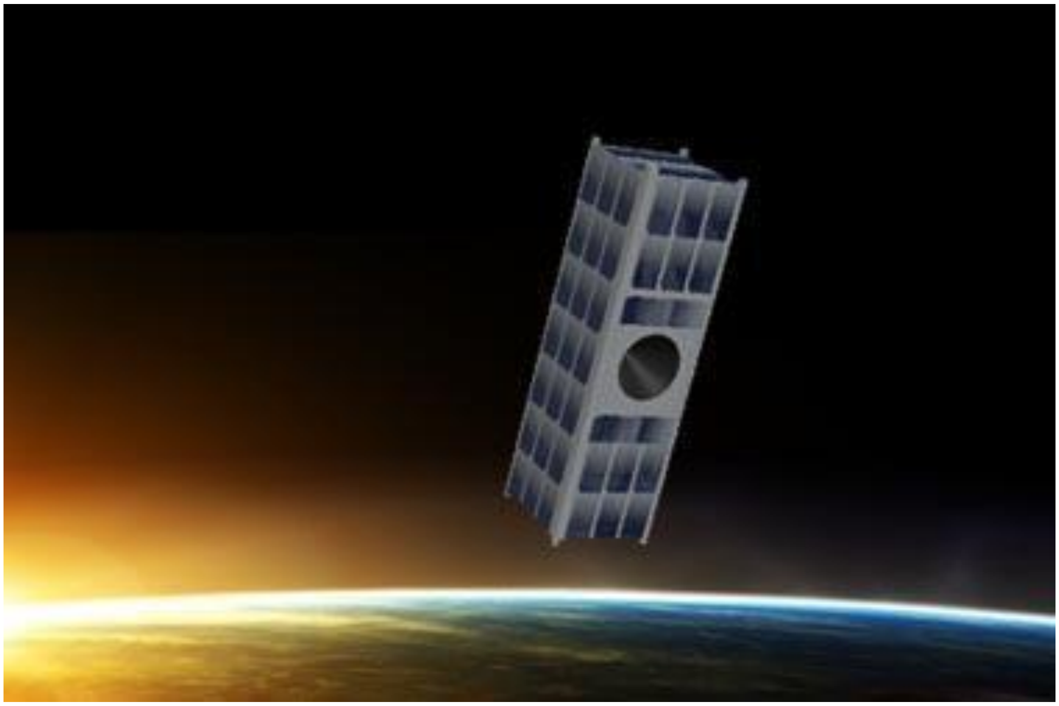
Attitude Control Design



AE 426

Aerospace Engineering Department

Embry-Riddle Aeronautical University

Daytona Beach, FL

Christopher Cabrera

Overview

The goal of this final project was to essentially demonstrate our understanding of attitude control of a satellite. In this case a 3U cubesat that is approximately at a 400km circular orbit above the earth. We are told that initially our satellite is aligned with the Local Vertical Local Horizon (LVLH) frame. We will first graph the response of said satellite with a disturbance under torque-free conditions with our own initial condition for the attitude. Once the cubesat's spin is stabilized the attitude trajectory along time while using quaternions will be found. Then the gravity gradient torque for attitude dynamics will be considered, and the appropriate attitude response of the satellite with respect to said disturbance found. A comparison of the stability analysis by linearization results will then be presented and discussed.

Methodology

Before we begin the process of understanding and explaining the project we must understand the general concepts of Euler angles, quaternions, direction cosine matrices. Understanding these concepts we will then derive how to convert from DCM to quaternions, quaternions to Euler angles as well as quaternions back to a DCM. The following general formula demonstrates a conversion from the body fixed frame to the inertial frame of a body. As well as what torque free motion is and gravity gradient.

$$[\mathbf{Q}]_{Xx} = [\mathbf{R}_1(\psi)][\mathbf{R}_2(\theta)][\mathbf{R}_3(\phi)]$$

With the following angles (psi, phi, theta) representing yaw pitch and roll angles also known as the classic Euler sequence. Each matrix is represented as shown;

$$[\mathbf{R}_1(\psi)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \quad [\mathbf{R}_2(\theta)] = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad [\mathbf{R}_3(\phi)] = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Which then results in this DCM that transforms from XYZ to xyz, and its transpose matrix which transforms xyz to XYZ;

$$[\mathbf{Q}]_{Xx} = \begin{bmatrix} \cos\phi\cos\theta & \sin\phi\cos\theta & -\sin\theta \\ \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\theta\sin\psi \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\theta\cos\psi \end{bmatrix}$$

$$[\mathbf{Q}]_{xX} = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \sin\phi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix}$$

Now that DCM's have been clearly defined quaternions are next. Quaternions as we know were introduced in 1843 by Sir William Hamilton as an alternative way to use a DCM to describe the orientation of a body in a 3-D reference frame, while avoiding the issue of the singularities run into by DCM's. Quaternions are made up of the classic Euler Sequence, yaw, and pitch and roll angles, the vector portion and q4 the last component being a scalar.

$$\{\hat{\mathbf{q}}\} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} \mathbf{q} \\ q_4 \end{Bmatrix}$$

If given the quaternion and a DCM is needed conversion from one to the other is simple and as follows.

$$[\mathbf{Q}]_{Xx} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

Similarly with DCM to a quaternion, with the given equations conversion is simple.

$$q_1 = \frac{Q_{23} - Q_{32}}{4q_4} \quad q_2 = \frac{Q_{31} - Q_{13}}{4q_4} \quad q_3 = \frac{Q_{12} - Q_{21}}{4q_4} \quad q_4 = \frac{1}{2}\sqrt{1 + Q_{11} + Q_{22} + Q_{33}}$$

For the project, Appendix D in Curtis' text all of these conversions are written out in Matlab and were utilized throughout the project.

Torque free motion is simply catergorized as a satellite orbiting with gravity as its only force. Since the Moment about the center of mass on said satellite is 0 the satellite is considered torque free.

$$\dot{\mathbf{H}}_G = \mathbf{0}$$

Thus resulting in general equations of motion (EOM) as follows;

$$\begin{aligned} A\dot{\omega}_x + (C - B)\omega_z\omega_y &= 0 \\ B\dot{\omega}_y + (A - C)\omega_x\omega_z &= 0 \\ C\dot{\omega}_z + (B - A)\omega_y\omega_x &= 0 \end{aligned}$$

And since in our particular case our 3U cube sat our geometrical A component is equal to that of B we can say. Where angular velocity is considered a constant.

$$\begin{aligned} A\dot{\omega}_x + (C - A)\omega_y\omega_z &= 0 \\ A\dot{\omega}_y + (A - C)\omega_z\omega_x &= 0 \\ C\dot{\omega}_z &= 0 \end{aligned}$$

Now for the gravity gradient portion equations result as follows, in this scenario we do have a moment acting on our body.

$$A\dot{\omega}_x + (C - B)\omega_y\omega_z = \frac{3\mu R_y R_z}{R^5}(C - B)$$

$$B\dot{\omega}_y + (A - C)\omega_z\omega_x = \frac{3\mu R_x R_z}{R^5}(A - C)$$

$$C\dot{\omega}_z + (B - A)\omega_x\omega_y = \frac{3\mu R_x R_y}{R^5}(B - A)$$

In order to start with the problem at hand we are given two sets of equations, one of which is called the kinematics equations which describe the attitude/orientation change due to angular velocity changes in the body (Eq. 1). The second one is called the dynamics equations, this equation describes the angular velocity changes due to the applied torque/moment at the body (Eq. 2). As we were told in the assignment, Eq 1 utilizes a quaternion instead of conventional Euler angles to eliminate the singularity of Gimbal lock that could exist with arbitrary rotations.

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_s & -q_3 & q_2 \\ q_3 & q_s & -q_1 \\ -q_2 & q_1 & q_s \end{bmatrix}\boldsymbol{\omega} \qquad (1)$$

$$I_{\mathrm{moi}}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (I_{\mathrm{moi}}\boldsymbol{\omega}) + \boldsymbol{M}_{\mathrm{torq}} \qquad (2)$$

For part 1-3 the general code was provided by utilizing Example 9.23 in Curtis' Orbital Mechanics for Engineering Students, then altering it to produce the results required by the project. In order to begin we must first specify the initial orientation of the XYZ axes of the satellites body frame. In other words we must define initial values for the DCM (direction cosine matrix) as done here in the Matlab code.

```
z = [sind(theta) 0 cosd(theta)]; % Initial z-axis direction:
p = [0 1 0]; % Initial x-axis direction (or a line defining x-z plane)
y = cross(z,p); % y-axis direction (normal to x-z plane)
x = cross(y,z); % x-axis direction (normal to y-z plane)
```

This was done in this way because the y direction on our body fixed frame is always normal to the x-z plane, similarly with the x direction which is always normal to the y-z plane. With the z direction being the reference direction. After this had been done the quaternion is computed and then initial value for the angular velocity is specified in body frame components as demonstrated below.

```
QXx = [i; j; k]; % Initial direction cosine matrix
q0 = q_from_dcm(QXx);% Initial quaternion
w0 = [dx dy dz+wspin]'; % Initial body-frame angular velocities (rad/s)
```

Finally to linearize these results and retrieve the kind of data we want to retrieve we have to solve the resulting differential equations. Matlab offers ODE45 that is directly incorporated in the program so no separate functions have to be written to approximate the ode, as we can see below.

```
[t,f] = ode45(@rates, [t0,tf], f0); %ODE solver in Matlab.
```

After running and computing the quaternions as well as the angular velocity in n-steps, then converting those solutions from the quaternions into DCMs then finally to Euler equations we retrieve our final solutions. Results were then graphed as seen below in the results section

These are the general steps that were taken for all 3 parts of the project. For task 1 initial parameters that were chosen were a theta of 30 degrees and a time iteration of 10 seconds with a spin rate (rad/s) of 0. Figure 1 shows the nutation, precession and spin rate angels for task 1.

Task 2 was nearly the same procedure but in this instance the spin rate was constant and had a value of 5 rad/s and theta remained the same as well as the time iteration. The results for task 2 can be seen in Figure 2 in the results section.

Task 3 proved to be more difficult since gravity gradient torque was to be applied to the satellite. Where firstly the equilibrium status was to be found. For these initial conditions the spin rate was the same as task 2's but the theta angle was set to 60 degrees. With a time iteration of 10 seconds. For this task since the moment about the center of gravity of our satellite had a torque applied to it a different rates function was used for the calculating the solution of the ODE.

```
wx_dot = M(1)/A - (C - B)*wy*wz/A; % time derivative of wx
wy_dot = M(2)/B - (A - C)*wz*wx/B; % time derivative of wy
wz_dot = M(3)/C - (B - A)*wx*wy/C; % time derivative of wz
```

We can see here that when compared to case 1 and 2 the scenario here is with gravity gradient, resulting in the EOM's not being equal to zero.

Results

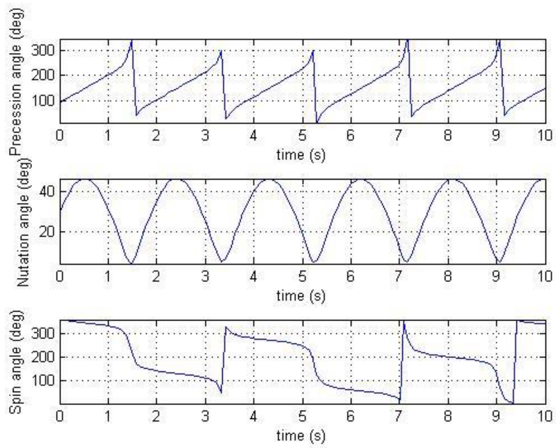


Figure 1: Results of Part I showing nutation, precession and spin angles varying with time (s)

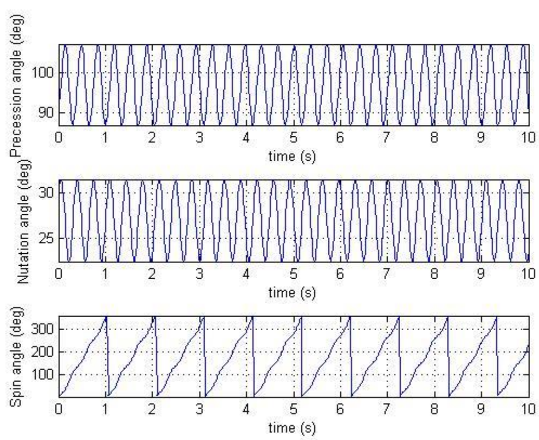


Figure 2: Results of Part II showing nutation, precession and spin angles varying with time (s)

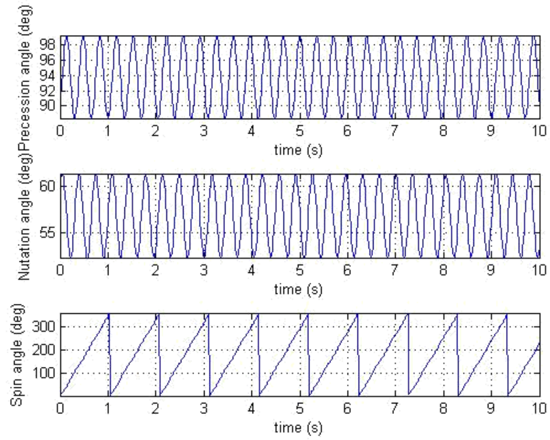


Figure 3: Results of Part III showing nutation, precession and spin angles varying with time (s)

When all three of the tasks are compared we see several similarities and differences. Task 3 for obvious reasons is different from task 2 and 1 because the gravity gradient applied. Task 1 and 2 were only different in the sense that task 2 only had a constant spin rate but still remained a torque free system. These simulation results when compared to analysis done by linearization demonstrate that we can retrieve the same answer but when done by linearization only general results of the ODE are obtained where in the simulations case we are able to find our actual attitude trajectory and see the nutation, precession and spin angles change as a function of time.

Conclusion

In conclusion not only was a general understanding of attitude control design was demonstrated as well as the application of several attitude scenarios. The 3U cubesat behavior under different conditions would be unattainable without the use of both DCM's and quaternions.

References

[1] Curtis, Howard D. *Orbital Mechanics for Engineering Students*. 3rd ed. Oxford: Elsevier Butterworth-Heinemann, 2005. Print.