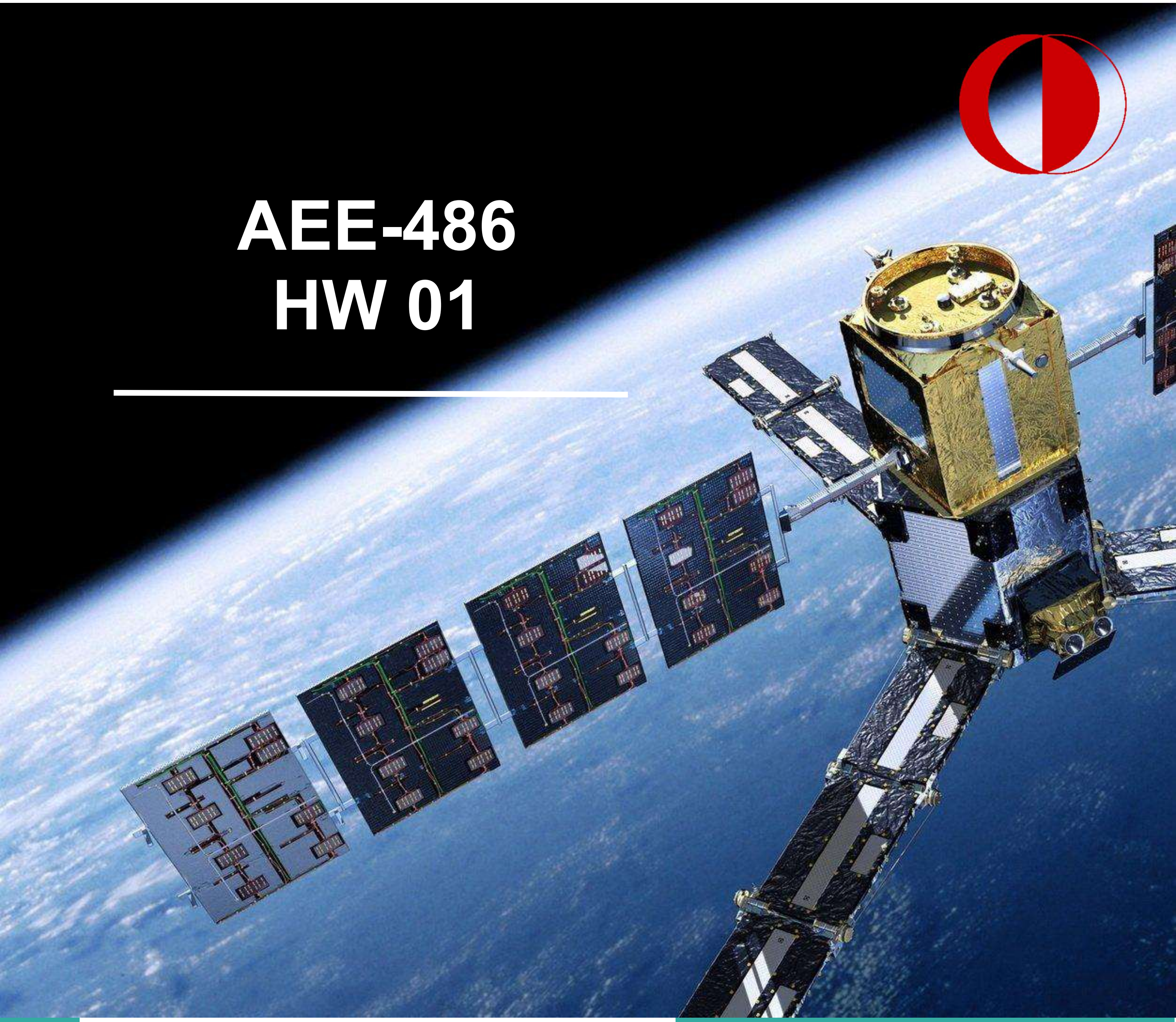
AEE-486
HW 01



Muhammad Owais Usman
2487437

Q1

(a)

In this part we were expected to find the DCM matrix that transforms a vector in the chief orbit frame to the ECI frame.

In order to that the DCM matrix we need is as follows:

$$A_{io_c} = [\overrightarrow{O_1} \quad \overrightarrow{O_2} \quad \overrightarrow{O_3}]$$

Where O_1, O_2, O_3 are column vectors of the size 3×1. The values of these column vectors can be found by calculating the below expressions:

$$\overrightarrow{O_3} = \frac{-\vec{R}}{\|\vec{R}\|}$$

$$\overrightarrow{O_2} = \frac{-\vec{R} \times \overrightarrow{v_c}}{\|\vec{R} \times \overrightarrow{v_c}\|}$$

$$\overrightarrow{O_1} = \overrightarrow{O_2} \times \overrightarrow{O_3}$$

Where $\overrightarrow{v_c}$ is the velocity vector of the chief satellite and $\vec{R}$ is the vector connecting the chief orbit satellite to the center of the earth.

(b)

In this part we were expected to find the DCM matrix that transforms a vector in the deputy orbit frame to the ECI frame.

In order to that the DCM matrix we need is as follows:

$$A_{io_d} = [\overrightarrow{O_1} \quad \overrightarrow{O_2} \quad \overrightarrow{O_3}]$$

Where O_1, O_2, O_3 are column vectors of the size 3×1. The values of these column vectors can be found by calculating the below expressions:

$$\overrightarrow{O_3} = \frac{-\vec{r}}{\|\vec{r}\|}$$

$$\overrightarrow{O_2} = \frac{-\vec{r} \times \overrightarrow{v_d}}{\|\vec{r} \times \overrightarrow{v_d}\|}$$

$$\overrightarrow{O_1} = \overrightarrow{O_2} \times \overrightarrow{O_3}$$

Where $\overrightarrow{v_d}$ is the velocity vector of the chief satellite and $\vec{r}$ is the vector connecting the chief orbit satellite to the center of the earth.

(c)

In order to find the DCM which relates the chief orbital matrix to the deputy orbital matrix, $A_{o_c o_d}$, we have to follow the following sequence of matrix multiplication:

$$A_{o_c o_d} = A_{o_c i}\, A_{i o_d}$$

We know the value of the DCM $A_{i o_d}$ from part (b), but to find the value of $A_{o_c i}$ we have to find the inverse of the matrix $A_{i o_c}$, DCM found in part (a). To find the inverse, we can just take the transpose of $A_{i o_c}$, as it is a rotation matrix which is orthonormal:

$$A_{o_c i} = \left(A_{i o_c}\right)^{-1} = \left(A_{i o_c}\right)^{T}$$

Now substituting in the above equation:

$$A_{o_c o_d} = \left(A_{i o_c}\right)^{T} A_{i o_d}$$

(d)

In order to find the DCM that transforms a vector in the ECI frame to the body frame, $A_{b_c i}$, we have to follow the following multiplication order using $A_{b_c o_c}$ and the DCM matrices obtained in the previous parts:

$$A_{b_c i} = A_{b_c o_c}\, A_{o_c i}$$

As we saw earlier,

$$A_{o_c i} = \left(A_{i o_c}\right)^{-1} = \left(A_{i o_c}\right)^{T}$$

Hence:

$$A_{b_c i} = A_{b_c o_c} \left(A_{i o_c}\right)^{T}$$

(e)

In order to show the relative attitude of the two spacecraft, we need the following matrix multiplication:

$$A_{b_c b_d} = A_{b_c i}\, A_{i b_d}$$

We have the DCM $A_{b_c i}$ from the previous part. But in order to find $A_{i b_d}$, we need $A_{o_d b_d}$, which is the DCM relating the deputy orbit frame to the attitude of the deputy satellite. Hence, unless we are given this DCM, it is impossible to find the DCM which relates the attitudes of the two satellites to each other.

Q2

In this question we were asked to make functions in MATLAB to convert DCM to quaternions. This was done using two methods, which were then checked with MATLAB's inbuilt function to convert DCM to quaternions, "dcm2quat".

First of all, three random DCMs were generated by using the MATLAB function "angle2dcm". Three sets of Euler angles were generated of which 2 were asymmetric and one was symmetric:

DCM-01

-0.6285	-0.3687	0.6848
0.0611	-0.9012	-0.4291
0.7754	-0.2279	0.5889

DCM-02

0.2973	-0.9137	-0.2770
-0.8282	-0.3912	0.4014
-0.4751	0.1101	-0.8730

DCM-03

-0.1833	-0.4016	0.8973
-0.2924	-0.8492	-0.4398
0.9386	-0.3430	0.0382

- **The first method was used by implementing the following formulation to the function:**

$$q_4 = \pm\frac{1}{2}\sqrt{1 + \text{Trace}(A)}$$

$$q_1 = \frac{1}{4q_4}(a_{23} - a_{32})$$
$$q_2 = \frac{1}{4q_4}(a_{31} - a_{13})$$
$$q_3 = \frac{1}{4q_4}(a_{12} - a_{21})$$

This resulted in the following quaternions presented in a 3 x 3 matrix, where each column represents the corresponding quaternion:

Quaternions by method 1

-0.4134	0.8002	-0.6379
0.1860	-0.5442	0.2720
-0.8830	-0.2350	-0.7195
0.1217	0.0910	0.0379

- The second method was implemented by first converting the DCM to the Euler axis/angle convention using the following formulation provided in the lecture notes:

$$\cos\vartheta = \frac{1}{2}[\mathrm{Trace}(A) - 1] \quad \text{and} \quad \hat{\mathbf{e}} = \frac{1}{2\sin\vartheta}\begin{bmatrix} a_{23} - a_{32} \\ a_{31} - a_{13} \\ a_{12} - a_{21} \end{bmatrix}$$

and then converting the Euler axis/angle convention to the quaternion convention using the following formulation:

$$\mathbf{q} \equiv \begin{bmatrix} \boldsymbol{\varrho} \\ q_4 \end{bmatrix}, \quad \boldsymbol{\varrho} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\mathbf{e}}\,\sin\frac{\vartheta}{2}, \quad q_4 = \cos\frac{\vartheta}{2}$$

This resulted in the following quaternions presented in a 3 x 3 matrix, where each column represents the corresponding quaternion:

Quaternions by method 2

-0.4134	0.8002	-0.6379
0.1860	-0.5442	0.2720
-0.8830	-0.2350	-0.7195
0.1217	0.0910	0.0379

- **Both the methods were compared to the inbuilt MATLAB function “dcm2quat” which resulted in the following quaternions presented in a 3 x 3 matrix, where each column represents the corresponding quaternion:**

-0.1217	0.0910	-0.0379
0.4134	0.8002	0.6379
-0.1860	-0.5442	-0.2720
0.8830	-0.2350	0.7195

  However, there are 2 differences which are both acceptable.
  - **The first term is the scalar one in the MATLAB inbuilt “dcm2quat” function, while the fourth term is the scalar term in our functions. This was kept like this due to the need of these functions in later questions so as to not cause errors due to this rearrangement.**
  - **The second is that our functions give a negative quaternion compared to the MATLAB function “dcm2quat”. This is also perfectly fine as for quaternions, $-\vec{q} = \vec{q}$.**

The comparison between the first function and the second function can be made that the first one could give a singularity based on the fact that it requires the term $\frac{1}{q_4}$ in the equation to find the terms $q_1, q_2,$ and q_3. Hence in the further questions the use of the second function is preferred.

Q3

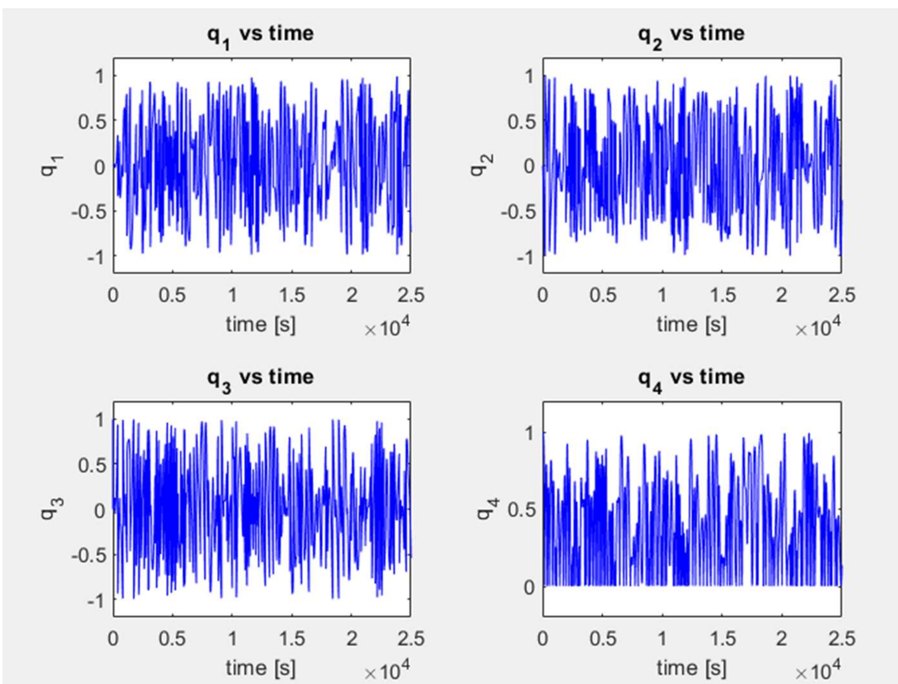
In this question we were asked to propagate the DCM $A_{bi} = I_3$, using attitude kinematics. The equation used to find $\dot{A_{bi}} = -[\omega_{bi}\times] * A_{bi}$. A function was created to calculate:

$$[\omega_{bi}\times] = \begin{matrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{matrix}$$

Then the DCM was propagated using the 4th order Runge-Kutta method. The new DCM at every time step was converted to its corresponding quaternion. The resulting end quaternion is given below:

-0.7346
-0.3839
-0.5425
0.1370

The plots for each quaternion term are given below. They all fluctuate at almost between -1 and 1 except for q_4, which rarely becomes negative as it is a scalar quantity.



Q4

This question is divided into 2 parts:

- The first parts require us to use quaternion kinematics to propagate instead of the DCM. Hence the following equation was used:

$$\dot{q} = \frac{1}{2}\Omega\big(\omega_{bi}(t)\big)q(t)$$

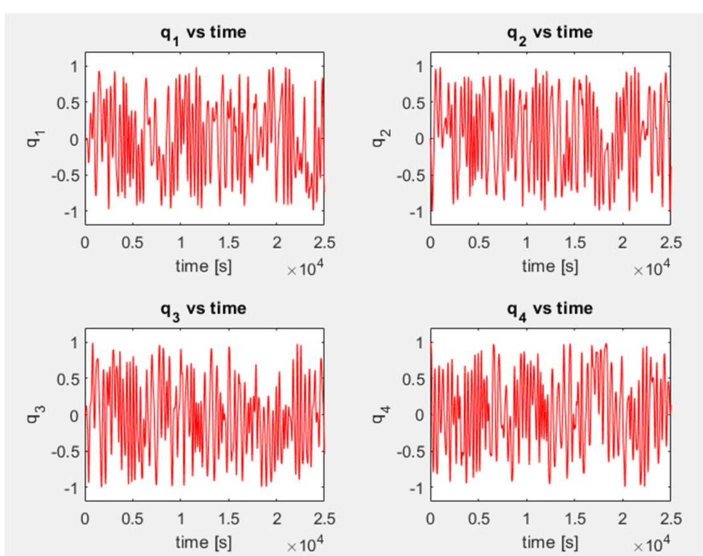
A function was created to calculate $\Omega\big(\omega_{bi}(t)\big)$ resulting in a 4 x 4 matrix:

$$\Omega\big(\omega_{bi}(t)\big) = \begin{matrix} -[\omega_{bi}\times] & \omega_{bi} \\ -\omega_{bi}^T & 0 \end{matrix}$$

Again, 4th order Rung-Kutta was used to propagate the quaternion. The quaternion was normalized at each iteration. The final quaternion was as follows:

-0.7346
-0.3839
-0.5425
0.1370

Which is the same as the one obtained in question 3. The plots of each term of the quaternion are given below:

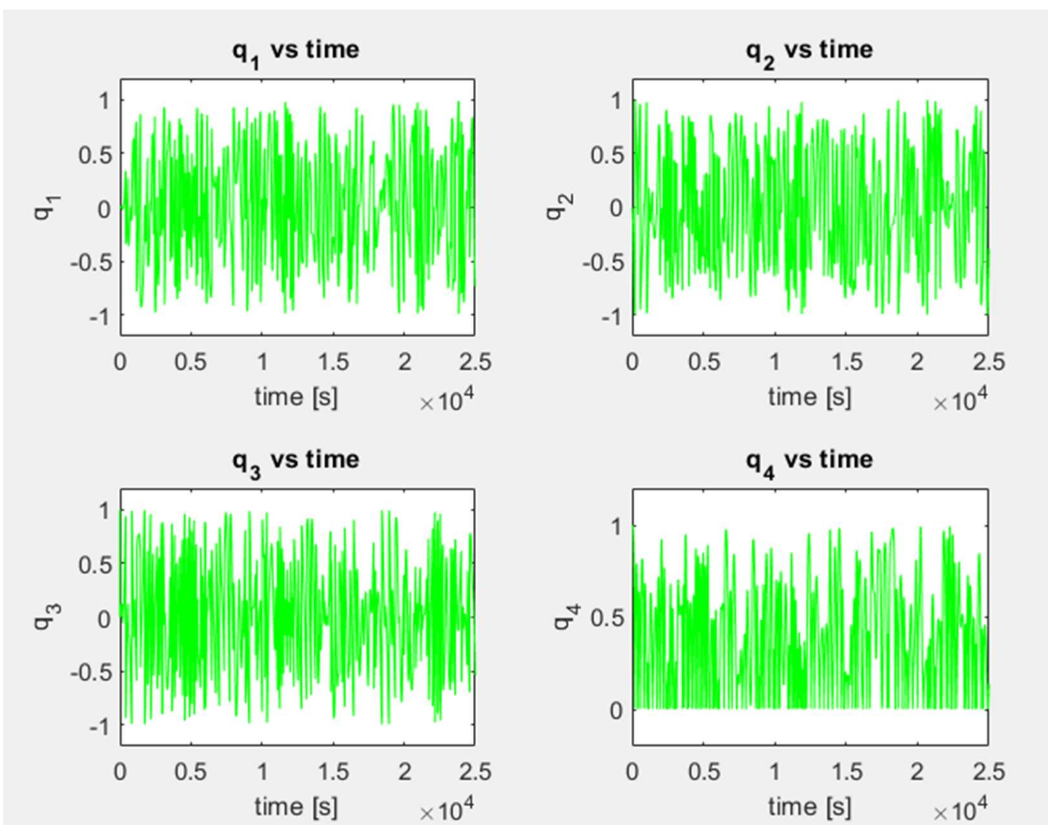


The reason that the plot of q_4 fluctuates between -1 and 1 instead of 0 and 1 like Q3, is because quaternion kinematics were used in Q4, while in Q3 DCM kinematics were used which were then converted to quaternions giving the value of q_4 as always positive.

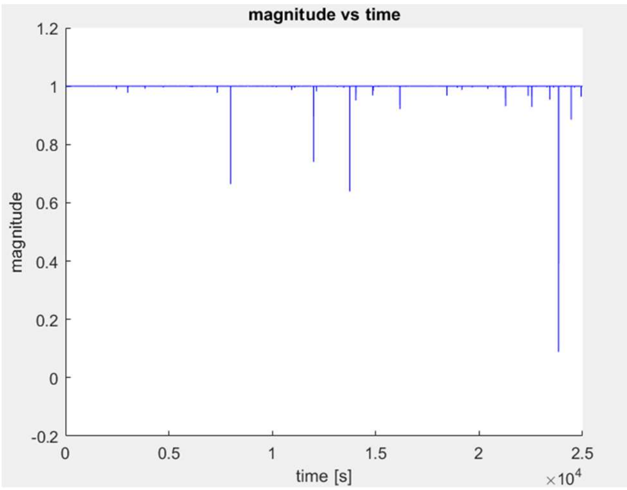
- The second part of the question asked us to orthonormalize the DCM at every iteration in Q3. This was done using the Gram–Schmidt orthonormalization process. The resulting final quaternion is given below:

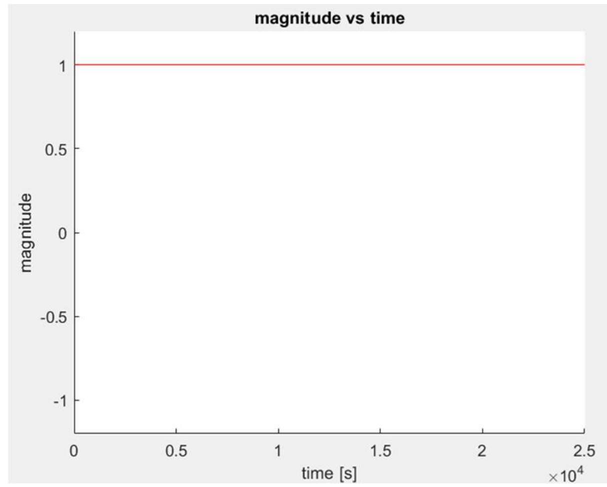
-0.7346
-0.3839
-0.5425
0.1370

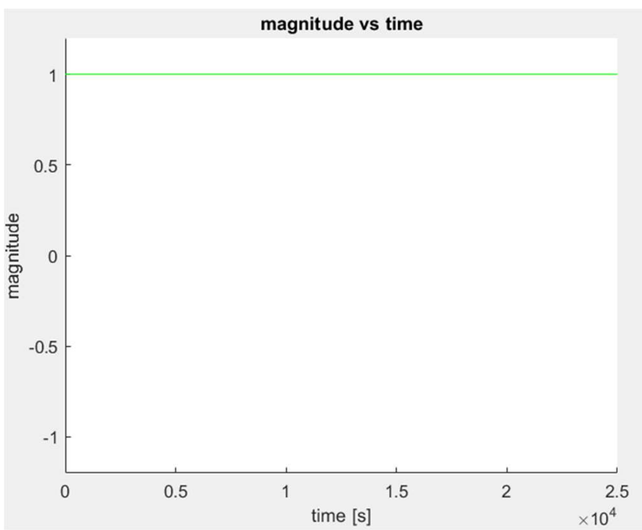
The resulting plots of the quaternion terms are given below:



In the end the plots of the magnitudes of the quaternions vs time are shown below with blue plot for Q3, red plot for Q4 part 1, and green plot for Q4 part 2.







As you can see, the fluctuation in magnitude is zero for Q4 due to normalizing and orthonormalizing in each iteration.