Lab 02: Complex Exponentials

Henry Mackay

ECE 3220

Lab Section 30

GWID: 36332480

Tuesday, October 3rd, 2022

Warmup

4.1- one_cos function

```
function [xx,tt] = one_cos(A,ff,p,dur)
   fs=2*ff;
   tt = 0:1/(2*40000):dur;
   xx = A.*cos(2*pi*tt+p);
end
```

4.2- syn_sin function

```
function [xx,tt] = syn_sin(fk, Xk, fs, dur, tstart)
%SYN_SIN Function to synthesize a sum of cosine waves
if nargin<5, tstart=0, end
ff= max(fk)*2;
tt = tstart:1/(fs*ff):dur;%-- gives fs samples per period
xx=zeros(1,length(tt));
for i = 1:length(fk)
       phase=atan2(imag(Xk(i)),real(Xk(i)));
       A=sqrt(imag(Xk(i))*imag(Xk(i)) + real(Xk(i))*real(Xk(i)));
       t_domain=A*cos(fk(i).*tt+phase);
       xx=xx+t_domain;
end
```

```
% #### Testing for one_cos function ####
A=95;
w=200*pi; %radians/s
p=pi/5; %radians
dur=.025; %sec
subplot(2,1,1)
[xx,tt] =one_cos(A,w,p,dur);
plot(tt,xx);
title("one\_cos Test")
xlabel('TIME (sec)')

%#### Testing for syn_sin function ####
subplot(2,1,2)
[xx0,tt0] = syn_sin([0,100,250],[10,14*exp(-j*pi/3),8*j],10000,1,0);
plot(tt0,xx0)
title("syn\_sin Test")
xlabel('TIME (sec)')
```

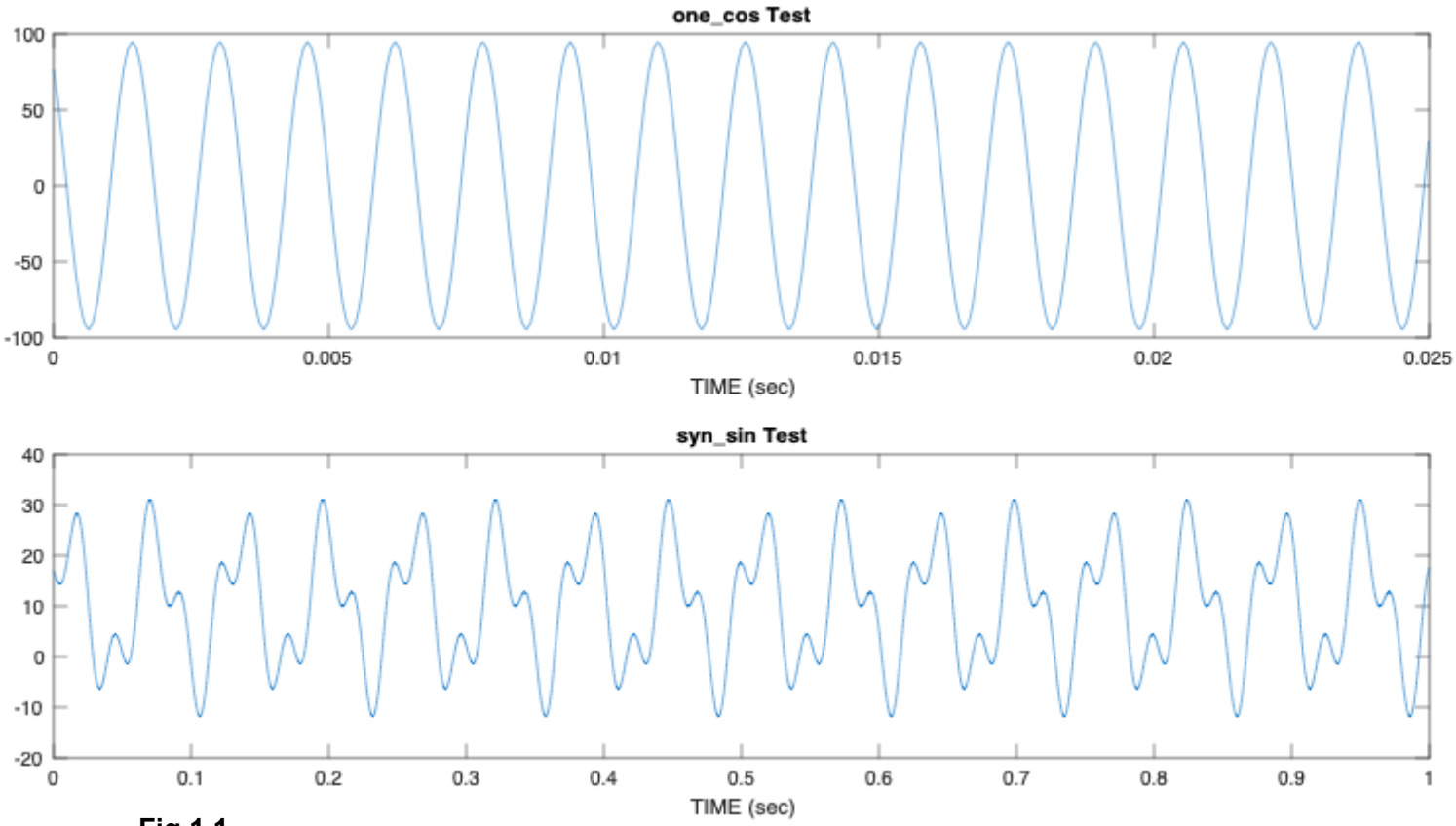

Fig 1.1

Lab Section

5 (b)

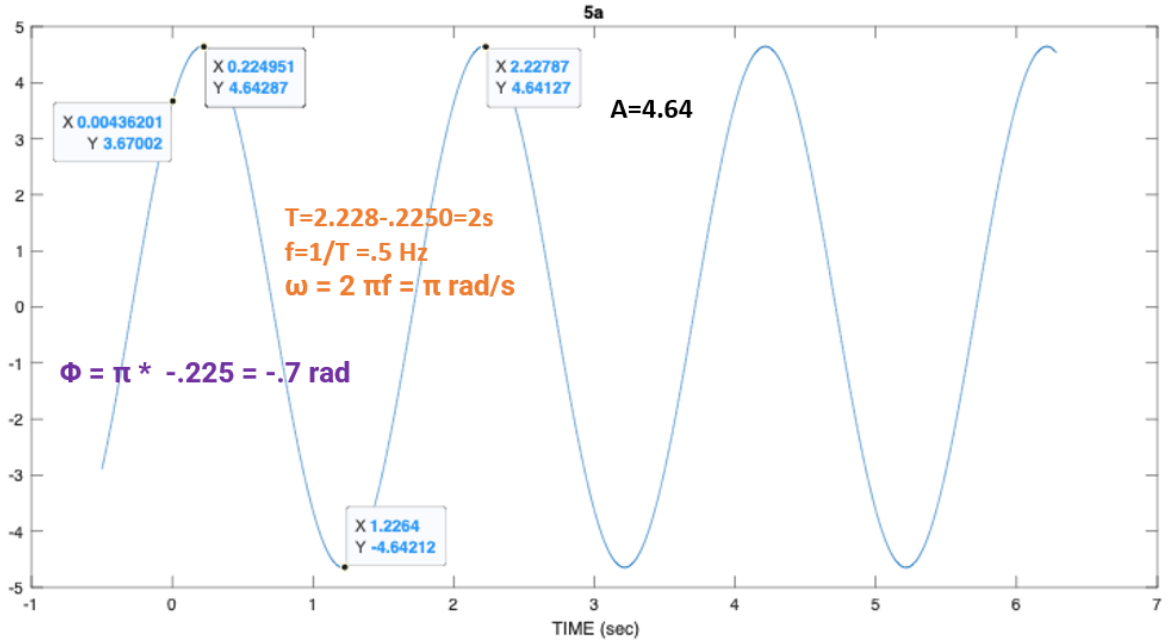

Fig 1.2

5 (a)

```
[xx0,tt0] = syn_sin([pi,pi,pi],[2*exp(j),2*exp(j*-1.25),(1-j)],10000,2*pi,-.5);
plot(tt0,xx0)
title("5a")
xlabel('TIME (sec)')
```

5 (c)

Hand Calculated Amplitude and Phase for $x(t) = Re\{2e^{j\pi t} + 2e^{j\pi(t-1.5)} + (1-j)e^{j\pi t}\}$

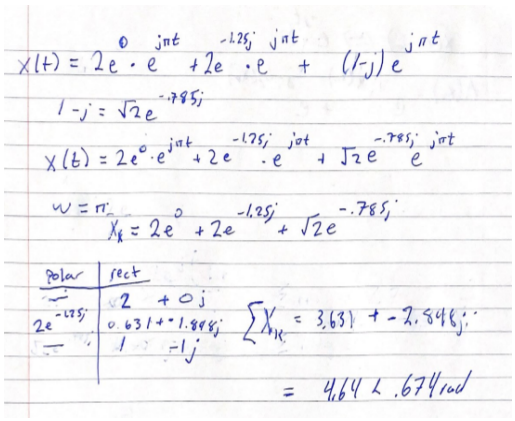

Fig 1.3

6 - Hand Calculated $t_1(x_v)$, $t_2(x_v)$, $A(x_v)$

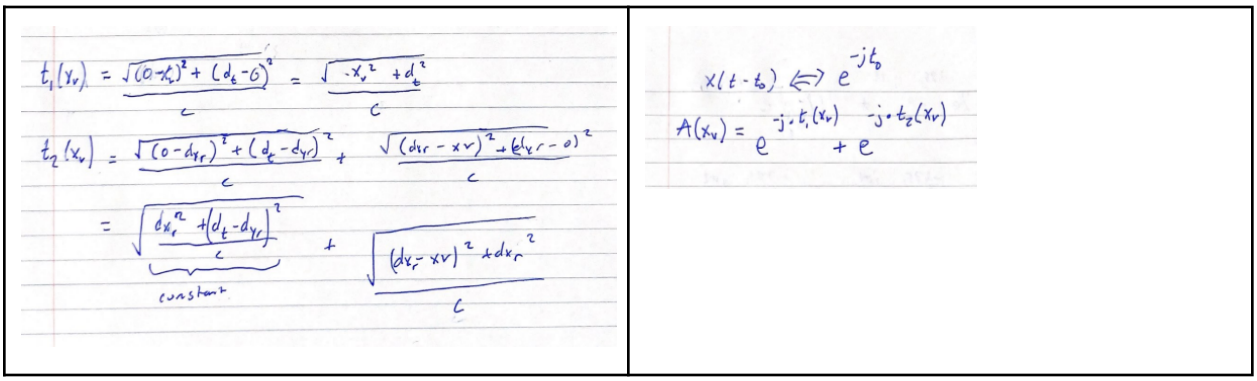

Fig 1.4

6 - Functions For Part (d)

```
%function that gets distance between two points
%(x1,y1) (x2,y2)
function k3 = distance(x1,y1,x2,y2);
           k3=sqrt((x2-x1)^2+(y2-y1)^2);

       end
```

```
%returns time delay given x pos of vehicle and y pos of transmitter
function k = t1(xv,dt)

      k=distance(xv,0,0,dt)/(2.998*10^8);

end
```

```
%returns time delay given x,y coords of transmitter and reflector and x
%coord of vehicle
function k1 = t2(dxr,dyr,xv,dt);
       k1=distance(xv,0,dxr,dyr)+distance(dxr,dyr,0,dt)/(2.998*10^8);

end
```

6 - Part (d)

```
f = 150*10^6; % frequency
T = 1/f; % Period
fs=90*f; % Sampling Frequency
t = 0 : 1/fs : 3*T;

yTrs=1500;% y pos of transmitter
xRf=100; % x pos of reflector
yRf=900; % y pos of reflector
xv=100; % x pos of vehicle

%getting time delays from Helper function
delay1=t1(xv,yTrs);
delay2=t2(xRf,yRf,xv,yTrs);

% ## TIME DOMAIN CALCULATIONS ##
%Time domain signals denoted with T_

%defining signals in TIME DOMAIN
T_reflec_sig1 = cos(2*pi*f*(t-delay1));
T_reflec_sig2 = cos(2*pi*f*(t-delay2));

%Received signal is difference of two reflected signals
T_received_sig = T_reflec_sig1-T_reflec_sig2
T_A = max(abs(T_received_sig));

%Plotting
hold on
subplot(3,1,1)
plot(t*.1e9,T_received_sig,"Red")
title('Time Domain Calculated Received Signal With Xv = 0 m')
xlabel('TIME (fsec)')
caption = sprintf('Max Amplitude = %.4f', T_A);
text(.2,.2,caption,"FontSize",10);
hold off

% ## Phasor DOMAIN CALCULATIONS ##

%Phasor domain signals denoted with P_
P_reflec_sig1 = exp(2*pi*j*f*-delay1);
P_reflec_sig2 = exp(2*pi*j*f*-delay2);

%Received signal is difference of two reflected signals
P_received_sig=P_reflec_sig1 - P_reflec_sig2;
P_A=abs(P_received_sig);

%Converting received signal back into the time domain for graphing
```

```
converted_sig=real( (P_received_sig) * exp(j*2*pi*f*t) );

%plotting
hold on
subplot(3,1,2)
plot(t*.1e9,converted_sig,"Blue")
title('Phasor Domain Calculated Received Signal With Xv = 0 m')
xlabel('TIME (fsec)')
caption = sprintf('Max Amplitude = %.4f', P_A);
text(.2,.2,caption,"FontSize",10);
hold off
```

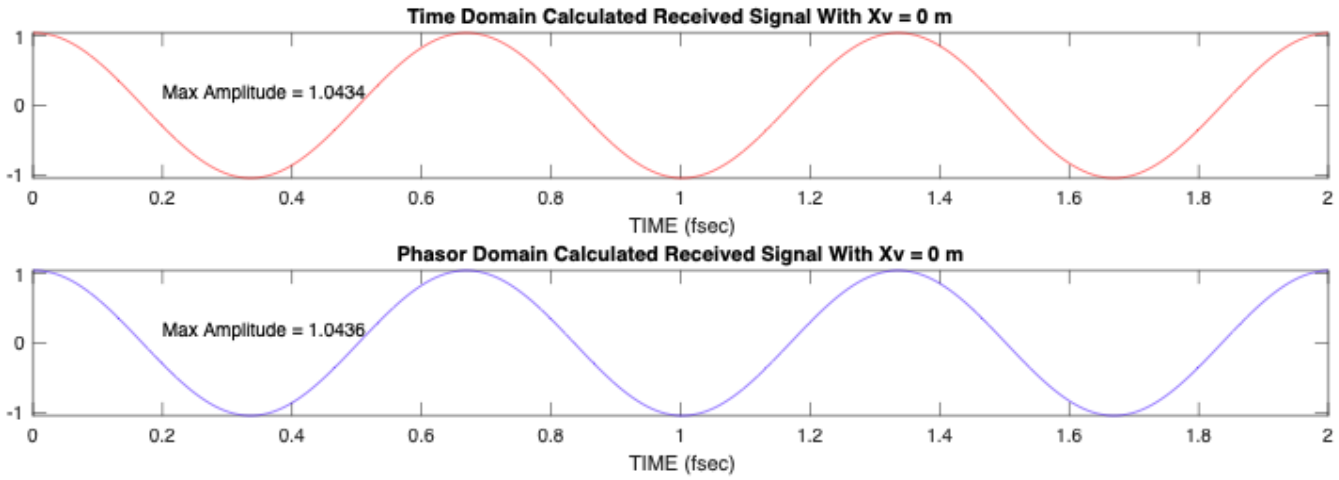

Fig 1.5

6 - Functions For Part (e)

```
%function that gets distance between two points
%(x1,y1) (x2,y2)
function k3 = distancev(x1,y1,x2,y2);
           k3=sqrt((x2-x1).^2+(y2-y1).^2);

       end
```

```
%specially designed for vector calcs
```

```
%returns time delay given x pos of vehicle and y pos of transmitter
function k = t1v(xv,dt)

      k=distancev(xv,0,0,dt)./(2.998*10^8);

end
```

```
%specially designed for vector calcs
%returns time delay given x,y coords of transmitter and reflector and x
%coord of vehicle
function k1 = t2v(dxr,dyr,xv,dt);
       k1=(distancev(xv,0,dxr,dyr)+distance(dxr,dyr,0,dt))./(2.998*10^8);

end
```

```
%returns  phasor from time delay and f
function  p = get_phasor(delay,f)
       p=exp(2*pi*j*f*-1*delay);
end
```

6 - Part (e)

```
f = 150*10^6; % frequency
yTrs=1500;% y pos of transmitter
xRf=100; % x pos of reflector
yRf=900; % y pos reflector
xv_all_pos= 0:300; % all positions of vehicle used
%this program uses a special version of the program made in part d
%functions with a 'v' at the end have been redesigned for vector
%calculations
%time delay vectors
delay1_all_pos=t1v(xv_all_pos,yTrs);
delay2_all_pos=t2v(xRf,yRf,xv_all_pos,yTrs);
%complex A of both signals
og_sig_all_pos=get_phasor(delay1_all_pos,f);
reflctd_sig_all_pos=get_phasor(delay2_all_pos,f);
%calc received signal
recvd_sig_all_pos= og_sig_all_pos - reflctd_sig_all_pos;
%calc A for every received signal
```

```
A_all_pos=abs(recvd_sig_all_pos);
plot(xv_all_pos,A_all_pos)
title('Signal Strength vs Distance')
ylabel('Signal Strength')
xlabel('Distance (m)')
```

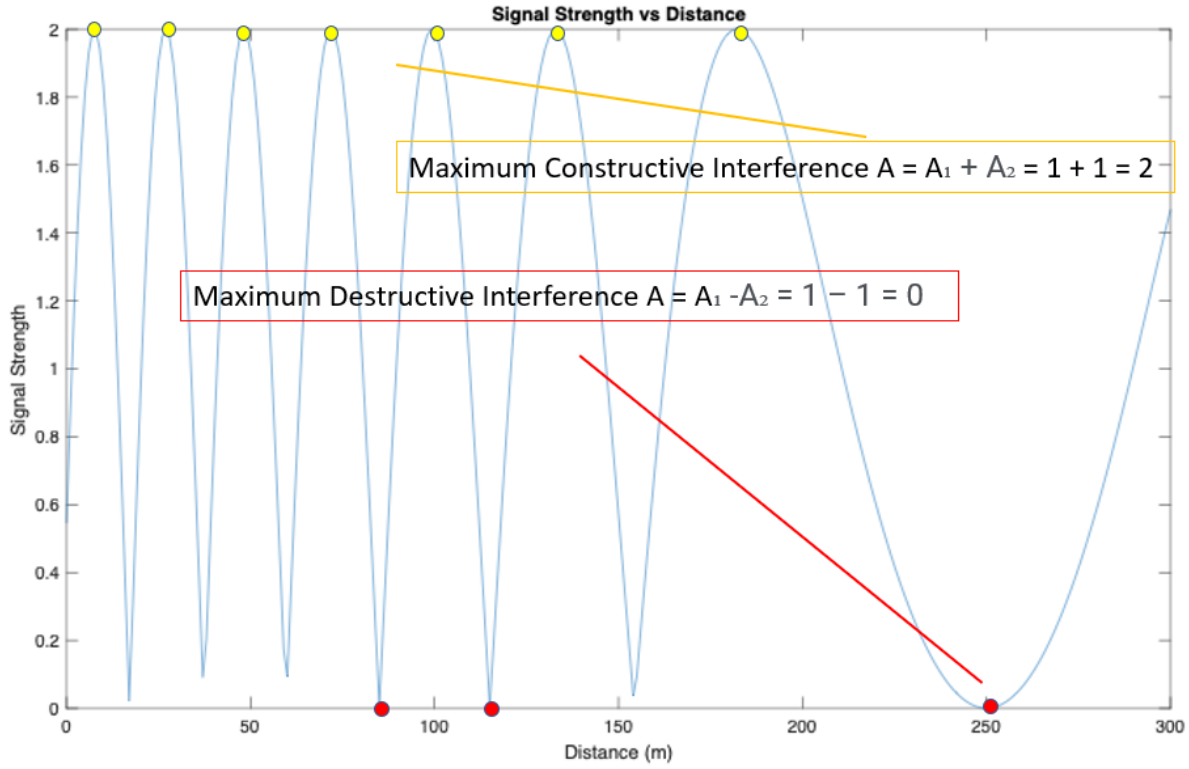

Fig 1.6

As is described above, the received signal has a maximum amplitude of 2 when signals are totally constructive and 0 when the two signals cancel.