



Norwegian University of Science and Technology  
Department of Electronics and Telecommunications

## TTT4120 Digital Signal Processing

### Solutions for Problem Set 5

#### Problem 1:

(a).

We have that  $S_{xx}(\omega) = |X(\omega)|^2$ . The DTFT of  $x[n]$  is found as follows.

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \sum_{n=0}^{\infty} (ae^{-j\omega})^n = \frac{1}{1 - ae^{-j\omega}}$$

Thus, we get

$$S_{xx}(\omega) = |X(\omega)|^2 = X(\omega)X^*(\omega) = \frac{1}{1 - ae^{-j\omega}} \cdot \frac{1}{1 - ae^{j\omega}} = \frac{1}{1 - ae^{-j\omega} - ae^{j\omega} + a^2} = \frac{1}{1 + a^2 - 2a \cos \omega}$$

$$S_{xx}(f) = S_{xx}(\omega)|_{\omega=2\pi f} = \frac{1}{1 + a^2 - 2a \cos(2\pi f)}.$$

(b).

Let  $l \geq 0$ . We get

$$r_{xx}[l] = \sum_{n=-\infty}^{\infty} x[n+l]x[n] = \sum_{n=-\infty}^{\infty} a^{n+l}u[n+l]a^n u[n] = a^l \sum_{n=0}^{\infty} a^{2n} = \frac{a^l}{1 - a^2}.$$

Let now  $l < 0$ . Using the symmetry of the autocorrelation function and the fact that  $-l > 0$  we get

$$r_{xx}[l] = r_{xx}[-l] = \frac{a^{-l}}{1 - a^2},$$

It follows that for any value of  $l$  we have

$$r_{xx}[l] = \frac{a^{|l|}}{1 - a^2},$$

We can now calculate  $S_{xx}(\omega)$  by taking the DTFT of  $r_{xx}[l]$

$$\begin{aligned}
S_{xx}(\omega) &= \sum_{l=-\infty}^{\infty} r_{xx}[l] e^{-j\omega l} = \sum_{l=-\infty}^{\infty} \frac{a^{|l|}}{1-a^2} e^{-j\omega l} \\
&= \frac{1}{1-a^2} \left( \sum_{l=-\infty}^{-1} a^{-l} e^{-j\omega l} + \sum_{l=0}^{\infty} a^l e^{-j\omega l} \right) = \frac{1}{1-a^2} \left( \sum_{l=1}^{\infty} a^l e^{j\omega l} + \sum_{l=0}^{\infty} a^l e^{-j\omega l} \right) \\
&= \frac{1}{1-a^2} \left( \sum_{l=0}^{\infty} a^{l+1} e^{j\omega(l+1)} + \sum_{l=0}^{\infty} a^l e^{-j\omega l} \right) = \frac{1}{1-a^2} \left( a e^{j\omega} \sum_{l=0}^{\infty} (a e^{j\omega})^l + \sum_{l=0}^{\infty} (a e^{-j\omega})^l \right) \\
&= \frac{1}{1-a^2} \left( \frac{a e^{j\omega}}{1-a e^{j\omega}} + \frac{1}{1-a e^{-j\omega}} \right) = \frac{1}{1-a^2} \cdot \frac{a e^{j\omega} - a^2 + 1 - a e^{-j\omega}}{(1-a e^{j\omega})(1-a e^{-j\omega})} \\
&= \frac{1}{(1-a e^{j\omega})(1-a e^{-j\omega})} = \frac{1}{1+a^2-2a \cos \omega},
\end{aligned}$$

which is the same as in 1a.

(c).

Plots of  $x[n]$ ,  $r_{xx}[l]$ , and  $S_{xx}(f)$  for different values of  $a$  are shown in Fig.1 – Fig.3.

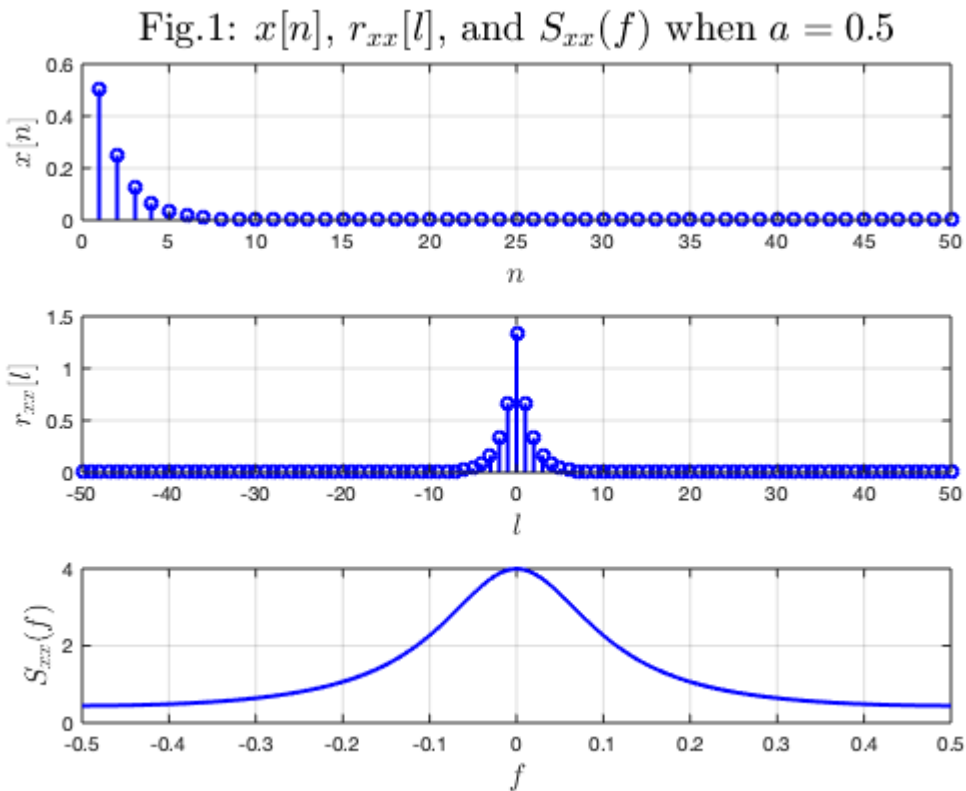
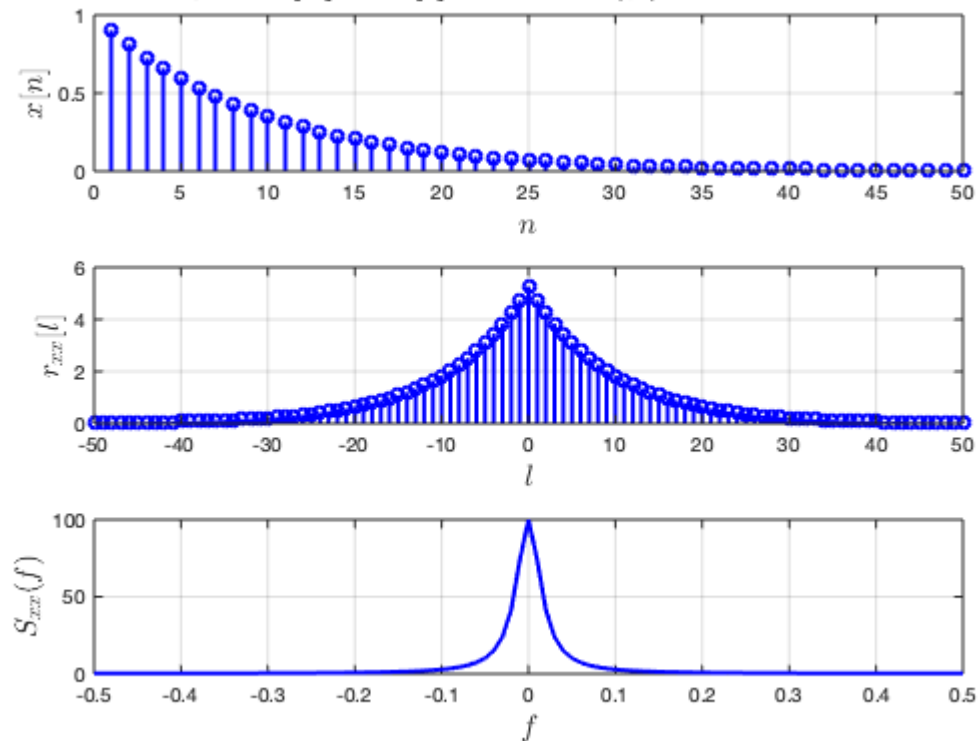
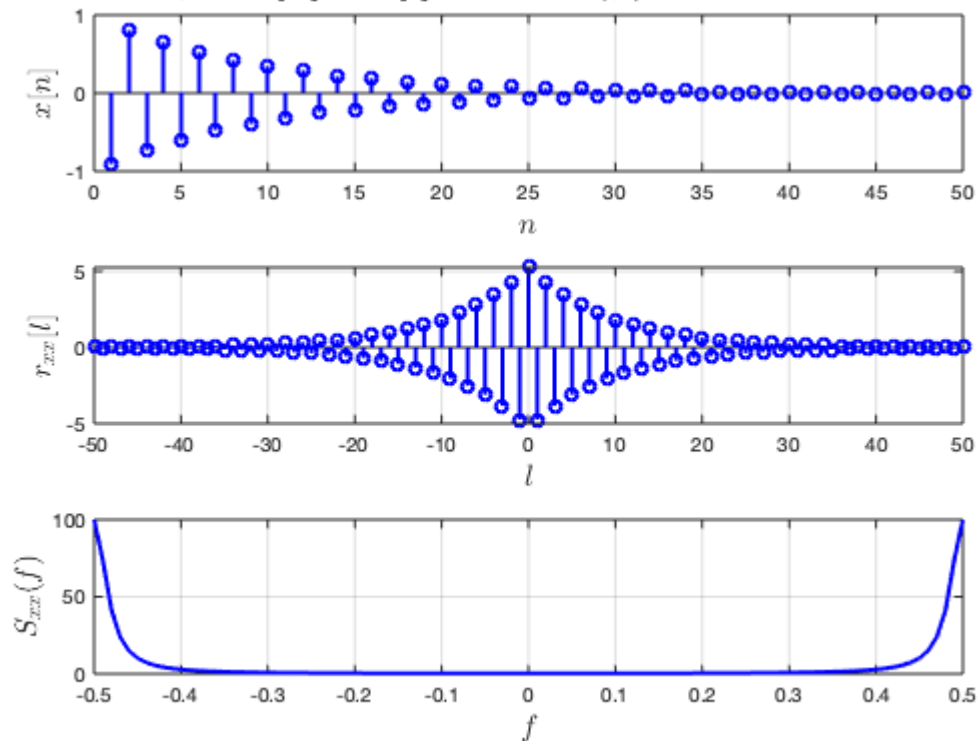


Fig.2:  $x[n]$ ,  $r_{xx}[l]$ , and  $S_{xx}(f)$  when  $a = 0.9$ Fig.3:  $x[n]$ ,  $r_{xx}[l]$ , and  $S_{xx}(f)$  when  $a = -0.9$ 

Let us first compare the plots for  $a = 0.5$  and  $a = 0.9$ . We observe that the signal  $x[n]$  varies much faster for  $a = 0.5$ . The correlation between two samples at distance  $l$  from each other is thus much smaller in this case. This can be verified by comparing the autocorrelation functions at some lag  $l$  (e.g.  $l = 10$ ). Furthermore, since the signal  $x[n]$  varies much faster for  $a = 0.5$ , it means that it contains

higher frequency components. This can be verified by comparing the energy spectral densities. It is easily seen that the energy is more concentrated at low frequencies for  $a = 0.9$  than for  $a = 0.5$ .

Let us now compare the plots for  $a = 0.9$  and  $a = -0.9$ . We see that for  $a = -0.9$  the signal changes very rapidly for adjacent samples ( $x[n+1] \approx -x[n]$ ), but the signal values at distance 2 from each other are rather similar. The same observation can be made by looking at the autocorrelation functions: the correlation between samples separated by a lag of 2, 4, 6, ... is as high as for the  $a = 0.9$ , while the correlation between samples at distance 1, 3, 5, ... from each other has a large negative value (it is large because the absolute value of the samples is similar, and negative because of the change in sign). The changes in sign between adjacent samples are quick variations in the signal, which means that the signal contains high frequency components. This is verified by looking at the energy spectral density, where it can be seen that all signal energy is concentrated around high frequencies. The two properties of the autocorrelation function that can be observed from the plots are that  $r_{xx}[l]$  is an even function (i.e.  $r_{xx}[-l] = r_{xx}[l]$ ), and that it attains its maximum at  $l = 0$ .

**(d).**

The energy of the signal is simply given by  $r_{xx}(0)$

$$E_x = \sum_{n=-\infty}^{\infty} x^2[n] = r_{xx}[0] = \frac{1}{1-a^2}.$$

**(e).**

When a signal is filtered, we have the following relation.

$$S_{yy}(f) = |H(f)|^2 S_{xx}(f)$$

The frequency response to the first filter is obtained as follows,

$$H_1(f) = \text{DTFT}\{\delta[n] - a\delta[n-1]\} = 1 - ae^{-j\omega}$$

and

$$|H_1(f)|^2 = (1 - ae^{-j\omega})(1 - ae^{j\omega}) = 1 + a^2 - 2a \cos(2\pi f)$$

Note that  $h_1(n)$  is actually the inverse of  $x(n)$ . If we denote the result of the first filtering stage as  $y_1(n)$ , then

$$S_{y_1 y_1}(f) = |H_1(f)|^2 S_{xx}(f) = 1$$

The second filter is lowpass filter with cut-off at  $f = 1/4$ . Thus, we get

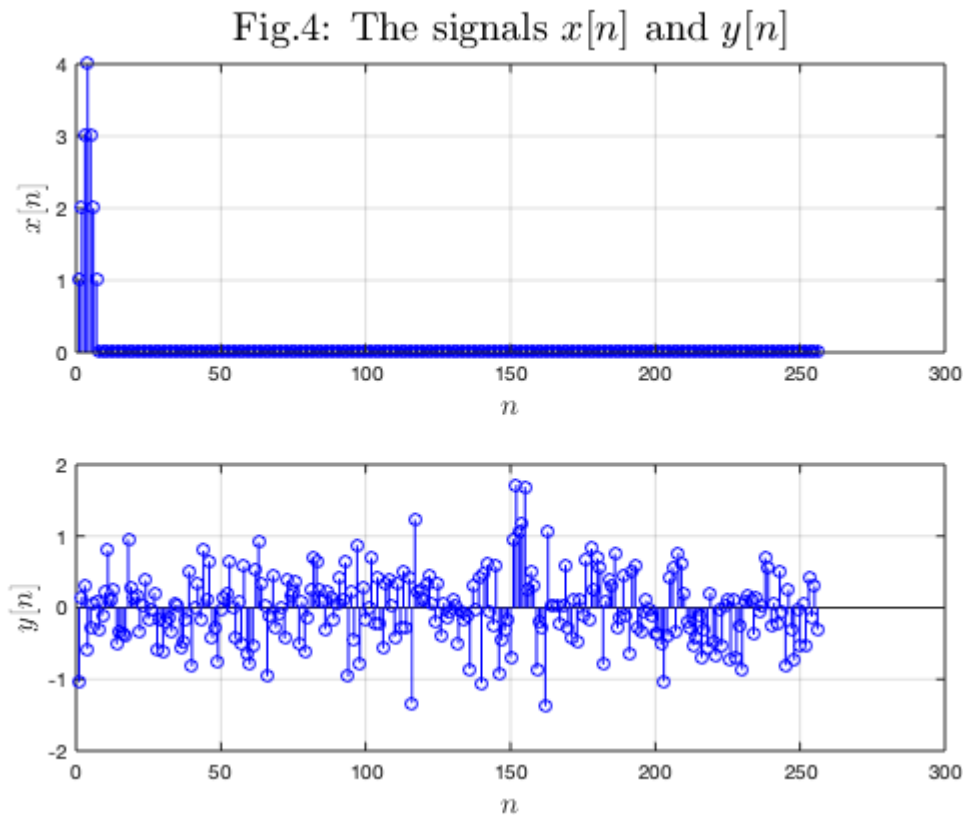
$$S_{yy}(f) = |H_2(f)|^2 S_{y_1 y_1}(f) = \begin{cases} \cos^2(2\pi f) & |f| \leq \frac{1}{4} \\ 0 & \frac{1}{4} < |f| \leq \frac{1}{2} \end{cases}.$$

Filtering by the lowpass filter removes all frequency components higher than the cut-off frequency. The entire energy of the output signal is thus contained in the frequency range  $|f| \leq \frac{1}{4}$ . The energy of the output signal can be found using

$$E_y = \int_{-\frac{1}{2}}^{\frac{1}{2}} S_{yy}(f) df = \int_{-\frac{1}{4}}^{\frac{1}{4}} \cos^2(2\pi f) df = \frac{1}{4}$$

**Problem 2:****(a).**

Plots of the signals  $x[n]$  and  $y[n]$  are shown in Fig.4.

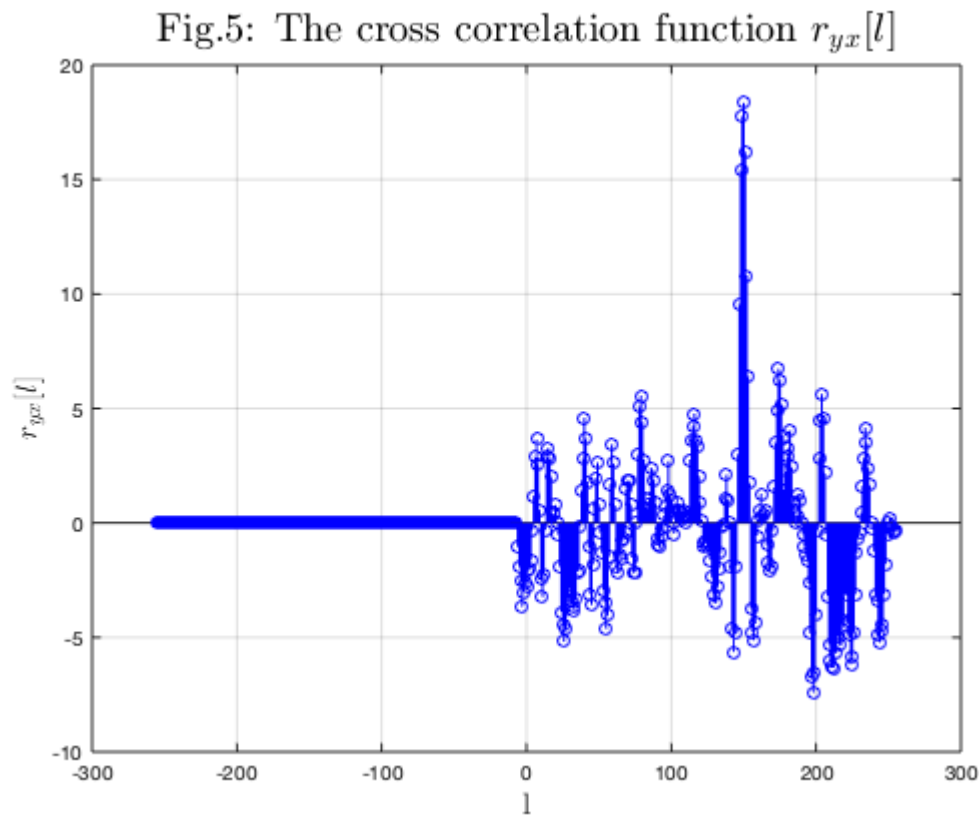


Due to the presense of a high noise component in the received signal  $y(n)$ , it is not easy to conclude wether  $y(n)$  also contains a reflection of the signal  $x(n)$ .

**(b).**

Matlab code to generate and plot  $r_{yx}(l)$  is given below.

```
r=xcorr(y, x);
L=length(y)-1;
l=-L:L;
stem(l,r,'b');
xlabel('l')
ylabel('r_yx[l]')
```



Plot of  $r_{yx}[l]$  is shown in *Fig.5*.

**(c).**

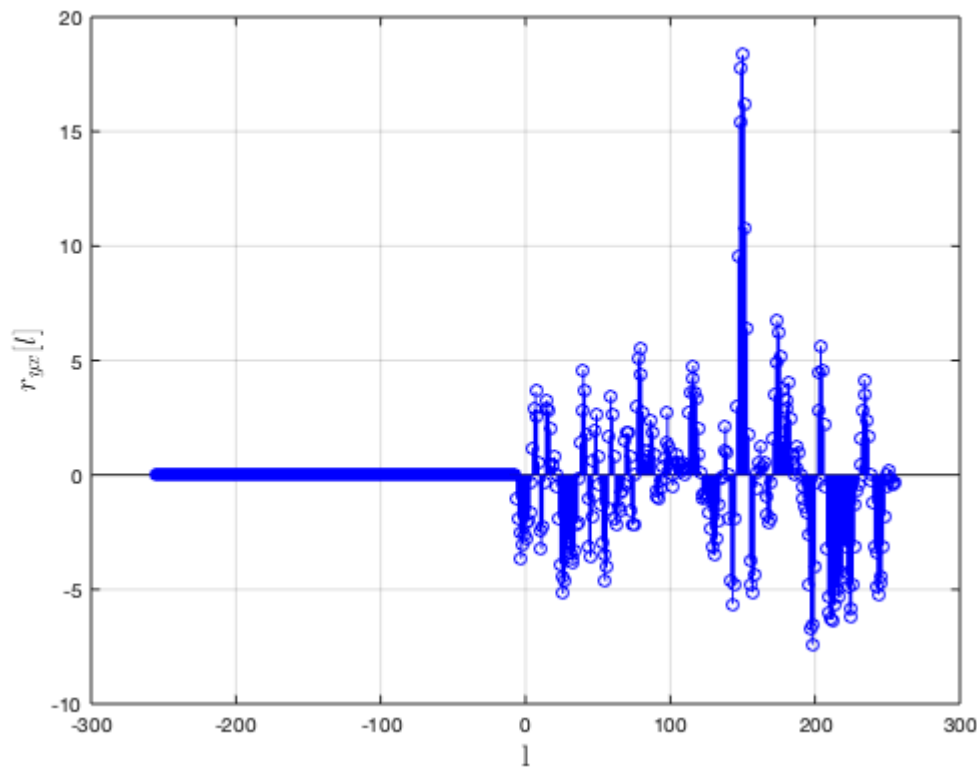
The crosscorrelation function is given by

$$r_{yx}[l] = y[l] * x[-l]$$

The following Matlab code can be used to calculate the crosscorrelation function.

```
% reverse the vector x
x_rev = fliplr(x);
r2 = conv(y, x_rev);
```

It produces the identical result as the one in **2b** as follows.



(d).

The crosscorrelation function has a distinct peak, which suggests that the signal  $y[n]$  contain a reflected component of the emitted signal  $x[n]$ . The crosscorrelation function attains its maximum at  $l = 150$ , so this value is our best estimate for the delay  $D$ . This method is much more reliable than the direct comparison of the plots of the emitted and the received signals.

### Problem 3:

(a).

We can find the transfer function as:

$$Y(z) = X(z) + \alpha z^{-R} X(z) = X(z)(1 + \alpha z^{-R})$$

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \alpha z^{-R}$$

(b).

The relationship between the delay in seconds,  $D_s$ , and the delay  $R$  is  $D_s = \frac{R}{F_s}$ .

(c).

The filter can be implemented as:

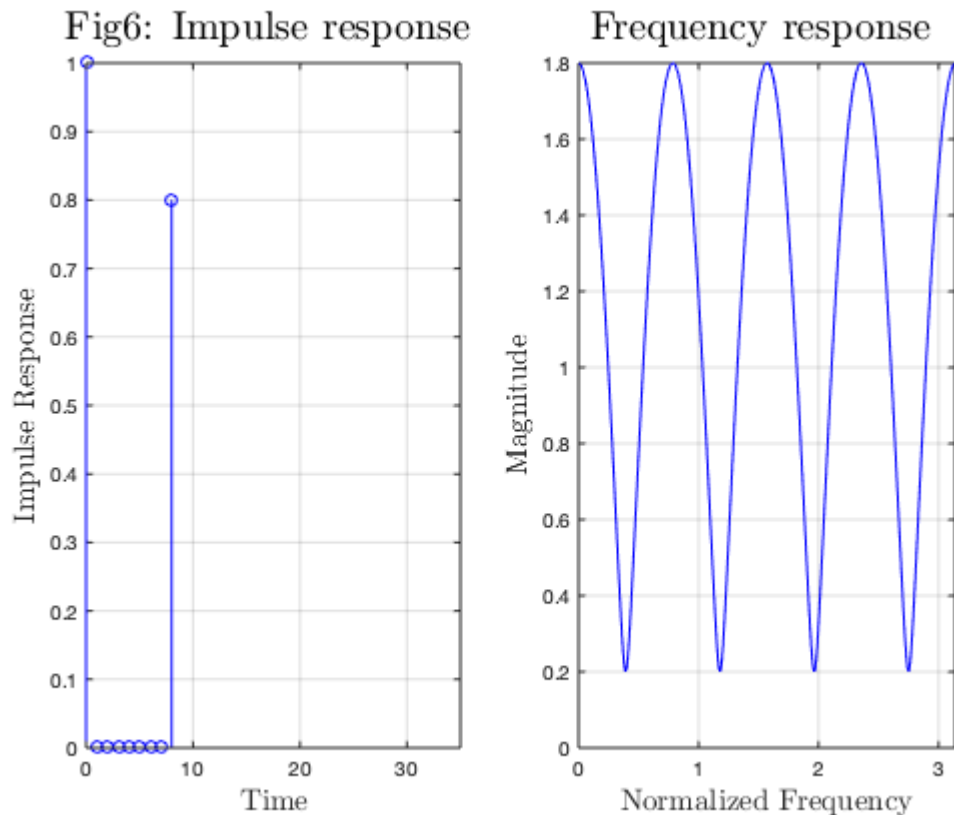
An example of the impulse/frequency response for single-echo filter with  $R = 8$  and  $\alpha = 0.8$  is given in Fig.6 left and right, respectively.

```

R=8;
alpha=.8;
num = [1,zeros(1,(R-1)),alpha];
den = [1];
[H,w] = freqz(num,den);
subplot(1,2,2)
plot(w,abs(H),'b')
ylabel('Magnitude')
subplot(1,2,1)
[h,n] = impz(num,den);
stem(n,h,'b')
ylabel('Impulse Response')

wavfile='piano.wav';
[x,fs]=audioread(wavfile);
y=filter(num,den,x);

```



(d).

A larger  $R$  and  $\alpha$  gives a more pronounced echo effect.

(e).

We find the transfer function as:



$$Y(z) = E(z) - E(z)z^{-NR}\alpha^N$$

$$E(z) = X(z) + E(z)z^{-R}\alpha$$

$$E(z)(1 - z^{-R}\alpha) = X(z) \rightarrow E(z) = \frac{X(z)}{1 - z^{-R}\alpha}$$

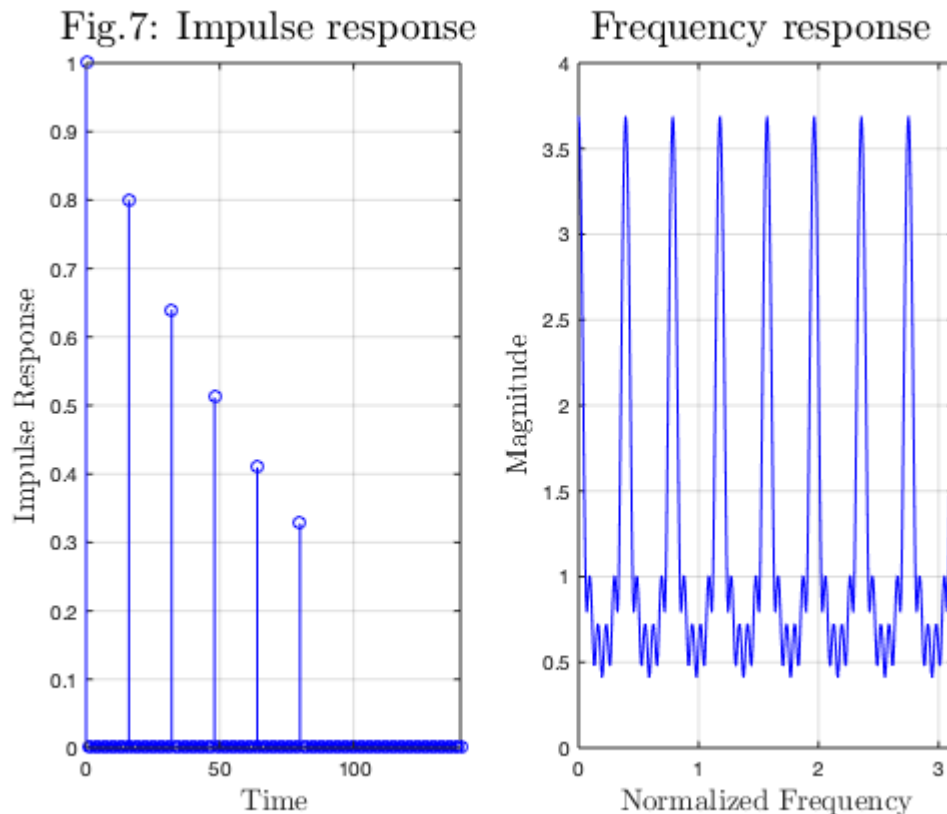
$$Y(z) = E(z)(1 - \alpha^N z^{-NR}) = X(z) \frac{1 - \alpha^N z^{-NR}}{1 - z^{-R}\alpha}$$

$$H(z) = \frac{1 - \alpha^N z^{-NR}}{1 - z^{-R}\alpha}$$

An example of the impulse/frequency response for the multiple echo filter with  $R = 16$ ,  $N = 6$  and  $\alpha = 0.8$  is given in Fig.7 left and right, respectively. This can be implemented in MATLAB as:

```
N=6;
R=16;
alpha=.8;
num_H = [1,zeros(1,(N*R-1)),-alpha^N];
den_H = [1,zeros(1,(R-1)),-alpha];
y=filter(num_H,den_H,x);
```

Playing \*filtered\* sound clip



(f).

We can in the multiple filter observe that we have multiple spikes in the impulse response. Each spike indicates an echo.

(g).

*After filtering the sound will have multiple echoes. This will give the sound a more spacious sound when compared to the original sound file, which only consist of the direct sound.*

**(h).**

*In a real room the sound will have multiple echoes from all the walls. Hence the filter with multiple echoes will give the most "natural" sound.*