.02$$

$$\ln\left(e^{-\zeta\omega_n t_s}\right) = -\zeta\omega_n t_s < \ln(0.02) = -3.912\ldots$$

$$\mathrm{t}_s > \frac{3.912\ldots}{\zeta\omega_n}$$

Since the true x(t) position function can be outside of the blue exponential envelope, and because 3.912… is not a rational number, we simply round up to 4:

$$t_s = \frac{4}{\zeta\omega_n}$$

The ratio of $\frac{1}{\zeta\omega_n}$ is often refered to as the Time Constant, τ, thus the settling time can also be said to be 4τ.

PID Control

Thursday, March 14, 2024

PID stands for Proportional-Integral-Derivative. Let's tackle each one of these independently first.

Proportional Control:

Proportional control means that the control, u(t), is proportional to the position state, x. This leads to the following ODE:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \frac{u(t)}{m} = \frac{-k_p x}{m}$$

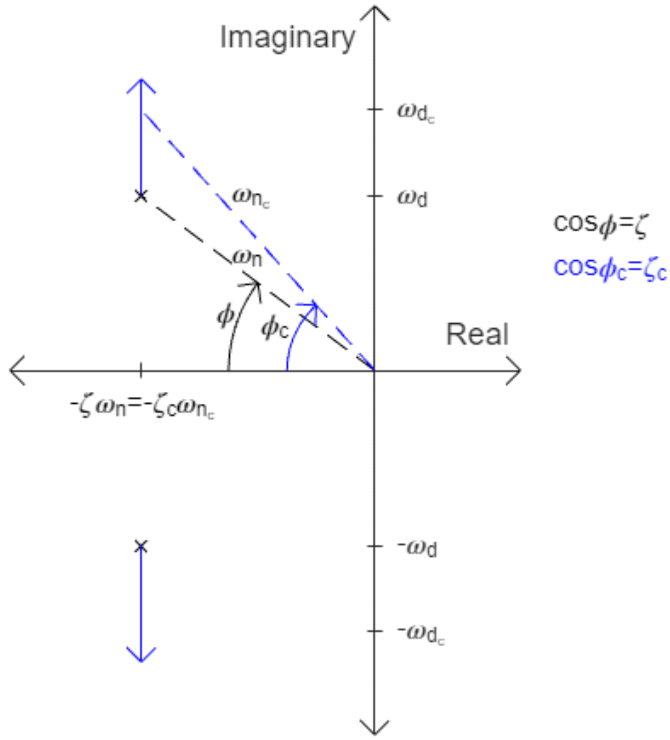
where *kp* is the proportionality constant known as the proportional gain. Now we can collect like terms on the left-hand side:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \left(\omega_n^2 + \frac{k_p}{m}\right)x = 0$$

This ODE has the same form as one we studied and solved in the Controls Introduction. This means that we can change the response of the system by simply changing the value of the proportional gain, k_p. Specifically we have:

$$\ddot{x} + \underbrace{2\zeta\omega_n}_{2\zeta_c\omega_{n_c}}\dot{x} + \underbrace{\left(\omega_n^2 + \frac{k_p}{m}\right)}_{\omega_{n_c}^2}x = 0$$

where ζ_c and ω_{n_c} are the damping ratio and natural frequency of the controlled system, *c*, respectively. So how do the values of the uncontrolled system compare to the values of the controlled system for changing values of the gain, k_p? First, assuming that kp is positive and increasing, looking at the position term, we see that the quantity that is ω_{n_c} must be larger than ω_n thus the natural frequency is increasing. Looking at the derivative term, we see that the product of $\zeta\omega_n$ stays the same, which also means that the settling time stays the same since the settling time is inversely proportional to $\zeta\omega_n$. So, since $\zeta\omega_n$ stays the same and ω_n is increasing, then ζ must be decreasing to keep the product the same, i.e., $\zeta_c < \zeta$. Lastly, since the damping ratio is decreasing, we will have more maximum percent overshoot, *%OS*. We can visualize these changes on a Root Locus plot:



Derivative Control:

Derivative control means that the control, u(t), is proportional to the derivative state, $\dot{x}$. This leads to the following ODE:

$$\ddot{x}+2\zeta\omega_n\dot{x}+\omega_n^2 x=\frac{u(t)}{m}=\frac{-k_d\dot{x}}{m}$$

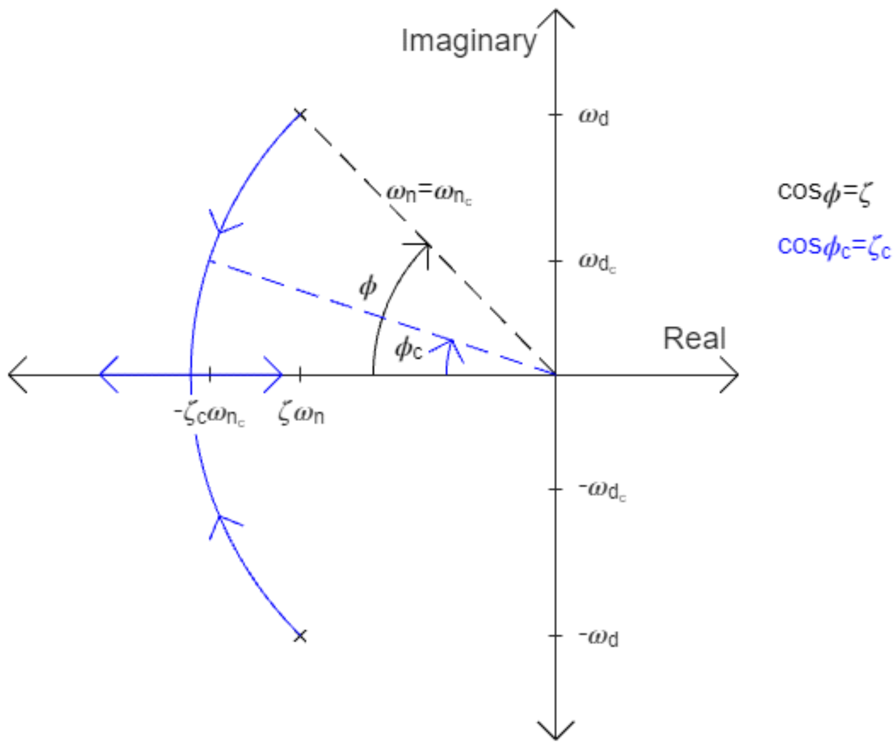
where k_d is the proportionality constant known as the derivative gain. Now we can collect like terms on the left-hand side:

$$\ddot{x}+\left(2\zeta\omega_n+\frac{k_d}{m}\right)\dot{x}+\omega_n^2 x=0$$

This ODE has the same form as one we studied and solved in the Controls Introduction. This means that we can change the response of the system by simply changing the value of the proportional gain, k_d. Specifically we have:

$$\ddot{x}+\underbrace{\left(2\zeta\omega_n+\frac{k_d}{m}\right)}_{2\zeta_c\omega_{n_c}}\dot{x}+\underbrace{\omega_n^2}_{\omega_{n_c}^2} x=0$$

where ζ_c and ω_{n_c} are the damping ratio and natural frequency of the controlled system, *c*, respectively. So how do the values of the uncontrolled system compare to the values of the controlled system for changing values of the gain, k_d? First, assuming that k_d is positive and increasing, looking at the position term, we see that the quantity that is ω_{n_c} must be the same as ω_n. Looking at the derivative term, we see that the product of $\zeta\omega_n$ must be increasing, which also means that the settling time is decreasing since the settling time is inversely proportional to $\zeta\omega_n$. So, since $\zeta\omega_n$ is increasing and ω_n is constants, then ζ must be increasing so that the product increases. Lastly, since the damping ratio is increasing, we will have less maximum percent overshoot, *%OS*. We can visualize these changes on a Root Locus plot:



Notice that if the derivative gain is too large, they system can transition from under-damped, to critically damped (a double root of the real axis), to over-damped (two distinct negative real roots).

Proportional-Derivative (PD) Control:

By utilizing both proportional and derivative control we can move the solution anywhere we need it to be to get the desired response, e.g., the correct damping ratio/maximum percentage overshoot and settling time/$\zeta\omega_n$ product. This is demonstrated in the PD Control Example lecture.

This raises the question, if PD control results in the desired response then what do we need integral control for? Well, the real-world is not an ideal/perfect world; the real-world has disturbances that are not taken into the perfect, linear, ODEs that we developed this analysis about. Let's discuss this in the context of our spacecraft attitude problem. Our linearized system of coupled ODEs from the Gravity-Gradient Torque Model lecture are:

$$\begin{bmatrix} \ddot{\theta}_3 + (k_1 - 1)n\dot{\theta}_1 + 4k_1 n^2 \theta_3 \\ \ddot{\theta}_2 + 3k_2 n^2 \theta_2 \\ \ddot{\theta}_1 + (1 - k_3)n\dot{\theta}_3 + k_3 n^2 \theta_1 \end{bmatrix} = \bar{0}$$

For a simplified analysis, we note that the mean motion, *n*, is order 1e-3 for low Earth orbits and thus any term that has the mean motion in it should be several orders of magnitude less than the control torque on the spacecraft. Ignoring these terms and including a torque on the right-hand side we have:

$$\begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_2 \\ \ddot{\theta}_1 \end{bmatrix} = \begin{bmatrix} T_1 / I_1 \\ T_2 / I_2 \\ T_3 / I_3 \end{bmatrix}$$

Recall that the third Euler Angle is about the first, x, axis and thus a Roll angle. The second Euler Angle is about the second, y, axis and thus is a Pitch angle. And the first Euler Angle is about the third, z axis and thus a Yaw angle. If the torque includes a control torque and disturbance torque term then we can write the simplified ODEs as:

$$\begin{bmatrix} \ddot{R} \\ \ddot{P} \\ \ddot{Y} \end{bmatrix} = \begin{bmatrix} \left(T_1 + T_{dist_1}\right) / I_1 \\ \left(T_2 + T_{dist_2}\right) / I_2 \\ \left(T_3 + T_{dist_3}\right) / I_3 \end{bmatrix}$$

Now let's look at a single axis. Because the Pitch, y, axis was decoupled from the Roll and Yaw axes in the linearized ODEs we started with, let's start with the Pitch axis. Let the control torque be a PD controller (currently designed to get to a Pitch of zero):

$$\ddot{P} = \left(T_2 + T_{dist_2}\right) / I_2 = \frac{1}{I_2}\left(-k_p P - k_d \dot{P}\right) + \frac{T_{dist_2}}{I_2}$$

Next, we collect position and velocity terms on the eft-hand side:

$$\ddot{P} + \frac{k_d}{I_2}\dot{P} + \frac{k_p}{I_2}P = \frac{T_{dist_2}}{I_2}$$

Under steady-state conditions, when $\ddot{P}, \dot{P} = 0$ we can solve for the final Pitch angle:

$$\frac{k_p}{I_2}P = \frac{T_{dist_2}}{I_2}$$

$$P = \frac{T_{dist_2}}{k_p}$$

This means that the steady-state error in our solution is directly proportional to the disturbance torque and inversely proportional to the proportional gain. We can increase the proportional gain to decrease the steady-state error but not without changing the response, e.g., maximum percentage overshoot and settling time, of the system. Thus, we need another type of control to eliminate any disturbances in the system. This is Integral Control.

Integral Control:

Integral Control is needed to drive steady-state error to zero in the presence of a disturbance. In general, the control term is:

$$u(t) = -k_i \int_0^t x_{error} dt = -k_i \int_0^t x - x_{target} dt$$

Let us look at a system that includes a disturbance torque with integral control:

$$\ddot{x} + 2\zeta\omega_n \dot{x} + \omega_n^2 x_{error} = \frac{u(t)}{m} = \frac{1}{m}\left(-k_i \int_0^t x_{error} dt + F_{dist}\right)$$

Since the integral term is typically order 1, especially for our spacecraft attitude problem, then the integral gain, k_i, should be on the same order of magnitude as the disturbance torque. The inclusion of the integral gain, while counteracting the disturbance torque, does change the equation to a 3rd order differential equation:

$$y = \int_0^t x_{error} dt$$

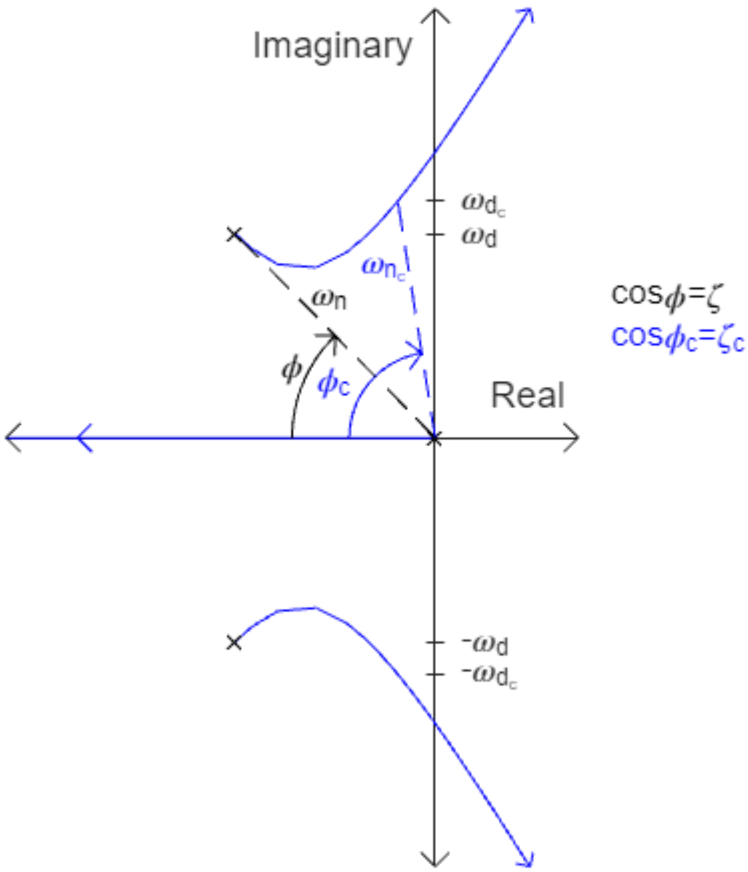
$$\dot{y} = x_{error} = x - x_{target}$$

$$\ddot{y} = \dot{x}$$

$$\dddot{y} = \ddot{x}$$

$$\dddot{y} + 2\zeta\omega_n \ddot{y} + \omega_n^2 \dot{y} + \frac{k_i}{m} y = \frac{F_{dist}}{m}$$

This means that the characteristic equation now has 3 roots as depicted in the following Root Locus plot:



This also shows that if the integral gain is too high, that the roots can reach the positive real axis. This leads to an x(t) solution that is exponentially increasing, instead of decreasing, an undesirable result. This also shows that the integral gain does change the damping ratio and natural frequency of the system, although, if the disturbance is small, then this effect will also be small.

Proportional-Integral-Derivative (PID) Control)

With all three types of controllers, we can control the overshoot and settling time of the response while achieving zero steady-state error in the presence of a disturbance.

PD Control Example

Friday, March 13, 2020 9:53 PM

The 1D spacecraft attitude problem can be approximated by:

$$I\ddot{\theta} = T$$

where torque is a PID controller. Here we will only use a PD controller with no disturbance torque:

$$T = -k_p\theta - k_d\dot{\theta}$$

In this example, we will start at a non-zero initial position and go to zero position and velocity. We will look at going to a specific, non-zero, position and velocity later in the course. Thus, the ODE can be written as:

$$\ddot{\theta} + \frac{k_d}{I}\dot{\theta} + \frac{k_p}{I}\theta = 0$$

This is of the form:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0$$

When we equate terms we have:

$$\frac{k_d}{I} = 2\zeta\omega_n, \quad \frac{k_p}{I} = \omega_n^2$$

Let's say the desired response is a settling time of 2 seconds and 5% maximum overshoot for a spacecraft with mass moment of inertia of 3 kg·m^2. This corresponds to the following two equations:

$$t_s = \frac{4}{\zeta\omega_n}, \quad \%OS = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

Solving the settling time equation we have:

$$\zeta\omega_n = \frac{4}{t_s} = \frac{4}{2} = 2$$

This product will be used later in the Root Locus plot and here by plugging this directly into the derivative gain equation above:

$$\frac{k_d}{I} = 2\zeta\omega_n = 2(2) = 4$$

$$k_d = 4I = 12$$

Next, we can re-arrange the *%OS* equation to solve for the damping ratio:

$$\%OS = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\ln\%OS = -\frac{\zeta\pi}{\sqrt{1-\zeta^2}}$$

$$\sqrt{1-\zeta^2}\ln\%OS = -\zeta\pi$$

$$\left(1-\zeta^2\right)\ln^2\%OS = \zeta^2\pi^2$$

$$\zeta^2\left(\pi^2+\ln^2\%OS\right) = \ln^2\%OS$$

$$\zeta = \sqrt{\frac{\ln^2\%OS}{\pi^2+\ln^2\%OS}}$$

This equation is not on the equation sheet explicitly, but, can be obtained using the above algebraic steps. Please do NOT memorize, or attempt to memorize, this equation. Plugging in 5% or 0.05:

$$\zeta = \sqrt{\frac{\ln^2 0.05}{\pi^2+\ln^2 0.05}} \approx 0.6901$$

Substitute this number into the settling time equation to solve for the natural frequency:

$$\zeta\omega_n = 2 \rightarrow \omega_n = \frac{2}{\zeta} \approx \frac{2}{0.6901} \approx 2.90\,\text{rad/s}$$

The natural frequency is used to determine the proportional gain:

$$\frac{k_p}{I} = \omega_n^2 \approx 2.90^2$$

$$k_p \approx 25.2$$

The corresponding Root Locus plot can be drawn. Solving for the damped natural frequency:

$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

$$\omega_d \approx 2.90\sqrt{1-0.6901^2} \approx 2.10\,\text{rad/s}$$

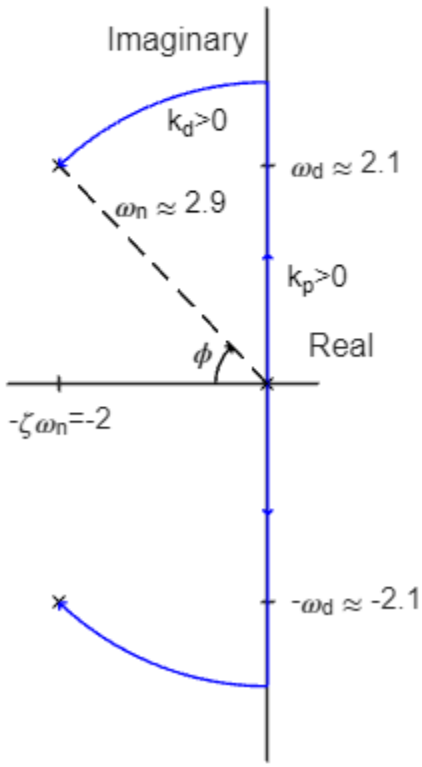
Recall, the general solution that is plotted on the Root Locus is:

$$r = -\zeta\omega_n \pm i\omega_d$$

Thus, for this example we are at the solution:

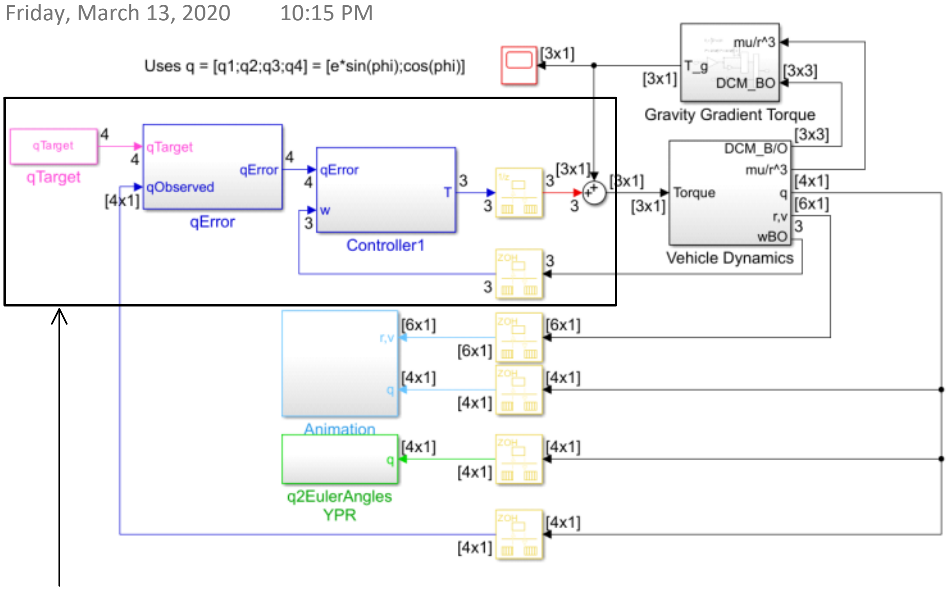
$$r = -2 \pm i2.10$$

If the gains were both zero, then the solution would have zero damping and zero natural frequency, hence, the solution begins at the origin of the root locus.


Imaginary
$k_d > 0$
$\omega_d \approx 2.1$
$\omega_n \approx 2.9$
$k_p > 0$
ϕ
Real
$-\zeta\omega_n = -2$
$-\omega_d \approx -2.1$

SIMULINK 6 Overview

Friday, March 13, 2020 10:15 PM



New for SIMULINK 6, add a PID controller to the SIMULINK 5 model.

First, if we have a target attitude in terms of the quaternion, $\bar{q}_{target}$, and an observed quaternion from our attitude sensors (in this model we assume perfect measurements so the actual attitude in terms of the quaternion coming from our Vehicle Dynamics block is our observed attitude), $\bar{q}_{observed}$, we need to determine the error in our attitude in terms of a quaternion, $\bar{q}_{error}$. This can be visualized first in terms of angles:

$$\theta_{target} + \theta_{error} = \theta_{observed}$$

$$\theta_{error} = \theta_{observed} - \theta_{target}$$

Recall that in quaternion space addition and subtraction do not exist thus we have to write this as quaternion multiplication where a subtraction is a negative, or inverse, rotation:

$$\bar{q}_{error} = \bar{q}_{target}^{-1} \otimes \bar{q}_{observed}$$

An inverse quaternion is simply a rotation of $-\phi$ about the Euler Axis $\bar{e}$:

$$\bar{q}^{-1} = \begin{bmatrix} \bar{e}\sin\left(\frac{-\phi}{2}\right) \\ \cos\left(\frac{-\phi}{2}\right) \end{bmatrix} = \begin{bmatrix} -\bar{e}\sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} = \begin{bmatrix} -\hat{q} \\ q_4 \end{bmatrix}$$

The Controller block must output a 3x1 torque vector based on the integral, position, and velocity (derivative) states:

$$\bar{T} = -[K]\bar{x} = -[K]\begin{bmatrix} \int_0^t \bar{\theta}_{error}\,dt \\ \bar{\theta}_{error} \\ \dot{\bar{\theta}} \end{bmatrix}$$

Notice that these states are in terms of the Euler Angles, not the quaternion. We can convert the quaternion to the Euler Angles but it requires nonlinear equations with 3 inverse trig functions; we would prefer to do something faster for real-time control. Thus, let's use small angle, and small angular velocity, approximations like we did in the Gravity-Gradient Torque Model:

$$DCM_O^B = R_{321} = R_1(\theta_3)R_2(\theta_2)R_3(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM_O^B \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \theta_3 \\ 0 & -\theta_3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\theta_2 \\ 0 & 1 & 0 \\ \theta_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \theta_1 & 0 \\ -\theta_1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\theta_2 \\ \cancelto{0}{\theta_2\theta_3} & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix} \begin{bmatrix} 1 & \theta_1 & 0 \\ -\theta_1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM_O^B \approx \begin{bmatrix} 1 & \theta_1 & -\theta_2 \\ \theta_1 & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & Y & -P \\ Y & 1 & R \\ P & -R & 1 \end{bmatrix}$$

Now we find the DCM in terms of the quaternion, also using small angle approximation:

$$\bar{q} = \begin{bmatrix} \bar{e}\sin\left(\dfrac{\phi}{2}\right) \\ \cos\left(\dfrac{\phi}{2}\right) \end{bmatrix} \approx \begin{bmatrix} \bar{e}\left(\dfrac{\phi}{2}\right) \\ 1 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{bmatrix}, \text{where } q_i q_j = 0 \text{ for } i, j = 1,2,3$$

Then:

$$DCM(\bar{q}) = \left(\cancelto{1}{q_4^2} - \cancelto{0}{\hat{q}^T\hat{q}}\right) I_{3x3} + 2\,\cancelto{0}{\hat{q}\hat{q}^T} + 2\,\cancelto{1}{q_4} \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \approx \begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix}$$

Equating these two DCM's we get:

$$2\hat{q} \approx \begin{bmatrix} R \\ P \\ Y \end{bmatrix}$$

This equation will be a better approximation as the error gets closer to zero.
For the angular velocity, also from the Gravity-Gradient Torque model we had:

$$\bar{\omega}_{B/O}^B \approx \begin{bmatrix} 1 & 0 & -\theta_2 \\ 0 & 1 & \theta_3 \\ \theta_2 & -\theta_3 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \theta_3 \\ 0 & -\theta_3 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\theta}_3 \\ 0 \\ 0 \end{bmatrix}$$

$$\bar{\omega}_{B/O}^B \approx \begin{bmatrix} -\theta_2 & 0 & 1 \\ \theta_3 & 1 & 0 \\ 1 & -\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

$$\theta_i \dot{\theta}_j \approx 0 \quad \text{for } i, j = 1,2,3$$

$$\bar{\omega}_{B/O}^B \approx \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_2 \\ \dot{\theta}_1 \end{bmatrix}$$

Thus we have:

$$\bar{\omega}_{B/O}^{B} \approx \begin{bmatrix} \dot{R} \\ \dot{P} \\ \dot{Y} \end{bmatrix}$$

Putting this together for the torque, we have:

$$\bar{T} = -[K]\bar{x} = -[K]\begin{bmatrix} \int_0^t \bar{\theta}_{error} dt \\ \bar{\theta}_{error} \\ \dot{\bar{\theta}} \end{bmatrix} = -[K]\begin{bmatrix} \int_0^t 2\hat{q}_{error} dt \\ 2\hat{q}_{error} \\ \bar{\omega}_{B/O}^{B} \end{bmatrix}$$

Since the torque vector is 3x1 and the x vector is size 9x1, the gain matrix [K] must be 3x9, totaling 27 gains. To simplify the control problem, we will assume that Roll torque is only based on Roll information, Pitch torque on Pitch information, and Yaw torque on Yaw information. Also, since the first 3 rows of x are the integral states, the first 3 columns of [K] must be integral gains. Rows 4-6 of x are position states so columns 4-6 of [K] are proportional states and rows 7-9 of x are derivative states so columns 7-9 of [K] are derivative states:

$$[K] = \begin{bmatrix} K_i & K_p & K_d \end{bmatrix}$$

Now, each 3x3 gain matrix, K_i, K_p, and K_d, will be diagonal:

$$K_p = \begin{bmatrix} k_{pR} & 0 & 0 \\ 0 & k_{pP} & 0 \\ 0 & 0 & k_{pY} \end{bmatrix}$$

where, row 1 column 2 would be the proportional gain for the Roll axis based on Pitch information and row 1 column 3 would be for the Roll axis based on Yaw information, hence, they are both zero. Similar arguments for the K_i and K_d matrices.

For the assignment, I suggest completing a 1D analysis for each axis to determine preliminary gains:

$$I\ddot{\theta} = -k_p\theta_{error} - k_d\dot{\theta}$$

for a desired response, settling time and percent overshoot.
Fine tune these gains, as necessary, when completing the full 3D maneuver.

Linear Quadratic Regulator (LQR)

Monday, October 7, 2019

PID control is widely used because of its simplicity, however, one difficulty remains, the tuning of the gains. This can be especially difficult for systems with coupled differential equations like the spacecraft attitude problem. Thus, an optimal method of finding gains is introduced with the Linear Quadratic Regulator (LQR) problem.

The LQR problem consists of linear equations of motion of the form:

$$\dot{\bar{x}} = \bar{f}\left(\bar{x},\bar{u}\right) = [A]\bar{x} + [B]\bar{u}$$

where $\bar{x}$ are the states and $\bar{u}$ are the controls. It also consists of a quadratic cost function:

$$J = \underbrace{\frac{1}{2}\bar{x}_f^T[S_f]\bar{x}_f}_{\Phi_f(\bar{x}_f)} + \int_0^t \underbrace{\frac{1}{2}\left(\bar{x}^T[Q]\bar{x} + \bar{u}^T[R]\bar{u}\right)}_{L(\bar{x},\bar{u})}dt$$

where $[S_f]$, $[Q]$, and $[R]$ are all symmetric weighting matrices. Φ_f is known as the terminal cost and L is known as the continuous cost. To minimize this cost function we use the Calculus of Variations. The exact detail of this are left to an optimization course but suffice it to say, the following are the necessary conditions such that dJ=0. These are known as the Euler-Lagrange Equations:

$$H = L + \bar{\lambda}^T\bar{f}$$

$$\dot{\bar{\lambda}}^T = -\frac{\partial H}{\partial \bar{x}}, \quad \bar{\lambda}^T(t_f) = \frac{\partial \Phi_f}{\partial \bar{x}_f}$$

$$\frac{\partial H}{\partial \bar{u}} = \bar{0}$$

Let's apply the linear EOMs and quadratic cost to the necessary conditions:

$$H = L + \bar{\lambda}^T\bar{f} = \frac{1}{2}\left(\bar{x}^T[Q]\bar{x} + \bar{u}^T[R]\bar{u}\right) + \bar{\lambda}^T\left([A]\bar{x} + [B]\bar{u}\right)$$

$$\dot{\bar{\lambda}}^T = -\frac{\partial H}{\partial \bar{x}} = -\left(\bar{x}^T[Q] + \bar{\lambda}^T[A]\right), \quad \bar{\lambda}^T(t_f) = \frac{\partial \Phi_f}{\partial \bar{x}_f} = \bar{x}_f^T[S_f]$$

$$\frac{\partial H}{\partial \bar{u}} = \bar{0} = \bar{u}^T[R] + \bar{\lambda}^T[B]$$

Solving the last equation for the control, $\bar{u}$:

$$\bar{u}^T[R] = -\bar{\lambda}^T[B]$$

$$[R]\bar{u} = -[B]^T\bar{\lambda}$$

$$\bar{u} = -[R]^{-1}[B]^T\bar{\lambda}$$

Recall that [R] is symmetric so its transpose is itself, but [B] in not necessarily symmetric so it has a transpose. This optimal control law can be used in the linear EOMs:

$$\dot{\bar{x}} = [A]\bar{x} + [B]\bar{u} = [A]\bar{x} - [B][R]^{-1}[B]^T\bar{\lambda}, \quad \bar{x}(0) = \bar{x}_0$$

This, along with the $\bar{\lambda}$ EOM, after a transpose:

$$\dot{\bar{\lambda}} = -\left([Q]\bar{x} + [A]^T\bar{\lambda}\right), \quad \bar{\lambda}(t_f) = [S_f]\bar{x}_f$$

results in what is called a Two-Point Boundary Value Problem (TPBVP) since we have half of the EOMs starting at time zero and half of the EOMs starting at the final time, t_f. To solve this system of equations, we notice that both EOMs are only a function of $\bar{x}$ and $\bar{\lambda}$ while the final $\bar{\lambda}$ is only a function of $\bar{x}$. Then, let's theorize that $\bar{\lambda}$ is always a function of $\bar{x}$:

$$\bar{\lambda}(t) = [S(t)]\bar{x}(t)$$

This clearly will satisfy the terminal constraint on $\bar{\lambda}$ but what must be true of the matrix [*S(t)*] such that the EOMs are satisfied? Let's begin by taking a derivative with respect to time and setting it equal to its EOM:

$$\dot{\bar{\lambda}}(t) = [\dot{S}]\bar{x} + [S]\dot{\bar{x}} = -[Q]\bar{x} - [A]^T\bar{\lambda}$$

Now, replace $\dot{\bar{x}}$ with its EOM and $\bar{\lambda}$ with $[S]\bar{x}$:

$$\dot{\bar{\lambda}}(t) = [\dot{S}]\bar{x} + [S]\left([A]\bar{x} - [B][R]^{-1}[B]^T[S]\bar{x}\right) = -[Q]\bar{x} - [A]^T[S]\bar{x}$$

Notice that every term is now post-multiplied by $\bar{x}$, thus everything that pre-multiplies $\bar{x}$ must be true:

$$[\dot{S}] + [S]\left([A] - [B][R]^{-1}[B]^T[S]\right) = -[Q] - [A]^T[S]$$

$$[\dot{S}] + [S][A] + [A]^T[S] - [S][B][R]^{-1}[B]^T[S] + [Q] = [0], \quad [S(t_f)] = [S_f]$$

This matrix EOM is known as the Matrix Riccati Equation. It is a symmetric matrix differential equation which means that you don't have to integrate all n^2 terms, just $\frac{1}{2}n(n+1)$ terms. To implement this, first we need to integrate the Matrix Riccati Equation backwards from the final time to the initial time. This is only possible if a final time exists. In the spacecraft attitude problem we never turn the controller off, so there is no final time. In this case the original quadratic cost function only has a continuous cost, not a terminal cost:

$$J = \int_0^\infty \underbrace{\frac{1}{2}\left(\bar{x}^T[Q]\bar{x} + \bar{u}^T[R]\bar{u}\right)}_{L(\bar{x},\bar{u})} dt$$

Everything we already did is still valid but now [S] must be a constant since if it has any slope, then as time goes to infinity, so will the matrix [S], thus:

$$[\dot{S}] = [0]$$

$$[S][A] + [A]^T[S] - [S][B][R]^{-1}[B]^T[S] + [Q] = [0]$$

This is known as the Algebraic Riccati Equation and can be solved numerically. It typically does not have algebraic solutions since it is a nonlinear equation. In MATLAB, the function "icare" can be used: [S,K]=icare(A,B,Q,R)
Once the Riccati Matrix, [S], is solved for then the optimal control is:

$$\bar{u} = -[R]^{-1}[B]^T\bar{\lambda} = -[R]^{-1}[B]^T[S]\bar{x} = -[K]\bar{x}$$

Where the optimal gain matrix, [K] is the second output of the "icare" function. Since the vector $\bar{x}$ is typically made up of position and velocity variables, the matrix [K] amounts to optimal PD control gains. If $\bar{x}$ includes integral states, then [K] are the PID gains.

Next we will investigate how to implement this general architecture on the spacecraft attitude problem.

LQR EOM and SIMULINK 7

Friday, March 13, 2020 11:12 PM

The LQR problem consists of linear equations of motion of the form:

$$\dot{\bar{x}} = \bar{f}\left(\bar{x},\bar{u}\right) = [A]\bar{x} + [B]\bar{u}$$

where $\bar{x}$ are the states and $\bar{u}$ are the controls. Luckily, we already have linearized equations of motion from out analysis of the gravity-gradient torque problem:

$$\begin{bmatrix} \ddot{\theta}_3 + (k_1 - 1)n\dot{\theta}_1 + 4k_1 n^2 \theta_3 \\ \ddot{\theta}_2 + 3k_2 n^2 \theta_2 \\ \ddot{\theta}_1 + (1 - k_3)n\dot{\theta}_3 + k_3 n^2 \theta_1 \end{bmatrix} = \bar{0}$$

These equations are only missing the control torque term which we can easily add to the right-hand side. Also, remember that each equation was divided by their respective mass moment of inertia value such that the coefficient multiplying the acceleration term was 1 and introducing the mass moment of inertia parameters, *k*. Thus, the control torques would also have to be divided by the mass moment of inertia values:

$$\begin{bmatrix} \ddot{\theta}_3 + (k_1 - 1)n\dot{\theta}_1 + 4k_1 n^2 \theta_3 \\ \ddot{\theta}_2 + 3k_2 n^2 \theta_2 \\ \ddot{\theta}_1 + (1 - k_3)n\dot{\theta}_3 + k_3 n^2 \theta_1 \end{bmatrix} = \begin{bmatrix} T_1 / I_1 \\ T_2 / I_2 \\ T_3 / I_3 \end{bmatrix}$$

Next, recall from the SIMULINK 6 Overview lecture we had the following approximations:

$$2\hat{q} \approx \begin{bmatrix} R \\ P \\ Y \end{bmatrix} = \begin{bmatrix} \theta_3 \\ \theta_2 \\ \theta_1 \end{bmatrix} \quad \text{and} \quad \bar{\omega}^B_{B/O} \approx \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_2 \\ \dot{\theta}_1 \end{bmatrix}$$

Via a derivative, we also have:

$$\dot{\bar{\omega}}^B_{B/O} \approx \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_2 \\ \ddot{\theta}_1 \end{bmatrix}$$

Now we can substitute ω and q values in for the Euler Angles are their derivatives:

$$\begin{bmatrix} \dot{\omega}_1 + (k_1 - 1)n\omega_3 + 4k_1 n^2 (2q_1) \\ \dot{\omega}_2 + 3k_2 n^2 (2q_2) \\ \dot{\omega}_3 + (1 - k_3)n\omega_1 + k_3 n^2 (2q_3) \end{bmatrix} = \begin{bmatrix} T_1 / I_1 \\ T_2 / I_2 \\ T_3 / I_3 \end{bmatrix}$$

Now, rearrange the equation to put it into the correct form we are looking for:

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} -8k_1 n^2 & 0 & 0 & 0 & 0 & (1-k_1)n \\ 0 & -6k_2 n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2k_3 n^2 & (k_3 - 1)n & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} 1/I_1 & 0 & 0 \\ 0 & 1/I_2 & 0 \\ 0 & 0 & 1/I_3 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

With $\bar{x}$ being defined by $\hat{q}$ and $\bar{\omega}$, we also need to define $\dot{\hat{q}}$:

$$2\hat{q} \approx \begin{bmatrix} R \\ P \\ Y \end{bmatrix} = \begin{bmatrix} \theta_3 \\ \theta_2 \\ \theta_1 \end{bmatrix} \rightarrow 2\dot{\hat{q}} \approx \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_2 \\ \dot{\theta}_1 \end{bmatrix} \approx \overline{\omega}$$

$$\dot{\hat{q}} \approx \frac{1}{2}\overline{\omega}$$

Putting this together with the other ODE yields the following [A] and [B] matrices:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ -8k_1n^2 & 0 & 0 & 0 & 0 & (1-k_1)n \\ 0 & -6k_2n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2k_3n^2 & (k_3-1)n & 0 & 0 \end{bmatrix}}_{[A]} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_1} & 0 & 0 \\ 0 & \frac{1}{I_2} & 0 \\ 0 & 0 & \frac{1}{I_3} \end{bmatrix}}_{[B]} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

These 2 matrices are conveniently put together in block format:

$$[A] = \begin{bmatrix} 0_{3x3} & \frac{1}{2}I_{3x3} \\ A_{2,1} & A_{2,2} \end{bmatrix}, \quad [A_{2,1}] = -2n^2\begin{bmatrix} 4k_1 & 0 & 0 \\ 0 & 3k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix}, \quad [A_{2,2}] = n\begin{bmatrix} 0 & 0 & (1-k_1) \\ 0 & 0 & 0 \\ (k_3-1) & 0 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} 0_{3x3} \\ I^{-1} \end{bmatrix}$$

The matrix [*B*] does assume that the Body Axes are colinear with the Principle Axes such that the mass moment of inertia matrix is diagonal. That is why rows 4 through 6 is the inverse of the mass moment of inertia matrix.

Now we can solve the Algebraic Riccati Equation using the MATLAB function "icare". This function will also output the optimal gain matrix [*K*].

[S,K] = icare(A,B,Q,R)

$$\overline{T} = -[K]\overline{x}$$

Since x contains positions, q, and velocities, ω, this LQR Controller is an optimal PD Controller

SIMULINK 7:

Q and R are both symmetric matrices. The magnitude of the values in Q and R are not important, only the ratio of the values in Q to the values in R. For example, if one set of values for Q and R result in a cost, J, and then multiply Q and R both by 10 it will not effect the states, x, or the controls, u, since they are weighted the same with respect to each other as the previous case. Thus I will end up with the same dynamical solution with a cost, J, 10 times bigger.

If you increase the values in Q, for constant R, it will make this more important in the cost function and drive the states, ω and q, to zero faster by using more torque. Likewise, if you decrease the values in Q, for constant R, it will drive the states to zero more slowly using less torque.

Suggest using R = eye(3) without the torque saturation block to start. Then adjust the values in Q by orders of magnitude until you are close to the desired solution. The first 3 values on the main diagonal of Q multiply the proportional states, q, and thus effect the proportional gains. The last 3 values on the main diagonal of Q multiply the derivative states, ω, and thus effect the derivative gains. You can still use your 1-dimensional analysis to know what gain values you need to achieve a given settling time and overshoot requirement.

Latitude/Longitude Pointing

Tuesday, October 29, 2019 4:08 PM

Target Quaternion

To point at a specific latitude and longitude location on the surface of the Earth, first we must determine that position in the Fixed Frame. This also involves calculating the position of Greenwich, located on the Prime Meridian (0° Longitude), with respect to the x-axis, θ_g. This problem is solved using Greenwich Mean Sidereal Time (GMST) that you probably learned in your Space Mechanics course.

$$GMST = 18.697374558 + 24.06570982441908D$$

where D is the number of days since January 1, 2000 at noon UT and GMST is in hours. GMST represents the number of hours into a full 24 hour rotation of Earth from the x-axis.

Example:

Find the GMST for April 11, 2024 at 11:30 AM PST. From PST to UT is a conversion of +7 hours (currently) so this time is 18:30 UT.

The Julian Date of April 11, 2024 at 18:30 UT is 2460412.27083333 days

The Julian Data of January 1, 2000 at 12 UT is 2451545 days

The difference is D = 8867.270833333 days

Then the GMST = 213415.8641841 hr

Since the Earth rotates once every 24 hours, we can modify the GMST to be between 0 and 24 hours by subtracting off an integer multiply of 24.

mod(GMST,24) = 7.8641841 hr

To calculate the angle of Greenwich relative to the x-axis, just multiply by 360°/24 hr.

$\theta_g = 117.963° = 2.05884$ rad

To determine the starting time of our SIMULINK simulation, in seconds, convert the GMST to seconds:

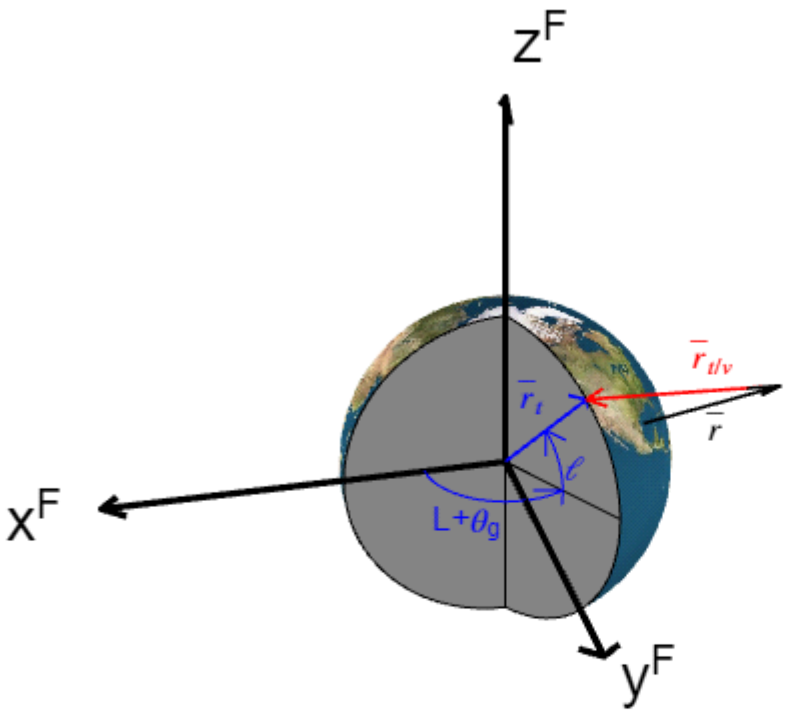
GMST = 28311 sec

Then, in our model, we have that:

$\theta_g = \omega_{\oplus} t$

where $\omega_{\oplus}$ is the angular velocity of the Earth in rad/s, i.e., 2π/86400 rad/s, and t is time in seconds. Note: When t = 0, in the equation and in our model, the Prime Meridian will be colinear with the x-axis. When t = 28311 sec, the angular position of Greenwich, θ_g, will be the desired 2.05884 rad.

With the angular position of Greenwich defined, we can determine the position of the target using its' latitude, ℓ, and longitude, *L*. Note: I recommend using radians so you don't have to use "sind" and "cosd" functions. Also, for latitude, South is negative and North is positive. For longitude, West is negative and East is positive.



$$\bar{r}_t^F = r_\oplus \begin{bmatrix} \cos(\ell)\cos(L+\theta_g) \\ \cos(\ell)\sin(L+\theta_g) \\ \sin(\ell) \end{bmatrix}$$

Next, we find the position of the target, *t*, relative to the vehicle, *v*, through vector algebra. This is done with the position vector of the spacecraft in the Fixed Frame, which we have in our SIMULINK model:

$$\bar{r}_{t/v}^F = \bar{r}_t^F - \bar{r}^F$$

Now, we rotate this from the Fixed Frame to the Orbit Frame using the same Direction Cosine Matrix from SIMULINK 4 Animation:

$$\bar{r}_{t/v}^O = DCM_{O/F}\bar{r}_{t/v}^F$$

We want the z-axis of the spacecraft to be pointing in this direction we simply need to determine the attitude of the $\bar{r}_{t/v}^O$ vector relative to $\hat{z}^O = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, nadir pointing. We do this using the formula for the quaternion between 2 vectors:

$$\bar{q} = \frac{1}{2r_1r_2\cos\frac{\phi}{2}} \begin{bmatrix} \bar{r}_1 \times \bar{r}_2 \\ r_1r_2(1+\cos\phi) \end{bmatrix}$$

where $\bar{r}_1 = \hat{z}^O$ and $\bar{r}_2 = \bar{r}_{t/v}^O$. This quaternion is the target attitude of the Body Frame relative to the Orbit Frame. To get the quaternion error that our LQR controller needs, use the same math as SIMULINK 6:

$$\bar{q}_{error} = \bar{q}_{target}^{-1} \otimes \bar{q}_{observed}$$

Target Angular Velocity

This attitude will result in the spacecraft pointing its z-axis towards the target, however, this quaternion is not constant; it changes as the spacecraft orbits the Earth and the Earth rotates. Next we have to determine the target angular velocity of the spacecraft to stay pointing at the target. We start by looking at the quaternion EOM:

$$\dot{\bar{q}} = \frac{1}{2}[\Omega]\bar{q} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}\bar{q}$$

In this equation, we can estimate the derivative of the quaternion using finite difference methods; in its' simplest form, at time t_k, this is:

$$\dot{\bar{q}}_k = \frac{\bar{q}_k - \bar{q}_{k-1}}{\Delta t}$$

Now, we just rearrange the quaternion EOM to solve for $\bar{\omega}$. Since there are only 3 independent values of the quaternion, we only need 3 of the 4 rows. Choosing the first 3 rows for mathematical convenience yields:

$$\dot{\bar{q}} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

$$\dot{\hat{q}} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

$$\bar{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = 2\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \end{bmatrix}^{-1}\dot{\hat{q}}$$

Explicitly showing the matrix inverse using the Minor-Cofactor-Adjugate method:

$$2\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \end{bmatrix}^{-1} = \frac{2}{\det}\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}^T$$

$$\det = q_4\left(q_4^2 + q_1^2\right) - (-q_3)(q_3q_4 - q_1q_2) + q_2(q_1q_3 + q_2q_4)$$

$$\det = q_4\left(q_4^2 + q_1^2 + q_2^2 + q_3^2\right) - q_1q_2q_3 + q_1q_2q_3 = q_4$$

$$C_{11} = q_4^2 + q_1^2 \qquad C_{12} = -(q_3q_4 - q_1q_2) \qquad C_{13} = q_1q_3 + q_2q_4$$

$$C_{21} = -(-q_3q_4 - q_1q_2) \qquad C_{22} = q_4^2 + q_2^2 \qquad C_{23} = -(q_1q_4 - q_2q_3)$$

$$C_{31} = q_1q_3 - q_2q_4 \qquad C_{32} = -(-q_1q_4 - q_2q_3) \qquad C_{33} = q_4^2 + q_3^2$$

$$2\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \end{bmatrix}^{-1} = \frac{2}{q_4}\begin{bmatrix} 2\left(q_4^2+q_1^2\right) & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right) \\ 2\left(q_1q_2-q_3q_4\right) & 2\left(q_4^2+q_2^2\right) & 2\left(q_2q_3+q_1q_4\right) \\ 2\left(q_1q_3+q_2q_4\right) & 2\left(q_2q_3-q_1q_4\right) & 2\left(q_4^2+q_3^2\right) \end{bmatrix}$$

The off-diagonal parts of this matrix are simply the Direction Cosine Matrix in terms of the quaternion. The main-diagonal components are a little different:

$$2\left(q_4^2+q_1^2\right)=\left(q_4^2+q_1^2-q_2^2-q_3^2\right)+\left(q_4^2+q_1^2+q_2^2+q_3^2\right)=DCM(1,1)+1$$

$$2\left(q_4^2+q_2^2\right)=\left(q_4^2-q_1^2+q_2^2-q_3^2\right)+\left(q_4^2+q_1^2+q_2^2+q_3^2\right)=DCM(2,2)+1$$

$$2\left(q_4^2+q_3^2\right)=\left(q_4^2-q_1^2-q_2^2+q_3^2\right)+\left(q_4^2+q_1^2+q_2^2+q_3^2\right)=DCM(3,3)+1$$

Given this, we can write the target angular velocity of the spacecraft as:

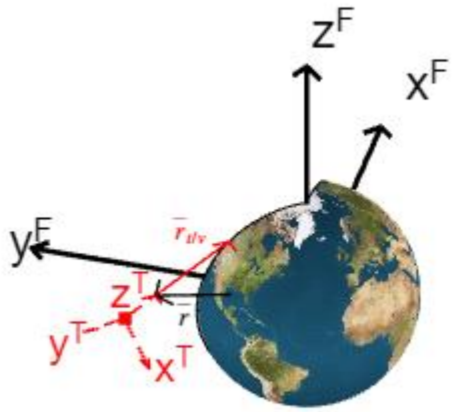
$$\overline{\omega}=\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}=2\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \end{bmatrix}^{-1}\dot{\hat{q}}=\left(DCM(\overline{q})+I_{3x3}\right)\dot{\hat{q}}$$

Similarly to the quaternion, the LQR requires angular velocity error so we need this target angular velocity relative to the angular velocity of the spacecraft. In this case, since the quaternion we are using is Body Frame relative to Orbit Frame, the angular velocity must be as well:

$$\overline{\omega}_{error}^{B}=\overline{\omega}_{target}^{B}-\overline{\omega}_{B/O}^{B}$$

Target View

Sunday, May 19, 2024 9:22 AM



Above is a depiction of the orientation of the Target (T) Frame. In the last lecture we determined the orientation of this frame relative to the Orbit Frame. Now, let us place a camera where the square is such that the direction of the camera points along the target relative to vehicle direction, i.e., z^T.

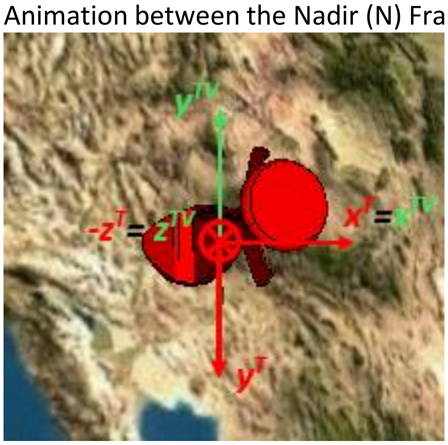
The position of the Target View (TV) camera is the position of the spacecraft minus a distance between the camera and the spacecraft, I like to use 0.3 DU for LEO satellites, along the target relative to the vehicle direction, i.e.:

$$\bar{r}_{TV}^{F} = \frac{\bar{r}}{6378} - 0.3\hat{r}_{t/v}^{F}$$

To get the correct orientation of the camera we need the following quaternion sequence:

$$\bar{q}_{TV/F} = \bar{q}_{O/F} \otimes \bar{q}_{T/O} \otimes \bar{q}_{TV/T}$$

The quaternion of the Orbit (O) Frame relative to the Fixed (F) Frame was discussed and calculated in the Simulink 4 Animation lecture. The quaternion of the Target (T) Frame relative to the Orbit (O) Frame is the quaternion derived in the previous lecture, Latitude/Longitude Pointing. Finally, the quaternion of the Target View (TV) Frame relative to the Target (T) Frame is the same as that seen in Simulink 4 Animation between the Nadir (N) Frame and the Orbit (O) Frame. Specifically:



$$\overline{q}_{TV/T} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \sin\left(\frac{\pi}{2}\right) \\ \cos\left(\frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\overline{q}_{TV/F} = \overline{q}_{T/F} \otimes \overline{q}_{TV/T} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \overline{q}_{T/F}$$

Once the quaternion is calculated, then use the algorithm to compute the Euler Parameters, $\bar{e}$ and ϕ, from the quaternion. Remember, the Euler Parameters are what the VR Sink in SIMULINK needs.

With this position and orientation the Target View Camera will point through the spacecraft to the target. This will be slightly different than the Nadir View Camera that points through the spacecraft to the center of the Earth.