Student: Marco Defilippi

ES1:

5.1 We first introduced the notion of *impulse response* in Chapter 1, and here in Chapter 5 we discussed the importance of knowing the impulse response of FIR filter networks. With that said, if the $y(n)$ output of a discrete system is equal to the system's $x(n)$ input sequence:

(a) Draw the unit impulse response of such a system.

(b) Draw the block diagram (structure) of that system.

(c) What is the frequency magnitude response of such a system? Prove your answer.

Answer:

For a system where the output y(n) is equal to the input x(n), the unit impulse response h(n) is simply an impulse itself. This is because an impulse input passes through the system unchanged.

a)

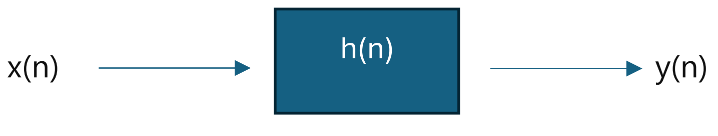
H(n) = delta(n)

Delta(n) is the delta function.

Delta(n) = $\begin{matrix} 1, \ if \ n = 0 \\ 0, \ otherwise \end{matrix}$

b)

The block diagram of such a system where the output is identical to the input is essentially a direct path from input to output without any filtering.



c)

For a system where the output y(n) = x(n), the frequency response is a constant across all frequencies. This is because the input signal is passed through unchanged, meaning the system has a flat frequency response.

|H(f)|=1, for all frequencies.

The DFT of the impulse response h(n) = delta(n) is:

H(n) = $\sum_{0}^{N-1} h(n)e^{-j\omega n} = \sum_{0}^{N-1} delta(n)e^{-j\omega n} = 1$

Since the impulse response h(n) is 1 at n=0 and 0 elsewhere, the sum evaluates to 1 for all frequencies ω. Thus, the magnitude response is 1, indicating a flat response across all frequencies.

ES2:

5.2 Consider a simple analog signal defined by $x(t) = \cos(2\pi 800t)$ shown in Figure P5–2. The FIR lowpass filter has a passband extending from –400 Hz to +400 Hz, a passband gain of unity, a transition region width of 20 Hz, and a stop-band attenuation of 60 dB.

(a) Draw the spectral magnitude of $x(n)$ showing all spectral components in the range of $-2f_s$ to $+2f_s$.
(b) Draw the spectral magnitude of $y(n)$ showing all spectral components in the range of $-2f_s$ to $+2f_s$.
(c) What is the time-domain peak amplitude of the sinusoidal $y(n)$ output?

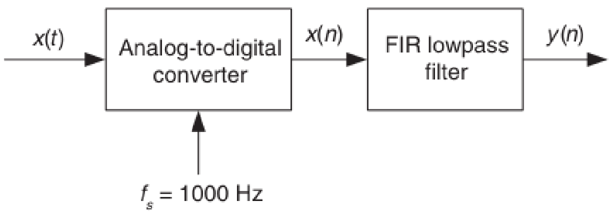


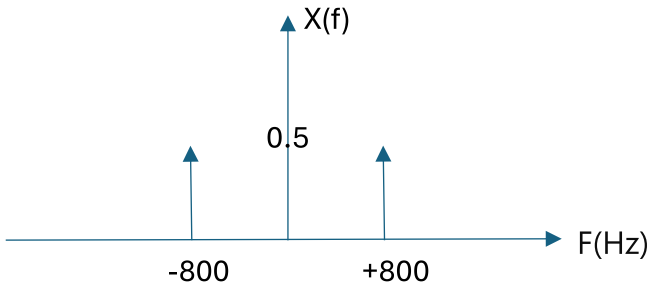
Figure P5–2

Answer:

a)

The analog signal X(t) = cos(2pi800t) has a single frequency component at 800 Hz.

Therefore, the diagram spectrum is:

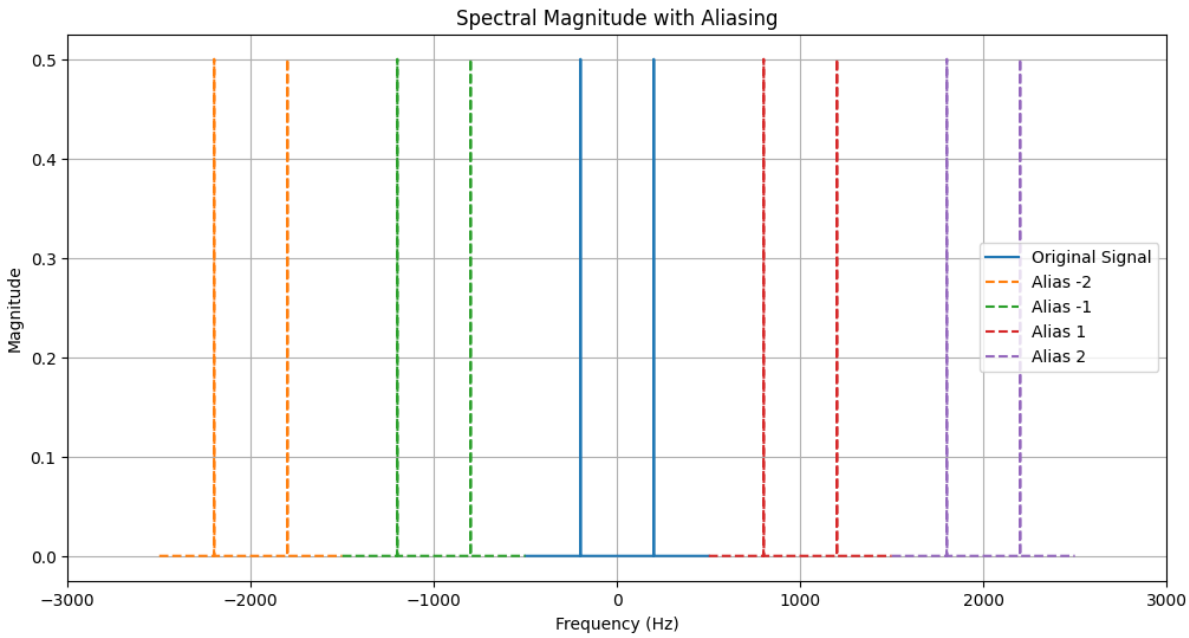


X(f) = 0.5[delta(f-800) + delta(f+800)]

In discrete time, the sampling frequency used is fs=1000 Hz.

The spectral magnitude of the signal x[n] is:

Spectral Magnitude of X(n):



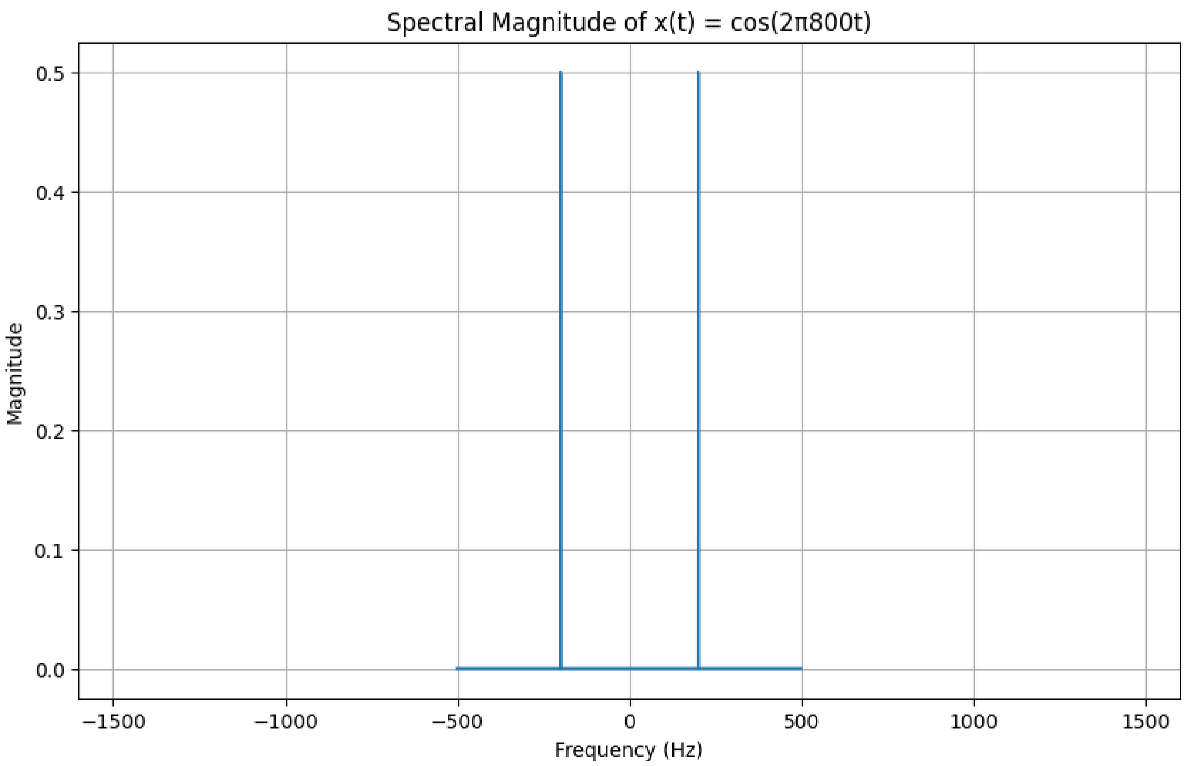
b)

X[n] is passed via the low-pass filter and the output y[n] produced will be from -400 Hz to 400 Hz.

The magnitude Y(f) has 0.5 values at -200Hz and 200Hz.

The expression for Y(f) is:

Y(f) = 0.5[delta(f-200) + delta(f+200)]



c)

Taking inverse Fourier transform:

Y[n] = F^{-1} [0.5[delta(f-200) + delta(f+200)]] = cos(2pi 200n) = cos(400n).

The peak amplitude of y[n] is 1.

ES3:

5.3 Assume we want to filter the audio signal from a digital video disc (DVD) player as shown in Figure P5–3. The filtered audio signal drives, by way of a digital-to-analog (D/A) converter, a speaker. For the audio signal to have acceptable time synchronization with the video signal, video engineers have determined that the time delay of the filter must be no greater than 6×10^{-3} seconds. If the f_s sample rate of the audio is 48 kHz, what is the maximum number of taps in the FIR filter that will satisfy the time delay restriction? (Assume a linear-phase FIR filter, and zero time delay through the D/A converter.)

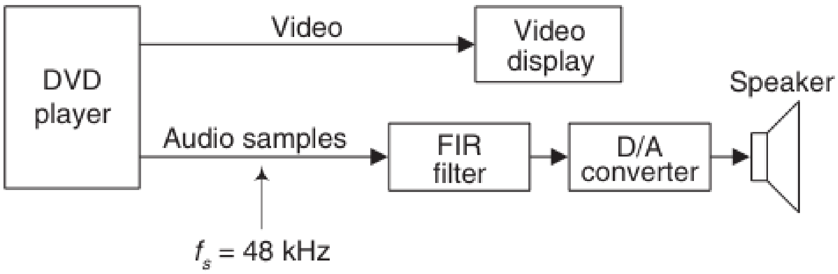


Figure P5-3

Answer:

To determine the maximum number of taps for the FIR filter that will satisfy the time delay restriction, it is needed to use the relationship between the number of taps in an FIR filter and its time delay.

For a linear-phase FIR filter, the time delay τ is given by:

τ =(n-1/2)(1/fs)

where:

N is the number of taps,

Fs is the sampling frequency.

Given:

$\tau \leq 6\times10^{-3}$ seconds,

fs=48 kHz

it is possible to rearrange the formula to solve for N:

$N <= 2\,\tau\,\mathrm{fs} + 1$

$N <= 2\text{x}\ 6\text{x}10^{-3}\text{x}\ 48000 + 1$

$N \leq 577$

Thus, the maximum number of taps N that the FIR filter can have to satisfy the time delay restriction is 577.

ES4:

5.4 There are times when we want to build a lowpass filter and a highpass filter that are *complementary*. By "complementary" we mean that a highpass filter's passband covers the frequency range defined by a lowpass filter's stopband range. This idea is illustrated in Figure P5–4(a). An example of such filters is an audio system, shown in Figure P5–4(b), where the low-frequency spectral components of an $x(n)$ audio signal drive, by way of a digital-to-analog (D/A) converter, a low-frequency speaker (woofer). Likewise, the high-frequency spectral components of $x(n)$ drive a high-frequency speaker (tweeter). Audio

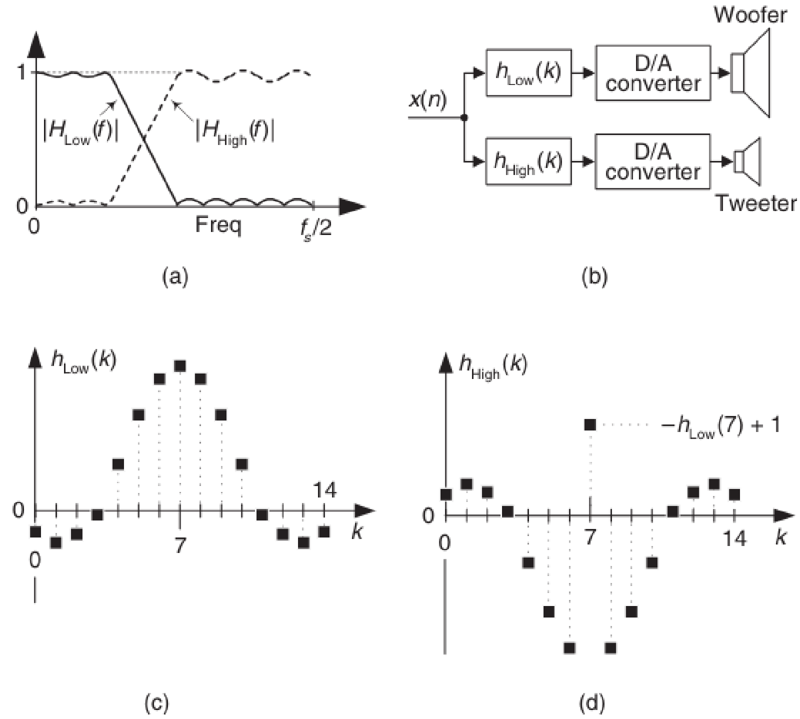


Figure P5–4

enthusiasts call Figure P5–4(b) a "crossover" network. Assuming that the lowpass filter is implemented with a 15-tap FIR filter whose $h_{\text{Low}}(k)$ coefficients are those in Figure P5–4(c), the complementary highpass filter will have the coefficients shown in Figure P5–4(d). Highpass coefficients $h_{\text{High}}(k)$ are defined by

$$h_{\text{High}}(k) = \begin{cases} -h_{\text{Low}}(k), & \text{when } k \neq 7 \\ -h_{\text{Low}}(k)+1, & \text{when } k = 7. \end{cases}$$

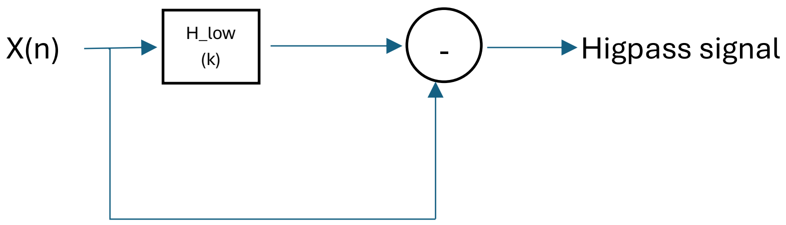
Here is the problem: Draw a block diagram of a system that performs the process in P5–4(b) where only the $h_{\text{Low}}(k)$ lowpass FIR filter need be implemented.

Answer:

To draw a block diagram for the system described in P5–4(b) where only the h_low(k) lowpass FIR filter needs to be implemented, it is possible to use the following approach:

- Use the h_low(k) filter to create the lowpass signal.
- Create the highpass signal by subtracting the output of the h_low(k) filter from the original input signal x(n).

Block diagram:



The input signal x(n) is passed through the h_low(k) lowpass filter to obtain the lowpass component.

To obtain the highpass component, the original signal x(n) is subtracted from the lowpass filter output.

ES5:

5.5 Think about a discrete System A, shown in Figure P5–5, that has an undesirable amplitude (gain) loss by a factor 0.5 (–6 dB), whose output requires lowpass linear-phase filtering. What can we do in the design of the lowpass FIR filter so the filter has an amplitude gain of 2 to compensate for System A's amplitude loss?

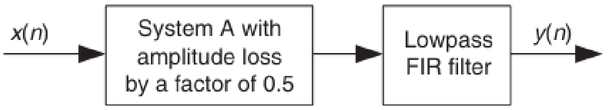


Figure P5–5

Answer:

To compensate for the amplitude loss by a factor of 0.5 (which corresponds to a -6 dB gain) in System A, it is possible to design the lowpass FIR filter to have an amplitude gain of 2. This will effectively negate the amplitude loss introduced by System A.

- System A has an amplitude loss of 0.5.
- To compensate for this loss, the FIR filter should have an amplitude gain of 2 (since 2×0.5=1, which restores the original amplitude).

If the original FIR filter coefficients are h(k), then the new FIR filter coefficients should be 2×h(k).

By scaling the FIR filter coefficients by 2, a gain of 2 cam be introduced, compensating for the 0.5 amplitude loss.

To make an example:

If the original FIR filter has coefficients: h(k) = [h0,h1,...,h_{N-1}], the new FIR filter should have the following coefficients:

H'(k) = 2 x h(k) = [2h0,2h1,...,2h_{N-1}].

The block diagram remains the same as in Figure, but with the FIR filter designed with the new scaled coefficients.

By scaling the coefficients of the FIR filter by 2, it is possible to compensate for the amplitude loss introduced by System A, ensuring that the overall system has the desired gain.

ES6:

5.6 Let's assume we have an $x(n)$ time sequence, whose f_s sample rate is 20 kHz, and its $|X(f)|$ spectral magnitude is that shown in Figure P5–6(a). We are required to design a linear-phase lowpass FIR filter that will attenuate the undesired high-frequency noise indicated in Figure P5–6(a). So we design a lowpass FIR filter whose frequency magnitude response is the $|H(f)|$ shown in Figure P5–6(b) and assume our filter design exercise is complete. Sometime later, unfortunately, we learn that the original $x(n)$ sequence's sample rate was *not* 20 kHz, but is in fact 40 kHz.

Here is the problem: What must we do to our lowpass filter's $h(k)$ coefficients, originally designed based on a 20 kHz sample rate, so that they will still attenuate $x(n)$'s undesired high-frequency noise when the f_s sample rate is actually 40 kHz?

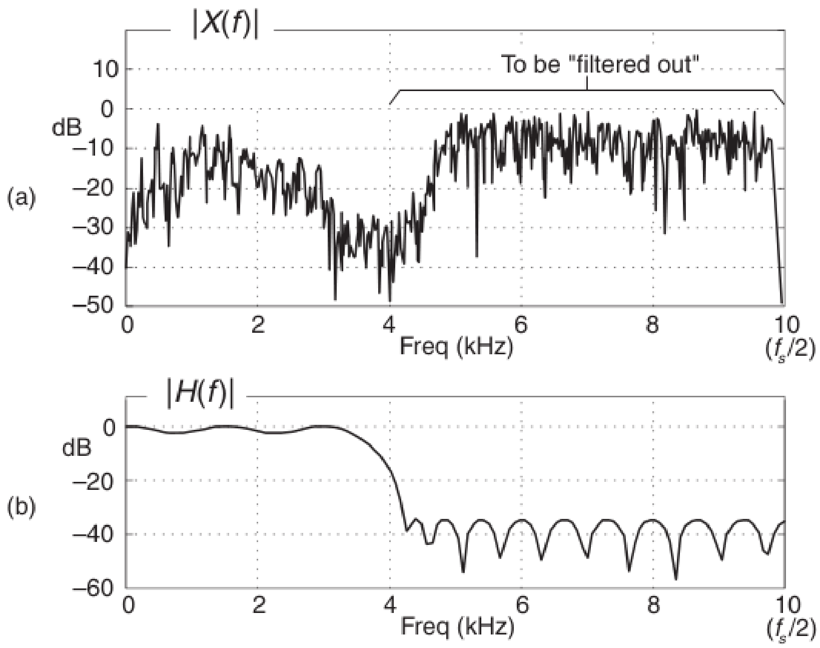


Figure P5–6

Answer:

To address the issue of having designed a lowpass FIR filter based on an incorrect sample rate, it is needed to adapt the filter coefficients to match the correct sample rate.

- The original design was based on a sample $f_{s,original}$ = 20 kHz.
- The actual sample rate is $f_{s,new}$ = 40 kHz.

- The filter needs to maintain its frequency characteristics relative to the new sample rate.

The frequency response of the filter is defined by its cut-off frequency relative to the sample rate. If the sample rate changes, the filter's coefficients must be adjusted to reflect this change.

The original cut-off frequency fc of the filter needs to be scaled to the new sample rate.

It is possible to use the relationship between the original and new sample rates to resample the filter coefficients.

The scaling factor is calculated as the ratio between $f_{s,new}$ and $f_{s,original}$.

Scaling factor = $f_{s,new}/ f_{s,original}$ = 40 kHz/ 20kHz = 2.

The original filter coefficients $h_{original}(k)$ need to be adjusted by the scaling factor.

This can be achieved by upsampling the filter coefficients by a factor of 2.

When upsampling by a factor of 2, insert a zero between each pair of the original coefficients.

Therefore, if the original filter length was N, the new filter length will be 2N -1.

To make an example:

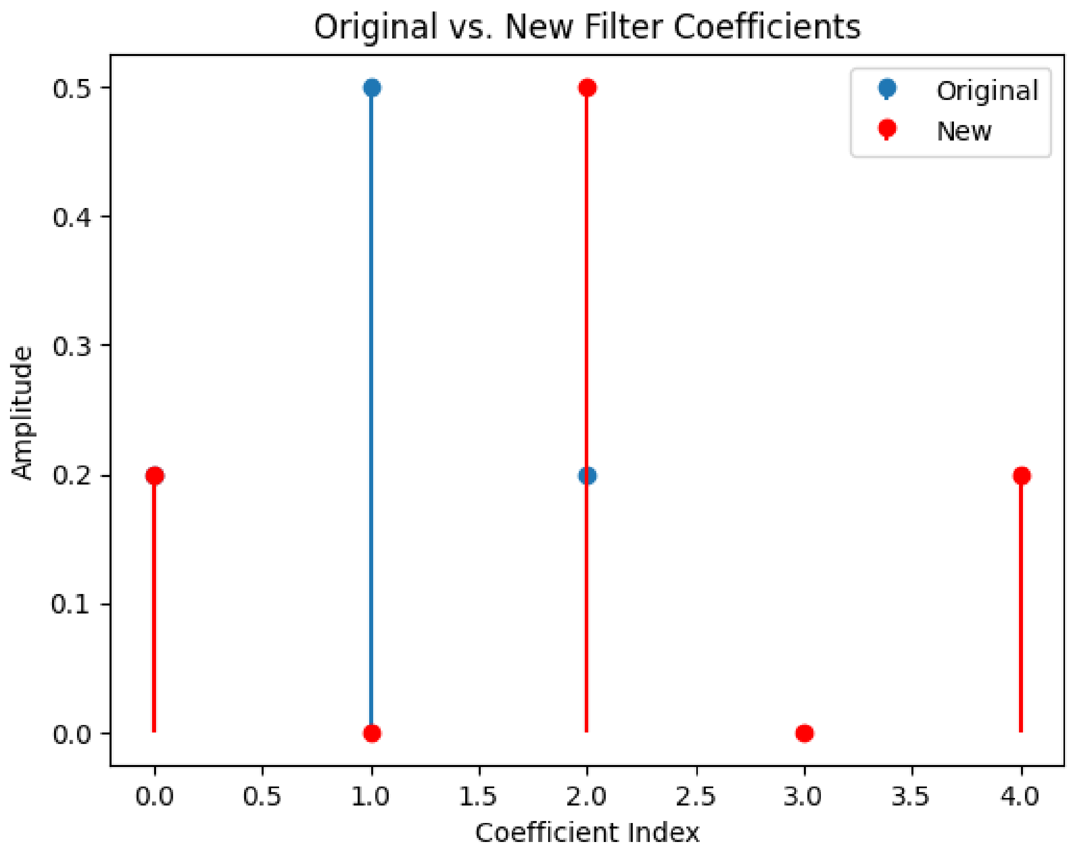
If the original filter coefficients are:

$H_{original}$ = [h0, h1,...,h_{N-1}]

The new filter coefficients will be:

H_{new}= [h0, 0, h1, 0,..., h_{N-1}, 0]

The adjustment of the filter coefficients ensures that the filter's frequency response matches the new sample rate. The block diagram of the filter system remains unchanged, but the internal filter coefficients are updated.



ES7:

5.7 Here is an interesting little problem. Think about applying the sinusoidal input sequence shown in Figure P5–7(a) to an 8-point moving average FIR filter. The filter's output sequence is that depicted in Figure P5–7(b).

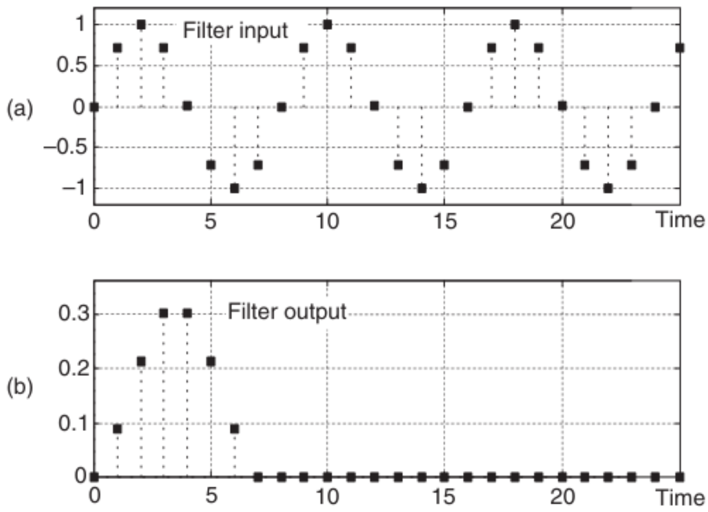


Figure P5–7

Answer:

a)

The 8-point moving average FIR filter has a specific characteristic in its frequency response that causes the filter's output sequence to go to zero. This characteristic is related to the filter's stopband.

A moving average FIR filter can be described by its impulse response:

H(n) = 1/N for n= 0,1,2,...,N-1, where N is the number of points in the moving average (N=8).

The frequency response of this filter is given by:

H(f) = (1/N) $\sum_{0}^{N-1} e^{-j2\pi fn}$

For the 8-point moving average filter, this becomes:

H(f) = (1/8) $\sum_{0}^{7} e^{-j2\pi fn}$

This frequency response has nulls (zeros) at certain frequencies. Specifically, for an 8-point moving average filter, the frequency response will be zero at:

f=k/8 for k=1,2,3,...,7

In other words, the filter has nulls at f=1/8, 2/8, 3/8, ..., 7/8 of the samplinf frequency fs. If the input sinusoidal sequence has a frequency that matches one of these nulls, the output of the filter will be zero at those frequencies. In Figure P5–7(a), the input sequence likely has a frequency component that aligns with one of these nulls, resulting in the output going to zero as seen in Figure P5–7(b).

b)

The initial nonzero-valued filter output samples are known as the "transient response" of the filter. When a filter is initially applied to a signal, the output includes a transient response period before it reaches its steady-state response.

During this transient period, the filter is adjusting to the input signal, and its output can have nonzero values even if the input signal frequency is at one of the filter's stopband frequencies. Once the filter settles, the output sequence reaches a steady state where the influence of the transient response diminishes, and the output becomes zero, as shown in Figure P5–7(b).

ES8:

5.8 Are abrupt (sudden) changes in the amplitude of a continuous, or discrete, signal associated with low or high frequencies?

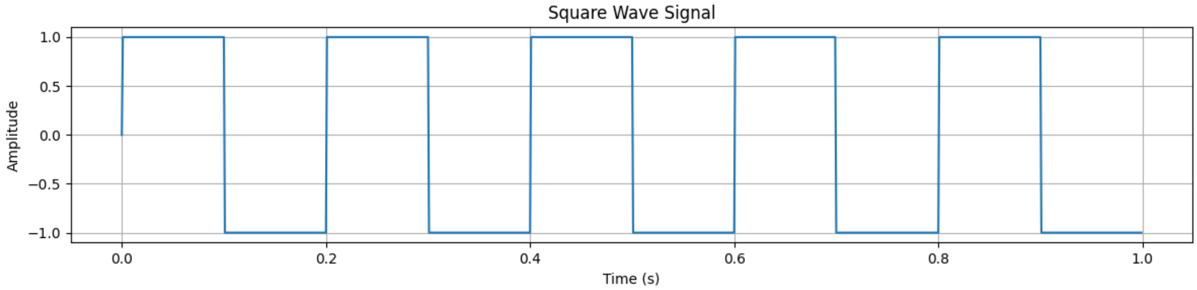
Answer:

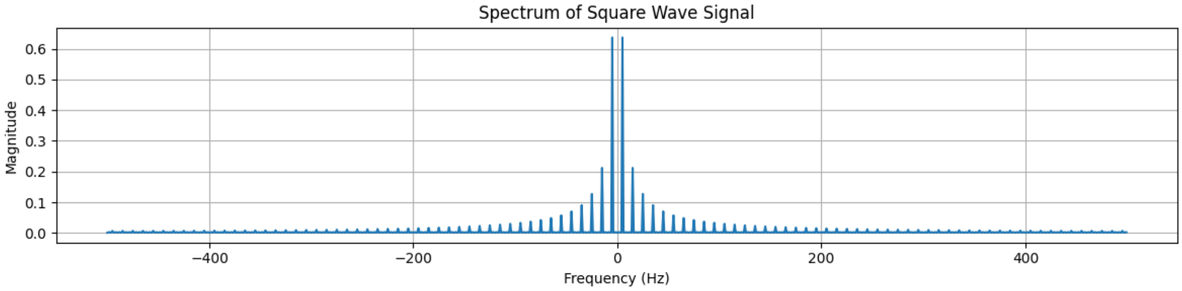
High-frequency components in a signal correspond to rapid changes in the signal's amplitude. This is because high-frequency sinusoids oscillate more quickly and can represent rapid transitions and sharp edges in the signal.

When a signal has abrupt changes, such as sharp edges or discontinuities, it contains significant high-frequency content. These high-frequency components are necessary to accurately represent the quick transitions in the signal.

For example, considering a square wave, which has abrupt changes between its high and low states, it is clearly observable in the Fourier series an infinite number of high-frequency harmonics indicating that a lot of high-frequency content is necessary to represent the abrupt transitions accurately.

On the other hand, a sinusoidal wave that varies smoothly over time contains only its fundamental frequency, which is a low-frequency component, and does not have abrupt changes.





ES9:

5.9 Consider an FIR filter whose impulse response is shown in Figure P5–9(a). Given the $x(n)$ filter input sequence shown in Figure P5–9(b):

(a) What is the length, measured in samples, of the nonzero-valued samples of the filter's output sequence?

(b) What is the maximum sample value of the filter's output sequence?

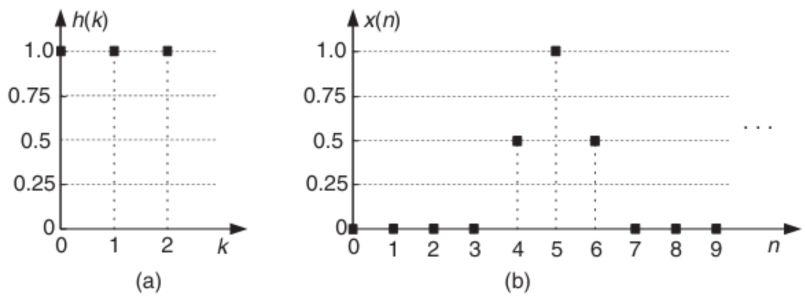


Figure P5–9

Answer:

a)

To determine the length of the nonzero-valued samples of the filter's output sequence, it is needed to understand the convolution process between the input sequence x(n) and the impulse response h(n) of the FIR filter.

Given:

- Impulse response h(k) = [1,1,1]
- Input sequence x(n) = [0,0,0,0,0.5,1,0.5,0,0,0]

Convolution of x(n) with h(k):

Y(n) = x(n) * h(k)

The length of the output sequence y(n) will be:

Length of y(n) = length of x(n) + length of h(k) -1

In this case:

Length of y(n) = 10 + 3 -1 = 12

To find the length of the nonzero valued samples, it is needed to identify the indeces where the output sequence y(n) is nonzero. Given the positions of nonzero elements in x(n) and considering the convolution with h(k), the nonzero output values will span from n=4 to n=6+2 = 8.

Thus, the length of the nonzero-valued samples of y(n) is: 8-4+1=5.

b)

To determine the maximum sample value of the filter's output sequence, it is needed to perform the convolution of x(n) and h(k):

$$y(n) = \sum_{k=0}^{2} h(k)x(n-k)$$

Calculating the convolution for the nonzero part:

1. At n=4:
   y(4) = 1 x(4) + 1 x(3) + 1 x(2) = 0.5 + 0 + 0 = 0.5
2. At n=5:
   y(5) = 1 x(5) + 1 x(4) + 1 x(3) = 1 x 1 + 1x0.5 + 0 = 1.5
3. At n=6:
   y(6) = 1 x(6) + 1 x(5) + 1 x(4) = 1x0.5 + 1x1 + 1x0.5 = 2
4. At n=7:
   y(7) = 1 x(7) + 1 x(6) + 1 x(5) = 0 + 0.5 + 0.5 = 1
5. At n=8:
   y(8) = 1 x(8) + 1x(7) + 1x(5) = 0 + 0 + 0.5 = 0.5

From the above calculations, the maximum sample value of the filter's output sequence is: 2.

ES10:

5.10 Consider an FIR filter whose impulse response is that shown in Figure P5–10(a). Given the $x(n)$ filter input sequence shown in Figure P5–10(b), draw the filter's output sequence.

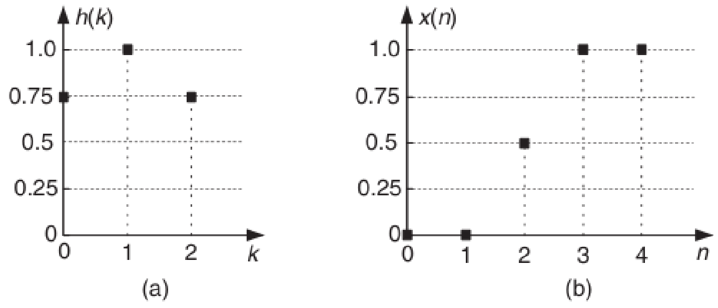


Figure P5–10

Answer:

To find the output sequence of the FIR filter, it is needed to perform the convolution of the input sequence x(n) with the impulse response h(k).

Given:

Impulse response h(k)=[0.75, 1, 0.75]

Input sequence x(n)=[0, 0, 0.5, 1.0, 1.0]

The convolution operation can be represented as:

y(n) = x(n)*h(n)

1. y(0) = h(0)x(0) = 0.75 x 0 = 0
2. y(1):
   y(1) = h(0)x(1) + h(1)x(0) = 0.75x0 + 1x0 = 0
3. y(2):
   y(2) = h(0)x(2) + h(1)x(1) + h(2)x(0) = 0.75x0.5 + 1x0 + 0.75x0 = 0.375
4. y(3):
   y(3) = h(0)x(3) + h(1)x(2) + h(2)x(1) = 0.75x1.0 + 1.0x0.5 + 0.75x0 = 1.25
5. y(4):
   y(4) = h(0)x(4) + h(1)x(3) + h(2)x(2) = 0.75x1.0 + 1.0x1.0 + 0.75x0.5 = 2.125
6. y(5):
   y(5) = h(1)x(4) + h(2)x(3) = 1x1 + 0.75x1.0 = 1.75
7. y(6):
   y(6) = h(2)x(4) = 0.75x1 = 0.75

The output sequence y(n) is:

Y(n) = [0, 0, 0.375, 1.25, 2.125, 1.75, 0.75]

Here below the plots: