DSP I - Spring 2021, Solution for Assignment 8

1.13 Consider the following 8-point signals, $0 \le n \le 7$.

(a) [1, 1, 1, 0, 0, 0, 1, 1]

(b) [1, 1, 0, 0, 0, 0, −1, −1]

(c) [0, 1, 1, 0, 0, 0, −1, −1]

(d) [0, 1, 1, 0, 0, 0, 1, 1]

Which of these signals have a real-valued 8-point DFT? Which of these signals have a imaginary-valued 8-point DFT? Do not use MATLAB or any computer to solve this problem and do not explicitly compute the DFT; instead use the properties of the DFT.

Solution

Signals (a) and (d) both have purely real-valued DFT. Signal (c) has a purly imaginary-valued DFT.

1.23 Find the DFT of the N-point discrete-time signal,

$$x(n) = \cos\Big(\frac{2\pi}{N}n + \theta\Big), \qquad n = 0, 1, \ldots, N-1.$$

Solution

$$x(n) = \cos\left(\frac{2\pi}{N}n + \theta\right)$$

$$= \frac{1}{2} e^{j\left(\frac{2\pi}{N}n\right)} e^{j\theta} + \frac{1}{2} e^{-j\left(\frac{2\pi}{N}n\right)} e^{-j\theta}$$

$$= \frac{1}{N}\left[\underbrace{\left(\frac{N}{2}e^{j\theta}\right)}_{X(1)} e^{+j\frac{2\pi}{N}n} + \underbrace{\left(\frac{N}{2}e^{-j\theta}\right)}_{X(N-1)} e^{-j\frac{2\pi}{N}n}\right]$$

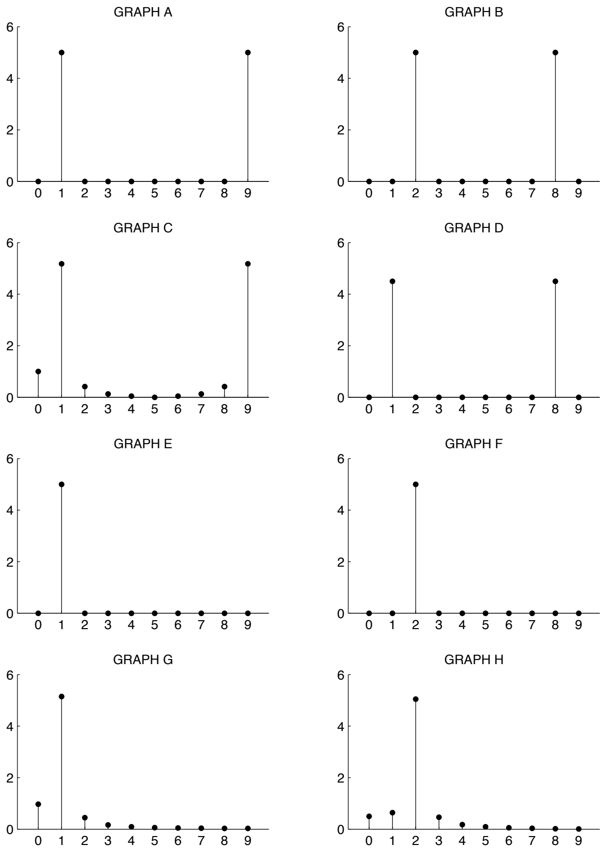
$$X(k) = \begin{cases} \frac{N}{2}e^{j\theta} & k = 1 \\ -\frac{N}{2}e^{-j\theta} & k = N-1 \\ 0 & \text{other } k \end{cases}$$

1.30 The following MATLAB commands define two ten-point signals and the DFT of each.

```
x1 = cos([0:9]/9*2*pi);
x2 = cos([0:9]/10*2*pi);
X1 = fft(x1);
X2 = fft(x2);
```

(a) Roughly sketch each of the two signals, highlighting the distinction between them.

(b) Which of the following four graphs illustrates the DFT $|X_1(k)|$? Explain your answer. Which graph illustrates the DFT $|X_2(k)|$?

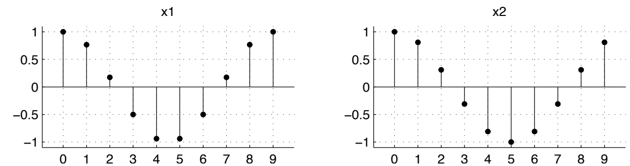


Give your answer by completing the table and provide an explanation for your answer. Note that there are more graphs shown than needed. Your answer will use only two of the eight graphs. Provide a brief explanation for your answer. You should be able to do this exercise without using MATLAB (this was an exam question).

DFT	Graph
X1	
X2	

Solution

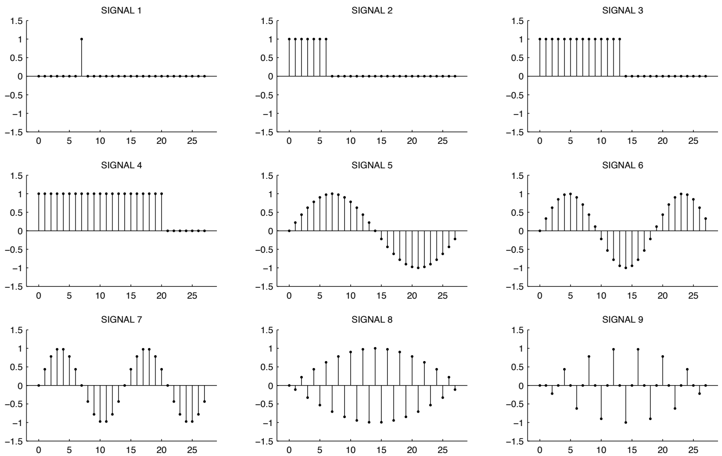
Note that $x_1(9) = \cos(9/9 \cdot 2 \cdot \pi) = 1$; this signal is slightly *more* than a full cycle of a cosine signal (it it were extended into a periodic signal by repeating the signal, then the '1' would appear twice in a row.). On the other hand, the signal $x_2(n)$ is exactly one cycle of a cosine signal.

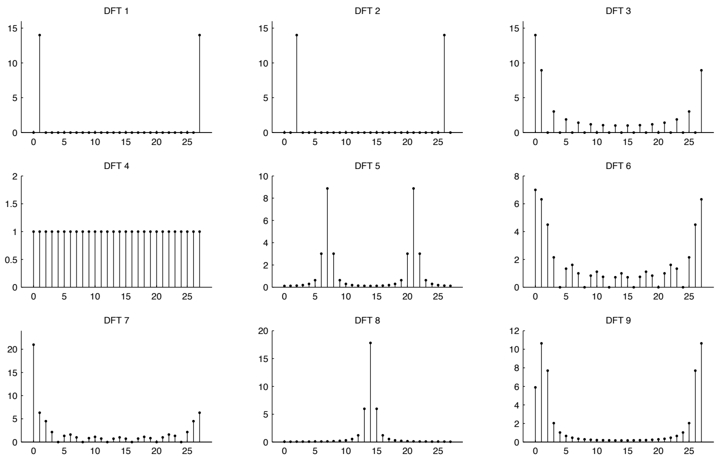


Therefore, signal $x_2(n)$ has the DFT illustrated in graph A. The DFT of signal $x_1(n)$ is similar but leakage is visible – which is visible in graph C.

DFT	Graph
X1	C
X2	A

1.39 **DFT Matching.** Match each of the following 28-point discrete-time signals with its DFT by completing a table. (The DFT plots show the magnitude of the complex DFT coefficients.)





Solution

Signal	DFT
1	4
2	6
3	3
4	7
5	1
6	9
7	2
8	8
9	5

1.43 What is the result of the following MATLAB command? Explain your answer.

```
>> fft([0 1 1 1 1 1])
```

Do not use direct computation of the DFT. Show your use of the appropriate DFT properties.

Solution

Note that the DFT of an N-point constant signal ($x(n) = 1$) is an impulse ($X(k) = N\,\delta(k)$ for $0 \le k \le N-1$) and that the DFT of an N-point impulse signal ($x(n) = \delta(n)$) is a constant ($X(k) = 1$). The signal given in this problem is a constant signal minus and impulse. So the DFT is a impulse signal minus a constant, $X(k) = N\,\delta(k) - 1$

```
[0 1 1 1 1 1] = [1 1 1 1 1 1] - [1 0 0 0 0 0]

DFT([0 1 1 1 1 1]) =  DFT([1 1 1 1 1 1]) - DFT([1 0 0 0 0 0])
                   = [6 0 0 0 0 0] - [1 1 1 1 1 1]
                   = [5 -1 -1 -1 -1 -1]
```

So the answer is

```
[5 -1 -1 -1 -1 -1]
```

1.49 What is the result of the following MATLAB commands? Explain your answer.

```
>> x = [6 5 4 3 2 1];
>> X = fft(x);
>> G = X .* (-1).^[0:5];
>> g = ifft(G)
```

Do not use direct computation of the DFT nor MATLAB. Show your use of the appropriate DFT properties.

Solution

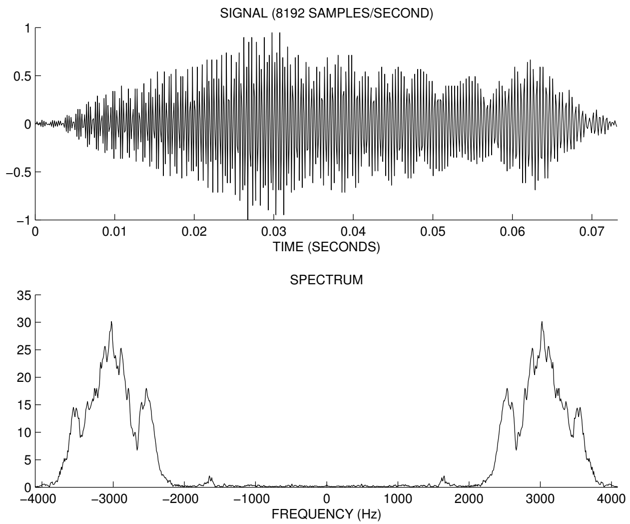
```
ans = [3 2 1 6 5 4]
```

This is multiplication of the DFT coefficients by $e^{j\pi n}$, equivalently by $e^{j\frac{2\pi}{N}(N/2)n}$, so it is a circular shift by $N/2$ in the time-domain.

1.61 An analog signal is sampled at 8192 Hz and 600 samples are collected. These 600 samples are available on the course webpage as the file `signal1.txt`. Plot the signal versus time in seconds; and using the DFT, plot the spectrum of the signal versus *physical frequency* in hertz with the DC component in the center of the plot. If your computer has sound capability, use the `soundsc` command to listen to the signal.

Solution

For this data set, zero padding is not essential. The spectrum below was calculated by zero padding the data set to length $1024 = 2^{10}$.



These figures were produced with the following Matlab code.

```
clear, close all

x = load('data1.txt');
N = 600;
n = 0:N-1;
Fs = 8192;

figure(1), clf
subplot(2,1,1)
plot(n/Fs,x)
xlabel('TIME (SECONDS)')
title('SIGNAL (8192 SAMPLES/SECOND)')
xlim([0 N/Fs])
box off

M = 2^10;
m = 0:M-1;
X = fft(x,M);
X = fftshift(X);
f = Fs*m/M-Fs/2;

subplot(2,1,2)
plot(f,abs(X));
xlabel('FREQUENCY (Hz)')
title('SPECTRUM')
xlim(f([1 end]))
box off

myprint('spectrum_bird')
```

1.64 (Porat 4.23) A bandlimited analog signal is sampled (with no aliasing) at 500 Hz and 980 samples are collected. The DFT of these 980 samples is computed. We wish to compute the value of the spectrum of the sampled signal at 120 Hz.

(a) Which DFT index k is nearest to 120 Hz, and what is its physical frequency in hertz?

(b) What is the minimum number of zeros we must pad onto the 980 samples to obtain a DFT value at 120 Hz exactly? What is the DFT index k then corresponding to 120 Hz?

Solution

(a) $F_s = 500$, $N = 980$, $f_k = 120\text{Hz}$.

$$f_k = \frac{F_s}{N}\,k$$

$$k = \frac{N}{F_s}\,f_k = \frac{980}{500}\,120 = 235.2$$

The DFT index closest to 120 Hz is $\boxed{k = 235.}$ Its physical frequency is $\boxed{119.898 \text{ Hz.}}$ $[(F_s/N)\,k = (500/980)\,235 \approx 119.898\text{Hz.}]$

(b) If we pad zeros so that the total length is M $(M \geq N)$, then $f_k = (F_s/M)\,k$. We want to find M and k such that

$$120 = \frac{500}{M}\,k$$

where both M and k are integers.

$$M = \frac{500}{120}\,k = \frac{25}{6}\,k.$$

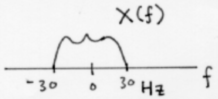
M is an integer only if k is a multiple of 6; in that case M will be a multiple of 25. The smallest multiple of 25 larger than N is 1000. We need to $\boxed{\text{pad 20 zeros.}}$ The DFT index then corresponding to 120 Hz is $\boxed{k = 240.}$ $[k = (M/F_s)\,f_k = (1000/500)\,120 = 240.]$

1.66 The analog signal $x(t)$ is bandlimited to 30 Hz.

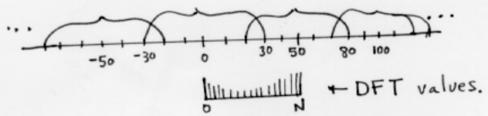
$$X(f) = 0 \quad \text{for} \quad |f| > 30 \text{ Hz}.$$

The signal $x(t)$ is sampled with a sampling rate of 50 Hz and 1000 samples are collected. You then take the DFT of these 1000 samples. Which DFT coefficients $X^d(k)$ are free of aliasing?

Solution



after sampling : $X_s(f)$



$X^d(k) = \text{DFT}\{x[n]\}$ represents $X_s(\frac{k}{N} 50)$

terms affected by aliasing : $20 \le \frac{k}{N} 50 \le 30$

OR $\frac{2}{5} \le K \le \frac{3}{5} N$

Should be 2N/5

here $N = 1000$

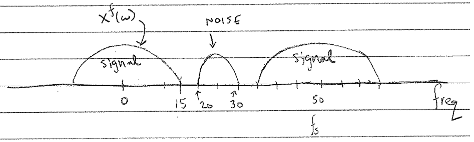
so $\frac{2}{5} 1000 \le k \le \frac{3}{5} 1000$

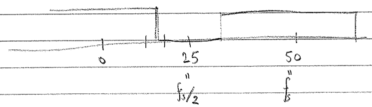
$400 \le k \le 600$

so $\boxed{X^d(k), \quad 0 \le k \le 399, \quad 601 \le k \le 999 \quad \text{are free of aliasing.}}$

3.28 **Filter specifications.** An analog signal, bandlimited to 15 Hz is corrupted by high-frequency noise. The spectrum of the noise is from 20 Hz to 30 Hz. The noisy analog signal is sampled at 50 Hz. A digital lowpass filter is to be designed so as to remove the noise from the signal. What would you choose for the desired frequency response $D(\omega)$ and weighting function $W(\omega)$? Sketch those functions for $0 \leq \omega \leq \pi$ where ω is normalized frequency (radians/sample).

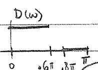
Solution



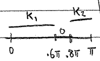


So LPF with $\omega_p = \frac{15}{50}\cdot 2\pi = 0.6\pi$

$\omega_s = \frac{20}{50}\cdot 2\pi = 0.8\pi$



$$\boxed{D(\omega) = \begin{cases} 1 & \text{for } |\omega| \leq 0.6\pi \\ 0 & \text{for } 0.8\pi \leq \omega \leq \pi \end{cases}}$$



$$\boxed{W(\omega) = \begin{cases} 1 & \text{for } |\omega| \leq 0.6\pi \\ 0 & \text{for } 0.6\pi \leq |\omega| \leq 0.8\pi \\ 1 & \text{for } 0.8\pi \leq |\omega| \leq \pi \end{cases}}$$

4 Linear-Phase FIR Digital Filters

4.1 (Porat 9.4) The impulse response of a causal linear-phase FIR filter starts with the values

$$h(0) = 1, \quad h(1) = 3, \quad h(2) = -2.$$

Find the shortest FIR impulse response $h(n)$ for each of the four types. ($h(3) =?$, $h(4) =?$, etc.)

Solution

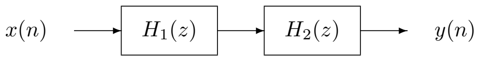
TYPE I: $h(n) = [1,\ 3,\ -2,\ 3,\ 1]$

TYPE II: $h(n) = [1,\ 3,\ -2,\ -2,\ 3,\ 1]$

TYPE III: $h(n) = [1,\ 3,\ -2,\ 0,\ 2,\ -3,\ -1]$

TYPE IV: $h(n) = [1,\ 3,\ -2,\ 2,\ -3,\ -1]$

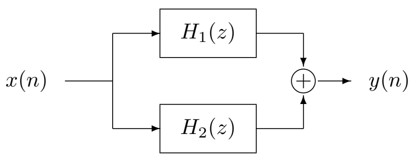
4.2 (Based on Porat 9.7) When two linear-phase FIR filters $h_1(n)$ and $h_2(n)$ are connected in cascade as shown the total filter linear-phase.



What is the filter type of the total filter as a function of the types of $h_1(n)$ and $h_2(n)$? (Fill in the table.)

TYPE (I, II, III, or IV)		
$h_1(n)$	$h_2(n)$	$h_1(n) * h_2(n)$
I	I	?
I	II	?
I	III	?
I	IV	?
II	II	?
II	III	?
II	IV	?
III	III	?
III	IV	?
IV	IV	?

If two linear-phase FIR filters $h_1(n)$ and $h_2(n)$ are connected in parallel as shown, is the total filter linear-phase?



Solution

The cascade of two linear-phase filters *is* linear-phase.

TYPE (I, II, III, or IV)		
$h_1(n)$	$h_2(n)$	$h_1(n) * h_2(n)$
I	I	I
I	II	II
I	III	III
I	IV	IV
II	II	I
II	III	IV
II	IV	III
III	III	I
III	IV	II
IV	IV	I

To make and remember this chart, it is helpful to keep in mind the simplest filter of each type.

TYPE I: $h(n) = [1]$

TYPE II: $h(n) = [1, 1]$

TYPE III: $h(n) = [1, 0, -1]$

TYPE IV: $h(n) = [1, -1]$

The parallel connection of two linear-phase filters is linear-phase only when they are the same length and have the same type of symmetry (when they are the same type).

4.3 (Porat 9.5) Let $H(z)$ be the transfer function of a linear-phase FIR filter with real coefficients. The filter is known to have zeros in the following locations:

$$z = \left\{1, \quad 0.5\,\mathrm{e}^{\mathrm{j}\pi/3}, \quad -5, \quad \mathrm{j}\right\}$$

(a) Because $H(z)$ is a linear-phase FIR filter it must have other zeros as well, because the zeros of a linear-phase FIR must exist in particular configurations. What are the other zeros of $H(z)$? Sketch the zero diagram of this filter.

(b) What is the minimal length of the filter?

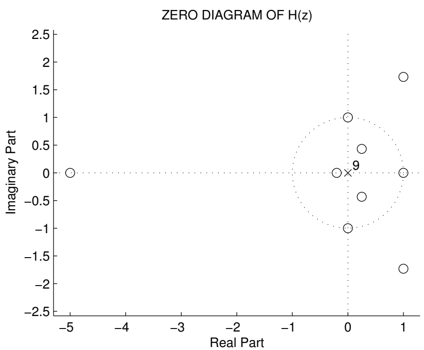
(c) What is the filter type of the minimal length filter? (I,II,III, or IV)

Solution

The zero $z = 1$ can be a zero by itself. The zero $z = 0.5\,\mathrm{e}^{\mathrm{j}\pi/3}$ must correspond to three other zeros: $0.5\,\mathrm{e}^{-\mathrm{j}\pi/3}$ (the zeros must occur in complex-conjugate pairs because the impulse response is real), $2\,\mathrm{e}^{\mathrm{j}\pi/3}$ and $2\,\mathrm{e}^{-\mathrm{j}\pi/3}$ (because if z_o is a zero of a linear-phase filter, then so is $1/\overline{z_o}$). The zero $z = -5$ must likewise correspond to one other zero: $z = -1/5$. The zero $z = \mathrm{j}$ must likewise correspond to one other zero: $z = -\mathrm{j}$. The total number of zeros is 9, so the length of the impulse response is 10. The transfer function is

$$H(z) = \prod_{k=1}^{9} (1 - z_k/z).$$

This is a Type IV filter because it has a zero at $z = 1$, no zero at $z = -1$, and is of even length. The zero diagram is shown in the following figure.



4.7 (a) List all of the four types of linear-phase FIR filters can be used for the implementation of a *bandpass* filter.

(b) List all of the four types of linear-phase FIR filters can be used for the implementation of a *bandstop* filter.

Solution

(a) $\boxed{\text{Any}}$ of the four linear-phase FIR filter types are fine for the implementation of a bandpass filter.

(b) Only a $\boxed{\text{Type-I}}$ linear-phase FIR filter is appropriate for the implementation of a bandstop filter, because the other three types must all have nulls at either DC, the Nyquist frequency, or both.

4.8 **The real-valued amplitude response.** The frequency response of a linear-phase FIR filter can be written as

$$H(\omega) = A(\omega)\, e^{j\theta(\omega)}$$

where amplitude response $A(\omega)$ is continuous and real-valued. Write a Matlab function that uses the FFT to compute $A(\omega)$ from the impulse response $h(n)$ at the frequency points

$$\omega_k = \frac{2\pi}{L} k \qquad 0 \le k \le L-1.$$

The input to your Matlab function should be a vector `h` containing the values of the impulse response, the filter Type (1,2,3,4), and the number of samples L. You may assume that L is greater than the length of the filter. The Matlab function should provide vectors `A` and `w` such that `plot(w,A)` gives a plot of $A(\omega)$.

An example Matlab header:

```
function [A,w] = firamp(h,type,L)
% [A,w] = firamp(h,type,L)
% Amplitude response of a linear-phase FIR filter
% A : amplitude response at the frequencies w
% w : [0:L-1]*(2*pi/L);
% h : impulse response
% type = [1,2,3,4]
```

Verify your Matlab function works correctly for each of the 4 filter types by using it to plot $A(\omega)$ for the following filters.

- Test for Type I using the following impulse response.

```
N = 29;
n = 0:N-1;
wo = 0.34*pi;
h1 = (wo/pi)*sinc((wo/pi)*(n-(N-1)/2));
```

- Test for Type II using the truncated sinc function again.

```
N = 28;
n = 0:N-1;
wo = 0.34*pi;
h2 = (wo/pi)*sinc((wo/pi)*(n-(N-1)/2));
```

- Test for Type IV using a differentiator with length 16

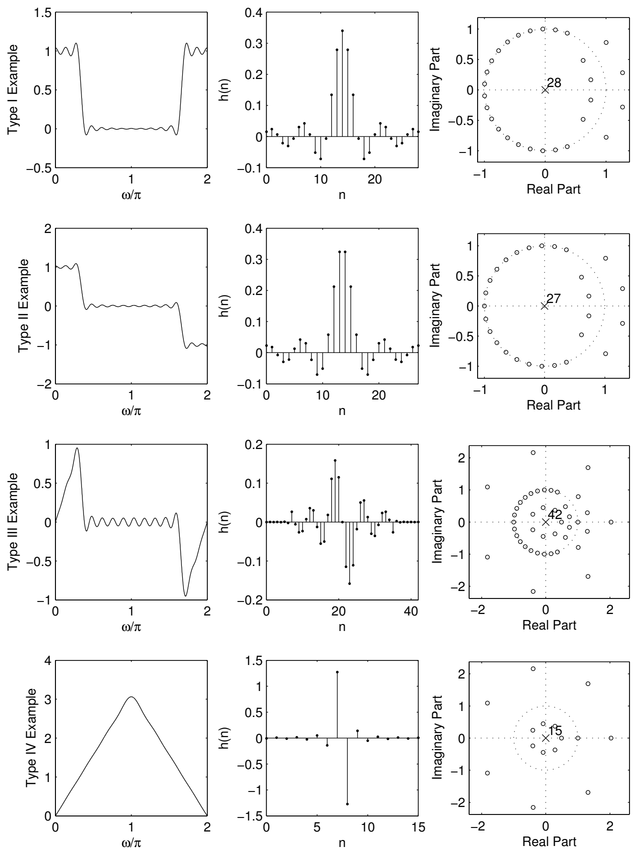
```
N = 16;
n = 0:N-1;
h4 = (-1).^(n-(N-2)/2)./(pi*(n-(N-1)/2).^2);
```

- Test for Type III using a *lowpass* differentiator — obtained by convolving the impulse response of a Type II lowpass filter and Type IV differentiator.

```
h3 = conv(h2,h4);
```

Solution

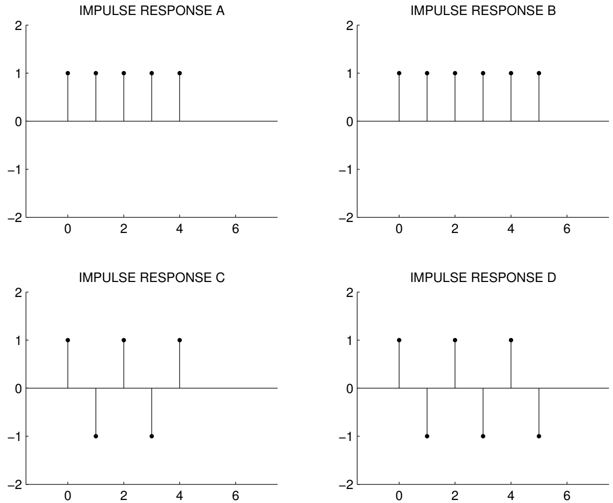
```
function [A,w] = firamp(h,type,L)
% [A,w] = firamp(h,type,L)
% Amplitude response of a linear-phase FIR filters
% A : amplitude response
% w : frequency grid [0:L-1]*(2*pi/L)
% h : impulse response
% type : [1,2,3,4]
% L : frequency density (optional, default = 2^10)

h = h(:)';                % make h a row vector
N = length(h);            % length of h
if nargin < 3
   L = 2^10;              % grid size
end
H = fft(h,L);             % zero pad and fft
w = [0:L-1]*(2*pi/L);     % frequency grid
M = (N-1)/2;
if (type == 1)|(type == 2)
  H = exp(M*j*w).*H;      % Type I and II
else
  H = -j*exp(M*j*w).*H;   % Type III and IV
end
A = real(H);              % discard zero imaginary part
```

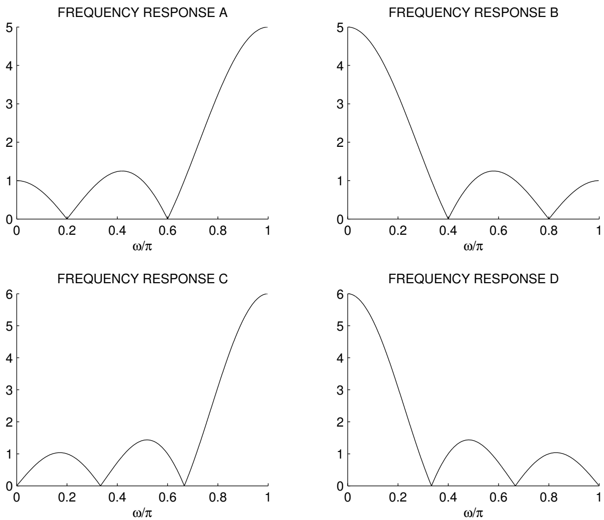


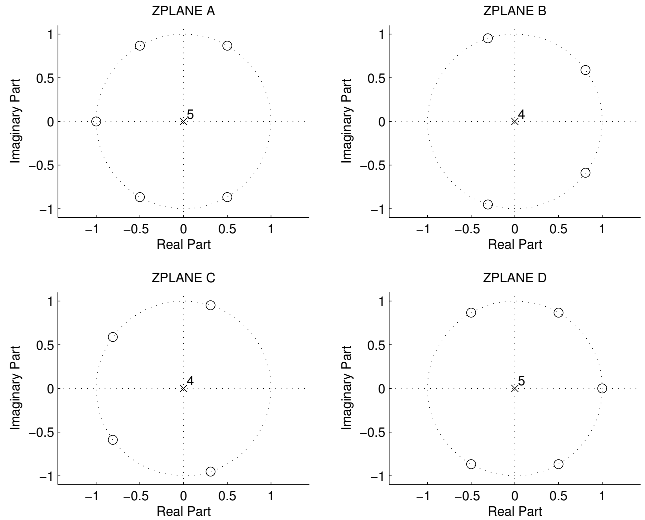
The amplitude response, impulse response, and zero diagram were plotted with the following Matlab commands.

```
% Type I
N = 29;
n = 0:N-1;
wo = 0.34*pi;
h1 = (wo/pi)*sinc((wo/pi)*(n-(N-1)/2));
[A1,w] = firamp(h1,1);

figure(1), clf
subplot(4,3,1),
plot(w/pi,A1),
ylabel('Type I Example'),
xlabel('\omega/\pi')

subplot(4,3,2),
stem(0:N-1,h1,'.'),
ylabel('h(n)')
xlabel('n')

subplot(4,3,3),
zp = zplane(h1);
set(zp,'markersize',3)
```

4.11 **Matching.** Match each impulse response with its frequency response and zero diagram by filling out the following table. You should do this problem with out using a computer.







Impulse Response	Zero Diagram	Frequency Response
A		
B		
C		
D		

Solution

Impulse Response	Zero Diagram	Frequency Response
A	C	B
B	A	D
C	B	A
D	D	C

4.15 **Design by DFT-based interpolation.** Implement the DFT-based interpolation approach for Type II. (Modify the code in the notes for the Type I case.) In particular, consider the design of a length $N = 12$ low-pass Type II FIR filter where the amplitude function $A(\omega)$ interpolates 1 and 0 as follows.

$$A\left(\frac{2\pi}{N}k\right) = \begin{cases} 1 & k = 0, 1, 2 \\ 0 & k = 3, 4, 5, 6 \end{cases}$$

What values should $A(2\pi k/N)$ interpolate for $7 \le k \le N - 1$? (Look at the characteristics of the $A(\omega)$ function in the notes.) Find the Type II filter $h(n)$ that interpolates those values. Make a plot of $A(\omega)$, $h(n)$, and the zeros of $H(z)$ using **zplane**. Check that $A(\omega)$ interpolates the specified points.

Note: the impulse response $h(n)$ should be *real*. If you get complex values for $h(n)$ then you have made a mistake.

Also, use this approach to design a Type II filter of length $N = 32$.

Optional: Implement the DFT-based interpolation approach for the design of Type III and IV FIR filters and design an example of each.

Solution

From the notes, the amplitude response of a TYPE II filter is antisymmetric around $\omega = \pi$. Therefore, the other interpolation points must be given by

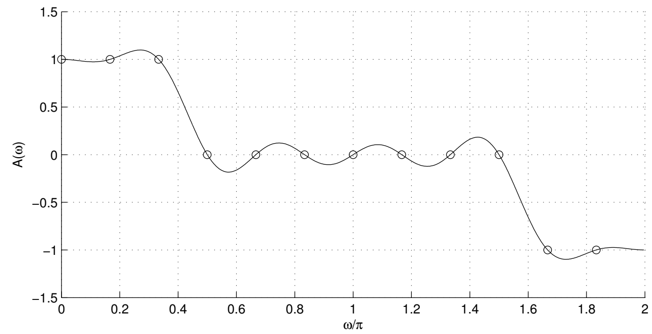
$$A\left(\frac{2\pi}{N}k\right) = \begin{cases} 0 & k = 7, 8, 9 \\ -1 & k = 10, 11 \end{cases}$$

or

$$A\left(\frac{2\pi}{N}k\right) = [1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ -1,\ -1]$$

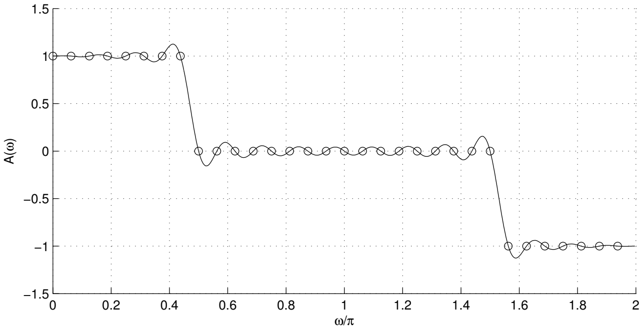
for $0 \le k \le N - 1$.

```
filename = 'fir_interp';

N = 12;
M = (N-1)/2;
Ak = [1 1 1 0 0 0 0 0 0 0 -1 -1];
k = 0:N-1;
h = ifft(Ak.*exp(-j*2*pi*M*k/N))
h = real(h);

L = 512;
[A,w] = firamp(h,2,L);
close all
figure(1), clf
subplot(3,1,[1 2])
plot(w/pi,A,2*[0:N-1]/N,Ak,'o')
xlabel('\omega/\pi')
ylabel('A(\omega)')
grid, box off
myprint(filename)
```



A length 32 Type II FIR filter is obtained similarly. Notice the large errors between the interpolation points, especially near the band-edge.

```
close all
clear

filename = 'fir_interp_2';

N = 32;
M = (N-1)/2;
Ak = [ones(1,8) zeros(1,17) -ones(1,7)];
k = 0:N-1;
h = ifft(Ak.*exp(-j*2*pi*M*k/N));
h = real(h);

L = 512;
[A,w] = firamp(h,2,L);
figure(1), clf
subplot(3,1,[1 2])
plot(w/pi,A,2*[0:N-1]/N,Ak,'o')
xlabel('\omega/\pi')
ylabel('A(\omega)')
grid
box off
myprint(filename)
```


1.5
1
0.5
0
−0.5
−1
−1.5
A(ω)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
ω/π