ECE113: DSP

Homework 7 Solutions

Problem 1:

Prove the following properties of linear-phase FIR filters.

1. If $H(z)$ has four zeros at $z_1 = re^{j\theta}$, $z_2 = \frac{1}{r}e^{-j\theta}$, $z_3 = re^{-j\theta}$, and $z_4 = \frac{1}{r}e^{-j\theta}$ then $H(z)$ represents a linear-phase FIR filter.
2. If $H(z)$ has two zeros at $z_1 = e^{j\theta}$ and $z_2 = e^{-j\theta}$ then $H(z)$ represents a linear-phase FIR filter.
3. If $H(z)$ has two zeros at $z_1 = r$ and $z_2 = \frac{1}{r}$ then $H(z)$ represents a linear-phase FIR filter.
4. If $H(z)$ has a zero at $z_1 = 1$ or a zero at $z_1 = -1$ then $H(z)$ represents a linear-phase FIR filter.

Solution:

1. The filter $H(z)$ has the following four zeros

$$z_1 = re^{j\theta}, \quad z_2 = \frac{1}{r}e^{j\theta}, \quad z_3 = re^{-j\theta}, \quad z_4 = \frac{1}{r}e^{-j\theta}$$

The system function can be written as

$$\begin{aligned}
H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right)\left(1 - z_3 z^{-1}\right)\left(1 - z_4 z^{-1}\right) \\
&= \left(1 - re^{j\theta}z^{-1}\right)\left(1 - \frac{1}{r}e^{j\theta}z^{-1}\right)\left(1 - re^{-j\theta}z^{-1}\right)\left(1 - \frac{1}{r}e^{-j\theta}z^{-1}\right) \\
&= \left\{1 - (2r\cos\theta)\, z^{-1} + r^2 z^{-2}\right\}\left\{1 - \left(2r^{-1}\cos\theta\right) z^{-1} + r^{-2}z^{-2}\right\} \\
&= 1 - 2\cos\theta\left(r + r^{-1}\right) z^{-1} + \left(r^2 + r^{-2} + 4\cos^2\theta\right) z^{-2} - 2\cos\theta\left(r + r^{-1}\right) z^{-3} + z^{-4}
\end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -2\cos\theta\left(r + r^{-1}\right), \left(r^2 + r^{-2} + 4\cos^2\theta\right), -2\cos\theta\left(r + r^{-1}\right), 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

2. The filter $H(z)$ has the following two zeros

$$z_1 = e^{j\theta} \text{ and } \quad z_2 = e^{-j\theta}$$

The system function can be written as

$$\begin{aligned} H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right) = \left(1 - e^{j\theta} z^{-1}\right)\left(1 - e^{-j\theta} z^{-1}\right) \\ &= \left\{1 - (2\cos\theta) z^{-1} + z^{-2}\right\} \end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -2\cos\theta, 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

3. The filter $H(z)$ has the following two zeros

$$z_1 = r \text{ and } z_2 = \frac{1}{r}$$

The system function can be written as

$$\begin{aligned} H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right) = \left(1 - r z^{-1}\right)\left(1 - \frac{1}{r} z^{-1}\right) \\ &= 1 - \left(r + r^{-1}\right) z^{-1} + z^{-2} \end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -\left(r + r^{-1}\right), 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

4. If $H(z)$ has a zero at $z_1 = 1$ or a zero at $z_1 = -1$ then $H(z)$ can be written as

$$H(z) = (1 - z^{-1}) \text{ or } H(z) = (1 + z^{-1})$$

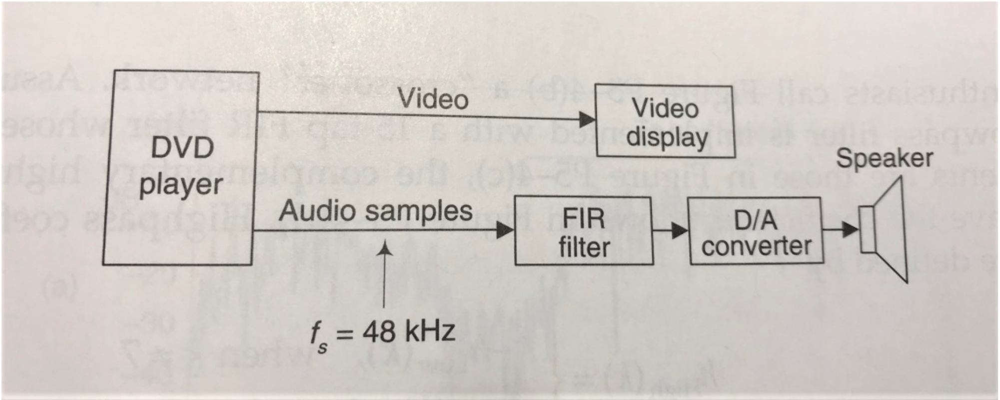
with impulse responses

$$h(n) = \left\{\underset{\uparrow}{1}, 1\right\} \text{ or } h(n) = \left\{\underset{\uparrow}{1}, -1\right\}$$

both of which are finite-duration with symmetric and antisymmetric impulse responses, respectively. This implies that the filter is a linear-phase FIR filter.

Problem 2:

Assume we want to filter the audio signal from a digital video disc (DVD) player as shown in the following diagram:



The filtered audio signal is converted to analog by a D/A and drives a speaker. For the audio signal to have acceptable time synchronization with the video signal, the video engineers have determined that the time delay for the FIR filter should not be greater than $6x10^{-3}$ seconds. If the F_s sampling rate of the audio signal is 48KHz, what is the maximum number of taps that we can have in the FIR filter that would satisfy the time delay restriction? (Assume the FIR filter has linear phase response and the D/A converter delay is zero).

Solution:

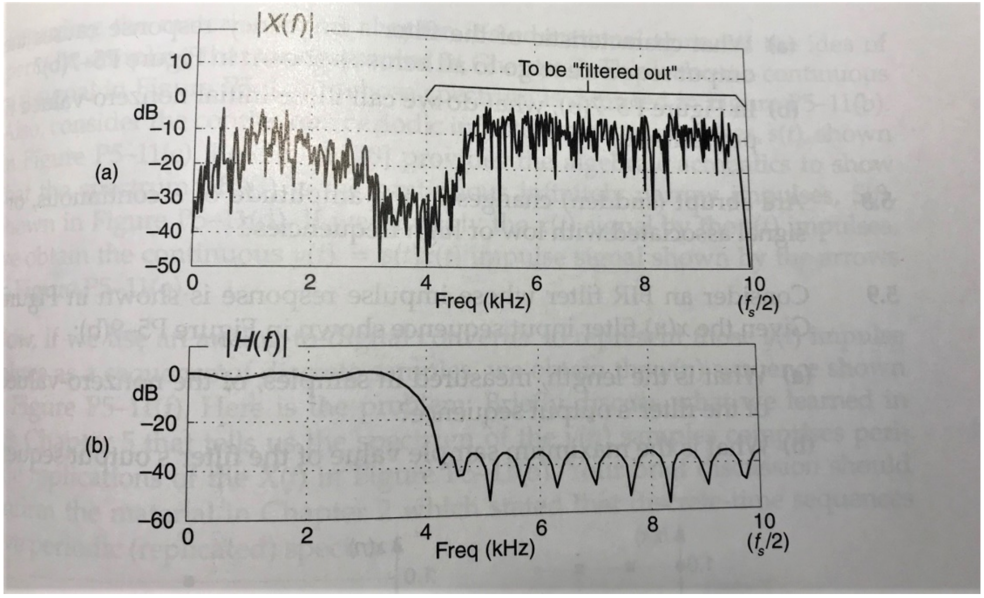
Time delay through a linear phase FIR filter is equal to the filter's group delay. The group delay of a linear-phase FIR filter of order N (i.e., N delay elements or N+1 taps) is N/2 samples, or equivalently (N/2)Ts seconds where Ts is the sampling period. Therefore:

(Max no. of delay elements/2)x(1/Fs) < $6x10^{-3}$ => N_{max} <$6x10^{-3}x2x48x10^{3}$=576

So the max number of taps would be N_{max} = 576

Problem 3:

Let's assume we were given an x(n) discrete-time sequence, whose sample rate is 20KHz, and its |X(f)| spectral magnitude is shown below. We were asked to design a linear-phase low-pass FIR filter to attenuate the undesired high-frequency noise, as seen in the spectral magnitude plot. Assume we already designed the filter and the frequency magnitude response of our filter is also shown below.



Sometime later, we unfortunately learned that our original sequence x(n) had actually been obtained at 40KHz sampling rate and not at 20KHz!

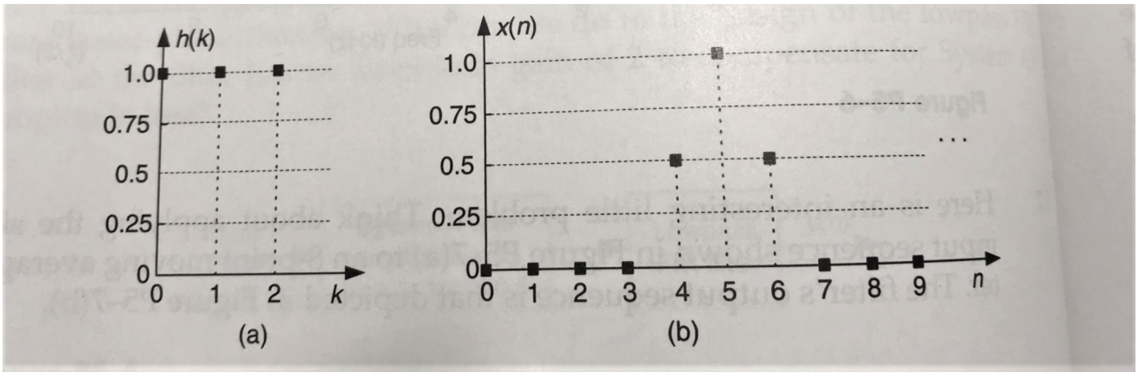
So do we need to do anything with our low-pass filter coefficients h(k)'s which were originally obtained based on the assumption of 20KHz sampling rate so that our filter still attenuates the high-frequency noise when the sample rate is actually 40KHz? If yes, what? If no, why not? Please discuss.

Solution:

The answer is no, we don't need to do anything, and the low-pass filter coefficients h(k) can remain the same. The reason is that when we design the digital filter based on our desired frequency response versus the discrete frequency in rad/sample or cycle/sample, we have effectively normalized our frequency with respect to the sampling frequency, and as such, have not used the sampling frequency explicitly in our filter design. In other words, if the sampling rate changes from 20KHz to 40KHz, the shape of the spectral plot of X(f) and H(f) would remain unchanged with the exception that their frequency axis values in Hz would be increased by a factor of two, and of course, the filter would have to run at double the clock rate.

Problem 4:

Consider an FIR filter whose impulse response h(k) and the input sequence x(n) are shown below. Find the output sequence y(n).



Solution:

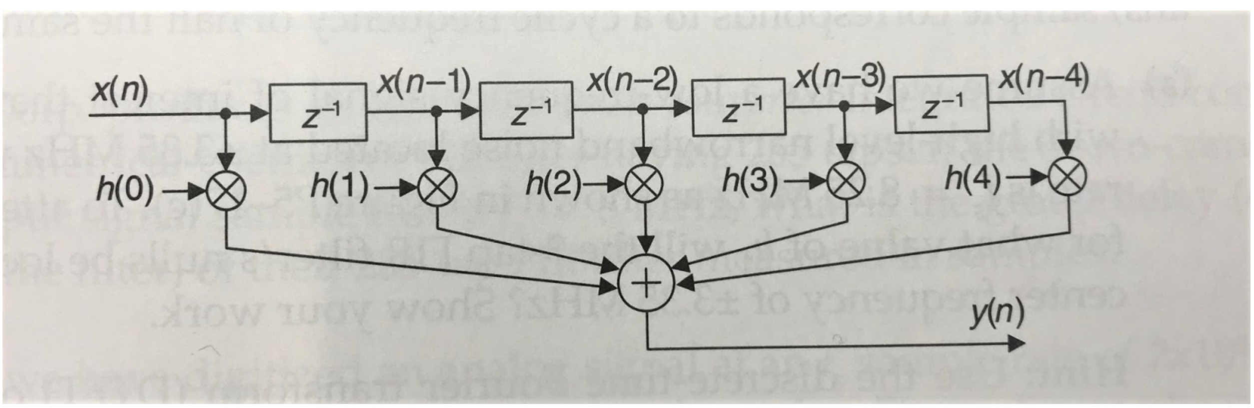
The nonzero output sequence values are obtained simply though time-domain convolution:

Y(n)={0.5,1.5,2.0,1.5,0.5}

As expected the length of the output sequence is N1+N2-1=3+3-1=5 samples. And the maximum value in the output sequence is equal to 2.0

Problem 5:

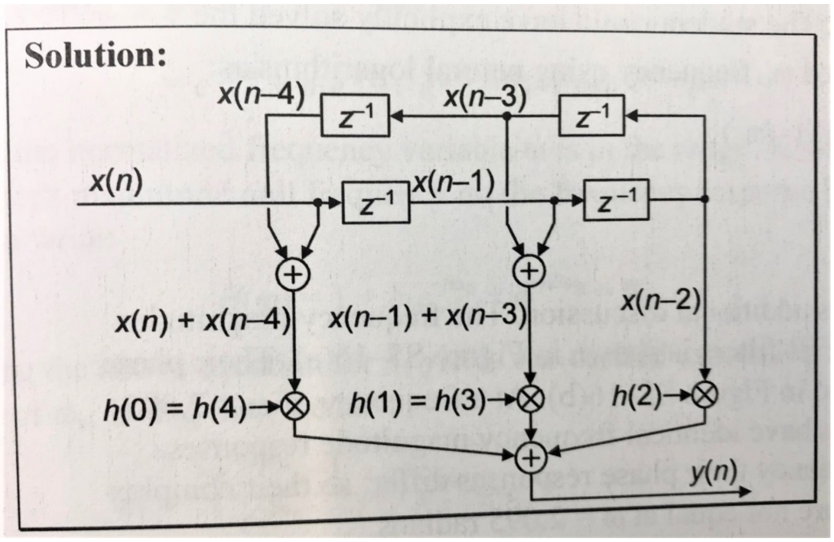
Consider the following linear-phase 5-tap FIR filter:



DSP engineers always seek to reduce the number of multipliers in their systems. Given what you know about linear-phase FIR filters, redesign the filter above with the reduced number of multiplications per output sample and draw its block diagram.

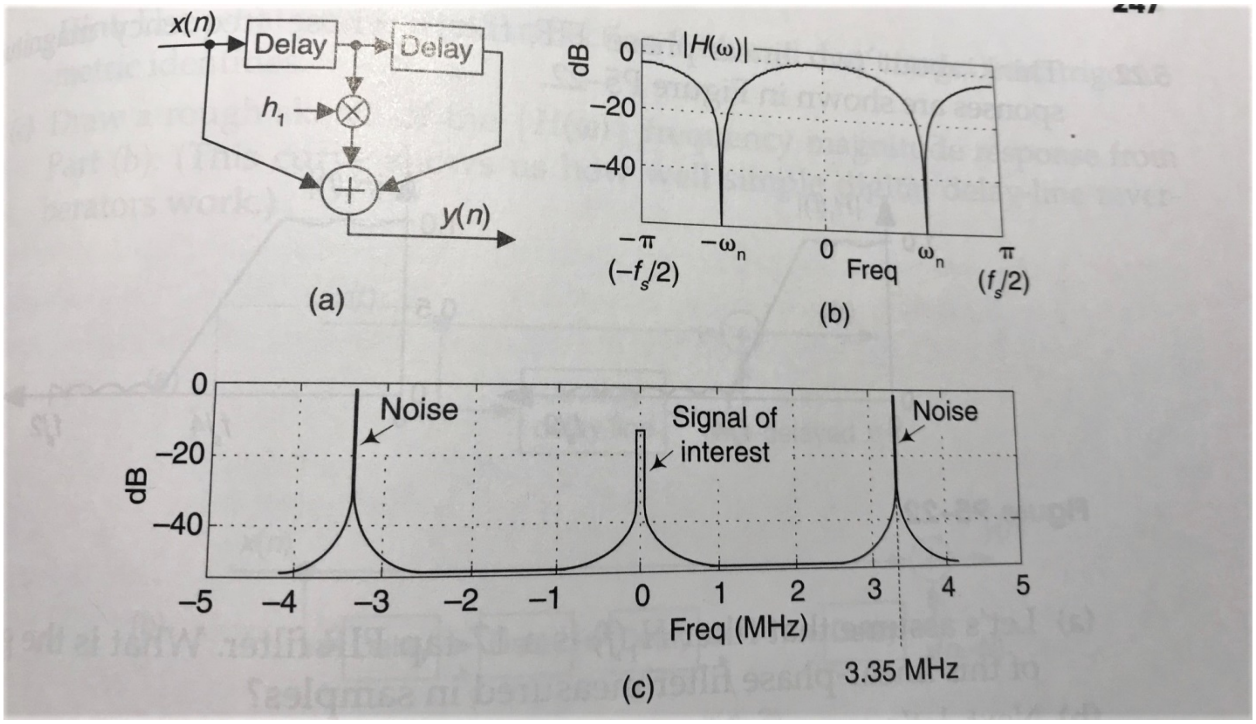
Solution:

Given that the filter is linear phase, the impulse response is either symmetric or antisymmetric, i.e., $h(n)=\pm h(4-n)$. This property can be exploited to reduce the number of multiplications per output sample as shown below:



Problem 6:

This picture below shows a simple 3-tap non-recursive FIR filter (Figure (a)):



The solution to this problem shows us how to design computationally efficient narrowband noise reduction filters.

If $|h1| \leq 2$, the filter will have an $|H(\omega)|$ frequency magnitude response with two nulls at $\pm\omega_n$ as shown in Figure (b).

(a) Assume we have a low-frequency desired signal that is contaminated by high-level narrowband noise at ± 3.35 MHz when the sample rate is *Fs*=8.25 MHz, as shown in Figure (c). Find the value of the filter coefficient h_1 such that the frequency response of the 3-tap FIR filter will have nulls at ± 3.35 MHz.

(b) What is the DC gain (i.e., the gain at zero Hz frequency) of our 3-tap filter?

(c) Does this filter have linear or nonlinear phase response? Why? Please discuss.

Solution:

(a) The filter's center coefficient is:

$$h_1 = 1.6617.$$

The derivation of h_1 is as follows: We write the discrete-time Fourier transform (DTFT) of the filter's impulse response, which is equal to the filter's $h(k) = [1, h_1, 1]$ coefficients, as:

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-jk\omega} = \sum_{k=0}^{2} h(k)e^{-jk\omega}$$

$$= e^{-j0\omega} + h_1 e^{-j\omega} + e^{-j2\omega} = 1 + h_1 e^{-j\omega} + e^{-j2\omega}$$

where the normalized frequency variable ω is in the range $-\pi \le \omega \le \pi$. At the filter's magnitude null frequency ω_n the frequency response is zero, so we can write:

$$H(\omega) = 1 + h_1 e^{-j\omega_n} + e^{-j2\omega_n} = 0.$$

Solving the above equation for h_1 yields our desired expression for h_1 in terms of ω_n. Doing that, we have

$$h_1 = \frac{-e^{-j2\omega_n} - 1}{e^{-j\omega_n}} = -e^{-j\omega_n} - e^{j\omega_n}.$$

Remembering Euler's equation of $\cos(\alpha) = (e^{j\alpha} + e^{-j\alpha})/2$, we can simplify the h_1 expression as:

$$h_1 = -2\cos(\omega_n)$$

where the normalized null frequency ω_n is in the range $0 \le \omega_n \le \pi$.

Because the normalized frequency of $\omega = 2\pi$ radians corresponds to a radian frequency of $2\pi f_s$ radians/second (where f_s is the signal data sample rate in Hz), the h_1 coefficient can also be defined using

$$h_1 = -2\cos[(2\pi f_n)\frac{2\pi}{2\pi f_s}] = -2\cos(2\pi f_n/f_s)$$

where cyclic null frequency f_n is in the range $0 \le f_n \le f_s/2$ Hz. So, given a null frequency of $3.35\text{x}10^6$ Hz, the desired h_1 filter coefficient is

$$h_1 = -2\cos[2\pi(3.35\text{x}10^6/8.25\text{x}10^6)] = -2\cos(2.5514) = 1.6617.$$

b) The DC gain of the filter is the sum of its coefficients, or:

$$\boxed{\text{DC gain} = 1 + 1.6617 + 1 = 3.6617.}$$

Using $h_1 = 1.6617$ yields a filter magnitude response shown by the solid curve in Figure S5–17.

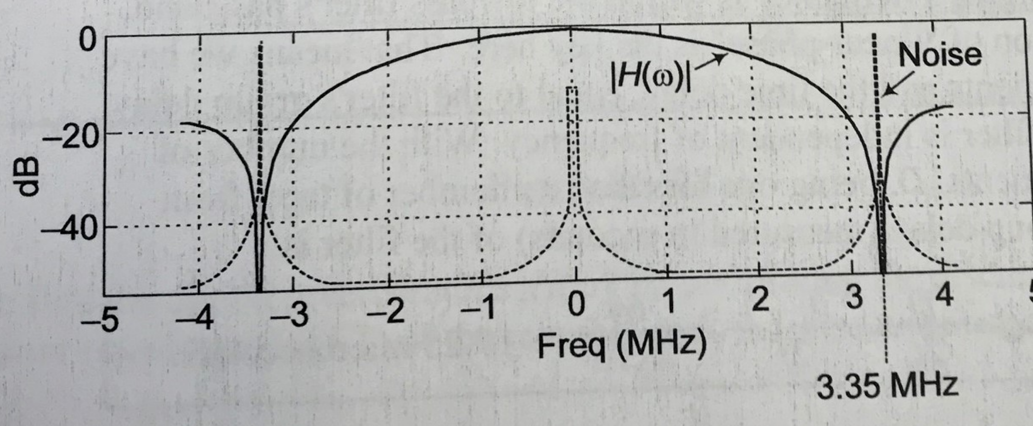
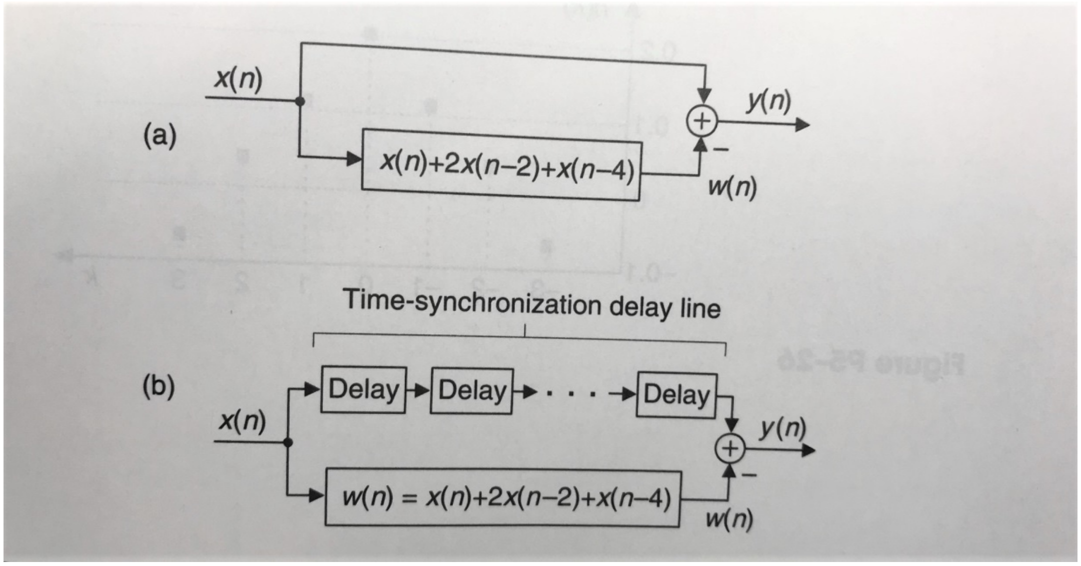


Figure S5–17

$\boxed{\text{(c) The 3-tap FIR filter has linear phase because its coefficients are symmetrical, } h(k) = [1, h_1, 1].}$

Problem 7:

There are digital filtering schemes that use the process conceptually shown in Figure (a) below, where the filter may be comprised of two parallel paths.



The actual implementation, however, will typically need a delay line as shown in Figure (b). The delay-line in the upper path is needed to compensate for the delay of the FIR filter in the bottom path.

How many unit-delay elements do you have to insert in the upper path for its delay to match that of the lower path?

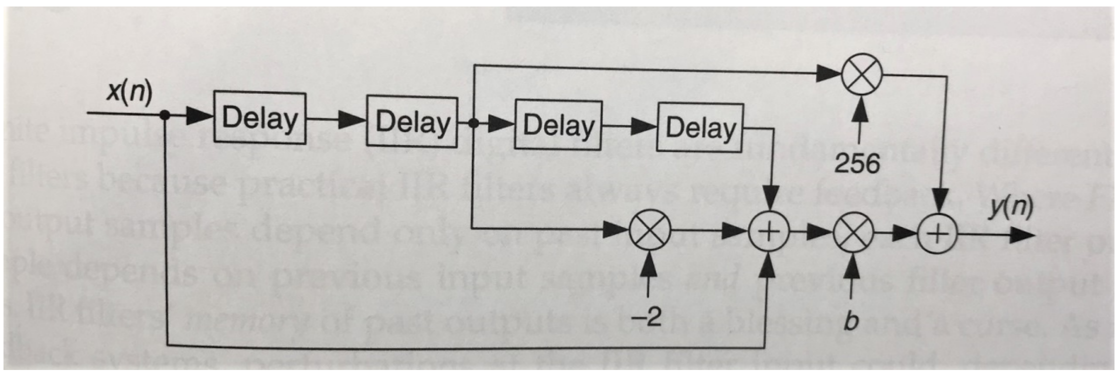
Solution:

The number of delay elements in the upper path should be equal to the group delay of the FIR filter in the bottom path as long as it has linear phase and thus a constant group delay, and its group delay is an integer (side note: fractional group delays can typically be avoided in situations like this, but otherwise, one could use various techniques to implement a fractional delay line. One such simple technique would be to linearly interpolate between every two consecutive samples).

The filter in the bottom path is a 4^{th} order filter (i.e., 5 taps, two of which happen to have their coefficients equal to 0) with a symmetric impulse response. Therefore it does indeed have a linear phase response, and its group delay is simply N/2=4/2=2 samples. Therefore we need two unit delay elements in the upper path as our time-synchronization delay line.

Problem 8:

Texas Instruments Inc. makes a Digital Media System-on-Chip (DMSoC) (Part# TMS320DM646x), which leverages TI's DaVinci™ technology to meet the networked media encode and decode application processing needs of next-generation embedded devices. It has a dual-core architecture providing the benefits of both a digital signal processor (DSP) and a reduced instruction set computer (RISC), incorporating a high-performance TMS320C64x+™ DSP core and an ARM926EJ-S core. Among other things, this chip contains an FIR filter structure shown below, where the coefficient *b*, as defined by the user, controls the frequency magnitude response of the filter.



(a) What is the time-domain LCCDE (Linear Constant Coefficient Difference Equation) for this filter?
(b) Does the filter have a linear-phase frequency response, or nonlinear? Why?
(c) What is the Group Delay of this filter measured in number of samples?

Solution:

(a) The time-domain difference equation for the filter can be found by way of inspection of the filter's block diagram, or by redrawing the block diagram to that shown in Figure S5–28.

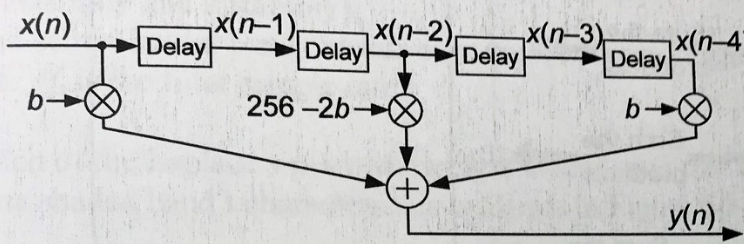


Figure S5–28

The difference equation is:

$$y(n) = bx(n) + (256 - 2b)\cdot x(n-2) + bx(n-4).$$

(b) The filter's five coefficients are:

$h(0) = b,$
$h(1) = 0,$
$h(2) = 256 - 2b,$
$h(3) = 0,$
$h(4) = b.$

Because the filter's coefficients are symmetrical, the filter will have a linear-phase frequency response.

(c) From Figure S5–28, the number of delay elements in the filter's tapped-delay line is $D = 4$. So the group delay, measured in samples, is:

$$\text{group delay} = G = \frac{D}{2} = \frac{4}{2} = 2 \text{ samples.}$$

MATLAB:

- This is a simple exercise for an FIR filter design and filtering of an audio signal in MATLAB. It was kindly provided to me by Mr. Rick Lyons, the author of R4 textbook.
- In this problem, you will be using "fir1" function in MATLAB to design your FIR filter. But, before you start, please do read about all FIR filter design capabilities in MATLAB here: https://www.mathworks.com/help/signal/ug/fir-filter-design.html
- The file 'CapnJ.wav' (posted under Week 8 on CCLE) contains discrete samples of an audio signal of a male voice speaking English words, and a value for the F_s sample rate of that signal.
- Please use the template 'ECE113_Spring18_FIR_Design_Template.m' posted under Week 8 on CCLE.
- After copying that ".wav" file to your Matlab Workspace, you can load and play the audio signal using "sound" function as shown in the template.
- The audio 'signal of interest', covering the frequency range of roughly 0 to 3000 Hz, is very badly corrupted with high-level noise. Your problem is to design a digital filter that will sufficiently attenuate the noise so that the audio speech is intelligible.

The Problem:

Design an FIR filter, filter the audio signal, and determine the English words being spoken.

Hints:

- Make sure you pause your computer using Matlab's *pause* command between any consecutive *sound(xx,Fs)* commands. This will avoid any "Unable to open sound device" error messages.
- As you start making progress in removing the corrupting characteristics of the original sound signal, don't be afraid to increase or decrease the amplitude (audio volume) of your time signal using e.g., *X = 20*X;* or *X = X/20;* commands. This may be necessary depending on your PC's sound card and what version of MATLAB you're using.