CECS 463 System On Chip II
FALL 2020

Kuldeep Gohil

Assignment #06 – Ch4&5 Problems

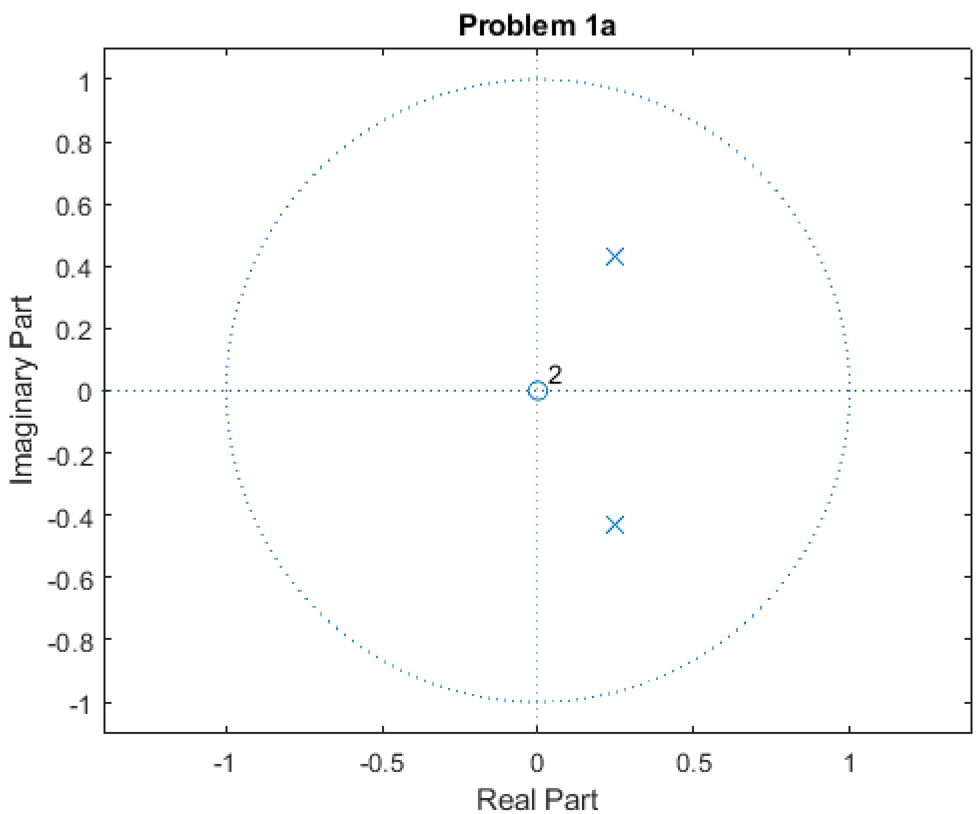
10/29/2020

```
% Kuldeep Gohil
% CECS 463 Fall20
% Assignment #06 Due: 10/29/2020

clc; close all;
```

Problem 6.1a

```
disp('Problem 1(a)')
b=[0.5,0];
a=[1,-0.5,0.25];
figure();
zplane(b,a);
title('Problem 1a');
```

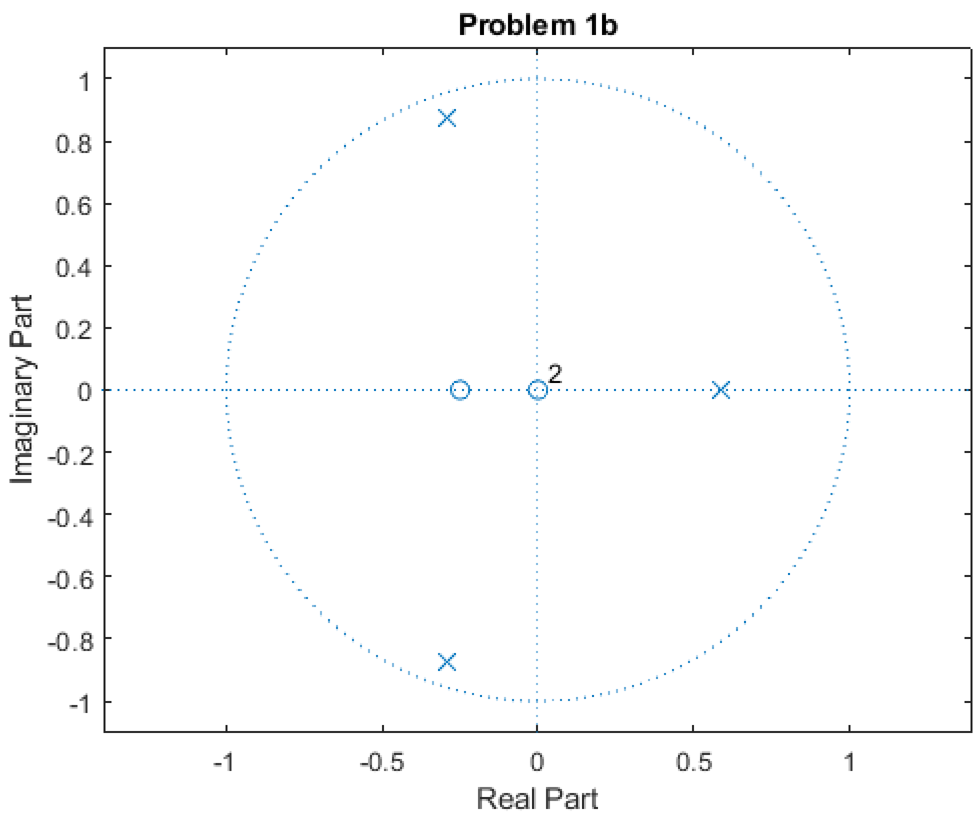
```
Problem 1(a)
```



Problem 6.1b

```
disp('Problem 1(b)')
b=[1,0.25,0];
a=[1,0,0.5,-0.5];
figure();
zplane(b,a);
title('Problem 1b');
```

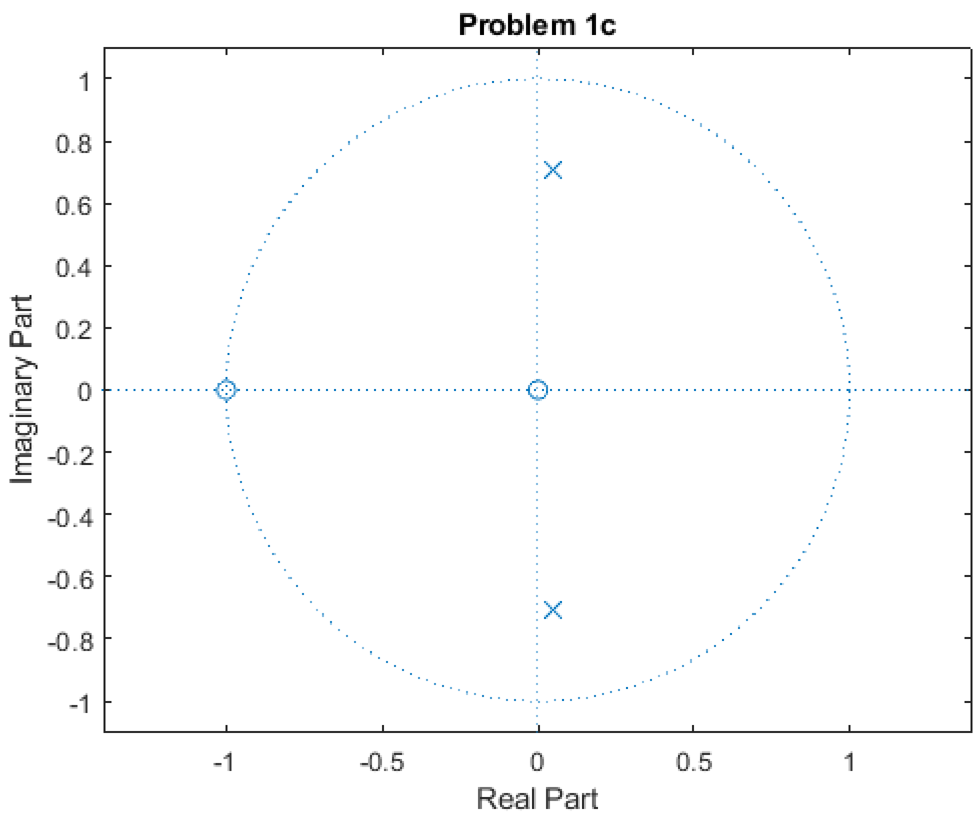
```
Problem 1(b)
```



Problem 6.1c

```
disp('Problem 1(c)')
b=[1,1,0,0];
a=[1,-0.1,0.5];
figure();
zplane(b,a);
title('Problem 1c');
```

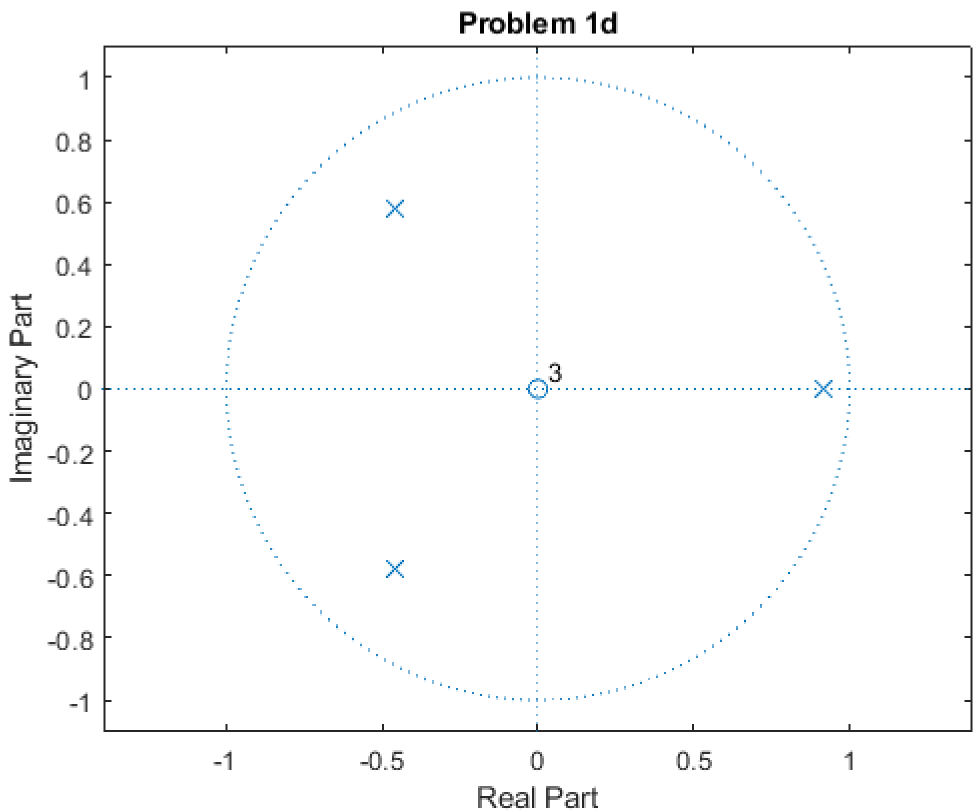
```
Problem 1(c)
```



Problem 6.1d

```
disp('Problem 1(d)')
b=[1];
a=[1,0,-0.3,-0.5];
figure();
zplane(b,a);
title('Problem 1d');
hold off;
```

```
Problem 1(d)
```

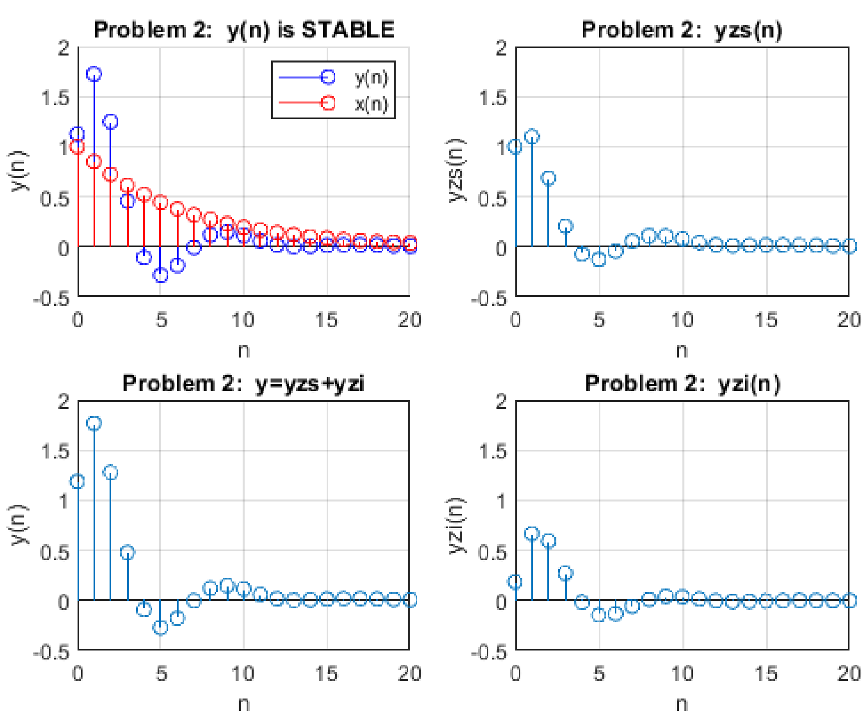


Problem 6.2

```
disp('Problem 2')
disp('  2(a) See figure for Problem 2');
disp('       There is only a transient solution.');
disp('  2(b) See figure for Problem 2');
disp('  2(c) See figure for Problem 2');
disp('  2(d) See figure for Problem 2');
disp('       The system is stable; y(n) --> 0 as n gets large.');
b=[1,-0.75];
a=[1,-1,0.5];
Y=[-1,-3];
X=[1/2];
q=0.85;
n=0:20;
x=(q).^n;
xic=filtic(b,a,Y,X);
y=filter(b,a,x,xic);

num=conv(xic,[1,-q]);
num=num+[b,zeros(1,length(num)-length(b))];
den=conv(a,[1,-q]);
[R,p,C]=residuez(num,den);
Mp=abs(p);
Ap=angle(p);
y1=R(1)*(Mp(1)).^n + Mp(2).^n.*(2*real(R(2))*cos(n*Ap(2))-2*imag(R(2))*sin(n*Ap(2)));

[Rzs,pzs,Czs]=residuez(b,den);
Mpzs=abs(pzs);
Apzs=angle(pzs);
yzs=Rzs(1)*(Mpzs(1)).^n + Mpzs(2).^n.*(2*real(Rzs(2))*cos(n*Apzs(2))-
2*imag(Rzs(2))*sin(n*Apzs(2)));

[Rzi,pzi,Czi]=residuez(xic,a);
Mpzi=abs(pzi);
Apzi=angle(pzi);
yzi=Rzi(1)*(Mpzi(1)).^n + Mpzi(2).^n.*(2*real(Rzi(2))*cos(n*Apzi(2))-
2*imag(Rzi(2))*sin(n*Apzi(2)));

figure();
grid on;
subplot(2,2,1);
hold on;
stem(n,y1,'b');
stem(n,x,'r');
hold off;
grid on;
axis([0,n(end),-0.5,2]);
title('Problem 2:  y(n) is STABLE');
xlabel('n');
ylabel('y(n)');
legend('y(n)','x(n)');
subplot(2,2,2);
stem(n,yzs);
axis([0,n(end),-0.5,2]);
grid on;
title('Problem 2:  yzs(n)');
xlabel('n');
ylabel('yzs(n)');
subplot(2,2,4);
stem(n,yzi);
axis([0,n(end),-0.5,2]);
```

```
grid on;
title('Problem 2:  yzi(n)');
xlabel('n');
ylabel('yzi(n)');
y2=yzs+yzi;
subplot(2,2,3);
stem(n,y2);
grid on;
axis([0,n(end),-0.5,2]);
title('Problem 2:  y=yzs+yzi');
xlabel('n');
ylabel('y(n)');
```

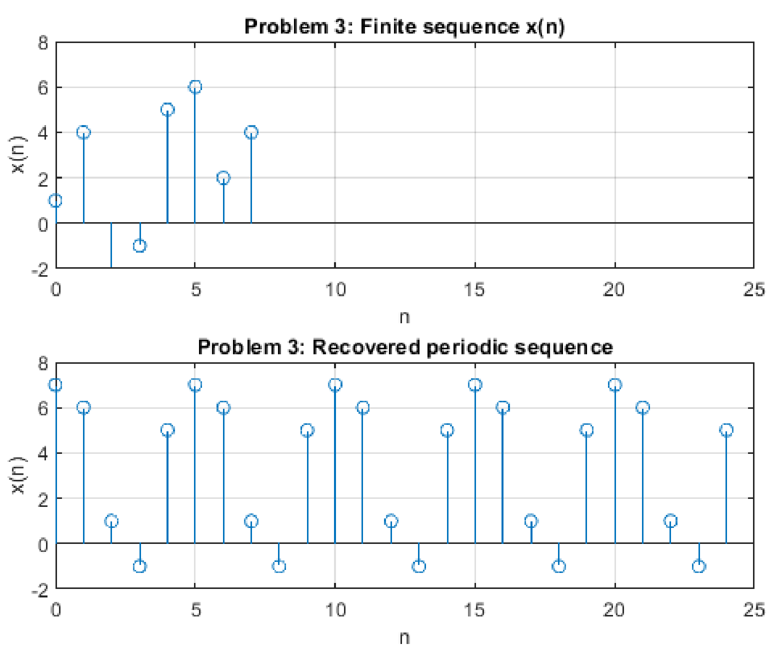
```
Problem 2

  2(a) See figure for Problem 2

       There is only a transient solution.

  2(b) See figure for Problem 2

  2(c) See figure for Problem 2

  2(d) See figure for Problem 2

       The system is stable; y(n) --> 0 as n gets large.

Warning: Using only the real component of complex data.

Warning: Using only the real component of complex data.
```



Problem 6.3

```
disp('Problem 3')
x=[1,4,-3,-1,5,6,2,4];
n=0:7;
figure();
subplot(2,1,1);
stem(n,x);
grid on;
axis([0,25,-2,8]);
title('Problem 3: Finite sequence x(n)');
xlabel('n');
ylabel('x(n)');
N=5;
k=0:4;
wk=2*pi*k/N;
M=100*N;
w=[0:2*M]*pi/M;
X=dtft(x,n,w);
for m=1:N
    Xk(m)=X(w==wk(m));
end
xn=round(real(idfs(Xk,N)));
xp=xn'*ones(1,5);
xp=xp(:)';
subplot(2,1,2);
stem(0:5*N-1,xp);
grid on;
axis([0,25,-2,8]);
title('Problem 3: Recovered periodic sequence');
xlabel('n');
ylabel('x(n)');
```

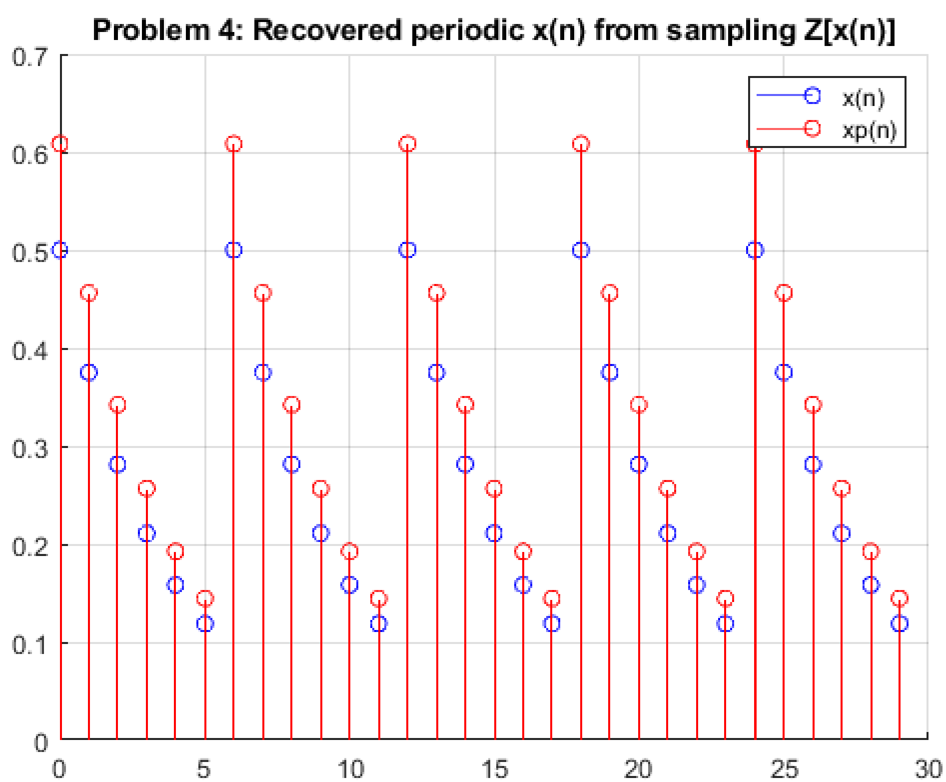
```
Problem 3
```



Problem 6.4

```
disp('Problem 4')
disp('    Figure 4 shows aliasing in time domain recovery of x(n)');
N=6;
k=0:N-1;
z=exp(1j*2*pi*k/N);
Xk=0.5./(1-(0.75)*z.^-1);
x=real(idfs(Xk,N));
P=5; xpr=x'*ones(1,P);
xpr=(xpr(:))';
npr=0:P*N-1;x=0.5*(0.75).^k;
xp=x'*ones(1,P);
xp=(xp(:))';
np=0:P*N-1;
figure();
hold on;
grid on;
stem(np,xp,'b');
stem(npr,xpr,'r');
legend('x(n)','xp(n)');
title('Problem 4: Recovered periodic x(n) from sampling Z[x(n)]');
```

```
Problem 4

    Figure 4 shows aliasing in time domain recovery of x(n)
```



Problem 6.5

```
disp('Problem 5')
disp('    Figure 5 shows very little aliasing in the recovered x(n)');
N=25;
k=0:N-1;
z=exp(1j*2*pi*k/N);
Xk=0.5./(1-(0.75)*z.^-1);
x=real(idfs(Xk,N));
P=3; xpr=x'*ones(1,P);
xpr=(xpr(:))';
npr=0:P*N-1;
x=0.5*(0.75).^k;
xp=x'*ones(1,P);
xp=(xp(:))';
np=0:P*N-1;
figure();
hold on;
grid on;
stem(np,xp,'b');
stem(npr,xpr,'r');
legend('x(n)','xp(n)');
title('Problem 5: Recovered periodic x(n) from sampling Z[x(n)]');
```

```
Problem 5

    Figure 5 shows very little aliasing in the recovered x(n)
```