ECE 220-203

Lab 1

Jordan Ditzler

9/15/19

Jordan Ditzler
ECE 220-203

Objective

The objective of this lab is to review how sinusoidal signals are calculated in MATLAB and how to plot these signals in MATLAB.

MATLAB code

See Appendix A.

Results

Figure 1

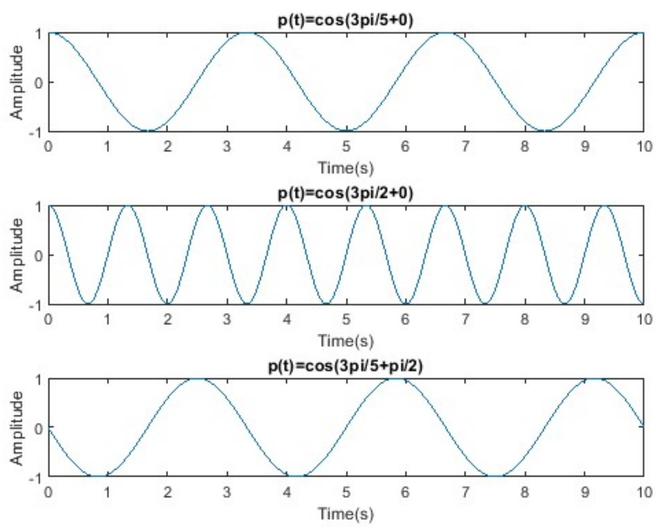


Figure 2

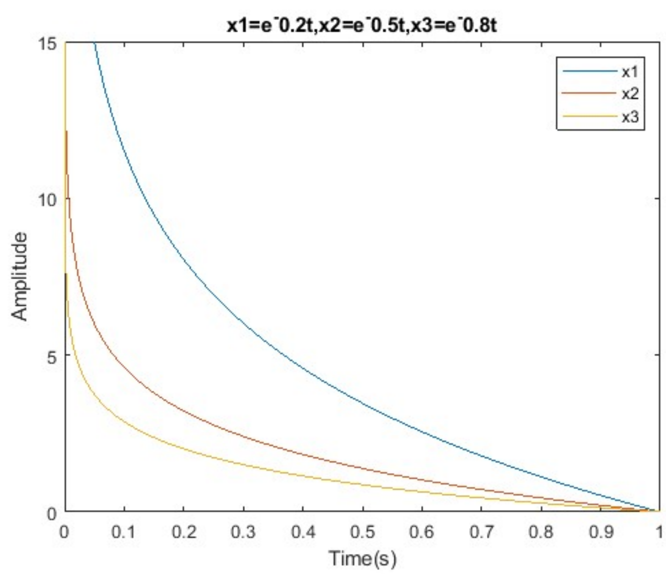


Figure 3

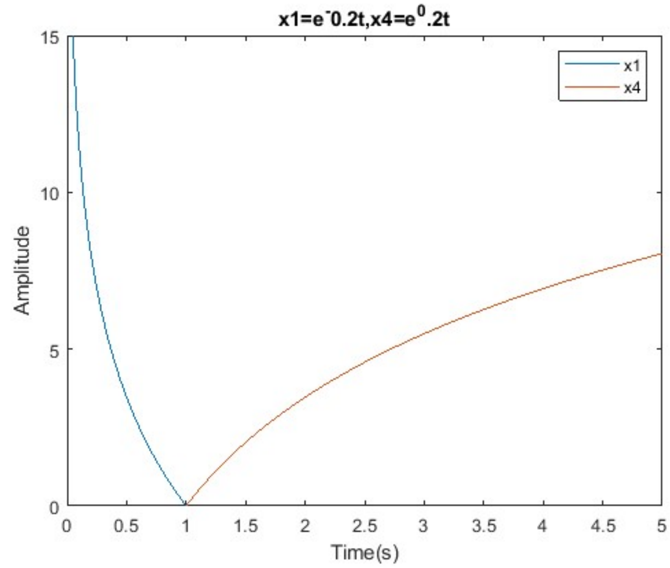


Figure 4

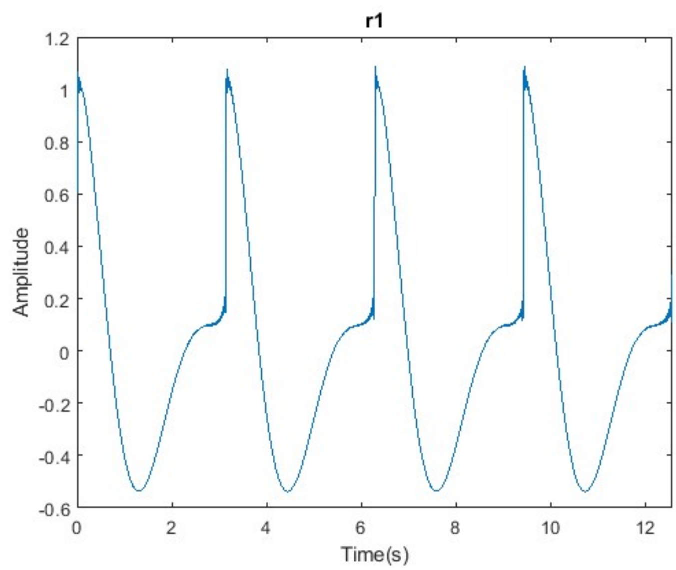


Figure 5

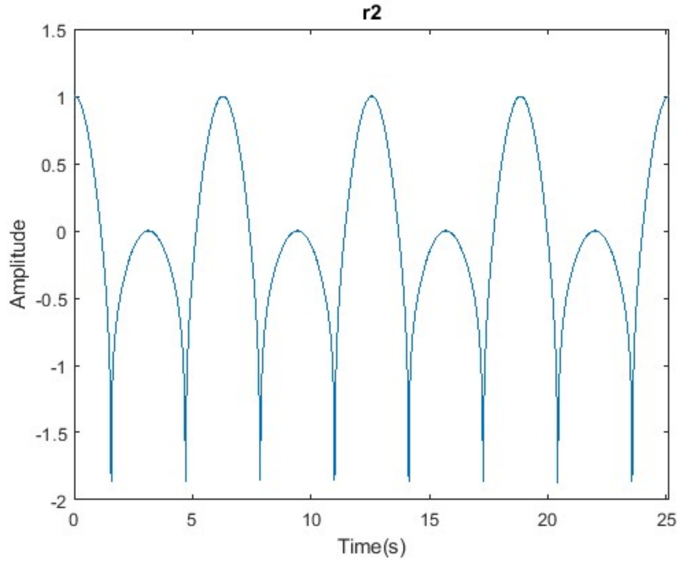
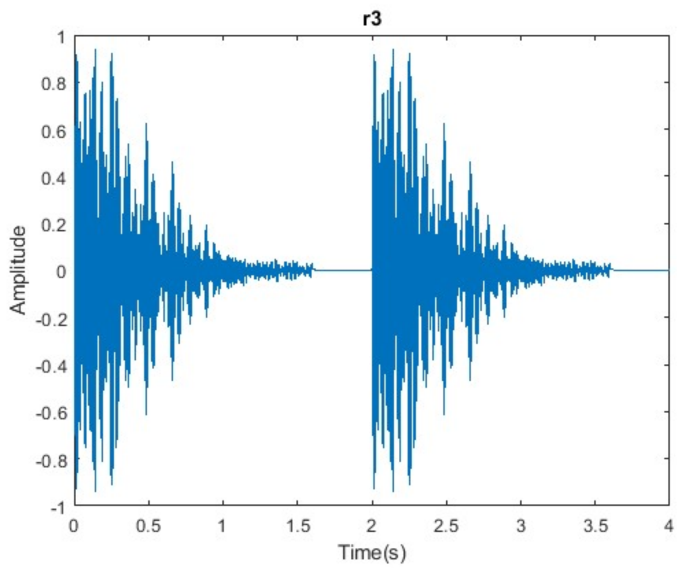


Figure 6



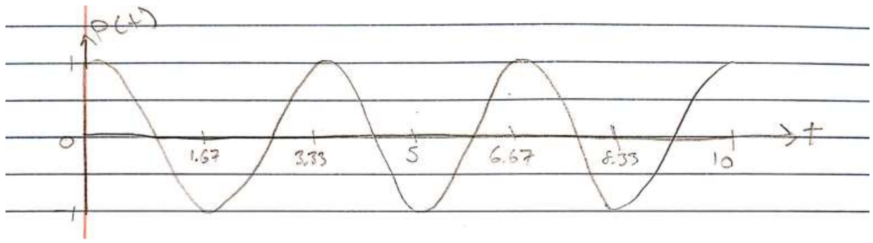
Jordan Ditzler
ECE 220-203

Questions

1.1

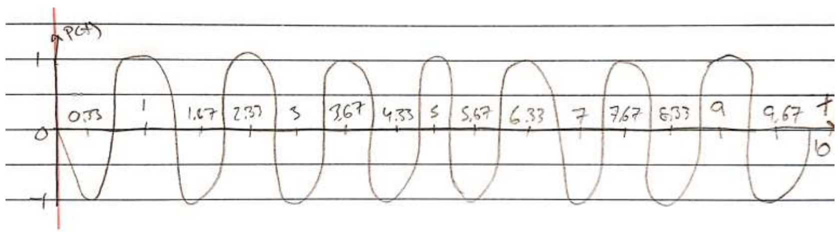
a)

i)



Frequency: $f\ =\ 0.3Hz$
Period: $T =\ 3.33s$

ii)



Frequency: $f =\ 0.75Hz$
Period: $T =\ 1.33s$

c) Period: $T =\ 3.33s$
d) Period: $T =\ 1.33s$
e) Period: $T =\ 3.33s$
f) Increasing ω_0 makes the signal oscillate more, thus making more waves in the same amount of time compared to a lower ω_0.
Decreasing ω_0 does the opposite; it makes less waves in the same amount of time compared to a higher ω_0.
The variable ω_0 is the angular frequency of a signal, so $\omega_0 = 2\pi * f$.
In order to get the period T, we get the inverse of f. First, we need to get f from ω_0. Using the above formula, we derive $f = \frac{\omega_0}{2\pi}$. Then we calculate T by finding the inverse of the frequency, so $T = \frac{1}{f}$.

1.2

d) As α varies, the plots (Figure 2, 3) can change dramatically. When α is negative, it loses amplitude as time passes. The plot is focused more vertically and loses much more in amplitude over a shorter period than if α would gain if it were positive. When α is positive, it gains amplitude as time passes. The plot is focused more horizontally and gains much less in amplitude over a longer period than if α would lose if it were negative. Both negative and positive α converge to the point (1,0), which is to be expected as $e^0 = 1$. The large or smaller effects how rapidly each plot gains or loses amplitude depending on if α is positive or negative respectively.

Discussion/Conclusion

This lab went smoothly, and I had no issues completing it. It has been awhile since I last used MATLAB, so it took some time before I became re-adjusted. All the calculations dealing with period, frequency, angular frequency, and exp() are familiar to me and I am confident my results are accurate. All the formulas used in this lab were already known, so no new formulas were discovered. I never figured out how to properly multiply the matrices together, which is the ideal and most efficient method of calculating sinusoidal signals; I instead used a for loop. I am still interested to figure out the method if time permits.

Appendix A

```
%1.1
figure(1)                       %designates plot as figure 1
t=0:0.01:10;                    %sets t as 1001 long vector
subplot(311)                    %sets plot as 1x3 grid of subplots
p1=cos(3*pi/5*t);               %calculates sinusoidal wave with w=3pi/5
plot(t,p1)                      %plots waveform vs t and labels it
xlabel('Time(s)')
ylabel('Amplitude')
title('p(t)=cos(3pi/5+0)')
subplot(312)                    %shifts down 1 subplot
p2=cos(3*pi/2*t);               %calculates sinusoidal wave with w=3pi/2
plot(t,p2);                     %plots new waveform and labels it
xlabel('Time(s)')
ylabel('Amplitude')
title('p(t)=cos(3pi/2+0)')
subplot(313)                    %shifts down 1 subplot
p3=cos(3*pi/5*t+pi/2);          %calculates sinusoidal wave with w=3pi/5 and
phase=pi/2
plot(t,p3)                      %plots new waveform and labels it
xlabel('Time(s)')
ylabel('Amplitude')
title('p(t)=cos(3pi/5+pi/2)')
%%
%1.2b
figure(2)                                   %designates plot as figure 2
t=0:0.01:15;                                %sets t as 1501 long vector
x1=exp(-0.2.*t);                            %calculates x1 as an exponential
with power of -0.2
x2=exp(-0.5.*t);                            %calculates x2 as an exponential
with power of -0.5
x3=exp(-0.8.*t);                            %calculates x3 as an exponential
with power of -0.8
plot(x1,t,x2,t,x3,t)                        %plots x1, x2, and x3 on the same
graph with labels and a legend
xlabel('Time(s)')
ylabel('Amplitude')
title('x1=e^-0.2t,x2=e^-0.5t,x3=e^-0.8t')
legend('x1','x2','x3')
%%
%1.2c
figure(3)                                   %designates plot as figure 3
t=0:0.01:15;                                %sets t as 1501 long vector
x1=exp(-0.2.*t);                            %calculates x1 as an exponential
with power of -0.2
x4=exp(0.2.*t);                             %calculates x4 as an exponential
with power of 0.2
plot(x1,t,x4,t)                             %plots x1 and x4 on the same
graph with labels and a legend
xlabel('Time(s)')
ylabel('Amplitude')
xlim([0 5])
title('x1=e^-0.2t,x4=e^0.2t')
legend('x1','x4')
%%
%2b
```

Jordan Ditzler
ECE 220-203

```
figure(4)                                        %designates plot as
figure 4
load('parameterSetOne.mat')                      %loads variables
from .mat file
tmax=8*pi/w0;                                    %calculates tmax as 4
periods of the signal
t=0:Tsample:tmax;                                %generates time vector
r1 = 0;                                          %initializes r1
for n=0:length(Cn)-1                             %calculates signal r1 via
for loop
    r1=(Cn(n+1)*cos((n+1)*w0*t+thetan(n+1)))+r1;
end
plot(t,r1)                                       %plots r1 vs with labels
xlim([0,tmax])
xlabel('Time(s)')
ylabel('Amplitude')
title('r1')
%%
%2c
figure(5)                                        %designates plot as
figure 5
load('parameterSetTwo.mat')                      %loads variables
from .mat file
tmax=8*pi/w0;                                    %calculates tmax as 4
periods of the signal
t=0:Tsample:tmax;                                %generates time vector
r2 = 0;                                          %initializes r2
for n=0:length(Cn)-1                             %calculates signal r2 via
for loop
    r2=(Cn(n+1)*cos((n+1)*w0*t+thetan(n+1)))+r2;
end
plot(t,r2)                                       %plots r2 vs t with
labels
xlim([0,tmax])
xlabel('Time(s)')
ylabel('Amplitude')
title('r2')
%%
%2d
figure(6)                                        %designates plot as
figure 6
load('parameterSetThree.mat')                    %loads variables
from .mat file
tmax=4*pi/w0;                                    %calculates tmax as 2
periods of the signal
t=0:Tsample:tmax;                                %generates time vector
r3 = 0;                                          %initializes r3
for n=0:length(Cn)-1                             %calculates signal r3 via
for loop
    r3=(Cn(n+1)*cos((n+1)*w0*t+thetan(n+1)))+r3;
end
plot(t,r3)                                       %plots r3 vs t with
labels
xlim([0,tmax])
xlabel('Time(s)')
ylabel('Amplitude')
title('r3')
```

```
soundsc(r3)                                              %plays signal r3 as a
sound
```