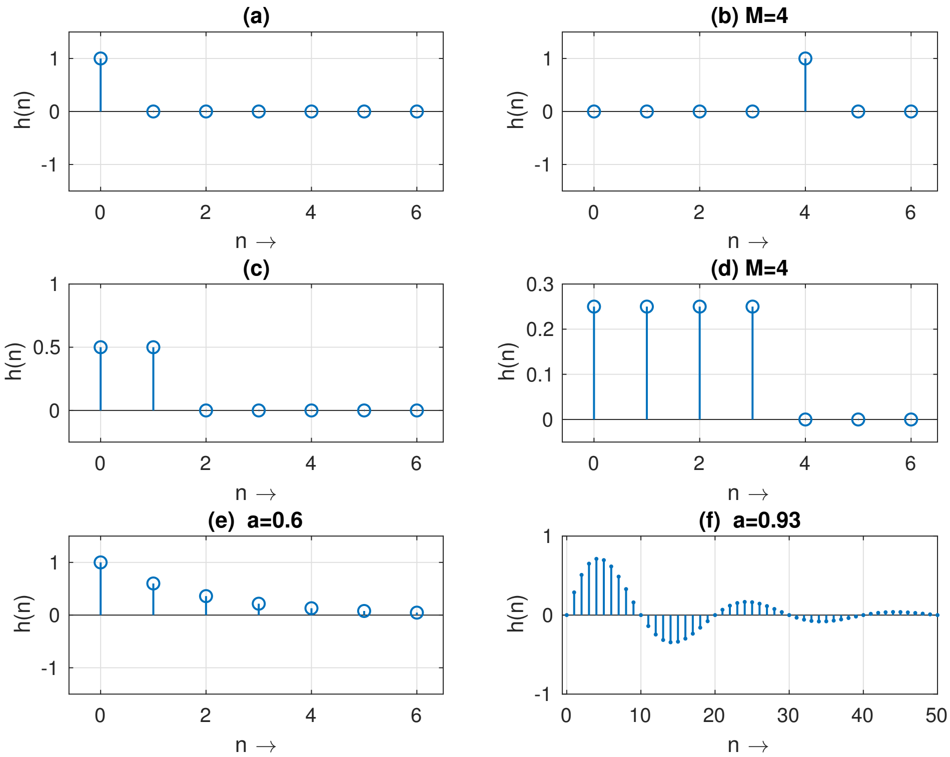
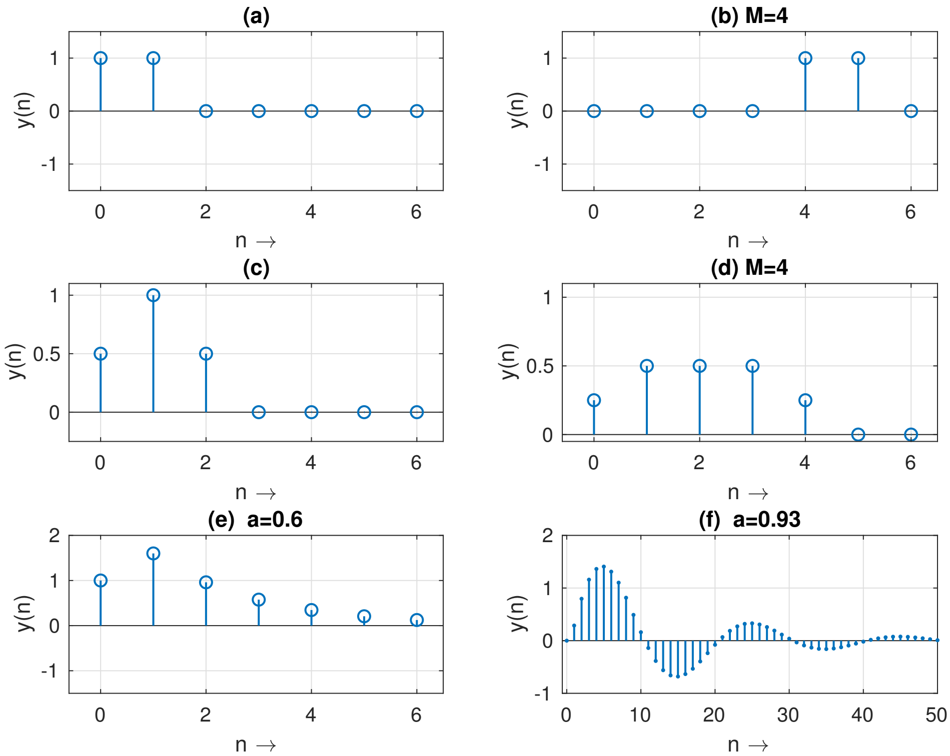
DASP3rd Chapter 1 - Exercises

1 Signals and Systems

1. Sketch the impulse responses (a-f)

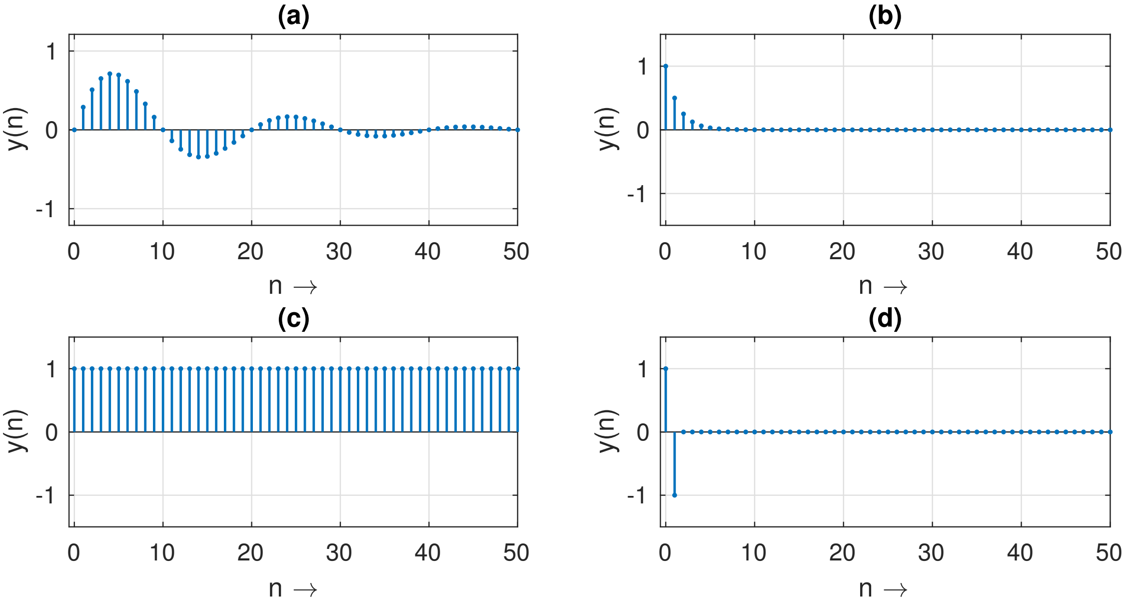


and compute the output signals (a-f) using convolution.

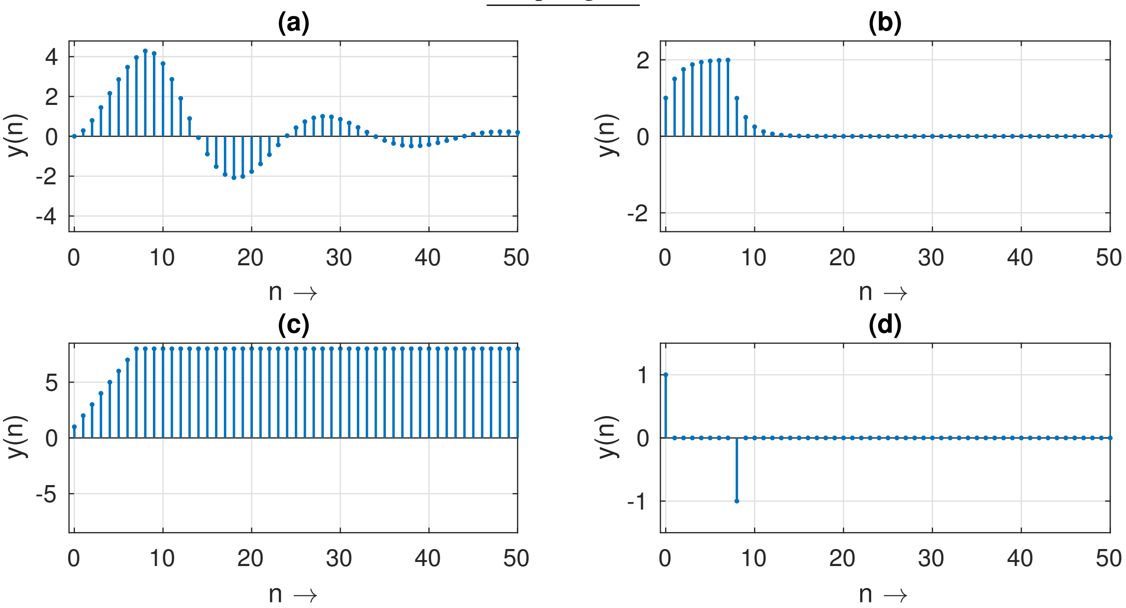


2. Compute the output signals (a-d) for two different input signals (unit pulse and 8-tap signal).

unit pulse



8-tap signal



2 Discrete-time Fourier Transform

1. Compute the discrete-time Fourier transform of (a-d)

$$(a) \qquad X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} \delta(n) e^{-j\Omega n} = 1$$

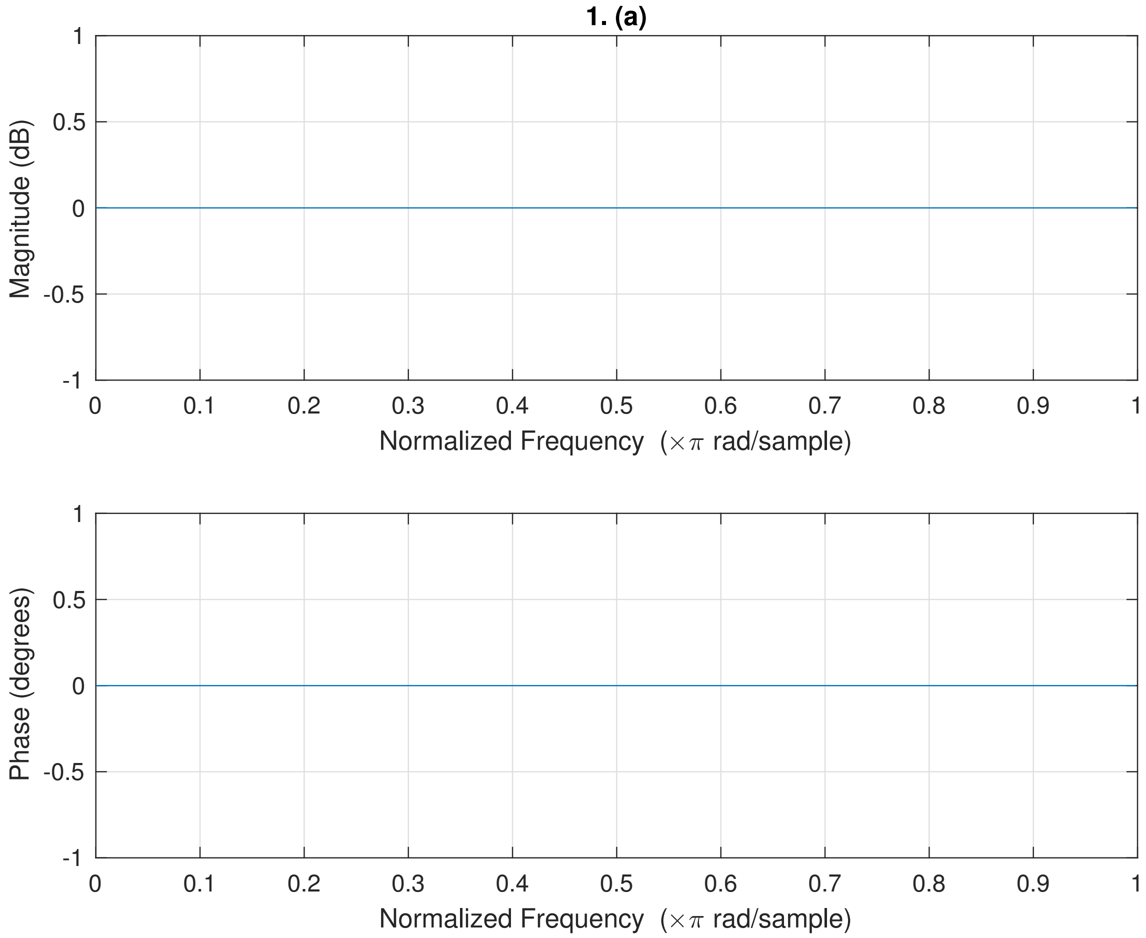
$$(b) \qquad X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} \delta(n-M) e^{-j\Omega n} = e^{-j\Omega M}$$

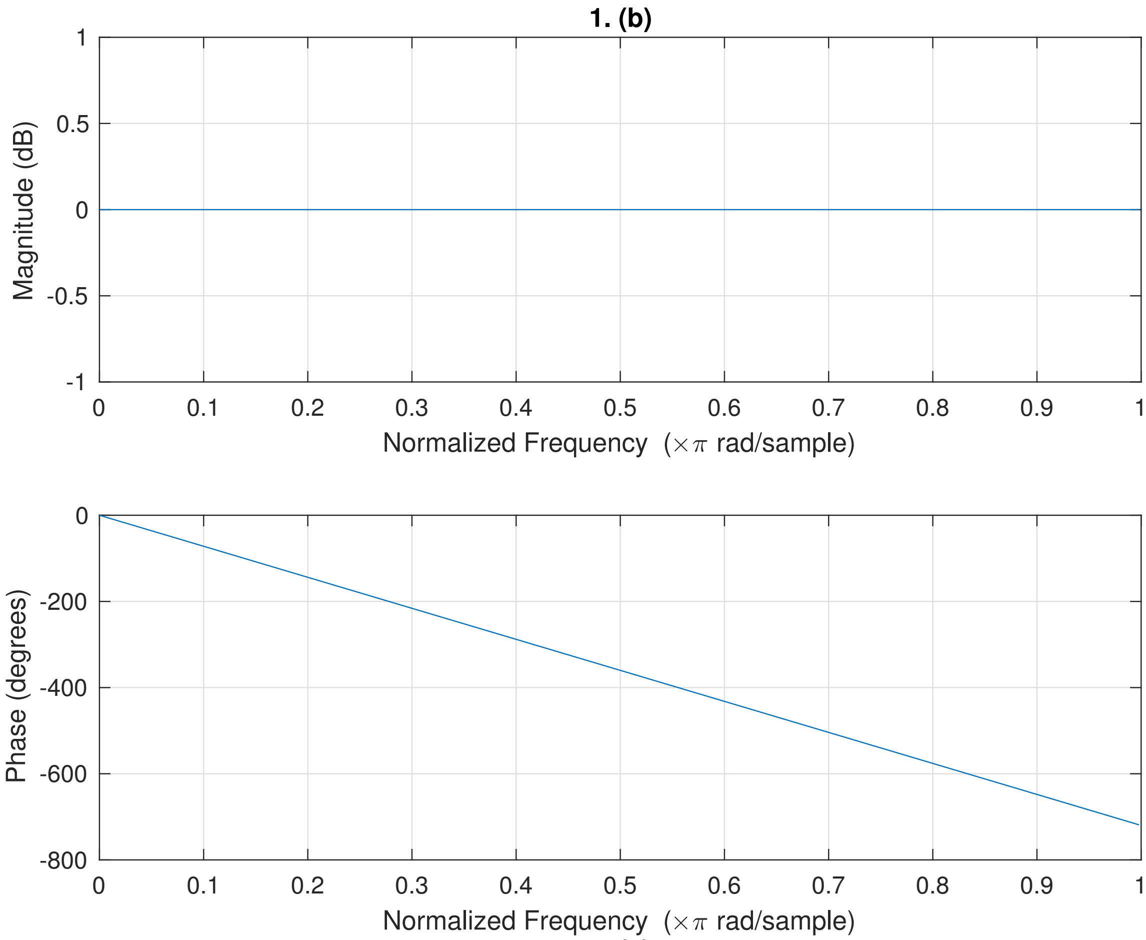
$$(c) \qquad X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} \frac{1}{2}\left[\delta(n) + \delta(n-1)\right] e^{-j\Omega n} = \frac{1}{2}(1 + e^{-j\Omega})$$

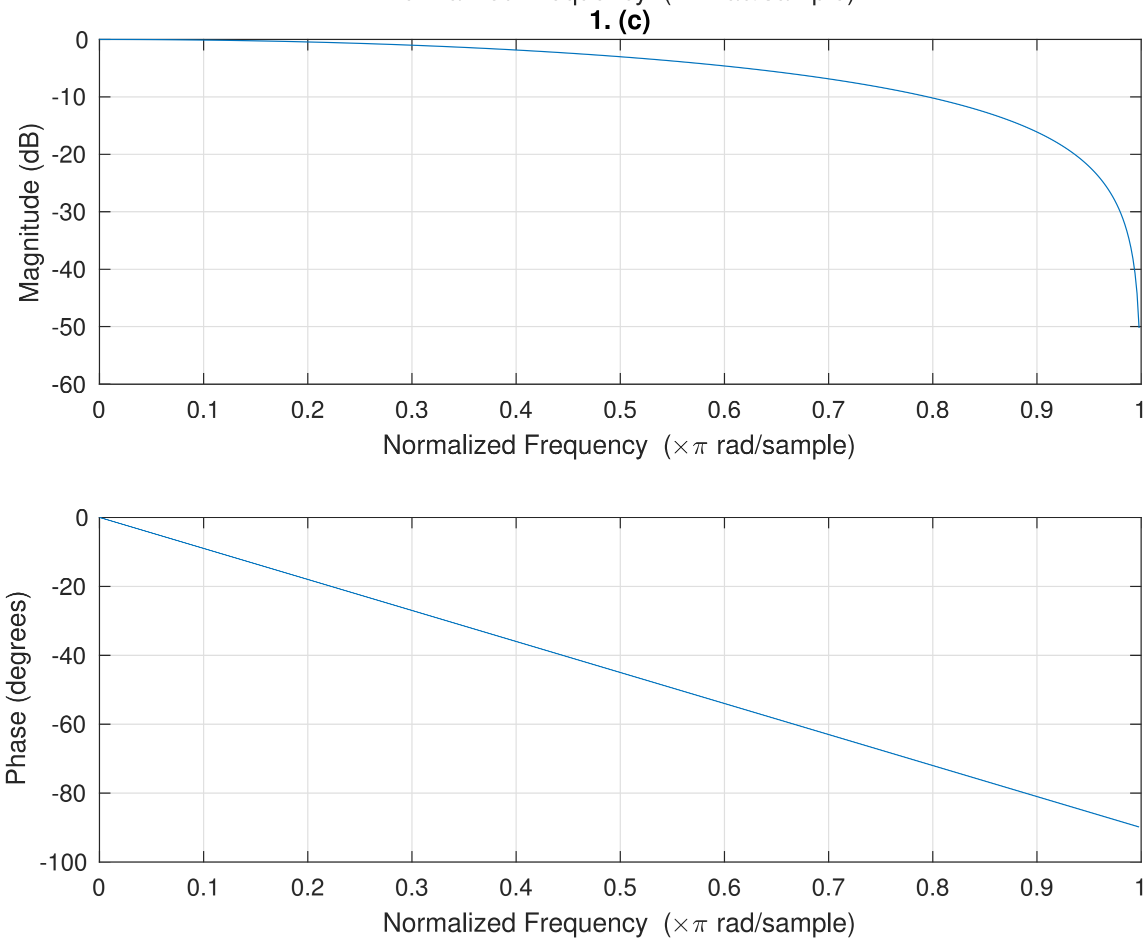
$$(d) \qquad X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} a^n \epsilon(n) e^{-j\Omega n} = \sum_{n=0}^{\infty} a^n e^{-j\Omega n}$$

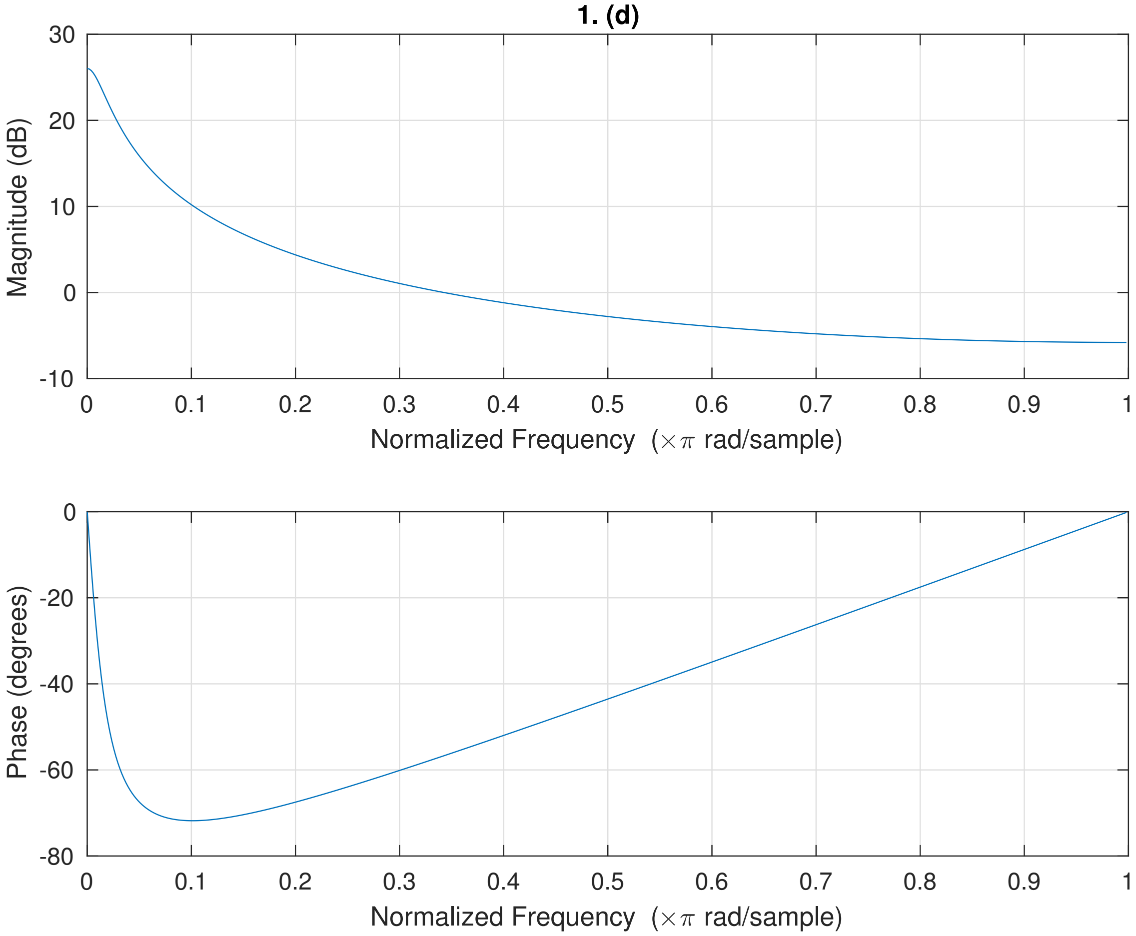
$$= \sum_{n=0}^{\infty} \left(\frac{a}{e^{j\Omega}}\right)^n = \frac{1}{1 - \frac{a}{e^{j\Omega}}} = \frac{1}{1 - ae^{-j\Omega}}, |\, a| < 1$$

and plot the results using Matlab.




1. (b)
Magnitude (dB)
1
0.5
0
-0.5
-1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
Phase (degrees)
0
-200
-400
-600
-800
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)

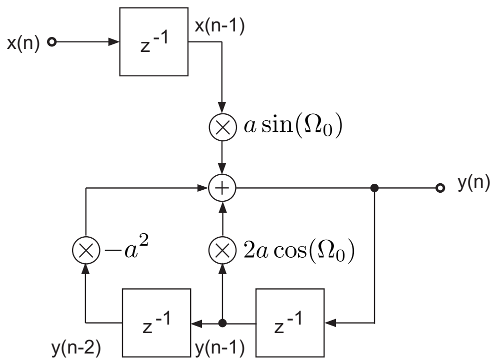

1. (c)
Magnitude (dB)
0
-10
-20
-30
-40
-50
-60
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
Phase (degrees)
0
-20
-40
-60
-80
-100
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)



2. Using the difference equation

$$y(n) = a\sin(\Omega_0)\cdot x(n-1) + 2a\cos(\Omega_0)\cdot y(n-1) - a^2 y(n-2)$$

(a) sketch the signal flow graph



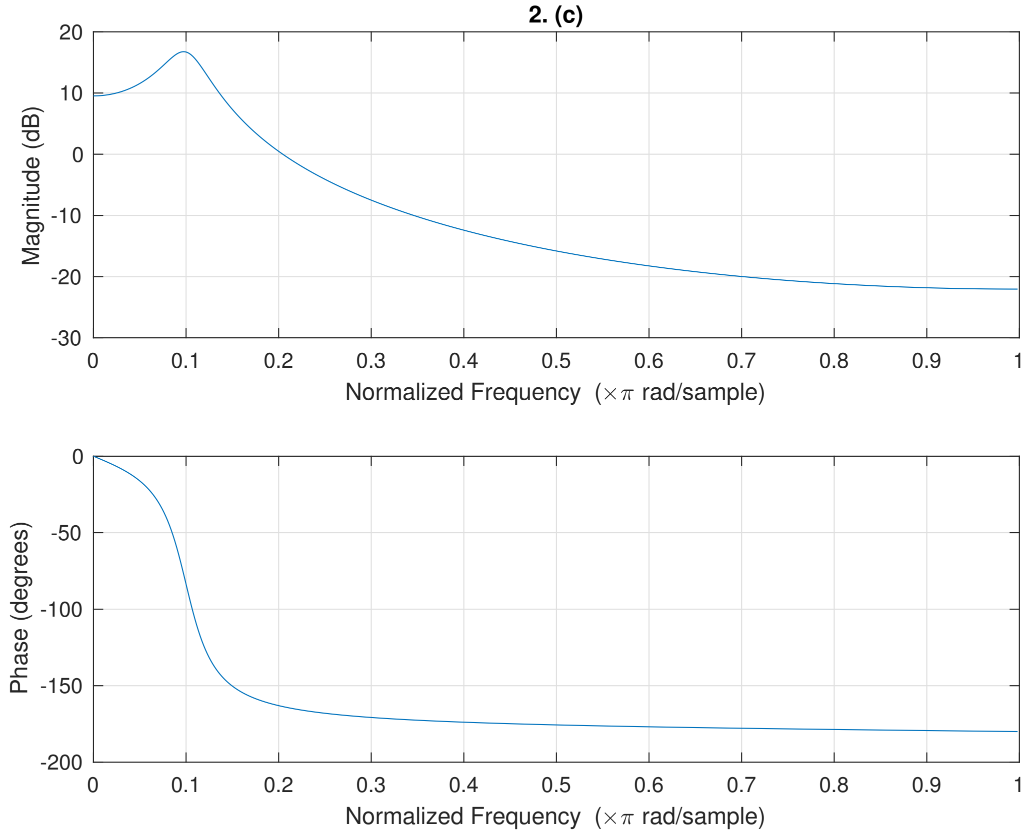
(b) compute the frequency response using the discrete-time Fourier transform of the difference equation gives

$$Y(e^{j\Omega}) = a\sin(\Omega_0)\cdot X(e^{j\Omega})e^{-j\Omega} + 2a\cos(\Omega_0)\cdot Y(e^{j\Omega})e^{-j\Omega} - a^2 Y(e^{j\Omega})e^{-2j\Omega}$$

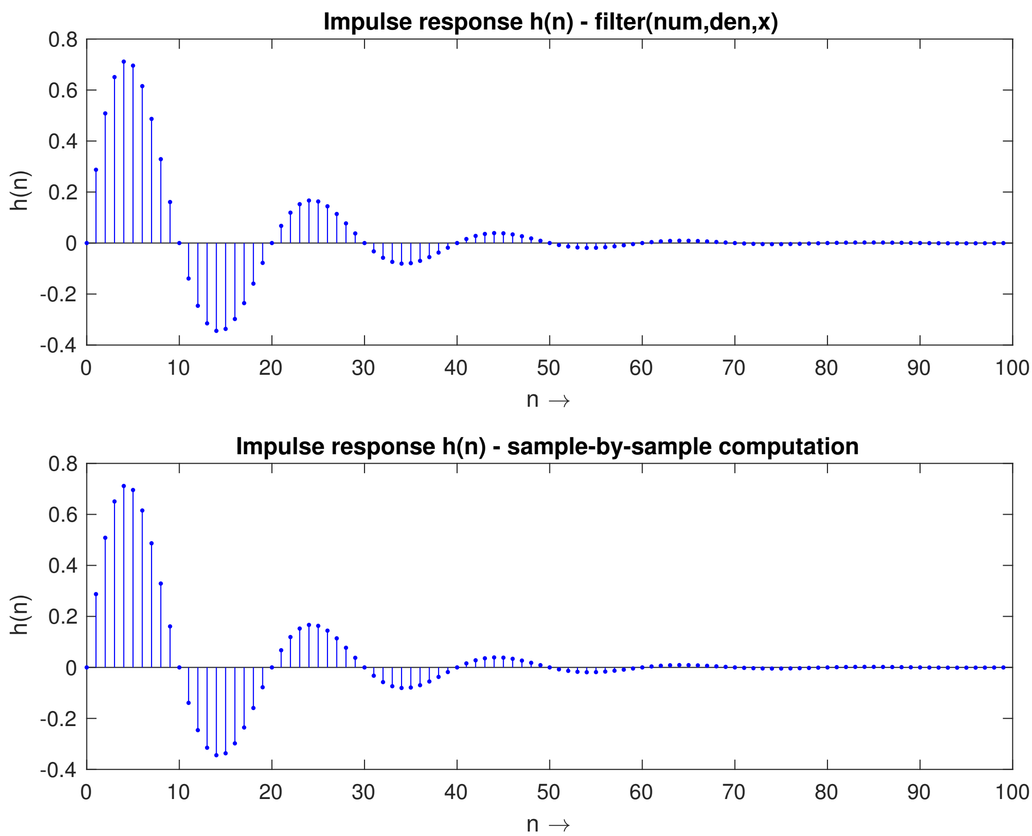
and thus the frequency response according to

$$H(e^{j\Omega}) = \frac{Y(e^{j\Omega})}{X(e^{j\Omega})} = \frac{a\sin(\Omega_0)e^{-j\Omega}}{1 - 2a\cos(\Omega_0)e^{-j\Omega} + a^2 e^{-2j\Omega}}$$

(c) plot the magnitude and phase response frequency using Matlab



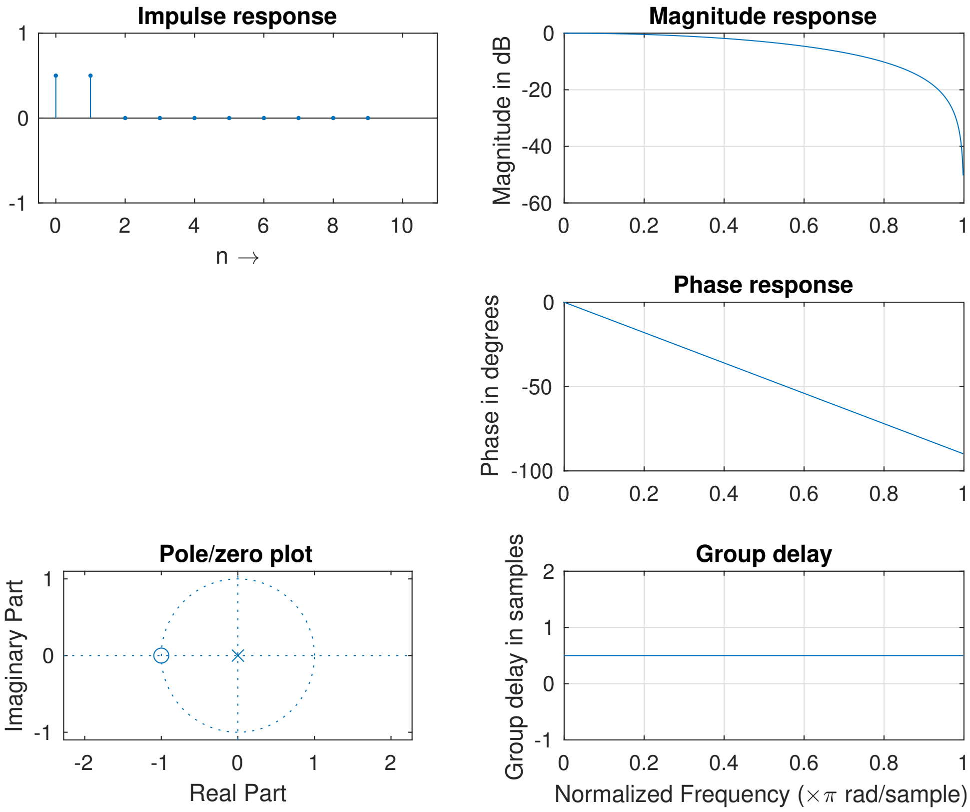
(d) program the difference equation and plot the impulse response



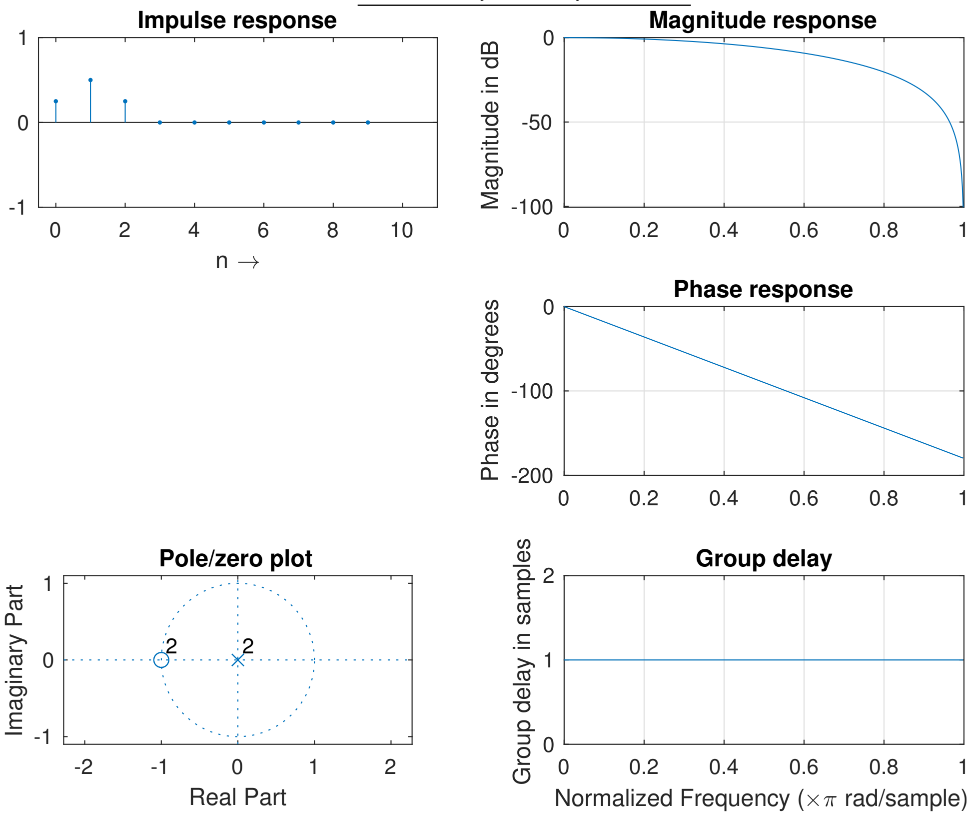
3 FIR and IIR Filters

1. FIR filters with linear phase (a-d)

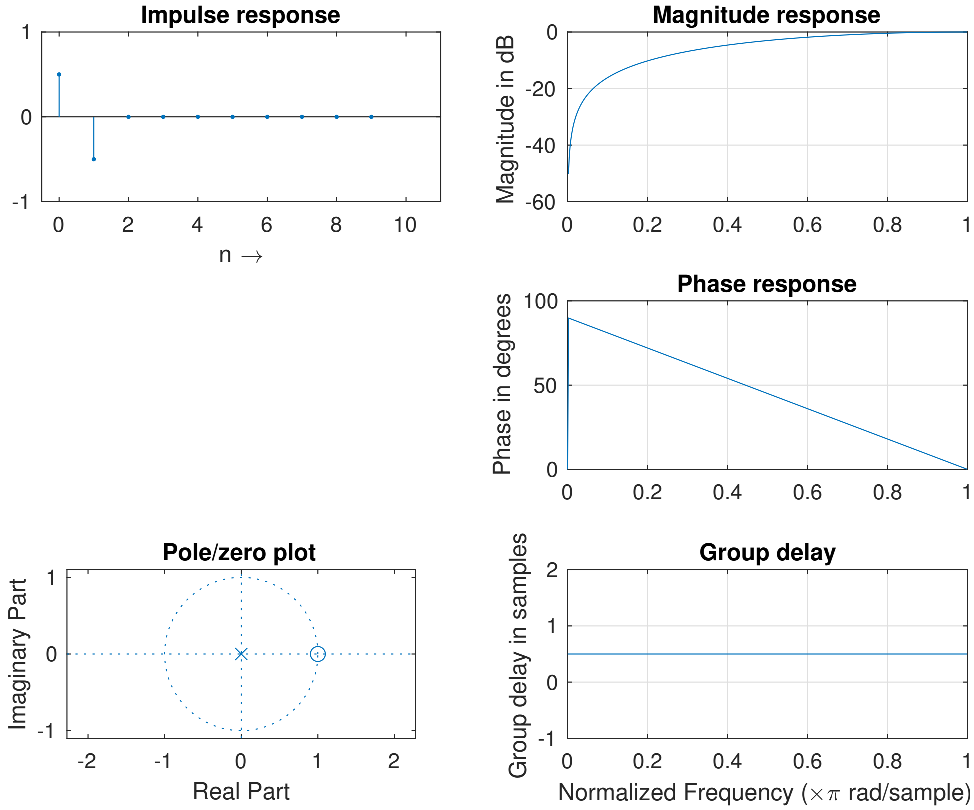
(a) even symmetry, $N = 2$



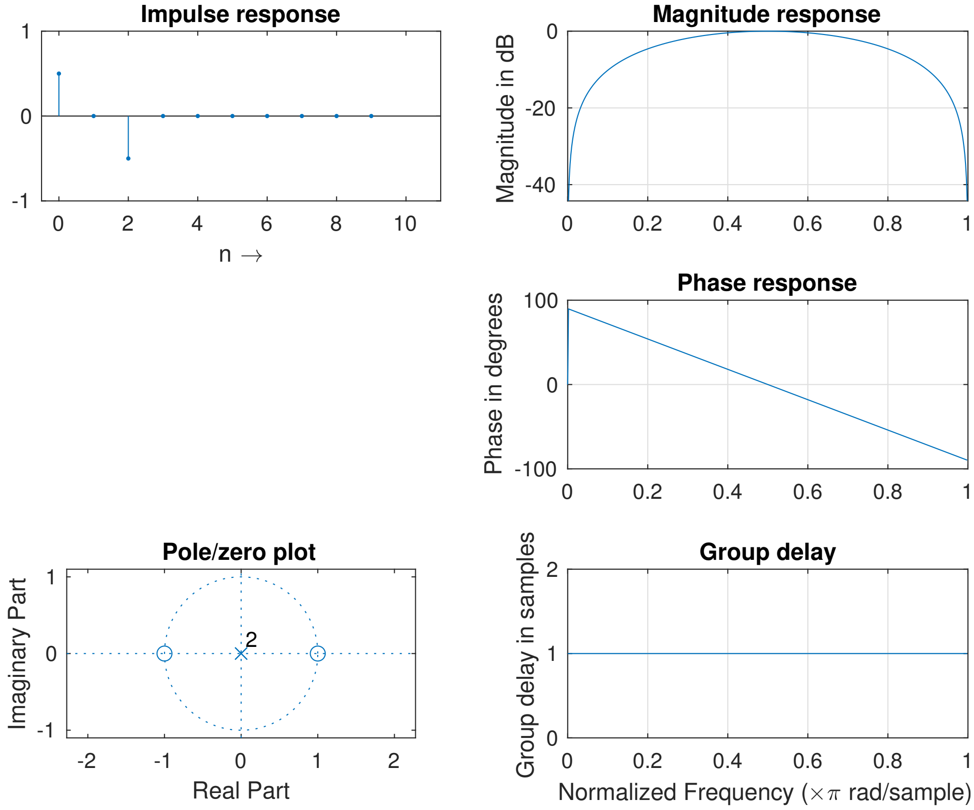
(b) even symmetry, $N = 3$



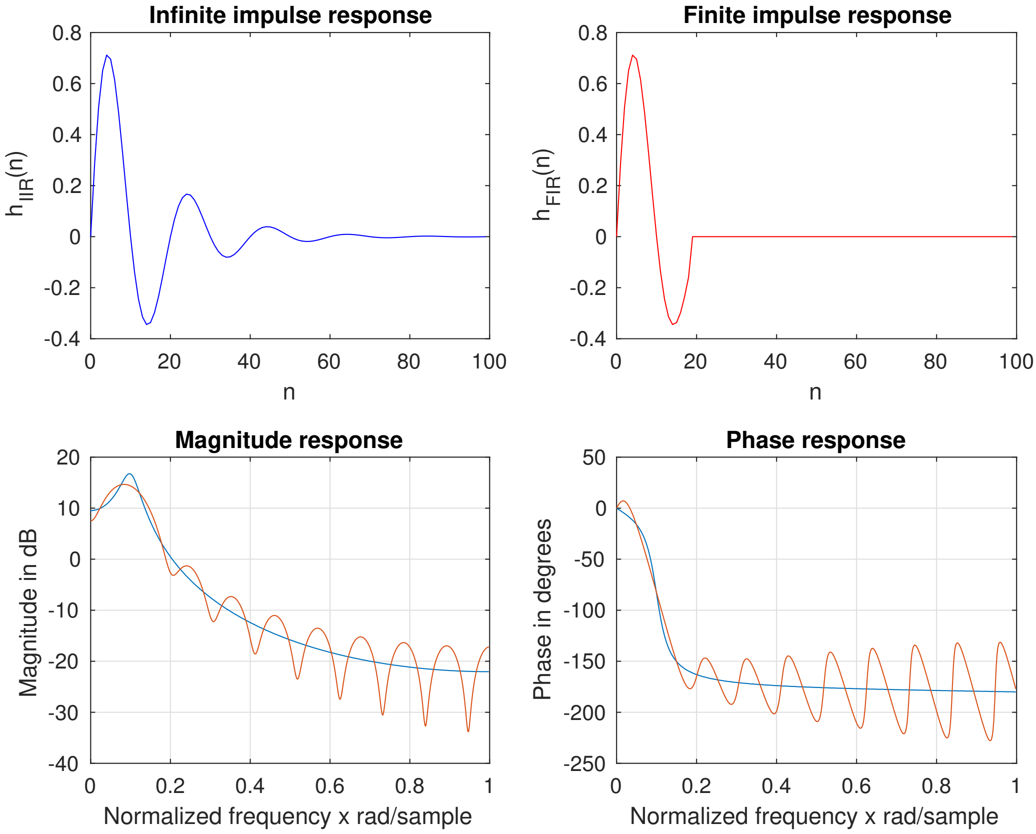
(c) odd symmetry, $N = 2$



(d) odd symmetry, $N = 3$

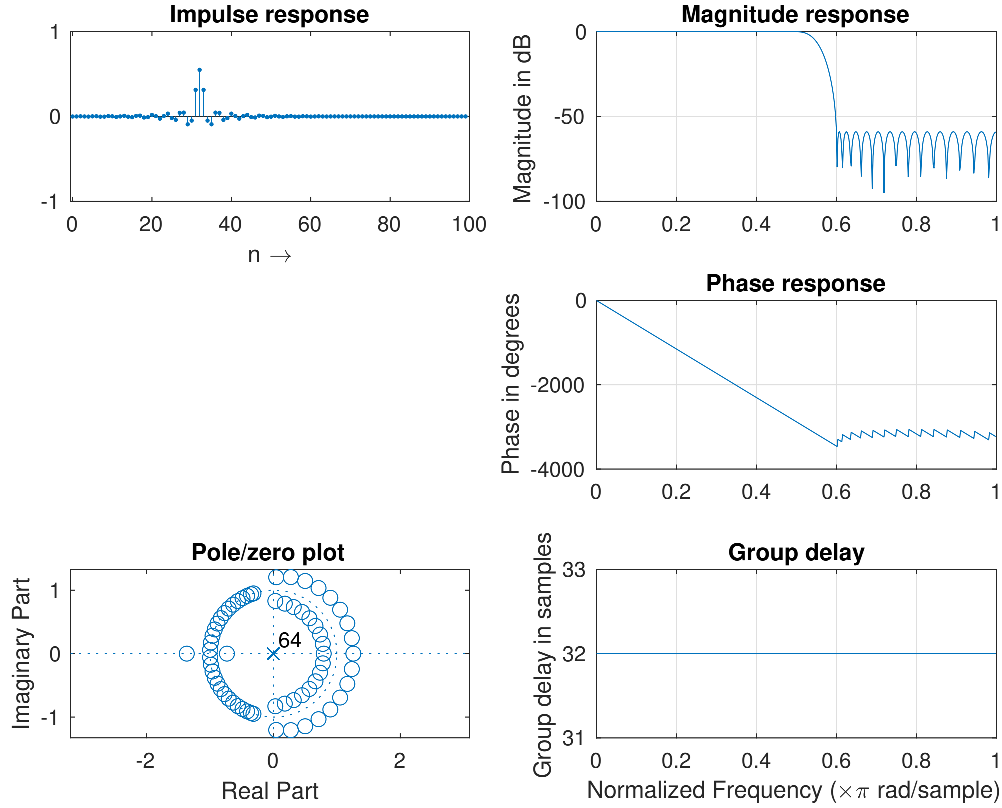


2. FIR filters using truncated impulse responses



3. FIR filters using PM algorithm lead to symmetric impulse responses and linear phase behavior

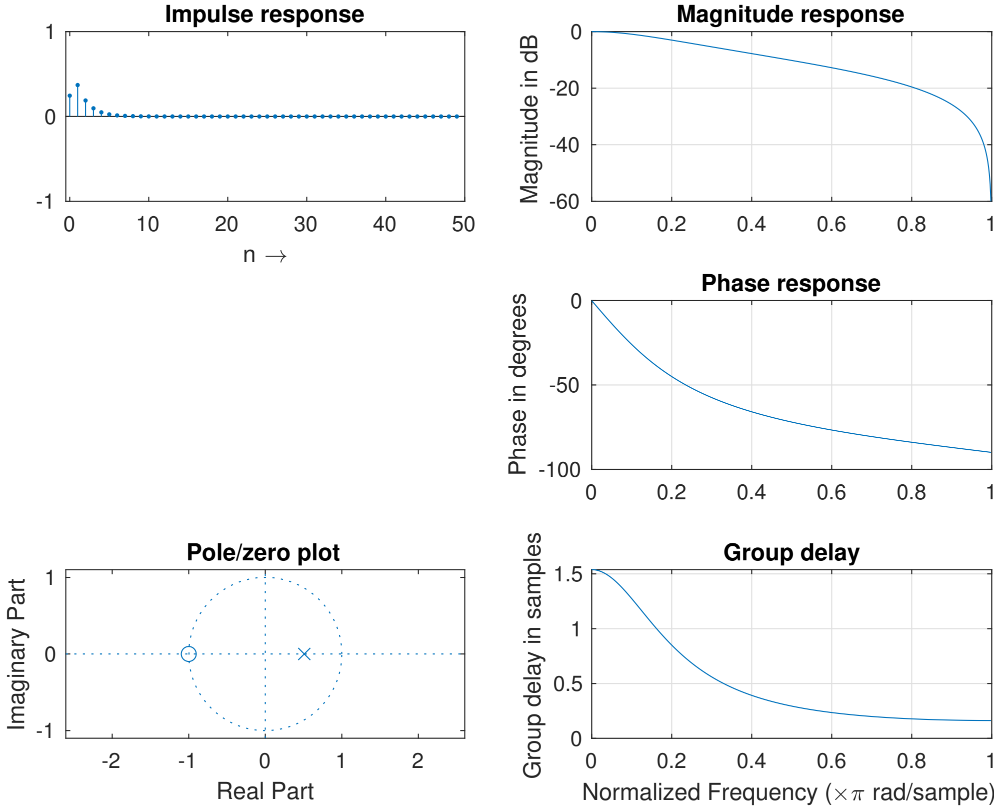
```
N=65, h=firpm(N-1,[0 0.5 0.6 1],[1 1 0 0],[1 1])
```



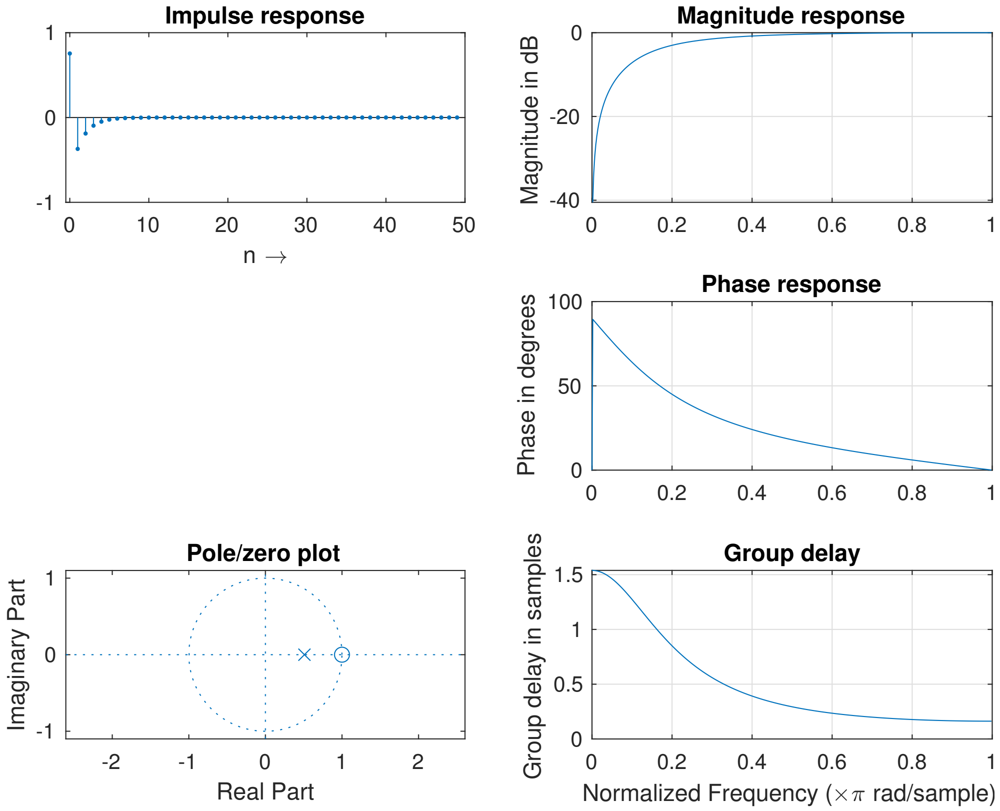
4. Simple IIR filters and their frequency responses

```
fc=4410;fs=44100;
a=(1-sin(2*pi*fc/fs))/cos(2*pi*fc/fs);
% Lowpass
num=(1-a)/2*[1 1];
den=[1 -a];
% Highpass
num=(1+a)/2*[1 -1];
den=[1 -a];
% Bandpass
fb= 4410;
a=(1-sin(2*pi*fb/fs))/cos(2*pi*fb/fs);
b=cos(2*pi*fc/fs);
num=(1-a)/2*[1 0 -1];
den=[1 -b*(1+a) a];
% plot magnitude and phase responses
```

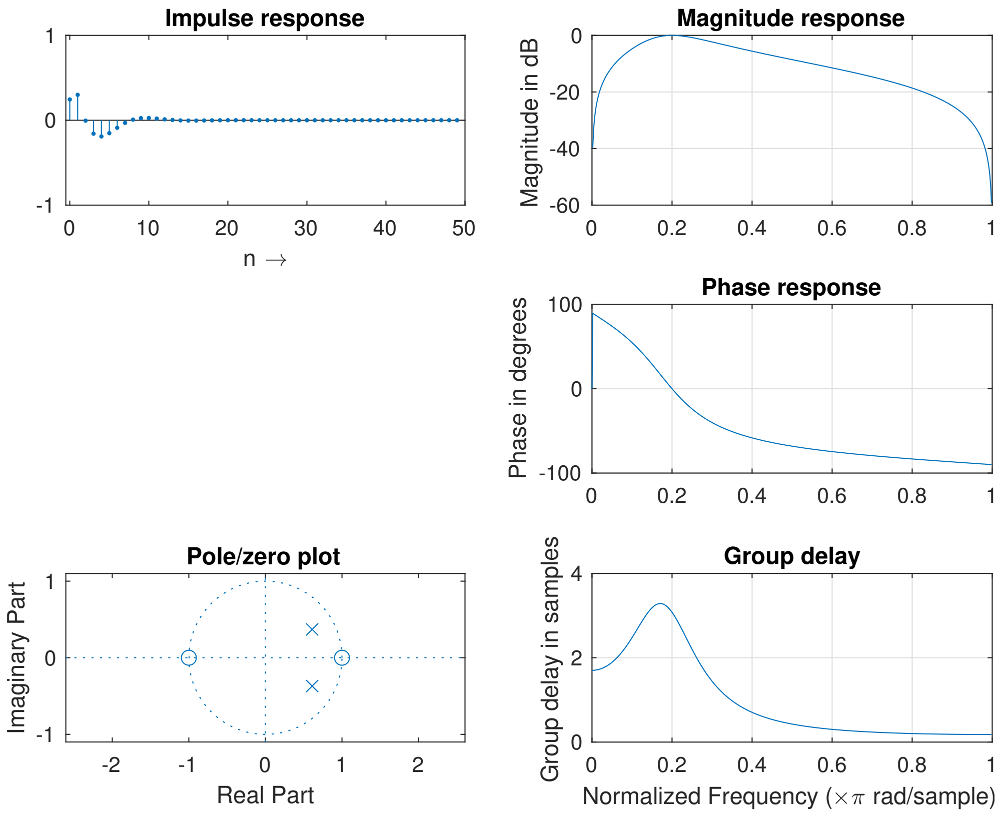
Lowpass



Highpass

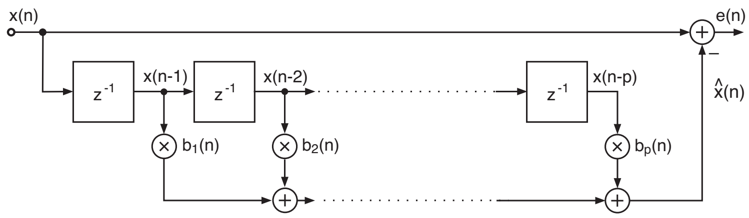


Bandpass

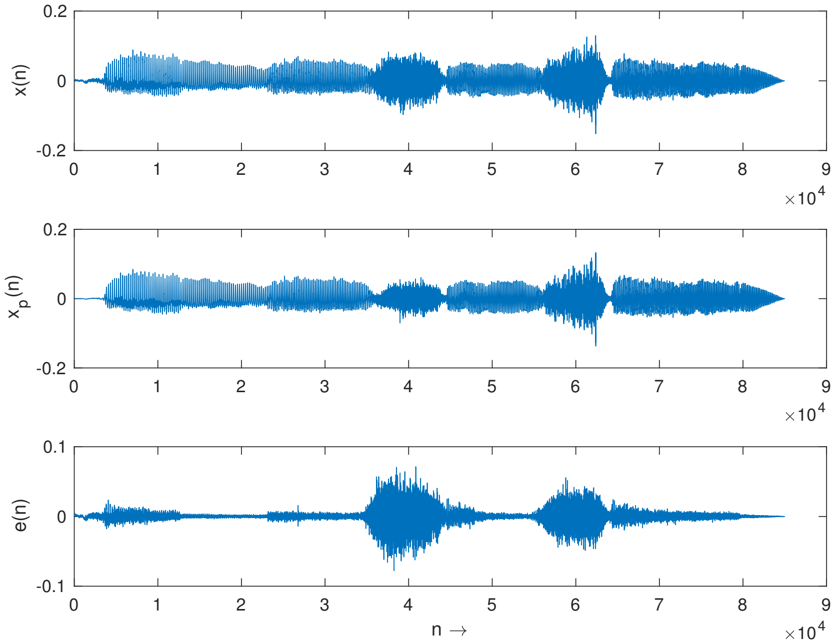


4 Adaptive Filters

1. Linear prediction using LMS algorithm

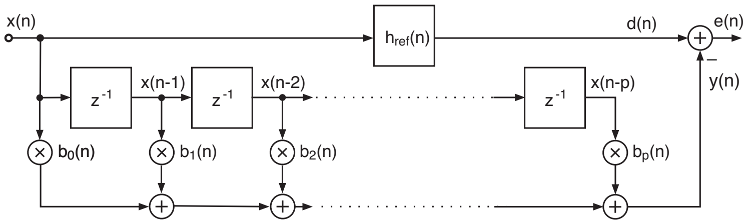



```
clf,clear;
[X, FS] = audioread('aint-no-sunshine.wav') ;
Nx=length(X);
N=32;
mu=1.5;

x_buff=zeros(1,N); % filter input
h=zeros(1,N);      % prediction coeffs
E=zeros(1,Nx);     % predicton error
Xp=zeros(1,Nx);    % prediction output

for n=1:Nx
Xp(n) = x_buff*h';   % prediction
E(n) = X(n) - Xp(n); % prediction error
for i=1:N % Update of coefficients with LMS
h(i) = h(i) + mu*x_buff(i)*E(n);
end
x_buff = [X(n), x_buff(1:N-1)]; %newest input sample first
end

subplot(311),plot(X);ylabel('x(n)');
subplot(312),plot(Xp);ylabel('x_p(n)');
subplot(313),plot(E);ylabel('e(n)');
```

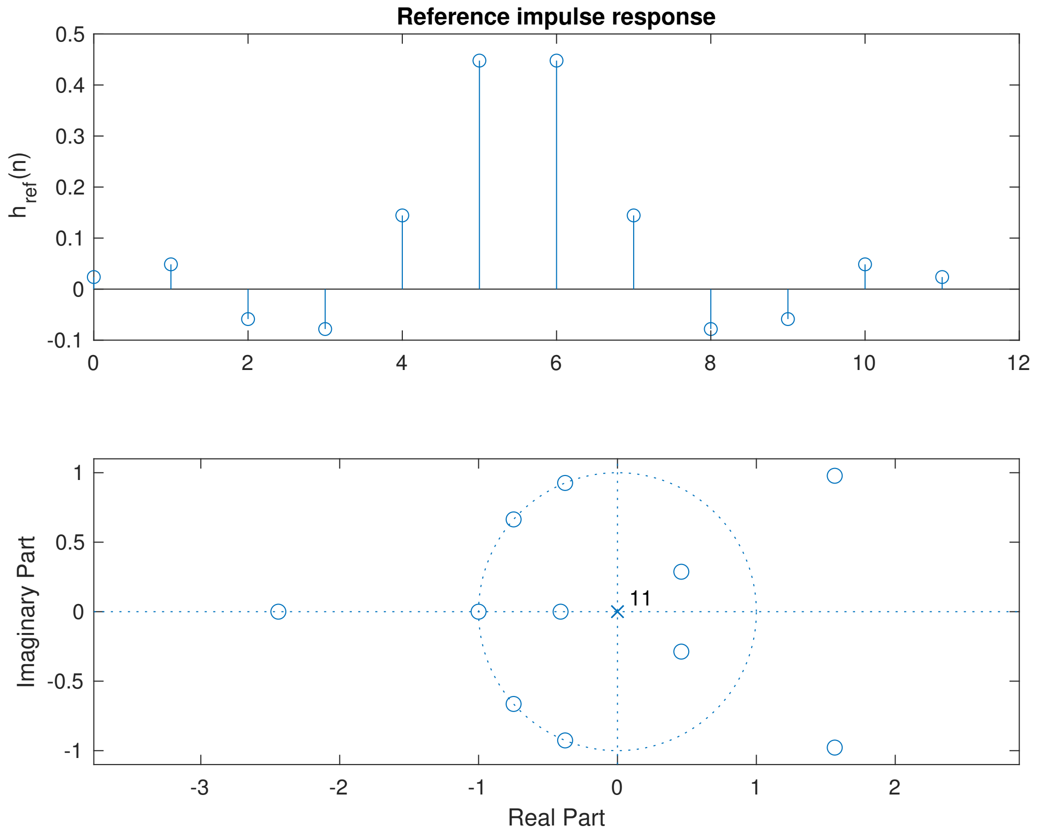


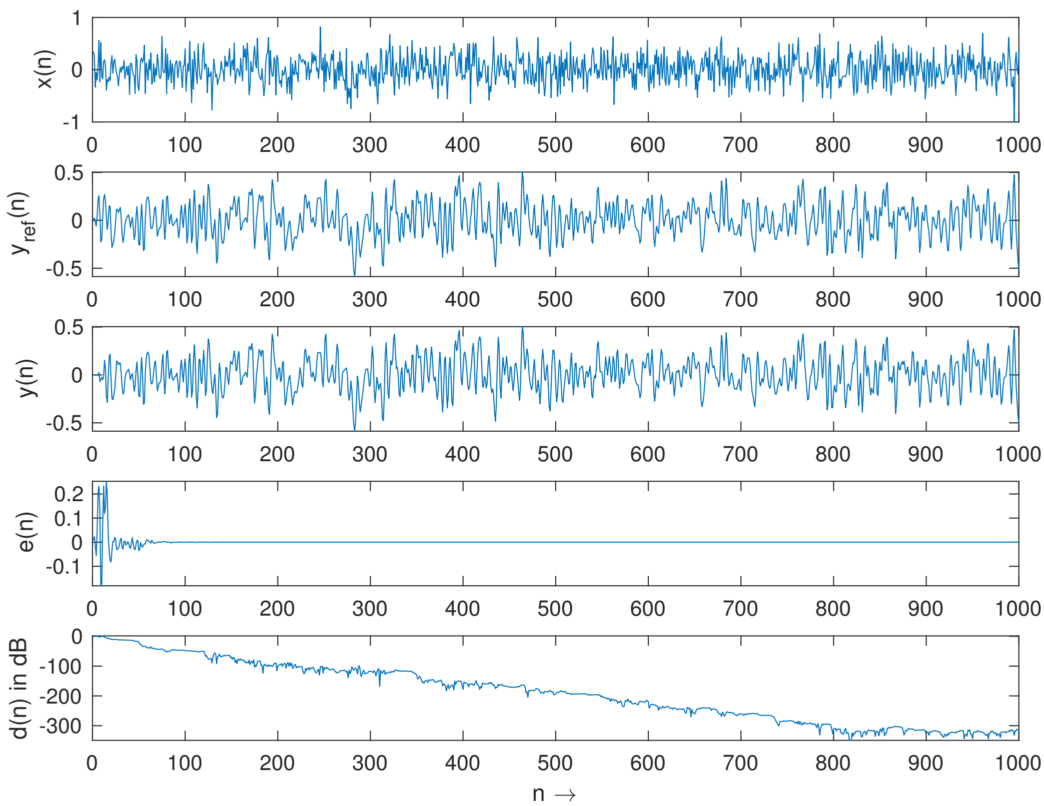
x(n)
x_p(n)
e(n)
n →
×10^4

2. System identification using LMS algorithm




```
h_ref = remez(11, [0 .4 .6 1], [1 1 0 0]);
N = length(h_ref);
figure(1)
clf
subplot(211)
stem(0:N-1, h_ref)
title('Reference impulse response')
ylabel('h_{ref}(n)')
subplot(212)
zplane(h_ref)

Nx=1000;
mu=1.4;

X = randn(1, Nx); % noise
X = X/max(abs(X));
Y_ref = filter(h_ref, 1, X);
x_buff=zeros(1,N); % filter input
h=zeros(1,N);  % direct FIR coeffs
E=zeros(1,Nx); % error
Y=zeros(1,Nx); % output
D=zeros(1,Nx); % coeff difference
for n=1:Nx
%shift filter input states:
x_buff = [X(n), x_buff(1:N-1)]; %newest input sample first
Y(n) = x_buff*h'; % filter result
E(n) = Y_ref(n) - Y(n); % prediction error
for i=1:N % Update of coefficients with LMS
h(i) = h(i) + mu*x_buff(i)*E(n);
end
D(n) = 10*log10( sum(h - h_ref).^2 );
end

subplot(511),plot(X);ylabel('x(n)');
subplot(512),plot(Y_ref);ylabel('y_{ref}(n)');
subplot(513),plot(Y);ylabel('y(n)');
subplot(514),plot(E);ylabel('e(n)');
subplot(515),plot(D);ylabel('d(n) in dB');
```


Reference impulse response
$h_{ref}(n)$
Imaginary Part
Real Part
11


x(n)
$y_{ref}(n)$
y(n)
e(n)
d(n) in dB
n →

5 Matlab Scripts

Table 1: Matlab scripts

1	dasp_ex1_1.m
2	dasp_ex1_2.m
3	dasp_ex1_3_1.m dasp_ex1_3_2.m dasp_ex1_3_3.m dasp_ex1_3_4.m
4	dasp_ex1_4_1.m dasp_ex1_4_2.m