Table of Contents

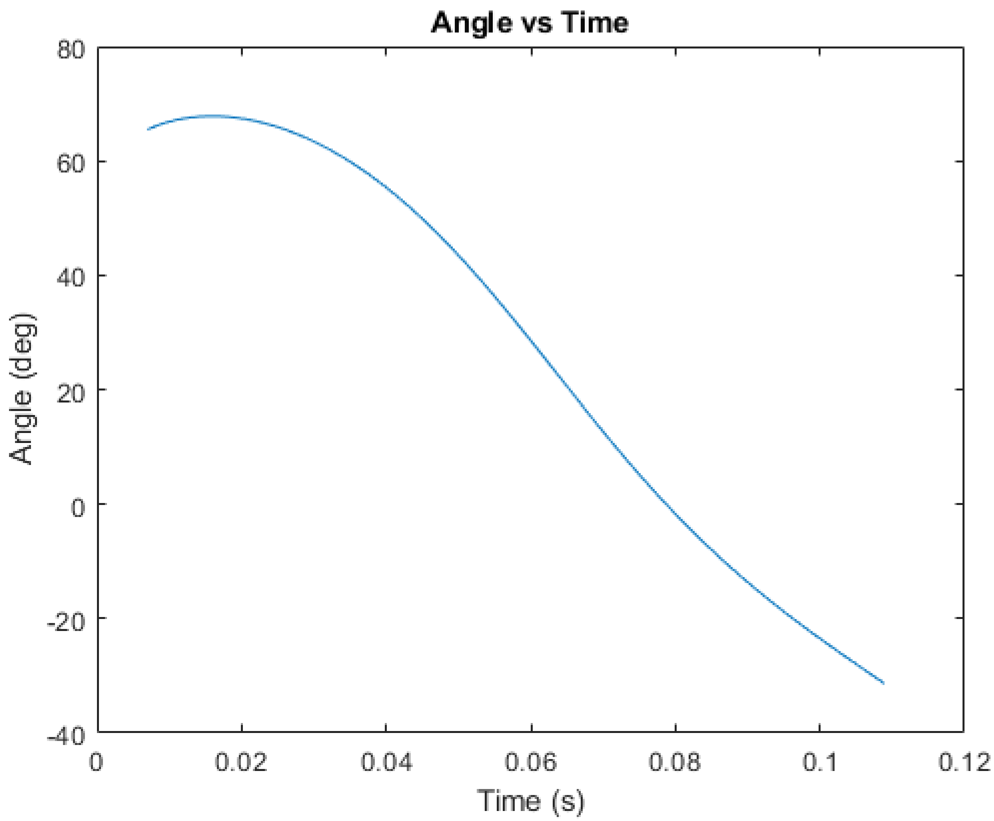
Shaggy Head Angle 1
Angular velocity 2
Cart velocity 3
Force, Work, Restitution 5

Shaggy Head Angle

```
%Set up
clear all ;
t(1) = .233 ;
ii = 2 ;

%Conversion factor from video seconds to real seconds
tc = .137 / 4.567 ;
lc = .0254 / 30.4 ;

%Position in the x-direction constants
a1 = -5.849 ;
b1 = 30.28 ;
c1 = -22.19 ;
d1 = 36.08 ;

%Position in the y-direction constants
a2 = 6.829 ;
b2 = -48.88 ;
c2 = 58.34 ;
d2 = 60.15 ;

%Position of head in x and y coordinates relative to shoulder
r_y(1) = a2 * t(1)^3 + b2 * t(1)^2 + c2 * t(1) + d2  ;
r_x(1) = a1 * t(1)^3 + b1 * t(1)^2 + c1 * t(1) + d1 ;
theta(1) = atand( r_y(1) / r_x(1) ) ;


%Angle of Shaggy's head relative to the horizontal
while t(ii-1) < 3.6

    t(ii) = t(ii-1) + .033 ;
    r_y = a2 * t(ii)^3 + b2 * t(ii)^2 + c2 * t(ii) + d2 ;
    r_x = a1 * t(ii)^3 + b1 * t(ii)^2 + c1 * t(ii) + d1 ;

    theta(ii) = atand( r_y / r_x ) ;

    ii = ii + 1 ;

end

%graph of the angle vs time
t_c = t * tc ;
```

```
    plot( t_c , theta )
    title( 'Angle vs Time' )
    xlabel( 'Time (s)' )
    ylabel( 'Angle (deg)' )

    %finding the total change in theta
    theta_max = max(theta) ;
    theta_min = min(theta) ;
    delta_theta = theta_max - theta_min ;
    disp( [ 'The maximum angular displacement was ' ,
 num2str( delta_theta ) , ' degrees' ] )
```

The maximum angular displacement was 99.2026 degrees

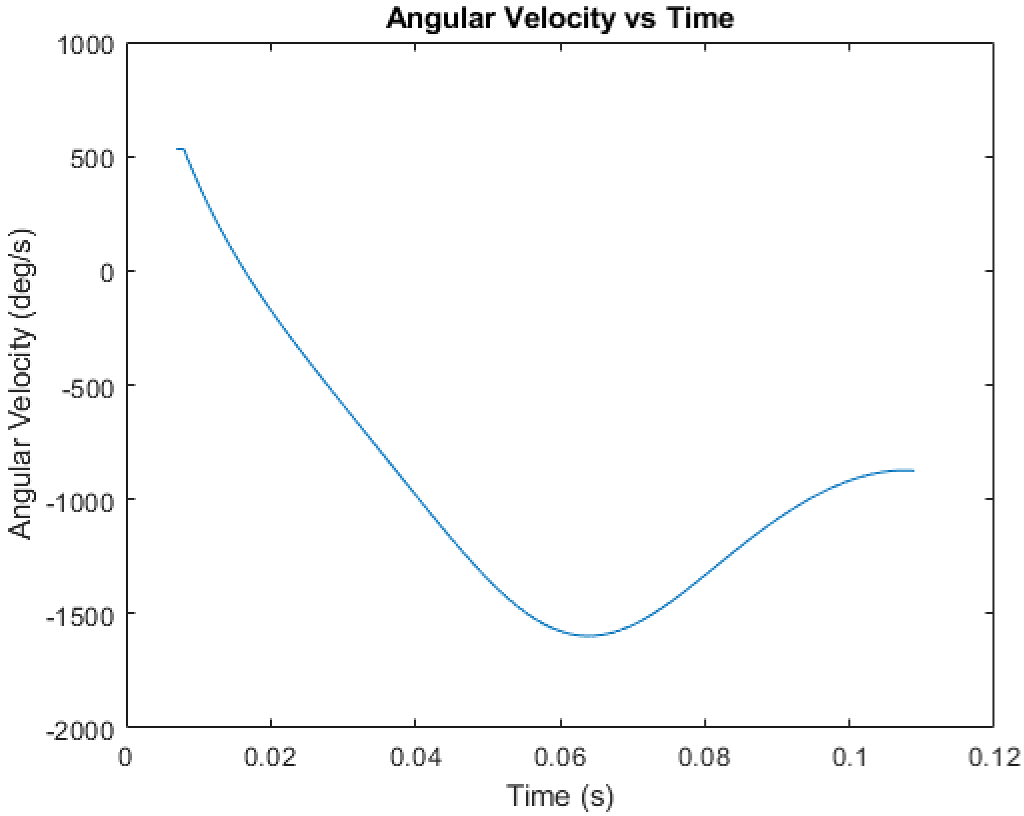


Angular velocity

```
kk = 2 ;
while kk < ii
    omega(kk) = (theta( kk ) - theta( kk - 1 )) / ( .033 * tc ) ;
    kk = kk + 1 ;
end

%This isn't an accurate piece of data but it makes the graph less
 weird
%looking than when it the first point is 0. There will be a missing
 point
```

```
%because I need a value on either side to calculate a difference.
omega(1) = omega(2) ;

%graph of omega vs time
figure
plot( t_c , omega )
title( 'Angular Velocity vs Time' )
ylabel( 'Angular Velocity (deg/s)' )
xlabel( 'Time (s)' )

%finding max angular velocity
max_omega = max( abs( omega ) ) ;
disp( [ 'The maximum angular velocity was ' , num2str( max_omega ) , '
 deg/s' ] )
```

The maximum angular velocity was 1597.3873 deg/s

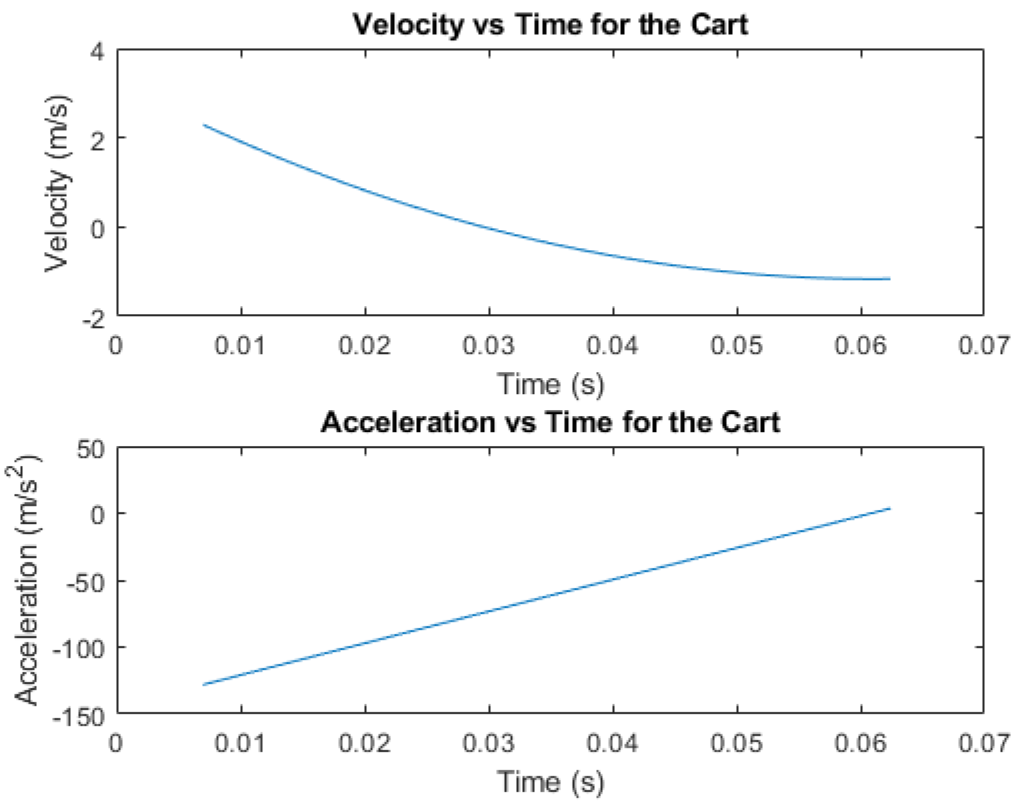


Cart velocity

```
%Constants for Cart kinematics
ac = 12.79 ;
bc = -77.91 ;
cc = 116.4 ;
dc = 35.23 ;

%Counter begins at 2
```

```
jj = 2 ;

%time begins at .233 subjective seconds
t2(1) = .233 ;

%Initial velocities of v and a
v_c(1) = ( 3 * ac * t2(1)^2 + 2 * bc * t2(1) + cc )  ;
a_c(1) = ( 6 * ac * t2(1) + 2 * bc ) ;

 while t2(jj-1) + .033 < 2.1 %calculates v and a from .233 to 2.1
subjective seconds

    t2(jj) = t2(jj-1) + .033 ;
    v_c(jj) = ( 3 * ac * t2(jj)^2 + 2 * bc * t2(jj) + cc )  ;
    a_c(jj) = ( 6 * ac * t2(jj) + 2 * bc ) ;

    jj = jj + 1 ;


 end

 %Applying conversion factors to fix scales and values
 t2_c = t2 * tc ;
 v_cc = v_c * ( lc / tc ) ;
 a_cc = a_c * ( lc / tc^2 ) ;

 %Acceleration and Velocity graphs
 figure
 subplot( 2 , 1 , 1 )
 plot( t2_c , v_cc )
 title( 'Velocity vs Time for the Cart' )
 xlabel( 'Time (s)' )
 ylabel( 'Velocity (m/s)' )
 subplot( 2 , 1 , 2 ) , plot( t2_c , a_cc )
 title( 'Acceleration vs Time for the Cart' )
 xlabel( 'Time (s)' )
 ylabel( 'Acceleration (m/s^2)' )

 %Finding intitial and final velocity as well as the maximum
 %acceleration
 v_initial = v_cc(1) ;
 v_final = v_cc(jj-1) ;
 disp( [ 'The velocity before the collision was ' ,
num2str( v_initial ) , ' m/s' ] );
 disp( [ 'The velocity after the collision was ' ,
num2str( v_final ) , ' m/s' ] );
 a_max = max( abs( a_cc ) ) ;
 disp( [ 'The maximum acceleration was ' , num2str( a_max ) , ' m/
s^2' ] )
```

```
The velocity before the collision was 2.2889 m/s
The velocity after the collision was -1.1614 m/s
The maximum acceleration was 128.0766 m/s^2
```



Force, Work, Restitution

```
%given
m = 2.4927 ;
g = 9.8 ;

%Crush of the bumper
whole = 61.65 * lc;
crushed = 33.00 * lc ;
crush = whole - crushed ;
disp( [ 'The bumpered was crushed ' , num2str( crush ) , ' meters'
] )

%Coeffecient of restitution
e = abs( v_final ) / abs ( v_initial ) ;
disp( [ 'The coeffecient of restitution is ' , num2str( e ) ] )

%Max force
f_max = m * a_max ;
disp( [ 'The maximum force is ' , num2str( f_max ) , ' N' ] )

%Work
% Work is the the difference in between the final and initial
energy
% states. It can be calculated as an integral of force dotted with
```

```
   % displacement. Or more easily we can calculate it by knowing that
 the
   % bumper was the only thing that slowed the cart. This means that
 the
   % work done by the bumper will explain all of the difference in
 kinetic
   % energy between each state
   % work_stop = E_2 - E_1
   % work_reform = E_3 - E_2
   % E_1 is kinetic energy before collision
   % E_2 is kinetic energy at the wall
   % E_3 is kinetic energy after the collision

   %Kinetic energy at all states
   E_1 = .5 * m * v_initial^2 ;
   E_2 = 0 ;
   E_3 = .5 * m * v_final^2 ;

   %Work done by bumper during deformation then reformation
   work_1 = E_2 - E_1 ;
   work_2 = E_3 - E_2 ;
   work = abs( work_1 ) + abs( work_2 ) ;
   disp( [ 'The work done by the bumper was ' , num2str( work ) , '
 joules' ] )
```

```
The bumpered was crushed 0.023938 meters
The coeffecient of restitution is 0.50741
The maximum force is 319.2566 N
The work done by the bumper was 8.2107 joules
```

Published with MATLAB® R2017b