Activity 1

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% This activity was just for creating a script with the proper header and
% common commands (lines 5-7).
```

Published with MATLAB® R2020b

Activity 2

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we are making a new script to make two variables, t and
% x, and assign them initial values. We will then use these variables to
% evaluate v and display the result using the display function. The output
% value will also be concatenated with a character vector to provide context.

% Setup initial variables
t = 0.4;
x = 2;

% Calculate output
v = (sin(2*pi*t)+x)^2+2*x+exp(t);

% Format output
t_char = num2str(t);
v_char = num2str(v);
output_char = strcat("The value at ",t_char," seconds is ",v_char);

% Display output
disp(output_char);
```

```
The value at 0.4 seconds is 12.1885
```

Published with MATLAB® R2020b

Activity 3

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we are creating a row vector v1 using the equal spacing
% command and row vector v2 through linspace. We will also create the
% column vector v3 through transposition. Then we will find matrix A
% through multiplaction of v2 and v3 and use index notation to pull
% variables x, v4, and B from matrix A.


% Create initial vectors
v1 = 2:2:8;
v2 = linspace(1,10,4);
v3 = [3 7 2].';

% Calculate A
A = v2.*v3;

% Data extractions from A
x = A(3,2);
v4 = A(2,:);
B = A(2:3,2:3);
```

Published with MATLAB® R2020b

Activity 4

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we will setup a matrix A and use the max function to
% find the maximum value within this matrix. We will then redefine A to a
% completely different row vector and use the diff function to identify the
% differences between all the integers within this row vector A. After
% doing this, we will define a new function activity4 (outside the script)
% and call it, inputting variables defined in this script.

% Setup initial matrix
A = [4 2;0 9];

% Find max
A_max = max(A,[],'all');

% Reset A
A = [1 5 6 2 85 14 6 4 23];

% Find difference
A_diff = diff(A);

% Setup initial functions for function activity4
x = 2;
t = 9;

% Evaluate activity4 with inputs
y = activity4(x,t); %y = 2sin(2t)-3x
```

Published with MATLAB® R2020b

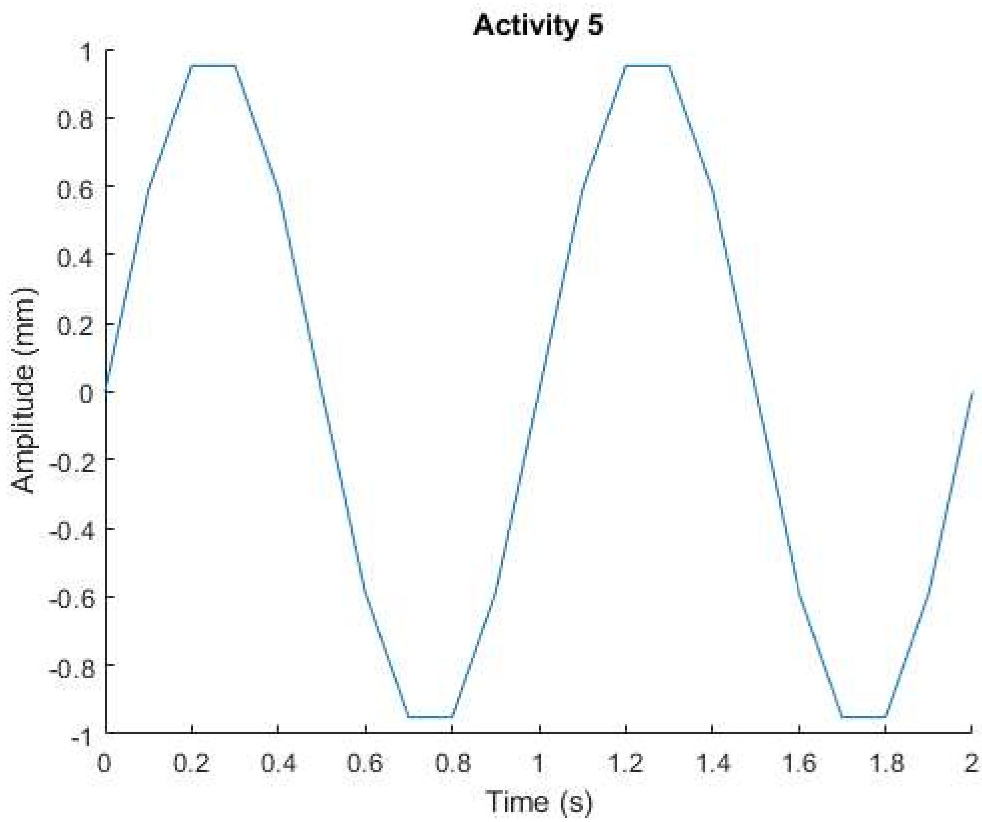
```
function [y] = activity4(x,t)
%ACTIVITY4
%   Take inputs for x and t and put into designated function below and
%   output

y = 2*sin(2*t)-3*x;

end
```

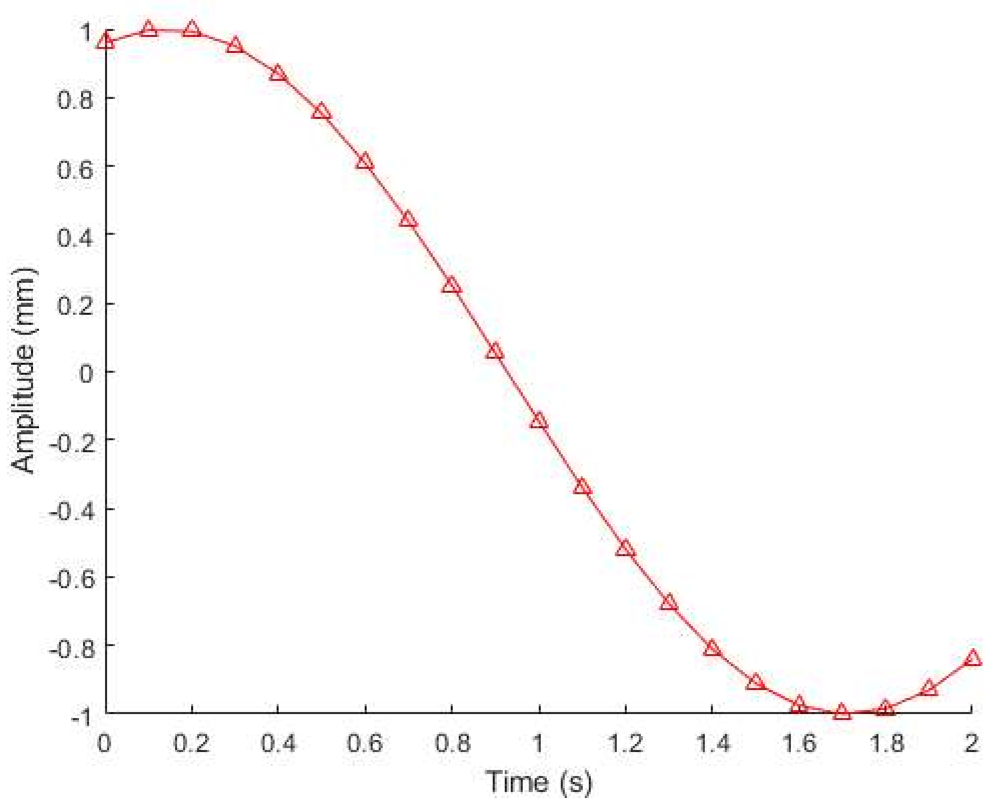
Activity 5

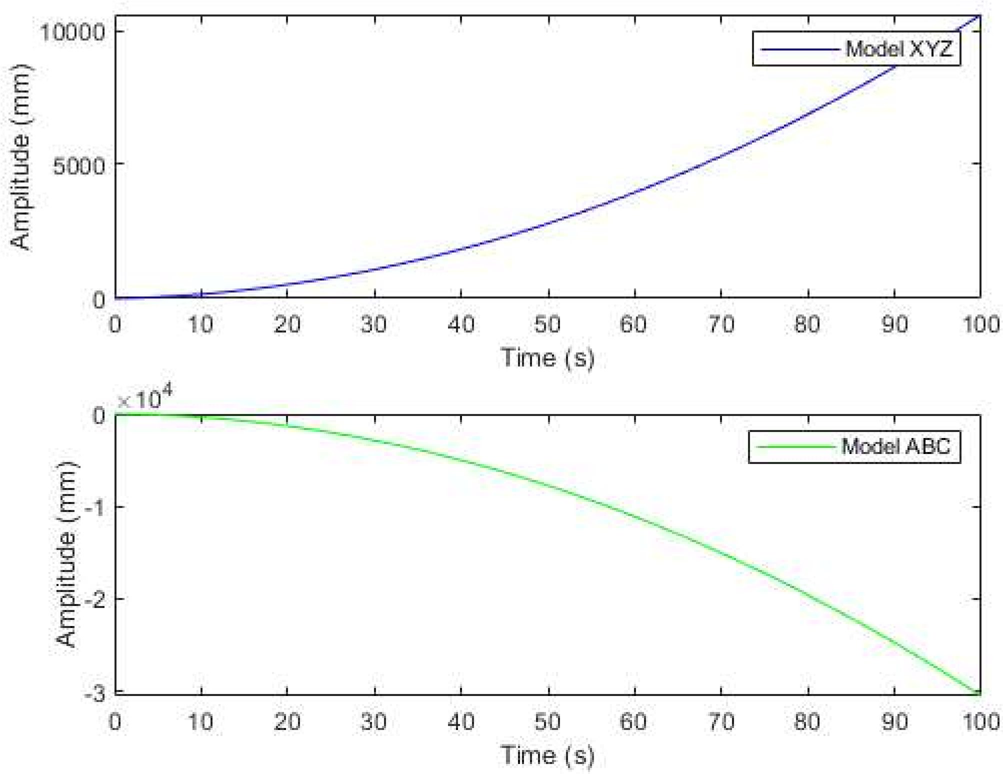
2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we will use the plot command to plot a sin function
% with a defined time step. We will title the figure and label the axes
% accordingly. A new plot will be made on another figure for a specific
% cosine function and the line characteristics will be edited using the
% appropriate name-value pairs. Finally, a 2x1 subplot will be created and
% populated with a separate equation being plotted in each one. These
% graphs will also be individually labeled with line properties affected on
% both. Also, a legend will be created to label the actual functions.

% Setup initial variables and functions
x = 0:0.1:2;
y = sin(2*pi*x);

x1 = 0:0.1:2;
y1 = cos(2*x1+6);

x2 = 0:0.1:100;
y2 = x2.^2+6*x2-2;

x3 = 0:0.1:100;
y3 = -3*x3.^2-5*x3+2;

% Generate "Activity 5"
figure;
hold on;
plot(x,y);
title('Activity 5');
xlabel('Time (s)');
ylabel('Amplitude (mm)');

% Generate solo figure
figure;
hold on;
plot(x1,y1,'Color','r','Marker','^');
xlabel('Time (s)');
ylabel('Amplitude (mm)');

% Generate subplot figure
figure;
hold on;
sp1 = subplot(2,1,1);
plot(x2,y2,'Color','b');
xlabel('Time (s)');
ylabel('Amplitude (mm)');
legend('Model XYZ');

sp2 = subplot(2,1,2);
plot(x3,y3,'Color','g');
xlabel('Time (s)');
```

```
ylabel('Amplitude (mm)');
legend('Model ABC');
```







Published with MATLAB® R2020b

Activity 6

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we are writing code to generate y, a function of x,
% where x goes from 0 to 10 in steps defined by h, which is 0.1. The
% initial value of y will also be set to 1. To assure the function is built
% correctly, the sizes of x and y will be compared (make sure they are the
% same length.

% Setup initial variables
h = 0.1;
x = 0:h:10;

% Generate y
for i = 1:length(x)
    if i > 1
        y(i) = y(i-1)+h*x(i);
    else
        y(i) = 1+h*x(i);
    end
end

if length(x) == length(y)
    disp('Activity complete...');
else
    disp('Check code, activity incomplete');
end
```

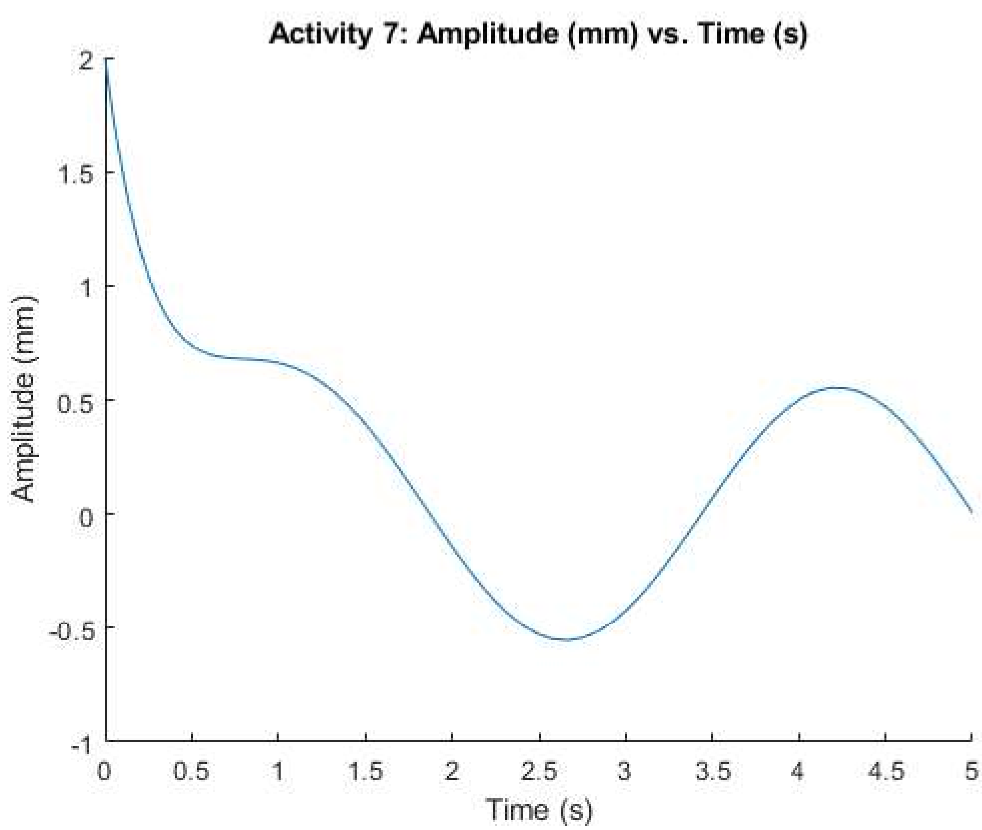
```
Activity complete...
```

Published with MATLAB® R2020b

Activity 7

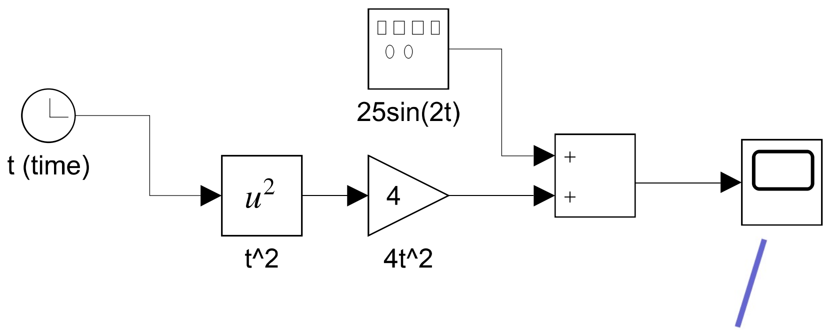
2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we will use ode45 to solve a simple, first order
% differential equation along t, which ranges from 0 to 5. The function is
% also given an initial value of x0 = 2.

% Setup initial variables
t = [0 5];
x0 = 2;

% Solve the ODE
[t,x] = ode45(@(t,x) (10/5)*sin(2*t)-(15/5)*x,t,x0);

% Plot and label results
figure;
hold on;
plot(t,x);
title('Activity 7: Amplitude (mm) vs. Time (s)');
xlabel('Time (s)');
ylabel('Amplitude (mm)');
```

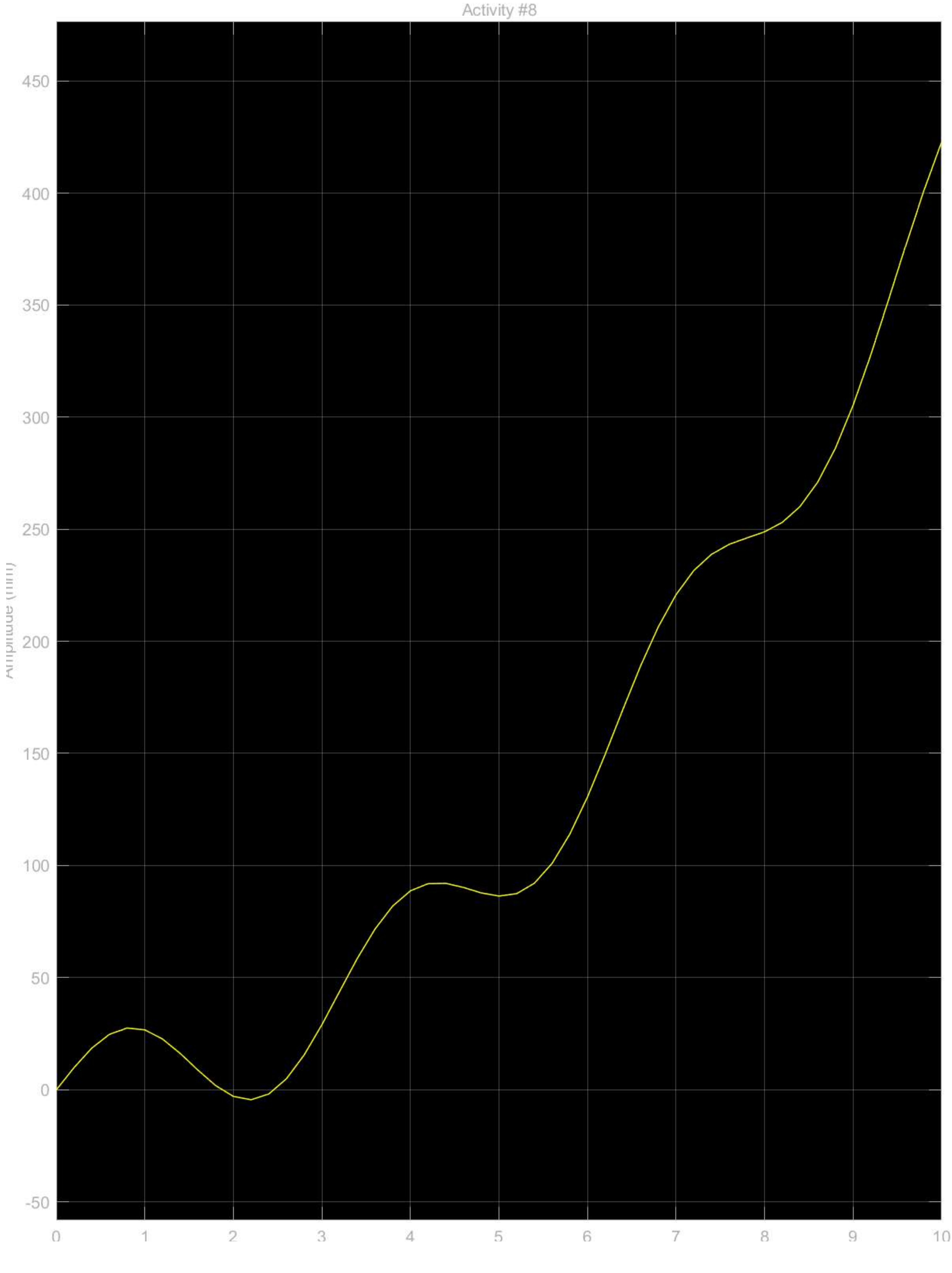


Published with MATLAB® R2020b

Activity Assignment Explanation:
In this activity, we are using only simulink to model the equation y = 25*sin(2*t)+4*t^2. We are then pushing this function to a scope block where we can take measurements at given times. This scope block will also be printed and submitted in the final report (shown below).



At t = 1.5, y = 1.25e+01
At t = 6.25, y = 1.546e+02

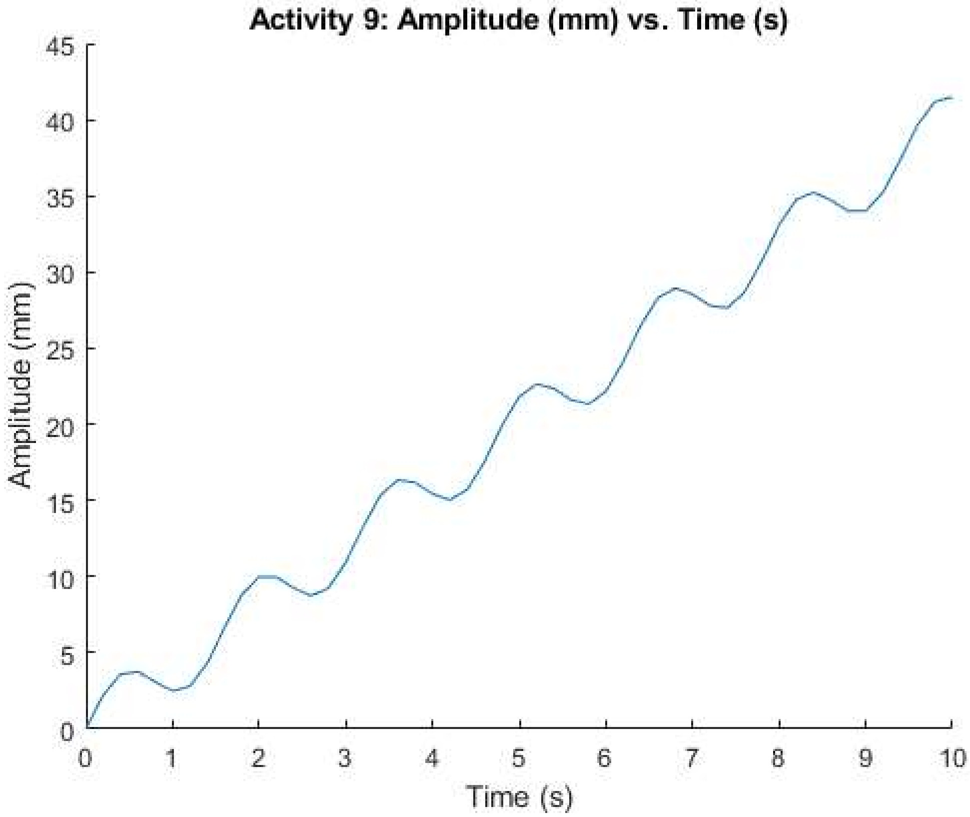


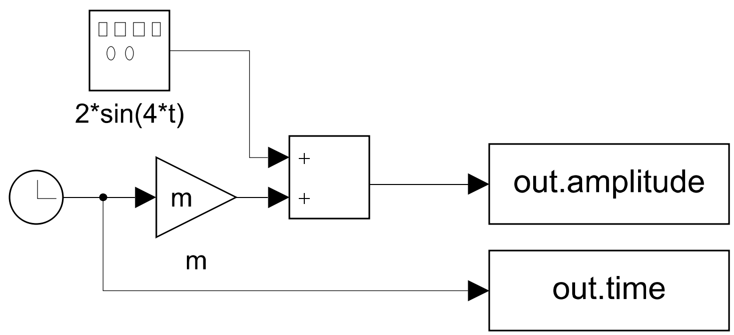
Activity #8
Amplitude (mm)
450
400
350
300
250
200
150
100
50
0
-50
0
1
2
3
4
5
6
7
8
9
10

Activity 9

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we will be creating a Simulink model to model the
% equation y = 2*sin(4*t)+m*t. We will then push the signals to the
% workspace to be used in plotting. Specifically, this script will define
% variable m to a value of 4, run the simulink model with the defined
% function, and plot the results.

% Define initial variables
m = 4;

% Simulate the model
simout = sim('White_Activity9_sim');

% Plot the results
figure;
hold on;
plot(simout.time,simout.amplitude);
title('Activity 9: Amplitude (mm) vs. Time (s)');
xlabel('Time (s)');
ylabel('Amplitude (mm)');

% Note: this lab is having us output the time from the "clock" in the
% simulink environment to graph, but this is unnecessary since "tout" is an output
% from sim inherently
```


Activity 9: Amplitude (mm) vs. Time (s)
Amplitude (mm)
Time (s)
45
40
35
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10

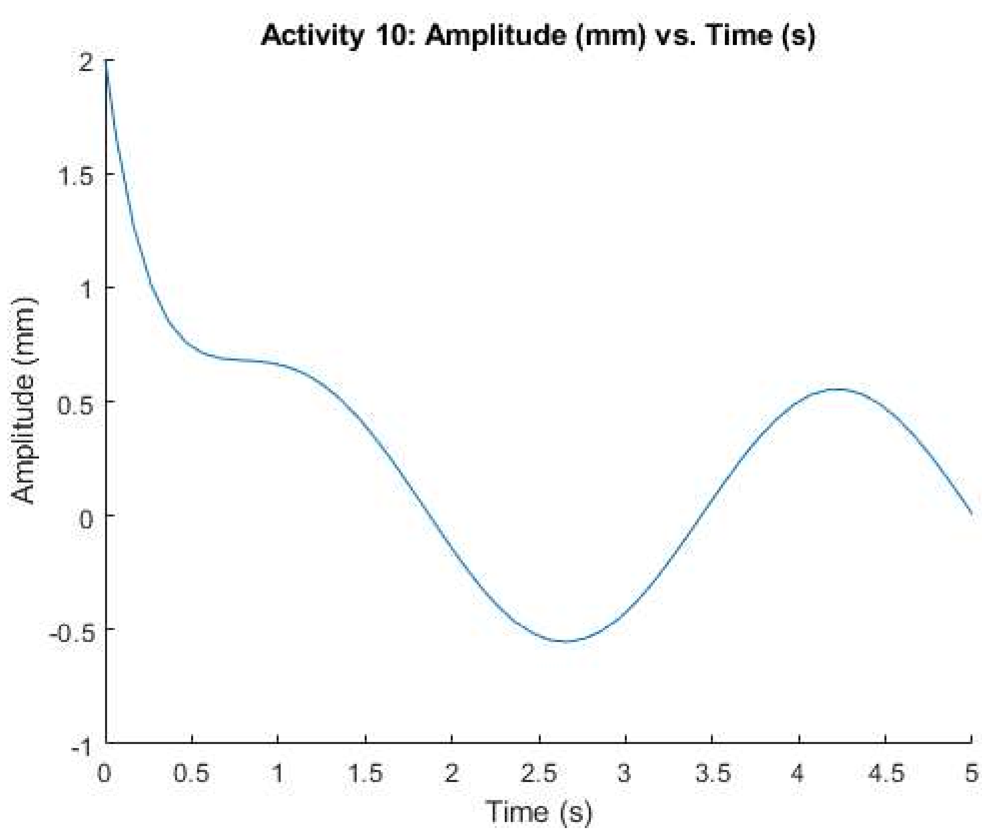


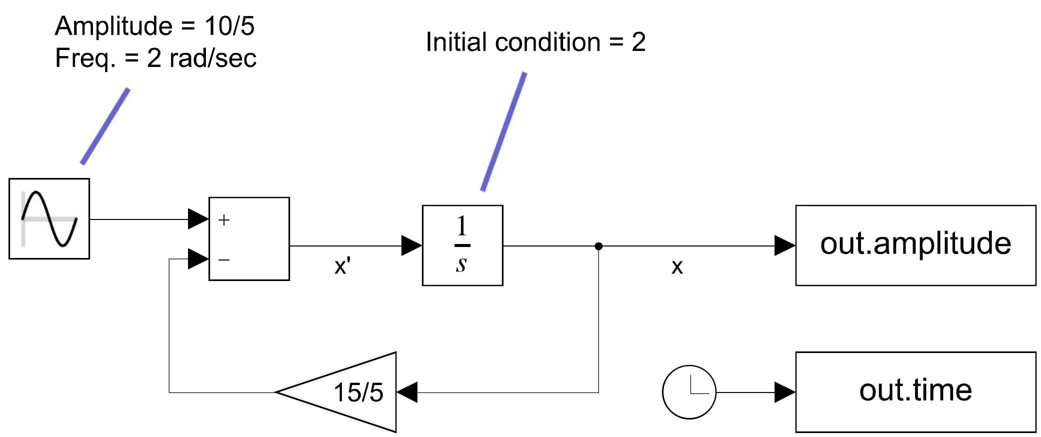
2*sin(4*t)
+
+
m
m
out.amplitude
out.time

Activity 10

2/12/21 Michael White Section 3 / Online

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we are solving an ODE within a simulink model, but
% calling that model again from a script and pushing the results to the
% workspace. From there, the results are taken to be graphed, just like
% what was done on Activiy 9. This activity result can also be compared to
% the result found in Activity 7, and doing so shows that these scripts are
% both successful.

% Run simulation for Activity 10 and output results to 'simout'
simout = sim('White_Activity10_sim');

% Generate figure and plot/label results
figure;
hold on;
plot(simout.time,simout.amplitude);
title('Activity 10: Amplitude (mm) vs. Time (s)');
xlabel('Time (s)');
ylabel('Amplitude (mm)');
```



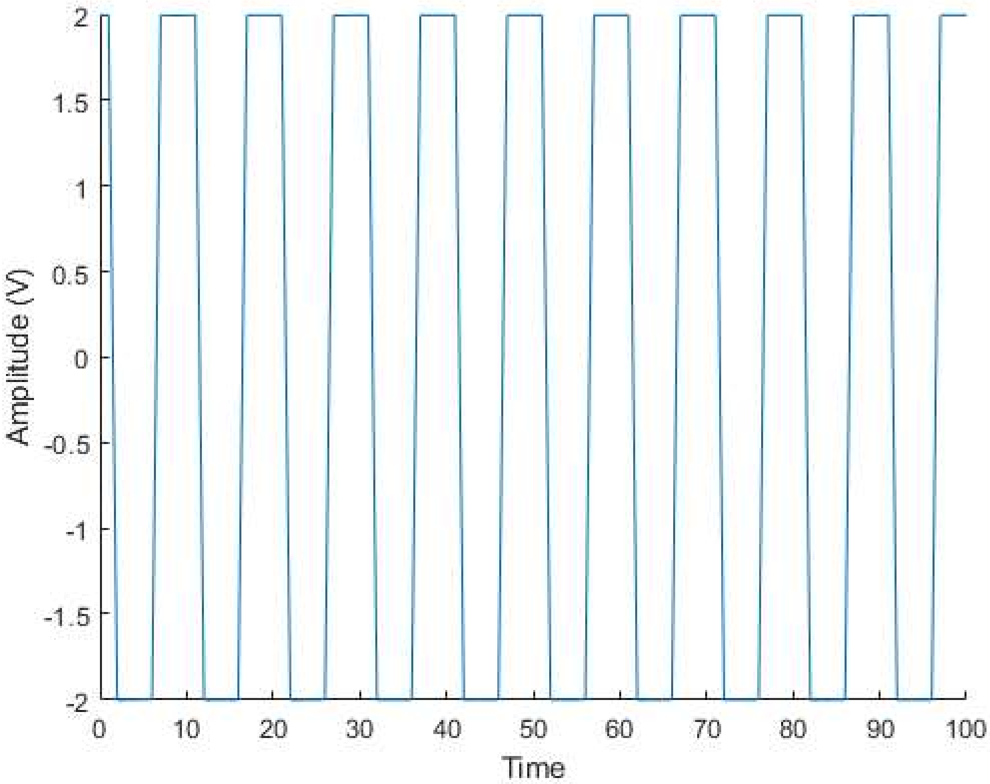


Amplitude = 10/5
Freq. = 2 rad/sec
Initial condition = 2
+
−
x'
1
s
x
out.amplitude
15/5
out.time

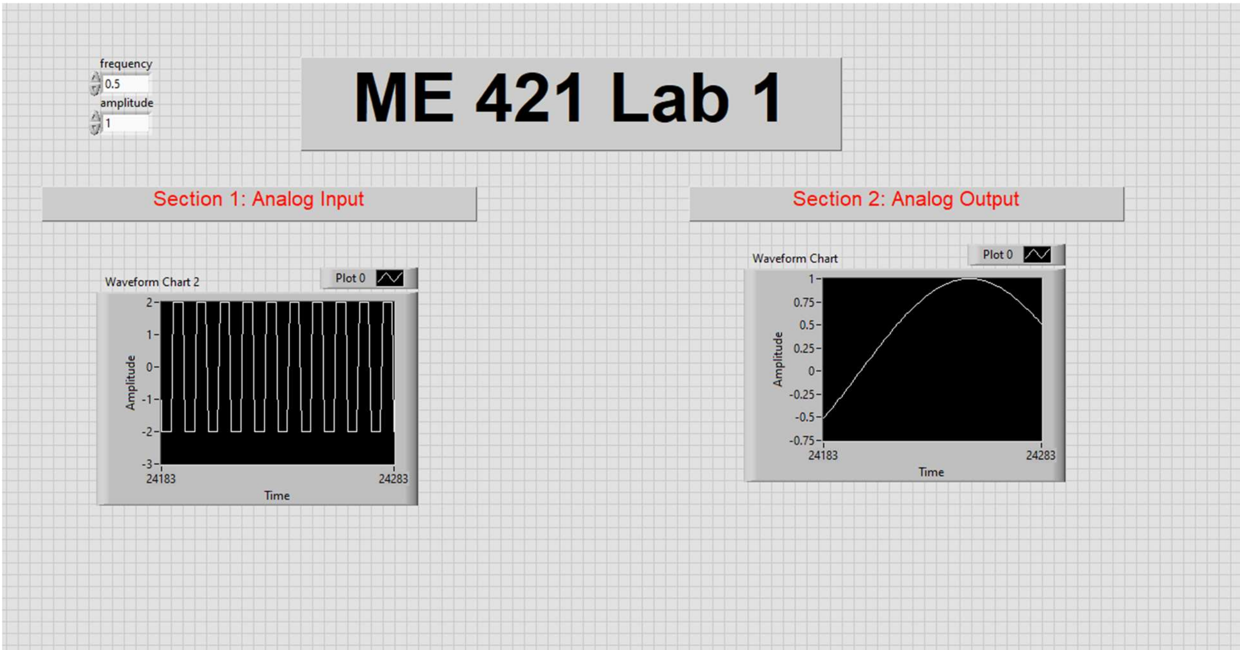
Activity 11

2/12/21 Michael White Section 3 / Section 3 Group B

```
close all;
clear all;
clc;

% Activity Assignment Explanation:
% In this activity, we use the lab equipment to generate a .csv file from
% LabView, which is reading the real-world signals from our lab equipment.
% This system is viewable in the screenshot taken below. The .csv file is
% then taken into this MATLAB code and pulled apart into usable data. This
% data is then plotted and labeled appropriately.

% Import table data from Activity11.csv
data = readtable('Activity11.csv');

% Setup data values from table
Amplitude = data(:,2);
Time = data(:,1);

% Convert tables to array
Amplitude = table2array(Amplitude);
Time = table2array(Time);

% Shift Time by starting value
Time = Time - Time(1);

% Plot data
figure;
hold on;
plot(Time,Amplitude);
xlabel('Time');
ylabel('Amplitude (V)');
```


Amplitude (V)
Time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
10
20
30
40
50
60
70
80
90
100

Activity 11 Screenshot:



Activity 12 Screenshot:

Here, the system was fixed using a “Waveform Chart” rather than a “Waveform Graph”. This success can be seen in the output of the waveform chart: