

---

```
clc
close all
clear all
```

# ME 292B - Homework #3 Problem 1

## Problem 1(d)

```
% Constants and input defined in stance_dynamics function

% Simulate controlled system for 5 ICs
figure(1)
hold on
figure(2)
hold on
for i = 1:5
    sim_10_bounces([2*i + 1;0])
end
figure(1)
title('Position vs Time')
xlabel(' Time[s]')
ylabel(' Position, y(t) [m]')
legend('x0 = 3m', 'x0 = 5m', 'x0 = 7m', 'x0 = 9m', 'x0 = 11m')
figure(2)
title('Velocity vs Time')
xlabel(' Time[s]')
ylabel(' Velocity [m/s]')
legend('x0 = 3m', 'x0 = 5m', 'x0 = 7m', 'x0 = 9m', 'x0 = 11m',...
       'Location','southeast')

function [] = sim_10_bounces(x0)
% Simulate 10 bounces

% Define time range to simulate the system
Tspan = linspace(0,10,100) ;
t0 = 0 ; % Initial Time

% Array to store data for all bounces
t_vec = [] ; x_vec = [] ;

for j=1:5

    % Define the events functions (stop integration when stance stage
    reached)
    options1 = odeset('Events', @ground_contact) ;

    % Simulate the system in flight
    [t_ode x_ode] = ode45(@flight_dynamics, t0+Tspan, x0, options1) ;

    % Save simulation data
    t_vec = [t_vec; t_ode] ;
```

---

```

x_vec = [x_vec; x_ode] ;

% Initialize x0,t for stance stage
x0 = [x_ode(length(x_ode),1); x_ode(length(x_ode),2)]; t0 =
t_vec(end);

% Define the events functions (stop integration when flight stage
reached)
options2 = odeset('Events', @leaves_ground) ;

% Compute stance dynamics

% Constants
k = 20000; % N/m
m = 80; % kg
L0 = 1; %m
c = 50000; %Ns/m
g = 9.81 ;

% Design input controller , t restarts at each start of stance
position

% From energy conservation, we can find target velocity at start
% of stance dynamics ( $1/2*m*v^2 = mgh$ ,  $h = 2m - 1m = 1m$ )
v0_target = sqrt(2*g*1);

% Calculate real time error from target velocity and actual
velocity
error = @(x) v0_target - x(2);

% PD-Controller
u = @(t,x) c*x(2) + 7500*error(x)*(x(2)>=0); % only apply u during
% second half of stance stage (gain of 7500 found from trial/
error)

xdot = @(t,x) [x(2) ;
((u(t,x)-k*(x(1)-L0)-c*x(2)-m*g)/m)] ;

% Simulate the system in stance stage
[t_ode x_ode] = ode45(xdot, t0+Tspan, x0, options2) ;

% Save simulation data
t_vec = [t_vec; t_ode] ;
x_vec = [x_vec; x_ode] ;

% Initialize x0,t for flight stage
x0 = [x_ode(length(x_ode),1); x_ode(length(x_ode),2)]; t0 =
t_vec(end);
end

% Plot position and velocity
figure(1)
plot(t_vec, x_vec(:,1), 'LineWidth',2) ; grid on ;
figure(2)

```

---

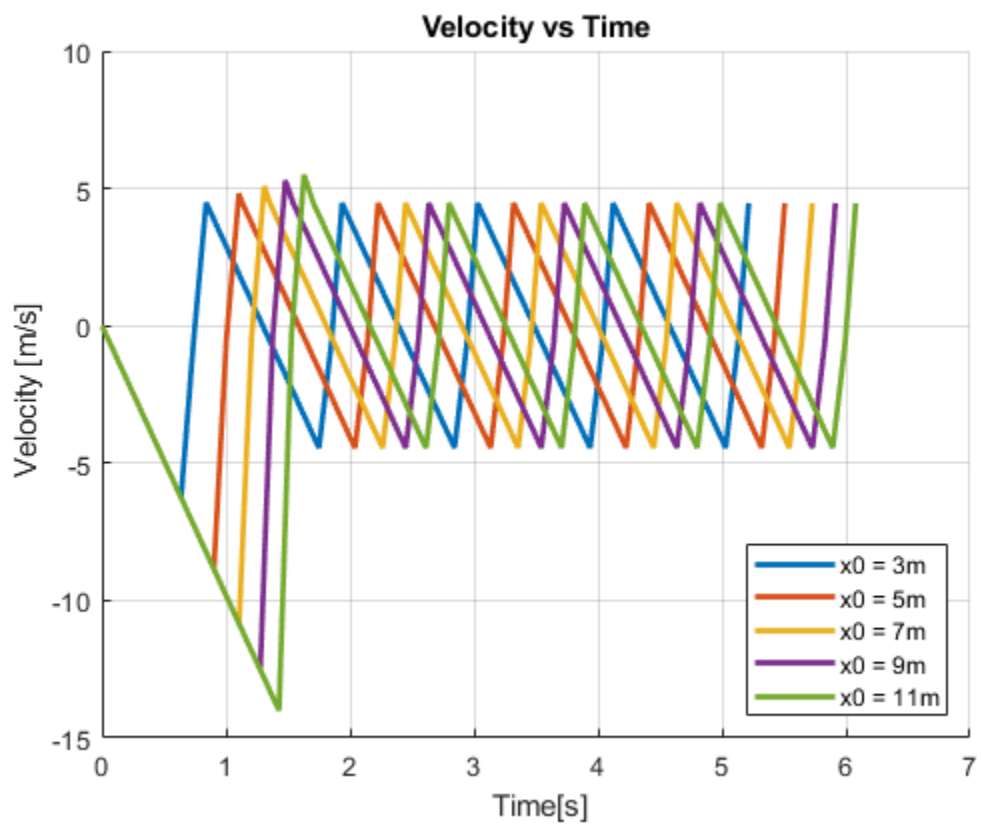
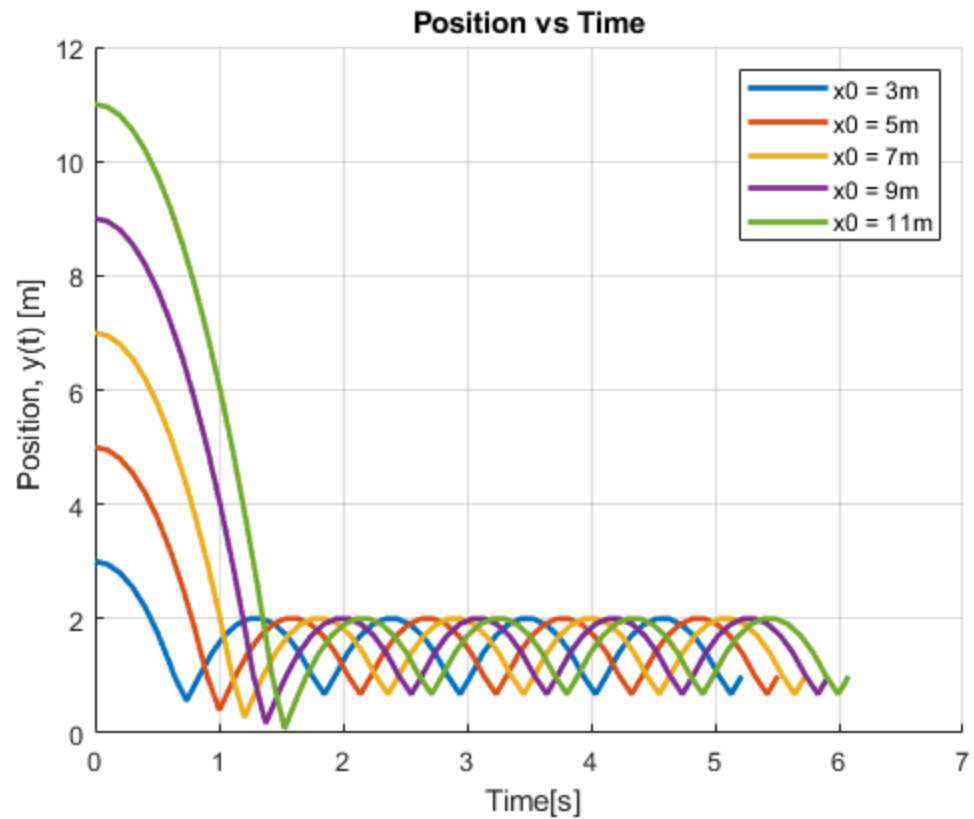
---

```
plot(t_vec, x_vec(:,2), 'LineWidth',2) ; grid on ;
end

% Function that describes flight dynamics
function dx = flight_dynamics(t, x)
    y = x(1) ;
    dy = x(2) ;
    g = 9.81 ;
    dx = [dy ;
          -g] ;
end

% Event function that describes when hopper hits the ground
function [value,isterminal,direction] = ground_contact(t,x)
    y = x(1);
    value = round(y - 1,2) ; % detect when y - L0 == 0
    isterminal = 1 ; % stop integration when y - L0 == 0
    direction = -1 ; % detect zero when function is decreasing
end

% Event function that describes when hopper leaves the ground from
stance
function [value,isterminal,direction] = leaves_ground(t,x)
    y = x(1);
    value = round(y - 1,2) ; % detect when y - L0 == 0
    isterminal = 1 ; % stop integration when y - L0 == 0
    direction = 1 ; % detect only +ve y to -ve y transitions
end
```



---

*Published with MATLAB® R2019a*