ELCE5180 Digital Signal Processing

Assignment 1: Discrete Fourier Transform (DFT)

Name______________, Class&Student ID_________________

Aim

1. To study the Discrete Fourier Transform.

2. Use DFT to analyze the DTFT.

3. Use the FFT to implement the fast convolutions.

Introduction

1) **Fast Fourier Transform (FFT):** FFT is an algorithm to reduce the amount of computations involved in the DFT. MATLAB provides a function called *fft* to compute the FFT.

2) DFT is the sampled version of DTFT

$$X(k) = X(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k}, k = 0,1,\cdots, N-1$$

3) **Fast convolution**: The *conv* function compute the convolution between two finite-duration sequence, and is very efficient for smaller values of N (<50). For large values of N it is possible to speed up the convolution using the FFT

$$x_1(n) * x_2(n) = IFFT\left[FFT\left[x_1(n)\right]\cdot FFT\left[x_2(n)\right]\right]$$

this algorithm is called fast convolution.

Requirements

1. Let $x(n)$ be a 4-point sequence:

$$x(n) = \{\underset{\uparrow}{2}, -1, 1, 1\}$$

a. Plot the magnitude and phase of discrete-time Fourier transform $X(e^{j\omega})$.

b. Computer the 4-point DFT of $x(n)$, use *stem* to plot its magnitude and phase. Verify that the above DFT is the sampled version of DTFT. It may be helpful to combine the above two plots in one graph using the *hold* function.

c. Let $x(n)$ be a 64-point sequence by appending 60 zeros (using the *zeros* function), then computer the 64-point DFT. Can we obtain other samples of the DTFT $X(e^{j\omega})$?

2. Let a finite-length sequence be given by

$$x(n) = 2e^{-0.9n}, 0 \le n \le 9$$

Plot the DTFT $X(e^{j\omega})$ of the above sequence using DFT as a computation tool. Choose the length N of DFT so that this plot appears to be a smooth graph.

3. Consider the sequence

$$x(n) = \cos(0.48\pi n) + \cos(0.52\pi n)$$

We want to determine its spectrum based on the finite number of samples.

a. Determine and plot the DTFT of $x(n)$, $0 \le n \le 10$.

b. Determine and plot the 100-point DFT of the above $x(n)$ by appending 89 zeros.

c. Determine and plot the DTFT of $x(n)$, $0 \le n \le 99$.

To illustrate the difference between the high-density spectrum and the high-resolution spectrum.

4. Let $x(n)$ be an *L*-point uniformly distributed random number between [0,1], and let $h(n)$ be an *L*-Point Gaussian random sequence with mean 0 and variance 1 (Using *rand* and *randn* function). Determine the average execution times of linear convolution and fast convolution for *L*=8,16,32,64,256,512,1024,2048, in which the average is computed over the 100 times realizations for each L.

ELCE5180 Digital Signal Processing

Assignment 2: IIR Filter Design

Name______________, Class&Student ID_________________

Aim

1. Design analog low-pass filters.

2. Study and apply filter transformations to obtain digital low-pass filters.

Introduction

1) The basic technique of IIR filter design is to transform well-known analog filters to digital filters using complex-valued mapping. There are two widely used transformations: namely impulse invariance transformation and bilinear transformation.

2) A typical specification of a low-pass digital filter is shown in Fig. 2-1, in which

- band $[0, \omega_p]$ is called passband, and δ_1 is the tolerance (or ripple) that we are willing to accept in the ideal passband response,
- Band $[\omega_{st}, \pi]$ is called stopband, and δ_2 is the corresponding tolerance (or ripple) ,
- band $[\omega_{p,} \omega_{st}]$ is called transition band,
- R_p is the passband ripple in dB, and
- A_s is the stopband attenuation in dB.

And we have

$$R_p = -20 \log_{10} \frac{1-\delta_1}{1+\delta_1}$$

$$A_s = -20 \log_{10} \frac{\delta_2}{1+\delta_1}$$

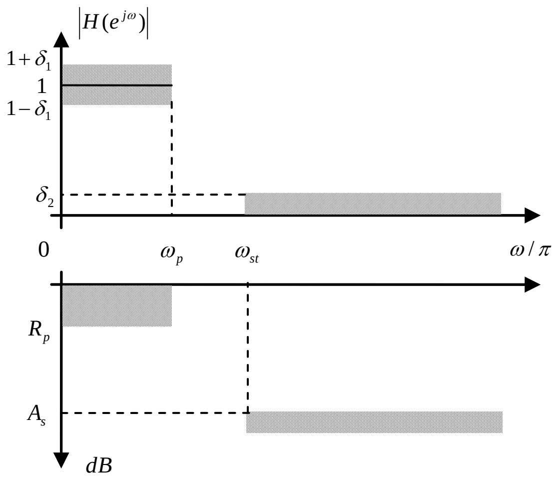
3) A typical specification of a low-pass analog filter is shown in Fig. 2-2, in which

- $|H_a(j\Omega)|^2$ is called magnitude-squared response,
- ε is a passband ripple parameter, Ω_p is the passband cutoff frequency in rad/sec,
- A is a stopband attenuation, and Ω_{st} is the stopband cutoff frequency in rad/sec.

And we have

$$R_p = -10\log_{10}\frac{1}{1+\varepsilon^2} \qquad \varepsilon = \sqrt{10^{R_p/10}-1}$$

$$A_s = -10\log_{10}\frac{1}{A^2} \qquad A = 10^{A_s/20}$$



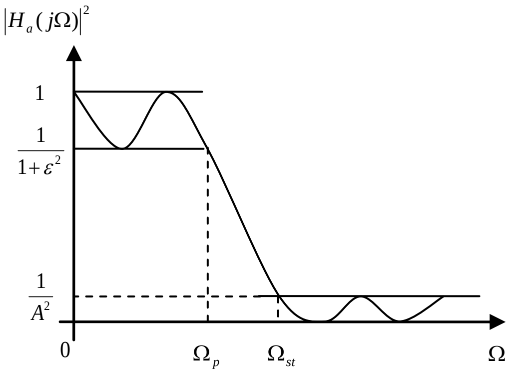


Fig. 2-1 Digital low-pass filter specifications Fig. 2-2 Analog low-pass filter specifications

4) **Butterworth loe-pass filters:** The magnitude-squared response of a *N*-order Butterworth analog loepass filter is given by

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

Where *N* is the order of the filter and Ω_c is the 3dB cutoff frequency in rad/sec.

The essence of the design in the case Butterworth filter is to obtain the order N and the cutoff frequency Ω_c from the parameters Ω_p, R_p, Ω_{st}, and A_s. And we have the design equations as follow

$$N = \left\lceil \frac{\log_{10}\left[\left(10^{R_p/10}-1\right)/\left(10^{A_s/10}-1\right)\right]}{2\log_{10}\left(\Omega_p/\Omega_{st}\right)} \right\rceil$$

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{10^{R_p/10}-1}}$$

$$\Omega_c = \frac{\Omega_{st}}{\sqrt[2N]{10^{A_s/10}-1}}$$

Where the operation $\lceil x \rceil$ means “choose the smallest integer lager than *x*”. And to satisfy the specifications exactly at Ω_p or Ω_{st}, we obtain Ω_c from the corresponding equation above. MATLAB provides a function called *butter* to design Butterworth analog filter, and the function is invoked by

[b,a] = butter(N,Wn,'s')

Here, *N* is the order of the filter, *Wn* is the 3dB cutoff frequency, and the string *'s'* means to design a analog filter.

5) **Chebyshev low-pass filters:** The magnitude-squared response of a Chebyshev-I filter is

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 c_N^2\left(\frac{\Omega}{\Omega_c}\right)}$$

Where $c_N(x)$ is the *N*-order Chebyshev polynomail.

We have the design equations as follow

$$\Omega_c = \Omega_p$$

$$N = \left\lceil \frac{\operatorname{arccos} h\left(\sqrt{A^2-1}/\varepsilon\right)}{\operatorname{arccos} h\left(\Omega_{st}/\Omega_p\right)} \right\rceil$$

MATLAB provides a function called *cheby1* to design Chebyshev-I analog filter, and the function is invoked by

[b,a] = cheby1(N,Rp,Wn,'s')

The magnitude-squared response of a Chebyshev-II filter is

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left[\varepsilon^2 c_N^2\left(\frac{\Omega_c}{\Omega}\right)\right]^{-1}}$$

The design equations for Chebyshev-II prototype are similar to those of Chebyshev-I except that

$$\Omega_c = \Omega_{st}$$

MATLAB provides a function called *cheby2* to design Chebyshev-II analog filter, and the function is invoked by

[b,a] = cheby2(N,As,Wn,'s')

6) **Analog -to-digital filter transformations:** MATLAB provides a function called *impinvar* to apply impulse invariance transformation, and a function called *bilinear* to apply bilinear transformation.

8) The *freqs* function returns the frequency response of analog filters, and the *freqz* function

returns the frequency response of digital filters.

9) It is a fact that the *butter*, *cheby1* and *cheby2* functions can be used to design digital filter directly. And there is also another set of filter design functions, namely the *buttord*, *cheb1ord*, and *cheb2ord* functions, which can provide filter order *N* and cutoff frequency *Wn*. These function use bilinear transformation because of its desirable advantages. However, you are not supposed to use these functions to design digital filter directly in this exercise.

Requirements

1. Design a low-pass digital filter to satisfy

$$\omega_p = 0.2\pi\ , R_p = 1dB$$

$$\omega_{st} = 0.3\pi\ , A_s = 15dB$$

(1) Use a Butterworth prototype and impulse invariance transformation.

(2) Use a Chebyshev-I prototype and impulse invariance transformation.

(3) Use a Chebyshev-II prototype and impulse invariance transformation.

(4) Use a Butterworth prototype and bilinear transformation.

(5) Use a Chebyshev-I prototype and bilinear transformation.

(6) Use a Chebyshev-II prototype and bilinear transformation.

Comment on the results. Is this a satisfactory design? Why?

ELCE5180 Digital Signal Processing

Assignment 3: FIR Filter Design

Name______________, Class&Student ID_________________

Aim

To study two design techniques, namely the window design, and the frequency sampling design for linear-phase FIR filters.

Introduction

1) **Linear-phase FIR filters.** The impulse response $h(n)$ must be symmetric, that is

$$h(n) = \pm h(N-1-n)$$

When the cases of symmetry and antisymmetry are combine with odd and even *N*, we obtain four types of linear-phase FIR filters. Frequency response functions for each of these types have some peculiar expressions and shapes. To study these responses, we write $H(e^{j\omega})$ as

$$H(e^{j\omega}) = \pm\left|H(e^{j\omega})\right|e^{j\theta(\omega)} = H(\omega)e^{j\theta(\omega)}$$

where $H(\omega)$ is an amplitude response function and not a magnitude response function. The amplitude response is a real function, but unlike the magnitude response, which is always positive, The amplitude response may be both positive and negative. And $\theta(\omega)$ is an phase function, which is a continuous linear function.

When the impulse response $h(n)$ is symmetry,

$$H(\omega) = \sum_{n=1}^{N-1} h(n)\cos\left[\left(\frac{N-1}{2}-n\right)\omega\right]$$

$$\theta(\omega) = -\left(\frac{N-1}{2}\right)\omega \qquad (3\text{-}1)$$

When the impulse response $h(n)$ is antisymmetry,

$$H(\omega) = \sum_{n=1}^{N-1} h(n)\sin\left[\left(\frac{N-1}{2}-n\right)\omega\right]$$

$$\theta(\omega) = -\left(\frac{N-1}{2}\right)\omega + \frac{\pi}{2} \qquad (3\text{-}2)$$

Type-1 Symmetrical impulse response, *N* odd

$$H(\omega) = \sum_{n=0}^{(N-1)/2} a(n)\cos(\omega n)$$

$$a(0) = h\left(\frac{N-1}{2}\right)$$

$$a(n) = 2h\left(\frac{N-1}{2} - n\right)\ ,\ n = 1,2,\cdots,\frac{N-1}{2}$$

Type-2 Symmetrical impulse response, *N* even

$$H(\omega) = \sum_{n=1}^{N/2} b(n)\cos\left[\left(n - \frac{1}{2}\right)\omega\right]$$

$$b(n) = 2h\left(\frac{N}{2} - n\right)\ ,\ n = 1,2,\cdots,\frac{N}{2}$$

Type-3 Antisymmetrical impulse response, *N* odd

$$H(\omega) = \sum_{n=0}^{(N-1)/2} c(n)\sin(n\omega)$$

$$c(n) = 2h\left(\frac{N-1}{2} - n\right)\ ,\ n = 1,2,\cdots,\frac{N-1}{2}$$

Type-4 Antisymmetrical impulse response, *N* evev

$$H(\omega) = \sum_{n=1}^{N/2} d(n)\sin\left[\omega\left(n - \frac{1}{2}\right)\right]$$

$$d(n) = 2h\left(\frac{N}{2} - n\right)\ ,\ n = 1,2,\cdots,\frac{N}{2}$$

MATLAB provides a function called *zerophase* to calculate the amplitude response function. And the phase function can be calculate by (3-1) or (3-2).

2) **Window design techniques.**

The basic idea behind the window design is to choose a proper ideal frequency-selective filter(which always has a noncausal,infinte-duration impulse response)and then to truncate(or window)its impulse response to obtain a linear-phase and causal FIR filter.Therefore the emphasis in this method is on selecting an appropriate windowing function and an appropriate ideal filter.We will denote an ideal frequency-selective filter by $H_d(e^{j\omega})$,which has a unity magnitude gain and linear-phase characteristics over its passband,and zero response over its stopband.An ideal low pass filter of bandwidth $\omega_c < \pi$ is given by

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega} & |\omega| \le \omega_c \\ 0 & \omega_c < |\omega| \le \pi \end{cases}$$

Then,the impulse response of this filter is given by

$$h_d(n) = F^{-1}\left[H_d(e^{j\omega})\right] = \frac{\sin\left[\omega_c(n-\alpha)\right]}{\pi(n-\alpha)}$$

Note that $h_d(n)$ is symmetric with respect to α ,a fact useful for linear-phase FIR filters.

To obtain an FIR filter from $h_d(n)$,one has to truncate $h_d(n)$ on both sides.That is, $h(n)$ can be thought of as being formed by the product of $h_d(n)$ and a window function $w(n)$ as follows:

$$h(n) = h_d(n)w(n)$$

There are several kinds of window functions.In Table 3.1 we provide a summary of fixed window function characteristics in terms of their transition widths(as a function of N)and their minimum stopband attenuations in dB.Both the approximate as well as the exact transition bandwidths are given.

Table 3.1 Summary of commonly used window function characteristics

Window name	Window function frequency characteristics		Specification	
	Side Lobe Magnitude（dB）	Main Lobe Width	Transition Width	Min.Stopband Attenuation（dB）
Rectangular	-13	$4\pi/N$	$1.8\pi/N$	-21
Hanning	-31	$8\pi/N$	$6.2\pi/N$	-44
Hamming	-41	$8\pi/N$	$6.6\pi/N$	-53
Blackman	-57	$12\pi/N$	$11\pi/N$	-74

MATLAB provides several functions to implement window functions above

Table 3.2 MATLAB functions for implementing window function

MATLAB function	**Window name**	**MATLAB function**	**Window name**
`boxcar`	Rectangular	`blackman`	Blackman
`hanning`	Hanning	`kaiser`	Kaiser
`hamming`	Hamming		

3)**Frequency sampling design techniques.**Given the ideal lowpass filter $H_d(e^{j\omega})$,choose the filter length N and then sample $H_d(e^{j\omega})$ at N equispaced frequencies between 0 and 2π

$$H(k) = H_d(k) = H(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k} \qquad k = 0,1,\cdots,N-1$$

Then we have

$$h(n) = IDFT[H(k)]$$

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} H(k)\phi\left(\omega - \frac{2\pi}{N}k\right)$$

$$\phi(\omega) = \frac{\sin(N\omega/2)}{N\sin(\omega/2)} \cdot e^{-j\omega\frac{N-1}{2}}$$

we write $H(k)$ as

$$H(k) = |H(k)|e^{j\theta(k)} = H_r(k)e^{j\theta(k)}$$

For a linear-phase FIR filter we have

$$h(n) = \pm h(N-1-n)$$

and $H(k) = H^*(N-k)$

When $h(n)$ is a real sequence,we have

$$H_r(k) = H_r(N-k)$$

$$\theta(k) = \begin{cases} -\left(\frac{N-1}{2}\right)\frac{2\pi k}{N} & k = 0,\cdots,\left\lfloor\frac{N-1}{2}\right\rfloor \\ \left(\frac{N-1}{2}\right)\frac{2\pi}{N}(N-k) & k = \left\lfloor\frac{N-1}{2}\right\rfloor + 1,\ldots,N-1 \end{cases} \quad \text{(Type-1 and Type-2)}$$

$$\theta(k) = \begin{cases} \pm\frac{\pi}{2} - \left(\frac{N-1}{2}\right)\frac{2\pi k}{N} & k = 0,\cdots,\left\lfloor\frac{N-1}{2}\right\rfloor \\ \mp\frac{\pi}{2} + \left(\frac{N-1}{2}\right)\frac{2\pi}{N}(N-k) & k = \left\lfloor\frac{N-1}{2}\right\rfloor + 1,\ldots,N-1 \end{cases} \quad \text{(Type-3and Type-4)}$$

4) The *freqz* function returns the frequency response of digital filters.

5) It is a fact that the *fir1*, and *fir2* functions can be used to design FIR digital filter directly. However, you are not supposed to use these functions to design digital filter directly in this exercise.

Requirements

1. We have $h(n) = \{3,-1,2,-3,5,-3,2,-1,3\}$, to plot the amplitude response function, phase

function, amplitude response, and phase response.

2. Design a low-pass digital filter to satisfy

$$\omega_p = 0.2\pi\ ,\quad R_p = 0.25dB$$
$$\omega_{st} = 0.3\pi\ ,\quad A_s = 50dB$$

Choose Rectangular window function, Hanning window function, Hamming window function, and Blackman window function to design the low-pass filter respectively.Comment on the results. Is this a satisfactory design? Why?

3. Design a band-pass digital filter with the following specifications

$$\omega_{st1} = 0.2\pi\ ,\quad A_s = 60dB$$

$$\omega_{p1} = 0.35\pi\ ,\quad R_p = 1dB$$

$$\omega_{p2} = 0.65\pi\ ,\quad R_p = 1dB$$

$$\omega_{st2} = 0.8\pi\ ,\quad A_s = 60dB$$

Choose an appropriate window function to design the necessary band-pass filter.

4. Design a low-pass digital filter using frequency sampling approach to satisfy

$$\omega_p = 0.2\pi\ ,\quad R_p = 0.25dB$$
$$\omega_{st} = 0.3\pi\ ,\quad A_s = 50dB$$

(1) Choose N=20,

(2) Choose N=40, so that we have one sample in the transition band, denote these samples by T=0.39,

(3) Choose N=60, so that we have two sample in the transition band, denote these samples by T1=0.5925, T2= 0.1099.

(4) Comment on the results. Is this a satisfactory design?