LAB #0 Modeling Dynamics System Using MATLAB and Simulink

PREPARED BY:
HADEN SANGREE

IN FULFILLMENT OF THE REQUIREMENTS FOR:
MTRE 4002L

MARCH 4th, 2015

College of Engineering
Kennesaw State University
Marietta, GA

Contents

Figures

1. INTRODUCTION

The goal of this lab is to further familiarize us with MATLAB and Simulink using our knowledge of Control Systems being taught in MTRE 4001. More specifically, we used our knowledge of circuit modeling to visualize a plot. Two techniques were used to realize this, the numerical approach, and the analytical approach. The former is a methodology that allows the user to realize a rough estimate of the output of a system and can be very powerful in analyzing complicated systems than cannot be analyzed using the latter approach, analytical. The analytical approach gives the user the exact output without expected error baked in. This is also a great tool but can be difficult to use.

2. QUESTION 1

Question 1 is where we were tasked with modeling the RC circuit given above, the bulk of the code was provided in the manual so just some formatting we used to furth clarify the figure. However, this code uses a sine wave input rather than a step wave input.

```
clc;
clear all;
close all;

R=30; %Resistance
C=0.0237; %Capacitance
dt=0.01; %time step size
t=0:dt:10; %simulation time
v=0.1*sin(2*pi*t); %input voltage
q(1)=0; %setting initial condition

for i=1:length(t) %Euler's method
      dq(i)=v(i)/R-q(i)/(R*C);
      q(i+1)=dq(i)*dt + q(i);
end

i=dq; %current
plot(t,i) % ploting current vs. time
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('Current (A)','FontSize',12,'FontWeight','bold','Color','b')
title('RC Circuit','FontSize',12,'FontWeight','bold')
```

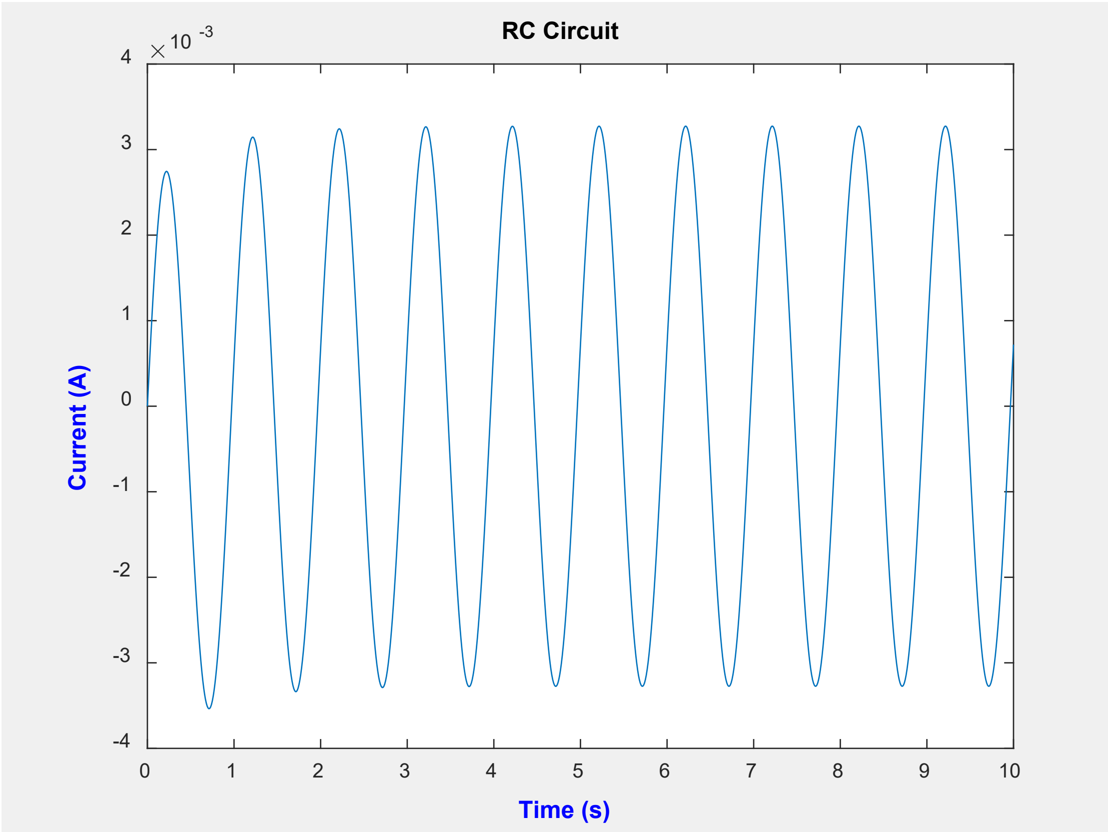

Figure A

3. QUESTION 2

In question 2 the same RC circuit was model but now strictly with Simulink. As we can see Figure C. the output is very similar to the MATLAB code's output in Figure A. The Simulink model can be seen in Figure B.

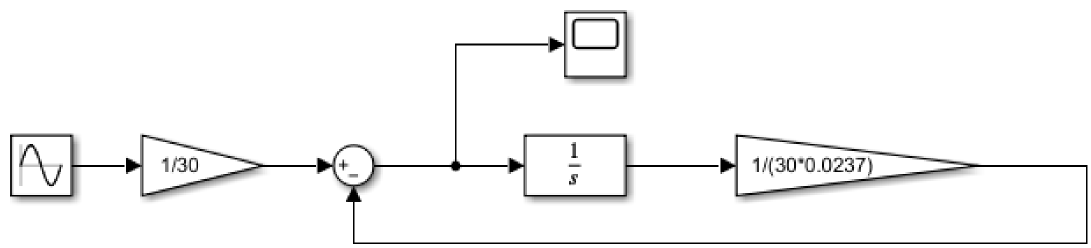

Figure B

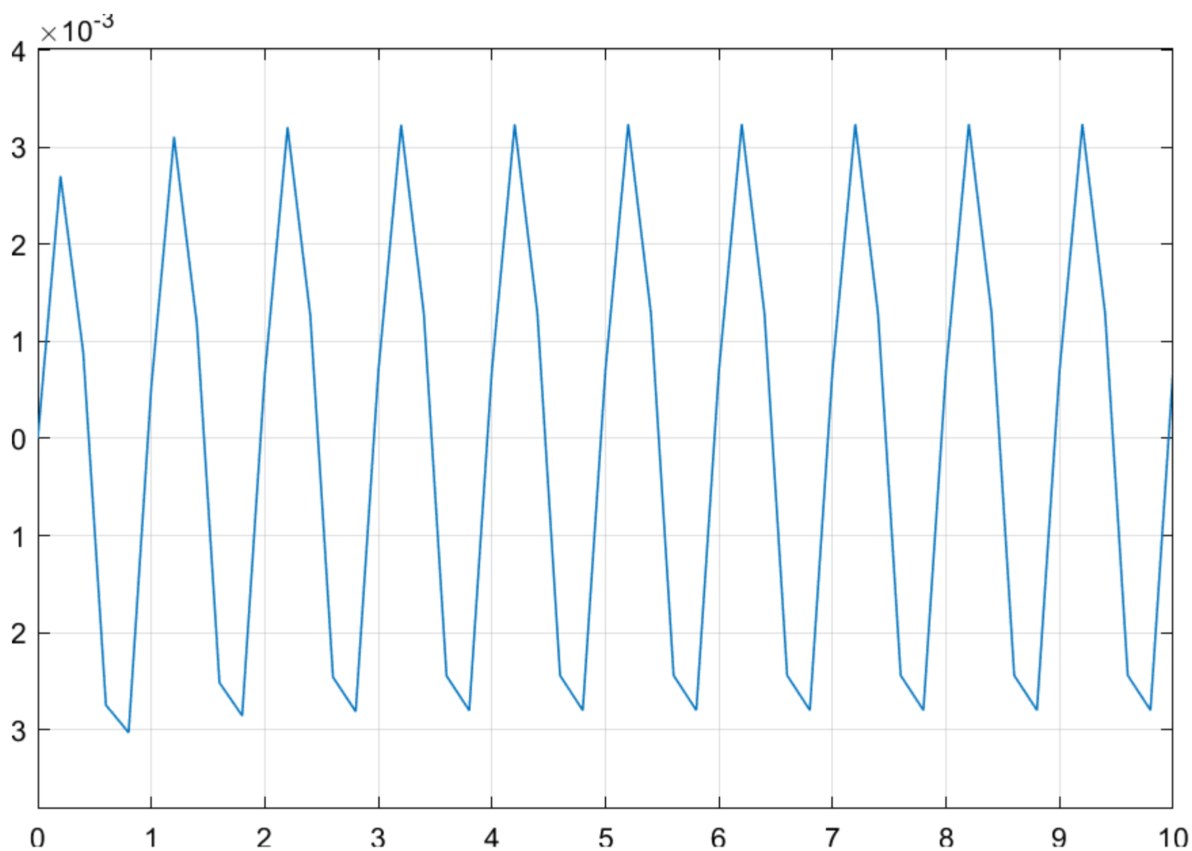

Figure C

4. QUESTION 3

In question 3 we utilized Euler's method to approximate the output of our function. Below is the MATLAB code used to generate Figure D. This figure displays the unit step response of the code below. The time step size is very narrow, thus giving a what appears to be a very bold blue trace; however those are lots of blue circles at each test interval.

```
clc;
clear all;
close all;

dt=0.01; %time step size
t=0:dt:10; %simulation time
v=0.1*sin(2*pi*t); %input voltage
u=ones(1,length(t)); %u(t) is the step input
dy(1)=0; %initializing
y(1)=0; %initializing

for i=1:length(t) %Euler's method
      ddy(i)=u(i) - 3*dy(i) - 2*y(i);
      dy(i+1)=ddy(i)*dt + dy(i);
      y(i+1)=dy(i)*dt + y(i);
end

y(end)=[]; %deleting last element in vector
plot(t,y,'--ob') %plotting y value vs. time
grid on %turns plot grid marks on
txt = ['\delta(t) = ' char(10) num2str(dt)]; %ignore the sqiggle
text(5,.25,txt) %adds text to center of plot
xticks(0:2:10) %sets tick marks on x axis
legend('Numerical Solution') %legend
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('Y','FontSize',12,'FontWeight','bold','Color','b')
title('Unit Step Function','FontSize',12,'FontWeight','bold')
```

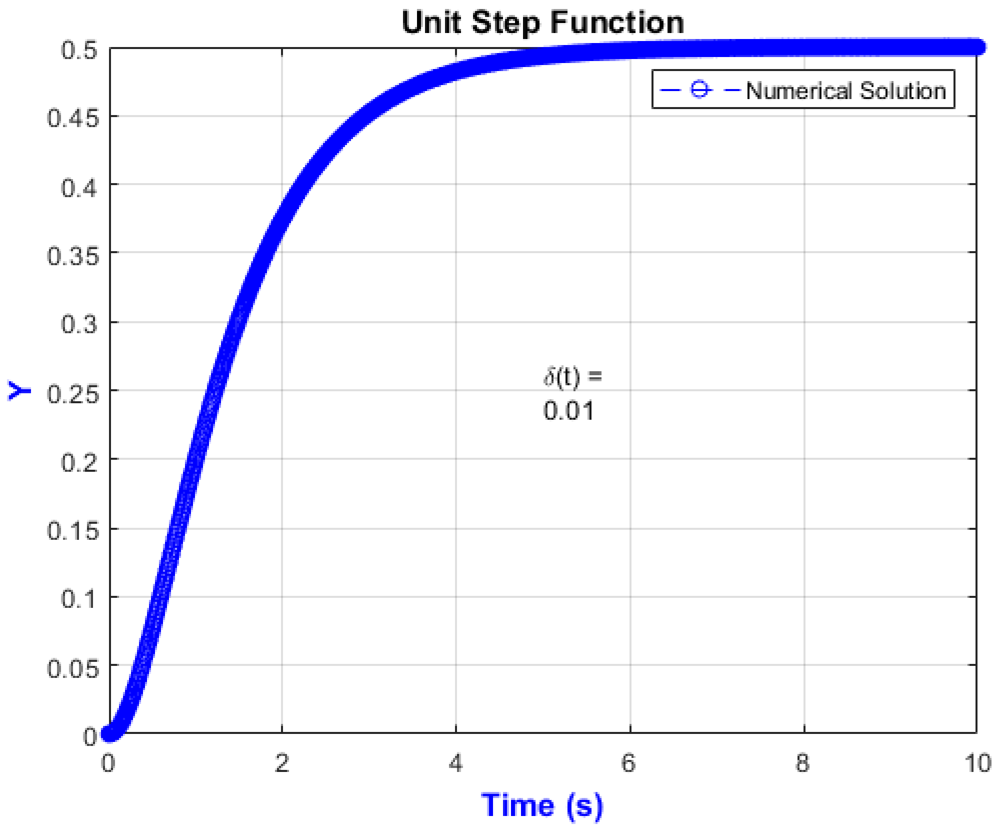

Figure D

5. QUESTION 4

Below is the MATLAB code that was used to generate the three figures below. The three figures below show the output of three values for "dt" in the MATLAB code. The values are commented next to what would be line 5, "0.5s then 0.1s then 0.01s". The figures below do a great job at displaying the different of the analytical and numerical approaches. When the time step size decreases the numerical approach's results comes very close to the numerical approach.

```
clc;
clear all;
close all;

dt=0.1; %time step size --- 0.5s then 0.1s then 0.01s
t=0:dt:10; %simulation time
v=0.1*sin(2*pi*t); %input voltage
u=ones(1,length(t)); %u(t) is the step input
dy(1)=0; %initializing
y(1)=(0); %initializing

for i=1:length(t) %Euler's method
      ddy(i)=u(i) - 3*dy(i) - 2*y(i);
      dy(i+1)=ddy(i)*dt + dy(i);
      y(i+1)=dy(i)*dt + y(i);
      z(i) = 0.5*u(i) + 0.5*exp(-2*t(i)) - exp(-t(i)) %analytical equation
```

```
end
y(end)=[]; %deleting last element in vector
error = z-y %calculate error between numerical and analytical

subplot(1,2,1) %plot in first position
plot(t,y,'--ob') %plotting y value vs. time
grid on %turns plot grid marks on
txt = ['\delta(t) = ' char(10) num2str(dt)];
text(5,.25,txt) %adds text to center of plot
xticks(0:2:10) %sets ticks to x axis
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('Y','FontSize',12,'FontWeight','bold','Color','b')
title('Unit Step Function','FontSize',12,'FontWeight','bold')

hold on %allows to plot two functions on one plot

plot(t,z,'--*r') %plotting z value vs. time
legend('Numerical Solution','Analytical Solution')

hold off %now plotting on seperate plot

subplot(1,2,2) %plot in second position
plot(t,error,'--ob') %plotting error value vs. time
grid on %turns plot grid marks on
xticks(0:2:10) %sets ticks to x axis
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('error','FontSize',12,'FontWeight','bold','Color','b')
title('Unit Step Function','FontSize',12,'FontWeight','bold')
```

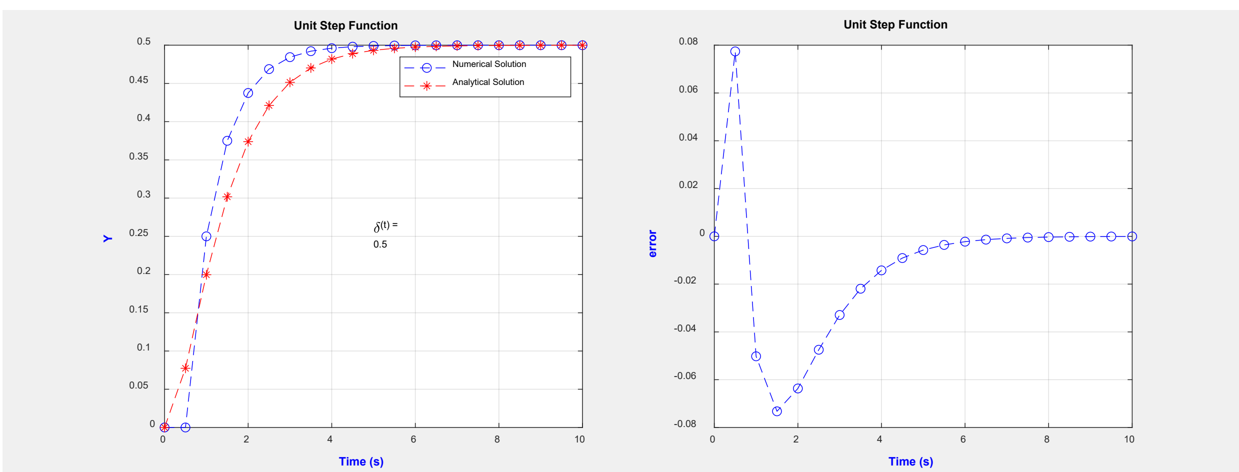

Figure E

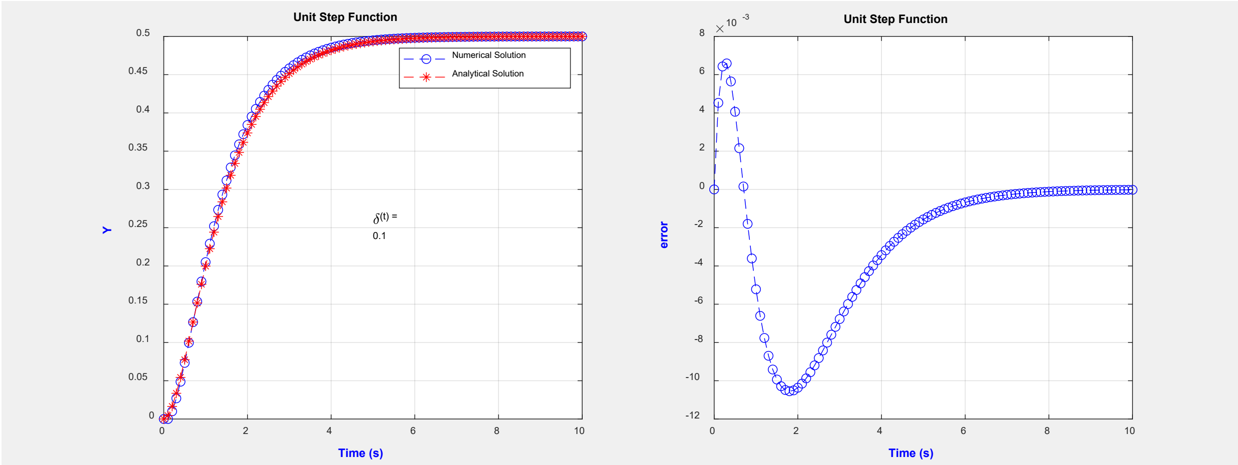

Figure F

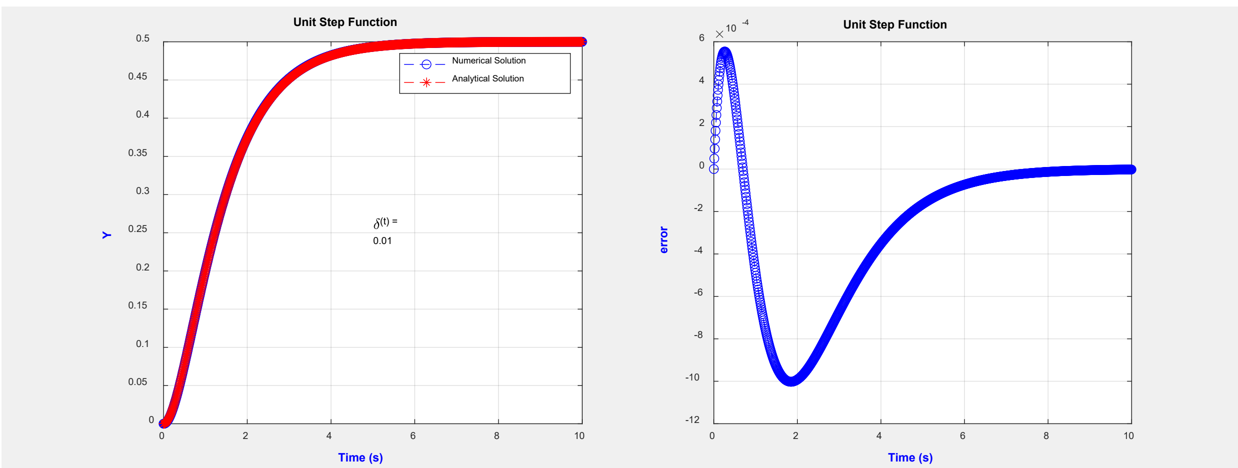

Figure G

<u>6. QUESTION 5</u>

Question 5 takes the same shape as question 4 but not with a sine input and with a different analytical solution. The same phenomenon is found here with the time step size. As the time step size decreases the error also decreases. This is important comparing analytical solutions to numerical solutions.

```
clc;
clear all;
close all;
dt=0.5; %time step size --- 0.5s then 0.1s then 0.01s
t=0:dt:10; %simulation time
u=sin(2*pi*t); %u(t) is the sine input
dy(1)=0; %initializing
y(1)=(0); %initializing

for i=1:length(t) %Euler's method
```

```
      ddy(i)=u(i) - 3*dy(i) - 2*y(i);
      dy(i+1)=ddy(i)*dt + dy(i);
      y(i+1)=dy(i)*dt + y(i);
      z(i) = (((2*pi*exp(-t(i))) / (4*pi^2+1)) -
      ((pi*exp(-2*t(i))) / (2*pi^2+2)) - ((3*pi*cos(2*pi*t(i))-
      sin(2*pi*t(i))+2*pi^2*sin(2*pi*t(i))) / ((pi^2+1)*(8*pi^2+2))))
end
y(end)=[];%deleting last element in vector
error = z-y %calculate error between numerical and analytical
subplot(1,2,1) %plot in first position
plot(t,y,'--ob') %plotting y value vs. time
grid on %turns plot grid marks on
txt = ['\delta(t) = ' char(10) num2str(dt)];
text(5,.25,txt) %adds text to center of plot
xticks(0:2:10) %sets ticks to x axis
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('Y','FontSize',12,'FontWeight','bold','Color','b')
title('Sine Function','FontSize',12,'FontWeight','bold')
hold on %allows to plot two functions on one plot
plot(t,z,'--*r') %plotting z value vs. time
legend('Numerical Solution','Analytical Solution')
hold off %now plotting on seperate plot
subplot(1,2,2) %plot in second position
plot(t,error,'--ob') %plotting error value vs. time
grid on %turns plot grid marks on
xticks(0:2:10) %sets ticks to x axis
xlabel('Time (s)','FontSize',12,'FontWeight','bold','Color','b')
ylabel('error','FontSize',12,'FontWeight','bold','Color','b')
title('Sine Function','FontSize',12,'FontWeight','bold')
```

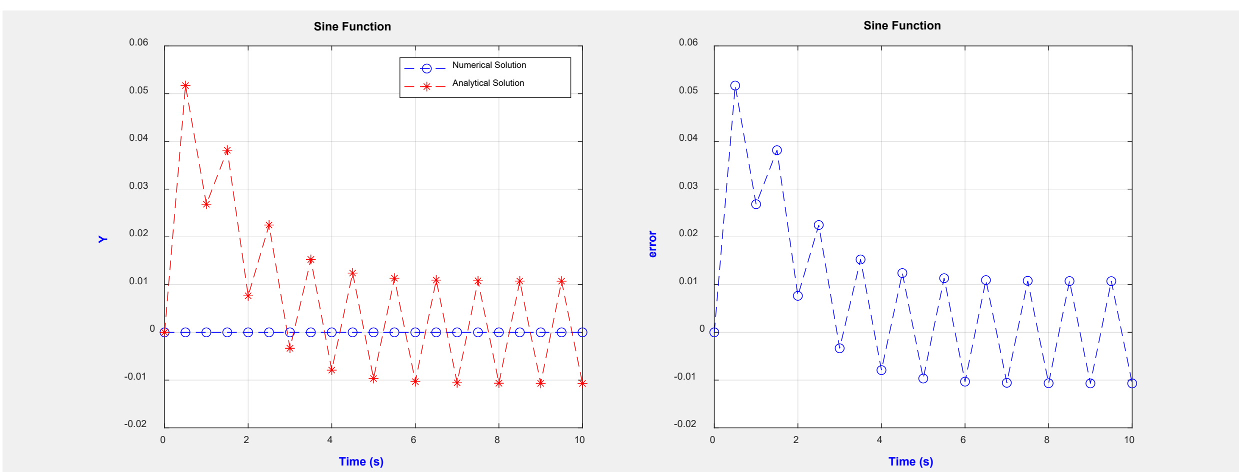

Figure H

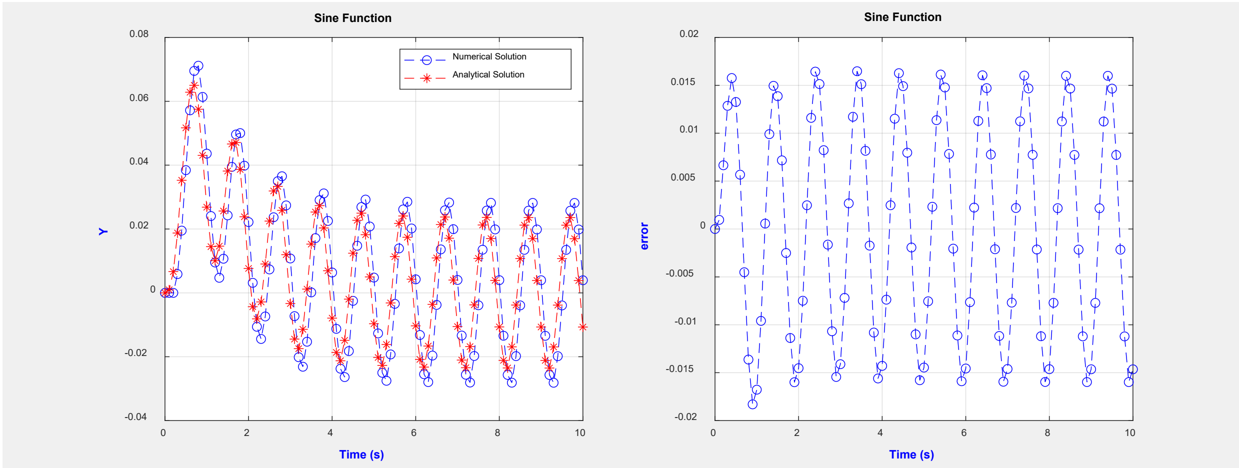

Figure I

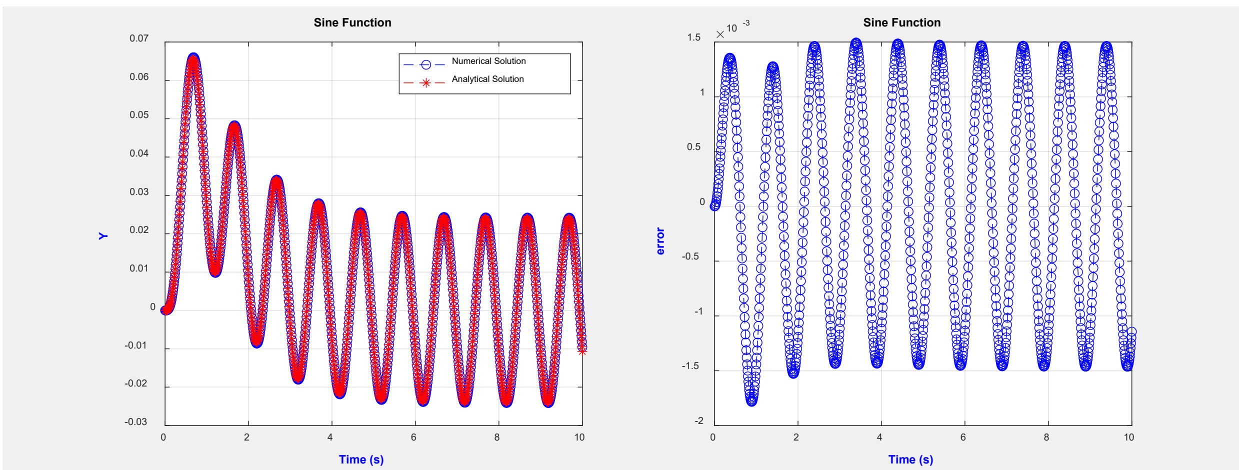

Figure J

7. QUESTION 6

Lastly, in question 6 the function in question 3 was modeled using Simulink with a unit step input and a sine wave input. Figure K and Figure M display the Simulink model, the former with a step input and the latter with a sine wave input. Figure L and Figure N display the output of Figure K and Figure respectively.

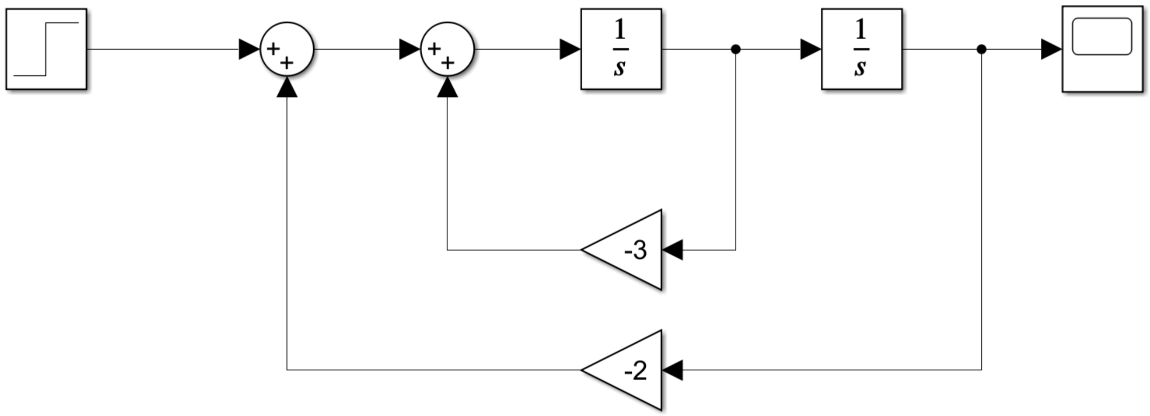

Figure K

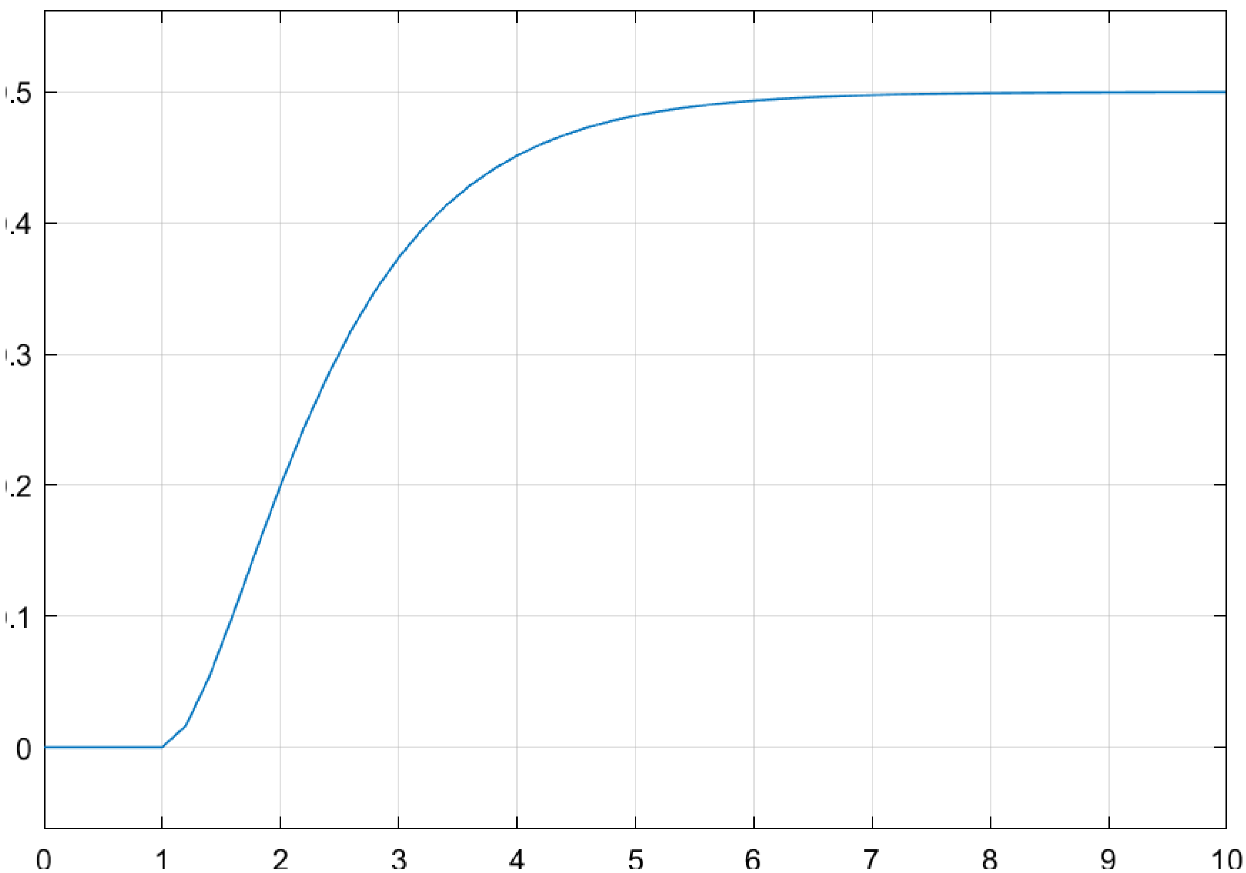

Figure L

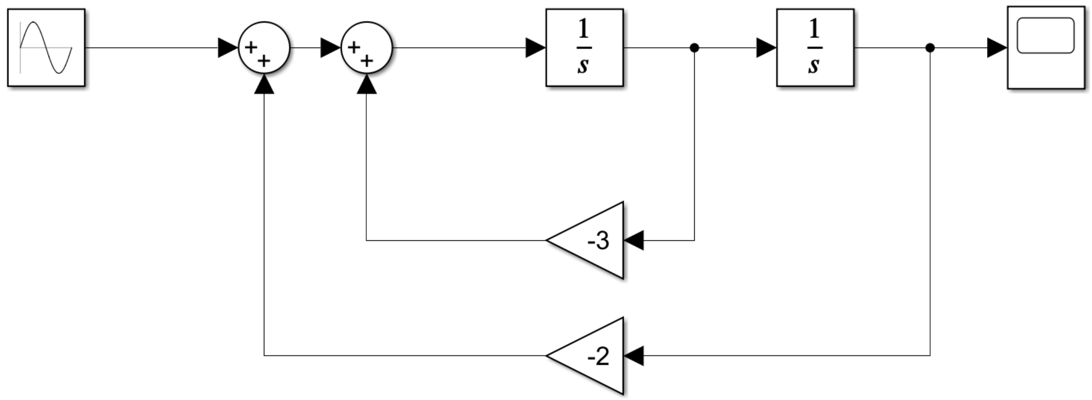

Figure M

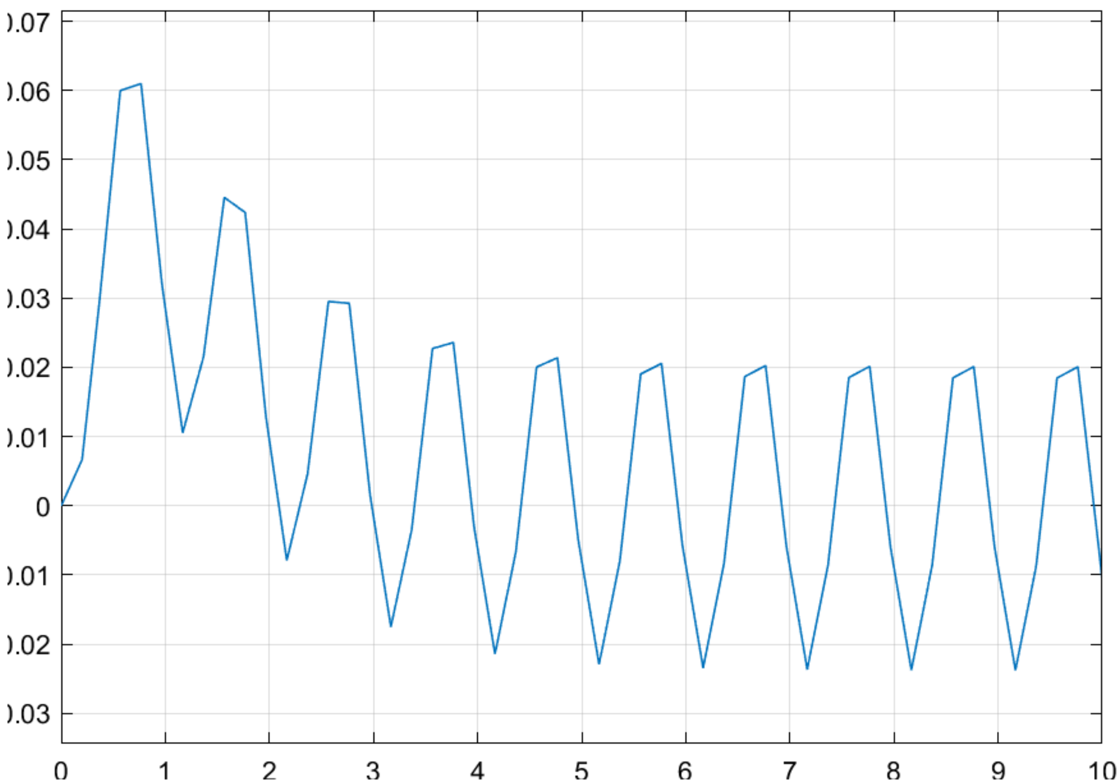

Figure N

<u>8. CONCLUSION</u>

In conclusion, this lab was a great exercise with showing the power of modeling and plotting with MATLAB and Simulink. There are many useful functions that help with getting straight to the answer with minimum software and syntax problems. Also, the difference shown between the numerical and analytical approaches on the plot was very helpful. Using Euler's method to generate a numerical solution was very insightful on the power of that technique.