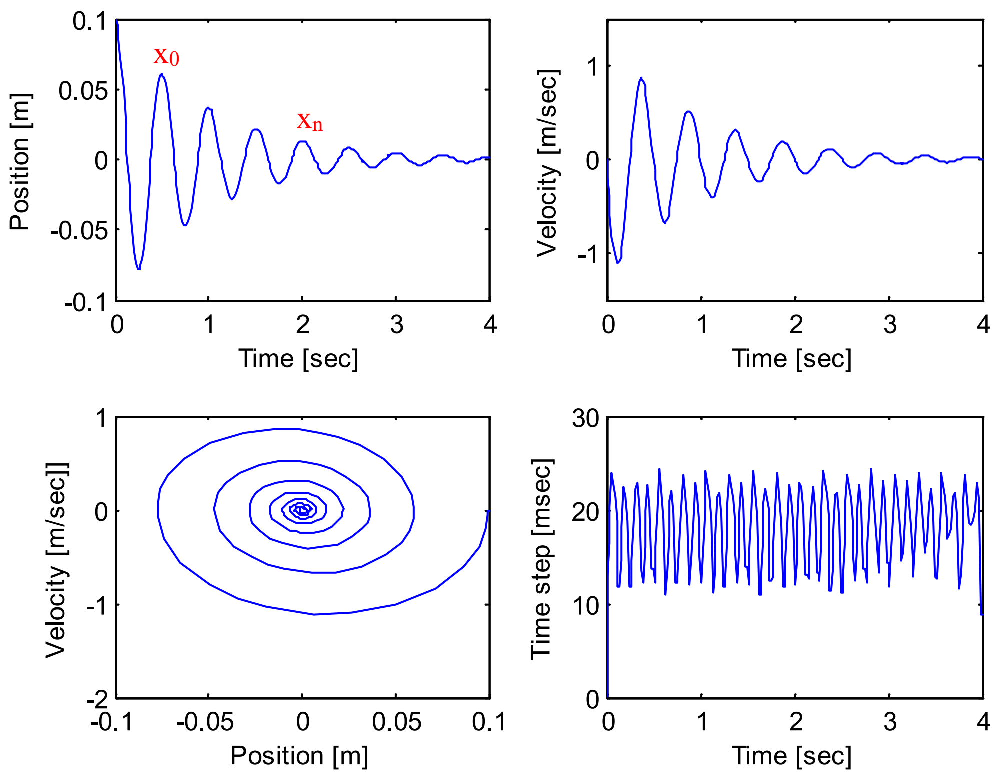
ME 481 – H14 **Name ________________________**

1) Copy the MATLAB files "ode_smd_main.m" and "ode_smd_yd.m" below into two separate M files in your working directory. Use the exact file names. Run "ode_smd_main.m" to provide a forward dynamic simulation of a spring-mass-damper. Provide MATLAB plots of position and velocity as a function of time and a MATLAB plot of time step as a function of time. Also provide a phase plane plot of velocity as a function of position.

viscous damping

mean time step = 17.05 msec
CPU time = 0.0466 sec



2) Compute damping ratio ζ based on spring-mass-damper coefficients in file "ode_smd_main.m" and using log-decrement methods from the decay envelope in your simulation data. Show your work.

ζ_{COEFF} ___0.0796___ $\zeta_{LOG\text{-}DEC}$ ___0.0802___

spring-mass-damper coefficients

m = 1 kg, k = 157.9 N/m, c = 2 N.sec/m

$$\omega_n = \sqrt{\frac{k}{m}} = \sqrt{\left(\frac{157.9\text{N}}{\text{m}}\right)\left(\frac{1}{1\text{ kg}}\right)\left(\frac{\text{kg.m}}{\text{N.sec}^2}\right)} = 12.56 \text{ rad/sec}$$

$\omega_n = 2\pi f_n$ $f_n = 2$ Hz $T = 0.5$ sec

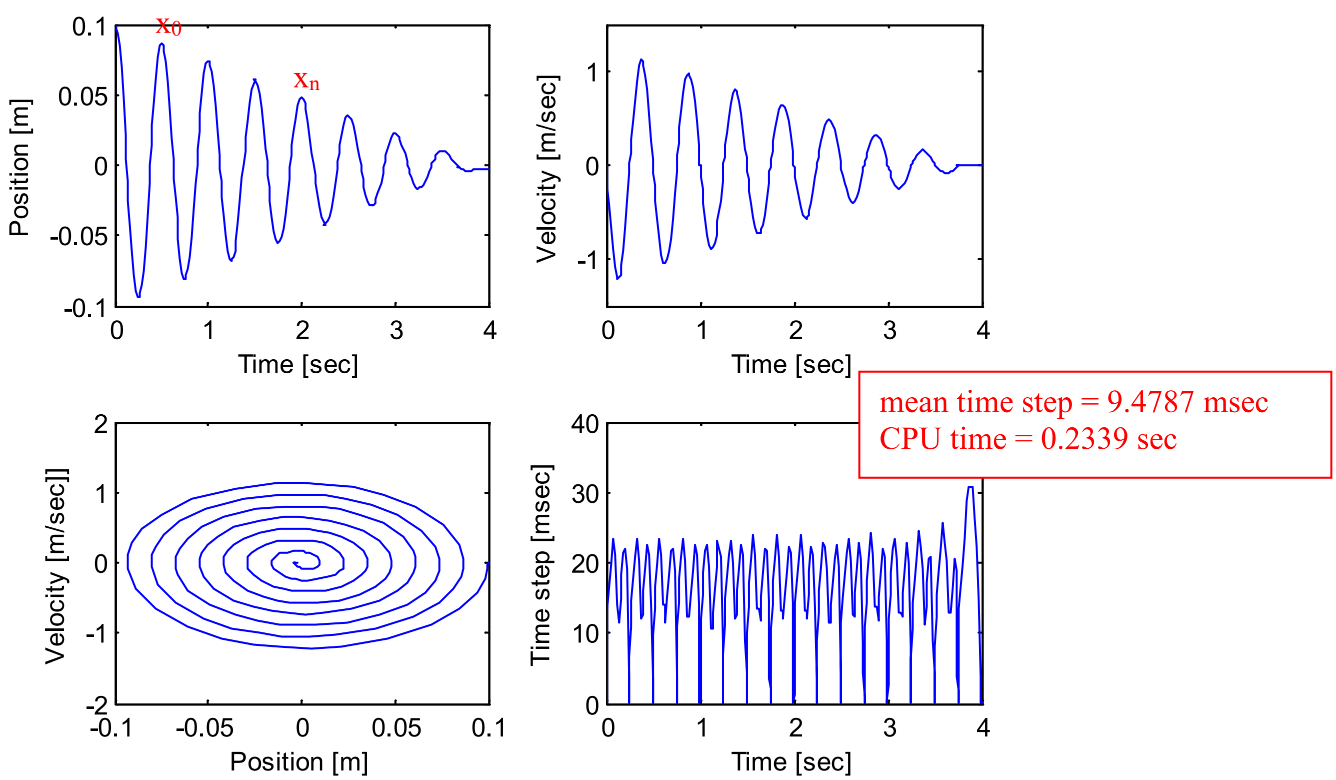
$$\zeta = \frac{c}{c_{CR}} = \frac{c}{2\, m\, \omega_n} = \left(\frac{1}{2}\right)\left(\frac{2\text{N.sec}}{\text{m}}\right)\left(\frac{1}{1\text{ kg}}\right)\left(\frac{\text{sec}}{12.56\text{rad}}\right)\left(\frac{\text{kg.m}}{\text{N.sec}^2}\right) = 0.0796$$

log-decrement

$n = 3$, $x_0 = 0.06031$ m, $x_n = 0.01323$ m $\quad \delta = \frac{1}{n}\ln\left(\frac{x_0}{x_n}\right) = 0.50567 \quad \zeta = \frac{\delta}{\sqrt{4\pi^2 + \delta^2}} = 0.0802$

3) Modify "ode_smd_yd.m" for Coulomb friction force equal to f = 0.5 Newtons and viscous damping c = 0. Use the simple friction model $F_{FRICTION} = -f * sign(\dot{x})$. Provide MATLAB plots of position and velocity as a function of time and a MATLAB plot of time step as a function of time. Also provide a phase plane plot of velocity as a function of position. Provide a listing of your code. Comment on the differences from part 1) above. Specifically address time step and CPU time.

simple Coulomb friction – friction force = - f * sign(velocity)



a) linear decay envelope for Coulomb versus exponential decay envelope for viscous

b) time step gets VERY small at velocity reversal, mean time step is about 1/2 for viscous, CPU time about 5 times slower

4) Compute Coulomb friction force based on the decay envelope in your simulation data from part 3). Show your work

$f_{COULOMB}$ 0.509 N

$n = 3$, $x_0 = 0.08703$ m, $x_n = 0.04837$ m $s = \frac{x_0 - x_n}{n\,T} = 0.02577$ m/sec

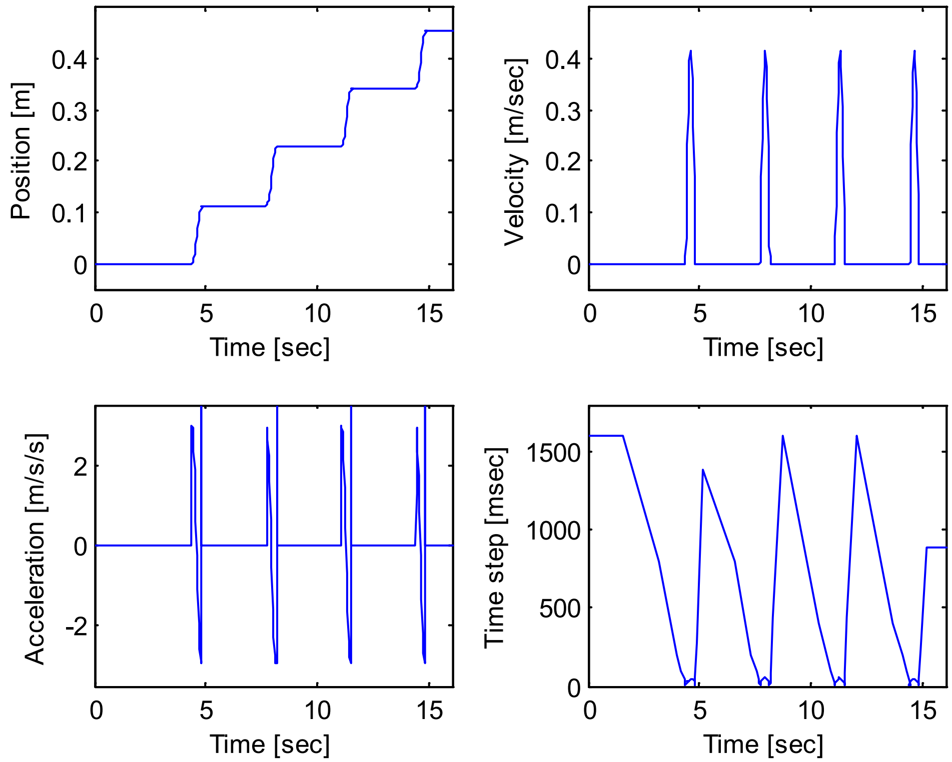
$$f = \frac{\pi\, k\, s}{2\, \omega_n} = \frac{\pi}{2}\left(\frac{157.9\text{N}}{\text{m}}\right)\left(\frac{0.02577\text{m}}{\text{sec}}\right)\left(\frac{\text{sec}}{12.56\text{rad}}\right) = 0.509\text{ N}$$

$$f = \frac{k(x_0 - x_n)}{4\, n} = \left(\frac{157.9\text{N}}{\text{m}}\right)\left(\frac{0.08703\text{m} - 0.04837\text{m}}{4(3)}\right) = 0.509\text{ N}$$

```matlab
% ode_smd_main.m - main program for example use of ODE solver
%   spring-mass-damper
% HJSIII, 20.04.09
%
% ODE coded in file ode_smd_yd.m  - m*xdd = Fext + Fspr + Fdamp
%
% {y} = { x  }          {yd} = { xd  }
%       { xd }                 { xdd }

clear

global  m  k  c

% physical constants
% x [m]
% xd [m/sec]
% xdd [m/sec^2]
m = 1;              % mass [kg]
k = 157.9;          % spring [N/m] - causes wn = 2Hz
c = 2;              % viscous [N.sec/m] - causes 4 sec exp decay

% initial conditions
y0 = [ 0.1  0  ]';   % free release

% time range
tspan = [ 0  4 ];

% measure CPU time
tic;
[ t, y ] = ode23( 'ode_smd_yd', tspan, y0 );
t_exe = toc

% time step
h = 1000 * diff(t);   % units [msec]
h = [ h ; h(end) ];   % repeat last value to make the same length as t

n_time_steps = length( t )
ave_time_step = mean( h )

% time domain results
figure( 1 )
  subplot( 2, 2, 1 )
    plot( t, y(:,1) )
    xlabel( 'Time [sec]' )
    ylabel( 'Position [m]' )
    axis( [ 0 4  -0.1 0.1 ] )
  subplot( 2, 2, 2 )
    plot( t, y(:,2) )
    xlabel( 'Time [sec]' )
    ylabel( 'Velocity [m/sec]' )
    axis( [ 0 4  -1.5 1.5 ] )
  subplot( 2, 2, 3 )
    plot( y(:,1), y(:,2) )
    xlabel( 'Position [m]' )
    ylabel( 'Velocity [m/sec]]' )
  subplot( 2, 2, 4 )
    plot( t, h )
    xlabel( 'Time [sec]' )
    ylabel( 'Time step [msec]' )

% bottom - ode_smd_main
```

```
function yd = ode_smd_yd( t, y )
% provides yd for integration
% spring-mass-damper
%    m*xdd = Fext + Fspr + Fdamp
% HJSIII, 20.04.09

global m  k  c

% free motion
Fext = 0;

% individual terms
x = y(1);
xd = y(2);

% ordinary differential equation (ODE)
Fspr = -k*x;
Fdamp = -c*xd;
xdd = ( Fext + Fspr + Fdamp ) / m;

% return values
yd(1,1) = xd;
yd(2,1) = xdd;

return
% bottom - ode_smd_yd

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

function yd = ode_smcoulomb_yd( t, y )
% provides yd for integration
% spring-mass-Coulomb friction     use simple friction model
%   m*xdd = Fext + Fspr + Ff
% HJSIII, 20.04.09

global  m  k  c

% free motion
Fext = 0;

% Coulomb friction
f = 0.5;           % [N]

% individual terms
x = y(1);
xd = y(2);

% ordinary differential equation (ODE)
Fspr = -k*x;
%Fdamp = -c*xd;
Ff = -f*sign( xd );  % simple friction model


%xdd = ( Fext + Fspr + Fdamp ) / m;
xdd = ( Fext + Fspr + Ff ) / m;

% return values
yd(1,1) = xd;
yd(2,1) = xdd;

return
% bottom - ode_smcoulomb_yd
```

5) Modify your MATLAB code to simulate the stick-slip drag-sled from H12 part 2). You must use the more detailed friction model on page 1 of Notes_08_06. The simple friction model used above will not suffice. Provide MATLAB plots of position, velocity and acceleration of the drag-sled as a function of time and a MATLAB plot of time step as a function of time. Provide a listing of your code. Comment on the differences from H12.



a) very similar curves

b) MATLAB	about 3.5 sec apart	max velocity ≈ 0.42 mps	max accel ≈ 2.95 mps^2
WM	about 3.5 sec apart	max velocity ≈ 0.45 mps	max accel ≈ 2.95 mps^2

c) time step gets VERY small during slip

```
% ode_stickslip_main.m - main program for example use of ODE solver
%   stick slip drag sled
% HJSIII, 20.04.09
%
% ODE coded in file ode_stickslip_yd.m  - m*xdd = Fext + Fspr + Ff
%
% {y} = { x  }           {yd} = { xd  }
%       { xd }                  { xdd }

clear

global  m  k  mu_s  mu_d  g  v_driver  eps

% physical constants
m = 12.5;          % mass [kg]
k = 742.5;         % spring [N/m]
mu_s = 0.9;        % static coefficient of friction
mu_d = 0.6;        % dynamic coefficient of friction
g = 9.81;          % gravity [mpss]
v_driver = 0.034;  % driver velocity [mps]
eps = 1e-7;        % very small value

disp( ' ' )
disp( 'Stick-slip drag-sled' )

% initial conditions
y0 = [ 0  0 ]';

% time range
tmax = 16;
tspan = [ 0  tmax ];

% call ODE solver
tic;
[ t, y ] = ode23( 'ode_stickslip_yd', tspan, y0 );
%[ t, y ] = ode23( 'ode_stickslip_pacejka_yd', tspan, y0 );
t_exe = toc

% time step
h = diff(t);          % units [sec] - do not scale to msec yet
h = [ h ; h(end) ];   % repeat last value to make the same length as t

% all values
x_all = y(:,1);
xd_all = y(:,2);
del_xd = diff( xd_all );
xdd_all = [ del_xd ; del_xd(end) ] ./ h;  % approximate accleration - same length as xd

n_time_steps = length( t )
ave_time_step = mean( h )

% time domain results
figure( 1 )
  clf
  subplot( 2, 2, 1 )
    plot( t, x_all )
    xlabel( 'Time [sec]' )
    ylabel( 'Position [m]' )
    axis( [ 0 tmax  -0.05 0.5 ] )

  subplot( 2, 2, 2 )
    plot( t, xd_all )
    xlabel( 'Time [sec]' )
    ylabel( 'Velocity [m/sec]' )
    axis( [ 0 tmax  -0.05 0.5 ] )

  subplot( 2, 2, 3 )
    plot( t, xdd_all )
    xlabel( 'Time [sec]' )
    ylabel( 'Acceleration [m/s/s]' )
    axis( [ 0 tmax  -3.5  3.5 ] )
```

```
  subplot( 2, 2, 4 )
    plot( t, h*1000 )          % scale to msec here
    xlabel( 'Time [sec]' )
    ylabel( 'Time step [msec]' )
    axis( [ 0 tmax  0 1800 ] )

% bottom - ode_stickslip_main
```

++

```
function yd = ode_stickslip_yd( t, y )
% provides yd for integration
% stick slip drag sled
%   m*xdd = Fext + Fspr + Ff      friction model from Notes_08_06
% HJSIII, 20.04.09

global  m  k  mu_s  mu_d  g  v_driver  eps

% free motion
Fext = 0;

% individual terms
x = y(1);
xd = y(2);

% spring force
x_driver = v_driver * t;
Fspr = k * ( x_driver - x );

% friction force - simple model
Ff = -mu_d * m * g * sign( xd );

% special handling for zero velocity - need for stick-slip
if abs(xd) < eps,
  Ff = -Fspr;
  if abs(Ff) > (mu_s*m*g) ,
    Ff = -mu_s * m * g * sign( Fspr );
  end
end

% ordinary differential equation (ODE)
xdd = ( Fext + Fspr + Ff ) / m;

% return values
yd(1,1) = xd;
yd(2,1) = xdd;

return
% bottom - ode_stickslip_yd
```

ME 481 – H14 **Name __________________________**

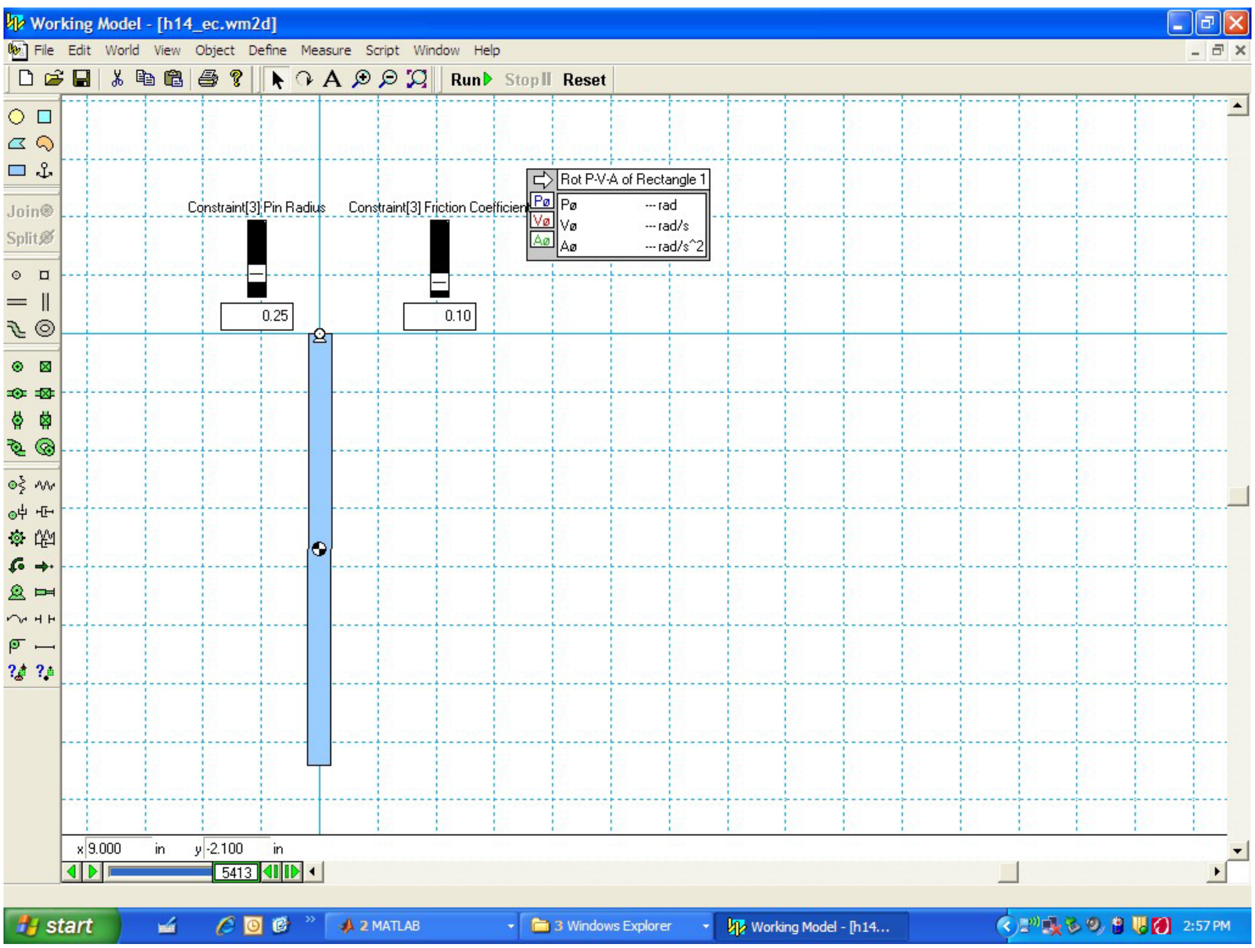
EXTRA CREDIT

Modify your pendulum from H13 using "Script Pin Friction" with "Pin Radius = 0.25 inch" and "Friction Coefficient = 0.1" for ±10 degrees of motion. Attach a screen shot of your mechanism. Provide a phase plane plot and estimate joint friction torque from the time decay envelope.

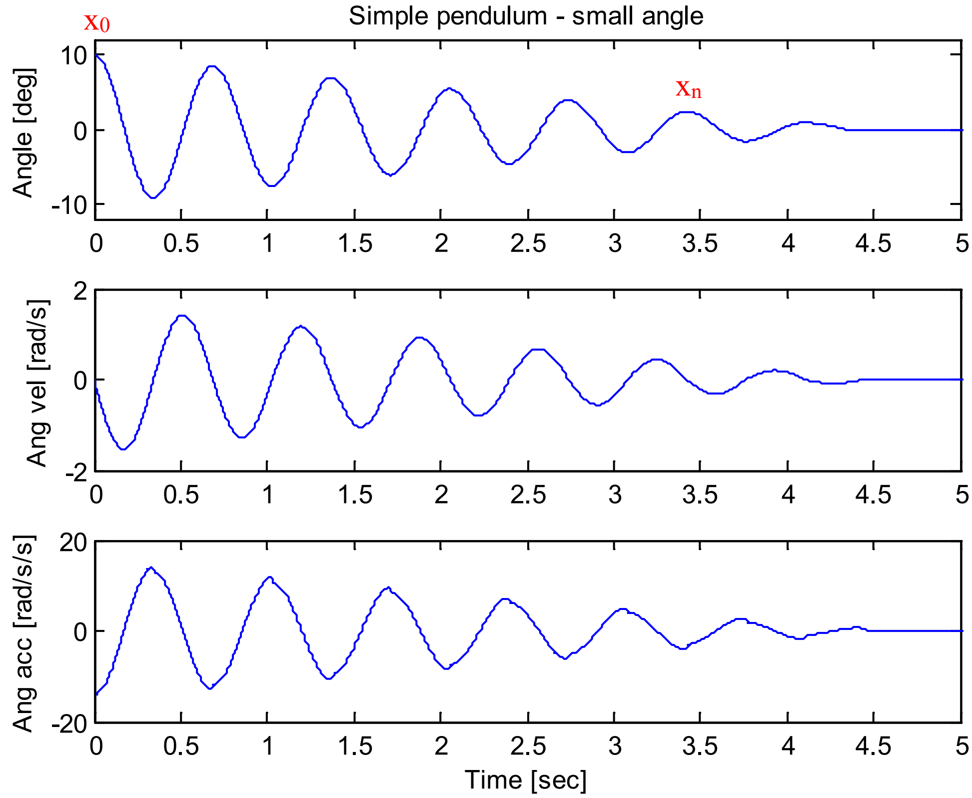
$$T_f = \mu\, r\, m\, g = (0.1)\,(0.25\text{ inch})\,(0.46\text{ lbf}) = 0.0115\text{ in.lbf}$$

$$T_f = \frac{m\, g\, a\, (\theta_0 - \theta_n)}{4\, n} \qquad \theta_0 = 10.02° \qquad \theta_5 = 2.337° \qquad n = 5 \qquad (m\, g) = 0.46\text{ lbf} \qquad a = 3.7\text{ inch}$$

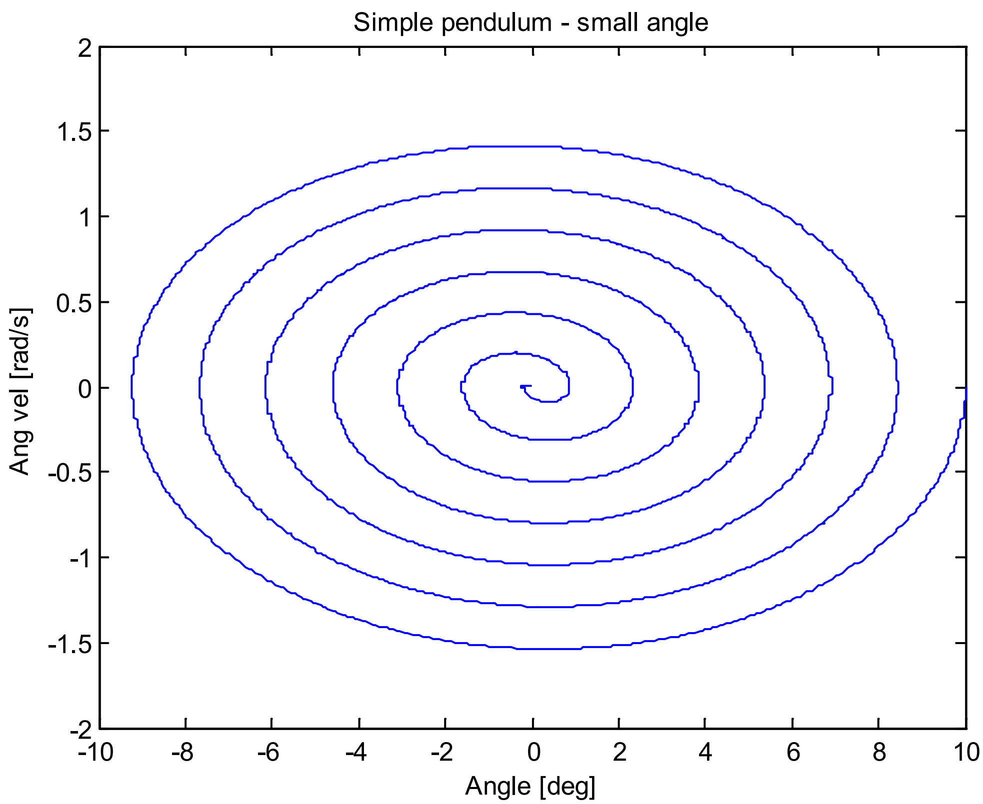
$$T_f = \frac{(0.46\text{ lbf})(3.7\text{ inch})(10.02° - 2.337°)}{4(5)}\left(\frac{\pi\text{ rad}}{180°}\right) = 0.01141\text{ in.lbf}$$



Name ________________________



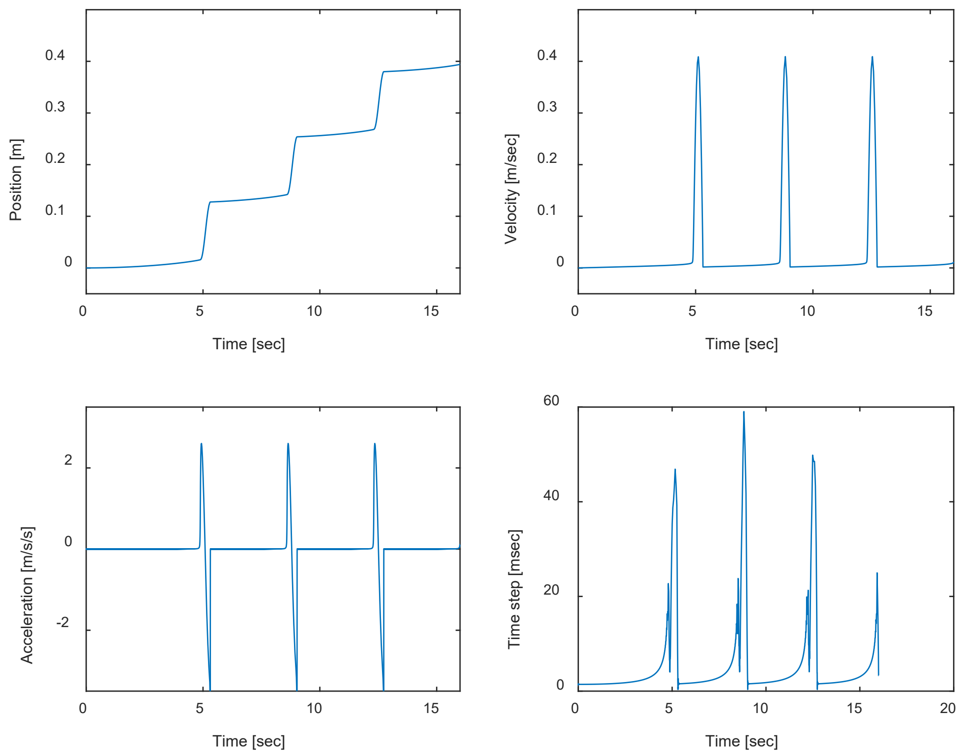
x_0
x_n
Simple pendulum - small angle
Angle [deg]
10
0
-10
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Ang vel [rad/s]
2
0
-2
Ang acc [rad/s/s]
20
0
-20
Time [sec]



Simple pendulum - small angle
Ang vel [rad/s]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-10
-8
-6
-4
-2
0
2
4
6
8
10
Angle [deg]

```
% h14_ec.m - ME 481 HW03 - read WM data for pendulum
% HJSIII, 13.04.06

% general constants
d2r = pi / 180;

% pendulum constants
m = 0.46;   % [lbm]
a = 3.7;    % [in]
JG = 1.5;   % [lbm.in.in]
g = 386;    % [ips2]

tau = sqrt( 4*pi*pi*( JG + m*a*a ) / ( m * g * a ) )
fn = 1 / tau

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% load WM data for pendulum with small angle
load h14_ec_cut.txt;

t = h14_ec_cut(:,1);
th = h14_ec_cut(:,2);
phi_deg = th/d2r + 90;
phid = h14_ec_cut(:,3);
phidd = h14_ec_cut(:,4);

% plot small angle results
figure( 1 )
  clf
  subplot( 3, 1, 1 )
  plot( t,phi_deg,'b' )
  axis( [ 0 5  -12 12 ] )
  ylabel( 'Angle [deg]' )
  title( 'Simple pendulum - small angle' )

  subplot( 3, 1, 2 )
  plot( t,phid,'b' )
  axis( [ 0 5  -2 2 ] )
  ylabel( 'Ang vel [rad/s]' )

  subplot( 3, 1, 3 )
  plot( t,phidd,'b' )
  axis( [ 0 5  -20 20 ] )
  xlabel( 'Time [sec]' )
  ylabel( 'Ang acc [rad/s/s]' )

figure( 2 )
  clf
  plot( phi_deg, phid, 'b' )
  axis( [ -10 10  -2 2 ] )
  xlabel( 'Angle [deg]' )
  ylabel( 'Ang vel [rad/s]' )
  title( 'Simple pendulum - small angle' )
```

Friction Model using Pacejka Magic Formula



friction model	CPU time [sec]	time stpes	ave time step [msec]
pseudo-code	1.7690	10972	1.5145
Pacejka xm=0.01 mps	0.7313	6667	2.4004
Pacejka xm=0.001 mps	7.3105	68107	0.2349

+ faster execution
- velocity is not exactly zero during stick phase
- asymmetric acceleration

```
function yd = ode_stickslip_pacejka_yd( t, y )
% provides yd for integration
% stick slip drag sled
%   m*xdd = Fext + Fspr + Ff      Pacejka friction model from Notes_08_06
% HJSIII, 20.04.26

global  m  k  mu_s  mu_d  g  v_driver  eps

% free motion
Fext = 0;

% individual terms
x = y(1);
xd = y(2);

% spring force
x_driver = v_driver * t;
Fspr = k * ( x_driver - x );
```

```
% Pacejka constants
xm_pacj = 0.01;
yp_pacj = mu_s;
ya_pacj = mu_d;

% Pacejka coefficients
D = yp_pacj;
C = 1 + (1 - 2 * asin( ya_pacj/D ) / pi);
slope = yp_pacj / (xm_pacj/2);
B = slope /C /D;
E = (B*xm_pacj - tan(pi/2/C)) / (B*xm_pacj - atan(B*xm_pacj));

% Pacejka variables
x_pacj = xd;
mu = D*sin( C*atan( B*x_pacj - E*(B*x_pacj-atan(B*x_pacj)) ) );

% Pacejka friction model
Ff = -mu * m * g;

% ordinary differential equation (ODE)
xdd = ( Fext + Fspr + Ff ) / m;

% return values
yd(1,1) = xd;
yd(2,1) = xdd;

return
% bottom - ode_stickslip_pacejka_yd
```