Norwegian University of Science and Technology
Department of Electronics and Telecommunications

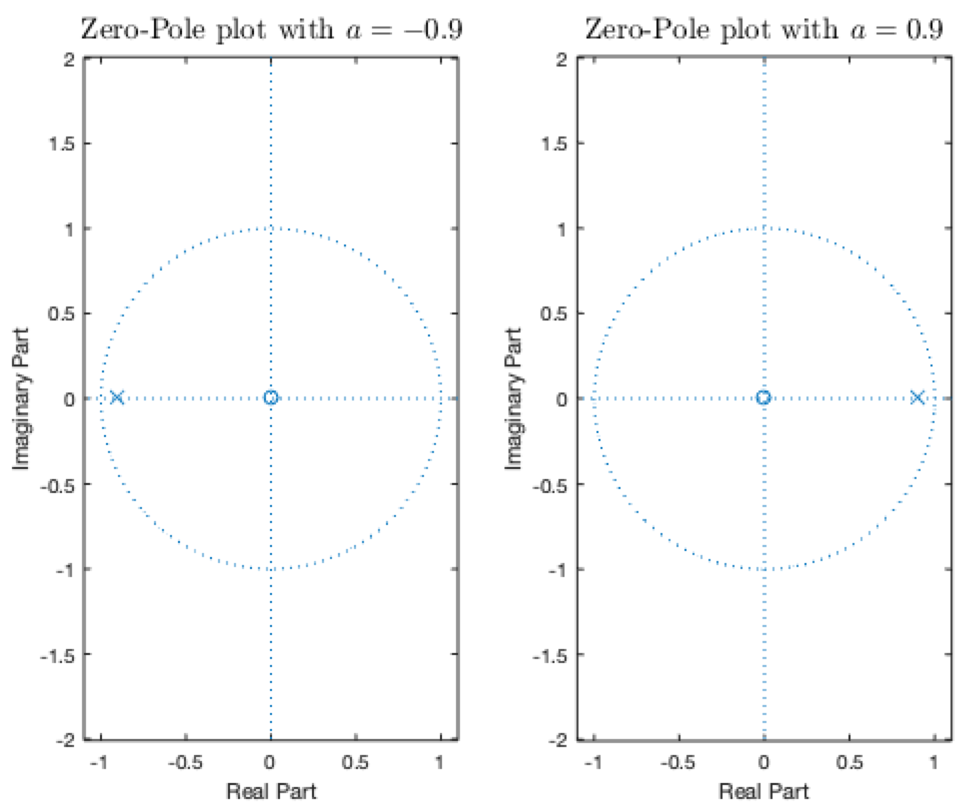
TTT4120 Digital Signal Processing

Solutions for Problem Set 4

Problem 1:

(a).

The pole-zero plots are shown in the figure below.



The frequency response is the z-transform evaluated along the unit circle. The magnitude of the response will be high at frequencies close to the pole, and lower at frequencies further away from the pole. We can then see from the plot that when $a = 0.9$ the filter is a lowpass filter, and when $a = -0.9$ the filter is a highpass filter.

(b).

Note that `pezdemo` or Pole-Zero Editor is a program that helps visualize the effect of pole-zero placements on a system's frequency and impulse response and it is only valid for causal filters.

Problem 2:

(a).

The inverse filter is given by

$$H_I(z) = H^{-1}(z) = \left(1 - \frac{1}{2}z^{-1}\right)\left(1 + \frac{1}{2}z^{-1}\right) = 1 - \frac{1}{4}z^{-2}.$$

(b).

Since this filter is an FIR filter, it is stable.

(c).

Since filter $H_I(z)$ has zeros at $z_1 = 1/2$ and $z_2 = -1/2$, which is inside the unit circle, the filter is a minimum phase filter.

(d).

For a filter with a zero at z to be linear phase, the filter must also have a zero at $1/z$. The filter $H_I(z)$ has zeros at $z_1 = 1/2$ and $z_2 = -1/2$, but not at 2 or -2. Thus, the filter is not linear phase.

Problem 3:

(a).

$A(z)$ is $allpass$ because it has pole and zero respectivly in α and α^{-1}.

(b).

By examining the block diagram we see that, for the upper branch,

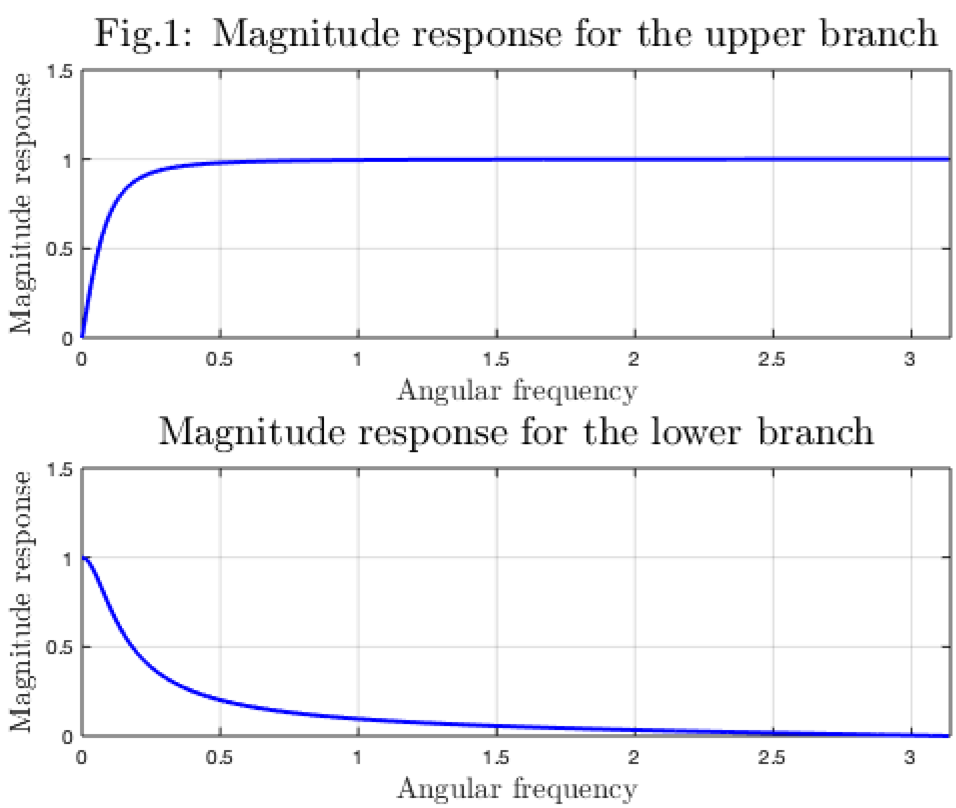
$$\begin{aligned}
Y_{ub}(z) &= \frac{1}{2}[1 + A(z)]X(z) \\
H_{ub}(z) &= \frac{Y_{ub}(z)}{X(z)} = \frac{1}{2}[1 + A(z)] = \frac{1}{2}[1 + \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}}] \\
&= \frac{1}{2}[\frac{1 - \alpha z^{-1} + \alpha - z^{-1}}{1 - \alpha z^{-1}}] = \frac{(1+\alpha) - (1+\alpha)z^{-1}}{2 - 2\alpha z^{-1}} = \frac{1.9 - 1.9z^{-1}}{2 - 1.8z^{-1}}.
\end{aligned}$$

For the lower branch

$$\begin{aligned}
Y_{lb}(z) &= \frac{K}{2}[1 - A(z)]X(z) \\
H_{lb}(z) &= \frac{Y_{lb}(z)}{X(z)} = \frac{K}{2}[1 - A(z)] = \frac{K}{2}[1 - \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}}] \\
&= \frac{K}{2}[\frac{1 - \alpha z^{-1} - \alpha + z^{-1}}{1 - \alpha z^{-1}}] = \frac{K(1-\alpha) + K(1-\alpha)z^{-1}}{2 - 2\alpha z^{-1}} = \frac{0.1 + 0.1z^{-1}}{2 - 1.8z^{-1}}.
\end{aligned}$$

Thus, the magnitude response can be calculated and plotted with the following code.

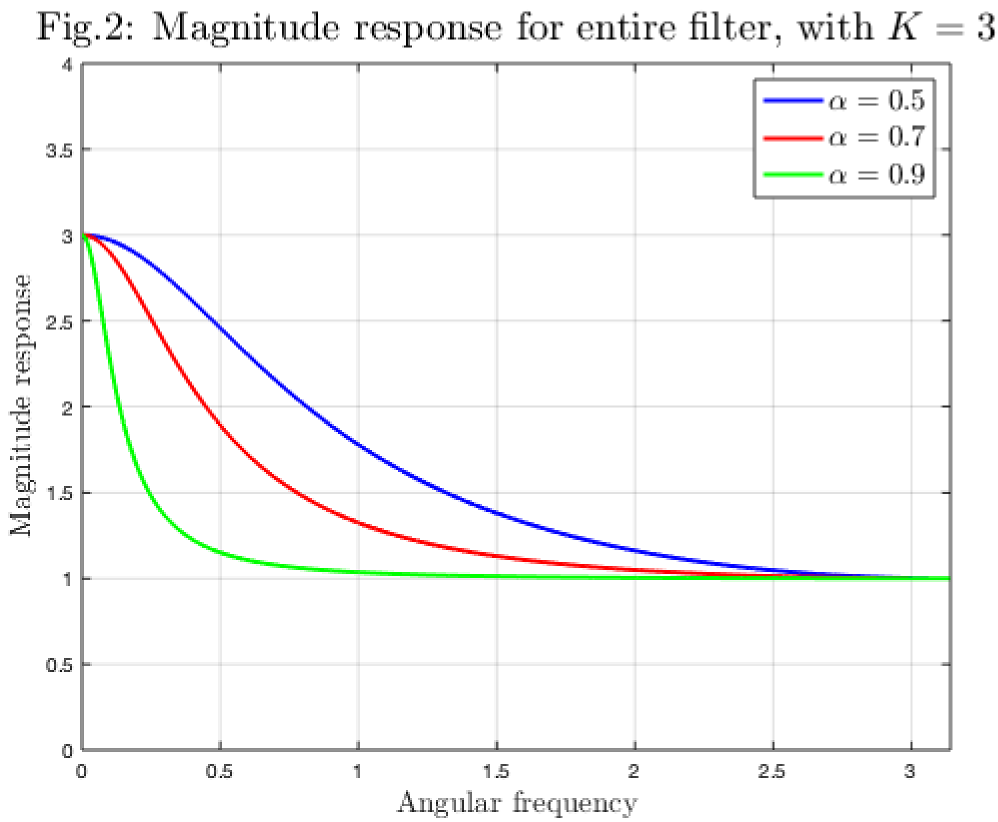
```
b_ub = [1.9,-1.9];
a = [2,-1.8];
b_lb = [0.1,0.1];
[h_ub,w] =freqz(b_ub,a);
[h_lb,w] = freqz(b_lb,a);
figure(1)
subplot(2,1,1)
plot(w,abs(h_ub),'LineWidth',2); grid on;
axis([0,pi,0,1.5])
xlabel('Angular frequency')
ylabel('Magnitude response')
title('Magnitude response for the upper branch')
subplot(2,1,2)
plot(w,abs(h_lb));grid on;
axis([0,pi,0,1.5])
xlabel('Angular frequency')
ylabel('Magnitude response')
title(' Magnitude response for the lower branch')
```



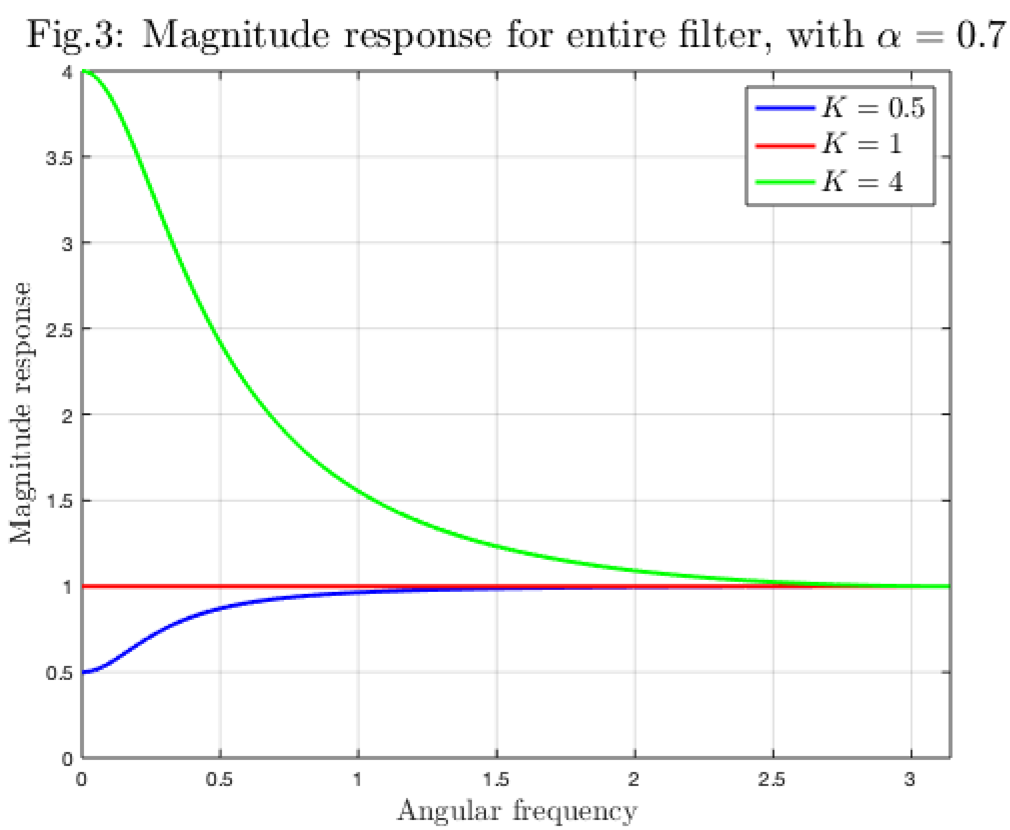
We can see that the upper branch represents a highpass filter, while the lower branch represents a lowpass filter.

(c).

Figure.2 is a plot of the magnitude response with K fixed at 3 and different αs,



while Figure.3 is a plot of the magnitude response with α fixed at 0.7 and different Ks.



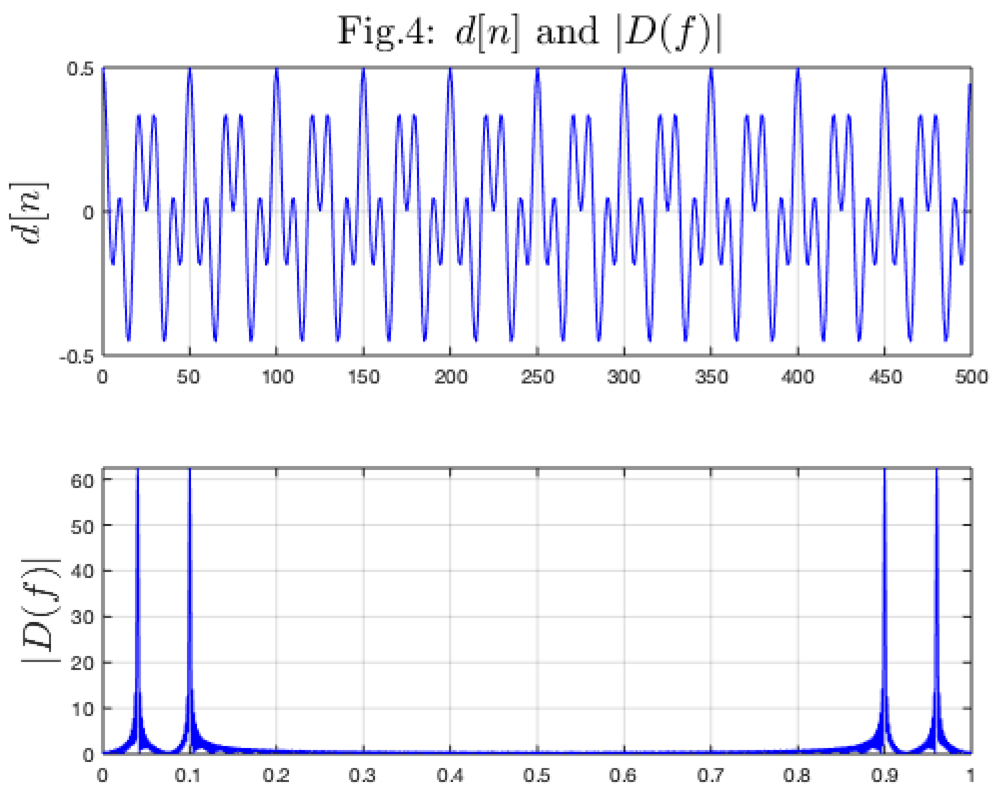
K is the gain of the lower branch (the lowpass filter). By adjusting K we regulate the contribution from the lowpass filter relative to the contribution from the highpass filter. Thus, K controls the boost or cut at low

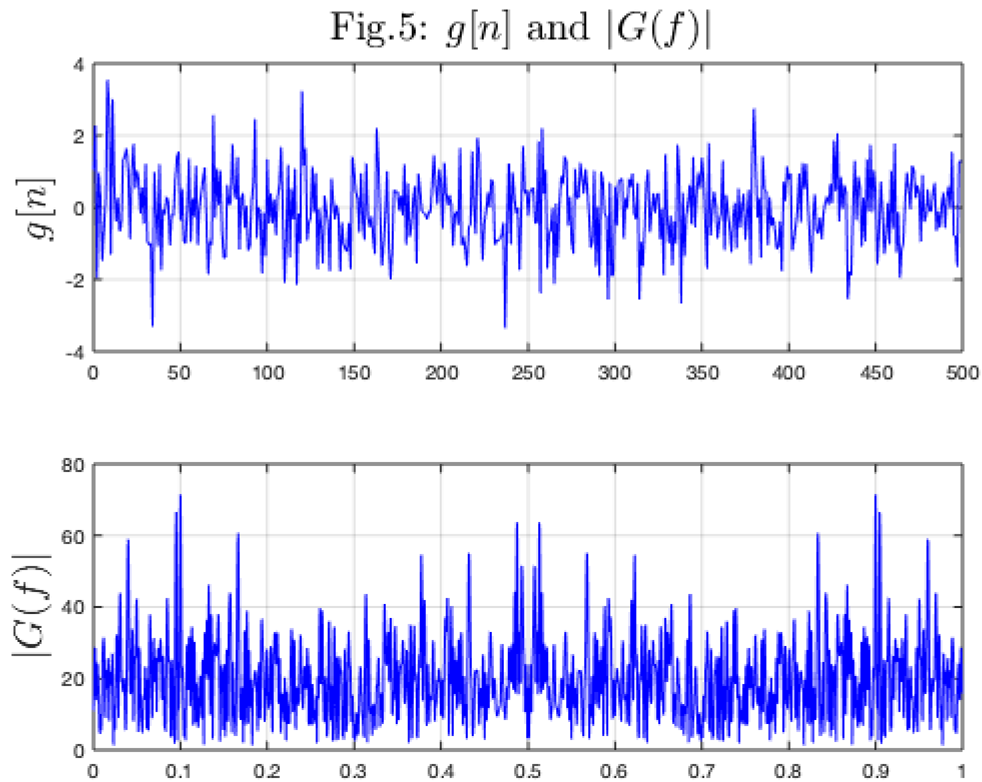
frequencies. As we can see from $Fig.2$, the parameter α controls the boost or cut bandwidth.

Problem 4:

(a).

Plot of the amplitude spectra for $G(f)$ and $D(f)$ are given below and shown in $Fig.4$ and $Fig.5$,





(b).

Transfer functions of the digital resonators are given by

$$
\begin{aligned}
H_x(z) &= \frac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j2\pi f_x}z^{-1})(1-0.99e^{-j2\pi f_x}z^{-1})} \\
H_y(z) &= \frac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j2\pi f_y}z^{-1})(1-0.99e^{-j2\pi f_y}z^{-1})}.
\end{aligned}
$$

The following MATLAB code generates plots of zeros and poles, and amplitude responses of the resonators. These plots are shown in $Fig.6-9$. From $|H_x(f)|$ and $|H_y(f)|$ in $Fig.8$ and $Fig.9$, we can see peaks at f_x and f_y respectively. This means that the resonators will remove all frequency components from a signal except components close to f_x and f_y.

```
px=[0.99*exp(j*2*pi*fx) 0.99*exp(-j*2*pi*fx)]';
py=[0.99*exp(j*2*pi*fy) 0.99*exp(-j*2*pi*fy)]';
z=[-1 1]';
zplane(z,px)
zplane(z,py)
[Hx,w]=freqz(poly(z),poly(px));
[Hy,w]=freqz(poly(z),poly(py));
f=w/2/pi;
plot(f,abs(Hx));
plot(f,abs(Hx));
```

Fig.6: Zeros and poles for $H_x(z)$

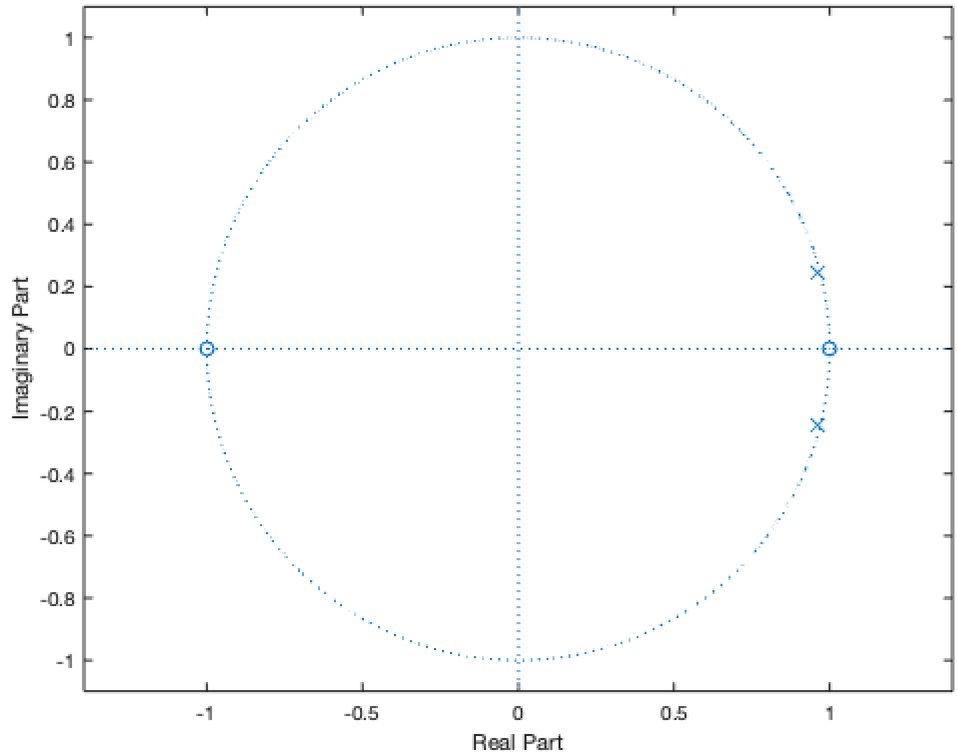
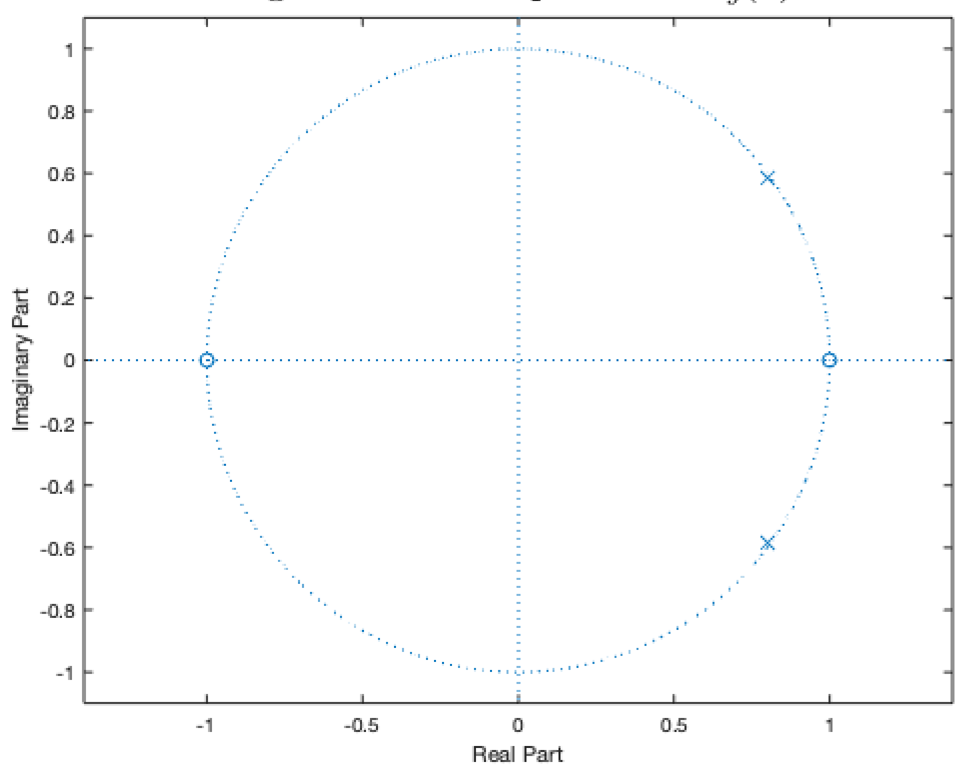
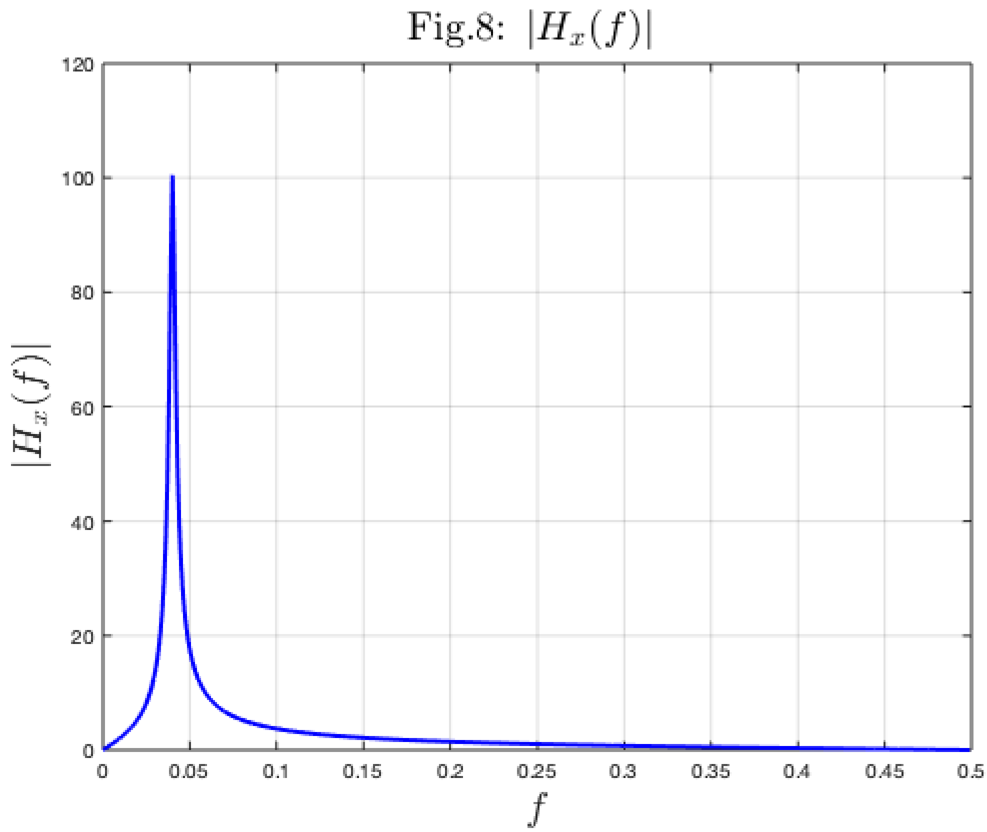
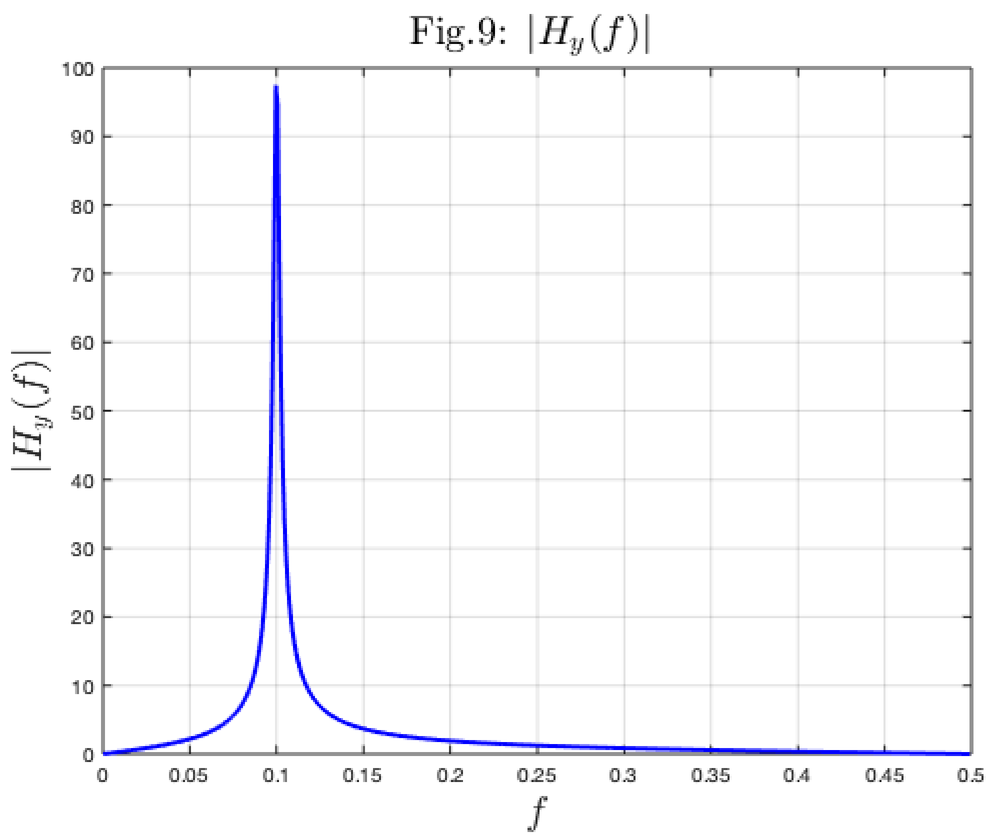


Fig.7: Zeros and poles for $H_y(z)$







(c).

The MATLAB code for filtering the noise contaminated signal with $H_x(z)$ and $H_y(z)$ are given below. Plots of the output from the filter as well as their amplitude spectra can be found in $Fig.10$ and $Fig.11$,

```
qx=filter(poly(z),poly(px),gn);
Qxf=fft(qx,N);
qy=filter(poly(z),poly(py),gn);
Qyf=fft(qy,N);
```

Fig.10: Output signal $q_x[n]$ and $|Q_x(f)|$

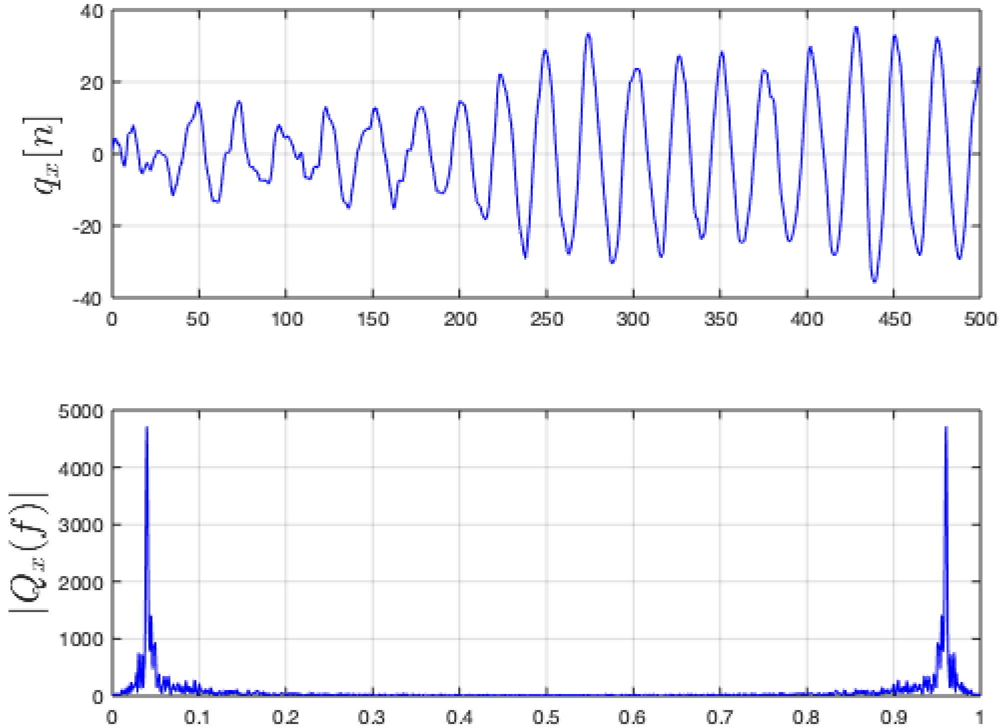
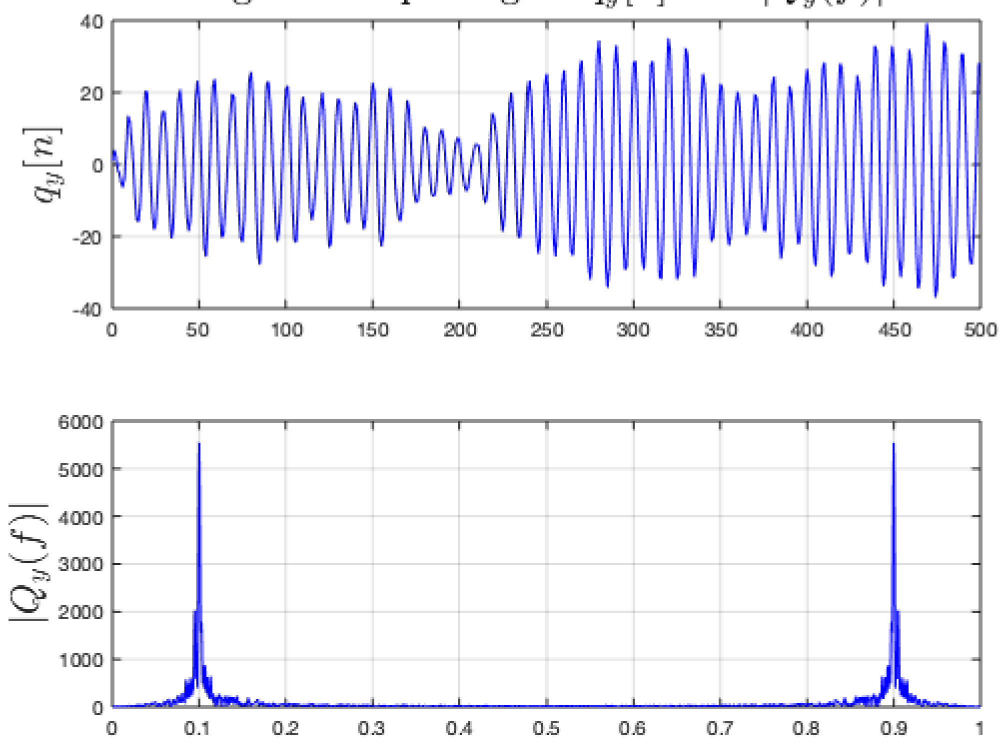


Fig.11: Output signal $q_y[n]$ and $|Q_y(f)|$



In $Fig.10$ and $Fig.11$, we see the output signals after filtering with $H_x(z)$ and $H_y(z)$ respectively. From $q_x[n]$ we can see that the frequency component f_x has been preserved, while f_y and most of the noise have been removed. In the signal $q_y[n]$, only f_y has been preserved. The same observations can be made from the corresponding spectra.

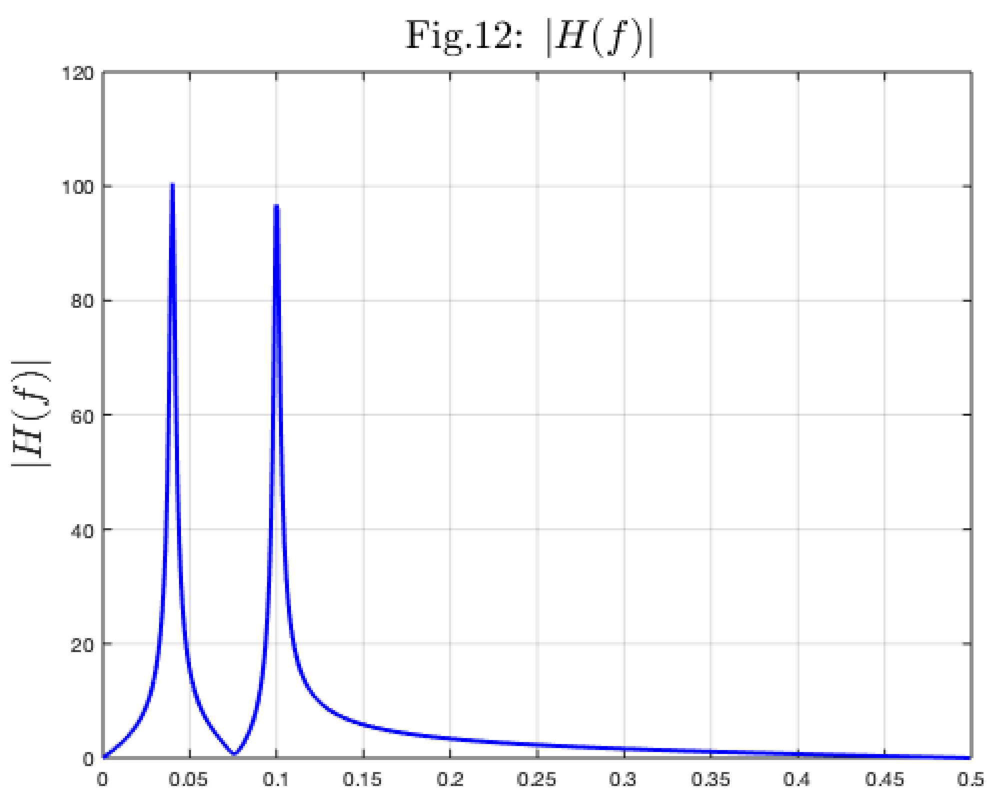
(d).

Since we want to isolate the two sinusoids with the resulting filter we need to combine then in parallel. The expression for $H(z)$ can then be written as

$$H(z) = H_x(z) + H_y(z)$$

The following MATLAB code will plot the magnitude response of the resulting system.

```
H=Hx+Hy;
plot(f,abs(H));
```

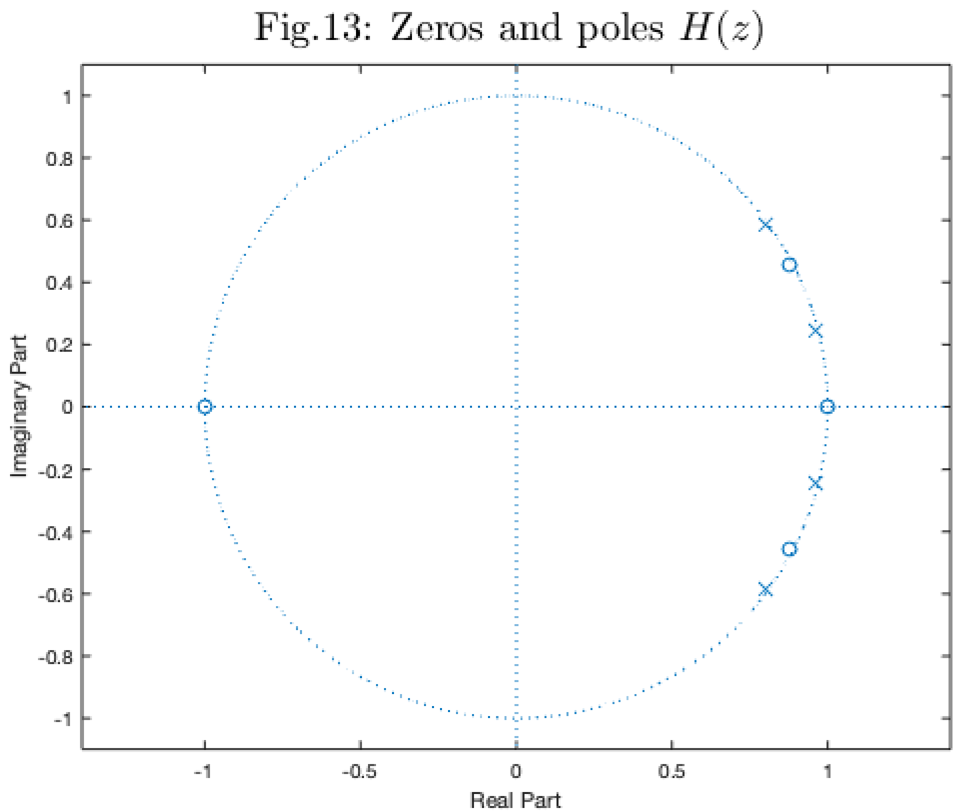


From $|H(f)|$ in $Fig.12$ we can see that there are peaks at both f_x and f_y. In addition, the response is zero between the peaks. To find the poles and zeros we can write the expression for $H(z)$ as

$$\begin{aligned} H(z) &= H_x(z) + H_y(z) = \frac{(1-z^{-1})(1+z^{-1})}{(1-p_x z^{-1})(1-p_x^* z^{-1})} + \frac{(1-z^{-1})(1+z^{-1})}{(1-p_y z^{-1})(1-p_y^* z^{-1})} \\ &= \frac{(1-z^{-1})(1+z^{-1})\big((1-p_x z^{-1})(1-p_x^* z^{-1}) + (1-p_y z^{-1})(1-p_y^* z^{-1})\big)}{(1-p_x z^{-1})(1-p_x^* z^{-1})(1-p_y z^{-1})(1-p_y^* z^{-1})} \end{aligned}$$

We can see from this equation that the filter will have all the poles and zeros from H_x and H_y. In addition to two new zeros from the second degree polynomial, which can be found by using the MATLAB functions `poly` and `roots`. The MATLAB code for this is given below.

```
z(3:4)=roots(poly(px)+poly(py));
p=[px;py];
zplane(z,p)
```



We can se from the plot in $Fig.13$ that all of the poles and zeros from both $H_x(z)$ and $H_y(z)$, in addition to two new zeros between the poles.

MATLAB code for filtering the noise contaminated signal with $H(z)$ is given below, and resulting plot is shown in $Fig.14$. We can see the result after filtering with $H(z)$. Although $q[n]$ is a somewhat distorted version of $d[n]$, we can observe similarity. The spectrum $|Q(f)|$ shows that most of the noise has been removed.

```
q=filter(poly(z),poly(p),gn);
Qf=fft(q,N);
plot(n,q)
f=0:1/N:1-1/N;
plot(f,abs(Qf))
```

Fig.14: Output signal $q[n]$ and $|Q(f)|$