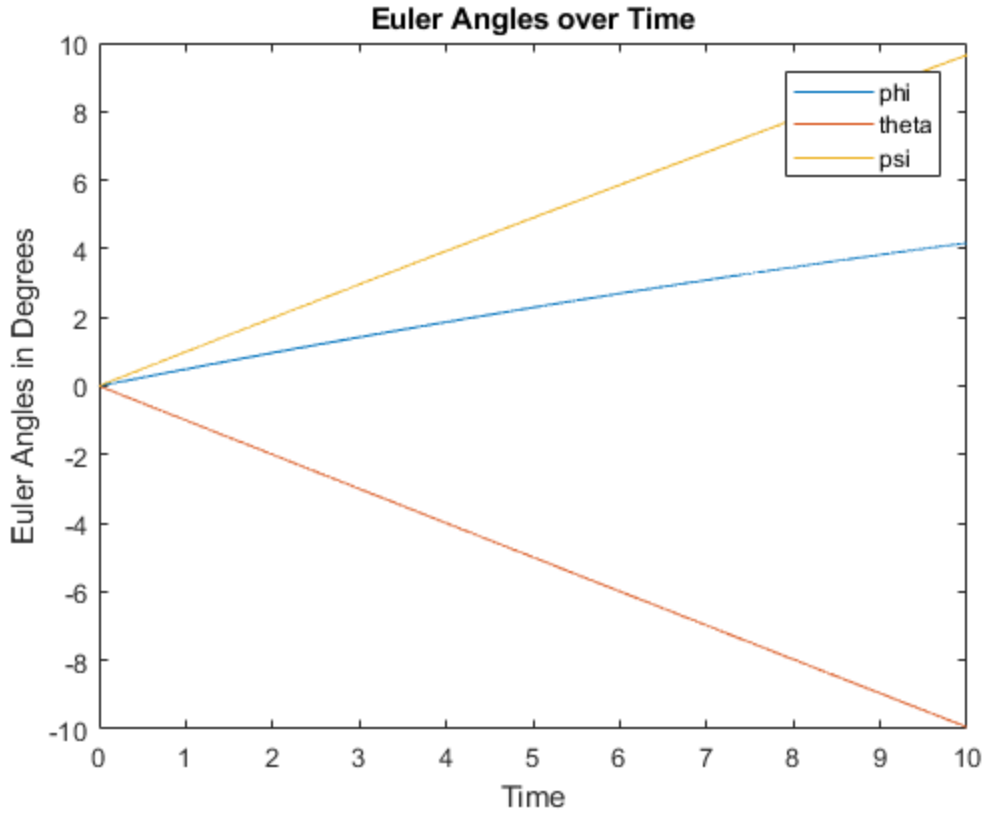
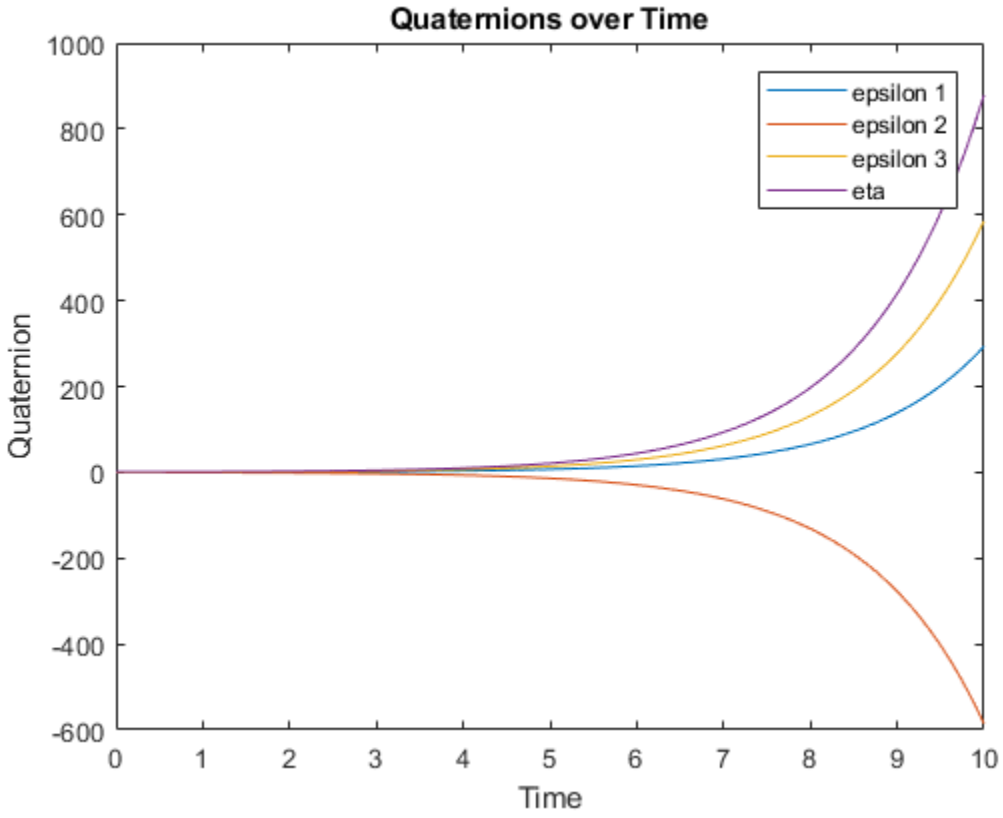
```
w21 = [ .5 ; -1 ; 1 ] ;
phi = 0 ;
theta = 0 ;
psi = 0 ;

disp( 'a' )
C21 = EulerAngle321( phi , theta , psi ) ; % Rotation when euler
 angles are 0
[ eta , eps ] = Quaternion( C21 ) ;
disp( 'Quaternions when Euler angles are 0' )
disp( 'eta' )
disp( eta )
disp( 'epsilon' )
disp( eps )

disp( 'b' )
euler = [ phi ; theta ; psi ] ;
tstep = 10 / 10000 ;
t = 0 ;
ii = 1 ;
while t(ii)< 10-tstep
    rates(:,ii) = EulerRates( euler(:,ii) , w21 ) ;
    euler(:,ii+1 ) = euler(:,ii) + rates(:,ii)*tstep ;
    t(ii+1) = t(ii) + tstep ;
    ii = ii + 1 ;
end
plot( t , euler(1,:) , t , euler(2,:) , t , euler(3,:) )
title( 'Euler Angles over Time' )
ylabel( 'Euler Angles in Degrees' )
xlabel( 'Time' )
legend( 'phi' , 'theta' , 'psi' )

disp( 'c' )
tstep = 10 / 1000 ;
t = 0 ;
ii = 1 ;
while t(ii)< 10-tstep
    [ epsdot , etadot ] = QRates( eps(:,ii) , eta(ii) , w21 ) ;
    eta(ii+1) = eta(ii) + etadot*tstep ;
    eps(:,ii+1) = eps(:,ii) + epsdot*tstep ;
    t(ii+1) = t(ii) + tstep ;
    ii = ii + 1 ;
end
figure
plot( t , eps(1,:) , t , eps(2,:) , t , eps(3,:) , t , eta )
title( 'Quaternions over Time' )
ylabel( 'Quaternion' )
xlabel( 'Time' )
legend( 'epsilon 1' , 'epsilon 2' , 'epsilon 3' , 'eta' )


function [ RotationMatrix ] = EulerAngle321( phi , theta , psi )
```

```
    Cx = [ 1 0 0 ; 0 cosd(phi) sind(phi) ; 0 -sind(phi) cosd(phi) ] ;
    Cy = [ cosd(theta) 0 -sind(theta) ; 0 1 0 ; sind(theta) 0
 cosd(theta) ] ;
    Cz = [ cosd(psi) sind(psi) 0 ; -sind(psi) cosd(psi) 0 ; 0 0 1 ] ;
    RotationMatrix = Cx * Cy * Cz ;
end

function [ eta , eps ] = Quaternion( RotationMatrix )
    C = RotationMatrix ;
    eta = .5*sqrt( 1+trace(C) ) ;
    eps = zeros( 3 , 1 ) ;
    if eta ~= 0
        eps(1) = (C(2,3)-C(3,2))/4*eta ;
        eps(2) = (C(3,1)-C(1,3))/4*eta ;
        eps(1) = (C(1,2)-C(2,1))/4*eta ;
    else
        eps(1) = sqrt( (C(1,1)+1)/2 ) ;
        eps(2) = sqrt( (C(2,2)+1)/2 ) ;
        eps(3) = sqrt( (C(3,3)+1)/2 ) ;
    end
end

function [ Rates ] = EulerRates( euler , w )
    Rates = (1/cosd(euler(2)))*[ cosd(euler(2))
 sind(euler(1))*sind(euler(2)) cosd(euler(1))*sind(euler(2)) ; 0
 cosd(euler(1))*cosd(euler(2)) -sind(euler(1))*sind(euler(2)) ; 0
 sind(euler(1)) cosd(euler(1)) ]*w ;
end

function [ epsdot , etadot ] = QRates( eps , eta , w )
    epscross = [ 0 -eps(3) eps(2) ; eps(3) 0 -eps(1) ; -eps(2) eps(1)
 0 ];
    epsdot = .5*(eta*eye(3)+epscross)*w ;
    etadot = .5*eps'*w ;
end

a
Quaternions when Euler angles are 0
eta
     1

epsilon
     0
     0
     0

b
c
```


Euler Angles over Time
Euler Angles in Degrees
Time
phi
theta
psi
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
1
2
3
4
5
6
7
8
9
10


Quaternions over Time
Quaternion
Time
epsilon 1
epsilon 2
epsilon 3
eta
1000
800
600
400
200
0
-200
-400
-600
0
1
2
3
4
5
6
7
8
9
10

*Published with MATLAB® R2017b*