Chapter 2

Review of Fundamentals of Digital Signal Processing

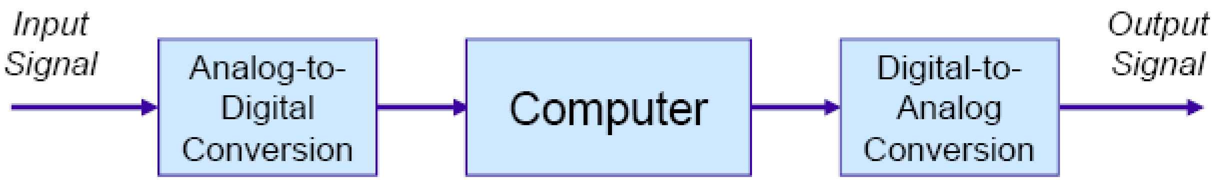
数字信号处理基础回顾

Outline

- DSP and Discrete Signals
- LTI Systems
- z-Transform Representations
- Discrete-Time Fourier Transform (DTFT)
- Discrete Fourier Transform (DFT)
- Digital Filtering
- Sampling

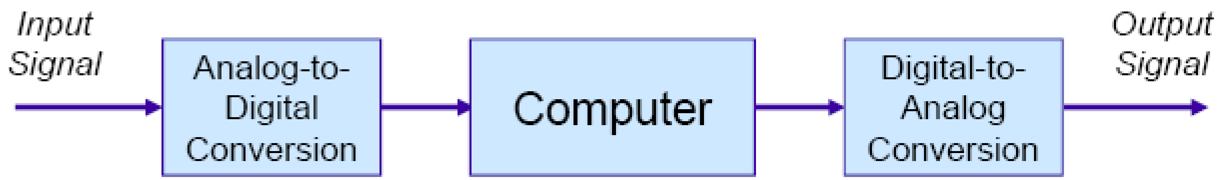
DSP and Discrete Signals

What is DSP?



- Digital
  - Method to represent a quantity, a phenomenon or an event
- Signal
  - something (e.g., a sound, gesture, or object) that carries information
  - a detectable physical quantity (e.g., a voltage, current, or magnetic field strength) by which messages or information can be transmitted
- Processing
  - Filtering/spectral analysis
  - Analysis, recognition, synthesis and coding of real world signals
  - Detection and estimation of signals in the presence of noise or interference

Digital Processing of Analog Signals



- A-to-D conversion: bandwidth control, sampling and quantization
- Computational processing: implemented on computers or ASICs(专用集成电路) with finite-precision arithmetic
  - basic numerical processing: add, subtract, multiply (scaling, amplification, attenuation), mute, …
  - algorithmic numerical processing: convolution or linear filtering, non-linear filtering (e.g., median filtering), difference equations, DFT, inverse filtering, MAX/MIN, …
- D-to-A conversion: re-quantification and filtering (or interpolation) for reconstruction

Discrete-Time Signals

- A sequence of numbers
- Mathematical representation

$$x[n], -\infty < n < \infty$$

- Sampled from an analog signal, $x_a(t)$, at time $t = nT$

$$x[n] = x_a(nT), -\infty < n < \infty$$

- T is called the sampling period(采样周期), and its reciprocal $F_s = 1/T$ is called the sampling frequency(采样频率)

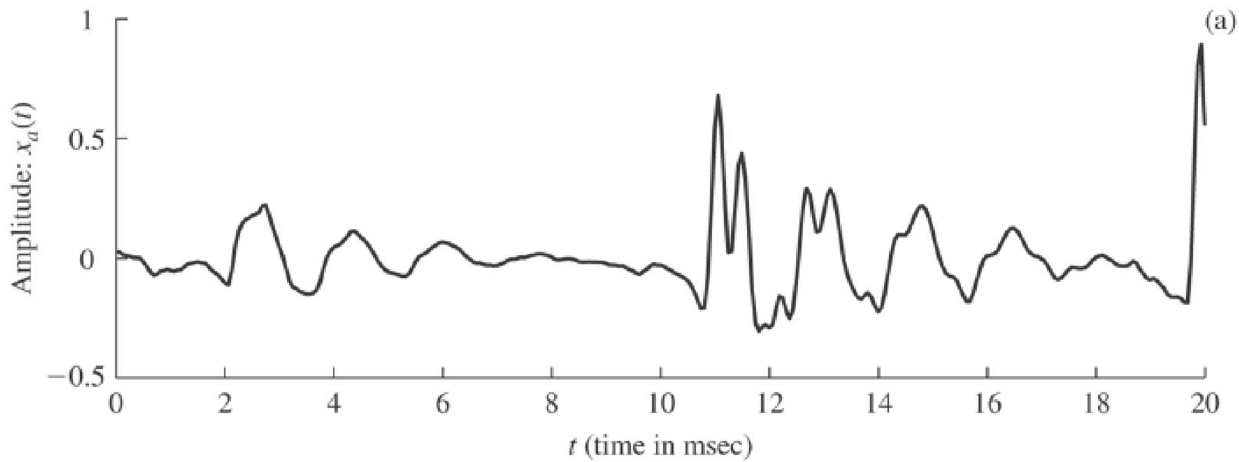
$$F_s = 8000\text{Hz} \leftrightarrow T = 1/8000 = 125\mu\sec$$

$$F_s = 10000\text{Hz} \leftrightarrow T = 1/10000 = 100\mu\sec$$

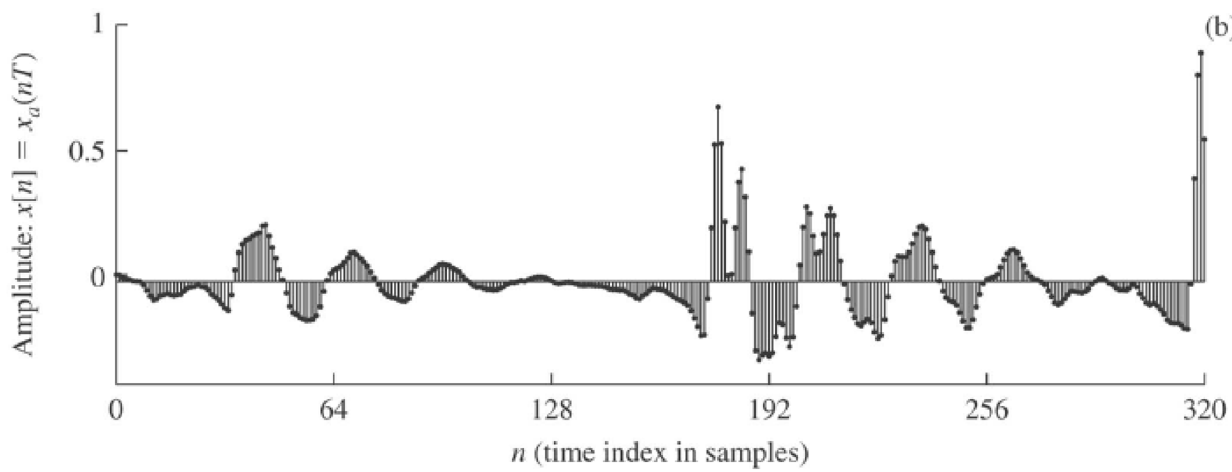
$$F_s = 16000\text{Hz} \leftrightarrow T = 1/16000 = 62.5\mu\sec$$

$$F_s = 20000\text{Hz} \leftrightarrow T = 1/20000 = 50\mu\sec$$

Speech Waveform Display

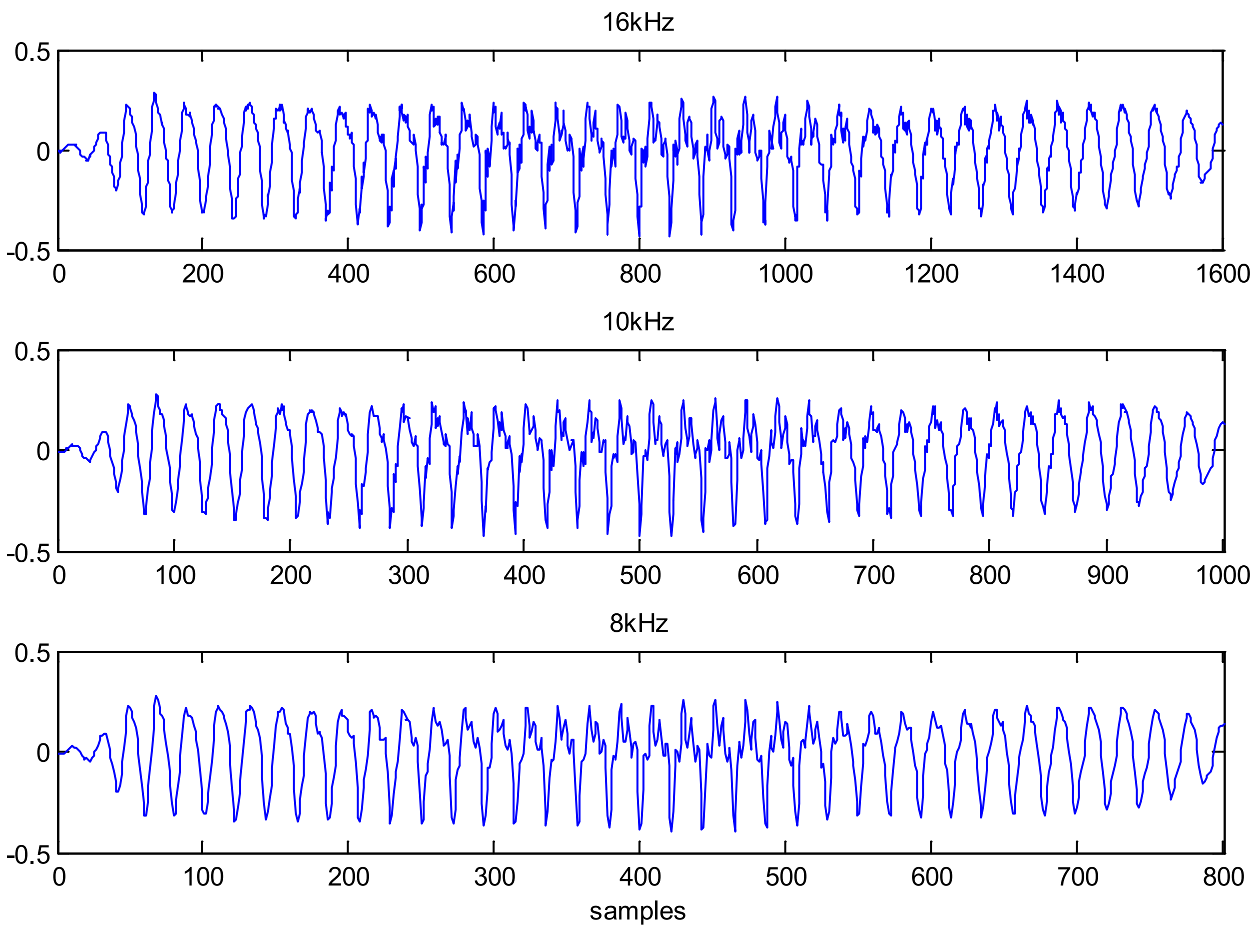


plot( );

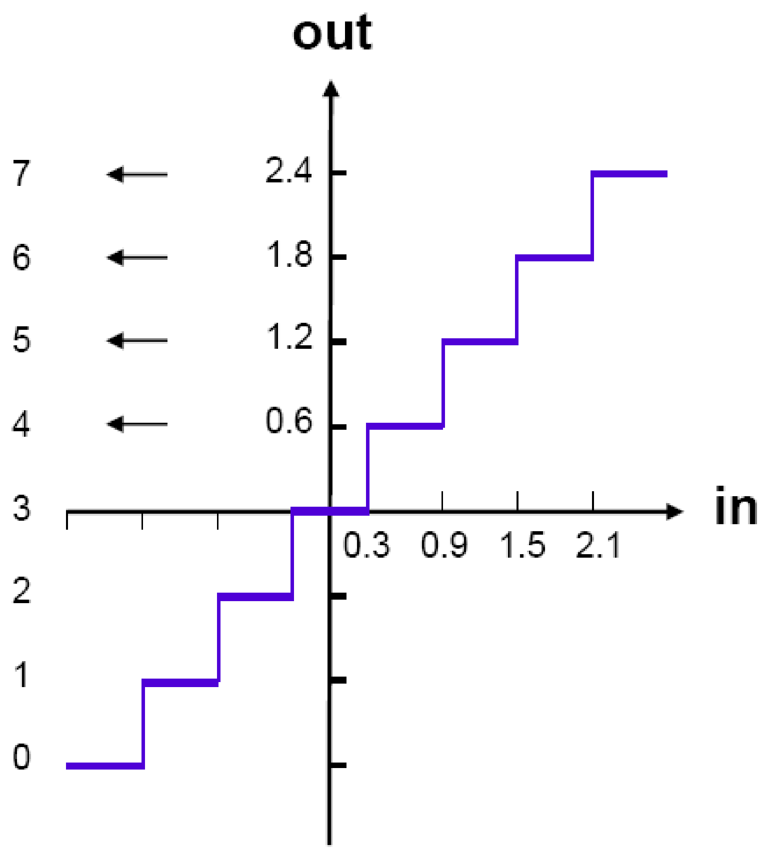


stem( );

Varying Sampling Rates



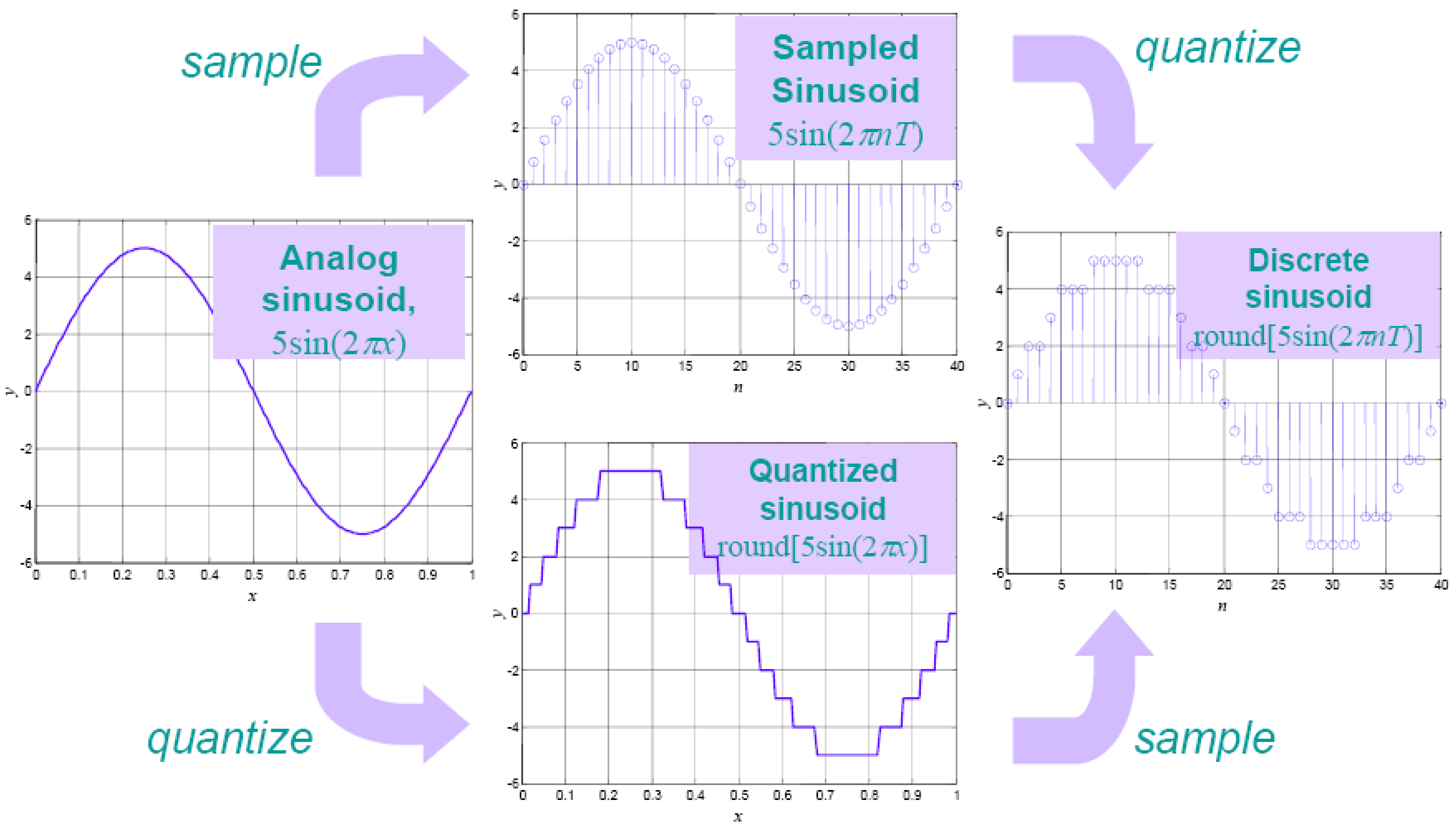
Quantization



A 3-bit uniform quantizer

- Transforming a continuously valued input into a representation that assumes one out of a finite set of values
- The finite set of output values is indexed; e.g., the value 1.8 has an index of 6, or (110) in binary representation
- Storage or transmission uses binary representation; a quantization table is needed

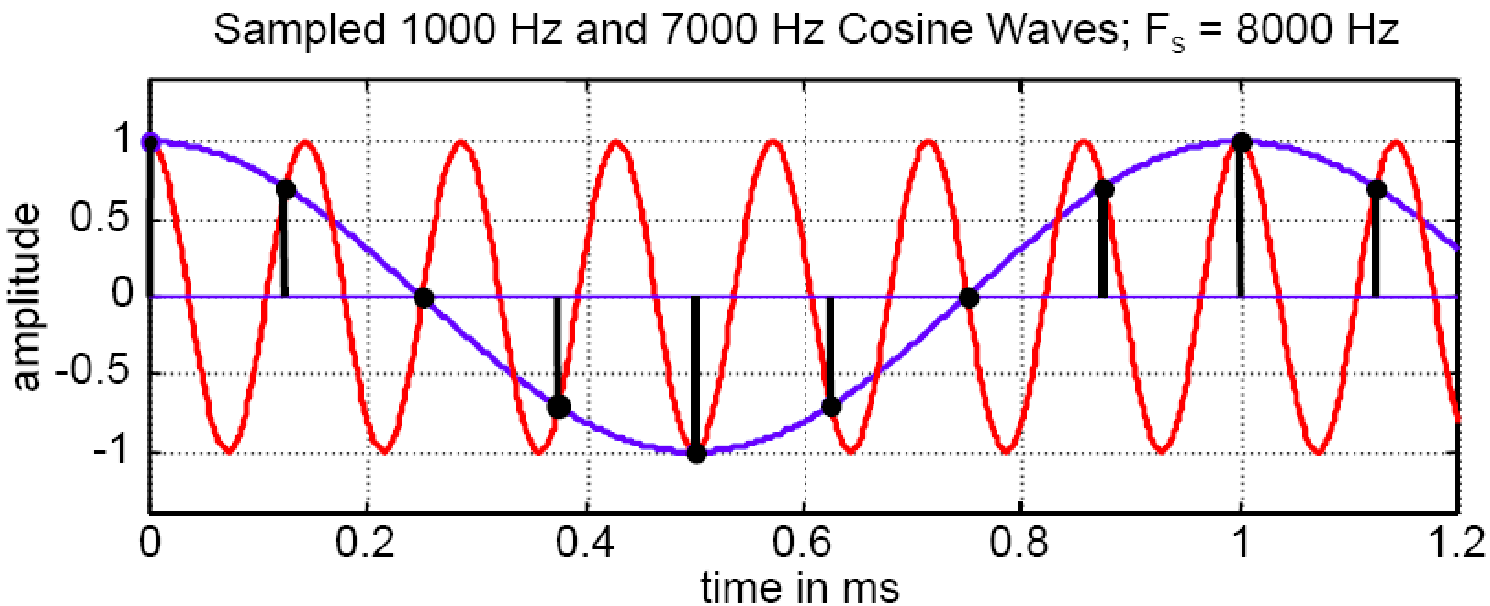
Discrete Signals



Issues with Discrete Signals

- what sampling rate *is appropriate*
  - 6.4 kHz (telephone bandwidth), 8 kHz (extended telephone BW), 10 kHz (extended bandwidth), 16 kHz (Hi-Fi speech)
- how many quantization levels *are necessary at* each bit rate (bits/sample)
  - 16, 12, 8, … => ultimately determines the S/N ratio of the speech
  - speech coding is concerned with answering this question in an optimal manner

The Sampling Theorem



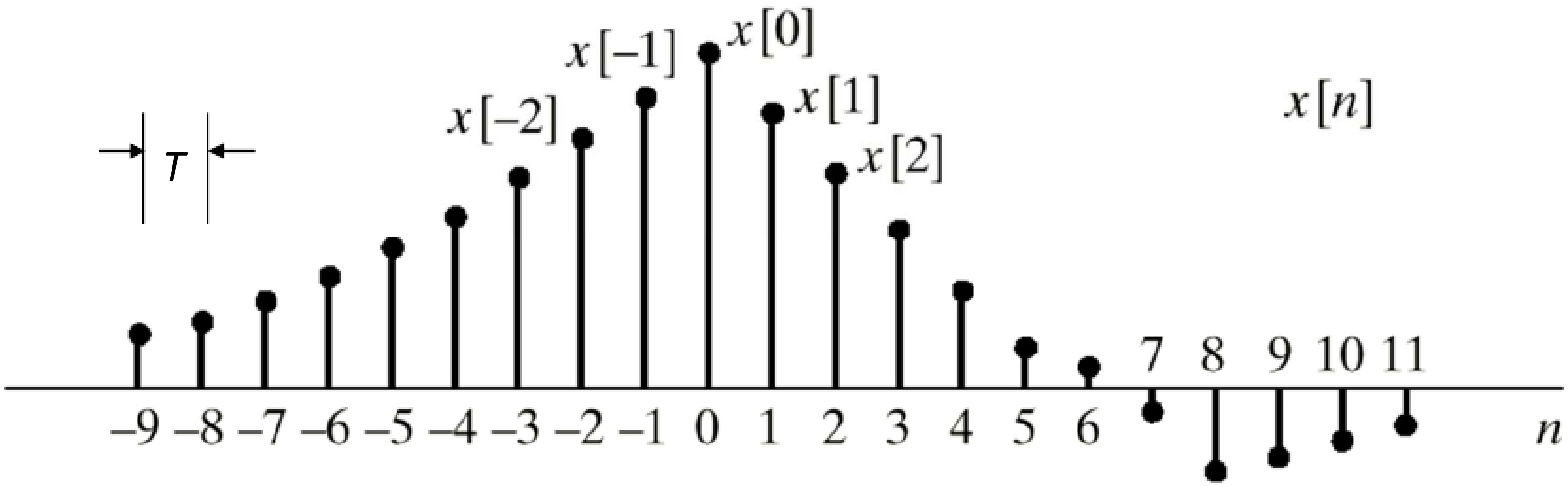
- A bandlimited signal can be reconstructed exactly from samples taken with sampling frequency

$$\frac{1}{T} = F_s \geq 2 f_{\max} \quad \text{or} \quad \frac{2\pi}{T} = \omega_s \geq 2\omega_{\max}$$

Demo Examples

- 5 kHz analog bandwidth
  - sampled at 10, 5, 2.5, 1.25 kHz (notice the aliasing that arises when the sampling rate is below 10 kHz)
- quantization to various levels
  - 16,12,8, and 4 bit quantization (notice the distortion introduced when the number of bits is too low)

Discrete-Time (DT) Signals are Sequences



- $x[n]$ denotes the “sequence value at ‘time’ n”
- Sources of sequences
  - Sampling a continuous-time signal

$$x[n] = x_c(nT) = x_c(t)\big|_{t=nT}$$

  - Mathematical formulas – generative system

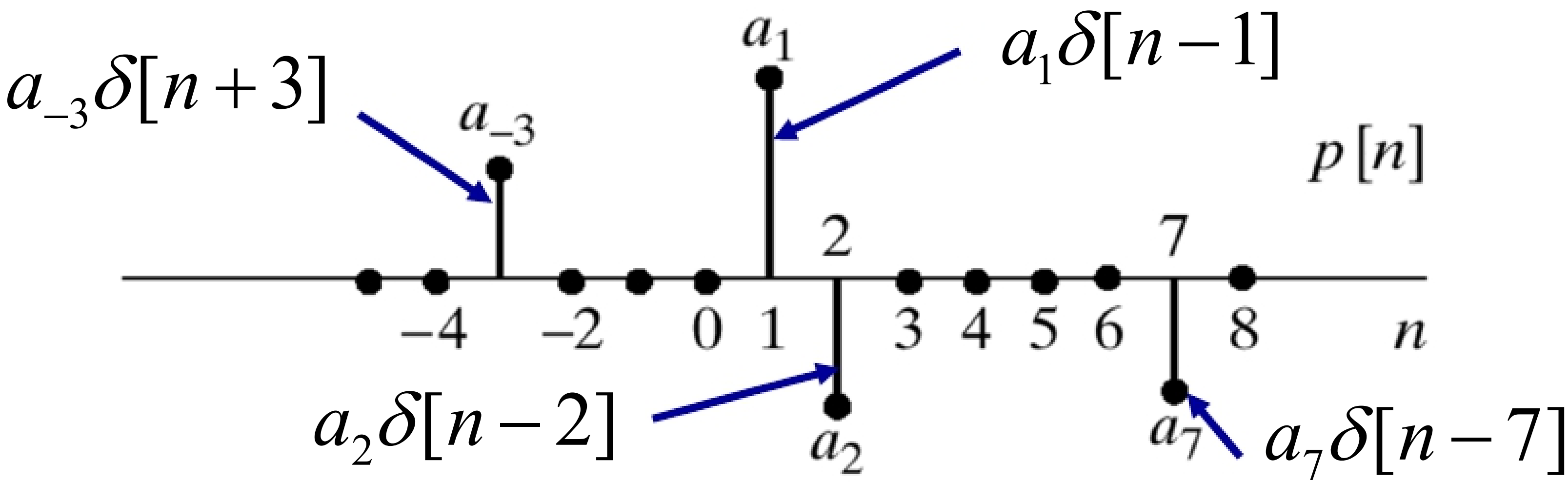
$$\text{e.g.,} \quad x[n] = 0.3x[n-1] - 1; \quad x[0] = 40$$

Impulse Representation of Sequences

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

A sequence, a function

Value of the function at k

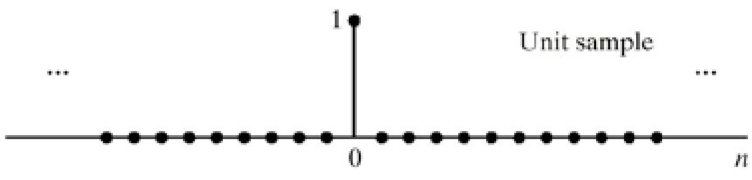


$$x[n] = a_{-3}\delta[n+3] + a_1\delta[n-1] + a_2\delta[n-2] + a_7\delta[n-7]$$

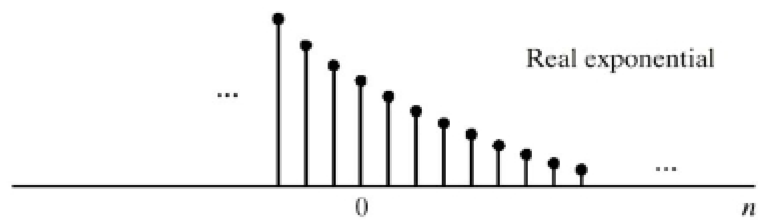
Some Useful Sequences

unit sample $\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$

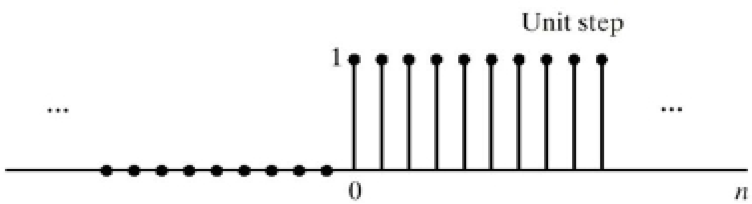
real exponential $x[n] = \alpha^n$



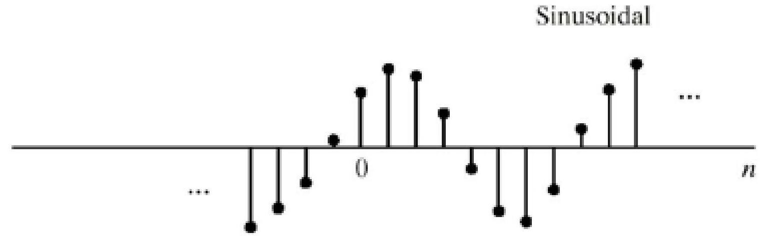
(a)



(c)



(b)

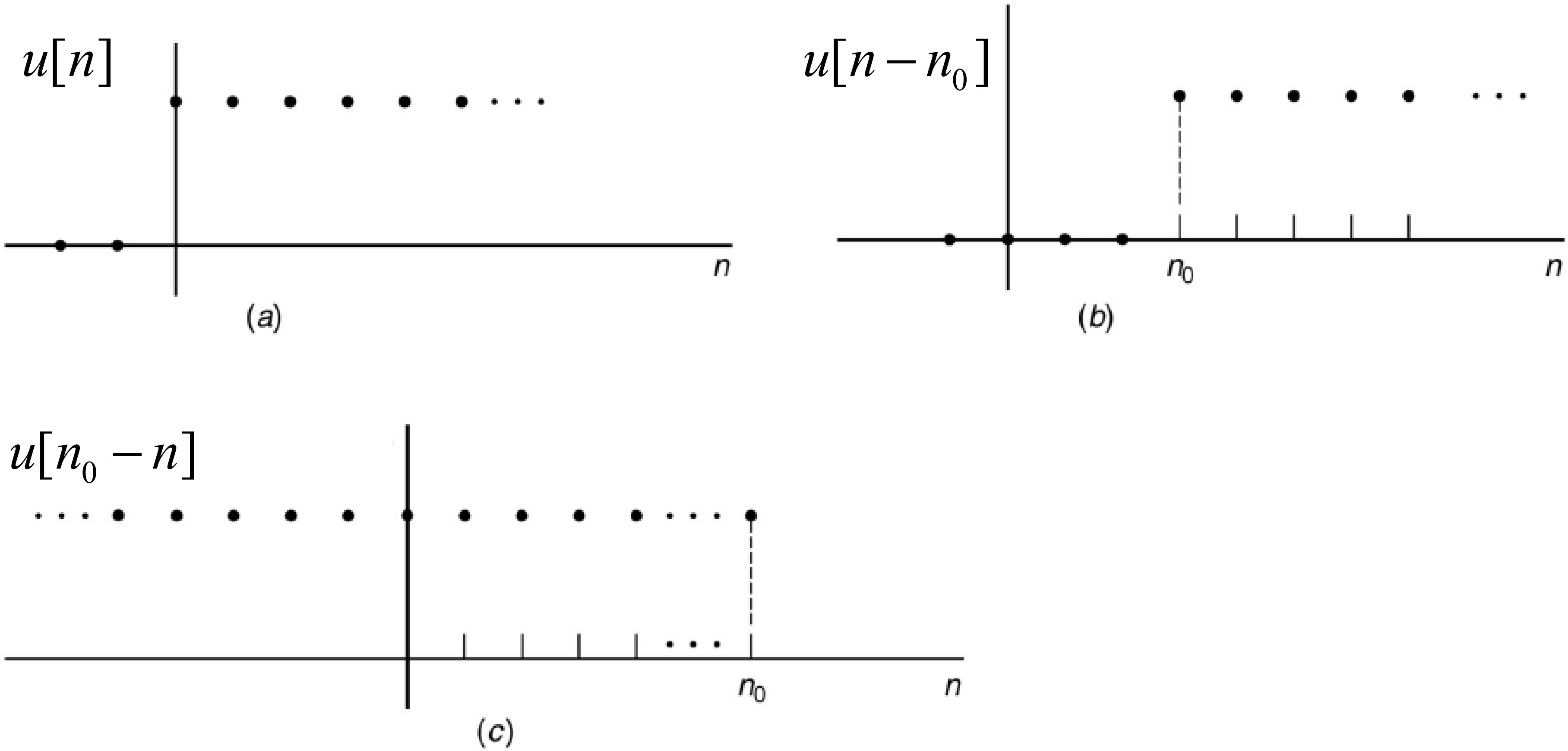


(d)

unit step $u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$

sine wave $x[n] = A\cos(\omega_0 n + \phi)$

Variants on Discrete-Time Step Function

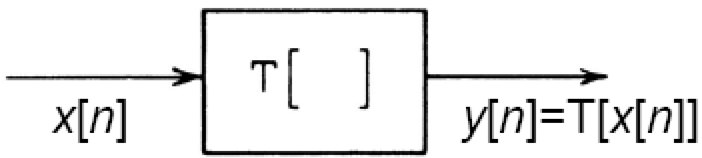


$n \rightarrow -n \quad \Leftrightarrow$ signal flips around 0

LTI Systems

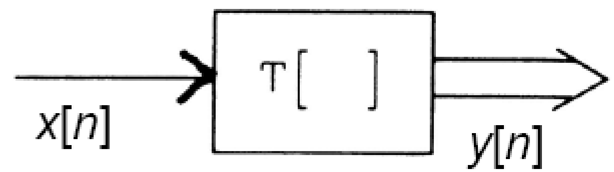
Signal Processing

- Transform digital signal into more desirable form



(a)

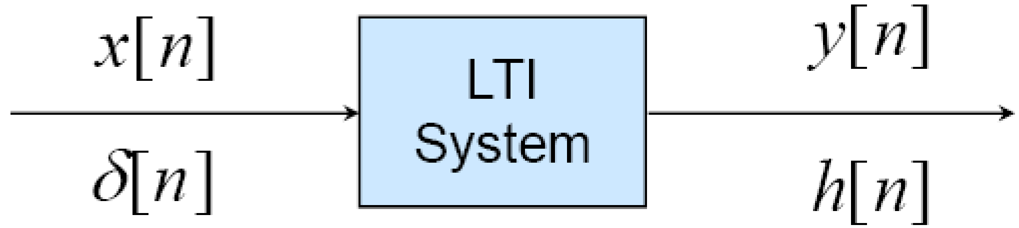
single input—single output



(b)

single input—multiple output, e.g., filter bank analysis, etc.

LTI Discrete-Time Systems



- Linearity (superposition)

$$\mathrm{T}\{ax_1[n]+bx_2[n]\} = a\mathrm{T}\{x_1[n]\} + b\mathrm{T}\{x_2[n]\}$$

- Time-Invariance (shift-invariance)

$$x_1[n] = x[n-n_d] \;\Rightarrow\; y_1[n] = y[n-n_d]$$

- LTI implies discrete convolution

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n]*h[n] = h[n]*x[n]$$

LTI Discrete-Time Systems

<u>Example</u>

Is system $y[n] = x[n] + 2x[n+1] + 3$ linear?

$$x_1[n] \to y_1[n] = x_1[n] + 2x_1[n+1] + 3$$

$$x_2[n] \to y_2[n] = x_2[n] + 2x_2[n+1] + 3$$

$$x_1[n] + x_2[n] \to$$

$$y_3[n] = x_1[n] + x_2[n] + 2x_1[n+1] + 2x_2[n+1] + 3 \neq y1[n] + y2[n]$$

$\Rightarrow$ Not a linear system!

Is system $y[n] = x[n] + 2x[n+1] + 3$ time/shift invariant?

$$y[n] = x[n] + 2x[n+1] + 3$$

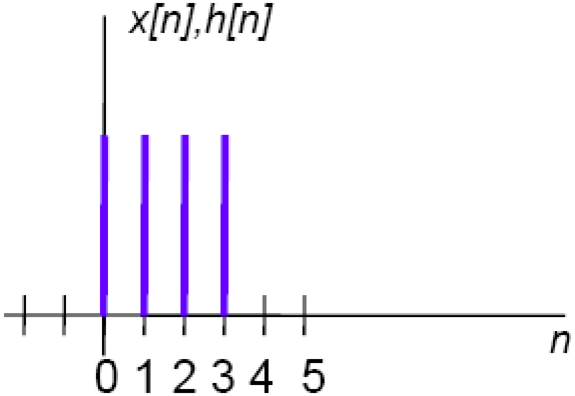
$y[n-n_0] = x[n-n_0] + 2x[n-n_0+1] + 3 \Rightarrow$ System is time invariant!

Is system $y[n] = x[n] + 2x[n+1] + 3$ causal?

$y[n]$ depends on $x[n+1] \Rightarrow$ System is not causal !

Convolution Example

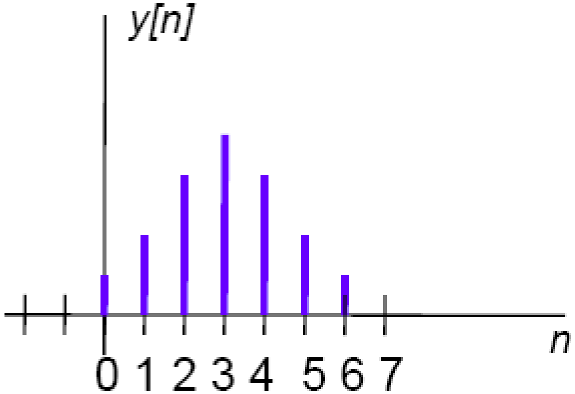
$$x[n] = \begin{cases} 1, & 0 \le n \le 3 \\ 0, & \text{otherwise} \end{cases} \qquad h[n] = \begin{cases} 1, & 0 \le n \le 3 \\ 0, & \text{otherwise} \end{cases}$$

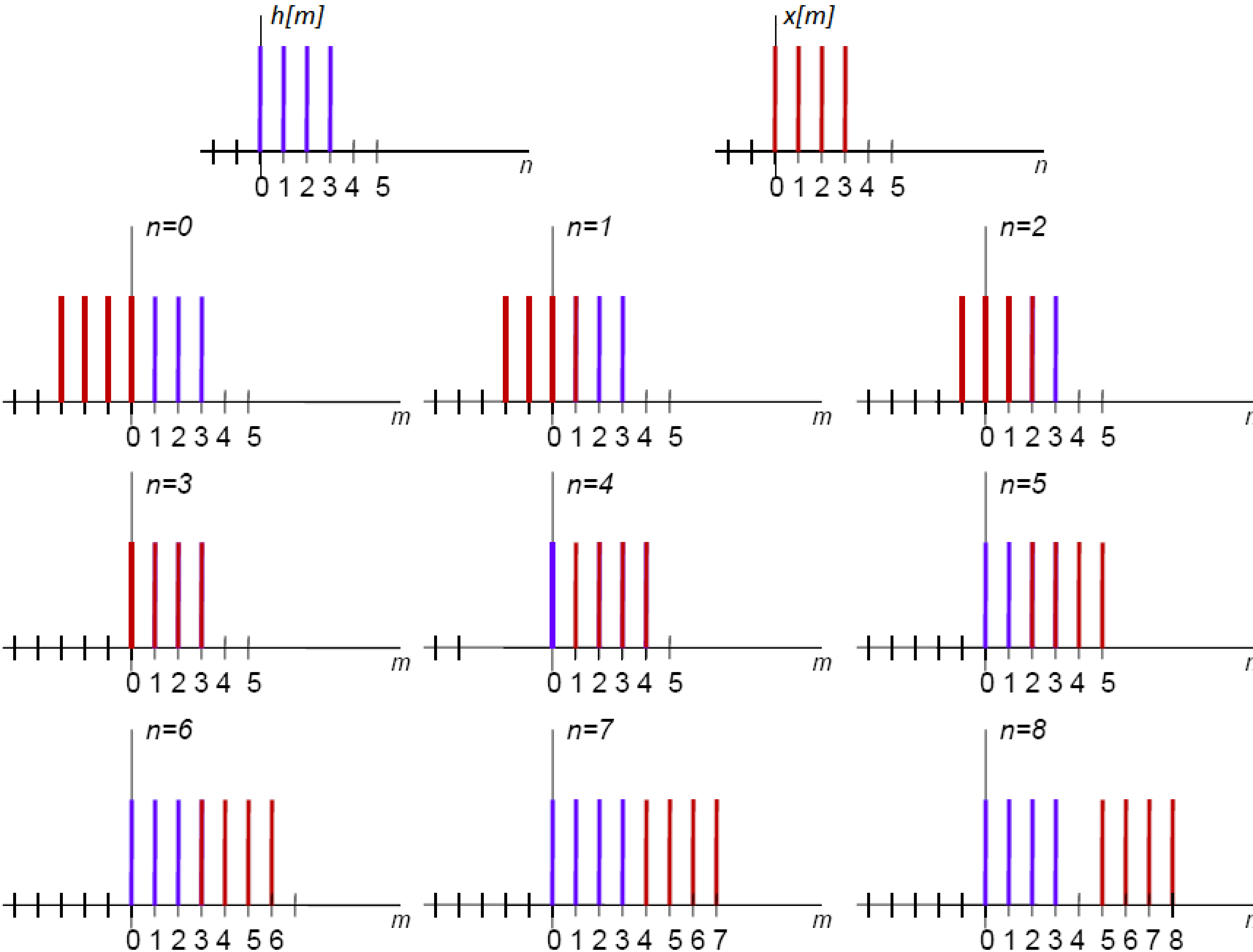


What is $y[n]$ for this system?

Solution $y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$

$$= \begin{cases} \sum_{m=0}^{n} 1 \cdot 1 = n+1, & 0 \le n \le 3 \\ \sum_{m=n-3}^{3} 1 \cdot 1 = 7-n, & 4 \le n \le 6 \\ 0, & \text{otherwise} \end{cases}$$




h[m]
n
0 1 2 3 4 5
x[m]
n
0 1 2 3 4 5
n=0
m
0 1 2 3 4 5
n=1
m
0 1 2 3 4 5
n=2
m
0 1 2 3 4 5
n=3
m
0 1 2 3 4 5
n=4
m
0 1 2 3 4 5
n=5
m
0 1 2 3 4 5
n=6
m
0 1 2 3 4 5 6
n=7
m
0 1 2 3 4 5 6 7
n=8
m
0 1 2 3 4 5 6 7 8

Convolution Example

The impulse response of an LTI system is of the form

$$h[n] = a^n u[n] \qquad |a| < 1$$

and the input to the system is of the form

$$x[n] = b^n u[n] \qquad |b| < 1, b \neq a$$

Determine the output of the system using the formula for discrete convolution.

Solution $$y[n] = \sum_{m=-\infty}^{\infty} a^m u[m]\, b^{n-m} u[n-m]$$

$$= b^n \sum_{m=0}^{n} a^m b^{-m} u[n] = b^n \sum_{m=0}^{n} (a/b)^m u[n]$$

$$= b^n \left[\frac{1-(a/b)^{n+1}}{1-(a/b)} \right] u[n] = \left[\frac{b^{n+1} - a^{n+1}}{b-a} \right] u[n]$$

Convolution Example

Consider a digital system with input $x[n] = 1$ for $n=0,1,2,3$ and 0 everywhere else, and with impulse response $h[n] = a^n u[n], |a| < 1$

Determine the response $y[n]$ of this linear system.

Solution

- We recognize that $x[n]$ can be written as the difference between two step functions, i.e., $x[n]= u[n]- u[n-4]$.
- Hence we can solve for $y[n]$ as the difference between the output of the linear system with a step input and the output of the linear system with a delayed step input.
- Thus we solve for the response to a unit step as:

$$y_1[n] = \sum_{m=-\infty}^{\infty} u[m] a^{n-m} u[n-m] = \left[\frac{a^n - a^{-1}}{1 - a^{-1}} \right] u[n]$$

$$y[n] = y_1[n] - y_1[n-4]$$

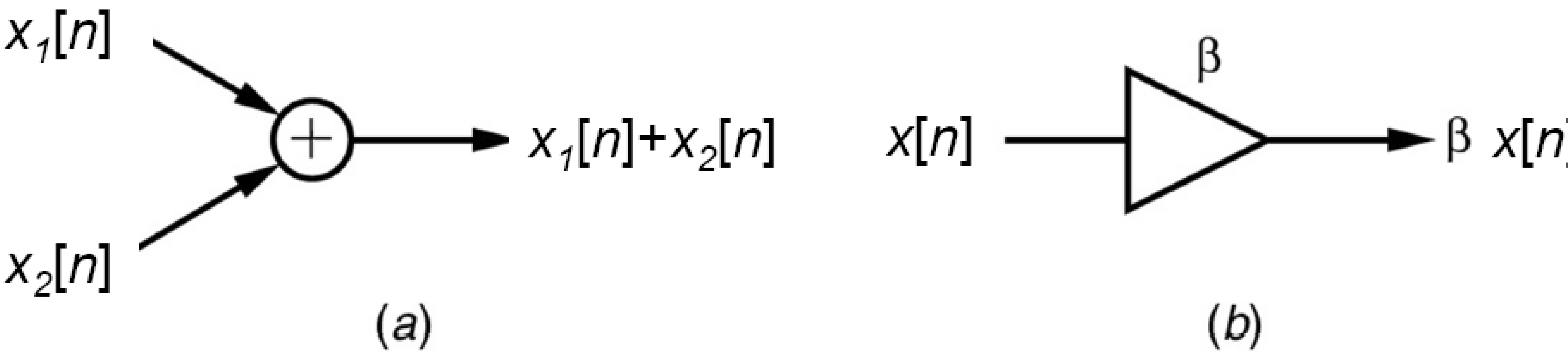
Linear Time-Invariant Systems

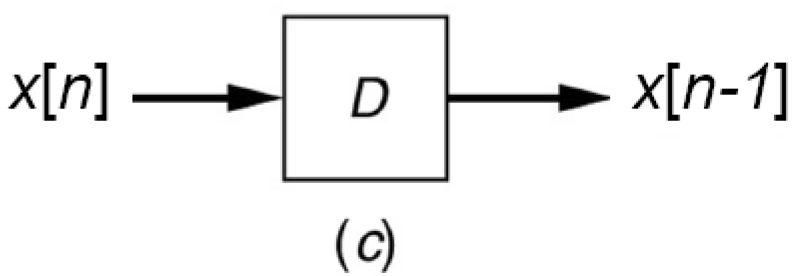
- easiest to understand
- easiest to manipulate
- powerful processing capabilities
- characterized completely by their response to unit sample, $h[n]$, via convolution relationship

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = h[n] * x[n]$$

- basis for linear filtering
- used as models for speech production (source convolved with system)

Signal Processing Operations

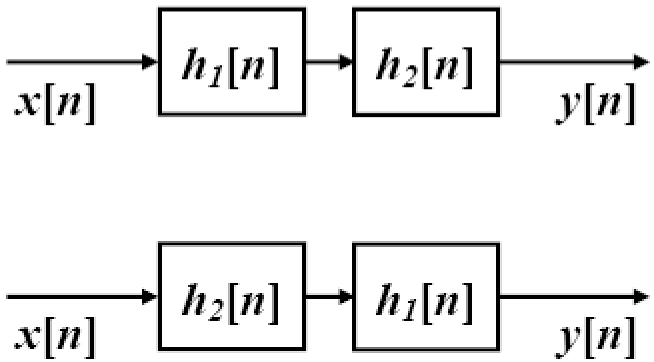


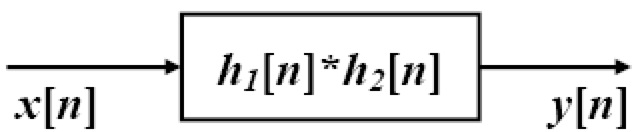


D is a delay of 1-sample

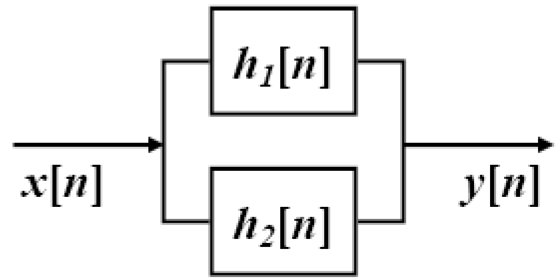
Can replace D with delay element z^{-1}

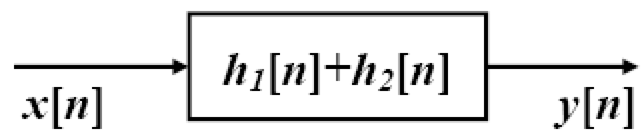
Equivalent LTI Systems





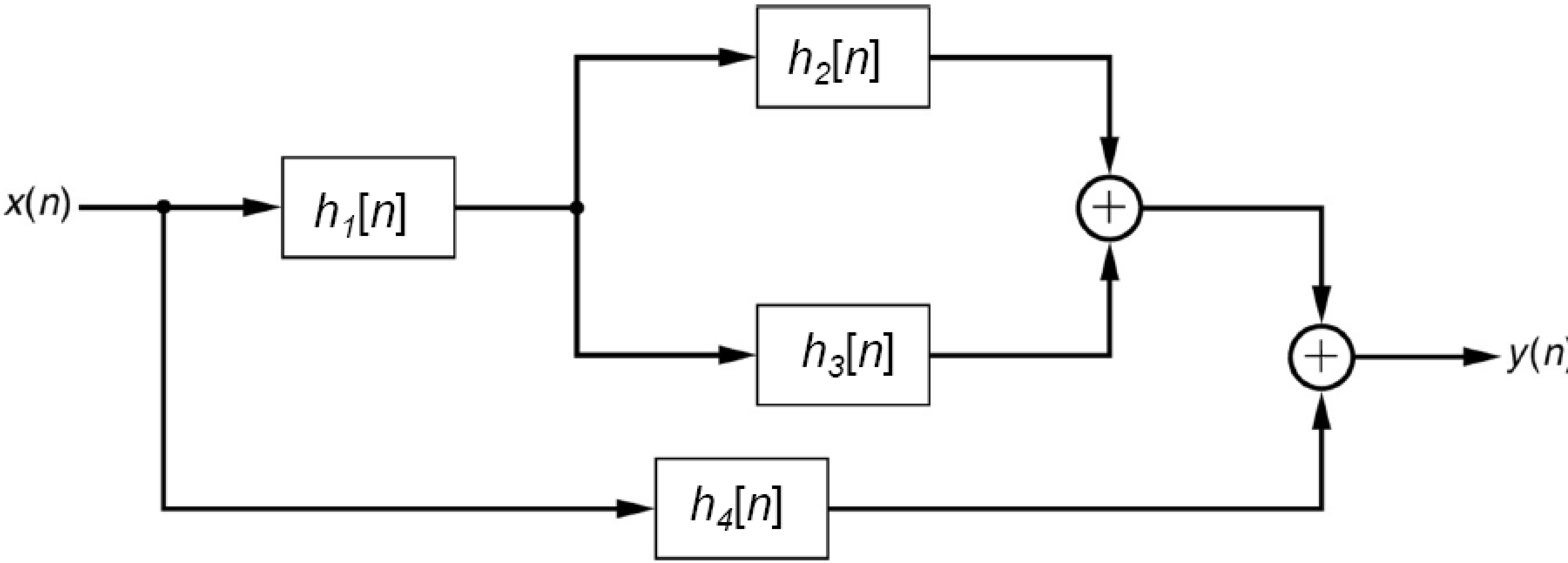
$$h_1[n] * h_2[n] = h_2[n] * h_1[n]$$





$$h_1[n] + h_2[n] = h_2[n] + h_1[n]$$

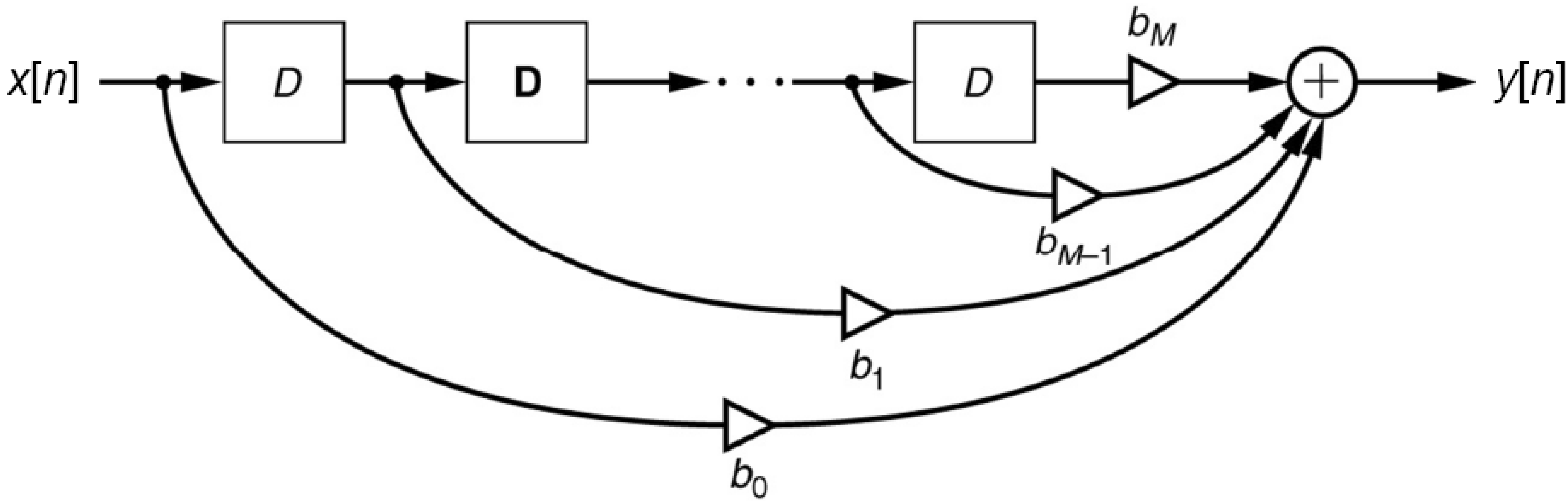
More Complex Filter Interconnections



$$y[n] = x[n] * h_c[n]$$

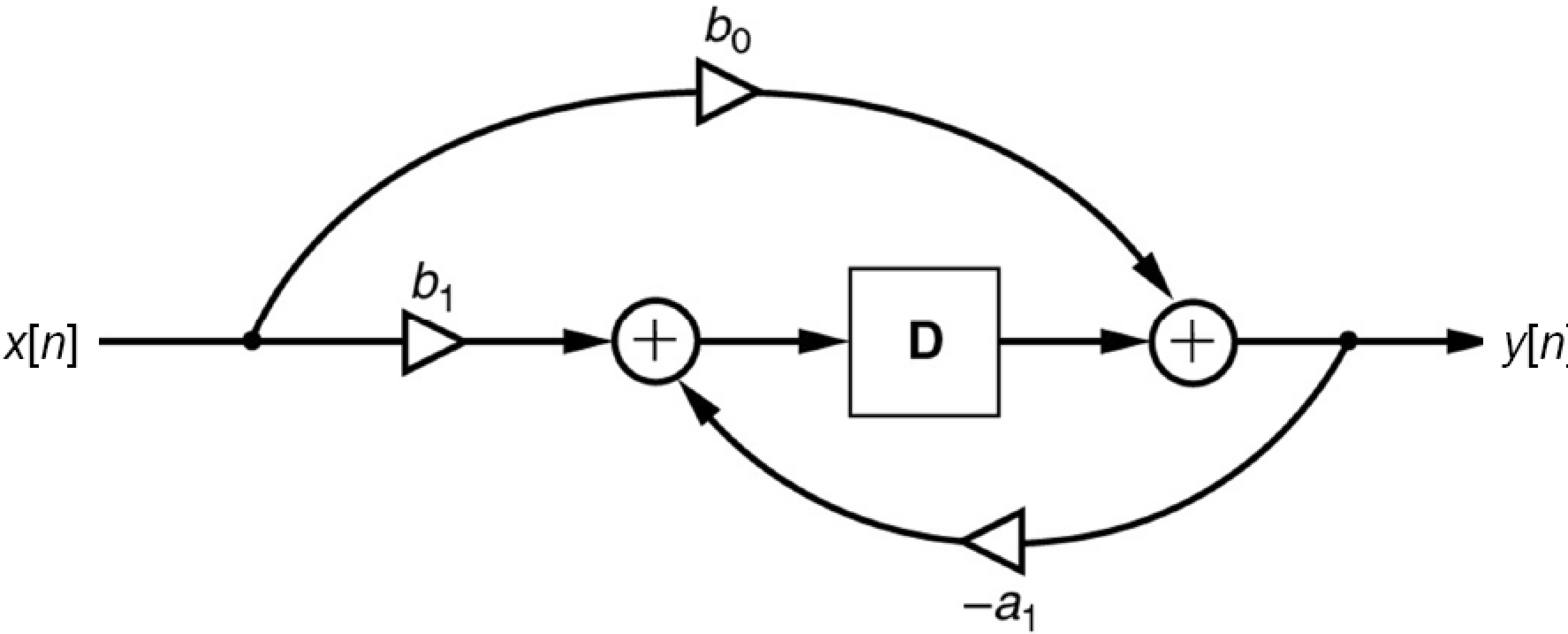
$$h_c[n] = h_1[n] * (h_2[n] + h_3[n]) + h_4[n]$$

Network View of Filtering (FIR Filter)



$$y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_{M-1} x[n-M+1] + b_M x[n-M]$$

Network View of Filtering (IIR Filter)



$$y[n] = -a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

z-Transform Representations

Transform Representations

- z-Transform

$$x[n] \leftrightarrow X(z)$$

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

$$x[n] = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz$$

Infinite power series in z^{-1}, with $x[n]$ as coefficients of term in z^{-n}

- direct evaluation using residue theorem (留数定理)
- partial fraction expansion (部分分式展开)of $X(z)$
- long division (长除法)
- power series expansion(幂级数展开)

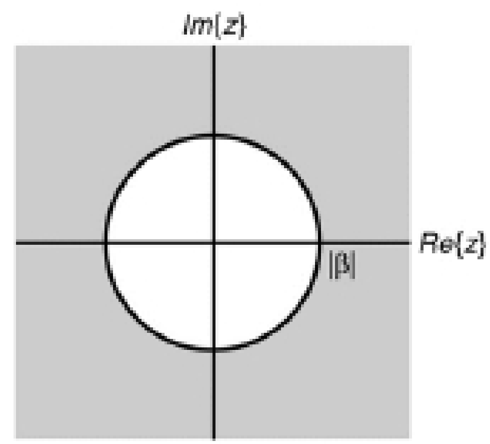
Transform Representations

- $X(z)$ converges (is finite) only for certain values of z
  - Sufficient condition for convergence

$$\sum_{n=-\infty}^{\infty} |x[n]| |z^{-n}| < \infty$$

- region of convergence

$$R_1 < |z| < R_2$$



Examples of Converge Regions

1. Delayed impulse $x[n] = \delta[n - n_0]$

   $X(z) = z^{-n_0}$ converges for $|z| > 0, n_0 > 0; |z| < \infty, n_0 < 0; \forall z, n_0 = 0$

2. Box pulse $x[n] = u[n] - u[n - N]$

   $$X(z) = \sum_{n=0}^{N-1} z^{-n} = \frac{1 - z^{-N}}{1 - z^{-1}}$$ converges for $0 < |z| < \infty$

<u>all finite length sequences converge in the region</u> $0 < |z| < \infty$

3. $x[n] = a^n u[n]$ $(|a| < 1)$

   $$X(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \frac{1}{1 - az^{-1}}$$ converges for $|z| > |a|$

<u>all infinite duration sequences which are non-zero for $n \geq 0$ converge in the region</u> $|z| > R_1$

Examples of Converge Regions

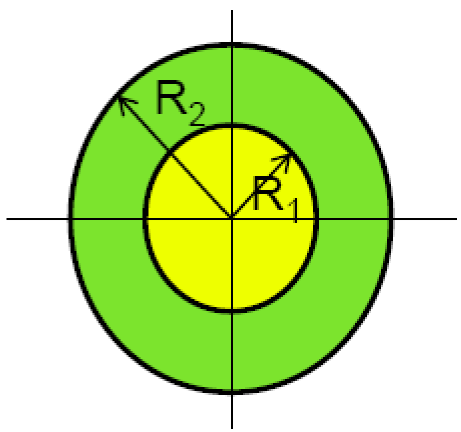
4. $x[n] = -b^n u[-n-1]$

$$X(z) = \sum_{n=-\infty}^{-1} -b^n z^{-n} = \frac{1}{1-bz^{-1}}$$ converges for $|z| < |b|$

all infinite duration sequences which are non-zero for n<0 converge in the region $|z| < R_2$

5. $x[n]$ non-zero for -∞<n< ∞ can be viewed as a combination of 3 and 4, giving a convergence region of the form $R_1 < |z| < R_2$

- sub-sequence for n≥0 $\Rightarrow |z| > R_1$
- sub-sequence for n<0 $\Rightarrow |z| < R_2$
- total sequence $\Rightarrow R_1 < |z| < R_2$



Examples

If $x[n]$ has z-transform $X(z)$ with ROC of $r_i<|z|<r_o$, find the z-transform, $Y(z)$, and the region of convergence for the sequence $y[n]=a^n x[n]$ in terms of $X(z)$

Solution

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$Y(z) = \sum_{n=-\infty}^{\infty} y[n]z^{-n} = \sum_{n=-\infty}^{\infty} a^n x[n]z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x[n](z/a)^{-n} = X(z/a)$$

$$\text{ROC}: \quad |a| r_i < |z| < |a| r_o$$

Examples

The sequence $x[n]$ has z-transform $X(z)$. Show that the sequence $nx[n]$ has z-transform $-z\mathrm{d}X(z)/\mathrm{d}z$

Solution

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

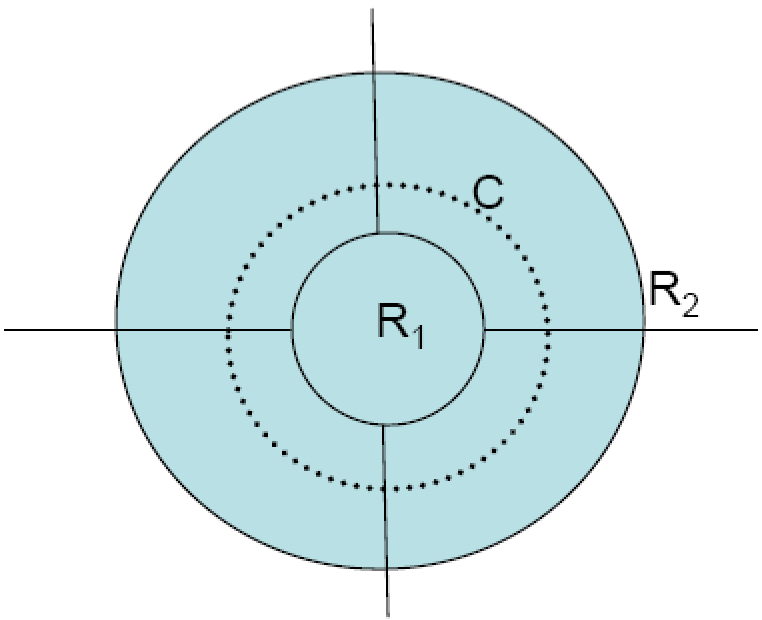
$$\frac{dX(z)}{dz} = -\sum_{n=-\infty}^{\infty} nx[n]z^{-n-1} = -\frac{1}{z}\sum_{n=-\infty}^{\infty} nx[n]z^{-n}$$

$$= -\frac{1}{z} Z(nx[n])$$

Inverse z-Transform

$$x[n] = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz$$

where C is a closed contour that encircles the origin of the z-plane and lies inside the region of convergence



for $X(z)$ rational(有理), can use a partial fraction expansion (部分分式展开) for finding inverse transforms

Partial Fraction Expansion

$$H(z) = \frac{b_0 z^M + b_1 z^{M-1} + \ldots + b_M}{z^N + a_1 z^{N-1} + \ldots + a_N}$$

$$= \frac{b_0 z^M + b_1 z^{M-1} + \ldots + b_M}{(z - p_1)(z - p_2)\ldots(z - p_N)}; \quad (N \geq M)$$

$$H(z) = \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2} + \ldots + \frac{A_N}{z - p_N}$$

$$\frac{H(z)}{z} = \frac{A_0}{z - p_0} + \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2} + \ldots + \frac{A_N}{z - p_N}; \; p_0 = 0$$

$$A_i = (z - p_i)\frac{H(z)}{z}\bigg|_{z = p_i} \quad i = 0, 1, \ldots, N$$

Example of Partial Fractions

Find the inverse z-transform of $H(z) = \dfrac{z^2+z+1}{(z^2+3z+2)} \quad 1<|z|<2$

$$\frac{H(z)}{z} = \frac{z^2+z+1}{z(z+1)(z+2)} = \frac{A_0}{z} + \frac{A_1}{z+1} + \frac{A_2}{z+2}$$

$$A_0 = \left.\frac{z^2+z+1}{(z+1)(z+2)}\right|_{z=0} = \frac{1}{2} \qquad A_1 = \left.\frac{z^2+z+1}{z(z+2)}\right|_{z=-1} = -1$$

$$A_2 = \left.\frac{z^2+z+1}{z(z+1)}\right|_{z=-2} = \frac{3}{2}$$

$$H(z) = \frac{1}{2} - \frac{z}{z+1} + \frac{(3/2)z}{z+2} \quad 1<|z|<2$$

$$h[n] = \frac{1}{2}\delta[n] - (-1)^n u[n] - \frac{3}{2}(-2)^n u[-n-1]$$

Transform Properties

Linearity	$ax_1[n]+bx_2[n]$	$aX_1(z)+bX_2(z)$
Shift	$x[n-n_0]$	$z^{-n_0}X(z)$
Exponential Weighting	$a^n x[n]$	$X(a^{-1}z)$
Linear Weighting	$nx[n]$	$-z\mathrm{d}X(z)/\mathrm{d}z$
Time Reversal	$x[-n]$	$X(z^{-1})$
Convolution	$x[n]*h[n]$	$X(z)H(z)$
Multiplication of Sequences	$x[n]w[n]$	$\frac{1}{2\pi j}\oint_C X(v)W(z/v)v^{-1}dv$

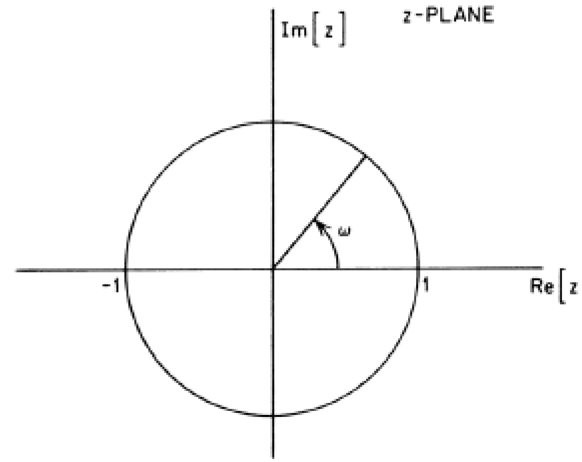
Discrete-Time Fourier Transform (DTFT)

Discrete-Time Fourier Transform

- evaluation of $X(z)$ on the unit circle in the z-plane

$$X(e^{j\omega}) = X(z)\Big|_{z=e^{j\omega}} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

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



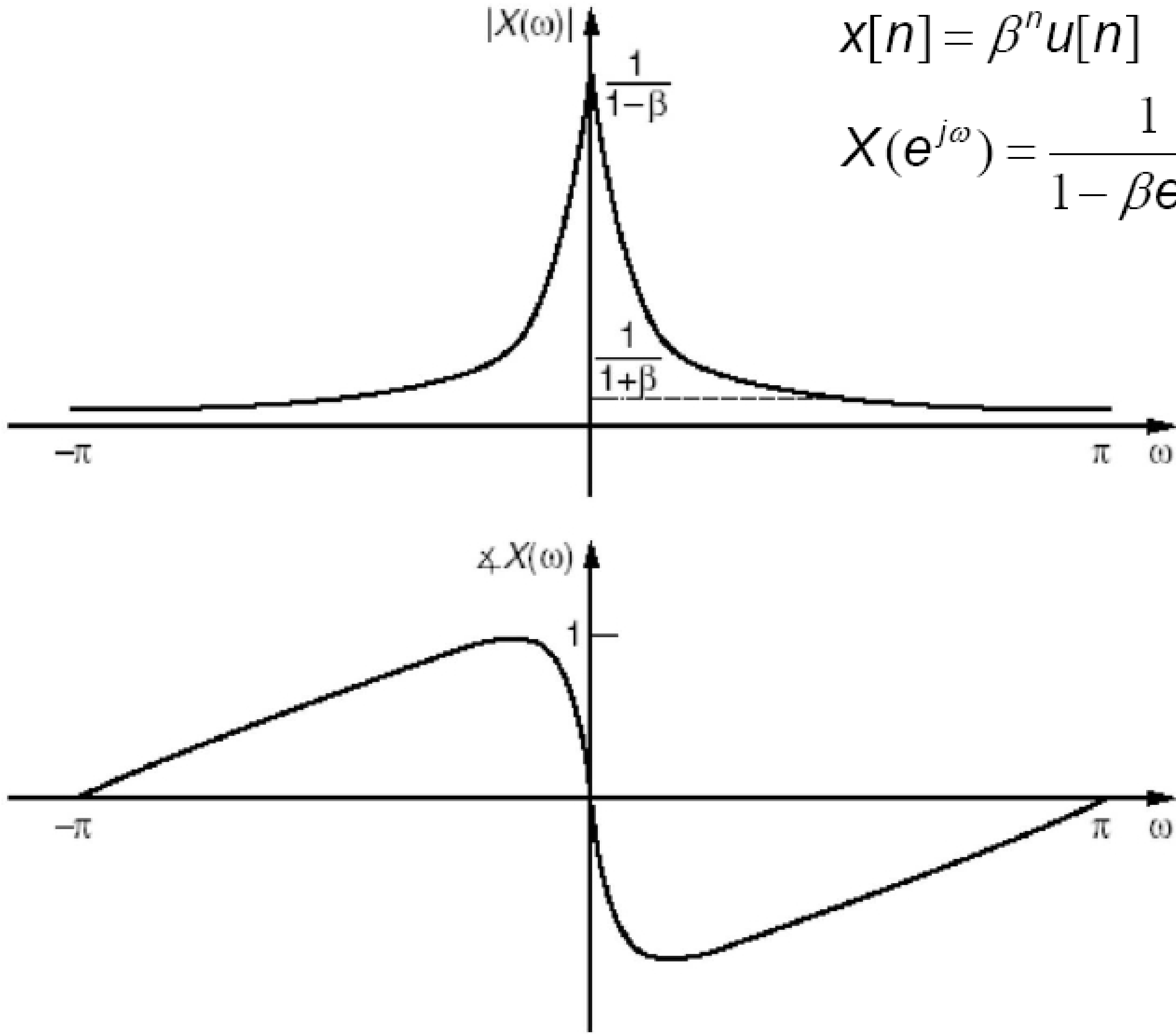
- sufficient condition for existence of Fourier transform is

$$\sum_{n=-\infty}^{\infty} |x[n]||z^{-n}| = \sum_{n=-\infty}^{\infty} |x[n]| < \infty, \text{ since } |z| = 1$$

Simple DTFTs

Impulse	$x[n] = \delta[n],$	$X(e^{j\omega}) = 1$
Delayed impulse	$x[n] = \delta[n - n_0],$	$X(e^{j\omega}) = e^{-j\omega n_0}$
Step function	$x[n] = u[n],$	$X(e^{j\omega}) = \dfrac{1}{1 - e^{-j\omega}}$
Rectangular window	$x[n] = u[n] - u[n - N],$	$X(e^{j\omega}) = \dfrac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$
Exponential	$x[n] = a^n \ u[n],$	$X(e^{j\omega}) = \dfrac{1}{1 - ae^{-j\omega}}, a < 1$
Backward exponential	$x[n] = -b^n \ u[-n-1],$	$X(e^{j\omega}) = \dfrac{1}{1 - be^{-j\omega}}, b > 1$

DTFT Examples



$$x[n] = \beta^n u[n] \quad \beta = 0.9$$

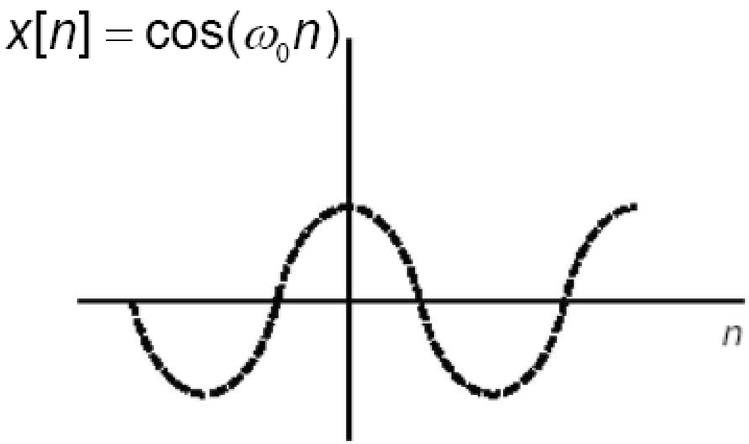
$$X(e^{j\omega}) = \frac{1}{1-\beta e^{-j\omega}} \quad |\beta| < 1$$

DTFT Examples

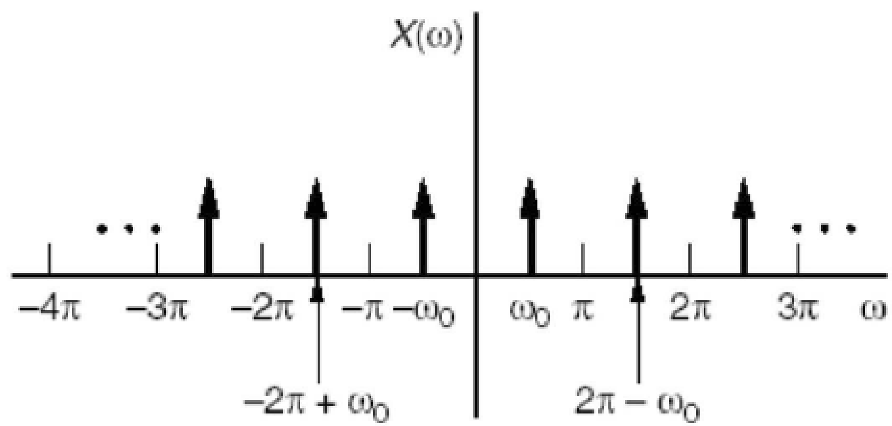
$$x[n] = \cos(\omega_0 n), \qquad -\infty < n < \infty$$

$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} [\pi\delta(\omega - \omega_0 + 2\pi k) + \pi\delta(\omega + \omega_0 + 2\pi k)]$$

Within interval $-\pi < \omega < \pi$, $X(e^{j\omega})$ is comprised of a pair of impulses at $\pm\omega_0$

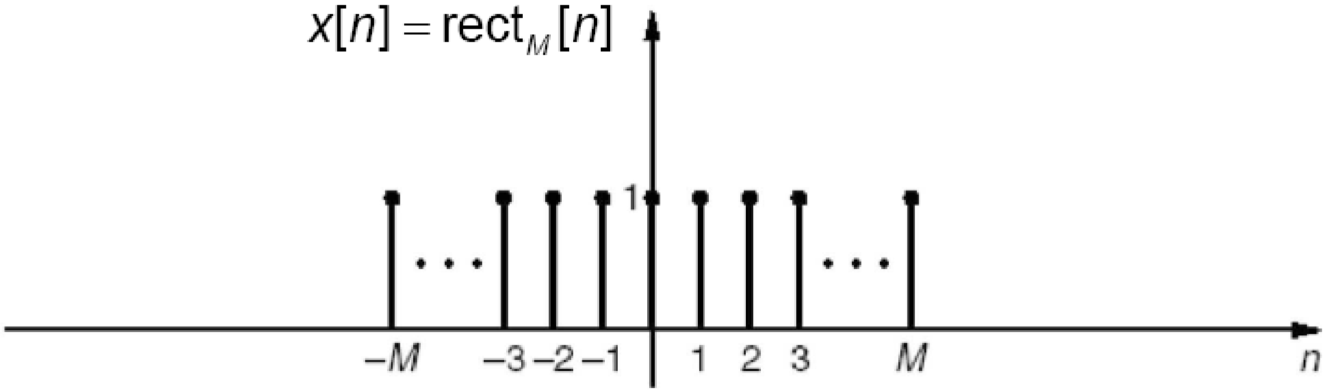


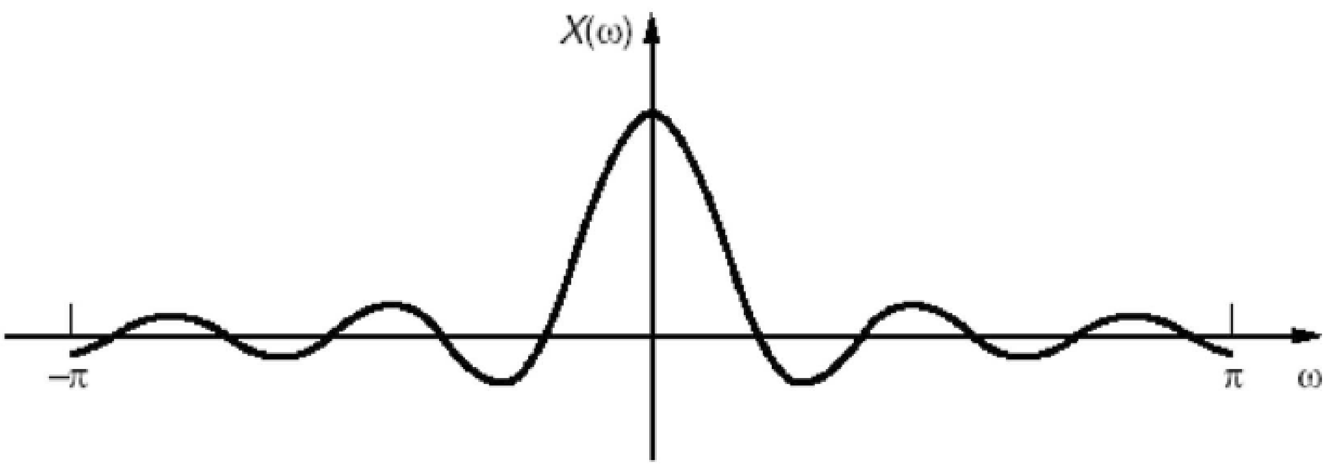
(a)



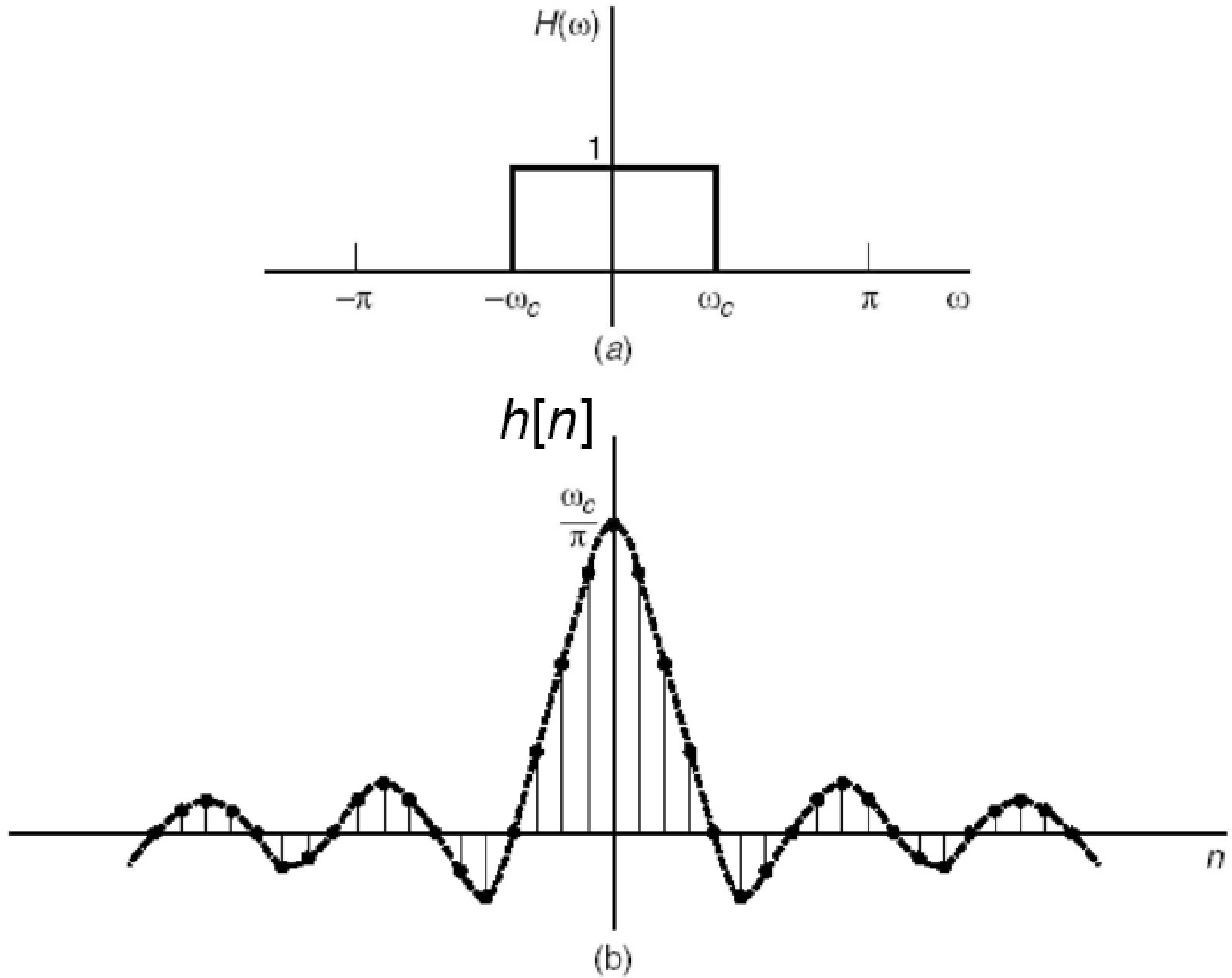
(b)

DTFT Examples

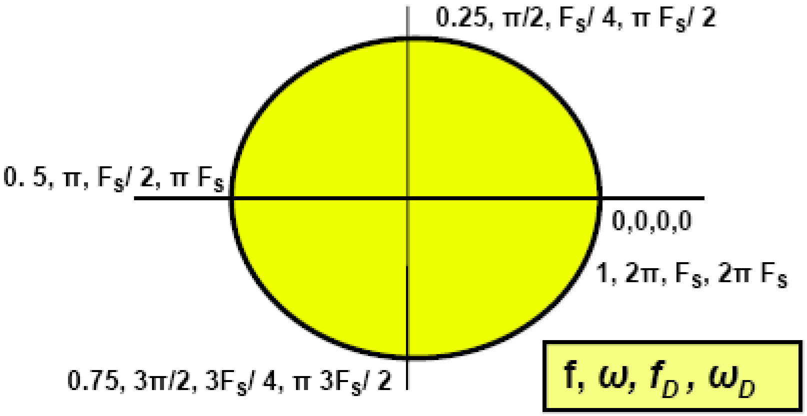




DTFT Examples



Fourier Transform Properties



- Periodicity in ω

$$X(e^{j\omega}) = X(e^{j(\omega+2\pi k)})$$

- Period of 2π corresponds to once around unit circle in the z-plane

• normalized frequency: f, 0→0.5 →1 (independent of Fs)

• normalized radian frequency: ω, 0→π →2π (independent of Fs)

• digital frequency : f_D=f*Fs, 0→0.5Fs →Fs

• digital radian frequency : ω_D= ω*Fs, 0→πFs→2πFs

Discrete Fourier Transform (DFT)

Discrete-Time Fourier Series

- consider a periodic signal with period N (samples)

$$\tilde{x}[n] = \tilde{x}[n+N], -\infty < n < \infty$$

$\tilde{x}[n]$ can be represented exactly by a discrete sum of sinusoids

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi kn/N}$$

- *N* Fourier series coefficients

$$\tilde{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j2\pi kn/N}$$

- *N*-sequence values

Finite Length Sequences

- consider a finite length (but not periodic) sequence, $x[n]$, that is zero outside the interval $0 \le n \le N-1$

$$X(z) = \sum_{n=0}^{N-1} x[n] z^{-n}$$

- evaluate $X(z)$ at equally spaced points on the unit circle,

$$z_k = e^{j2\pi k/N}, k = 0, 1, \ldots, N-1$$

$$X[k] = X(e^{j2\pi k/N}) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, k = 0, 1, \ldots, N-1$$

  - looks like discrete-time Fourier series of periodic sequence

Relation to Periodic Sequence

- consider a periodic sequence, $\tilde{x}[n]$, consisting of an infinite sequence of replicas of $x[n]$

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n+rN]$$

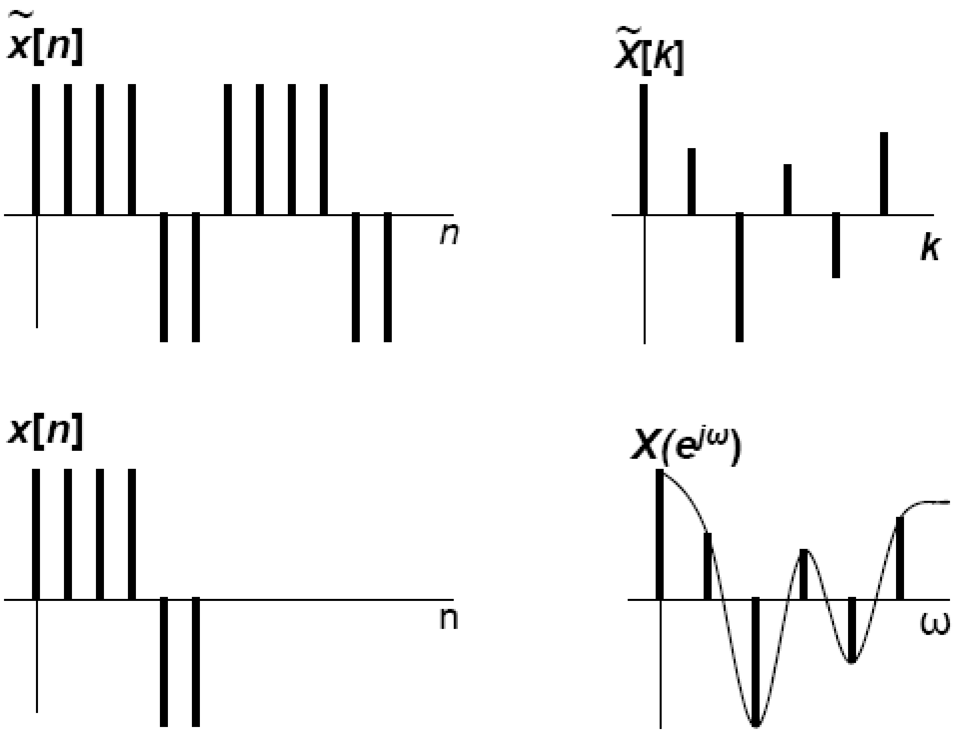
- The Fourier coefficients, $\tilde{X}[k]$, are then identical to the values of $X(e^{j2\pi k/N})$ for the finite duration sequence $\Rightarrow$

<u>a sequence of N length can be exactly represented by a DFT representation of the form</u>

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, 0 \le k \le N-1$$

$$x[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[k]e^{j2\pi kn/N}, 0 \le n \le N-1$$

Periodic and Finite Length Sequences



periodic signal => line spectrum in frequency

finite duration => continuous spectrum in frequency

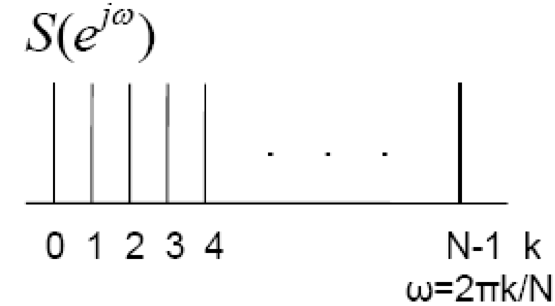
Sampling in Frequency (Time Domain Aliasing)

Consider a finite duration sequence:

$x[n] \neq 0$ for $0 \leq n \leq L-1$

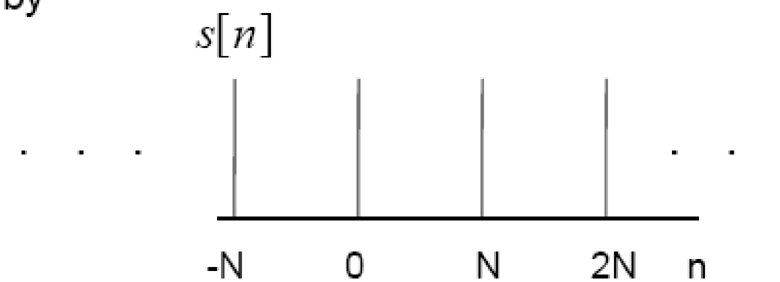
i.e., an $L-$point sequence, with discrete time Fourier transform

$$X(e^{j\omega}) = \sum_{n=0}^{L-1} x[n] e^{-j\omega n} \quad 0 \leq \omega \leq 2\pi$$



Consider sampling the discrete time Fourier transform by multiplying it by a signal that is defined as:

$$S(e^{j\omega}) = \sum_{k=0}^{N-1} \delta[\omega - 2\pi k / N]$$



with time-domain representation

$$s[n] = \sum_{r=-\infty}^{\infty} \delta[n - rN]$$

Thus we form the spectral sequence

$$\tilde{X}(e^{j\omega}) = X(e^{j\omega}) \cdot S(e^{j\omega})$$

which transforms in the time domain to the convolution

$$\tilde{x}[n] = x[n] * s[n] = x[n] * \sum_{r=-\infty}^{\infty} \delta[n - rN] = \sum_{r=-\infty}^{\infty} x[n - rN]$$

$$\tilde{x}[n] = x[n] + x[n-N] + x[n+N] + \ldots$$

Sampling in Frequency (Time Domain Aliasing)

If the duration of the finite duration signal satisfies the relation $N \geq L,$ then only the first term in the infinite summation affects the interval $0 \leq n \leq L-1$ and there is no time domain aliasing, i.e.,

$$\tilde{x}[n] = x[n] \quad 0 \leq n \leq L-1$$

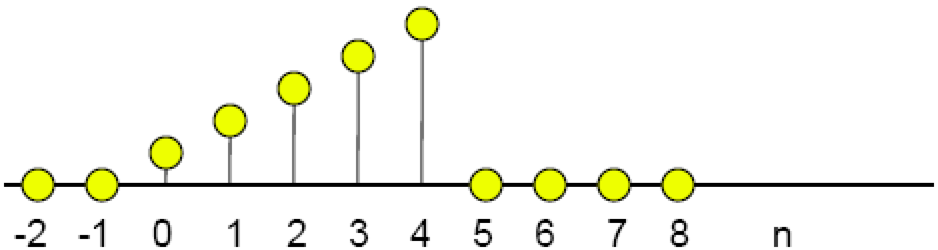
If $N < L,$ i.e., the number of frequency samples is smaller than the duration of the finite duration signal, then there is time domain aliasing and the resulting aliased signal (over the interval $0 \leq n \leq L-1$) satisfies the aliasing relation:

$$\tilde{x}[n] = x[n] + x[n+N] + x[n-N] \quad 0 \leq n \leq N-1$$

Time Domain Aliasing Example

Consider the finite duration sequence

$$x[n] = \sum_{m=0}^{4} (m+1)\,\delta[n-m] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3] + 5\delta[n-4]$$



The discrete time Fourier transform of $x[n]$ is computed and sampled at N frequencies around the unit circle. The resulting sampled Fourier transform is inverse transformed back to the time domain. What is the resulting time domain signal, $\tilde{x}[n]$, (over the interval $0 \le n \le L-1$) for the cases $N=11$, $N=5$ and $N=4$.

SOLUTION:

For the cases $N=11$ and $N=5$, we have no aliasing (since $N \ge L$) and we get $\tilde{x}[n] = x[n]$ over the interval $0 \le n \le L-1$. For the case $N=4$, the $n=0$ value is aliased, giving $\tilde{x}[0]=6$ (as opposed to 1 for $x[0]$) with the remaining values unchanged.

DFT Properties

Periodic Sequence	Finite Sequence
Period = N	Length = N
Sequence defined for all n	Sequence defined for n = 0,1,…, N-1
DFS defined for k = 0,1,…, N-1	DTFT defined for all ω

- when using DFT representation, <u>all sequences behave as if they were infinitely periodic</u> $\Rightarrow$ DFT is really the representation of the extended periodic function $\tilde{x}[n] = \sum_{-\infty}^{\infty} x[n + rN]$

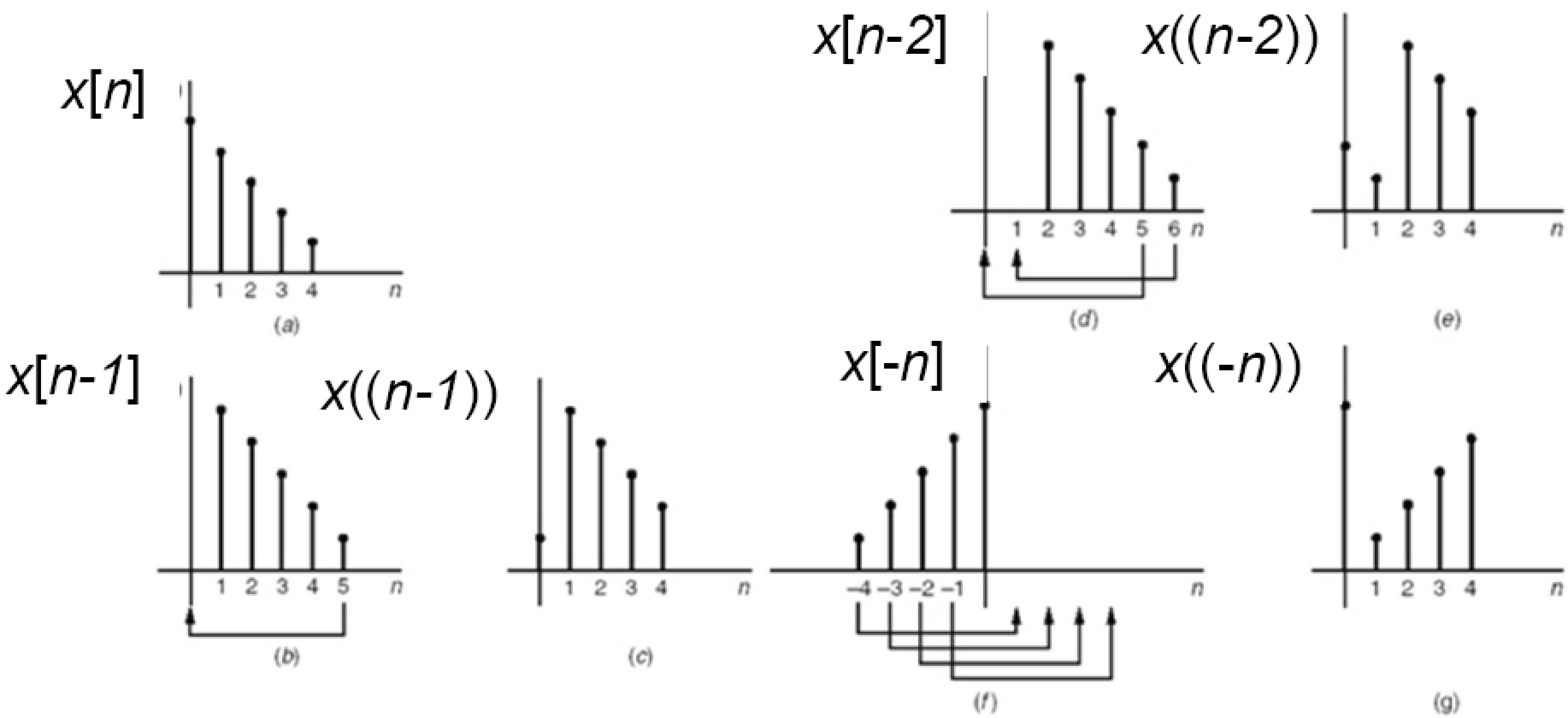
DFT Properties for Finite Sequences

- $X[k]$, the DFT of the finite sequence $x[n]$, can be viewed as a sampled version of the z-transform (or Fourier transform) of the finite sequence (used to design finite length filters via frequency sampling method)
- the DFT has properties very similar to those of the z-transform and the Fourier transform
- the N values of $X[k]$ can be computed very efficiently (time proportional to $N \log N$) using the set of FFT methods
- DFT used in computing spectral estimates, correlation functions, and in implementing digital filters via convolutional methods

DFT Properties

	N-point sequences	N-point DFT
Linearity	$ax_1[n]+bx_2[n]$	$aX_1[k]+bX_2[k]$
Shift	$x([n-n_0])_N$	$e^{-j2\pi kn_0/N}X[k]$
Time Reversal	$x([-n])_N$	$X^*[k]$
Convolution	$\sum_{m=0}^{N-1} x[m]h([n-m])_N$	$X[k]H[k]$
Multiplication	$x[n]w[n]$	$\frac{1}{N}\sum_{r=0}^{N-1} X[r]W([k-r])_N$

Circular Shifting Sequences



Digital Filters

Digital Filters

- digital filter is a discrete-time linear, shift invariant system with input-output relation

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

$$Y(z) = X(z) \cdot H(z)$$

- $H(z)$ is the system function (系统函数) with $H(e^{j\omega})$ as the complex frequency response (频率响应)

$$H(e^{j\omega}) = H_r(e^{j\omega}) + jH_i(e^{j\omega})$$

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\arg[H(e^{j\omega})]}$$

$$\log H(e^{j\omega}) = \log\left|H(e^{j\omega})\right| + j\arg[H(e^{j\omega})]$$

$$\log\left|H(e^{j\omega})\right| = \mathrm{Re}[\log H(e^{j\omega})]$$

$$\arg[H(e^{j\omega})] = \mathrm{Im}[\log H(e^{j\omega})]$$

Digital Filters

- causal linear shift-invariant

  $\Rightarrow h[n]$=0 for n<0
- stable system

  $\Rightarrow$ every bounded input produces a bounded output

  $\Rightarrow$ a necessary and sufficient condition for stability and for the existence of $H(e^{j\omega})$

$$\sum_{n=-\infty}^{\infty} \left| h[n] \right| < \infty$$

Digital Filters

- input and output satisfy linear difference equation (线性差分方程) of the form

$$y[n]-\sum_{k=1}^{N}a_k y[n-k]=\sum_{r=0}^{M}b_r x[n-r]$$

- evaluating z-transforms of both sides gives:

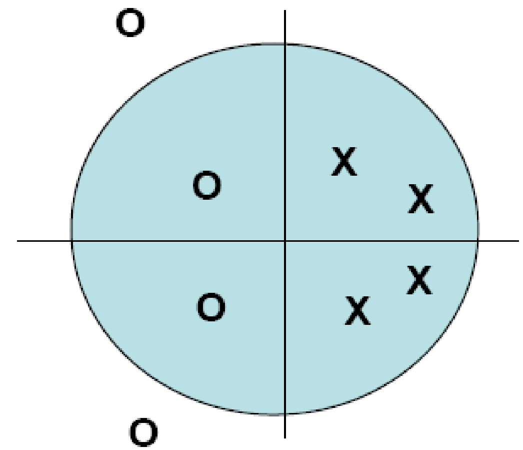
$$Y(z)-\sum_{k=1}^{N}a_k z^{-k}Y(z)=\sum_{r=0}^{M}b_r z^{-r}X(z)$$

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{r=0}^{M}b_r z^{-r}}{1-\sum_{k=1}^{N}a_k z^{-k}}$$

Digital Filters

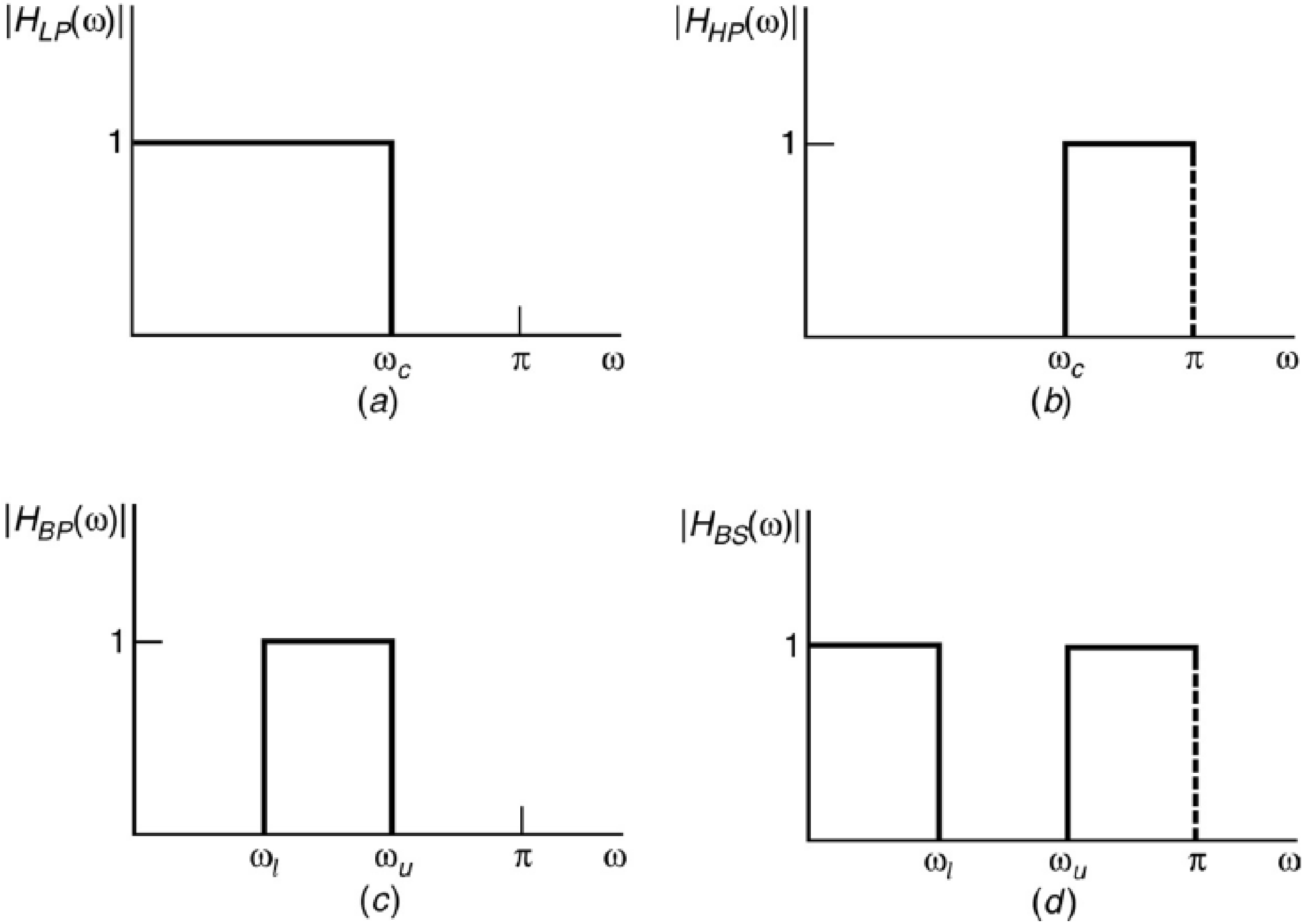
- $H(z)$ is a rational function of z^{-1} with M zeros and N poles

$$H(z) = \frac{A\prod_{r=1}^{M}(1-c_r z^{-1})}{\prod_{k=1}^{N}(1-d_k z^{-1})}$$



- converges for $|z|>R_1$, with $R_1<1$ for stability $\Rightarrow$
  <u>all poles of $H(z)$ inside the unit circle for a stable, causal system</u>

Ideal Filter Responses



FIR System

- If a_k=0, all k, then

$$y[n] = \sum_{r=0}^{M} b_r x[n-r] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M] \Rightarrow$$

1) $$h[n] = b_n \quad 0 \le n \le M$$
$$= 0 \quad \text{otherwise}$$

2) $$H(z) = \sum_{n=0}^{M} b_n z^{-n} = \prod_{m=1}^{M} (1 - c_m z^{-1}) \Rightarrow M$$ zeros

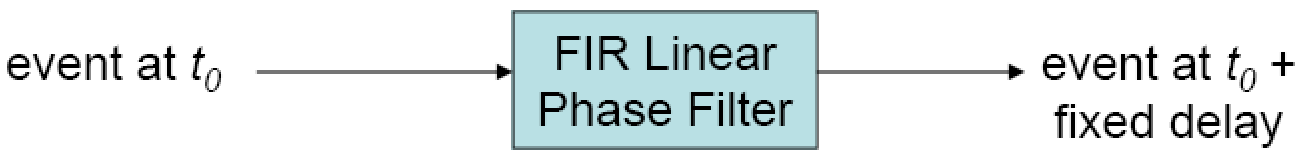
3) if $h[n] = \pm h[M-n]$ (symmetric, anti-symmetric)

$$H(e^{j\omega}) = A(e^{j\omega}) e^{-j\omega M/2}$$

$A(e^{j\omega}) =$ real(symmetric), imaginary(anti-symmetric)

Linear Phase Filter

- no signal dispersion (散布) because of non-linear phase ⇒ precise time alignment of events in signal



FIR Filters

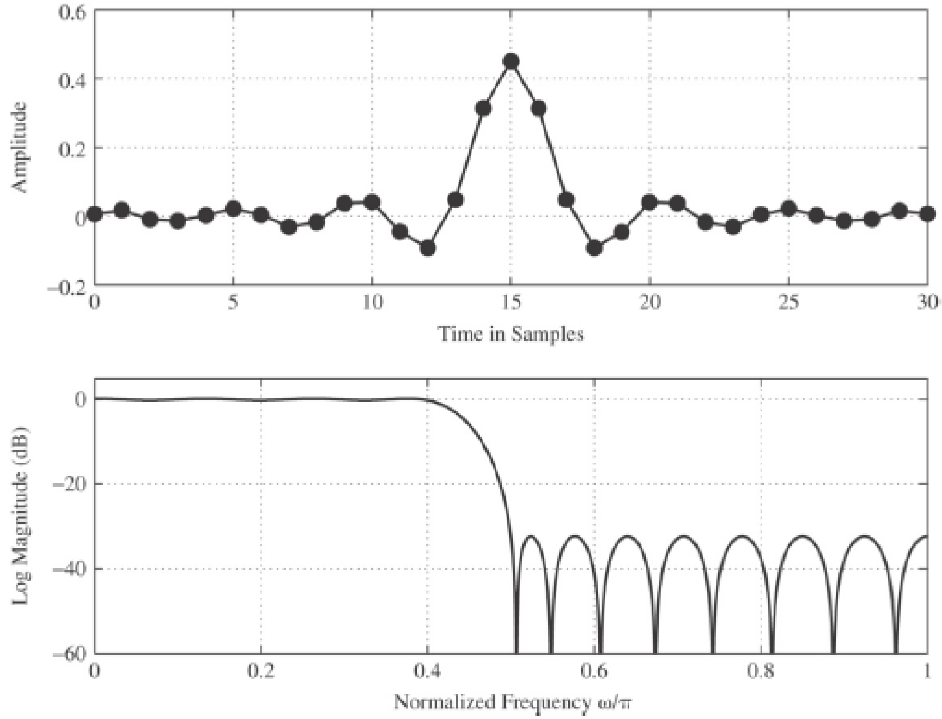
- cost of linear phase filter designs
  - can theoretically approximate any desired response to any degree of accuracy
  - requires longer filters than non-linear phase designs

- FIR filter design methods
  - window approximation $\Rightarrow$ analytical, closed form method
  - frequency sampling approximation $\Rightarrow$ optimization method
  - optimal (minimax error) approximation $\Rightarrow$ optimization method

Matlab FIR Design

1. Use **fdatool** to design digital filters
2. Use **firpm** to design FIR filters
   - B=firpm(N,F,A)
   - N+1 point linear phase, FIR design
   - B=filter coefficients (numerator polynomial)
   - F=ideal frequency response band edges (in pairs) (normalized to 1.0)
   - A=ideal amplitude response values (in pairs)
3. Use **freqz** to convert to frequency response (complex)
   - [H,W]=freqz(B,den,NF)
   - H=complex frequency response
   - W=set of radian frequencies at which FR is evaluated (0 to pi)
   - B=numerator polynomial=set of FIR filter coefficients
   - den=denominator polynomial=[1] for FIR filter
   - NF=number of frequencies at which FR is evaluated
4. Use **plot** to evaluate log magnitude response
   - plot(W/pi, 20log10(abs(H)))

Lowpass Filter Design Example

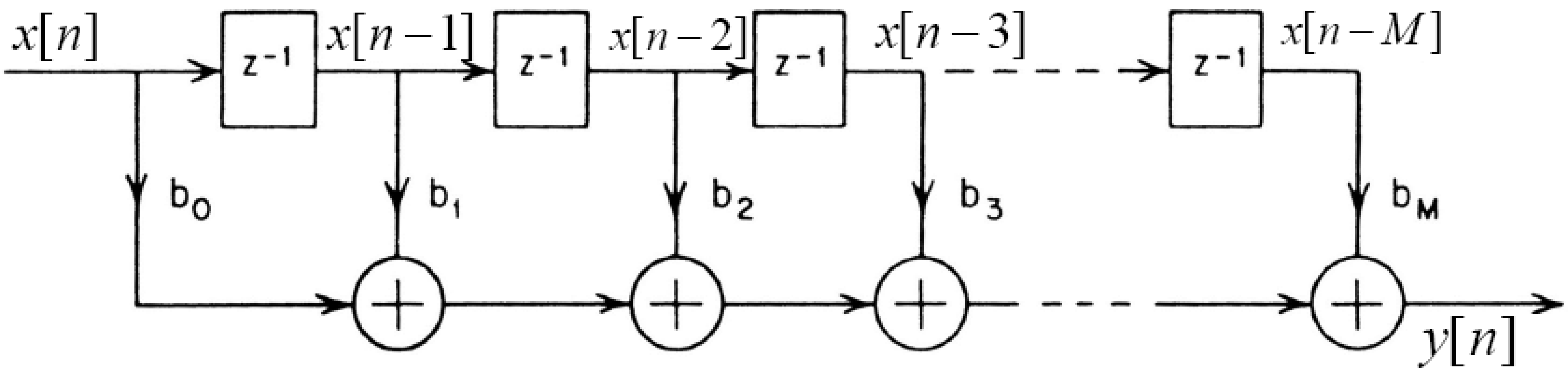



```
N=30
F=[0 0.4 0.5 1];
A=[1 1 0 0];
B=firpm(N,F,A)

NF=512;  number of frequency points
[H,W]=freqz(B,1,NF);

plot(W/pi,20log10(abs(H)));
```

FIR Implementation



- linear phase filters can be implemented with half the multiplications (because of the symmetry of the coefficients)

IIR Systems

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x[n-r]$$

- $y[n]$ depends on $y\,[n\text{-}1]$,…, $y[n\text{-}N]$ as well as $x[n]$, …, $x[n\text{-}M]$
- for $M<N$

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{1-\sum_{k=1}^{N} a_k z^{-k}} = \sum_{k=1}^{N} \frac{A_k}{1-d_k z^{-1}}$$ (partial fraction expansion)

$$h[n] = \sum_{k=1}^{N} A_k (d_k)^n u[n]$$ (for casual systems)

<u>an infinite duration impulse response</u>

IIR Filters

- IIR filter issues
  - efficient implementations in terms of computations
  - can approximate any desired magnitude response with **arbitrarily small error**
  - non-linear phase $\Rightarrow$**time dispersion of waveform**

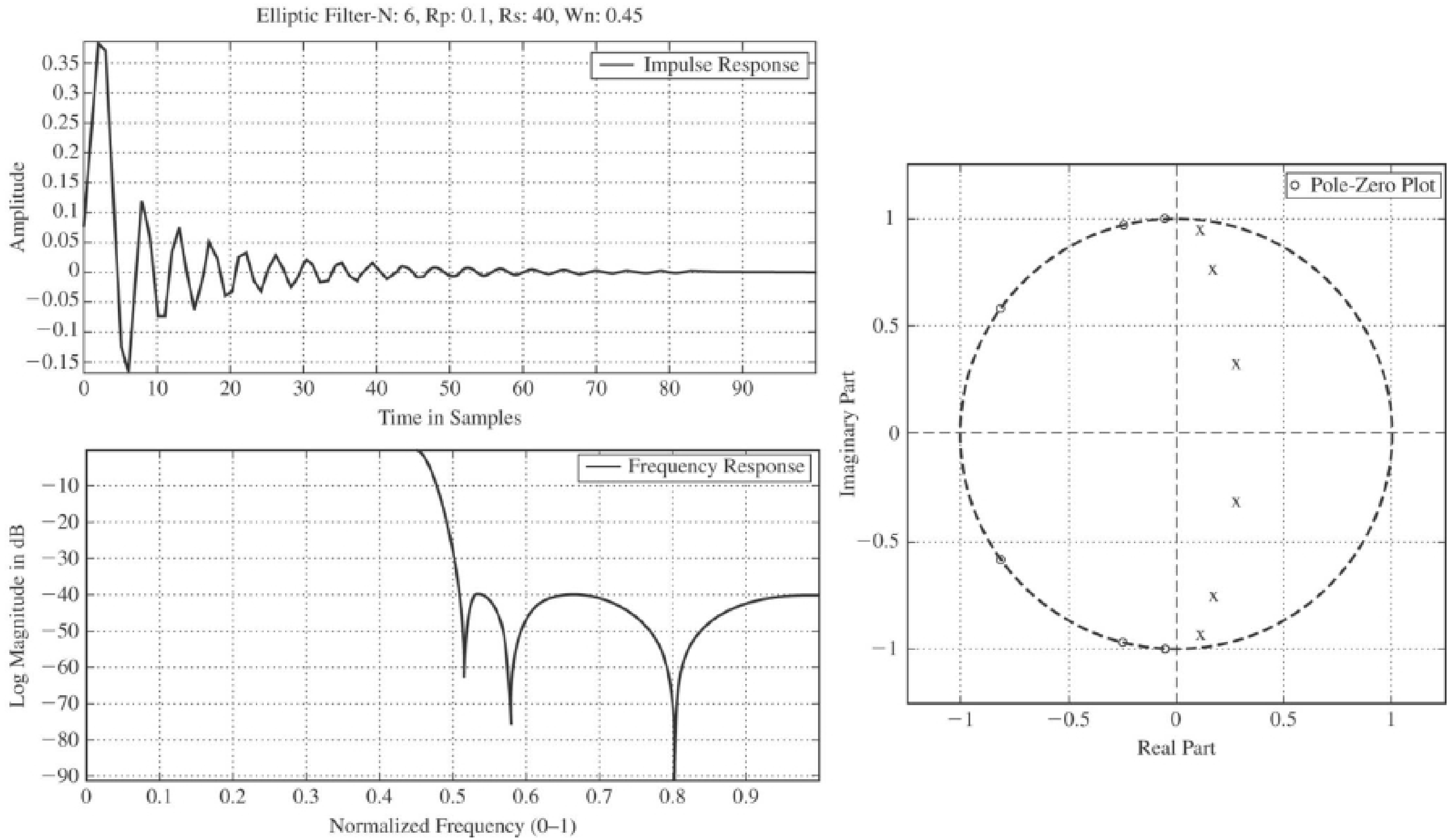
IIR Design Methods

- Analog filter design
  - Butterworth designs: maximally flat amplitude
  - Bessel designs: maximally flat group delay
  - Chebyshev designs: equi-ripple in either passband or stopband
  - Elliptic designs: equi-ripple in both passband and stopband
- Transform to digital filter
  - Impulse invariant transformation 冲击不变法
    - match the analog impulse response by sampling
    - resulting frequency response is aliased version of analog frequency response
  - Bilinear transformation 双线性变换法
    - use a transformation to map an analog filter to a digital filter by warping the analog frequency scale (0 to infinity) to the digital frequency scale (0 to pi)
    - use frequency pre-warping to preserve critical frequencies of transformation (i.e., filter cutoff frequencies)

Matlab Elliptic Filter Design

- use **ellip** to design elliptic filter
  - [B,A]=ellip(N,Rp,Rs,Wn)
  - B=numerator polynomial—N+1 coefficients
  - A=denominator polynomial—N+1 coefficients
  - N=order of polynomial for both numerator and denominator
  - Rp=maximum in-band (passband) approximation error (dB)
  - Rs=out-of-band (stopband) ripple (dB)
  - Wp=end of passband (normalized radian frequency)
- use **filter** to generate impulse response
  - y=filter(B,A,x)
  - y=filter impulse response
  - x=filter input (impulse)
- use **zplane** to generate pole-zero plot
  - zplane(B,A)

Matlab Elliptic Filter Design




```
[b,a]=ellip(6,0.1,40,0.45);  [h,w]=freqz(b,a,512);  x=[1,zeros(1,511)];  y=filter(b,a,x);  zplane(b,a);
appropriate plotting commands;
```

IIR Filter Implementation

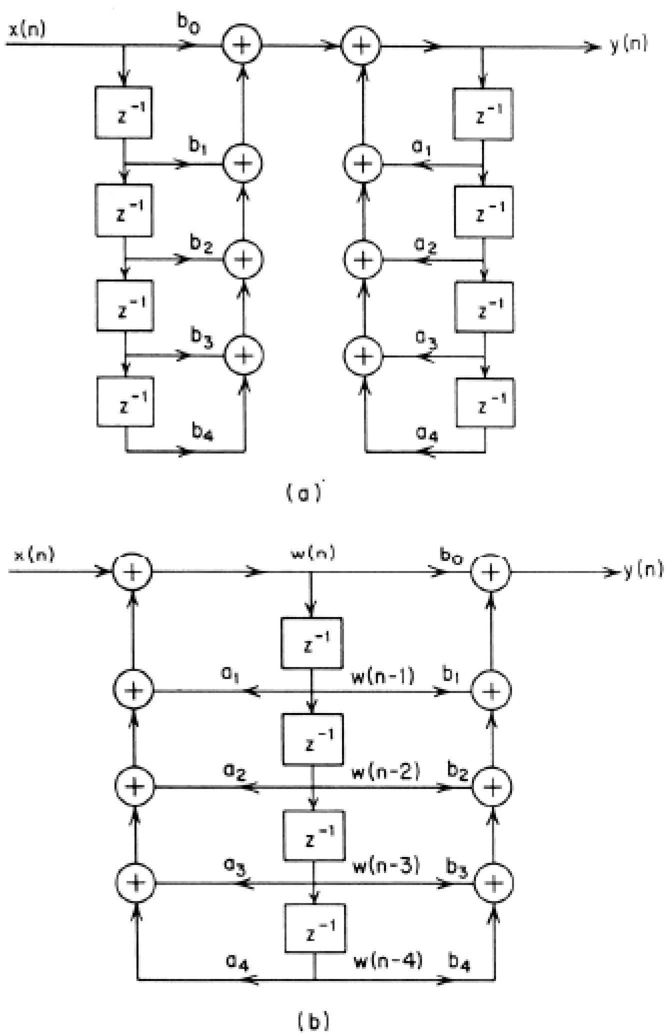


Fig. 2.6 (a) Direct form IIR structure; (b) direct form structure with minimum storage.

M=N=4

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x[n-r]$$

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$

$$y[n] = \sum_{r=0}^{M} b_r w[n-r]$$

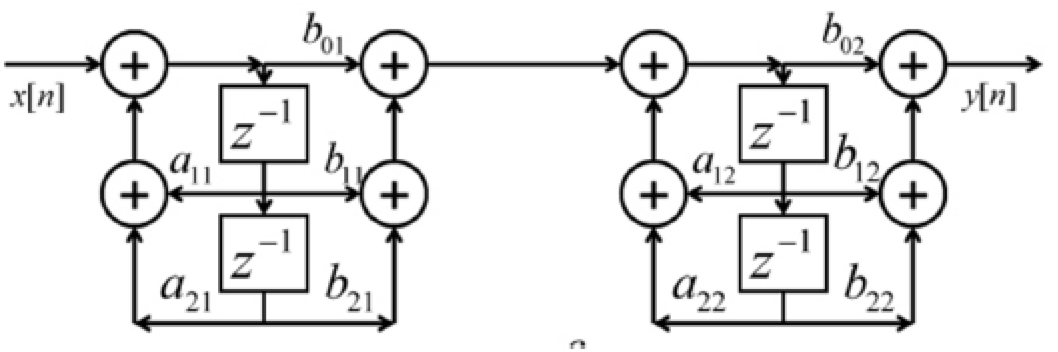
IIR Filter Implementation

$$H(z) = \frac{A\prod_{r=1}^{N}(1-c_r z^{-1})}{\prod_{k=1}^{N}(1-d_k z^{-1})}$$ - zeros at $z = c_r$, poles at $z = d_k$

- since a_k and b_r are real, poles and zeros occur in complex conjugate pairs $\Rightarrow$

$$H(z) = A\prod_{k=1}^{K}\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}}, \quad K = \left\lfloor \frac{N+1}{2} \right\rfloor$$

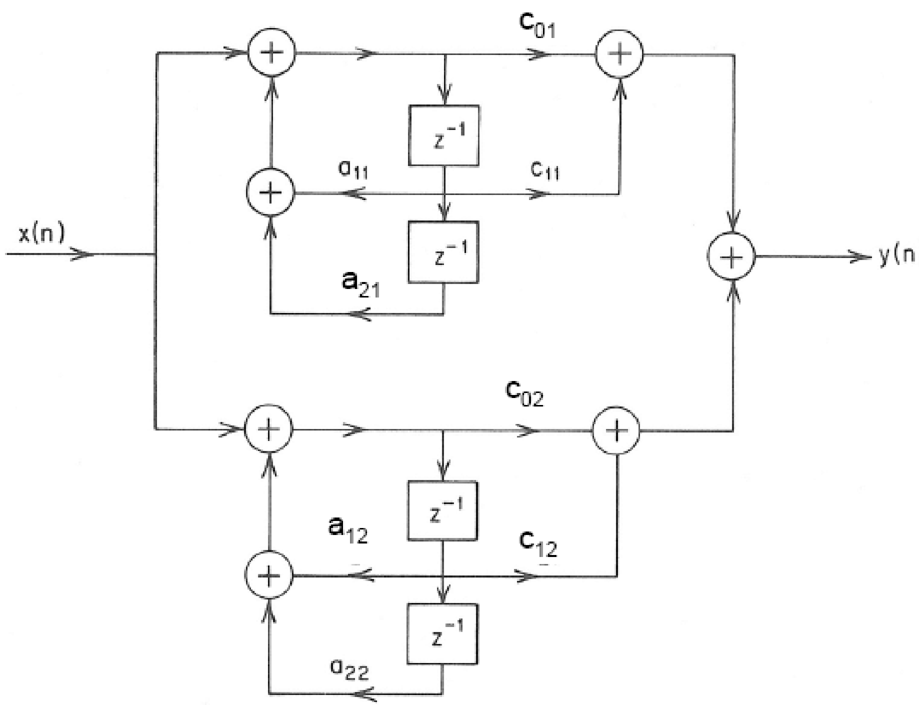
- cascade of second order systems



Used in formant synthesis systems based on ABS methods

IIR Filter Implementation

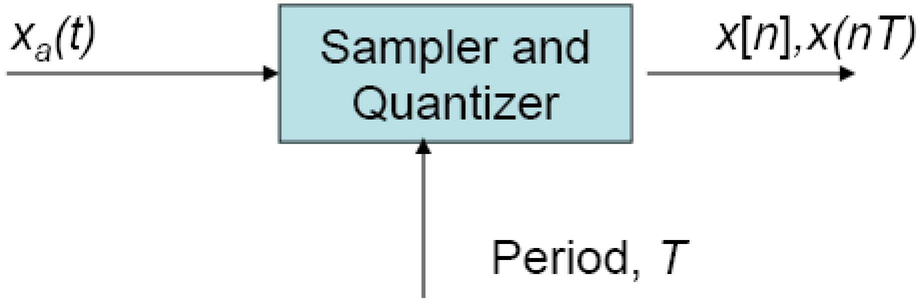
$$H(z)=\sum_{k=1}^{K}\frac{c_{0k}+c_{1k}z^{-1}}{1-a_{1k}z^{-1}-a_{2k}z^{-2}},\quad \text{parallel system}$$



Common form for speech synthesizer implementation

Sampling

Sampling of Waveforms



$$x[n] = x_a(nT), -\infty < n < \infty$$

$$F_s = 8000\text{Hz} \leftrightarrow T = 1/8000 = 125\mu\sec$$

$$F_s = 10000\text{Hz} \leftrightarrow T = 1/10000 = 100\mu\sec$$

$$F_s = 16000\text{Hz} \leftrightarrow T = 1/16000 = 62.5\mu\sec$$

$$F_s = 20000\text{Hz} \leftrightarrow T = 1/20000 = 50\mu\sec$$

The Sampling Theorem

If a signal $x_a(t)$ has a bandlimited Fourier transform $X_a(j\Omega)$ such that $X_a(j\Omega) = 0$ for $\Omega \geq 2\pi F_N$, then $x_a(t)$ can be uniquely reconstructed from equally spaced samples $x_a(nT)$, $-\infty < n < \infty$, if $1/T \geq 2F_N$ ($F_S \geq 2F_N$) (A-D or C/D converter)

Sampling Theorem Equations

$$x_a(t) \longleftrightarrow X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t)\, e^{-j\Omega t} dt$$

$$x[n] \longleftrightarrow X(e^{j\Omega T}) = \sum_{n=-\infty}^{\infty} x_a(nT) e^{-j\Omega nT}$$

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(j\Omega + j2\pi k / T)$$

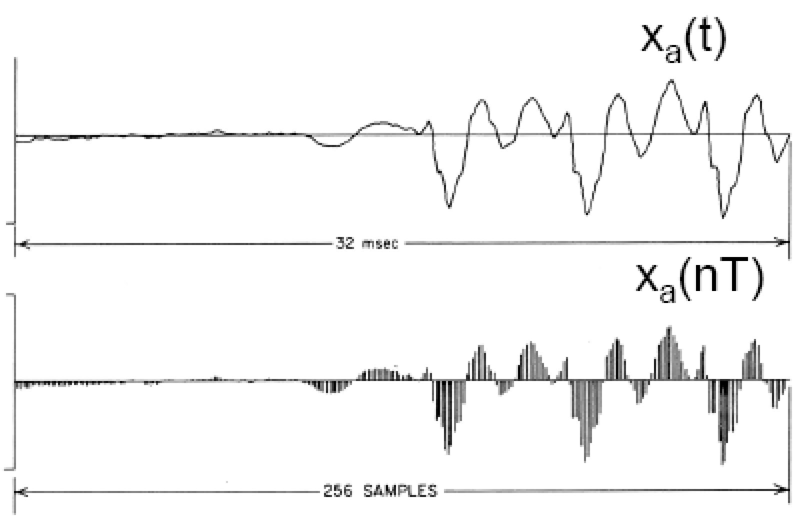
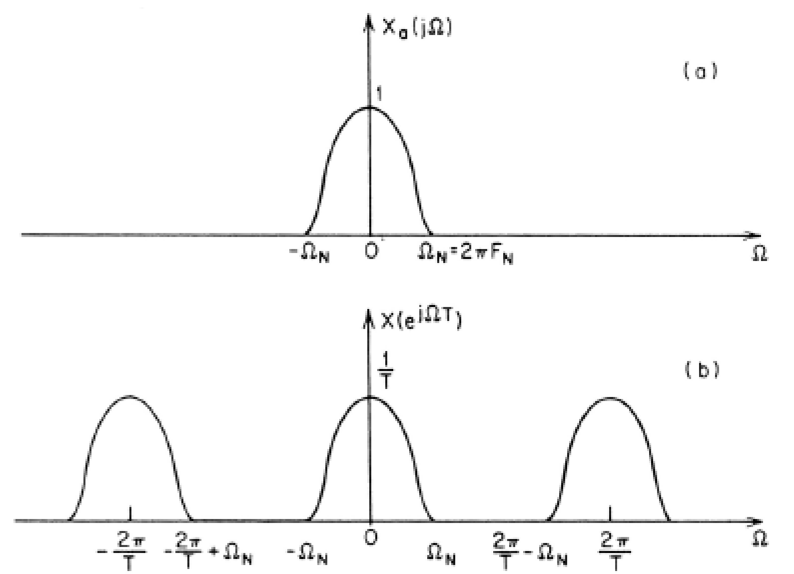
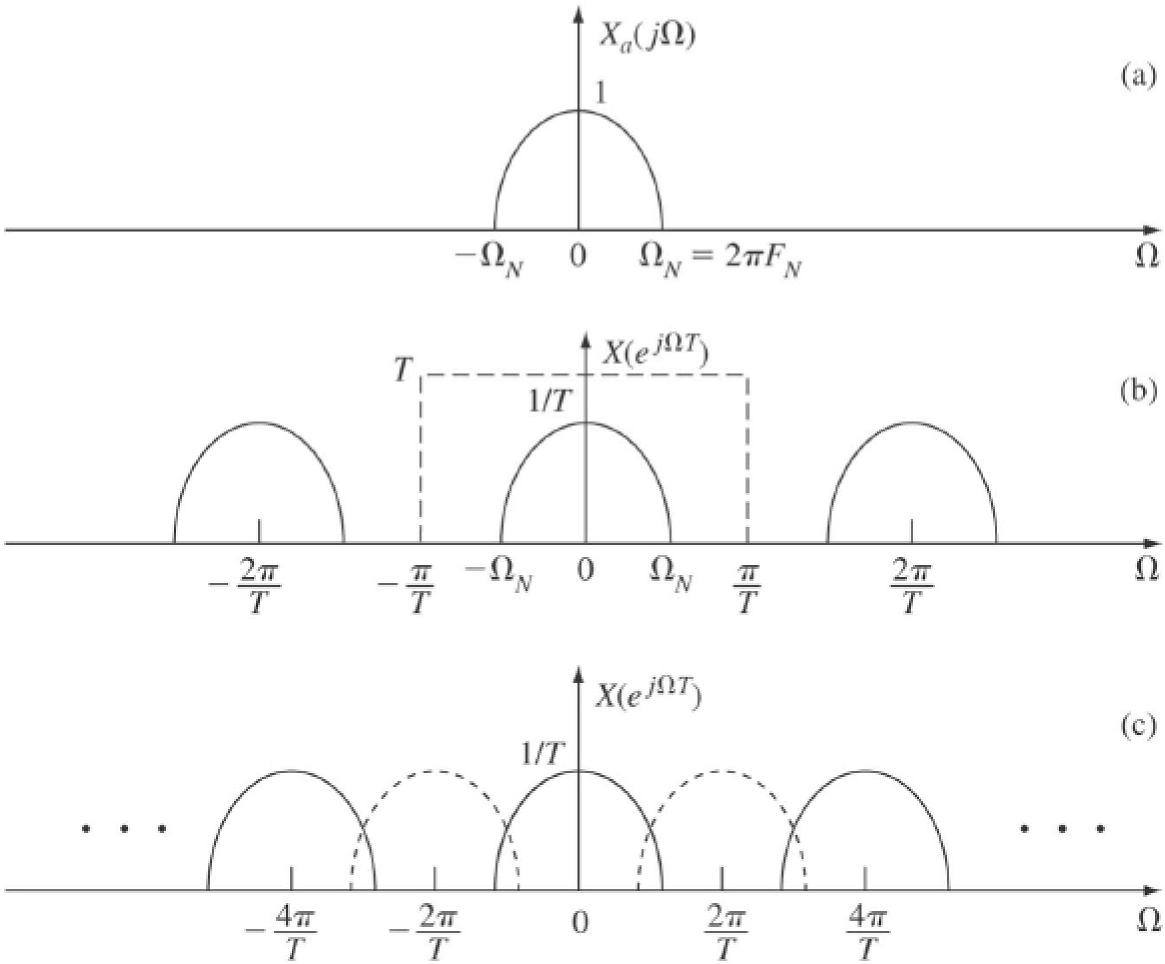


Fig. 2.1 Representations of a speech signal.



Sampling Theorem Interpretation



To avoid aliasing need:

$2\pi / T - \Omega_N > \Omega_N$

$\Rightarrow \; 2\pi / T > 2\,\Omega_N$

$\Rightarrow \; F_s = 1/T > 2F_N$

case where $1/T < 2F_N$, aliasing occurs

Sampling Rates

- F_N = Nyquist frequency (highest frequency with significant spectral level in signal)
- must sample at least twice the Nyquist frequency to prevent aliasing (frequency overlap)
  - telephone speech (300-3200 Hz)$\Rightarrow F_S$ =6400 Hz
  - wideband speech (100-7200 Hz) $\Rightarrow F_S$ =14400 Hz
  - audio signal (50-21000 Hz) $\Rightarrow F_S$ =42000 Hz
  - AM broadcast (100-7500 Hz) $\Rightarrow F_S$ =15000 Hz
- can always sample at rates higher than twice the Nyquist frequency (but that is wasteful of processing)

Recovery from Sampled Signal

- If $1/T > 2\,F_N$, the Fourier transform of the sequence of samples is proportional to the Fourier transform of the original signal in the baseband, i.e.,

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega), \; |\Omega| < \frac{\pi}{T}$$

- can show that the original signal can be recovered from the sampled signal by interpolation using an ideal LPF of bandwidth π / T, i.e.,

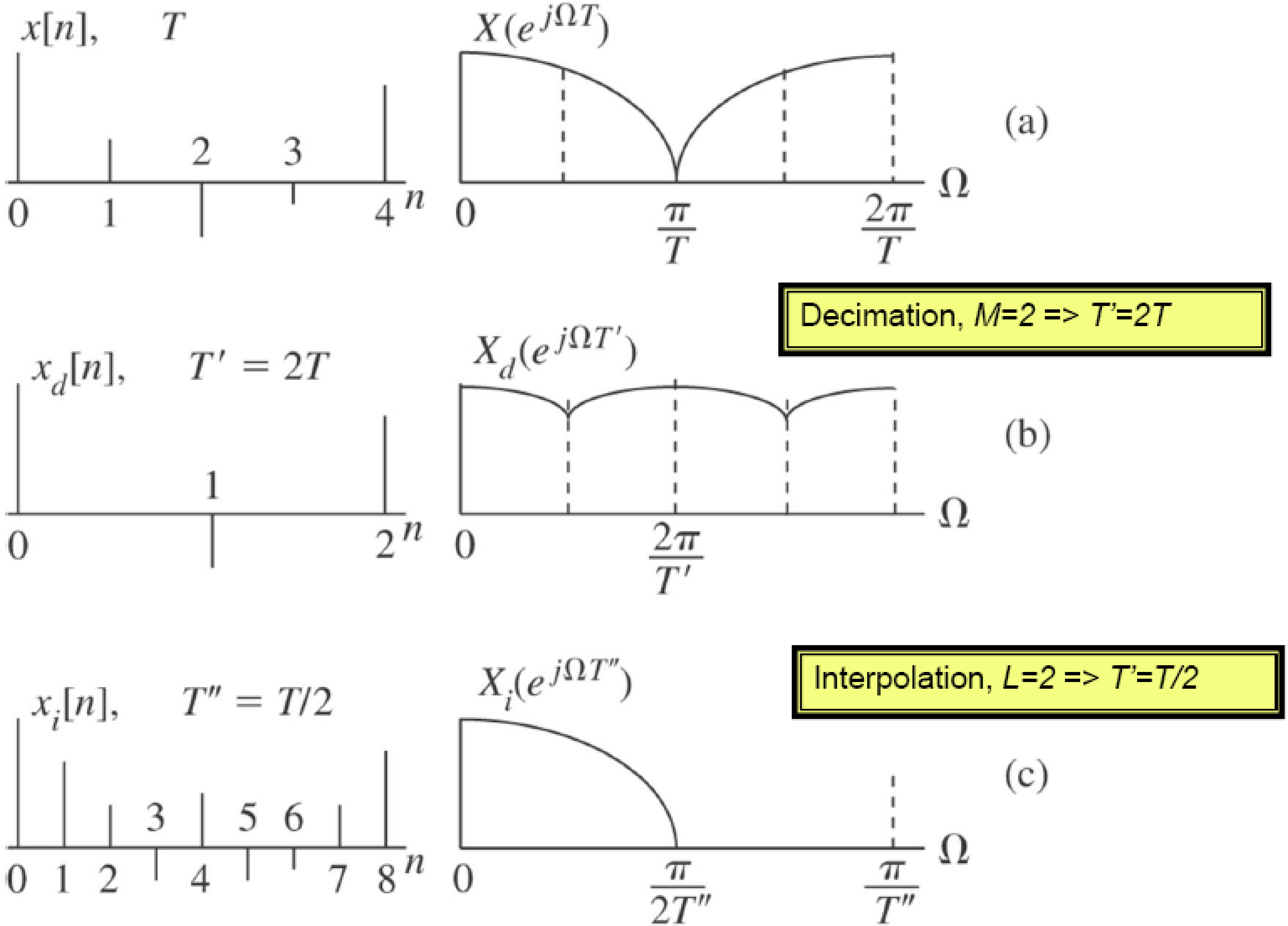
$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT) \left[\frac{\sin(\pi(t-nT)/T)}{\pi(t-nT)/T} \right]$$

  - digital-to-analog converter

Decimation(抽取) and Interpolation(内插) of Sampled Waveforms

- CD rate (44.06 kHz) to DAT rate (48 kHz)—media conversion
- Wideband (16 kHz) to narrowband speech rates (8kHz, 6.67 kHz)—storage
- oversampled to correctly sampled rates--coding

Decimation and Interpolation of Sampled Waveforms



Decimation

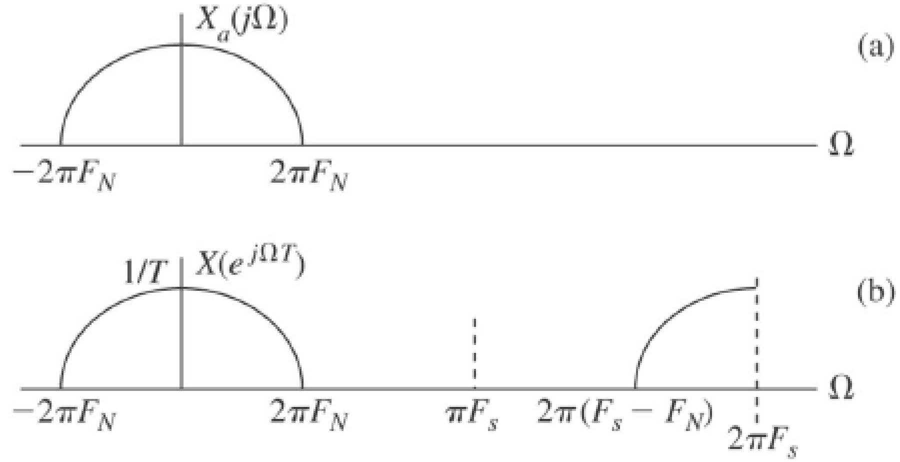
- Standard Sampling: begin with digitized signal

$$x[n] = x_a(nT) \leftrightarrow X_a(j\Omega) = 0, \quad |\Omega| \geq 2\pi F_N$$

$$F_s = \frac{1}{T} \geq 2F_N$$

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega), \quad |\Omega| < \frac{\pi}{T}$$

$$X(e^{j\Omega T}) = 0, \; 2\pi F_N \leq |\Omega| \leq 2\pi(F_s - F_N)$$



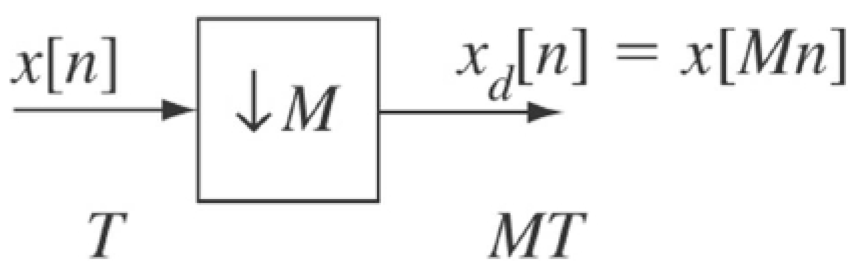
- can achieve perfect recovery of $x_a(t)$ from digitized sample under these conditions

Decimation

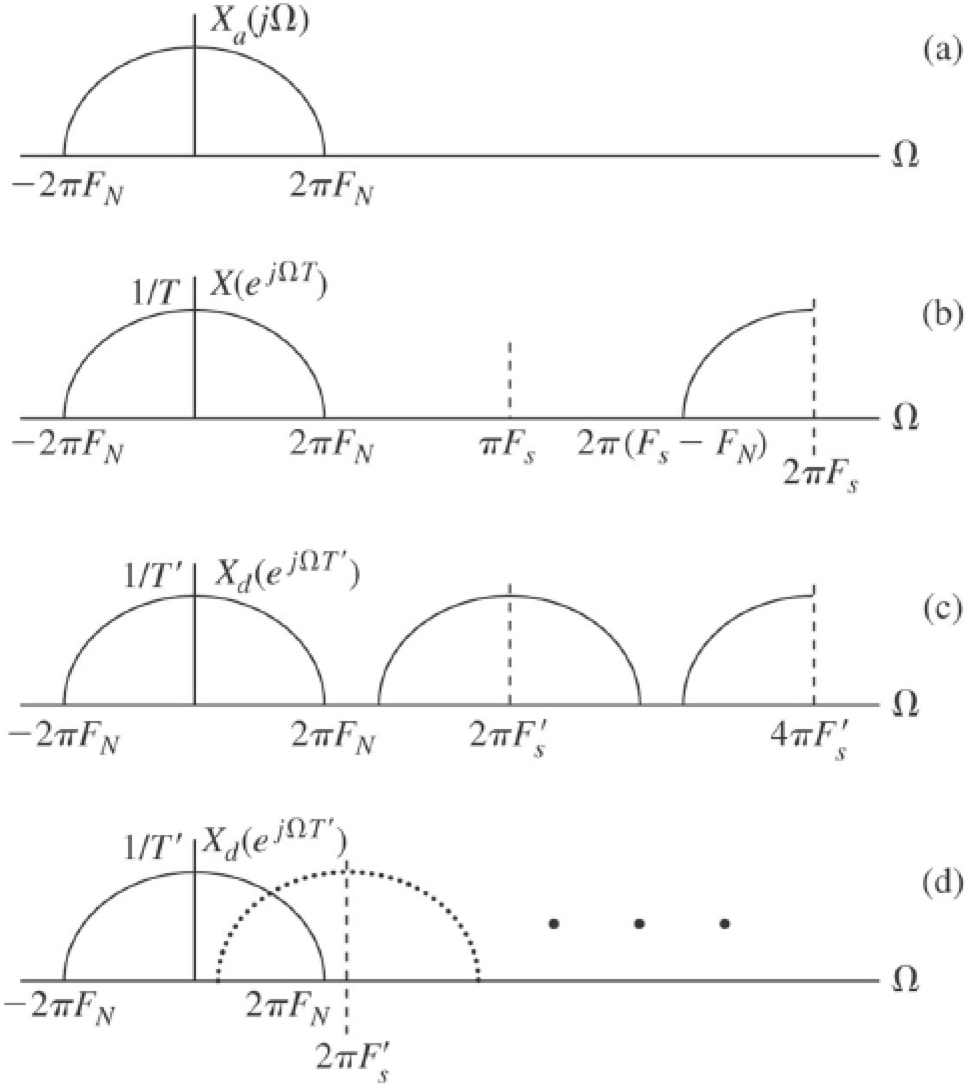
- to reduce sampling rate of sampled signal by factor of $M \geq 2$
- to compute new signal $x_d[n]$ with sampling rate

$$F_s' = 1/T' = 1/(MT) = F_s / M$$

  such that $x_d[n] = x_a(nT')$ with no aliasing
- one solution is to downsample $x[n] = x_a(nT)$ by retaining one out of every M samples of $x[n]$, giving $x_d[n] = x[nM]$



Decimation



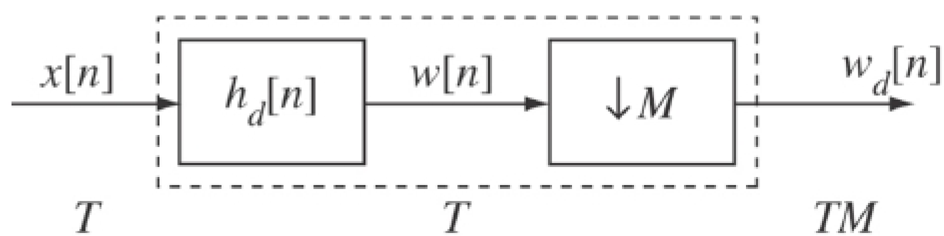
- need $F_s' \geq 2\ F_N$ to avoid aliasing for M=2
- when $F_s' < 2\ F_N$, we get aliasing for M=2

Decimation

- to decimate by factor of M with no aliasing, need to ensure that the highest frequency in $x[n]$ is no greater than $F_s/(2M)$
- thus we need to filter $x[n]$ using an ideal lowpass filter with response

$$H_d(e^{j\omega}) = \begin{cases} 1 & |\omega| < \pi / M \\ 0 & \pi / M < |\omega| \le \pi \end{cases}$$

- using the appropriate lowpass filter, we can downsample the reuslting lowpass-filtered signal by a factor of M without aliasing



Interpolation

- assume we have $x[n]=x_a(nT)$ (no aliasing) and we wish to increase the sampling rate by the integer factor of L
- we need to compute a new sequence of samples of $x_a(t)$ with period $T'' = T/L$, i.e., $x_i[n]=x_a(nT'')=x_a(nT/L)$
- It is clear that we can create the signal

  $x_i[n]=x[n/L]$ for $n = 0, \pm L, \pm 2L, \ldots$

  but we need to fill in the unknown samples by an interpolation process
- can readily show that what we want is

$$x_i[n]=x_a(nT'')=\sum_{k=-\infty}^{\infty} x_a(kT)\left[\frac{\sin(\pi(nT''-kT)/T)}{\pi(nT''-kT)/T}\right]$$

- equivalently with $T''=T/L$, $x[n]=x_a(nT)$, we get

$$x_i[n]=x_a(nT'')=\sum_{k=-\infty}^{\infty} x[k]\left[\frac{\sin(\pi(n/L-k))}{\pi(n/L-k)}\right]$$

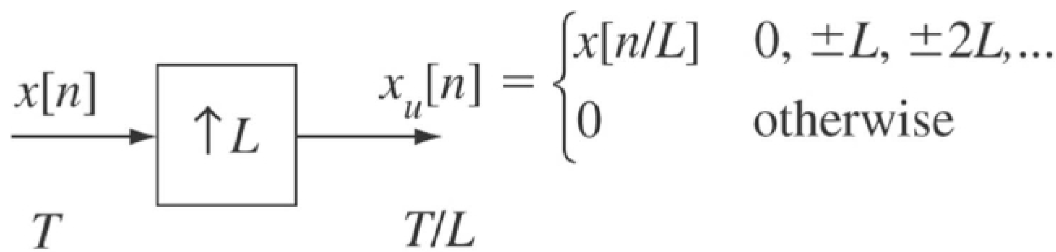
which relates $x_i[n]$ to $x[n]$ directly

Interpolation

- implementing the previous equation by filtering the upsampled sequence

$$x_u[n] = \begin{cases} x[n/L] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

- $x_u[n]$ has the correct samples for n = 0, $\pm L$, $\pm 2L$, … , but it has zero-valued samples in between (from the upsampling operation)



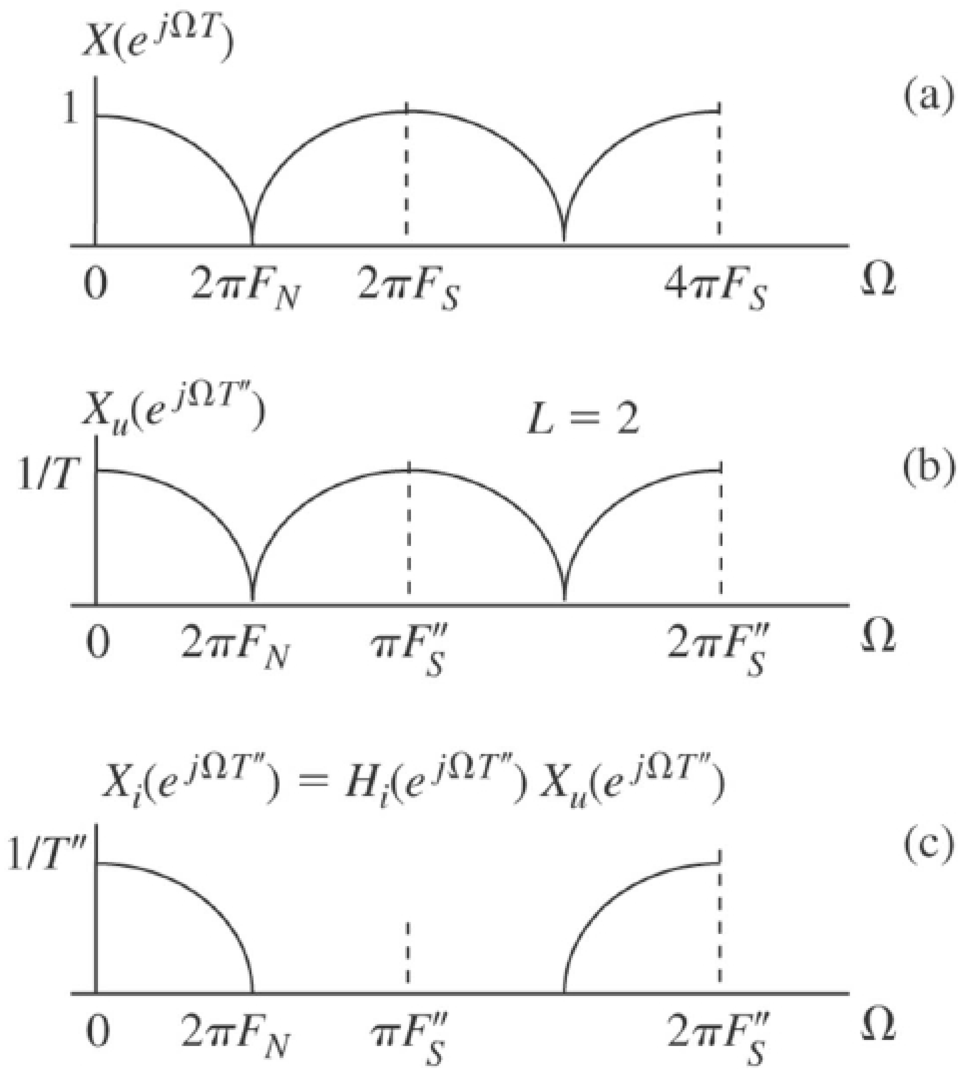
- The Fourier transform of $x_u[n]$ is simply

$$X_u(e^{j\omega}) = X(e^{j\omega L})$$

$$X_u(e^{j\Omega T''}) = X(e^{j\Omega T'' L})$$

- Thus $X_u(e^{j\Omega T''})$ is periodic with two periods, namely with period 2π/L due to upsampling and 2π due to being a digital signal

Interpolation



(a) Plot of $X(e^{j\Omega T})$

(b) Plot of $X_u(e^{j\Omega T''})$ showing double periodicity for $L = 2,\ T'' = T/2$

(c) DTFT of desired signal with

$$X_i(e^{j\Omega T''}) = \begin{cases} (2/T)X_a(j\Omega) & |\Omega| \le 2\pi F_N \\ 0 & 2\pi F_N < |\Omega| \le \pi / T'' \end{cases}$$

can obtain results of (c) by applying ideal lowpass filter with gain L (to restore amplitude) and cutoff frequency $2\pi F_N = \pi / T$, giving:

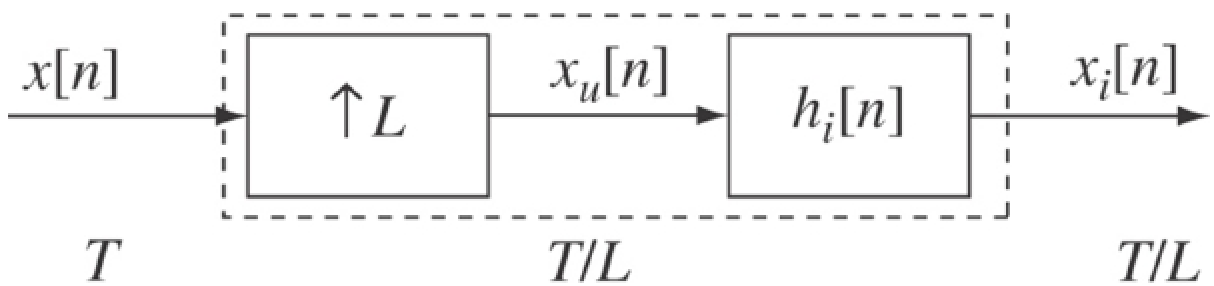
$$X_i(e^{j\omega}) = \begin{cases} (1/T'')X(e^{j\omega L}) & 0 \le |\omega| < \pi / L \\ 0 & \pi / L \le |\omega| \le \pi \end{cases}$$

$$H_i(e^{j\omega}) = \begin{cases} L & |\omega| < \pi / L \\ 0 & \pi / L \le |\omega| \le \pi \end{cases}$$

Interpolation

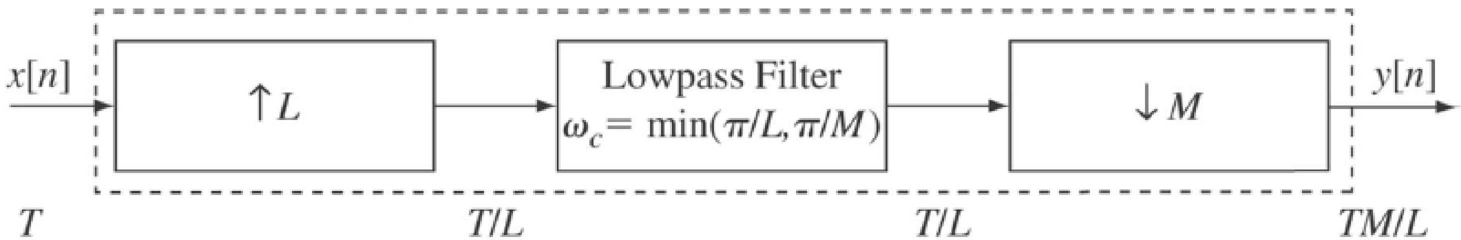
- Original signal, $x[n]$, at sampling period, T, is first upsampled to give signal $x_u[n]$ with sampling period $T''= T / L$
- lowpass filter removes images of original spectrum giving

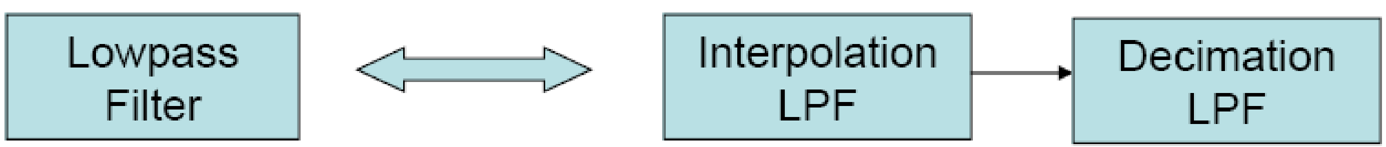
$x_i\,[n]= x_a(nT'')= x_a(nT/L)$



SR Conversion by Non-Integer Factors

- $T' = MT/L \Rightarrow$ convert rate by factor of M/L
- need to interpolate by L, then decimate by M (why can't it be done in the reverse order?)





need to combine specifications of both LPFs and implement in a single stage of lowpass filtering

  - can approximate almost any rate conversion with appropriate values of *L and M*
  - for large values of *L*, or *M*, or both, can implement in stages, i.e., L =1024, use $L1$=32 followed by $L2$=32

Summary - 1

- speech signals are inherently bandlimited => must sample appropriately in time and amplitude
- LTI systems of most interest in speech processing; can characterize them completely by impulse response, $h(n)$
- the z-transform and Fourier transform representations enable us to efficiently process signals in both the time and frequency domains
- both periodic and time-limited digital signals can be represented in terms of their Discrete Fourier transforms
- sampling in time leads to aliasing in frequency; sampling in frequency leads to aliasing in time => when processing time-limited signals, must be careful to sample in frequency at a sufficiently high rate to avoid time-aliasing

Summary - 2

- digital filtering provides a convenient way of processing signals in the time and frequency domains
- can approximate arbitrary spectral characteristics via either IIR or FIR filters, with various levels of approximation
- can realize digital filters with a variety of structures, including direct forms, serial and parallel forms
- once a digital signal has been obtain via appropriate sampling methods, its sampling rate can be changed digitally (either up or down) via appropriate filtering and decimation or interpolation