EE6401

Advanced Digital Signal Processing

Semester 1 Weeks 1 - 7

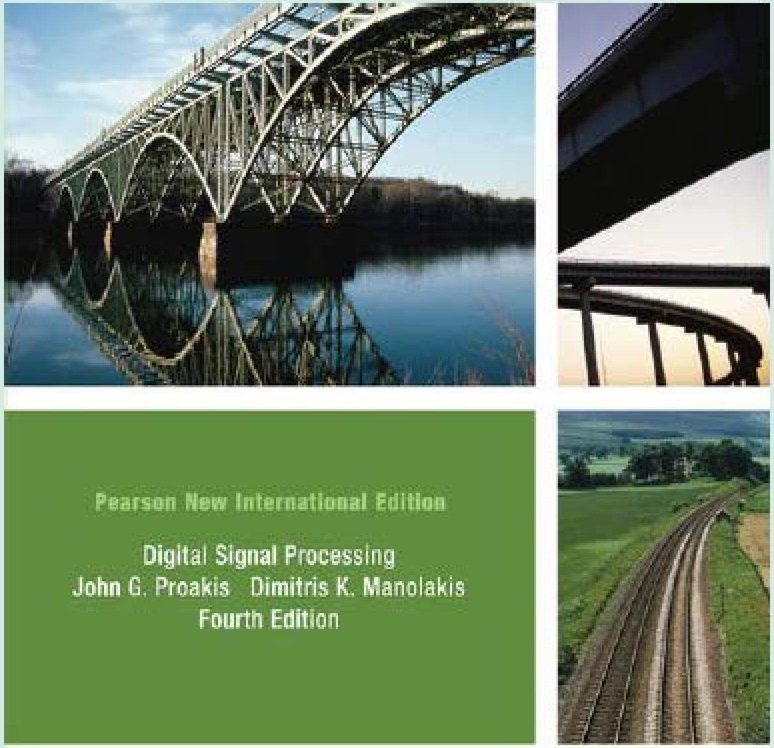
Dr Bi Guoan

School of EEE

Division of Information Engineering

Email:EGBI@NTU.EDU.SG

Tel: 67904823



EE6401 Advanced Digital Signal Processing

Textbook author: John G. Proakis

Textbook title: Digital Signal Processing: Pearson New International Edition

Textbook edition: 4

ISBN: 9781292025735

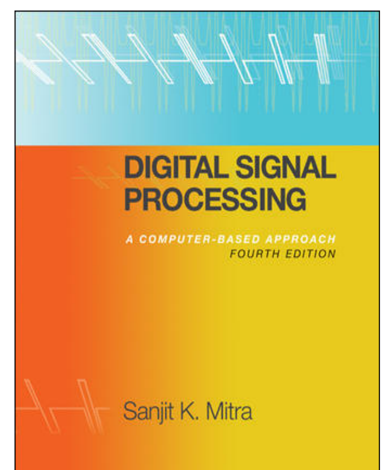
Reference Book:

Digital Signal Processing: A computer based Approach, 4/e

Author : S. K. Mitra

Publisher : Mcgraw-Hill

ISBN : 9780071289467



- Part 1 (weeks 1 – 2) briefly reviews the foundations on digital signal processing. Two main topics are:
  - Sampling process
  - Signal representation in frequency domain.
- Part 2 (weeks 3 – 4) is on the topics of analysis, representations of signals and digital filter design.
- Parts 3,4,5 (weeks 5 - 6) focuses on various techniques for implementation of multi-rate signal processing systems and applications.
- Week 7 for revision, Weeka 8-13 will be taught by Prof Lin Zhiping.
- Relevant questions are also given for each week. Their solutions will be given in Week 7.
- All materials are available in NTULearn.ntu.edu.sg
- **Evaluation:** Two assignments (10% each) and one final examination (80%).

Useful Information

- The coverage of this subject contains a large amount theoretical materials that require sufficient math knowledge, although efforts have been made to be less mathematical.
- Reading notes are available from most slides.
- Weekly revision on the lecture notes is very much needed.
- Study groups among students are beneficial for better understanding
- The most direct way to clear any confusion is to ask the lecturer
- Use the given questions in each week to test your understanding on the lecture materials. Solutions will be given in week 7.
- You may have learnt some basics of DSP, which are reviewed in the first 3 weeks
- Lecture videos are given after each lecture.

Digital Processing of Continuous-Time Signals

- Digital processing of a continuous-time signal involves the following basic steps:

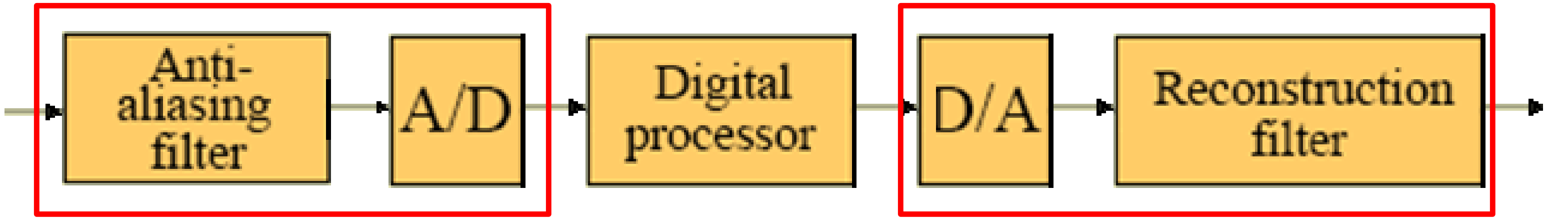


Figure 1 General block diagram of a digital system

- Anti-aliasing filter prevents aliasing before the A/D conversion
- Analog-to-Digital (A/D) converter converts each sampled value into one of a finite number of discrete values
- DSP system processes the discrete-time signal for the specific application, which is our **main focus**
- Digital-to-Analog (D/A) converter converts the processed discrete sequence into a continuous-time signal. Reconstruction filter eliminates any components outside the baseband.
- These functional blocks are often located at different places, for example, the audio applications

Digital Signal Processing

- One main objective of DSP is to extract the useful information from the signals – **Signal Analysis**
- Method of information extraction depends on the type of signals and the characteristics of the information carried by the signals.
- The **analysis outputs** and **various expressions** of signals, and **application requirements** in both time and transform domains are essential for DSP studies.
- Digital filtering is the most widely used operation to pass certain frequency components, with a minimum distortion, and block other frequency components.
- We will focus on specification, design and implementation of digital filters for both mono- and multi-rate systems.
- It is assumed that you have learn the topics presented in the first two weeks. These topics are the foundations for the more advanced topics.

Sampling Process

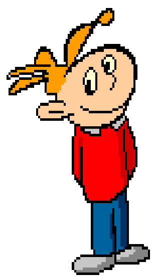
What makes the following signals different?

Speech signal 1 and Speech signal 2.

This difference between the sounds of the **same signal** is due to the use of different sampling frequencies.

A/D Conversion

- There exist **many** different continuous-time (analog) signals whose sampled sequences lead to the **same** discrete-time signal, see the example below



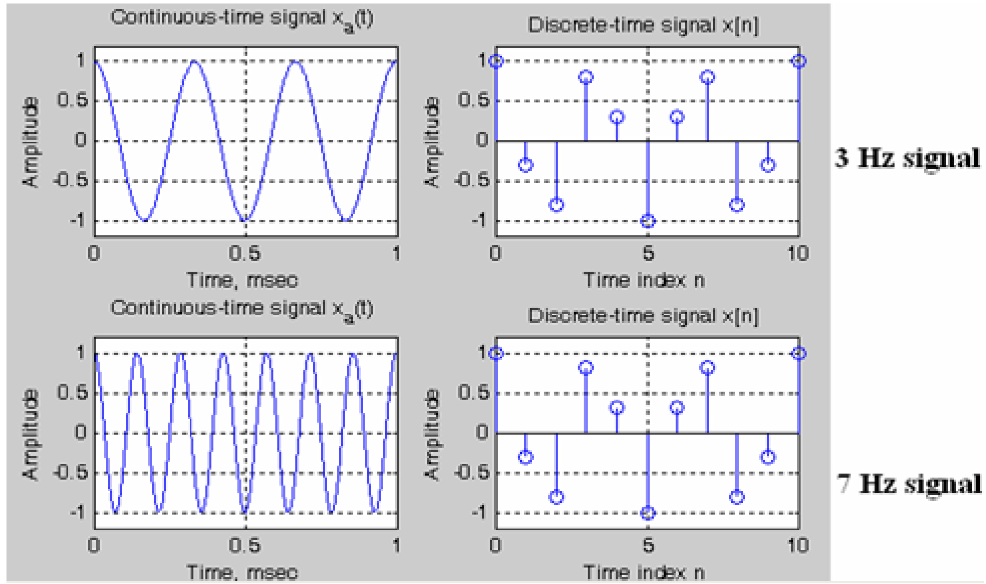


Figure 2 sampling two signals at sampling frequency of 5 Hz

- The **key requirement** for sampling is to relate a ***unique*** continuous-time signal $x(t)$ to a given discrete-time signal $x(n)$ and to *truthfully* ***recover*** $x(t)$ from its sampled values $x(n)$.

- Because the digital form, $x(n)$, is an ***approximation* to** $x(t)$, the basic requirement for the analogue to digital conversion process is to achieve ***acceptable accuracy*** of the approximation at an ***acceptable cost***

Analogue to Digital Conversion

- The two key parameters are:
  - Sampling frequency, and
  - The number of bits to represent each sample

- Contradictory Requirements:
  - The sampling rate should be **as *low* as** possible to minimize the computational loads (or costs). For better accuracy, however, the sampling rate should be **as *high* as** possible.
  - The no. of bits/sample should be **as *small* as** possible to minimize the computational time/hardware costs. For better accuracy, however, it should be **as *high* as** possible.
- The original signal $x(t)$ should be recovered from its digital form $x(n)$ with ***acceptable distortions***

Aliasing FORMULA Reveals the Sampling Theorem

Effects of Sampling in the Frequency Domain

- Let $g_a(t)$ be a *continuous-time* signal that is sampled at the time instants $t = nT$, generating the ***digital*** sequence

$$\boxed{g[n]= g_a(nT)=g_a(n/F_T), \qquad -\infty < n < \infty}$$

with T being the sampling period.

- The *sampling frequency* is $F_T = 1/T$
- The frequency-domain representation of $g_a(t)$ is given by its continues-time Fourier transform (CTFT):

$$G_a(\Omega) = \int_{-\infty}^{\infty} g_a(t) e^{-j\Omega t} dt$$

- The frequency-domain representation of $g[n]$ is given by its discrete-time Fourier transform (DTFT)

$$G_p(\omega) = \sum_{n=-\infty}^{\infty} g[n] e^{-j\omega n}$$

Effects of Sampling in the Frequency Domain

- It is proved that the DTFT of the sampled signal is expressed as

$$\boxed{G_p(\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(\Omega - k\Omega_T)}$$

where $G_a(\Omega)$ is the CTFT of $g_a(t)$.

- This equation is known as ***Aliasing Formula***. Understanding this formula is essential to appreciate the ***Nyquest Sampling Theorem***.

- $G_p(\Omega)$ is a periodic function of Ω consisting of a sum of *shifted* and *scaled* replicas of $G_a(\Omega)$, shifted by integer multiples of $\Omega_T = 2\pi F_T$ (radian sampling frequency) and scaled by $1/T$.

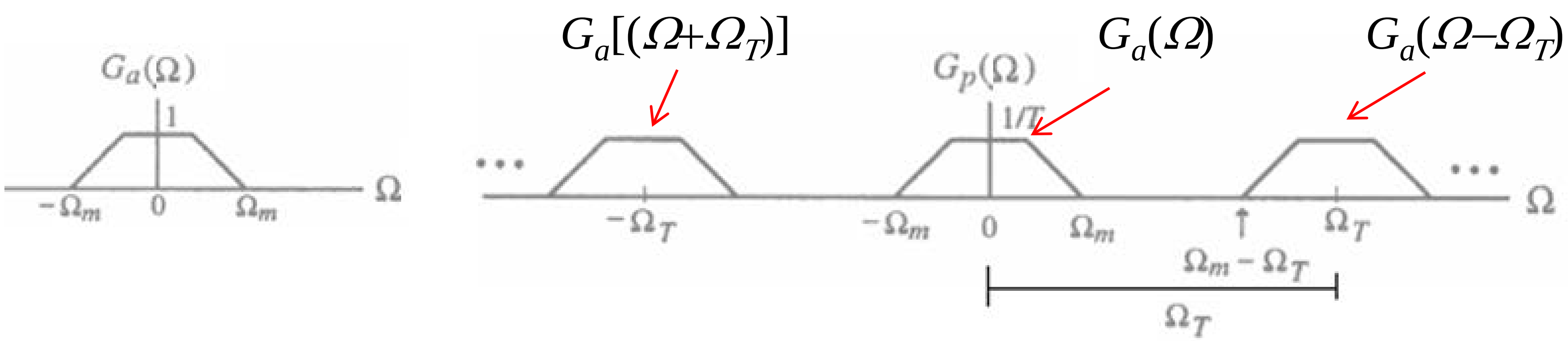


Figure 3 Spectra of an analog and digital signals

Effects of Sampling in the Frequency Domain

- The term of $G_p(\Omega)$ for $k = 0$ is the baseband portion (the same as $G_a(\Omega)/T$) and each of the other terms is the frequency shifted portions of $G_a(\Omega)/T$ when $|k| > 0$.
- The frequency range $-\frac{\Omega_T}{2} \le \Omega \le \frac{\Omega_T}{2}$ is called *baseband* or *Nyquist band*, which has the entire information of the signal
- One type of possible spectra $G_p(j\Omega)$ is shown below:

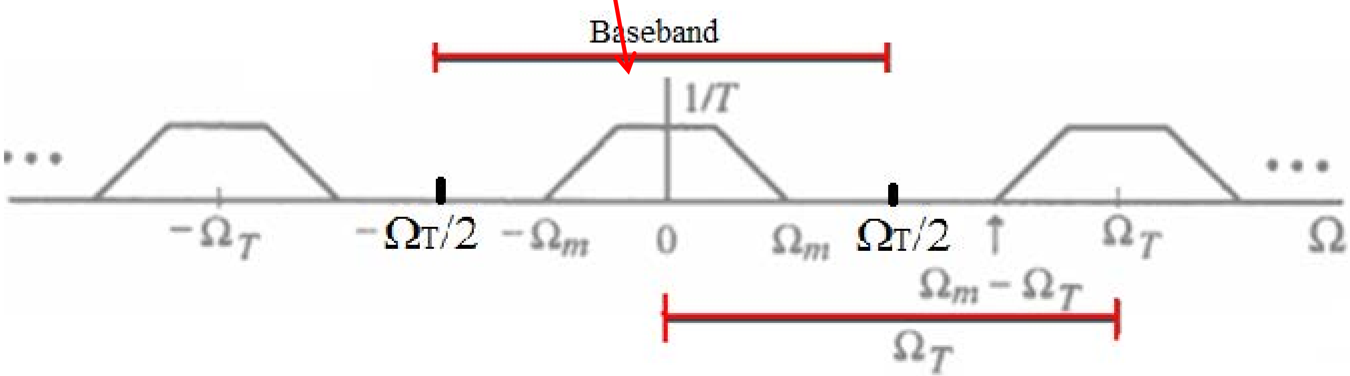


Figure 4 Spectrum of a digital signal

- It is evident that if $\Omega_T > 2\Omega_m$, where Ω_m is the maximum frequency in the signal, there is no overlap between the shifted replicas of $G_a(\Omega)/T$ generating $G_p(\Omega)$.

Effects of Sampling in the Frequency Domain

- On the other hand, as indicated by the figure below, if $\Omega_T < 2\Omega_m$, overlaps, known as ***aliasing effects***, between the shifted replicas of $G_a(\Omega)/T$ occur.

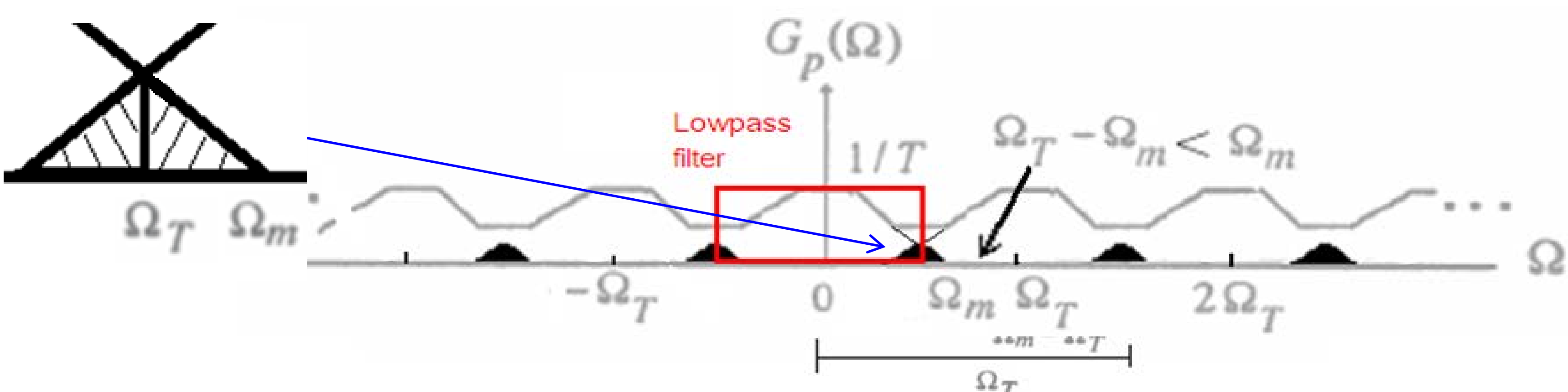
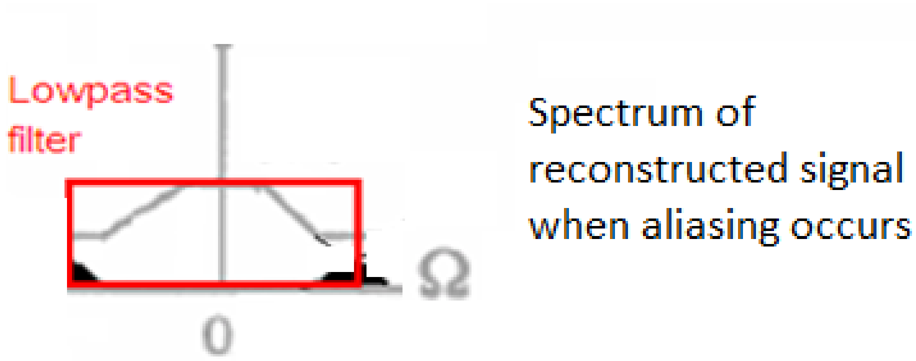


Figure 5 Aliased spectrum of a digital signal

- If $\Omega_T < 2\Omega_m$, due to the overlap of the shifted replicas, the spectrum $G_a(\Omega)$ cannot be recovered by filtering in DAC.
- This is because the distortion from a part of the replicas immediately outside the baseband folded back or aliased into the baseband (see the dark area).

Effects of Sampling in the Frequency Domain

- The portion within the low-pass bandwidth of the filter is the spectrum of the recovered signal.
- When aliasing occurs, the original signal cannot be recovered.

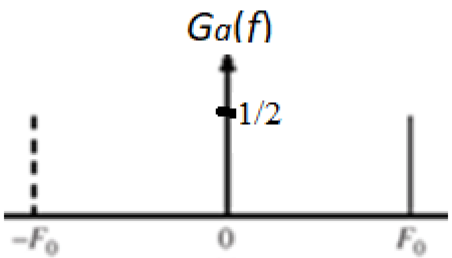


- The amount of distortion due to aliasing depends on the sampling frequency.

Effects of Sampling in the Frequency Domain

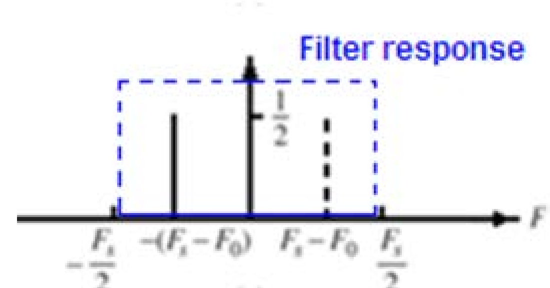
Example: Aliasing of sinusoidal signals

- Consider $x_a(t) = \cos(\omega_0 t)$, where $\omega_0 = 2\pi F_0$, which has spectrum lines (CTFT) at $F = \pm F_0$ (See Figure (a)).

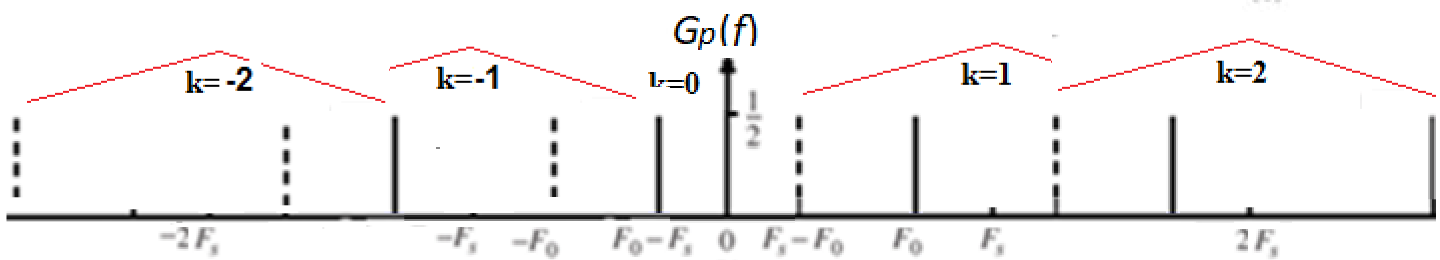


(a)

$$G_p(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a[(f - kF_s)]$$



(c)



(b)

- Figure (b) shows the line spectrum of the sampled signal $x(n)$ when $F_s/2 < F_0 < F_s$, i.e., $F_s < 2F_0$, where F_s is the sampling frequency
- Reconstruction is always based on the fundamental frequency range $|F| \leq F_s/2$ (See Figure (c)). Other frequencies are suppressed by the low pass filter.

Aliasing

- Therefore, the aliasing frequencies are used for signal reconstruction if sampling frequency is not sufficiently large (see $\pm(F_s\text{-}F_0)$, instead of $\pm F_{0,}$ is used in our previous example).

- **Demonstration**
  - We can observe the change of aliasing frequency when the sampling frequency is not sufficiently large.
  - It should be noted that the reconstructed output is not the original signal when the sampling rate does not satisfy the *Nyquist condition*, resulting in *aliasing*.
  - Therefore the sound of reconstructed signal changes with aliasing frequency (or sampling frequency). This should not happen since the same signal is used.
  - If $F_s > 2F_{max,}$ the sound is not changed with sampling frequency

- Speech signal without aliasing and Speech signal with aliasing.

Sampling Theorem

- **Sampling Theorem** - Let $g_a(t)$ be a bandlimited signal with CTFT $G_a(\Omega) = 0$ for $|\Omega| > \Omega_m$. Then $g_a(t)$ is uniquely determined by its samples $g_a(nT)$, $-\infty<n<\infty$ if $\Omega_T > 2\Omega_m$ where $\Omega_T = 2\pi/T$.
- The condition $\Omega_T > 2\Omega_m$ is often referred to as the ***Nyquist condition***
- The frequency $\Omega_T/2$ is usually referred to as the ***folding frequency***
- The highest frequency Ω_m, contained in $g_a(t)$ is usually called the ***Nyquist frequency*** since it determines the minimum sampling frequency $\Omega_T = 2\Omega_m$ that must be used to fully recover $g_a(t)$ from its samples
- The frequency $2\Omega_m$ is called the ***Nyquist rate***

Sampling Theorem

- **Oversampling** - The sampling frequency is higher than the Nyquist rate, which will be considered in latter sections
- **Undersampling** - The sampling frequency is lower than the Nyquist rate, which will be always avoided
- **Critical sampling** - The sampling frequency is equal to the Nyquist rate, which is not often used in practice
- **Application Examples :**
  - In digital telephony, a 3.4 *k*Hz signal bandwidth is acceptable for telephone conversation, and a sampling rate of 8 *k*Hz, greater than twice the signal bandwidth, is used.
  - In high-quality analog music signal processing, a bandwidth of 20 *k*Hz has been determined to preserve the fidelity. Hence, in compact disc (CD) music systems, a sampling rate of 44.1 *k*Hz, slightly higher than twice the signal bandwidth, is used.
  - The sampling frequency is often affected by system implementation limitations.

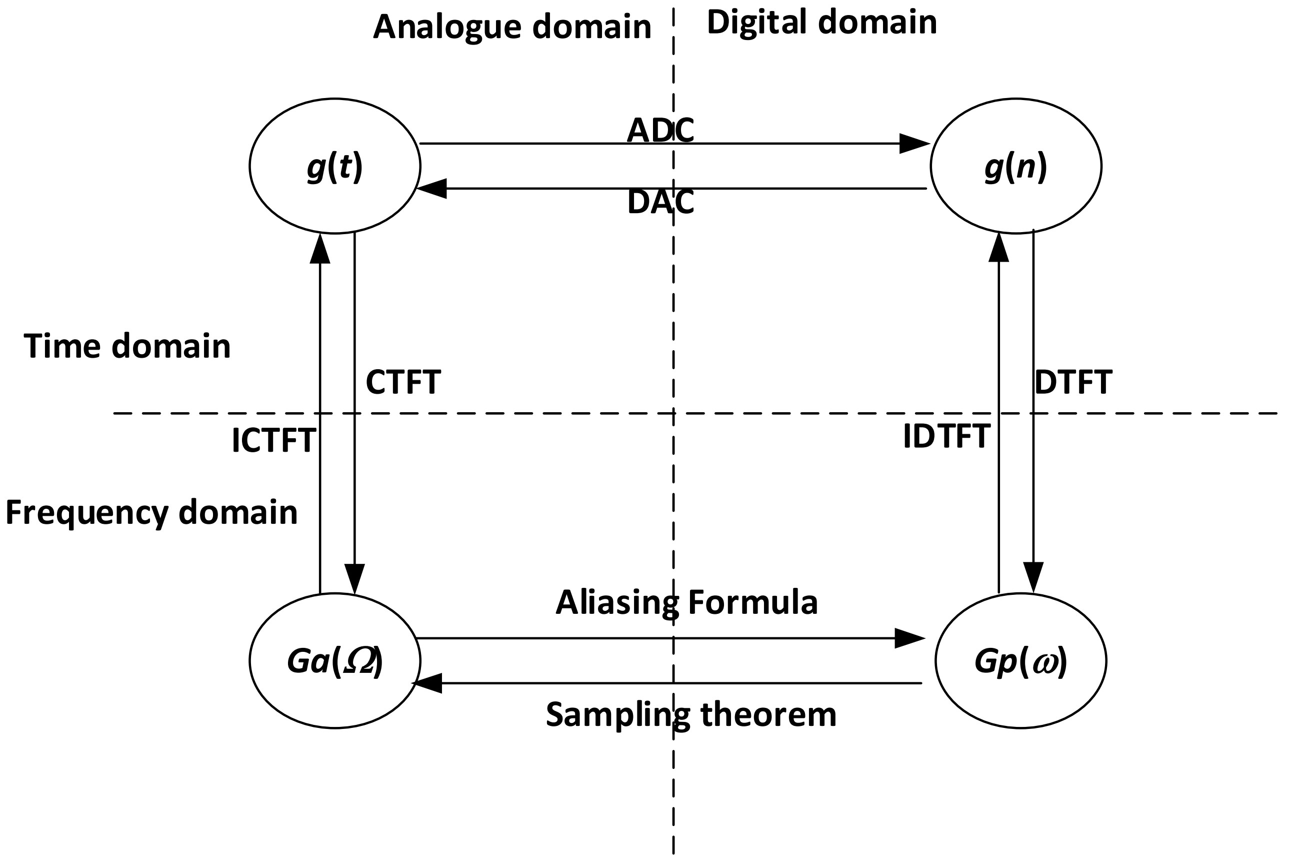
Sampling Theorem

- Understanding aliasing effects is very important for DSP studies because sampling frequency is directly related to the performance and computational cost.
- Without full understanding of this effect, it is difficult to understand the spectrum such as DFT of a digital sequence or a system transfer function.
- The tutorial questions provide very good materials to test you appreciation of the sampling process.
- Also see Section 6.1 of Proakis book for more information.

Main Points:

- Appreciate aliasing formula and the sampling theorem
- What is the condition and how is the original signal recovered from its digital samples?
- What are the aliasing phenomena for audio and image signals?

An important Figure



Signals in the Frequency Domain

What makes the difference between signals?

Demonstration 1

Demonstration 2

A Very Basic Question

- What are the most *basic elements* of various signals?
  - Not possible to describe all signals individually, and requiring a *unified approach.*
- One solution was given by Jean Baptiste Fourier (1768-1830).
  - He was the first one using the concept of GREEN HOUSE EFFECTS to describe earth warming phenomena
  - Study heat transmission model and the temperature of the globe
- He concluded that **all signals can be generated by combining sinusoidal waves with different frequencies and amplitudes**

DTFT - Representation in Frequency Domain

- The DTFT is generally used to observe the frequency content of signal $x(n)$.

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

- The signal $x(n)$ can be recovered from $X(\omega)$ by

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$$

Properties:

- **Periodic properties (a basic result of Aliasing formula):** We always observe one period of the spectrum, i.e., from 0 to F_T Hz, where F_T is the sampling frequency.

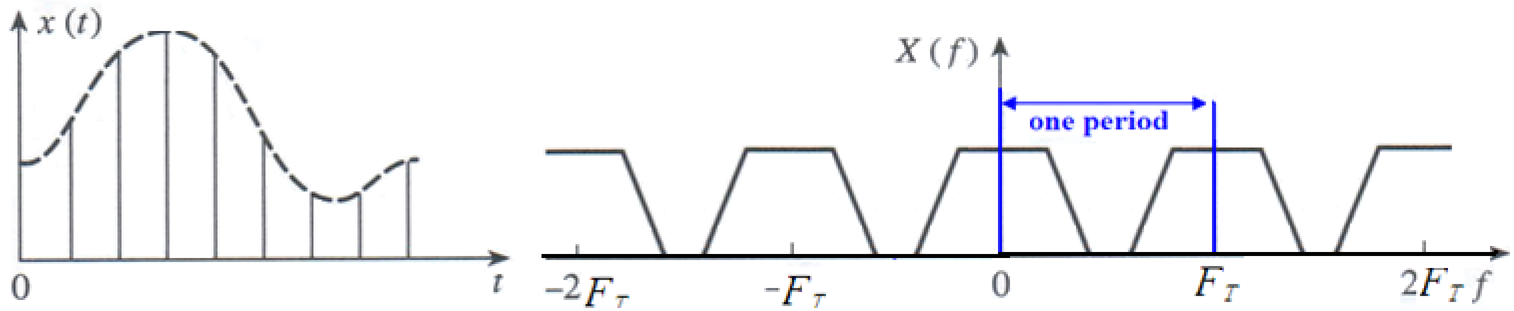


Figure 6 Periodic property of the DTFT

DTFT - Representation in Frequency Domain

Symmetric Properties:

Any real signal $x(n)$	$X(\omega) = X^*(-\omega)$	$X_R(\omega) = X_R(-\omega)$ $X_I(\omega) = -X_I(-\omega)$
	$\lvert X(\omega)\rvert = \lvert X(-\omega)\rvert$	$\measuredangle X(\omega) = -\measuredangle X(-\omega)$

- We often use only the first half of the spectrum

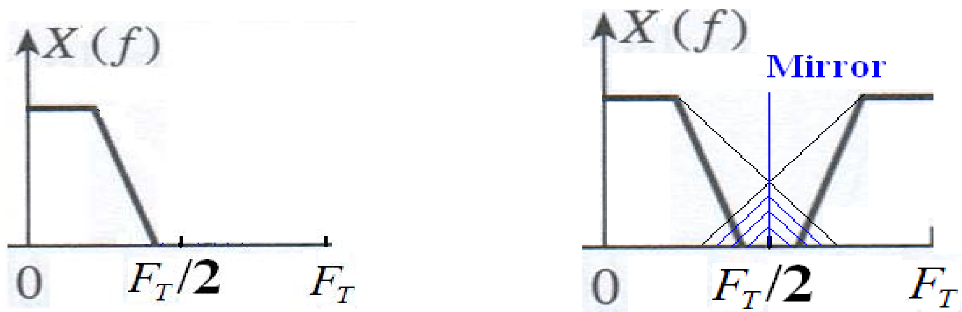


Figure 7 Symmetric property of the DTFT of a real signal

- The spectrum is often given in terms of frequency $0 \le f < F_T$ and radian frequency $0 \le \omega < \omega_T = 2\pi F_T$. Other parts of spectrum are the repetitions of this period

- Or normalized frequency $0 \le f / F_T < 1.0$ and normalized radian frequency $0 \le \omega < 2\pi$.

DTFT - Representation in Frequency Domain

- **Example :** Calculate the impulse response of the ideal low-pass filter defined by

$$X(\omega) = \begin{cases} 1 & |\omega| \le \omega_c < \pi \\ 0 & \text{Otherwise} \end{cases}$$

$$h(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} \frac{de^{j\omega n}}{jn}$$

$$= \frac{1}{j2\pi n} e^{j\omega n}\Big|_{-\omega_c}^{\omega_c}$$

$$= \frac{\sin n\omega_c}{\pi n}$$

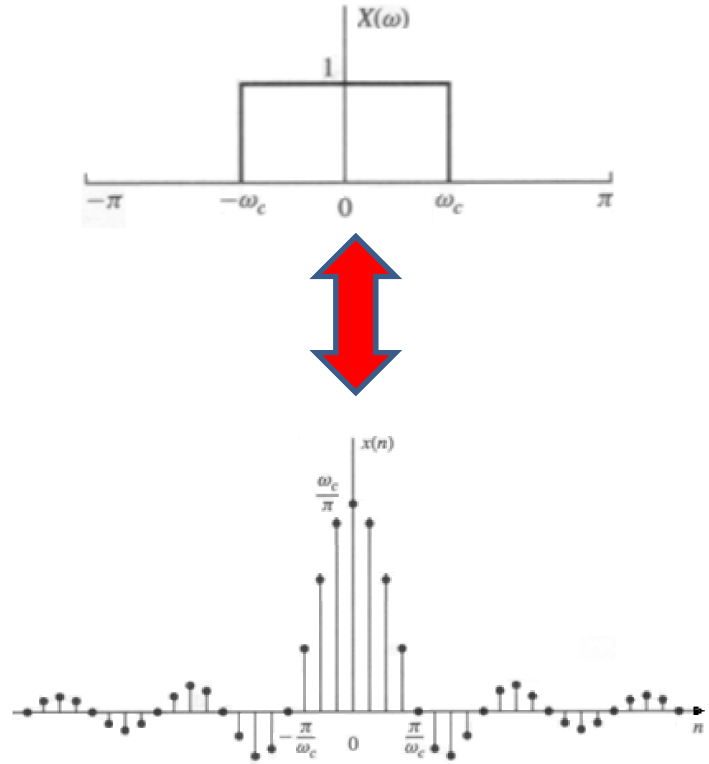


Figure 8 a rectangular pulse and its spectrum

- This response is not causal and not realizable (exists in theory only)

- A system is said to be ***causal*** if the present output depends only on the present and past values of the input. For example, $x[n+1]$ is the signal in the future and $x[n-1]$ is in the past, then

$$y[n] = x[n] + x[n-1] \quad \text{(causal)}$$

$$\text{or} \quad y[n] = x[n] + x[n+1] \text{ (noncausal)}$$

- Measures have to be taken to change the casualty of the system. For example, to add an additional phase in the frequency domain, or delay in time domain to the system impulse response.

Example 2

- Calculate the impulse response of the ideal low-pass filter defined by

$$X(\omega) = \begin{cases} e^{-j\omega N/2} & |\omega| \le \omega_c < \pi \\ 0 & \text{Otherwise} \end{cases}$$

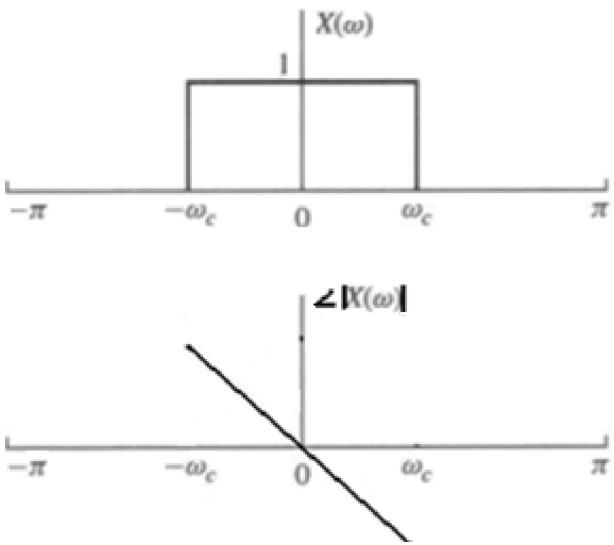


Figure 9 A spectrum with a phase

Solution:

$$h(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-j\omega N/2} e^{j\omega n} d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} \frac{de^{j\omega(n-N/2)}}{j(n-N/2)}$$

$$= \frac{1}{j2\pi(n-N/2)} e^{j\omega(n-N/2)} \Big|_{-\omega_c}^{\omega_c}$$

$$= \frac{\sin(n-N/2)\omega_c}{\pi(n-N/2)}$$

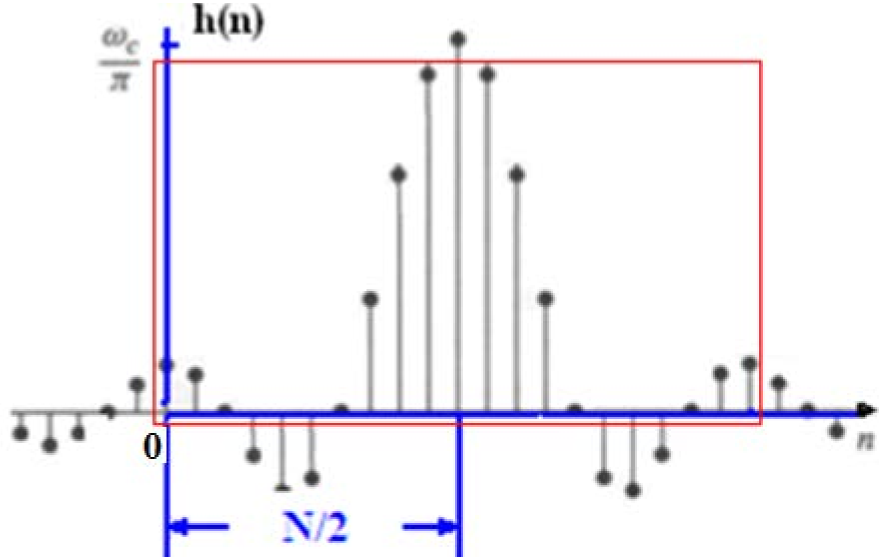


Figure 10 a delayed impulse response

- To be causal and realizable, $h(n)$ has to be within *a finite duration* and shifted by one half of the sequence length (see the response inside the box in the above figure)
- The *finite length* of the response will lead to
  - – Ripples in both pass and stop bands
  - – A transitional band between the stop and pass bands.

Discrete Fourier Transform

- Discrete Fourier transform (DFT) is for a periodic *N*-point input sequence, defined as

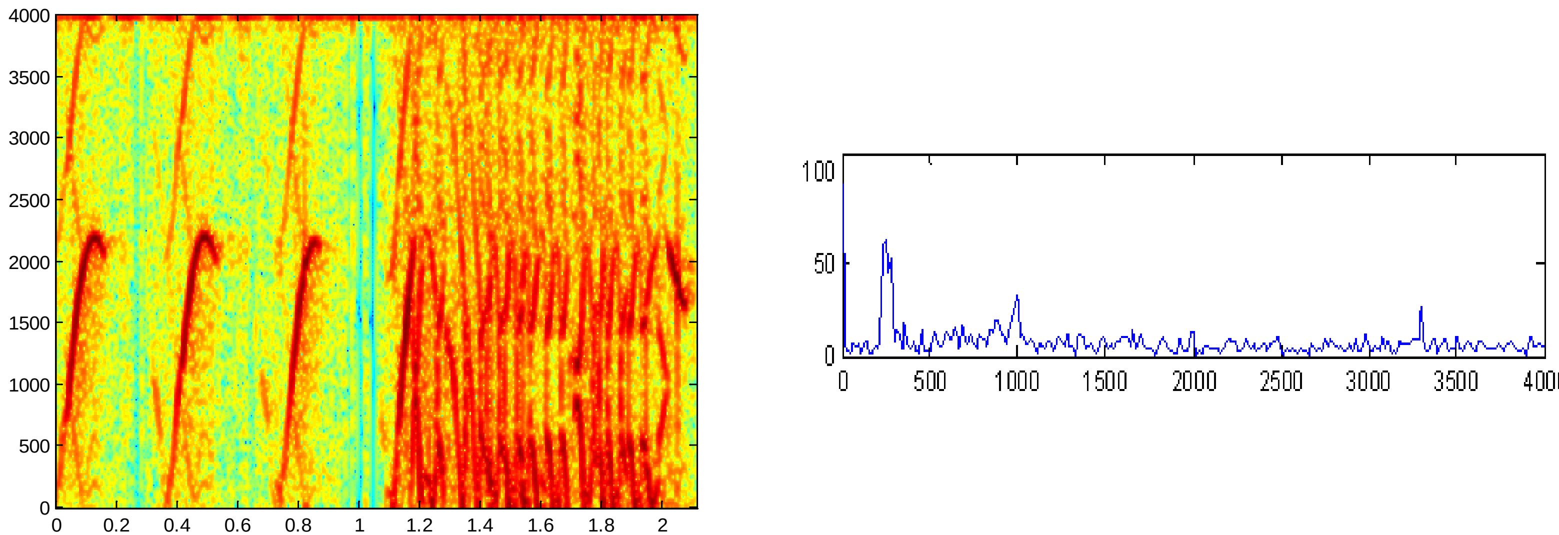
$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad 0 \le k < N$$

$$x(n) = \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N}, \quad 0 \le n < N$$

- $X(k)$ is discrete and is considered as the sampling of the DTFT in the frequency domain.
- Therefore, the resolution of the DFT is F_T/N (in *Hz*) or ($2\pi/N$ in radian), where F_T is the sampling frequency of the signal.
- For example, if a signal is sampled with 8 *k*Hz, a point of a 1024 point DFT represents a *resolution* of 8000/(1024-1) Hz.
- DFT is often used for practical computation with its fast algorithms (FFT). In contrast, DTFT is more for theoretical presentation.

Time-Frequency Representation

- The DTFT reveals the frequency contents of a signal segment only, but fails to tell when and which frequency starts and ends for many natural signals having time varying frequencies, such as voice, music, etc..
- See the spectrogram (left) and DFT (right) of bird sounds



STFT DFT

Figure 11 The time-frequency representation

Time-Frequency Representation

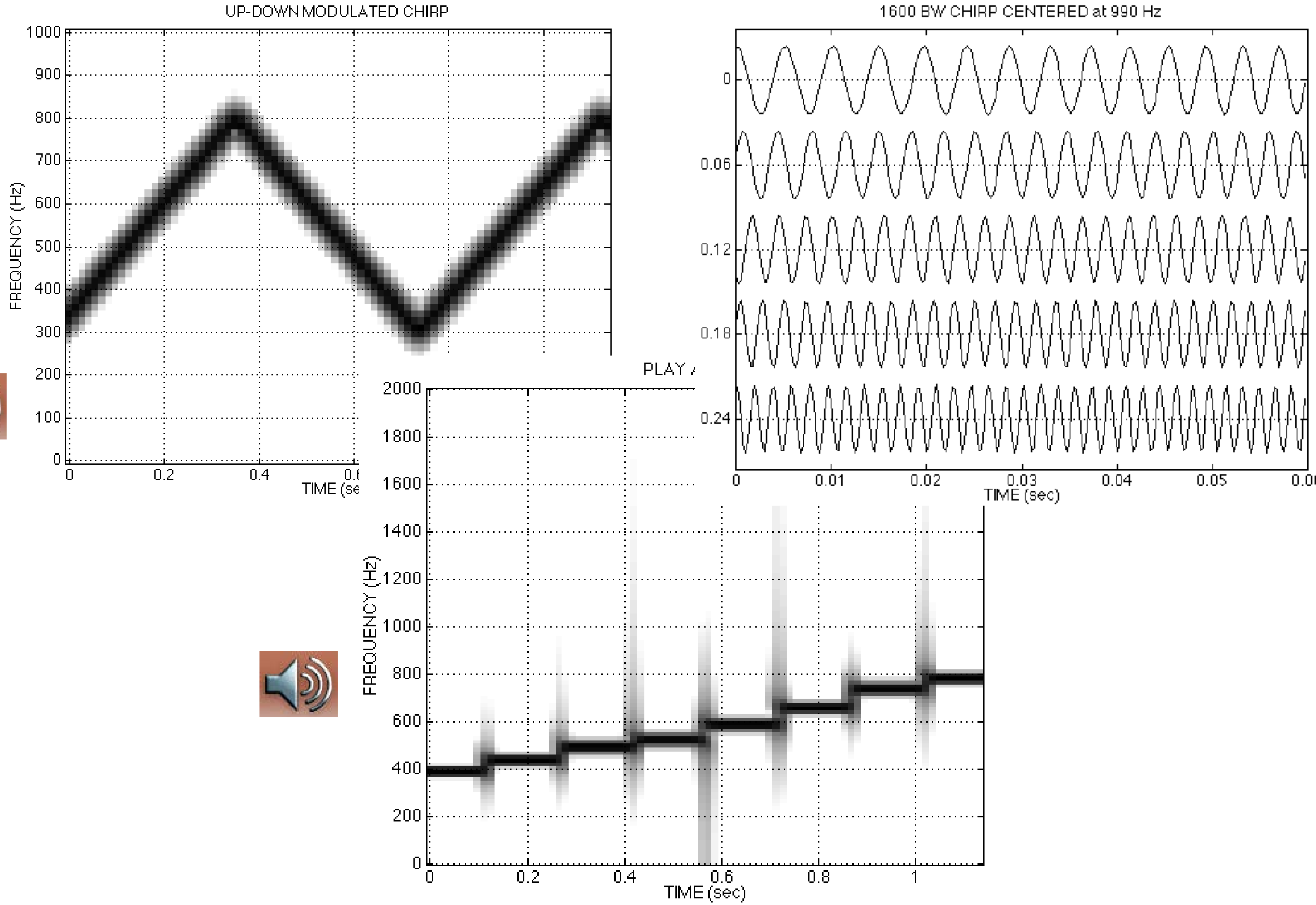


Figure 12 Time-frequency representation of chirp and piano signals

Discrete-Time Short-Time Fourier Transform

- Short-time discrete-time Fourier transform (STFT) is a simple method to reveal *time-varying* characteristics of signal spectrum in time-frequency domain.

Segmented input

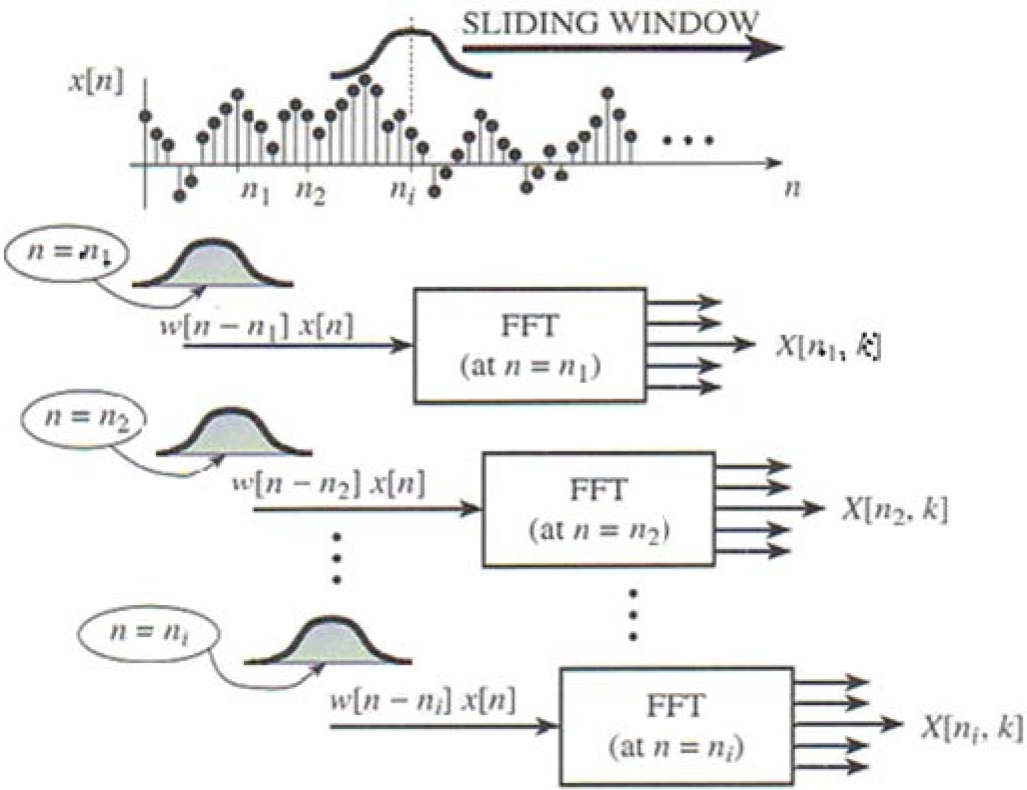
- The spectrum for segment n_0 is expressed as

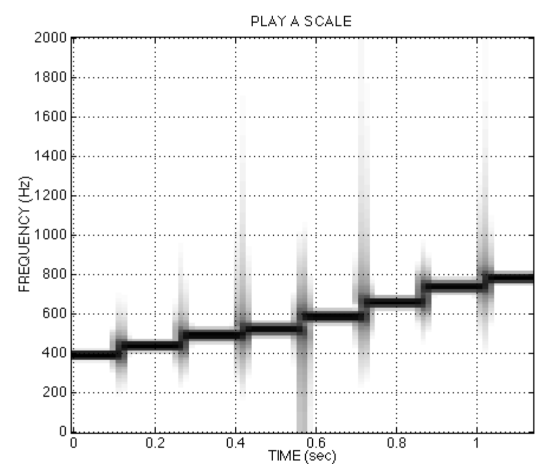
$$STFT(x, n_0, k) = \sum_{m=n_0-N+1}^{n_0} x[m]w[m-n_0]e^{-j(2\pi/N)km}$$

where $w[n]$ is an N-point window function to divide the input to be N-point segments.

- The STFT is a sequence of point-N DFTs. The spectrogram of $x(n)$ is defined as $|STFT(x, n_0, k)|$ or log $|STFT(x, n_0, k)|$, which is a function of n_0 and k.

Discrete-Time Short-Time Fourier Transform





- Example 1: Piano sounds  
- Example 2: Sound monitoring  

Fast Fourier Transform (FFT)

- In practice, fast Fourier transform (FFT) is widely used for the computation of the DFT. An *N*-point DFT is defined as

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk}, \qquad W_N = e^{-j2\pi/N}$$

- FFT is based on the symmetric and periodic properties of the transform kernel W_N. For example,

$$X(k)=\sum_{n=0}^{N/2-1} x(n)W_N^{nk} + \sum_{n=0}^{N/2-1} x(\frac{N}{2}+n)W_N^{(N/2+n)k}$$

$$=\sum_{n=0}^{N/2-1}\left[x(n)+W_N^{(N/2)k}x(\frac{N}{2}+n)\right]W_N^{nk} = \sum_{n=0}^{N/2-1}\left[x(n)+(-1)^k x(\frac{N}{2}+n)\right]W_N^{nk}$$

- Then, we have two *N*/2-point DFTs for *k* = 0, 1, …, *N*/2

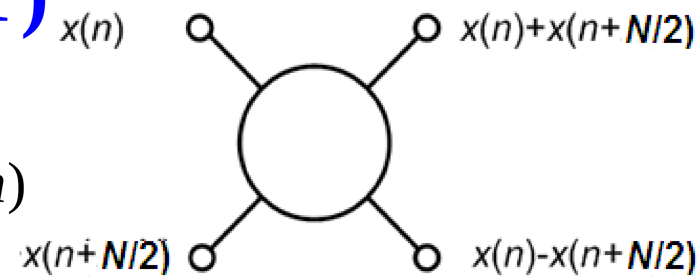
$$X(2k)=\sum_{n=0}^{N/2-1}\left[x(n)+x(N/2+n)\right]W_{N/2}^{nk},$$

$$X(2k+1)=\sum_{n=0}^{N/2-1}\left\{\left[x(n)-x(N/2+n)\right]W_N^{n}\right\}W_{N/2}^{nk},$$

Fast Fourier Transform (FFT)

- Butterfly computation is

$$x(n) + x(N/2+n), \qquad x(n) - x(N/2+n)$$



- Repeat the above decomposition steps, each $N/2$-point DFT can be decomposed into another two $N/4$-point DFTs.

- Such procedures can be repeated until 2-point DFTs are reached, which is again a butterfly computation (see the fig).

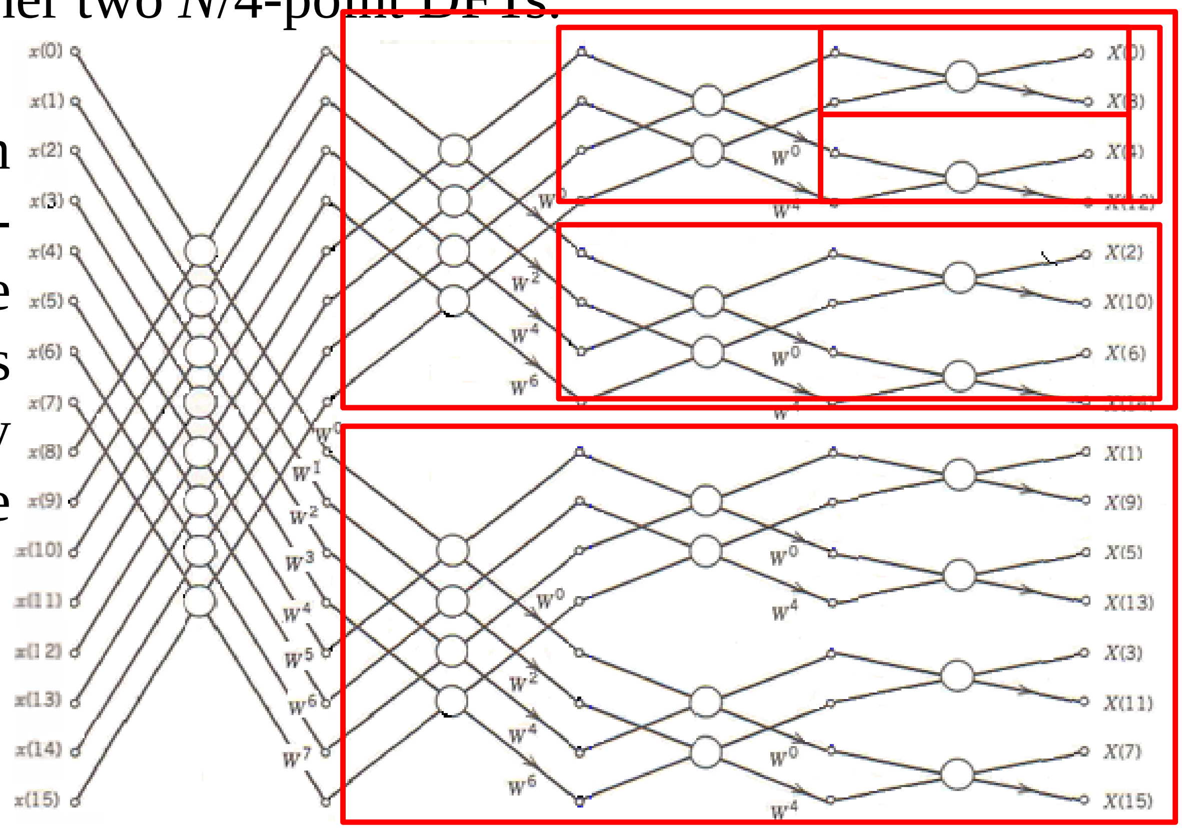


Figure 13 A 16-point radix-2 DIF FFT diagram

Fast Fourier Transform (FFT)

- In general, there will be $(N/2)\log_2 N$ butterflies for a N-point DFT
- Each butterfly needs two complex additions and one complex multiplication.
- In total, an N-point DFT needs $N\log_2 N$ complex additions and $(N/2)\log_2 N$ complex multiplications
- A direct computation of an N-point DFT needs N^2 complex multiplications and $N(N-1)$ complex additions
- $2N/\log_2 N$ times of complex multiplications saving by the FFT. For example, 204.8 times saving when N =1024.
- Each complex multiplication need 4 real multiplications and 2 real additions
- Radix-2 algorithm requires the size of N be a power of two.
- Many other types of the fast algorithms which can deal with N being a composite number.

Application of FFT

- One typical application of the FFT is filtering based on the property that the DFT of the convolution outputs between two input sequences in the time domain is equivalent to the product of the DFTs of the two sequences in the frequency domain.
- The definition of a finite impulse response (FIR) filter is

$$y[n] = \sum_{k=0}^{N-1} x[k]h[n-k] = \sum_{k=0}^{N-1} x[n-k]h[k]$$

  where $x[n]$ is the input signal and $h[n]$ is the filter impulse response.
- When N is large, for example, N > 100, the computational complexity is costly
- **Example:** Input sampling frequency = 20 *k*H of an audio signal, and N = 100, the number of multiplications is 20×100/2 =1,000,000 multiplications, where the factor 2 is due to filter symmetry property.

Application of FFT

- Based on the property that the DFT of the convolution outputs between two sequences in the time domain is equivalent to the product of the DFTs of these sequences in the frequency domain.
- For example, a filter can be implemented as in the figure below.

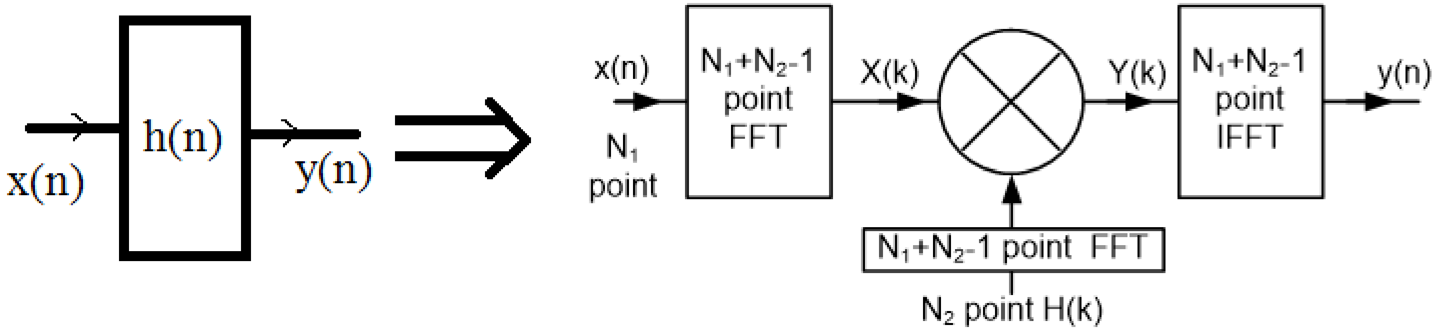


Figure 14 Block diagram of convolution based FFT.

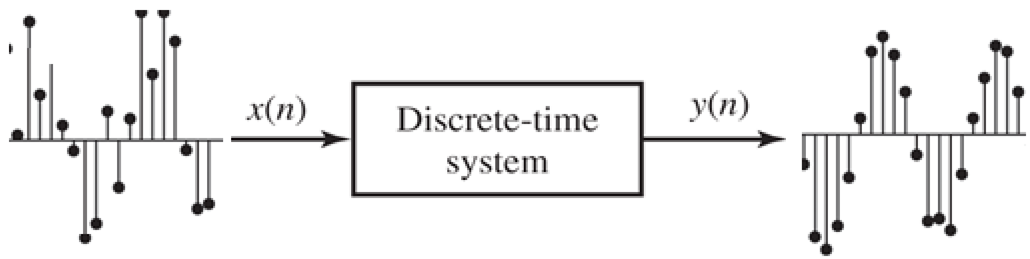
- Assuming N_1 point input sequence and N_2 point filter coefficients. The computation steps using the FFT are as follows:
  - Forming N_1+N_2-1 input sequence and coefficient sequence by ***padding zeros***
  - Computing the DFT of the N_1+N_2-1 input sequence $X(k)$ by using the FFT. The N_1+N_2-1 DFT of the coefficient sequence $H(k)$ can be pre-calculated.
  - Point-by-point multiplying the two DFTs to obtain $Y(k)$
  - Compute the N_1+N_2-1 point inverse FFT of $Y(k)$ to obtain $y(n)$.
  - More information on page 505 section 8.2.3 of Proakis book.

Summary for Lecture 1

- We are only able to review the basics of the DTFT and DFT. If necessary, self study on the text book is necessary.
- Sufficient understanding on signal representation in the frequency domain
- Able to use the DTFT and DFT to represent signals
- Be familiar with the properties of the DTFT and DFT
- Basic understanding on time-frequency representation STFT and FFT.

- CFT – continuous Fourier transform, for analogue signal, continuous in both time and frequency domain
- DTFT – discrete time Fourier transform, for digital signal, discrete in time domain and continuous in frequency domain
- DFT – discrete Fourier transform, for digital signal, discrete in both time and frequency domain.
- STFT – short time Fourier transform, for long signal with a window, discrete in both time and frequency domain.
- The use of radian freq. (radian) and freq. (Hz) in the frequency domain $\omega=2\pi f$ for analogue signal and normalised radians freq. and freq. for digital signal $\omega=2\pi f/F_T$

The Processing Systems of Digital Signals and Their Classifications



LTI Discrete-Time Systems

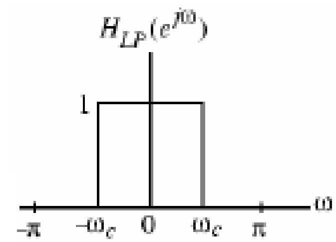
- It is important to mathematically describe the system that deals with the signals.

Types of Transfer Functions

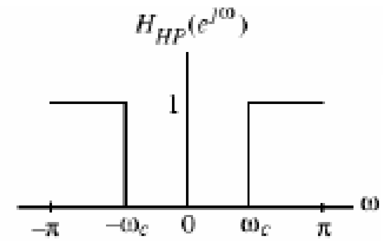
- Based on the length of its impulse response $h(n)$, a *linear time invariant* (LTI) digital transfer function is generally classified into:
  - Finite impulse response (FIR) transfer function, which is *nonrecursive*
  - Infinite impulse response (IIR) transfer function, which is *recursive*
- Based on digital transfer functions with frequency-selective responses, we have
  - Classification based on the shape of the magnitude function $|H(e^{j\omega})|$
  - Classification based on the form of the phase function $\theta(\omega)$

Classification Based on Mag. Characteristics

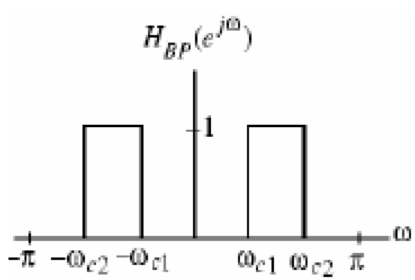
- A digital filter passing certain frequencies without distortion has a frequency response **equal to one** at these frequencies, and a frequency response **equal to zero** at all other frequencies
- The range of frequencies, where the frequency response takes the value of *one*, is called the ***passband***
- The range of frequencies, where the frequency response takes the value of *zero*, is called the ***stopband***
- Frequency responses of the four popular types of ideal digital filters with real impulse response are shown in these figures



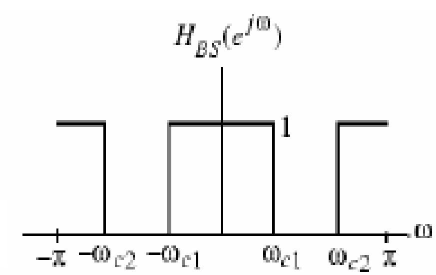
Ideal lowpass
Passband
$0 \leq \omega \leq \omega_c$
Stopband
$\omega_c \leq \omega \leq \pi$



Ideal highpass
Passband
$\omega_c \leq \omega \leq \pi$
Stopband
$0 \leq \omega \leq \omega_c$



Ideal bandpass
Passband
$\omega_{c1} \leq \omega \leq \omega_{c2}$
Stopband
Elsewhere



Ideal Bandstop
Passband
Elsewhere
Stopband
$\omega_{c1} \leq \omega \leq \omega_{c2}$

Figure 15 Frequency response of four types of filters

Classification Based on Mag. Characteristics

- The inverse DTFT of these characteristics is generally called the *impulse response*, $h(n)$, which is a time domain function.
- The frequencies ω_c, ω_{c1}, and ω_{c2} are called the *cutoff frequencies*
- These *ideal* filters have a magnitude response equal to one in the passband and zero in the stopband, and have a **zero phase** everywhere (or no delay to produce the output).
- The ideal filters with the "**brick wall**" frequency responses are **not causal** and cannot be realized with finite length LTI filter.
- To develop stable and realizable transfer functions, the ideal frequency response specifications are relaxed by including a *transition band* between the passband and the stopband

Classification Based on Mag Characteristics

- This approximation permits the magnitude response to decay slowly from its maximum value in the passband to the zero value in the stopband.
- The magnitude responses also have a small deviation from the ideal values both in passband and stopband
- Typical magnitude response specifications of a lowpass filter are shown in the Figure.
- The sources of these variations?

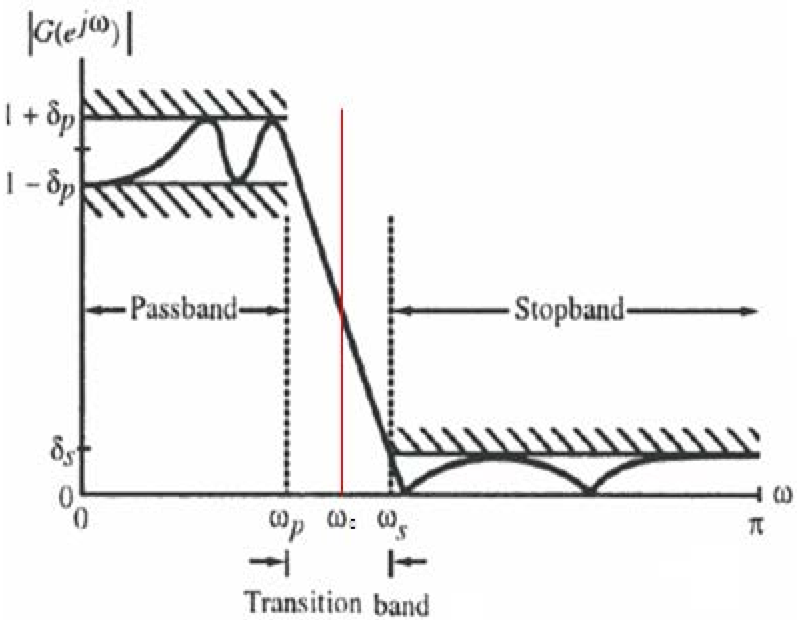


Figure 16 A typical lowpass filter magnitude frequency response

Classification Based on Phase Characteristics

- **Example:** Determine the impulse response of an ideal lowpass filter with a linear phase response (see the Figure in the next page):

$$H_{LP}(e^{j\omega}) = \begin{cases} e^{-j\omega n_0}, & 0 \le |\omega| < \omega_c \\ 0 & \omega_c \le |\omega| < \pi \end{cases}$$

- Applying the frequency shift property of the DTFT to the impulse response of an ideal zero-phase lowpass filter we arrive at

$$h_{LP}(n) = \frac{\sin \omega_c (n - n_0)}{\pi (n - n_0)} \qquad -\infty < n < \infty$$

- As before, the above filter is *noncausal* and of doubly infinite length, and hence, *unealizable*.
- By truncating the impulse response to a finite number of terms, a realizable FIR approximation to the ideal lowpass filter can be developed
- The truncated approximation may or may not exhibit linear phase, depending on the value of n_0 chosen

Classification Based on Phase Characteristics

- It is desired to pass input signal components in a certain frequency range undistorted in both magnitude and phase,
- The transfer function should exhibit a unity magnitude response and a linear-phase response in the band of interest, as shown in the figure.
- Since the signal components in the stopband are blocked, the phase response in the stopband can be of any shape.
- What is the time domain expression $h_{LP}(n)$ for $H_{LP}(e^{j\omega})$?

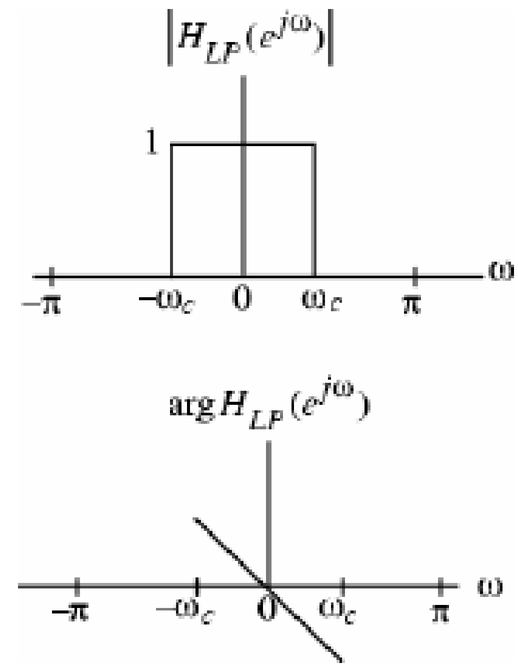
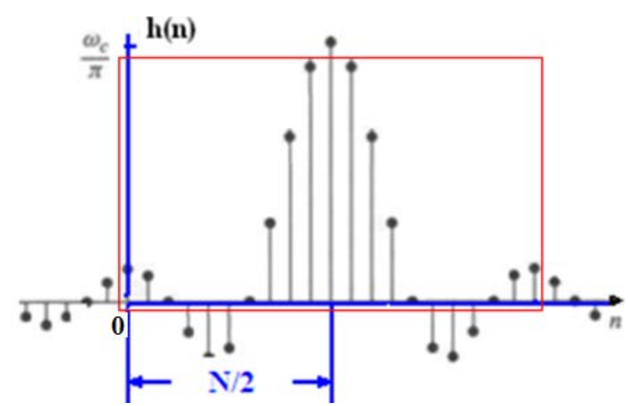


Figure 17 Ideal lowpass filter frequency responses



Classification Based on Phase Characteristics

- If we choose $n_0 = N/2$ with N a positive integer, the truncated and shifted approximation

$$\hat{h}_{LP}(n) = \frac{\sin \omega_c (n - N/2)}{\pi (n - N/2)} \qquad 0 \le n < N$$

will be a length N causal linear phase FIR filter

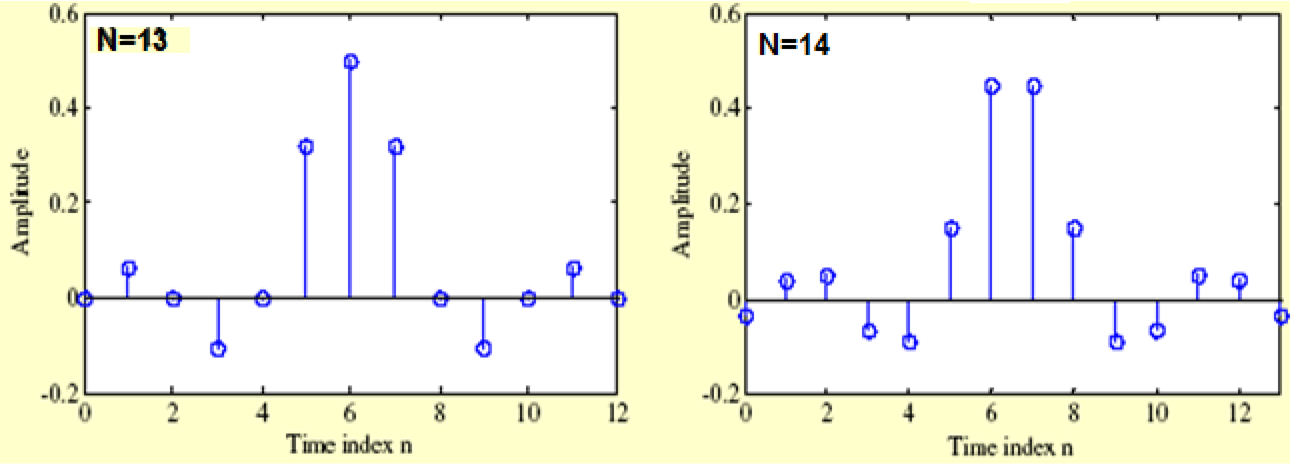


Figure 18 Examples of finite lowpass filter impulse responses

- It can be easily understood that the above specifications *approximate* the transfer function of $H_{LP}(e^{j\omega})$ with a linear phase.

Some Simple Filter Examples

Simple Digital Filters

- Let us consider a low-order digital filter with reasonable selective frequency responses that are satisfactorily used in practical applications
- Latter we shall review various methods of designing frequency-selective filters satisfying the prescribed specifications
- The digital filter considered here has *integer-valued* impulse response coefficients
- The main emphasis here is to familiarize the *way of analyzing them* rather than their applications
- Some parameter definitions that are common to all filters should be fully *understood* and *remembered*

Lowpass FIR Digital Filters

- Moving average filtering is the oldest filtering technique

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k]$$

- Used as indices in stock market, weather observation, air pollution index…
- The simplest lowpass FIR digital filter is the 2-point moving-average filter given by

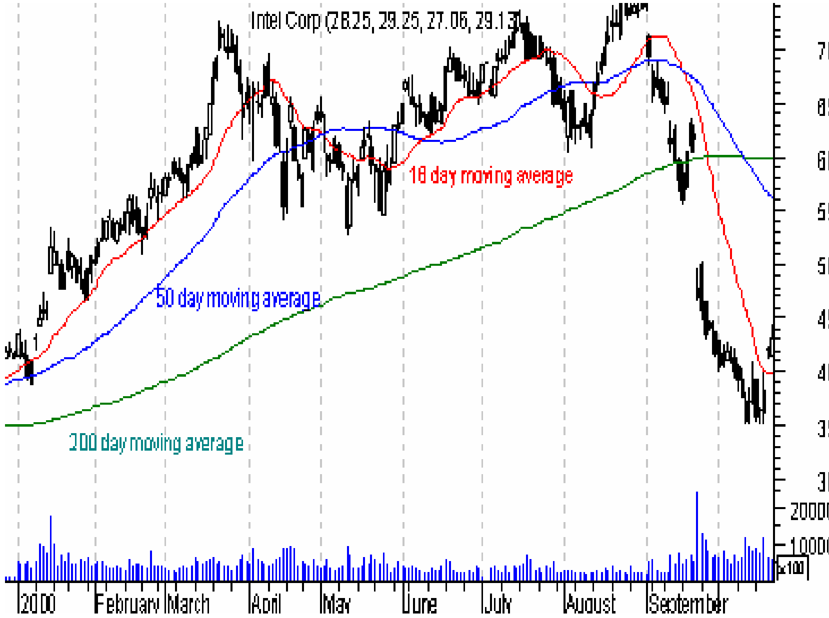


Figure 18 Example of using moving average filter for different values of *N*.

$$y[n] = \frac{1}{2}\{x[n] + x[n-1]\}$$

$$H_0(e^{j\omega}) = H_0(z)|_{z=e^{j\omega}} = e^{-j\omega/2}\cos(\omega/2)$$

$$H_0(z) = \frac{1+z^{-1}}{2} = \frac{z+1}{2z}$$

$$|H_0(e^{j0})| = 1, \quad |H_0(e^{j\pi})| = 0$$

Simple Digital Filters

- The transfer function has a pole at z = 0 (location of the max. value) and a zero at z = -1 (location of the min. value)
- The maximum value of the magnitude function is 1 at ω = 0, and the minimum value is 0 at $\omega = \pi$ (therefore a lowpass filter).
- The frequency response of the above filter is shown in the figure.
- The magnitude response $|H(e^{j\omega})| = \cos(\omega/2)$ can be seen to be a monotonically decreasing function of ω.

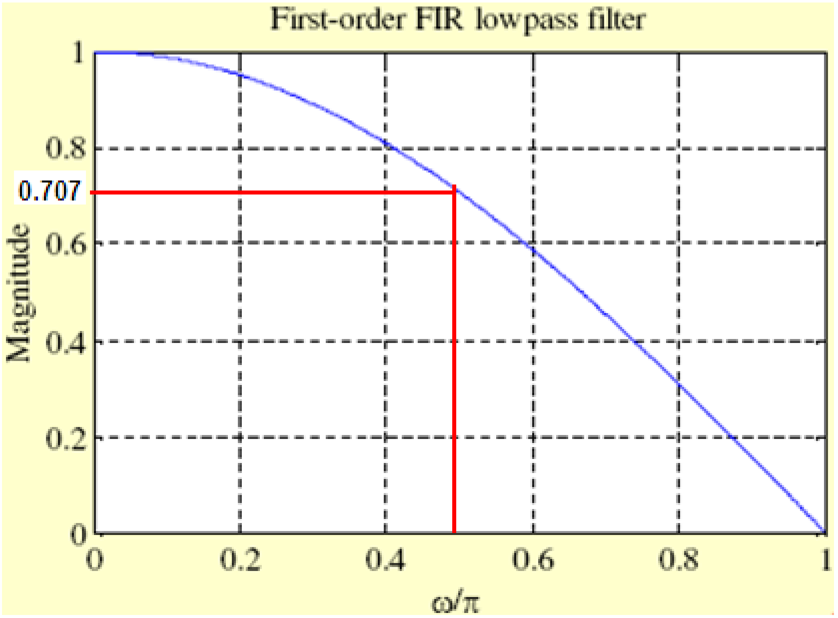


Figure 19 Frequency response of a simple filter

Simple Digital Filters

- The frequency $\omega = \omega_c$, at which
$$|H_0(e^{j\omega_c})| = \frac{1}{\sqrt{2}}|H_0(e^{j0})|$$
is of practical interest here. The gain $G(\omega_c)$ in dB is given by
$$G(\omega_c) = 20\log_{10}|H(e^{j\omega_c})|$$
$$= 20\log_{10}\left\{\frac{|H(e^{j0})|}{\sqrt{2}}\right\} = -20\log_{10}\sqrt{2} \cong -3\,dB$$
- Thus, the gain $G(\omega_c)$ at $\omega = \omega_c$ is approximately 3 dB less than the gain at $\omega = 0$.
- As a result, ω_c is called the *3-dB cut-off frequency*.
- To determine the value of ω_c we set $|H_0(e^{j\omega_c})|^2 = \cos^2(\omega_c/2) = \frac{1}{2}$ which yields $\omega_c = \pi/2$ for this case
- The moving average of order N has a system transfer function
$$H(z) = \frac{1}{N+1}\frac{1-z^{-(N+1)}}{1-z^{-1}}$$

How to Implement the Filters?

Structures of Filters

Digital Filter Structures

- In general, an infinite impulse response (IIR) finite-duration system is expressed as

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

which has an impulse response of *infinitive* length.

- A finite impulse response (FIR) filter is the special case considered as the convolution sum of a *linear time invariant* (LTI) discrete-time system

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k]$$

The actual implementation of an LTI digital filter can be either in software or hardware form, depending on applications

Digital Filter Structures

- One practical problem is that in either case, the signal samples and the filter coefficients cannot be represented with *infinite* precision due to the finite word length effects
- A direct implementation of a digital filter sum may not provide satisfactory performance due to the **finite precision** arithmetic
- It is of practical interest to develop alternate realizations and choose the structure that provides satisfactory performance under finite precision arithmetic
- A structural representation using interconnected basic building blocks is the *first step* in the hardware or software implementation of an LTI digital filter
- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid *computational algorithm*

Filter Structures – Basic Building Blocks

- The computational algorithm of an LTI digital filter can be conveniently represented in signal block diagram form using the basic building blocks shown below

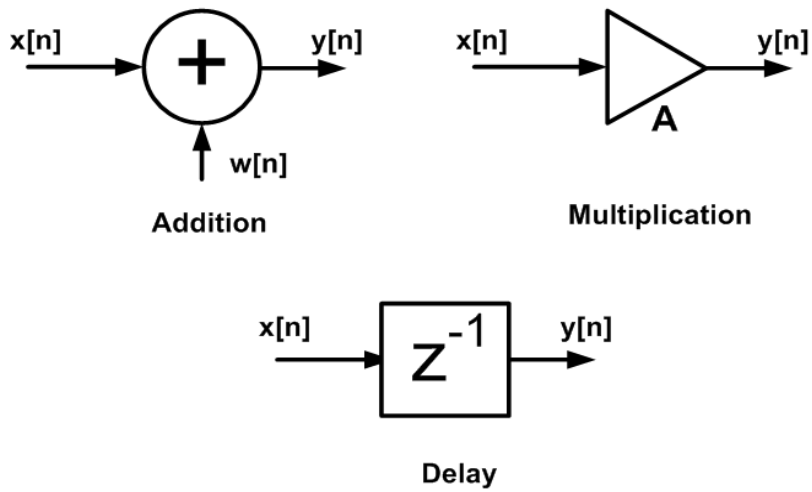


Figure 19 Basic building blocks for digital systems

- Advantages of block diagram representation
  - Easy to *write down* the computational algorithm by inspection
  - Easy to *analyze* the block diagram to determine the explicit relation between the output and input

Filter Structures – Analysis of Block Diagrams

– Easy to *manipulate* and derive other *equivalent* block diagrams yielding different computational algorithm

– Easy to *determine* the hardware requirements

– Easier to *develop* block diagram representations from the transfer function directly

Steps:

- Carried out by writing down the expressions for the output signal of each *adder* as a sum of its input signals,
- Developing a set of equations relating the filter input and output signals in terms of all internal signals
- Eliminating the unwanted internal variables and obtaining the expression for the output signal as a function of the input signal and the filter parameters that are the multiplier coefficients

Filter Structures - Analysis of Block Diagrams

- **Example:** Analyze the cascaded lattice structure below where the *z*-dependence of signal variables are not shown for brevity

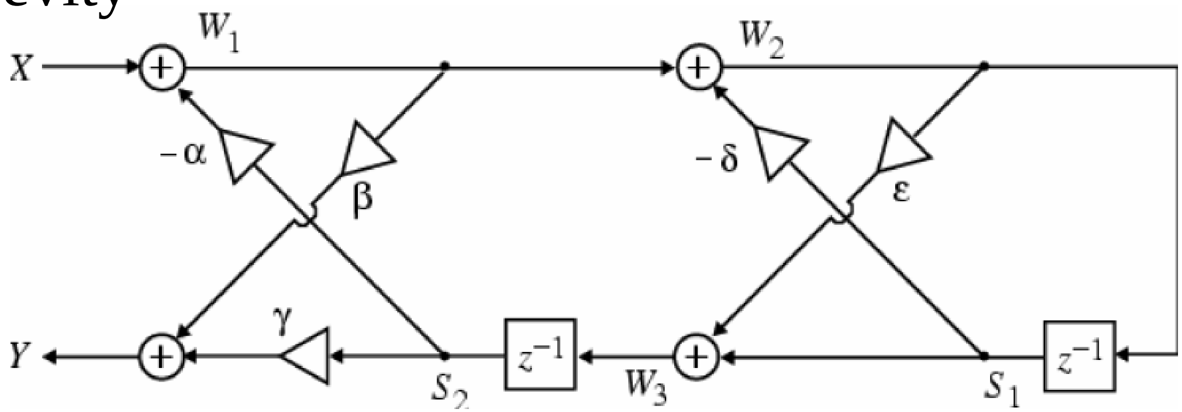


Figure 20 The cascaded structure of a lattice filter

- The output signals of the four adders are

$$W_1 = X - \alpha S_2, \qquad W_2 = W_1 - \delta S_1$$

$$W_3 = S_1 + \varepsilon W_2 \qquad Y = \beta W_1 + \gamma S_2$$

- From the figure, we observe

$$S_1 = z^{-1} W_2, \qquad S_2 = z^{-1} W_3$$

Filter Structures - Analysis of Block Diagrams

- Substituting the last two relations in first 4 equations, we get

$$W_1 = X - \alpha z^{-1} W_3, \qquad W_2 = W_1 - \delta z^{-1} W_2$$

$$W_3 = z^{-1} W_2 + \varepsilon W_2 \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

- From the second equation, we get $W_2 = W_1 / (1 + \delta z^{-1})$
- From the third equation we get $W_3 = (\varepsilon + z^{-1}) W_2$
- Combining the last two equations, we have

$$W_3 = \frac{(\varepsilon + z^{-1})}{1 + \delta z^{-1}} W_1$$

- Substituting the above equation in

$$W_1 = X - \alpha z^{-1} W_3, \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

- We finally arrive at:

$$H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon) z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon) z^{-1} + a z^{-2}}$$

Filter Structures – Canonic Structures

- A digital filter structure is ***canonic*** if the number of delays in the block diagram representation is equal to the **order** of the transfer function, otherwise, it is a noncanonic structure

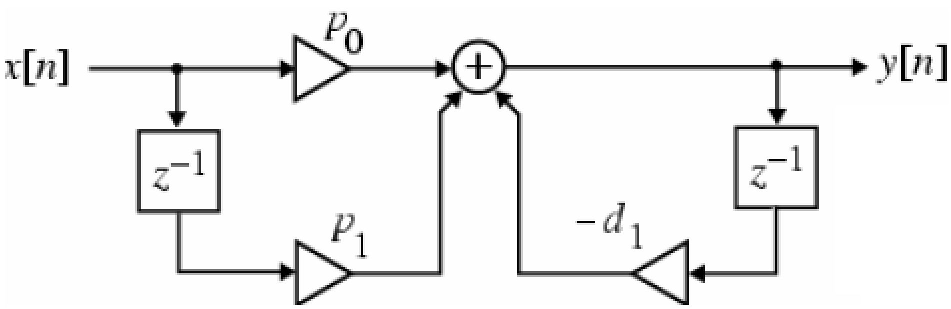


Figure 21 A non-canonic structure

- The structure shown above is ***noncanonic*** as it uses *two* delays to realize the *first* order difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

- It is often that a canonic structure requires the minimum cost of implementation.

Equivalent FIR Filter Structures

may help to reduce the world length effects and hardware complexity

Filter Structures - Equivalent Structures

- Two digital filter structures are defined to be ***equivalent*** if they have the ***same*** transfer function
- One simple way to generate an equivalent structure from a given realization is via the *transpose operation* as follows
  - Reverse all path directions
  - Replace pick-off nodes by adder, and vice versa
  - Interchange the input and output nodes

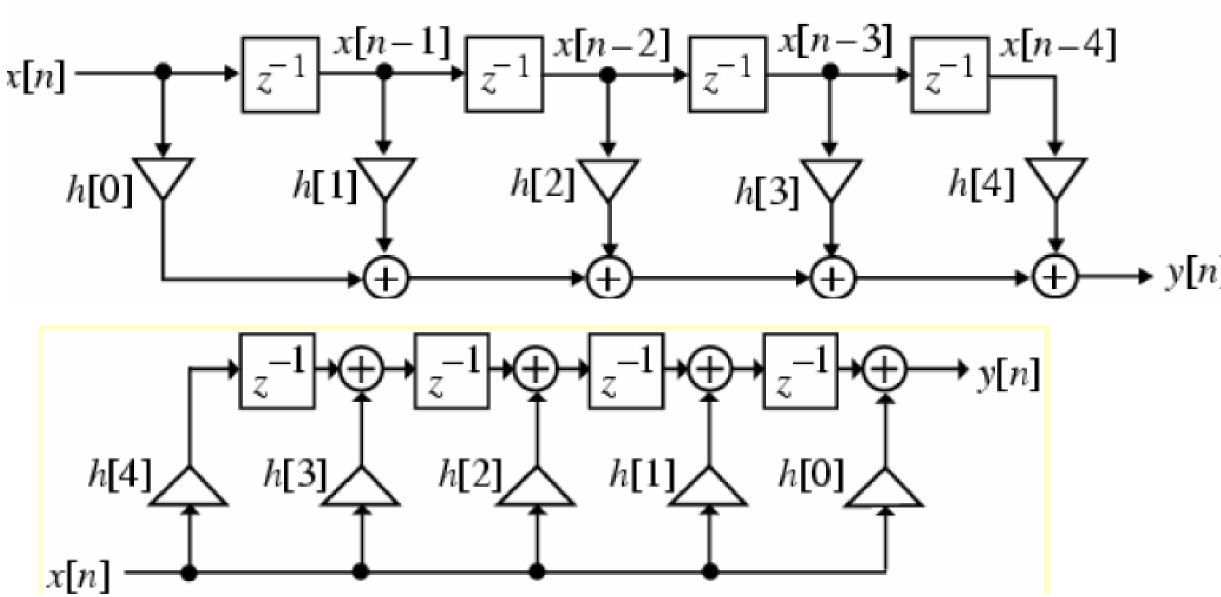


Figure 22 Transformation of a direct form filter

Digital Filter Structures - Equivalent Structures

- All other methods for developing equivalent structures are based on a specific algorithm for each structure

- There are an *infinite* number of equivalent structures realizing the same transfer function. Because it is impossible to develop all equivalent realizations, we mainly deal with some commonly used structures

Filter Structures – Basic FIR Filter Structure

- In certain cases, it is possible to develop a structure that has the **least quantization effects**.
- A causal FIR filter of order N is characterized by a transfer function $H(z)$ given by

$$H[z] = \sum_{n=0}^{N} h(n) z^{-n}$$

  which is a polynomial in z^{-1}
- In the time-domain the output-input relation of the above FIR filter is given by

$$y[n] = \sum_{k=0}^{N} h[k] x[n-k]$$

- An FIR filter of *order* N is characterized by N+1 coefficients and, in general, requiring N+1 multipliers and N 2-input adders

Filter Structures – Direct Form FIR

- Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called *direct form structures*
- A direct form realization of an FIR filter can be readily developed from the *convolution sum* description as indicated below for $N = 4$

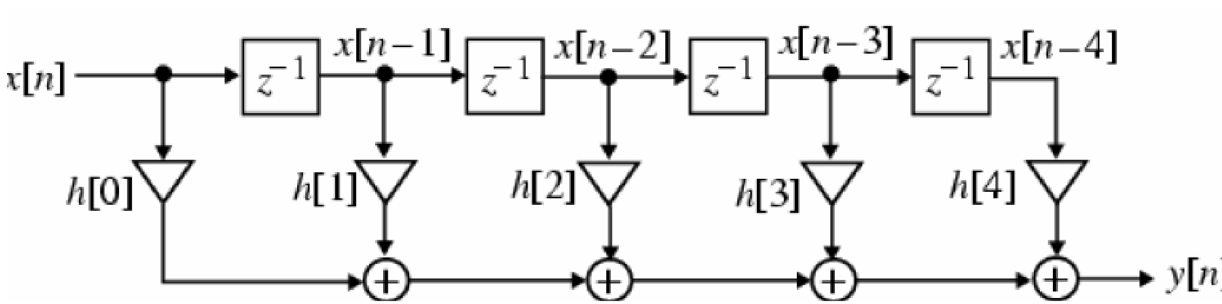


Figure 24 A direct form of an FIR filter

- An analysis of this structure yields

$$y[n] = h[0]x[n]+h[1]x[n\text{-}1]+h[2]x[n\text{-}2]+h[3]x[n\text{-}3]+h[4]x[n\text{-}4]$$

which is precisely the form of the convolution sum description

Filter Structures – Direct Form FIR

- The direct form structure is also known as a *tapped delay line* or a *transversal filter*
- The transpose of the direct form structure shown earlier is indicated below

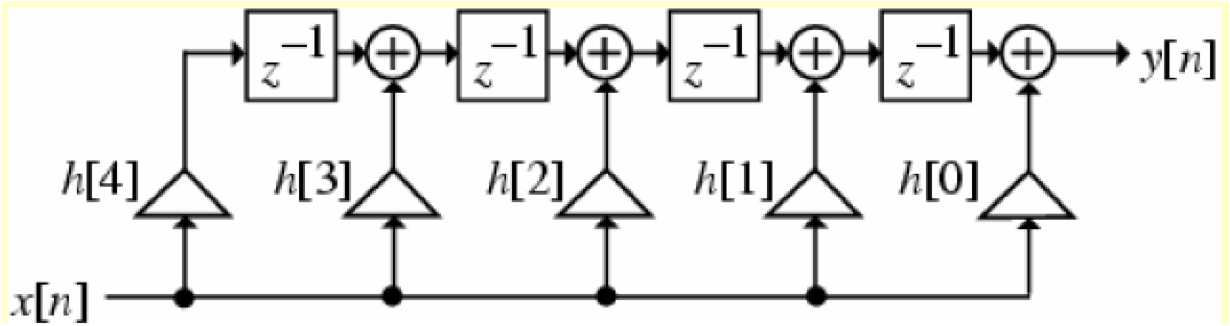


Figure 25 The transpose of the direct form FIR filter

- Both direct form structures are *canonic* with respect to delays
- Observe the arrangement of the delay elements and the location of the coefficient used in the tansposed structure to verify the transpose operation.

Filter Structures – Cascaded Form FIR

- A higher-order FIR transfer function can be realized as a cascade of 1st-order and 2nd-order FIR sections
- We express $H(z)$ as

$$H[z] = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$

where $K = N/2$ for even N, and $K = (N+1)/2$ for odd N with $b_{2K} = 0$.

- A cascaded realization for $N = 6$ is shown

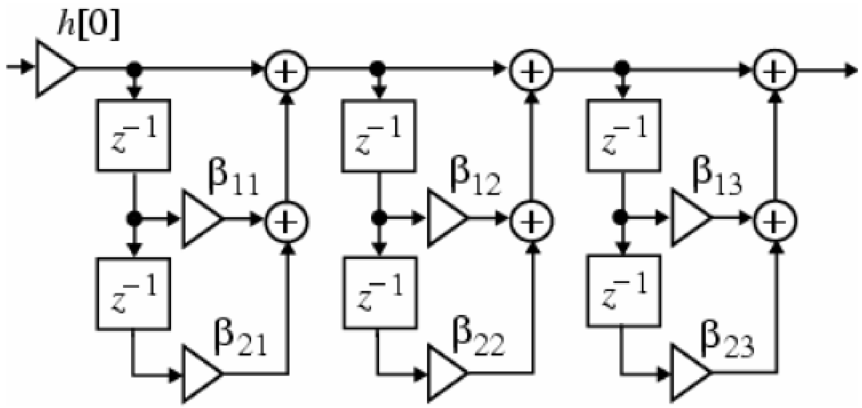


Figure 26 An example of cascading structure (N=6)

Filter Structures – Polyphase FIR

- The polyphase decomposition of $H(z)$ leads to a parallel form structure
- For illustration, consider a causal FIR transfer function $H(z)$ with N = 8:

$$H[z] = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3}$$
$$+ h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8}$$

- $H(z)$ can be expressed as a sum of two terms, with one term containing the even-indexed coefficients and the other containing the odd-indexed coefficients

$$H[z] = (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6} + h[8]z^{-8})$$
$$+ z^{-1}(h[1] + h[3]z^{-2} + h[5]z^{-4} + h[7]z^{-6})$$

Filter Structures – Polyphase FIR

- By using the notation

$$E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4}$$

$$E_1(z) = h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3}$$

- We have

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

- In a similar manner, by grouping the terms in $H(z)$ differently, we can have

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$$

where

$$E_0(z) = h[0] + h[3]z^{-1} + h[6]z^{-2}$$

$$E_1(z) = h[1] + h[4]z^{-1} + h[7]z^{-2}$$

$$E_2(z) = h[2] + h[5]z^{-1} + h[8]z^{-2}$$

Filter Structures – Polyphase FIR

- The decomposition $H(z)$ into the form

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

or

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$$

is commonly known as *polyphase decomposition*.

- In general, an L-branch polyphase decomposition of an FIR transfer function or order N is of the form

$$H(z) = \sum_{m=0}^{L-1} z^{-m}E_m(z^L)$$

where

$$E_m(z) = \sum_{n=0}^{\lfloor (N+1)/L \rfloor} h[Ln+m]z^{-n}$$

with $h[n] = 0$ for $n > N$.

Filter Structures – Polyphase FIR

- The Figures below show the 4-branch, 3-branch and 2-branch polyphase realization of an FIR transfer function $H(z)$.

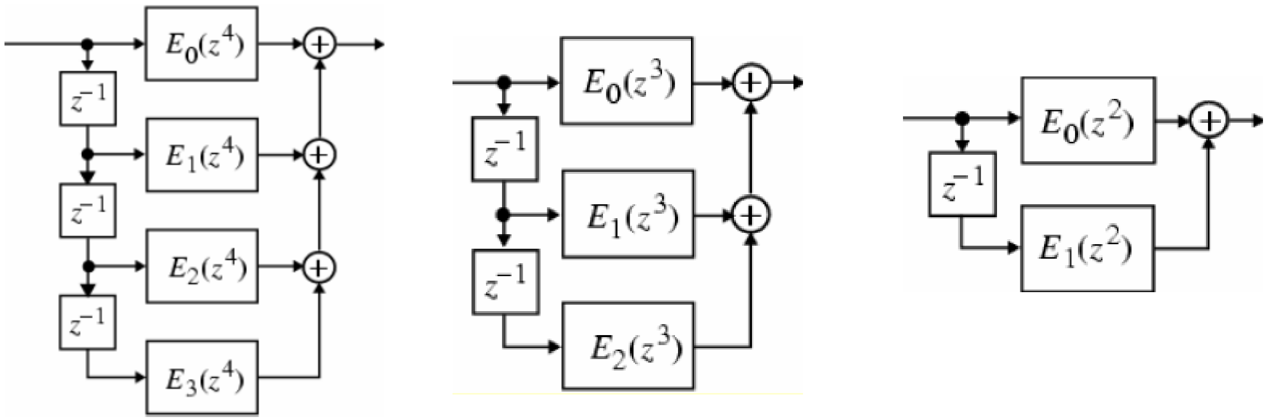


Figure 27 Polyphase structures with L = 2, 3 and 4.

- The expressions for the polyphase components $E_m(z)$ are different in each case
- The subfilters $E_m(z^L)$ in the polyphase realization of all FIR transfer function are also FIR filters and can be realized using any methods described so far
- To obtain a *canonic* realization, when $(N+1)/L$ is an integer, of the overall structure, the delays in all subfilters must be *shared.*

Filter Structures – Polyphase FIR

- The Figure below shows a canonic realization of length-9 FIR transfer function obtained using delay line sharing.

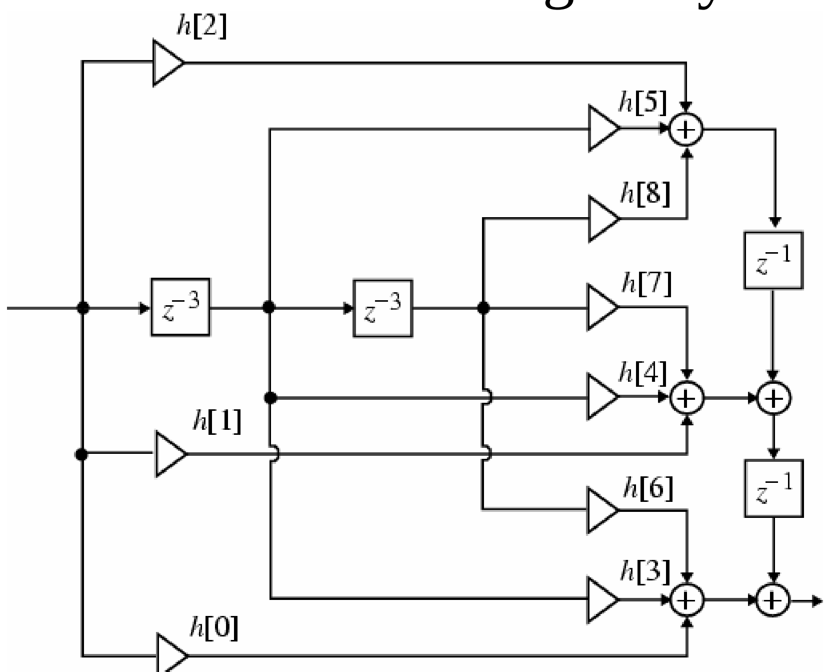


Figure 28 A canonic polyphase realization

- It requires 8 delays shared by subfilters.
- **Question**: What happens if $(N+1)/L$ is not an integer?

Filter Structures – Linear Phase FIR

- The symmetry (or antisymmetry) property of a linear-phase FIR filter can be exploited to reduce the no. of multipliers into almost half of that in the direct form implementations
- Consider a length-7 Type-I (for odd value of filter length) FIR transfer function with a *symmetric* impulse response:

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(2)z^{-4} + h(1)z^{-5} + h(0)z^{-6}$$

- Rewrite $H(z)$ into

$$H(z) = h(0)(1 + z^{-6}) + h(1)(z^{-1} + z^{-5}) + h(2)(z^{-2} + z^{-4}) + h(3)z^{-3}$$

- We obtain

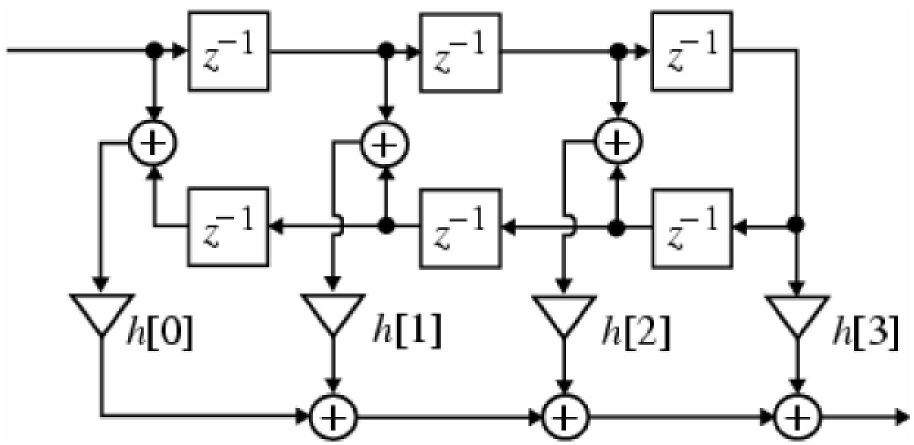


Figure 29 The implementation of FIR filter by using the symmetric property

Filter Structures – Linear Phase FIR

- A similar decomposition can be applied to a Type-II FIR transfer function for even value of filter length
- For example, a length-8 Type-II FIR transfer function can be expressed as

$$H(z) = h(0)(1+z^{-7}) + h(1)(z^{-1}+z^{-6}) + h(2)(z^{-2}+z^{-5}) + h(3)(z^{-3}+z^{-4})$$

- The corresponding realization is

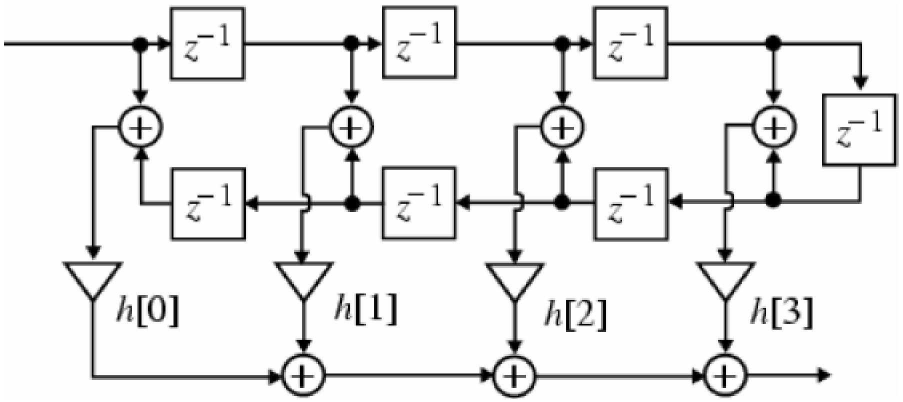


Figure 30 The implementation of FIR filter by using the symmetric property

Filter Structures – Linear Phase FIR

- Note: the type-I linear-phase structure for length-7 FIR filter needs 4 multipliers whereas a direct form realization uses 7 multipliers.
- Note: the type-II linear-phase structure for a length-8 FIR filter requires 4 multipliers, whereas a direct form realization requires 8 multipliers

Type	impulse response	
I	symmetric	length is odd
II	symmetric	length is even
III	anti-symmetric	length is odd
IV	anti-symmetric	length is even

Summary for Lecture 2

- Type and classification of Filters. In particular, the FIR filters are our main focus
- Understand that issues for practical implementation such as ripples, transitional band and word length effects.
- Able to analyze the type of filters based on their transfer functions expressed in *z*-domain
- Able to derive filter transfer functions from a given signal flow graph
- Able to derive equivalent structures from a given filter