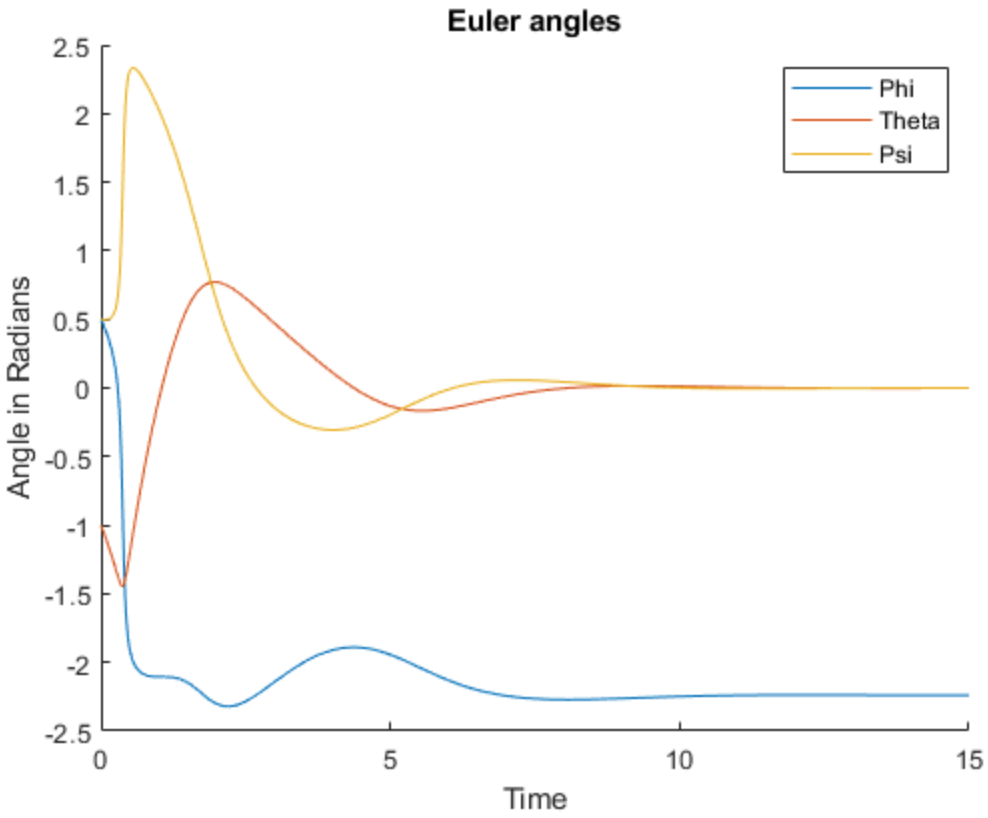
Table of Contents

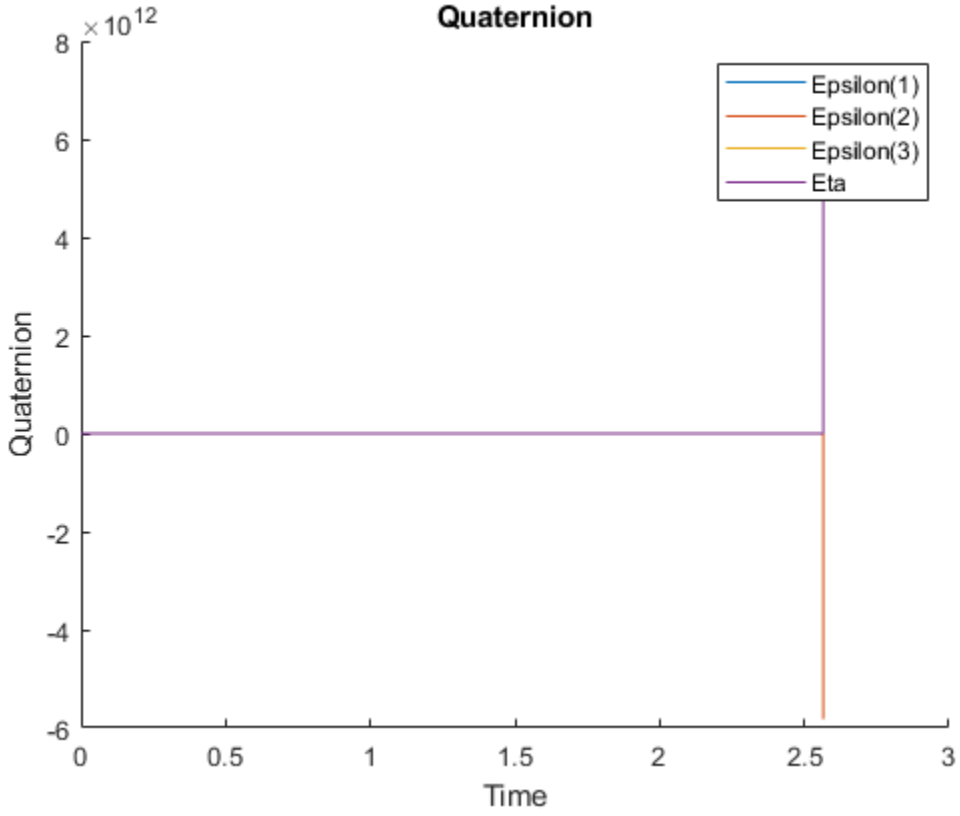
Liam Hood

```
clear
close all
clc

r = 1 ;
h = 3 ;
m = 1 ;
I = (1/12)*m*[ 3*r^2+h^2 0 0 ; 0 3*r^2+h^2 0 ; 0 0 6*r^2 ] ;
w0 = [ .5 ; -1 ; .5 ] ;
```

2

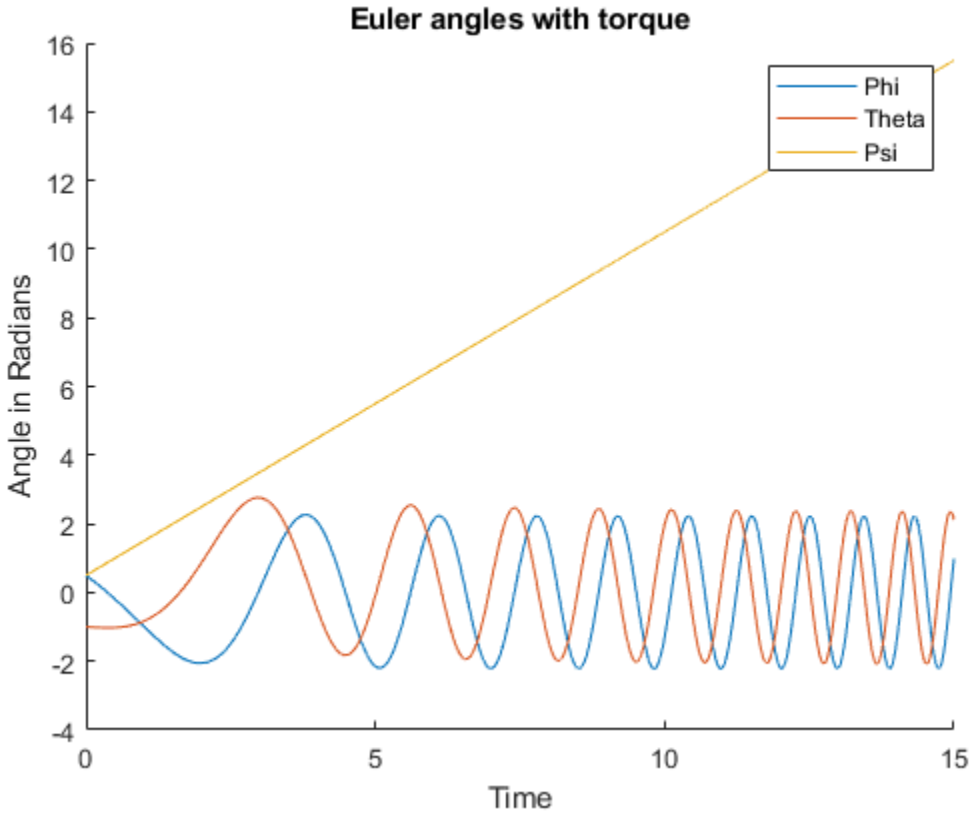
```
t = 0 ;
quat = [ w0 ; 0 ] ;
options = odeset( 'RelTol' , 10^-10 , 'AbsTol' , 10^-10 );
[ t , w ] = ode45( @EulerAngleRates , [ 0 , 15 ] , w0 , options );
figure
hold on
for ii = 1:3
    plot( t , w(:,ii) )
end
title( 'Euler angles' )
xlabel( 'Time' )
ylabel( 'Angle in Radians' )
legend( 'Phi' , 'Theta' , 'Psi' )
hold off

[ tq , quat ] = ode45( @QuaternionRates , [ 0 , 15 ] , quat ,
 options );
figure
hold on
for ii = 1:4
    plot( tq , quat(:,ii) )
end
title( 'Quaternion' )
xlabel( 'Time' )
ylabel( 'Quaternion' )
legend( 'Epsilon(1)' , 'Epsilon(2)' , 'Epsilon(3)' , 'Eta' )
hold off
```


Euler angles
Phi
Theta
Psi
Angle in Radians
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
5
10
15


Quaternion
×10^12
Epsilon(1)
Epsilon(2)
Epsilon(3)
Eta
Quaternion
Time
8
6
4
2
0
-2
-4
-6
0
0.5
1
1.5
2
2.5
3

4

```
[ tT , wT ] = ode45( @TorqueEuler , [ 0 , 15 ] , w0 , options ) ;
figure
hold on
for ii = 1:3
    plot( tT , wT(:,ii) )
end
title( 'Euler angles with torque' )
xlabel( 'Time' )
ylabel( 'Angle in Radians' )
legend( 'Phi' , 'Theta' , 'Psi' )
hold off
```



3

```
function [ dw ] = EulerAngleRates( t , w )
    % phi = w(1) ;
    % theta = w(2) ;
    % psi = w(3) ;
    % phidot = w(4) ;
    % thetadot = w(5) ;
    % psid0t = w(6) ;

    % Define constants
```

```
    c2 = cos( w(2) ) ;
    s2 = sin( w(2) ) ;
    c1 = cos( w(1) ) ;
    s1 = sin( w(1) ) ;

    dw = zeros(3,1) ; % Make output a column

    % Calculate change in angular velocity
    dw(1) = ( c2*w(2) + s1*s2*w(2) + c1*s2*w(3) )/c2 ;
    dw(2) = ( c1*c2*w(2) - s1*c2*w(3) )/c2 ;
    dw(3) = ( s1*w(2) + c1*w(3) )/c2 ;
%     w(1) = w(4)-s2*w(5) ;
%     w(2) = c1*w(5) + s1*s2*w(6) ;
%     w(3) = -s1*w(5) + c1*c2*w(6) ;

end

function [ dquat ] = QuaternionRates( t , quat )
    eps = quat(1:3) ;
    eta = quat(4) ;
    w = eps ;
    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 ;
    dquat = [ epsdot ; etadot ] ;
end

function [ dw ] = TorqueEuler( t , w )
r = 1 ;
m = 1 ;
h = 3 ;
I = [ (1/12)*m*(3*r^2+h^2) 0 0 ; 0 (1/12)*m*(3*r^2+h^2) 0 ; 0
 0 .5*m*r^2 ] ;
iI = inv(I) ;
wcross = [ 0 -w(3) w(2) ; w(3) 0 -w(1) ; -w(2) w(1) 0 ] ;
T = [ -1 ; 0 ; .5 ] ;
    dw = iI * ( T - wcross * I * w ) ;
end
```

Warning: Failure at t=2.565100e+00. Unable to meet integration tolerances without reducing the step size below the smallest value allowed (7.105427e-15) at time t.

Published with MATLAB® R2018b