# Quaternion Attitude Representation for Small Satellite Transformed from Euler Angles

## Attitude Dynamics and Control
## Project Report

**Serkan Burak Örs***

*orss19@itu.edu.tr

# Contents

# 1 Quaternion Attitude Representation for Small Satellite Transformed from Euler Angles

The subject of the homework is to characterize the attitude dynamics of the Low Earth Orbit Satellite in terms of quaternions. To investigate the behaviors of satellite, the graphs will include the propagation of quaternions in time. The mathematical model of the satellite's rotational motion was written in terms of the Euler angles and its angular velocities and it was solved by an iterative approach respect to the initial conditions of these parameters. One of the commercial engineering software will be used to plot the figures.
The quaternions

$$q = q_1 + q_2 i + q_3 j + q_4 k \tag{1}$$

- The quaternions expressed in terms of Euler angles

$$q_1 = cos\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \tag{2}$$
$$q_2 = sin\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} - cos\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \tag{3}$$
$$q_3 = cos\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} \tag{4}$$
$$q_4 = cos\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} + sin\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} \tag{5}$$

- The propagation of quaternions with time

$$\vec{\dot{q}} = 0.5 \cdot \vec{q} X \vec{P_n^b} \tag{6}$$
$$\vec{P_n^b} = [0, \vec{\omega_n^b}]^T \tag{7}$$

$$\vec{\dot{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{8}$$

$$\vec{\dot{q_1}} = -0.5(q_2\omega_x + q_3\omega_y + q_4\omega_z) \tag{9}$$
$$\vec{\dot{q_2}} = 0.5(q_1\omega_x - q_4\omega_y + q_3\omega_z) \tag{10}$$
$$\vec{\dot{q_3}} = 0.5(q_4\omega_x + q_1\omega_y - q_2\omega_z) \tag{11}$$
$$\vec{\dot{q_4}} = -0.5(q_3\omega_x - q_2\omega_y - q_1\omega_z) \tag{12}$$

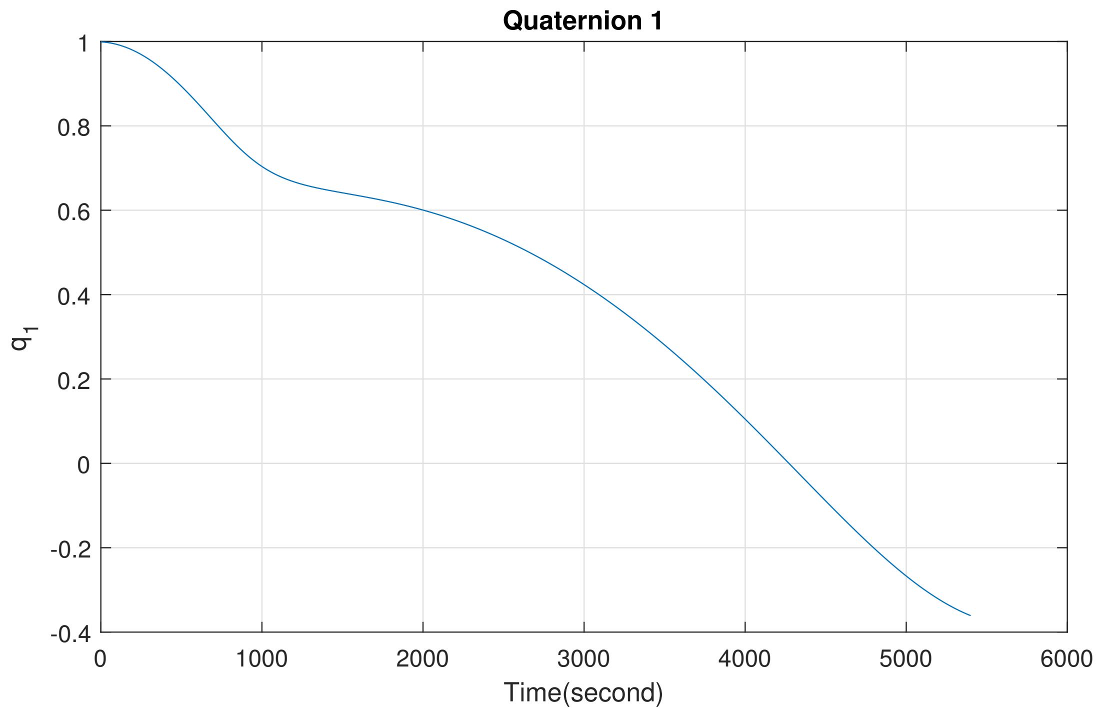
Quaternions, quaternion rates should be found and 8 figures (quaternions, quaternion rates) should be plotted before and after **nearest neighboring method** is applied.

**Nearest Neighboring Method**: Because of the nature of the quaternions, any spin expression can be expressed in q or - q , and these two expressions have the same meaning. Therefore, the founded values are close to the desired values. However, in order to be able to perform error analysis and to avoid errors in the subsequent steps of simulation, the uncertainty of the sign must be corrected. In order to overcome the problem, an approach should be made considering that the value in each step is as close to the previous value as possible.

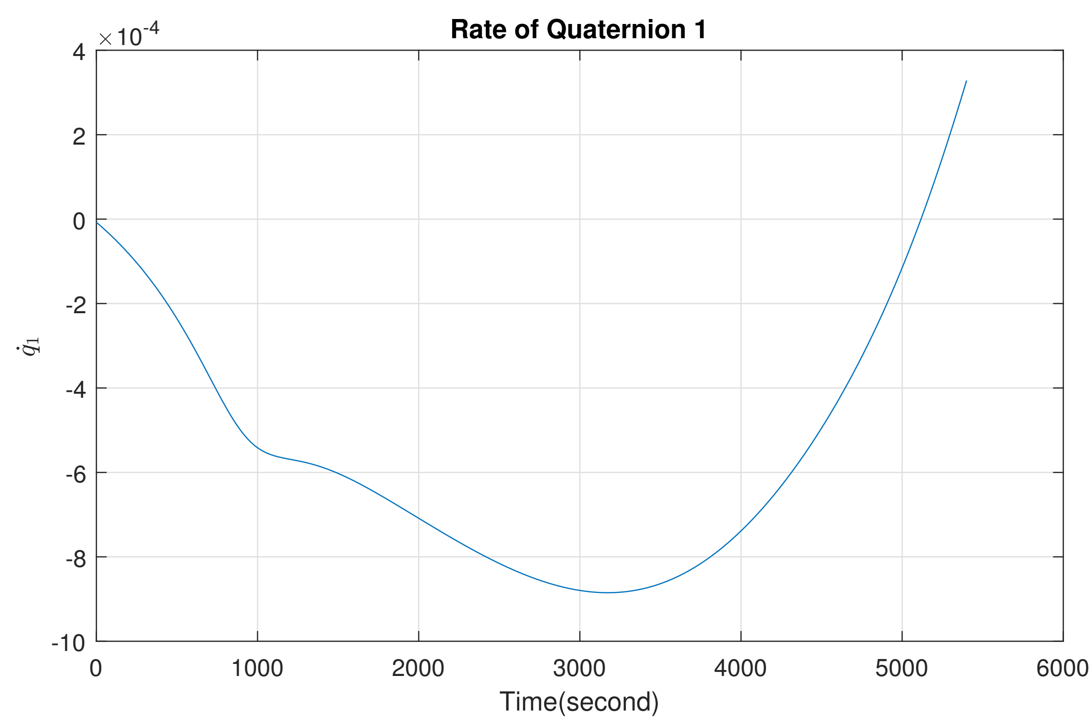
By following the steps below, the nearest neighboring method can be applied;
- the quaternion values in steps i and i +1 are considered,
- q(i) and q (i+1) are multiplied as scalar. If the multiplication result is negative, the value of q(i +1) is changed multiplying by -1. If the value is positive, the value is left unchanged without any operation.$\dot{q}_1$

**Before applying nearest neighboring method:**

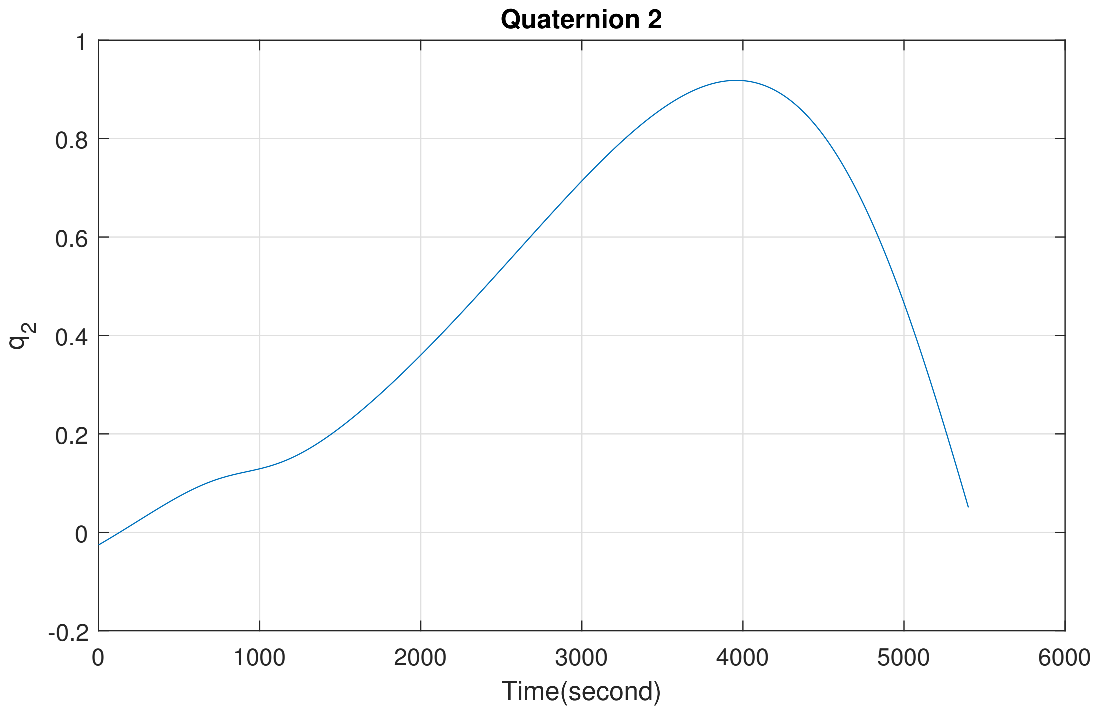
Graph of $1^{st}$ quaternion($q_1$) before applying nearest neighboring method:



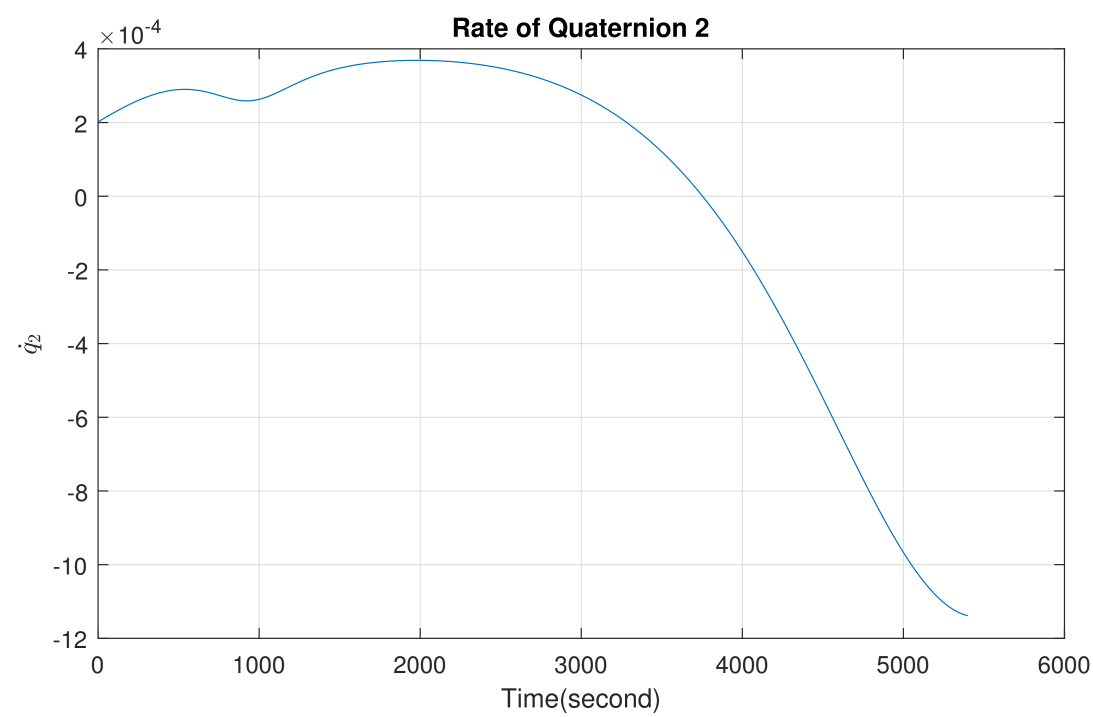

Graph of rate of $1^{st}$ quaternion($\dot{q}_1$) before applying nearest neighboring method:



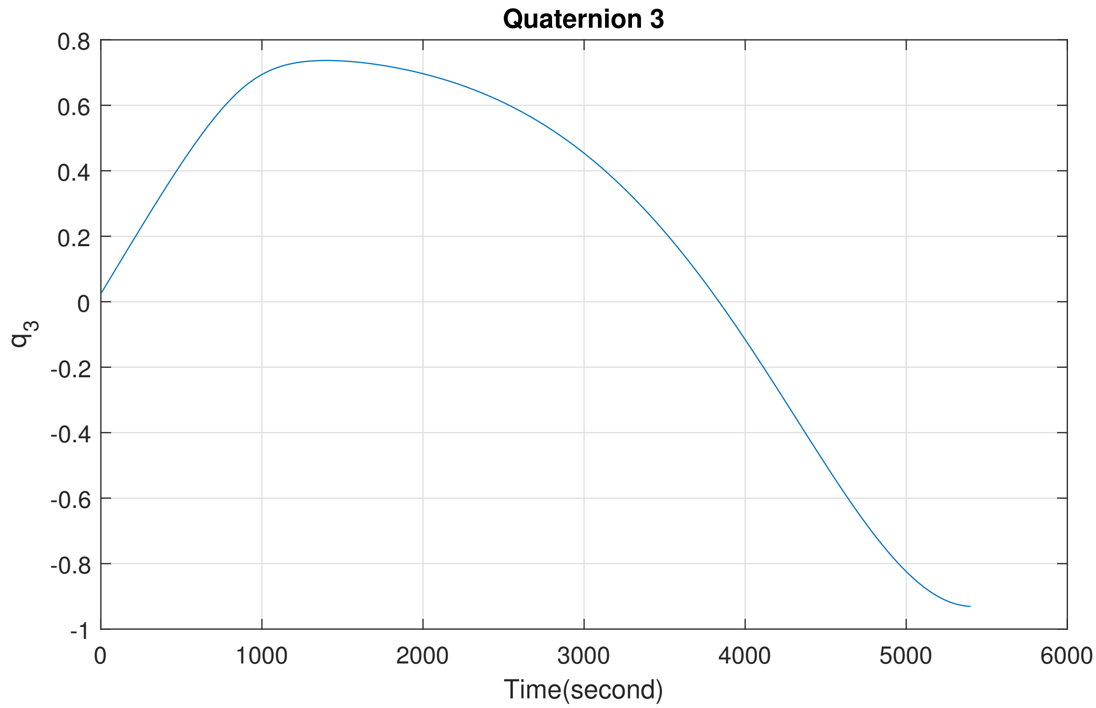
Graph of $2^{nd}$ quaternion($q_2$) before applying nearest neighboring method:



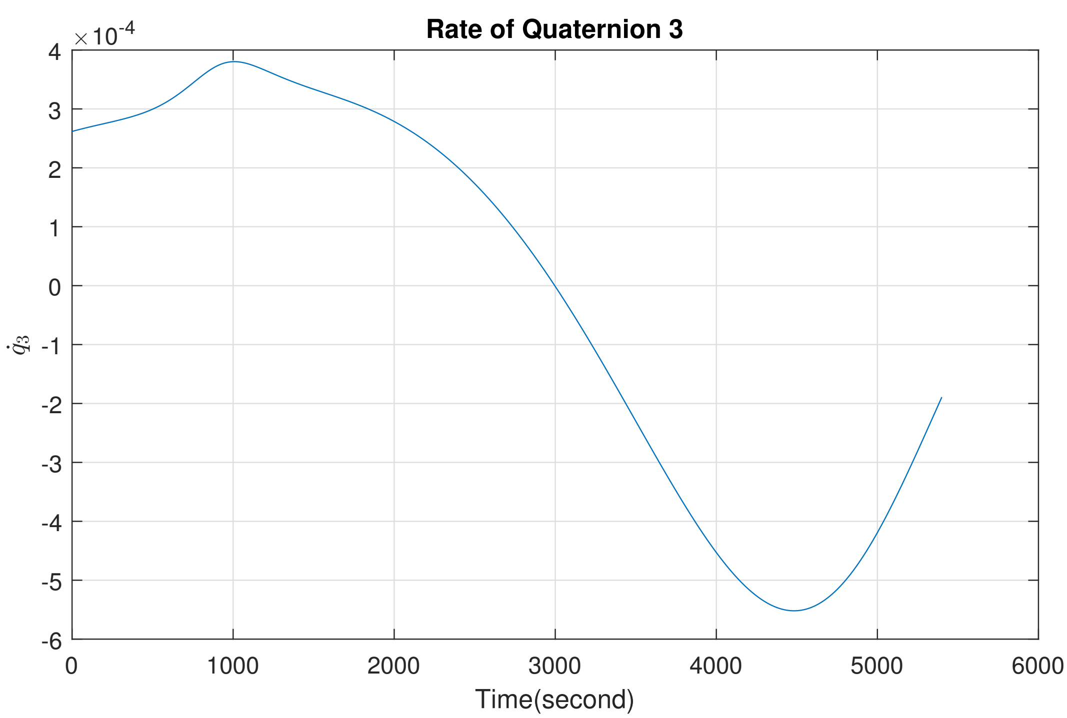

Graph of rate of $2^{nd}$ quaternion($\dot{q}_2$) before applying nearest neighboring method:



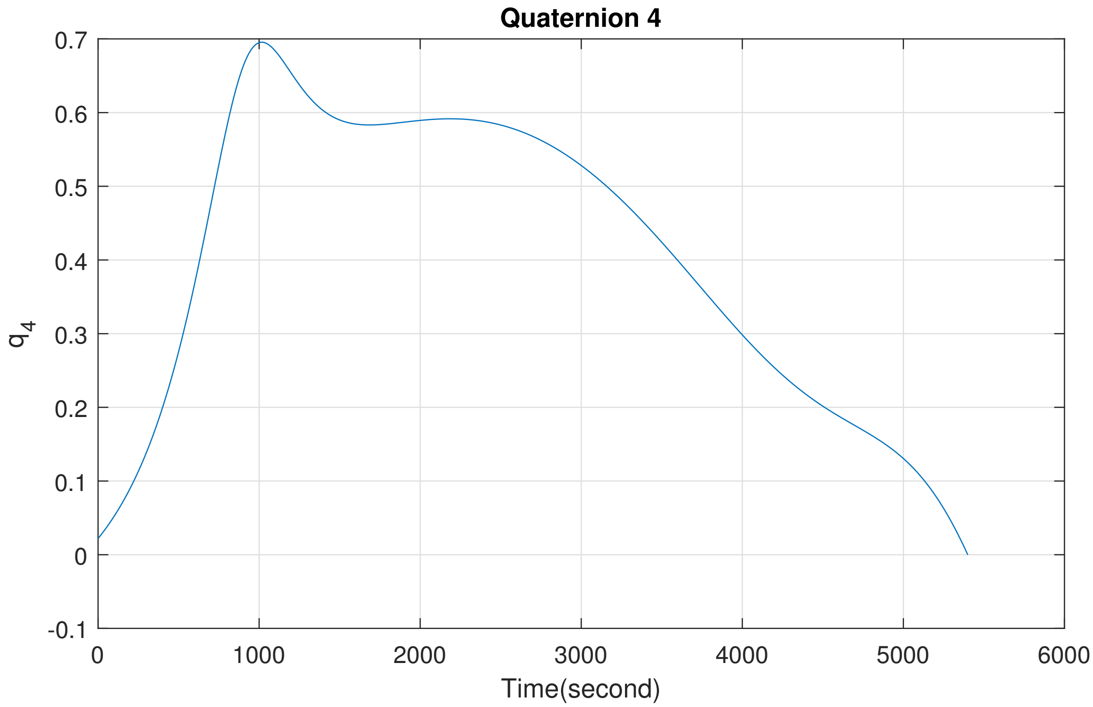
Graph of $3^{rd}$ quaternion($q_3$) before applying nearest neighboring method:



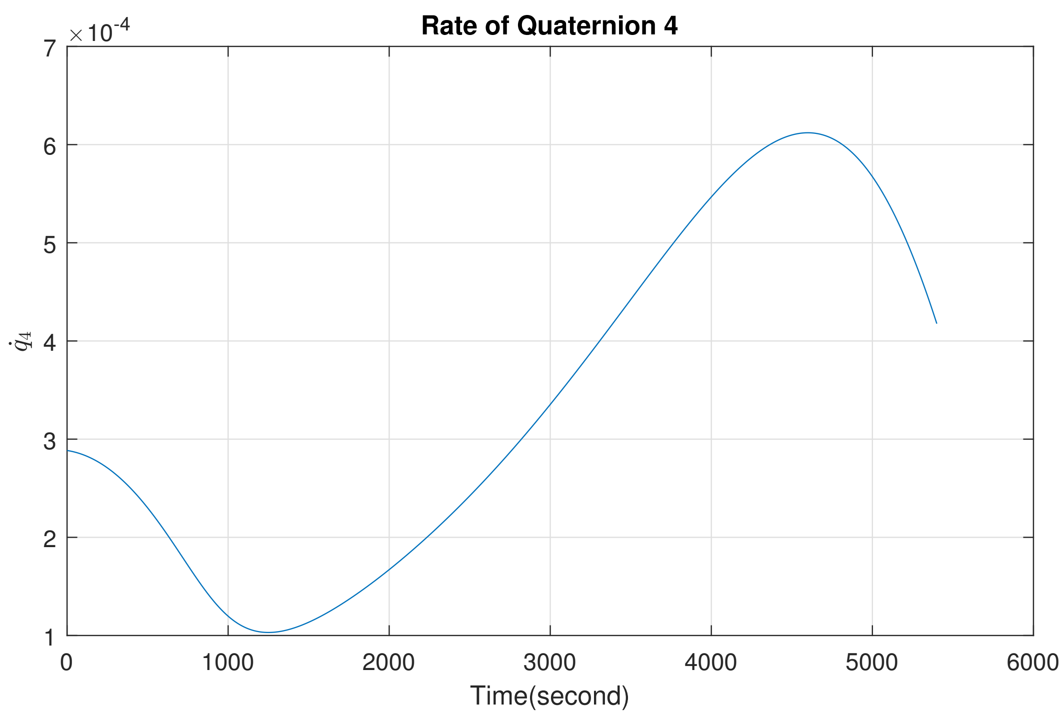

Graph of rate of $3^{rd}$ quaternion($\dot{q}_3$) before applying nearest neighboring method:



Graph of $4^{th}$ quaternion($q_4$) before applying nearest neighboring method:

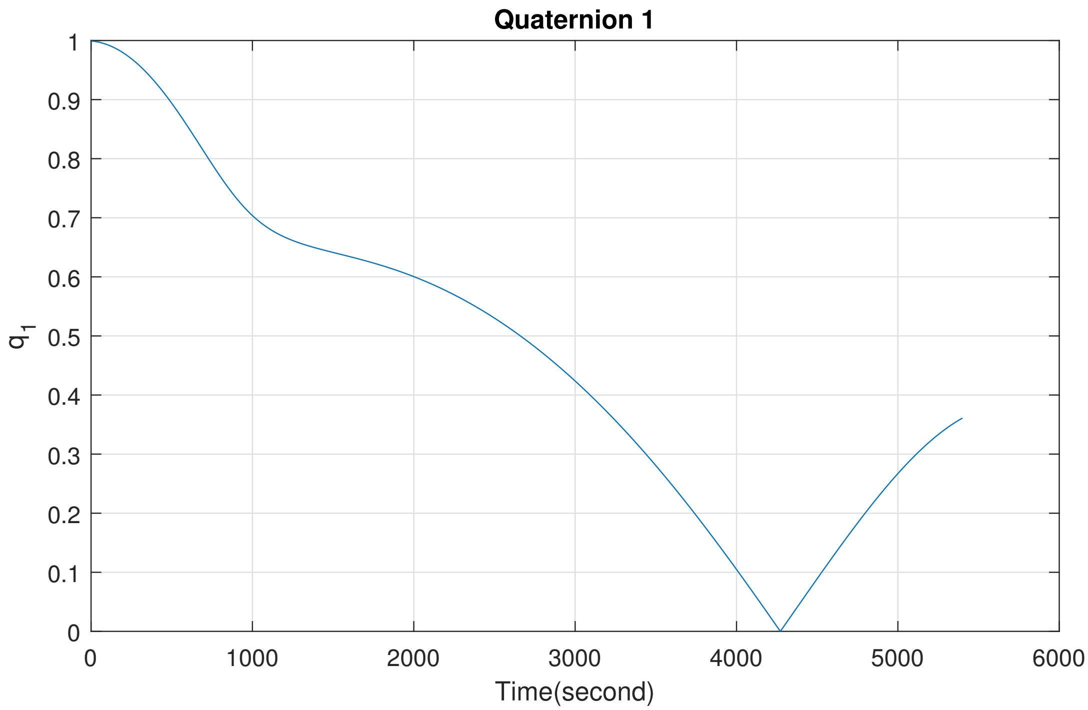



Graph of rate of $4^{th}$ quaternion($\dot{q}_4$) before applying nearest neighboring method:

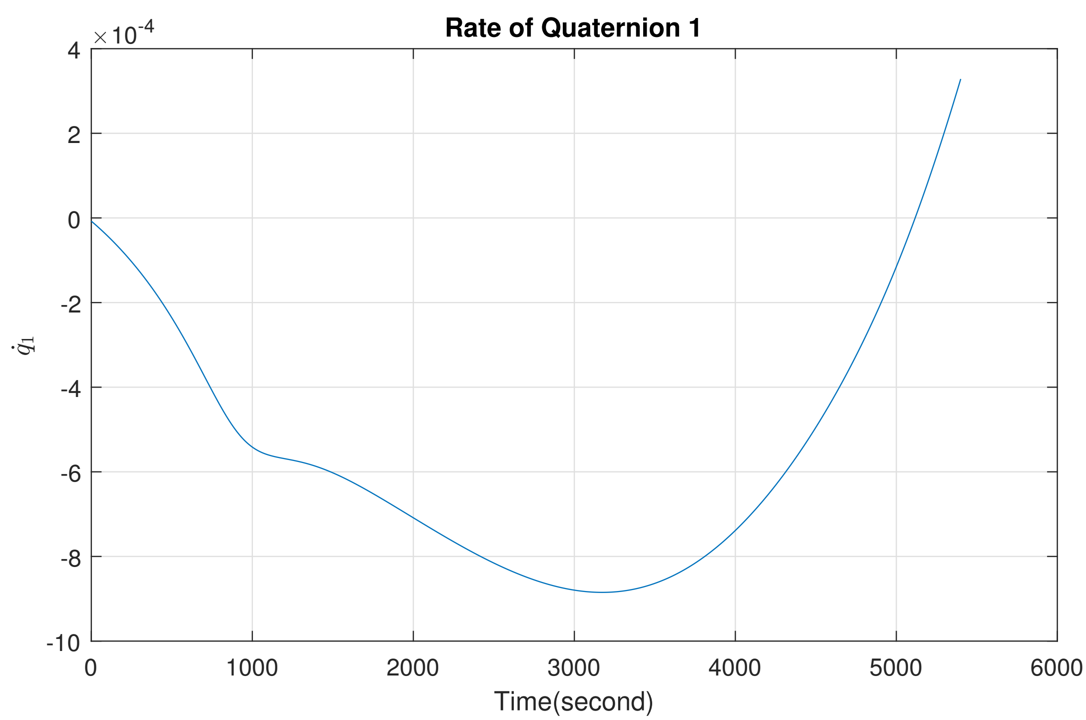


**After applying nearest neighboring method:**

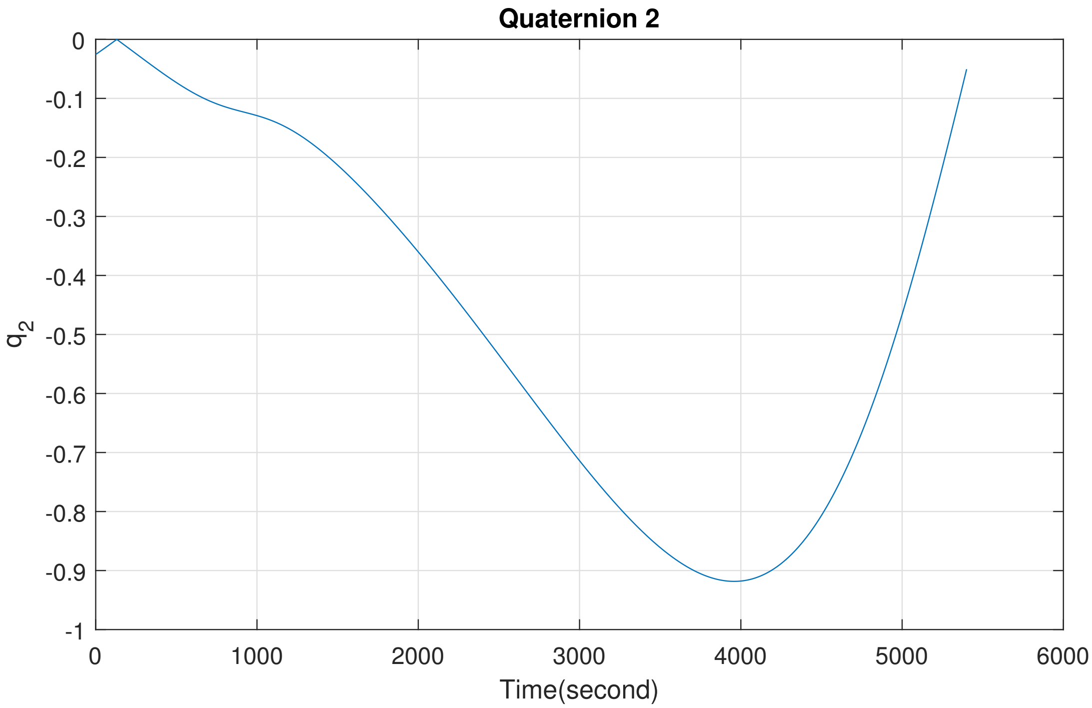
Graph of $1^{st}$ quaternion($q_1$) after applying nearest neighboring method:



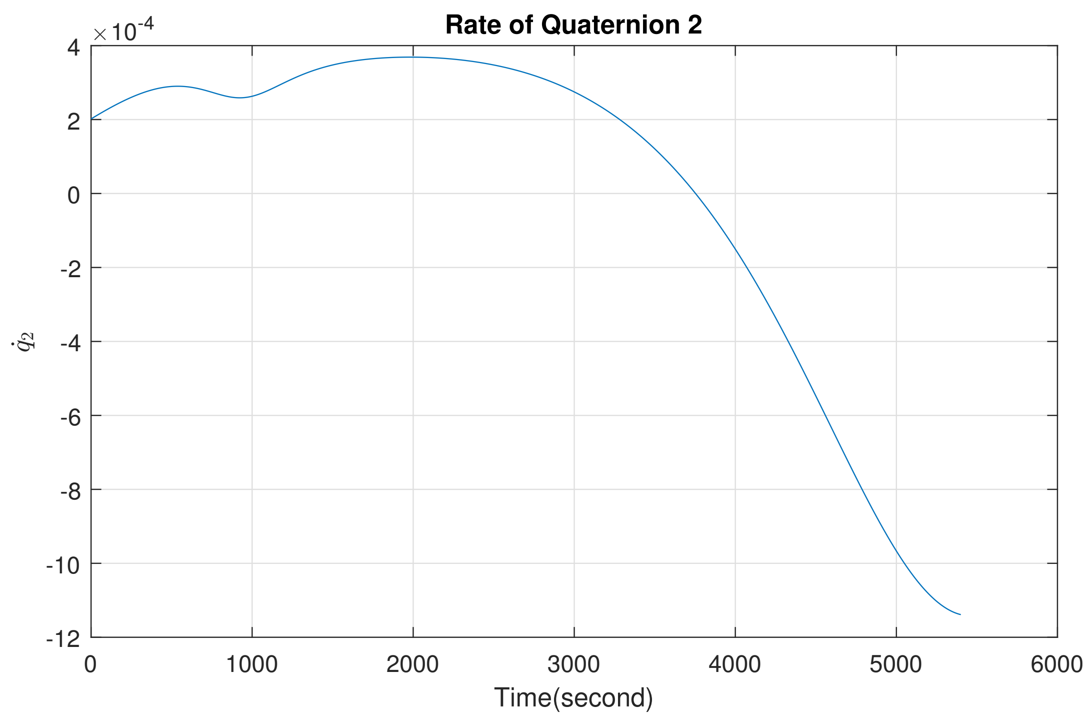

Graph of rate of $1^{st}$ quaternion($\dot{q}_1$) after applying nearest neighboring method:



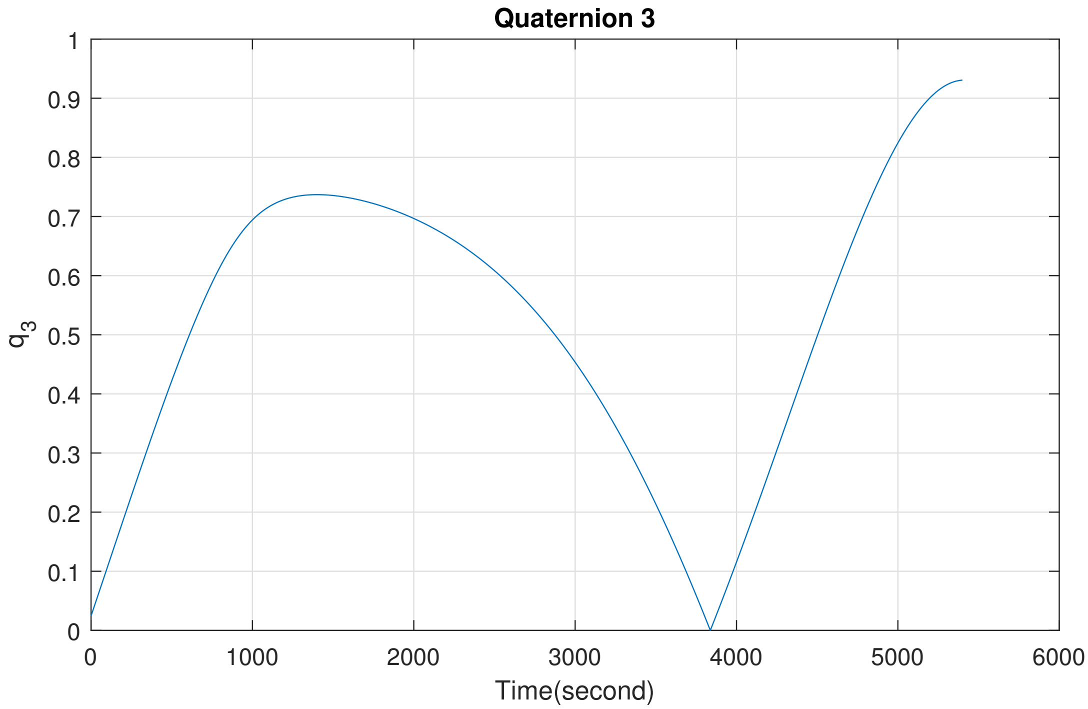
Graph of $2^{nd}$ quaternion($q_2$) after applying nearest neighboring method:



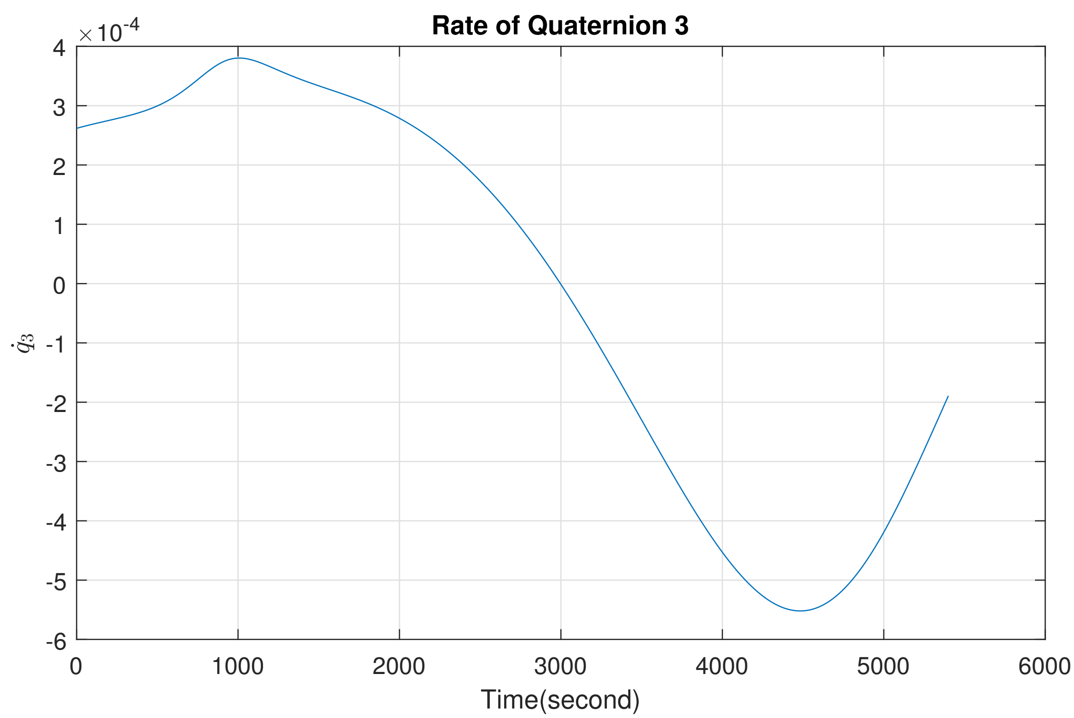

Graph of rate of $2^{nd}$ quaternion($\dot{q}_2$) after applying nearest neighboring method:



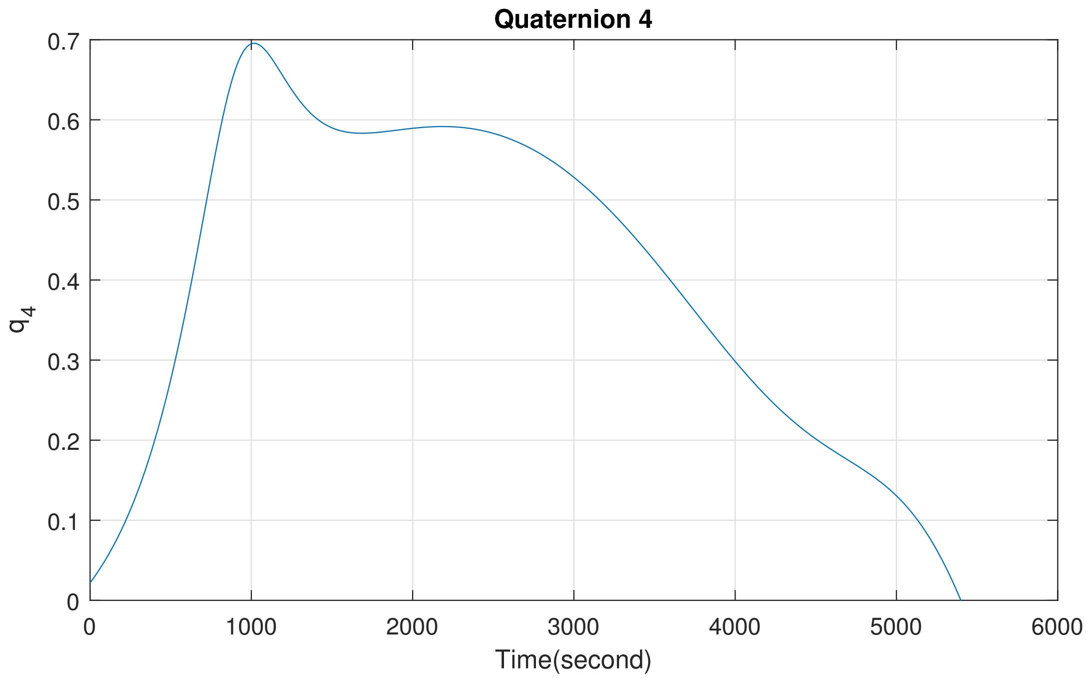
Graph of $3^{rd}$ quaternion($q_3$) after applying nearest neighboring method:



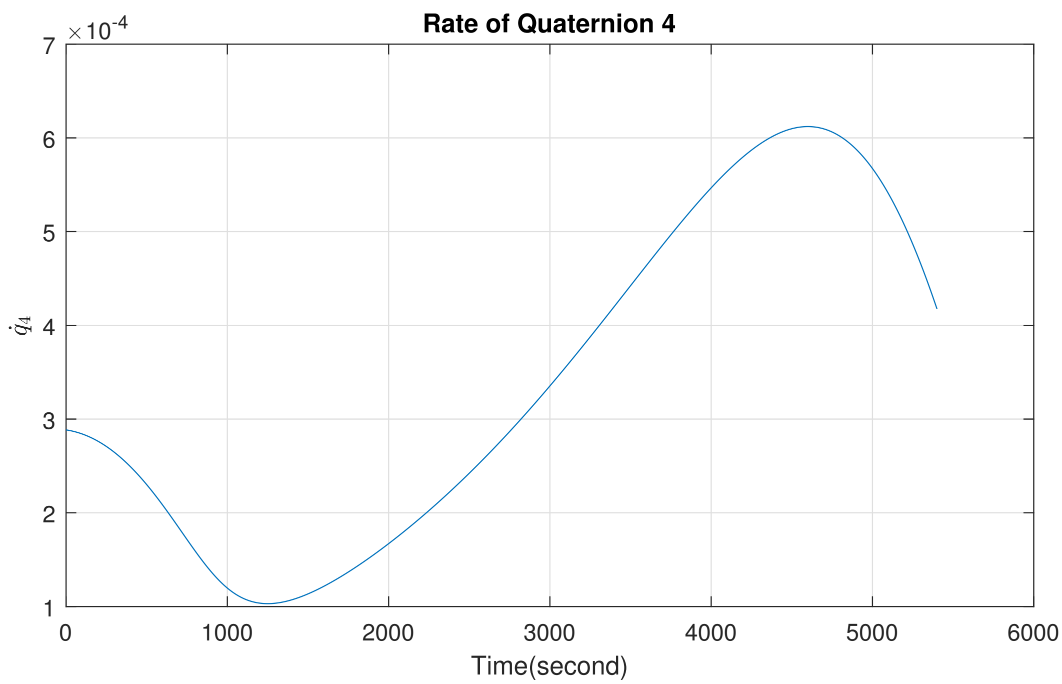

Graph of rate of $3^{rd}$ quaternion($\dot{q}_3$) after applying nearest neighboring method:



Graph of $4^{th}$ quaternion($q_4$) after applying nearest neighboring method:




Graph of rate of $4^{th}$ quaternion($\dot{q}_4$) after applying nearest neighboring method:




**Conclusion:** From the graphs, we can deduce that nearest neighboring method only affects quaternions, not quaternion rates.

# 2 Appendix: MATLAB Code

```
clc;clear, close all;

n = 20; %sequence number

%initial data of the attitude angles(rad)
phi = -0.01 - 0.002 * n;
theta = 0.01 + 0.002 * n;
psi = 0.005 + 0.002 * n;

%initial data of the angular velocities of the satellite
wX = 0.0002 + 0.00001 * n;
wY = 0.0003 + 0.00001 * n;
wZ = 0.0004 + 0.00001 * n;
w = [wX;wY;wZ];

%initial moments of inertia of the satellite(m^4)
Jx = 2.1*(10^(-3));
Jy = 2*(10^(-3));
Jz = 1.9*(10^(-3));

w_orbit = 0.0011; %The angular orbit velocity of satellite(rad/s)
N_T = 3.6*(10^(-10)); %The disturbance torque acting on the satellite(Nm)
dt = 0.1; %the sample time(s)
N = 54000; %iteration number
for k = 1:N
%the angular velocities
w(1) = w(1) + ((dt/Jx)*(Jy - Jz)*w(3)*w(2)) + ((dt/Jx)*N_T);
w(2) = w(2) + ((dt/Jy)*(Jz - Jx)*w(1)*w(3)) + ((dt/Jy)*N_T);
w(3) = w(3) + ((dt/Jz)*(Jx - Jy)*w(1)*w(2)) + ((dt/Jz)*N_T);

%the euler angles
phi = phi + dt*((((w(2)*sin(phi)) + (w(3)*cos(phi)))*tan(theta)) + w(1));
theta = theta + dt*((w(2)*cos(phi)) - (w(3)*sin(phi)) + (w_orbit));
psi = psi + dt*(((w(2)*sin(phi)) + (w(3)*cos(phi)))*sec(theta));

%quaternions
q1 = cos(phi/2)*cos(theta/2)*cos(psi/2)+sin(phi/2)*sin(theta/2)*sin(psi/2);
q2 = sin(phi/2)*cos(theta/2)*cos(psi/2)-cos(phi/2)*sin(theta/2)*sin(psi/2);
q3 = cos(phi/2)*sin(theta/2)*cos(psi/2)+sin(phi/2)*cos(theta/2)*sin(psi/2);
```

```
q4 = cos(phi/2)*cos(theta/2)*sin(psi/2)+sin(phi/2)*sin(theta/2)*cos(psi/2);
q = [q1;q2;q3;q4];

%quaternion rates
q1Dot = -0.5*((q2*w(1)) + (q3*w(2)) + (q4*w(3)));
q2Dot = 0.5*((q1*w(1)) - (q4*w(2)) + (q3*w(3)));
q3Dot = 0.5*((q4*w(1)) + (q1*w(2)) - (q2*w(3)));
q4Dot = -0.5*((q3*w(1)) - (q2*w(2)) - (q1*w(3)));
qDot = [q1Dot;q2Dot;q3Dot;q4Dot];

Q(:,k) = q;
Qdot (:,k) = qDot;
end

t = 0:dt:(N-1)*dt; %constructing time axis

%plotting before applying nearest neighboring method
figure(1);
plot(t,Q(1,:));
title('Quaternion 1');
xlabel('Time(second)');
ylabel('q_{1}');
grid on;

figure(2);
plot(t,Q(2,:));
title('Quaternion 2');
xlabel('Time(second)');
ylabel('q_{2}');
grid on;

figure(3);
plot(t,Q(3,:));
title('Quaternion 3');
xlabel('Time(second)');
ylabel('q_{3}');
grid on;

figure(4);
plot(t,Q(4,:));
title('Quaternion 4');
xlabel('Time(second)');
ylabel('q_{4}');
```

```
grid on;

figure(5);
plot(t,Qdot(1,:));
title('Rate of Quaternion 1');
xlabel('Time(second)');
ylabel('$\dot{q}_{1}$','interpreter','latex');
grid on;

figure(6);
plot(t,Qdot(2,:));
title('Rate of Quaternion 2');
xlabel('Time(second)');
ylabel('$\dot{q}_{2}$','interpreter','latex');
grid on;

figure(7);
plot(t,Qdot(3,:));
title('Rate of Quaternion 3');
xlabel('Time(second)');
ylabel('$\dot{q}_{3}$','interpreter','latex');
grid on;

figure(8);
plot(t,Qdot(4,:));
title('Rate of Quaternion 4');
xlabel('Time(second)');
ylabel('$\dot{q}_{4}$','interpreter','latex');
grid on;

%nearest neighboring method
for j = 1:1:4
for i = 1:N-1
if (Q(j,i)*Q(j,i+1))<0
Q(j,i+1) = -1*Q(j,i+1);
else
Q(j,i+1) = Q(j,i+1);
end
end
end

%plotting after applying nearest neighboring method

```

```
figure(9);
plot(t,Q(1,:));
title('Quaternion 1');
xlabel('Time(second)');
ylabel('q_{1}');
grid on;

figure(10);
plot(t,Q(2,:));
title('Quaternion 2');
xlabel('Time(second)');
ylabel('q_{2}');
grid on;

figure(11);
plot(t,Q(3,:));
title('Quaternion 3');
xlabel('Time(second)');
ylabel('q_{3}');
grid on;

figure(12);
plot(t,Q(4,:));
title('Quaternion 4');
xlabel('Time(second)');
ylabel('q_{4}');
grid on;

figure(13);
plot(t,Qdot(1,:));
title('Rate of Quaternion 1');
xlabel('Time(second)');
ylabel('$\dot{q}_{1}$','interpreter','latex');
grid on;

figure(14);
plot(t,Qdot(2,:));
title('Rate of Quaternion 2');
xlabel('Time(second)');
ylabel('$\dot{q}_{2}$','interpreter','latex');
grid on;

figure(15);
```

```
plot(t,Qdot(3,:));
title('Rate of Quaternion 3');
xlabel('Time(second)');
ylabel('$\dot{q}_{3}$','interpreter','latex');
grid on;

figure(16);
plot(t,Qdot(4,:));
title('Rate of Quaternion 4');
xlabel('Time(second)');
ylabel('$\dot{q}_{4}$','interpreter','latex');
grid on;
```

# 3 References

[1] Prof. Dr. Cengiz Hacızade, *Istanbul Technical University UCK421E Lecture Notes*, 2021.

[2] J.R.Wertz., *Space Attitude Determination and Control*, D.Reidel Publishing Company, Dordrecht, Holland, 2002.

[3] Hajiyev, C., & Soken, H.E., *Fault Tolerant Attittude Estimation for Small Satellites*, $1^{st}$ Ed., CRC Press, 2021.