REPORT

Homework II

Attitude Determination and Control

UZB421E - 21265

Nadir Doğan

110180807

n = 48

Euler Angle Attitude Representation for Small Satellite Transformed from Quaternions

I will explain the required information in the code.

```
n=48;                %Student Number
Worbit=0.0011;       %The angular orbit velocity of satellite
Nt=3.6*10^-10;       %The disturbance torque acting on the satellite
N=54000;             %The iteration number
dt=0.1;              %The sample time
t(1)=0;              %Initial Time

%The moments of inertia of the satellite
Jx=2.1*10^-3;
Jy=2*10^-3;
Jz=1.9*10^-3;

%Initial data of the attitude angles (rad)
q1(1)=0.002*n;
q2(1)=0.001*n;
q3(1)=0.005*n;
q4(1)=sqrt(1-q1(1)^2-q2(1)^2-q3(1)^2);

%The initial data of the satellite's angular velocities
Wx(1)=0.0002+0.0001*n;
Wy(1)=0.0003+0.0001*n;
Wz(1)=0.0004+0.0001*n;


C=cell(N,1);   %cell contains all transformation matrices C
A=cell(N,1);   %cell contains all transformation matrices A


for i=1:N  %for loop to perform iteration

    %Time Increase of 0.1 s per Iteration
    t(i+1)=t(i)+dt;

    %The Angular Velocities Iteration
    Wx(i+1)=Wx(i)+(dt/Jx)*(Wz(i)*Wy(i)+Nt)*(Jy-Jz);
    Wy(i+1)=Wy(i)+(dt/Jy)*(Wx(i)*Wz(i)+Nt)*(Jz-Jx);
    Wz(i+1)=Wz(i)+(dt/Jz)*(Wx(i)*Wy(i)+Nt)*(Jx-Jy);


    %The Quaternions
    q1(i+1)=q1(i)-0.5*dt*(q2(i)*Wx(i)+q3(i)*Wy(i)+q4(i)*Wz(i));
    q2(i+1)=q2(i)+0.5*dt*(q1(i)*Wx(i)-q4(i)*Wy(i)+q3(i)*Wz(i));
```

```
q3(i+1)=q3(i)+0.5*dt*(q4(i)*Wx(i)+q1(i)*Wy(i)-q2(i)*Wz(i));
q4(i+1)=q4(i)-0.5*dt*(q3(i)*Wx(i)-q2(i)*Wy(i)-q1(i)*Wz(i));


%The Transformation Matrix
c11(i)=q1(i)^2-q2(i)^2-q3(i)^2+q4(i)^4;
c12(i)=2*(q1(i)*q2(i)+q3(i)*q4(i));
c13(i)=2*(q1(i)*q3(i)-q2(i)*q4(i));
c21(i)=2*(q1(i)*q2(i)-q3(i)*q4(i));
c22(i)=-q1(i)^2+q2(i)^2-q3(i)^3+q4(i)^2;
c23(i)=2*(q2(i)*q3(i)-q1(i)*q4(i));
c31(i)=2*(q1(i)*q3(i)+q2(i)*q4(i));
c32(i)=2*(q2(i)*q3(i)-q1(i)*q4(i));
c33(i)=-q1(i)^2-q2(i)^2+q3(i)^2+q4(i)^2;

C{i}=[c11(i),c12(i),c13(i);c21(i),c22(i),c23(i);c31(i),c32(i),c33(i)];
```

There is nothing different from the first homework so far, in the same way, I put the given data into code.

The Euler angles expressed in terms of quaternions below.
ϕ is the roll angle, reference x- axis
θ is the pitch angle, reference y- axis
ψ is the yaw angle, reference z- axis

$$\phi = \arctan\left(\left(2\left(q_2 q_3 + q_1 q_4\right)\right) / \left(1 - 2\left(q_1^2 + q_2^2\right)\right)\right)$$

$$\theta = \arcsin\left(2\left[q_4 q_2 - q_1 q_3\right]\right)$$

$$\psi = \arctan\left(\left(2\left(q_4 q_3 + q_1 q_2\right)\right) / \left(1 - 2\left(q_2^2 + q_3^2\right)\right)\right)$$

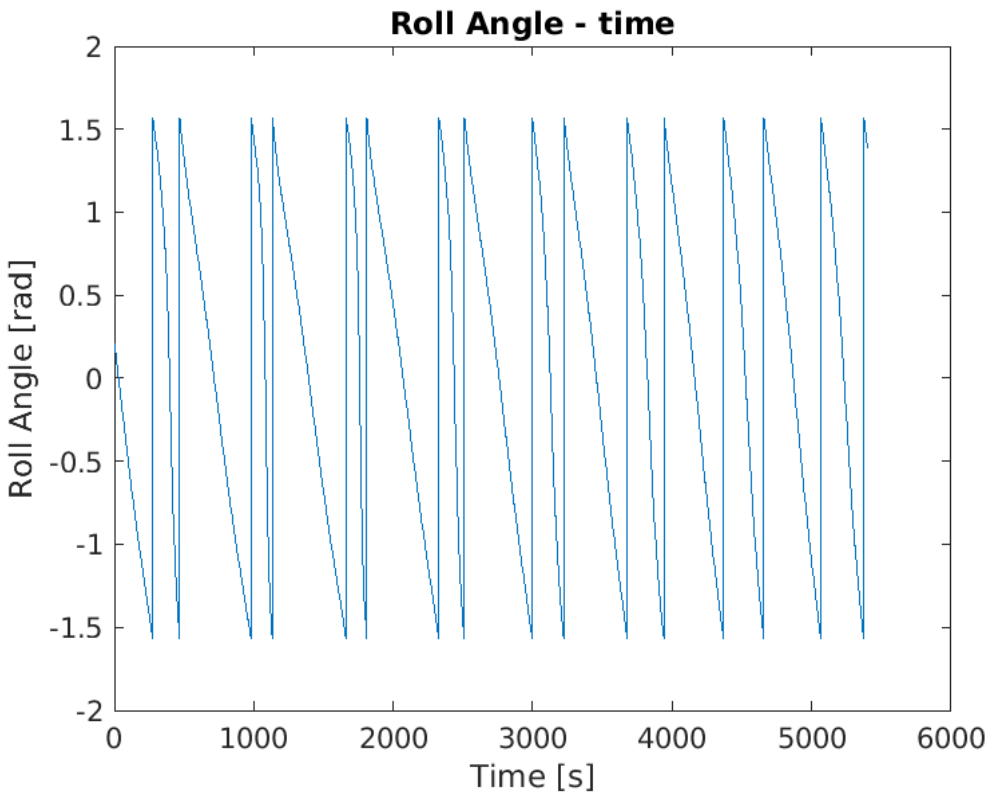
```
%The Euler Angles
%Roll Angle
phi(i)=atan((2*(q2(i)*q3(i)+q1(i)*q4(i)))/(1-2*(q1(i)^2+q2(i)^2)));
%Pitch Angle
tetha(i)=asin(2*(q4(i)*q2(i)-q1(i)*q3(i)));
%Yaw Angle
psi(i)=atan((2*(q4(i)*q3(i)+q1(i)*q2(i)))/(1-2*(q2(i)^2+q3(i)^2)));
```

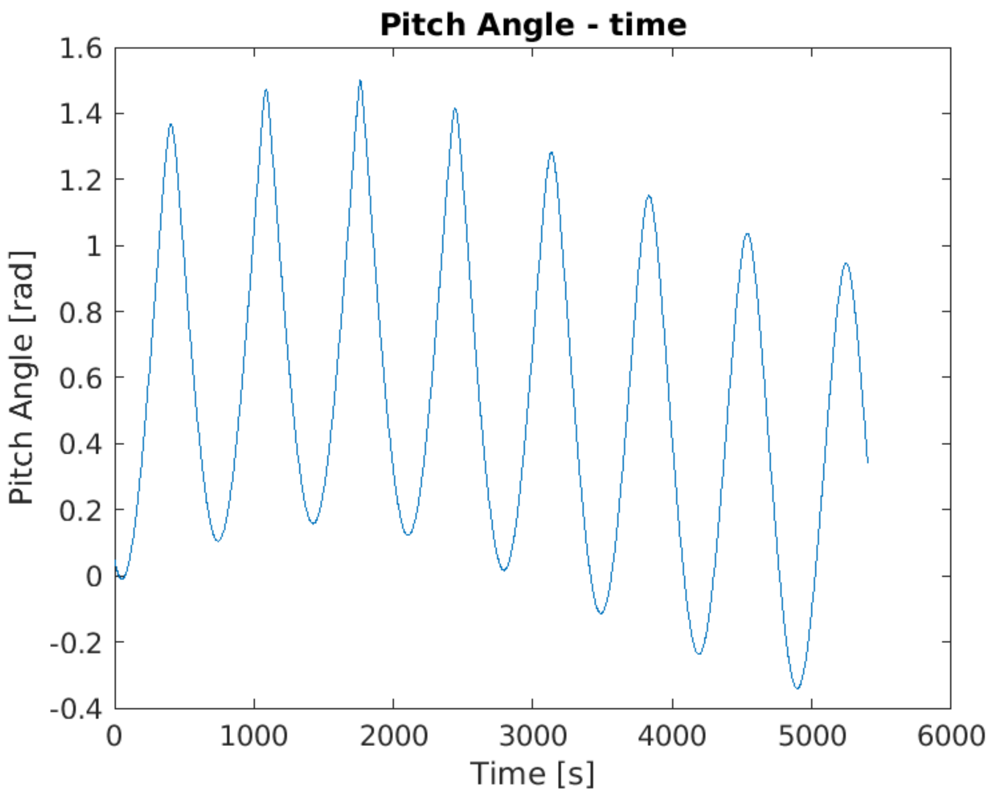
There is given transformation matrix below.

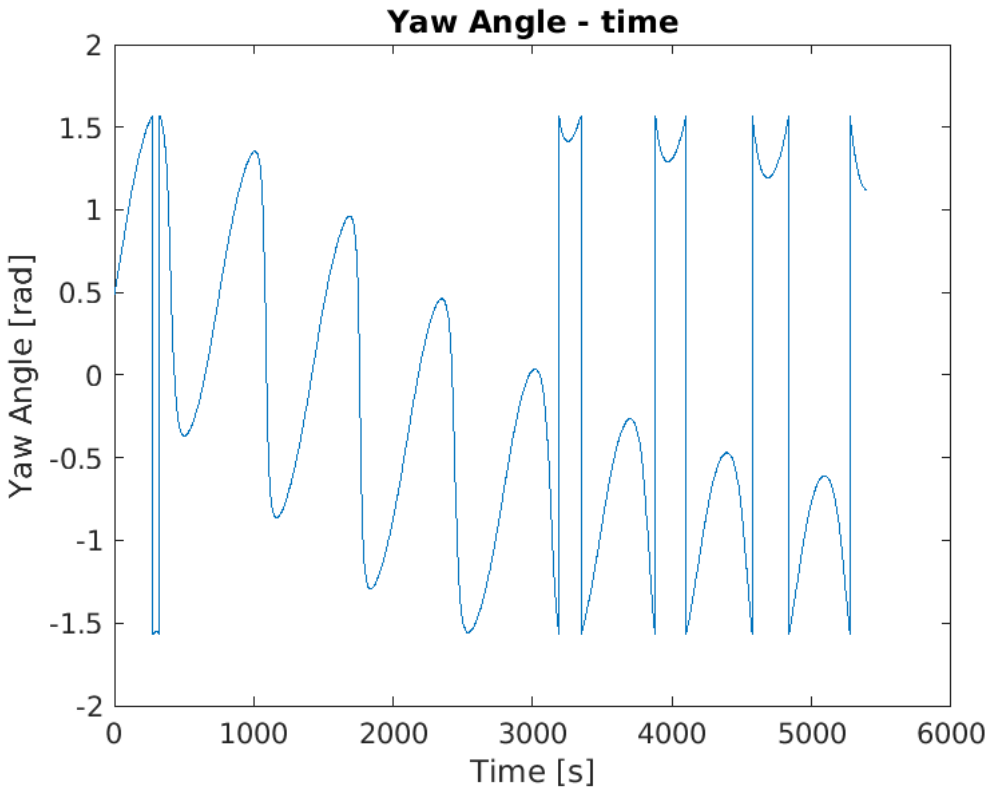
$$A=\begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ -c(\varphi)s(\psi)+s(\varphi)s(\theta)c(\psi) & c(\varphi)c(\psi)+s(\varphi)s(\theta)s(\psi) & s(\varphi)c(\theta) \\ s(\varphi)s(\psi)+c(\varphi)s(\theta)c(\psi) & -s(\varphi)c(\psi)+c(\varphi)s(\theta)s(\psi) & c(\varphi)c(\theta) \end{bmatrix}.$$

```
    %The Transformation Matrix Euler angle
    a11(i)=cos(tetha(i))*cos(psi(i));
    a12(i)=cos(tetha(i))*sin(psi(i));
    a13(i)=-sin(tetha(i));
    a21(i)=-cos(phi(i))*sin(psi(i))+sin(phi(i))*sin(tetha(i))*cos(psi(i));
    a22(i)=cos(phi(i))*cos(psi(i))+sin(phi(i))*sin(tetha(i))*sin(psi(i));
    a23(i)=sin(phi(i))*cos(tetha(i));
    a31(i)=sin(phi(i))*sin(psi(i))+cos(phi(i))*sin(tetha(i))*cos(psi(i));
    a32(i)=-sin(phi(i))*sin(psi(i))+cos(phi(i))*sin(tetha(i))*sin(psi(i));
    a33(i)=cos(phi(i))*cos(tetha(i));

    A{i}=[a11(i),a12(i),a13(i);a21(i),a22(i),a23(i);a31(i),a32(i),a33(i)];
end

%Euler Angles Graphs
figure
plot(t(1:54000),phi);
title('Roll Angle - time')
xlabel('Time [s]')
ylabel('Roll Angle [rad]')
figure
plot(t(1:54000),tetha);
title('Pitch Angle - time')
xlabel('Time [s]')
ylabel('Pitch Angle [rad]')
figure
plot(t(1:54000),psi);
title('Yaw Angle - time')
xlabel('Time [s]')
ylabel('Yaw Angle [rad]')
```

Graphs of Euler Angles






Yaw Angle - time
Yaw Angle [rad]
Time [s]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1000
2000
3000
4000
5000
6000