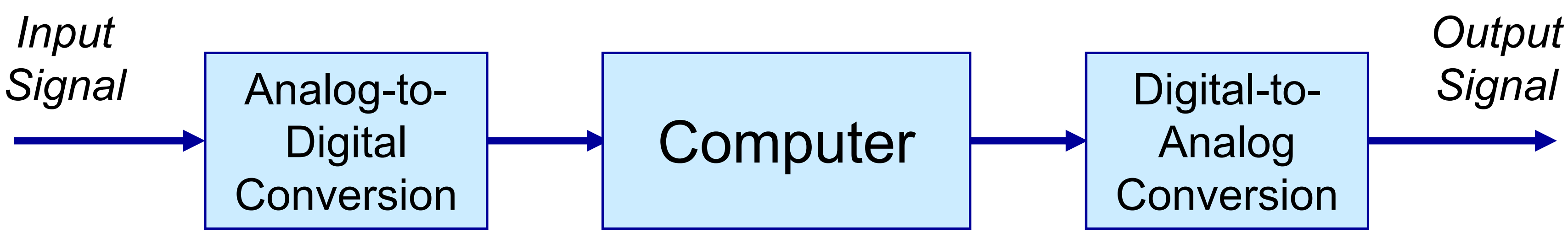
Digital Speech Processing—Lecture 2

Review of DSP Fundamentals

What is DSP?



Digital
- Method to represent a quantity, a phenomenon or an event
- Why digital?

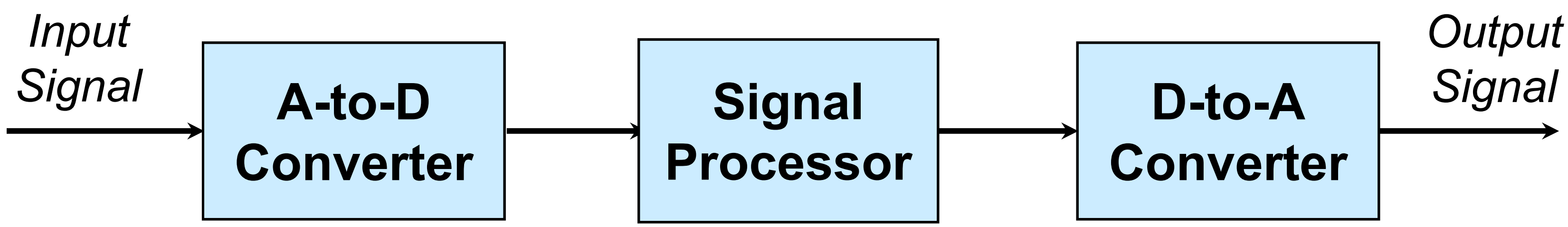
Signal
- What is a signal?
  - something (e.g., a sound, gesture, or object) that carries information
  - a detectable physical quantity (e.g., a voltage, current, or magnetic field strength) by which messages or information can be transmitted
- What are we interested in, particularly when the signal is speech?

Processing
- What kind of processing do we need and encounter almost everyday?
- Special effects?

Common Forms of Computing

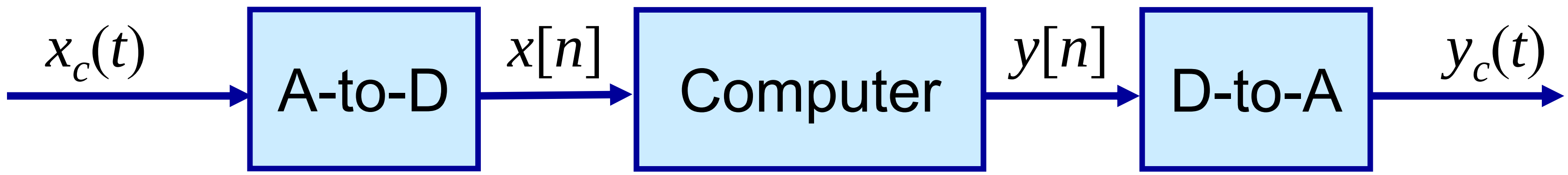
- **Text processing** – handling of text, tables, basic arithmetic and logic operations (i.e., calculator functions)
  - Word processing
  - Language processing
  - Spreadsheet processing
  - Presentation processing
- **Signal Processing** – a more general form of information processing, including handling of speech, audio, image, video, etc.
  - Filtering/spectral analysis
  - Analysis, recognition, synthesis and coding of real world signals
  - Detection and estimation of signals in the presence of noise or interference

Advantages of Digital Representations



- Permanence and robustness of signal representations; zero-distortion reproduction may be achievable
- Advanced IC technology works well for digital systems
- Virtually infinite flexibility with digital systems
  - Multi-functionality
  - Multi-input/multi-output
- Indispensable in telecommunications which is virtually all digital at the present time

Digital Processing of Analog Signals



- **A-to-D conversion:** bandwidth control, sampling and quantization
- **Computational processing:** implemented on computers or ASICs 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:

$$x = \{x[n]\}, \qquad -\infty < n < \infty$$

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

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

- T is called the **sampling period,** and its reciprocal, $F_S = 1/T$, is called the **sampling frequency**

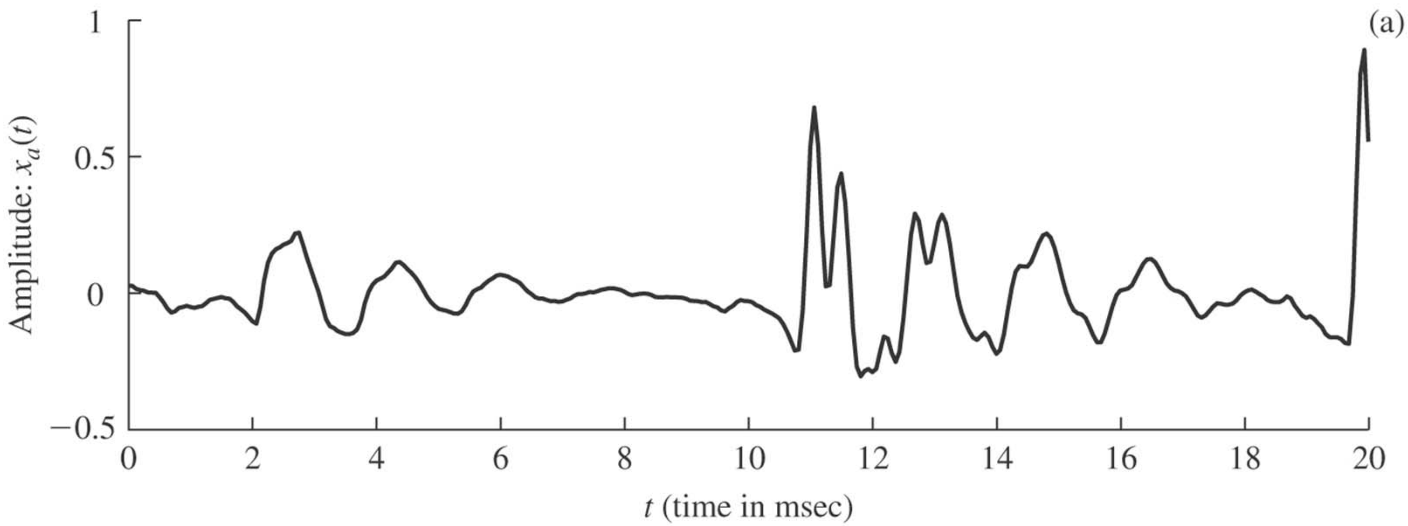
$$F_S = 8000 \text{ Hz} \quad \leftrightarrow \quad T = 1/8000 = 125\,\mu\text{sec}$$

$$F_S = 10000 \text{ Hz} \quad \leftrightarrow \quad T = 1/10000 = 100\,\mu\text{sec}$$

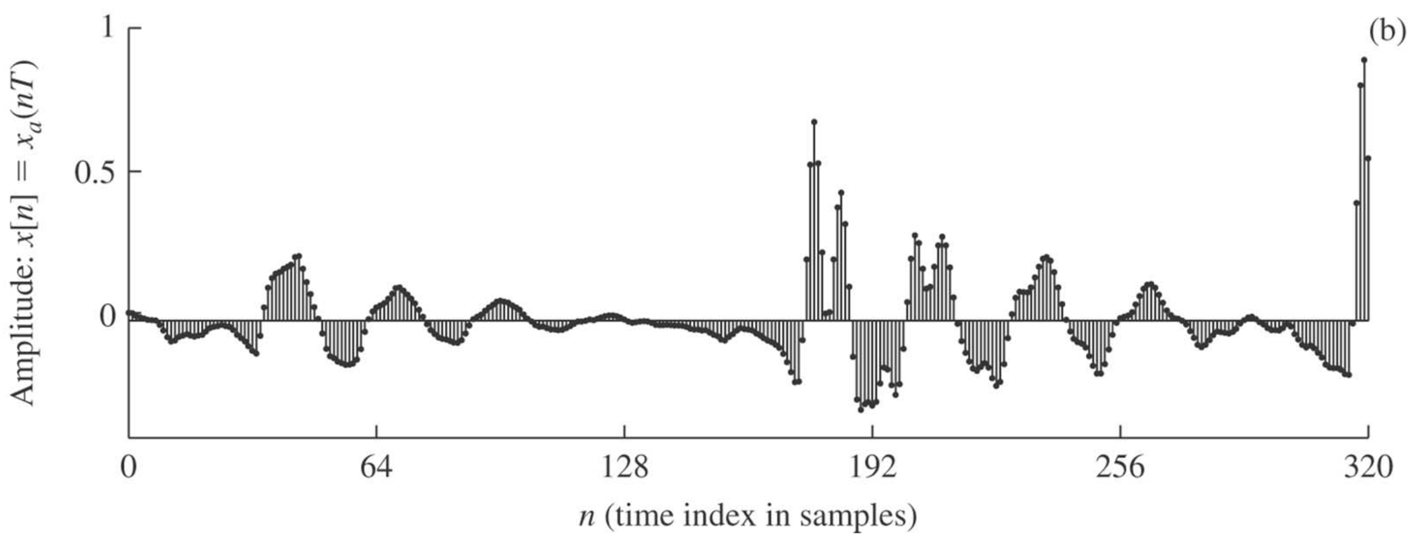
$$F_S = 16000 \text{ Hz} \quad \leftrightarrow \quad T = 1/16000 = 62.5\,\mu\text{sec}$$

$$F_S = 20000 \text{ Hz} \quad \leftrightarrow \quad T = 1/20000 = 50\,\mu\text{sec}$$

Speech Waveform Display

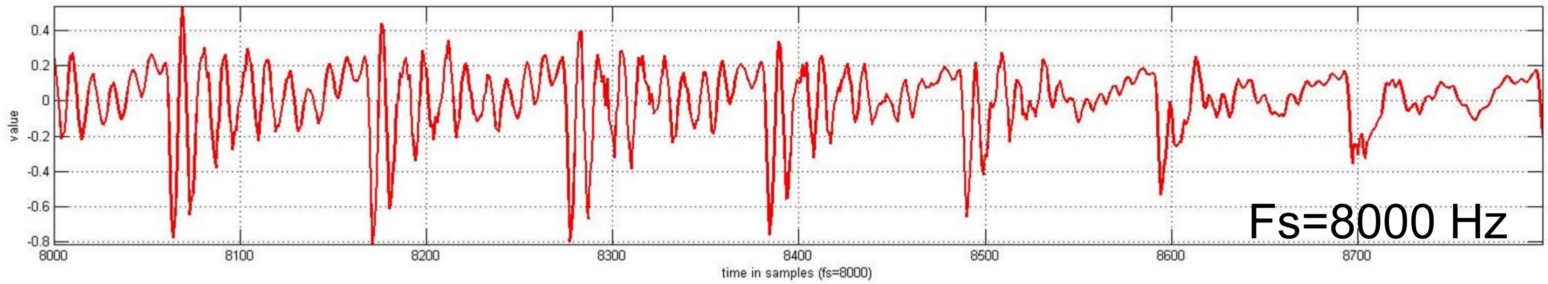


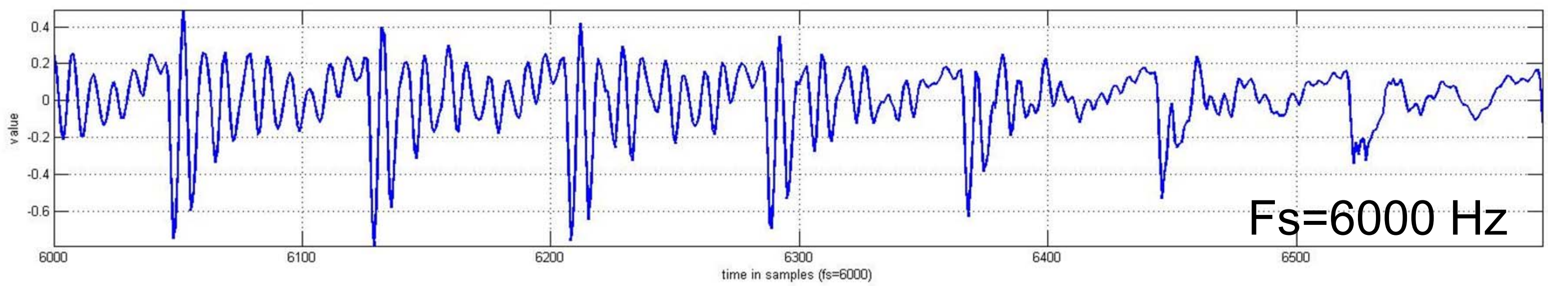
plot( );

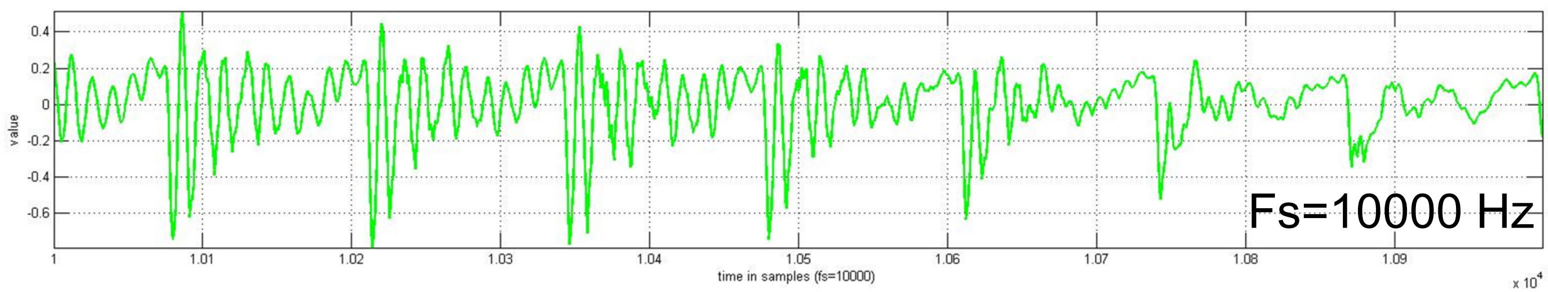


stem( );

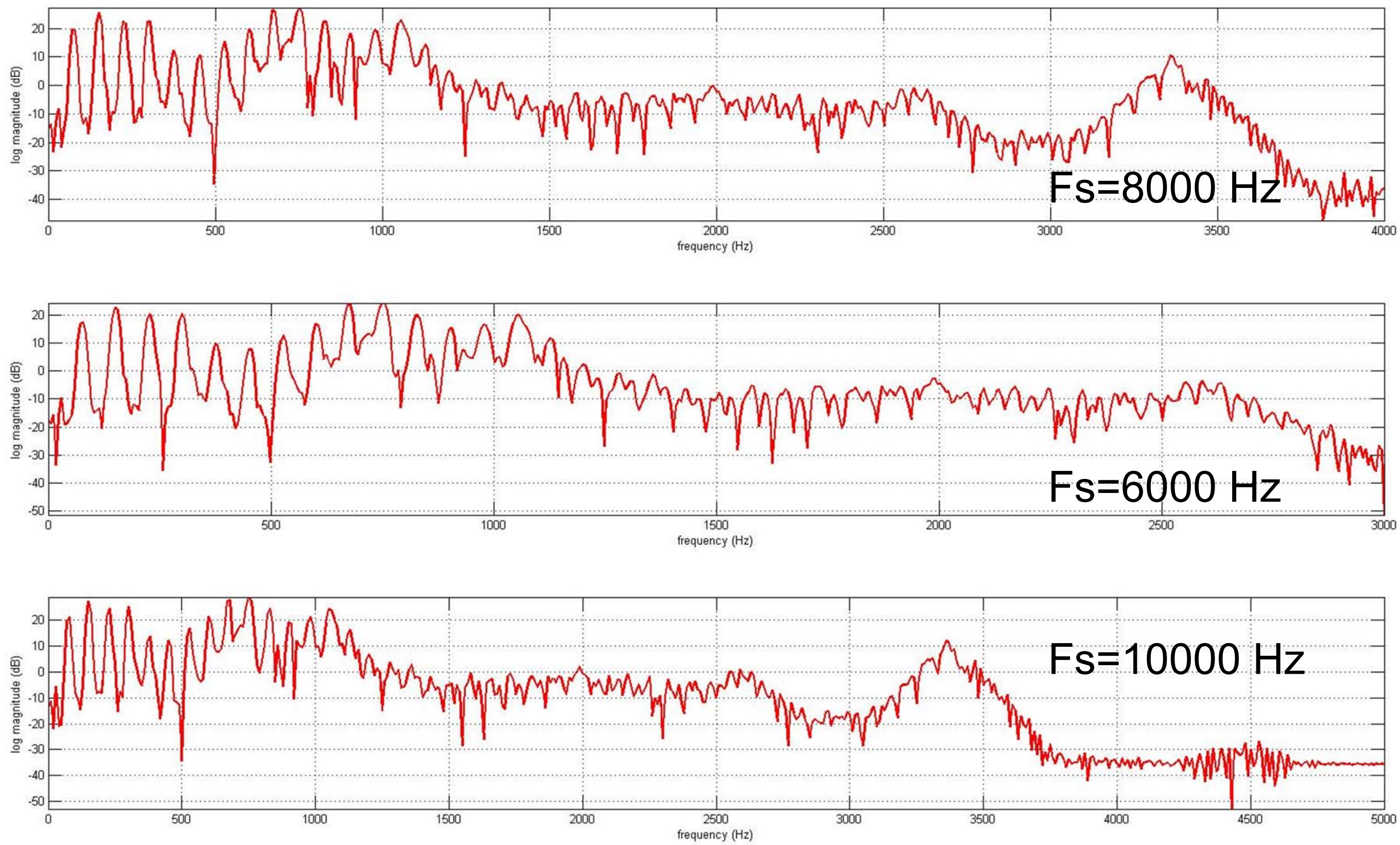
Varying Sampling Rates





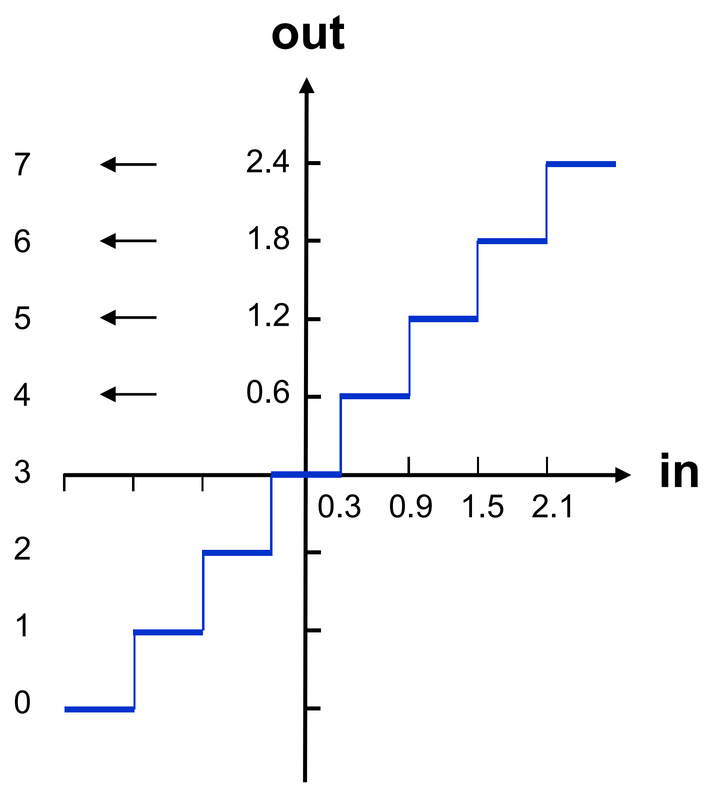


Varying Sampling Rates



Quantization

 $x[n]$ can be quantized to one of a finite set of values which is then represented digitally in bits, hence a truly digital signal; the course material mostly deals with discrete-time signals (discrete-value only when noted).

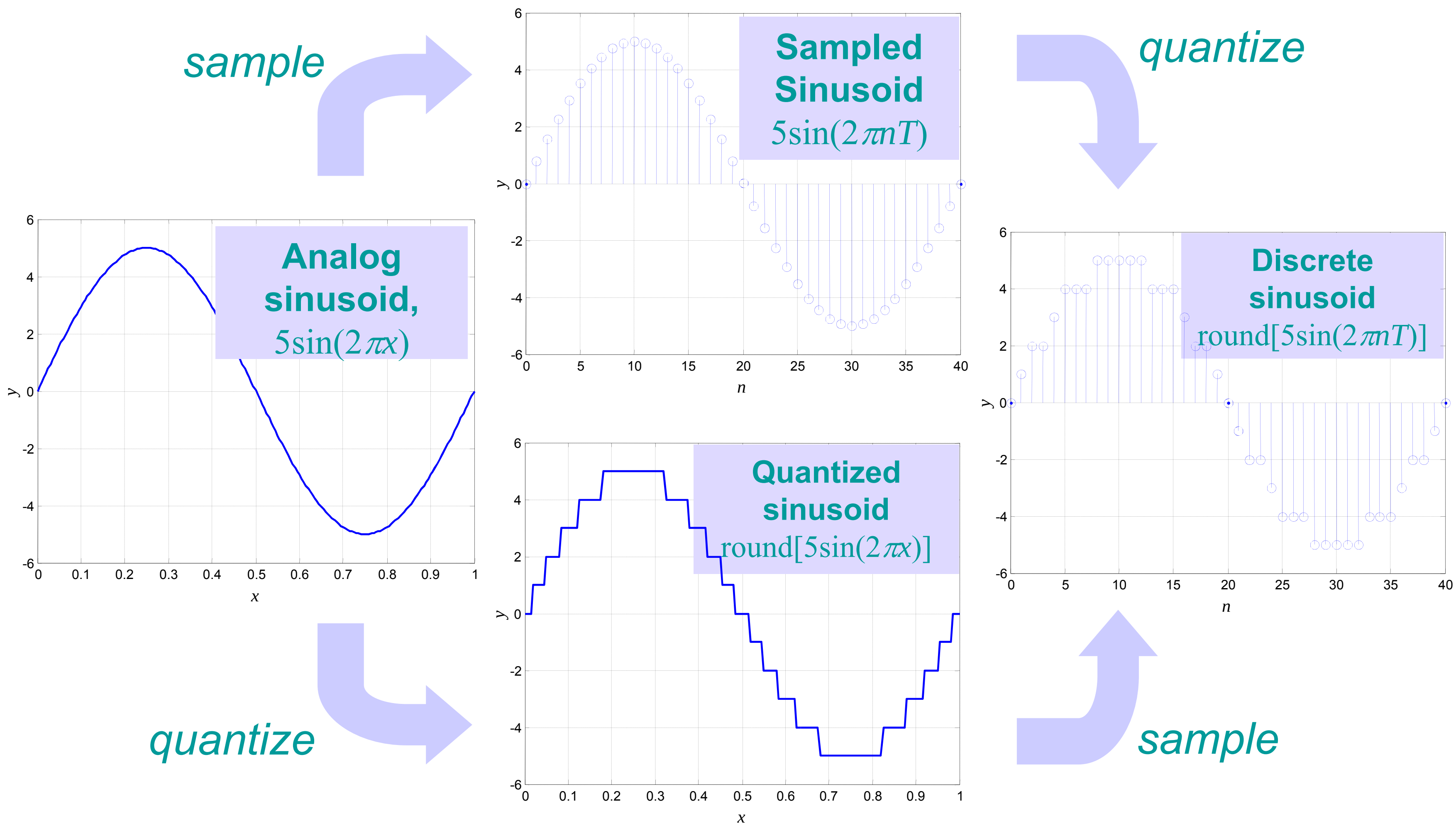


A 3-bit uniform quantizer

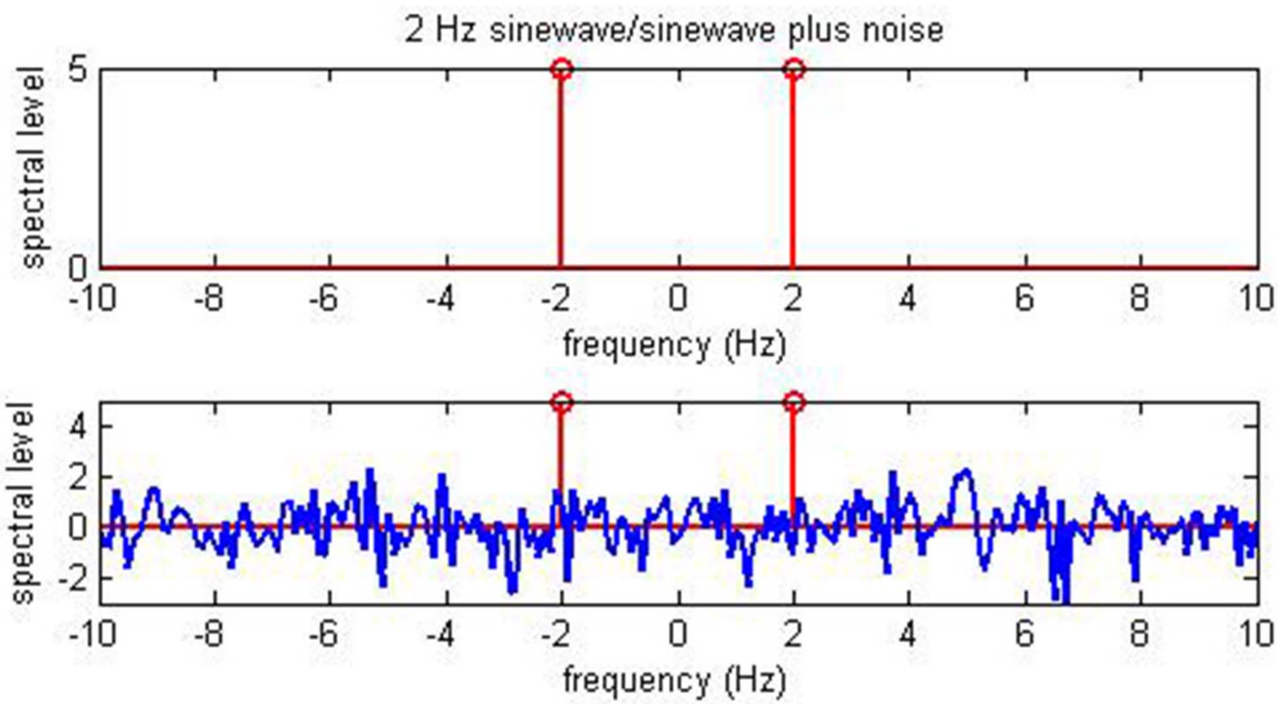
Quantization:

- 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 $(110)_2$ in binary representation
- Storage or transmission uses binary representation; a quantization table is needed

Discrete Signals



Sinewave Spectrum

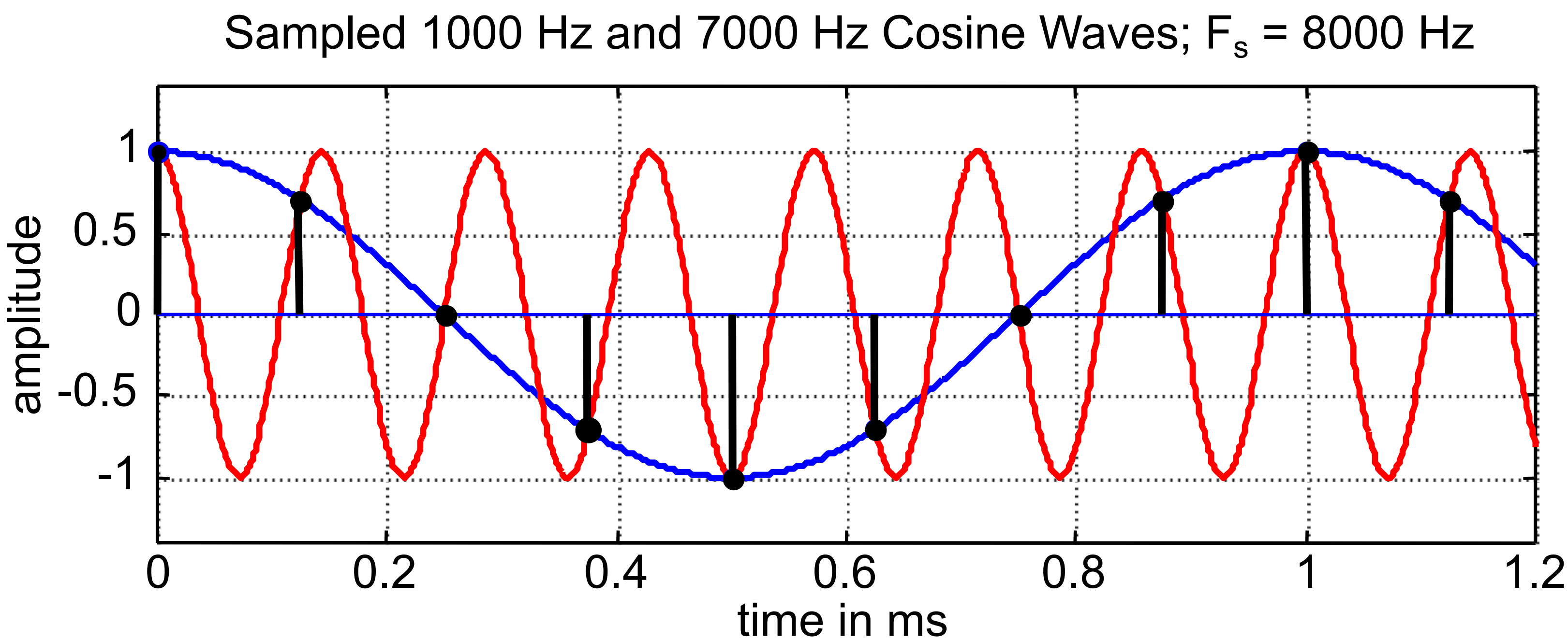


SNR is a function of *B*, the number of bits in the quantizer

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

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

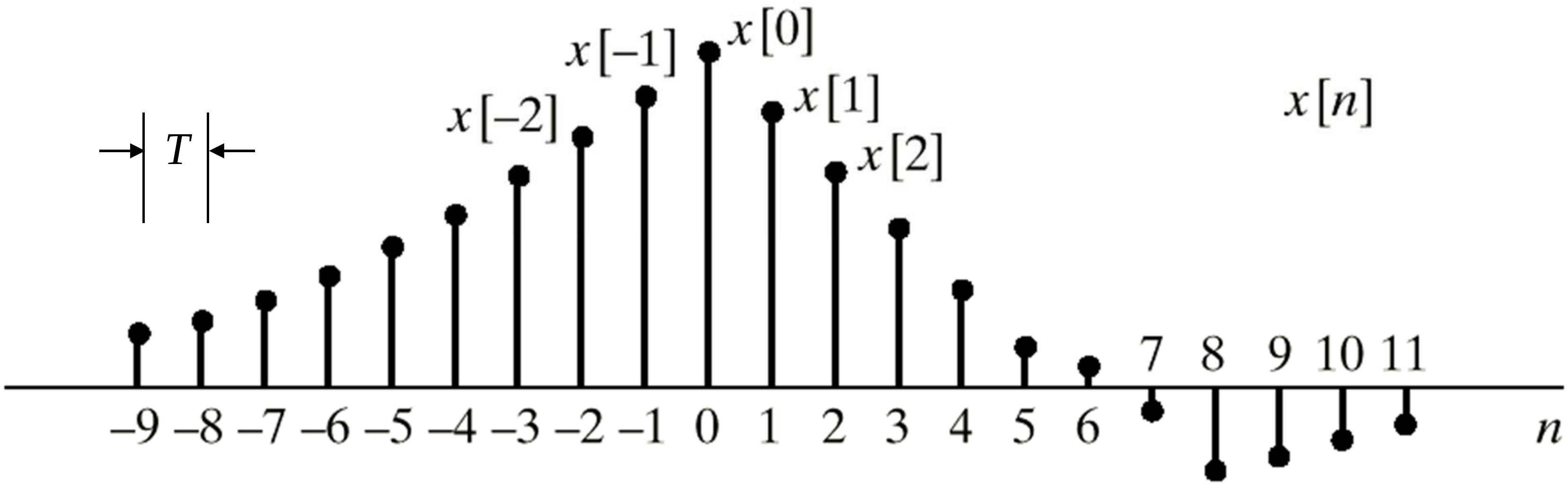
Demo Examples

1. **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)
2. **quantization to various levels** — 12,9,4,2, and 1 bit quantization (notice the distortion introduced when the number of bits is too low)
3. **music quantization** — 16 bit audio quantized to various levels:

Maple Rag: 16 bits, 12 bits, 10 bits, 8 bits, 6 bits, 4 bits;

Noise: 10 12 bits

Discrete-Time (DT) Signals are Sequences



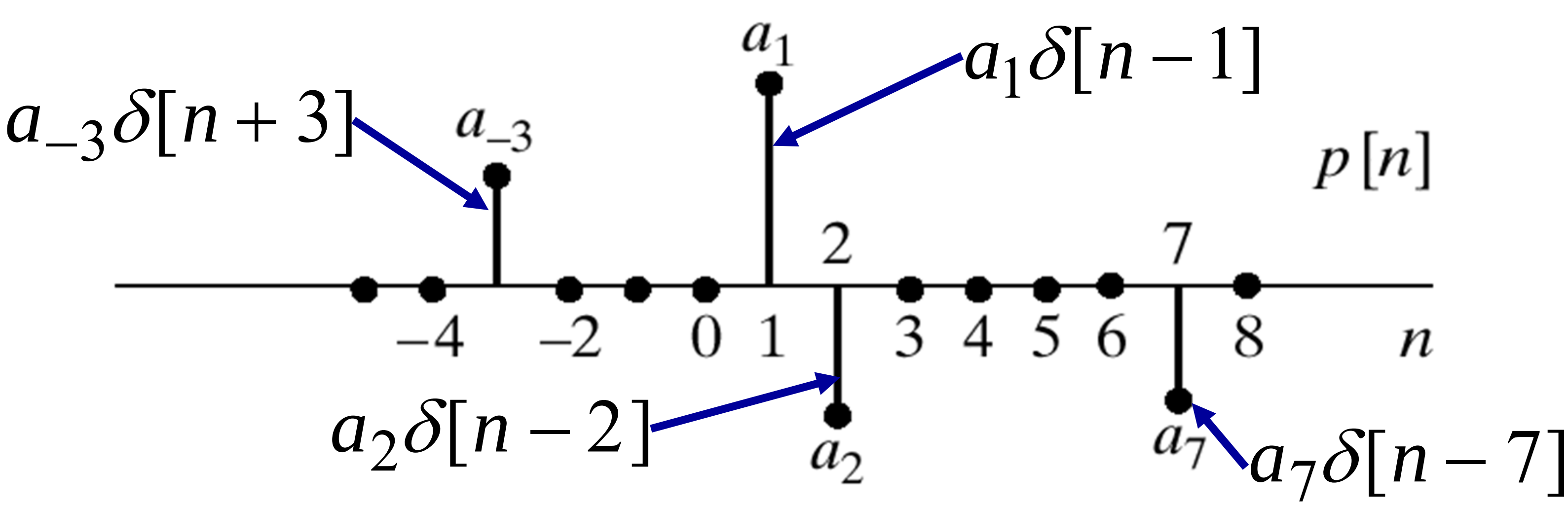
- $x[n]$ denotes the "sequence value at 'time' n"
- Sources of sequences:
  - Sampling a continuous-time signal
    $x[n] = x_c(nT) = x_c(t)|_{t=nT}$
  - Mathematical formulas – generative system
    e.g., $x[n] = 0.3 \bullet x[n\text{-}1] \text{ -}1;\quad x[0] = 40$

Impulse Representation of Sequences

A sequence,
a function

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

Value of the
function at k

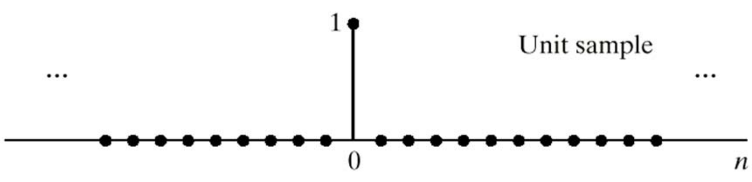


$$x[n] = a_{-3}\delta[n+3] + a_1\delta[n-1] + a_2\delta[n-2] + a_7\delta[n-7]$$

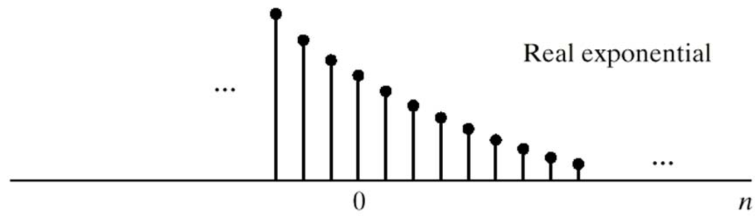
Some Useful Sequences

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

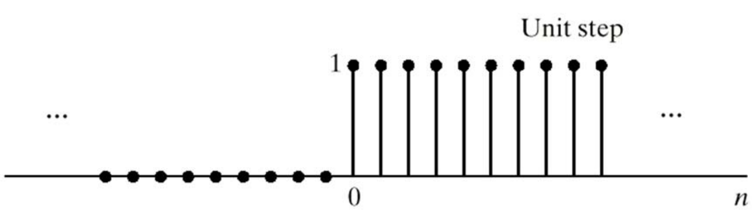
real exponential $$x[n] = \alpha^n$$



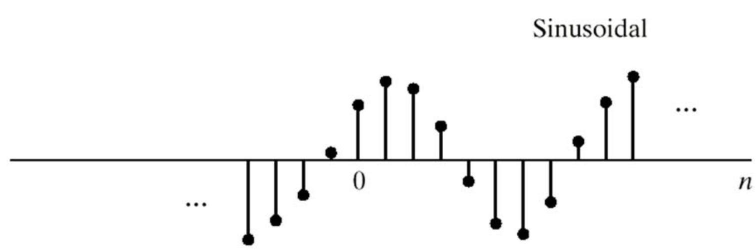
(a)



(c)



(b)

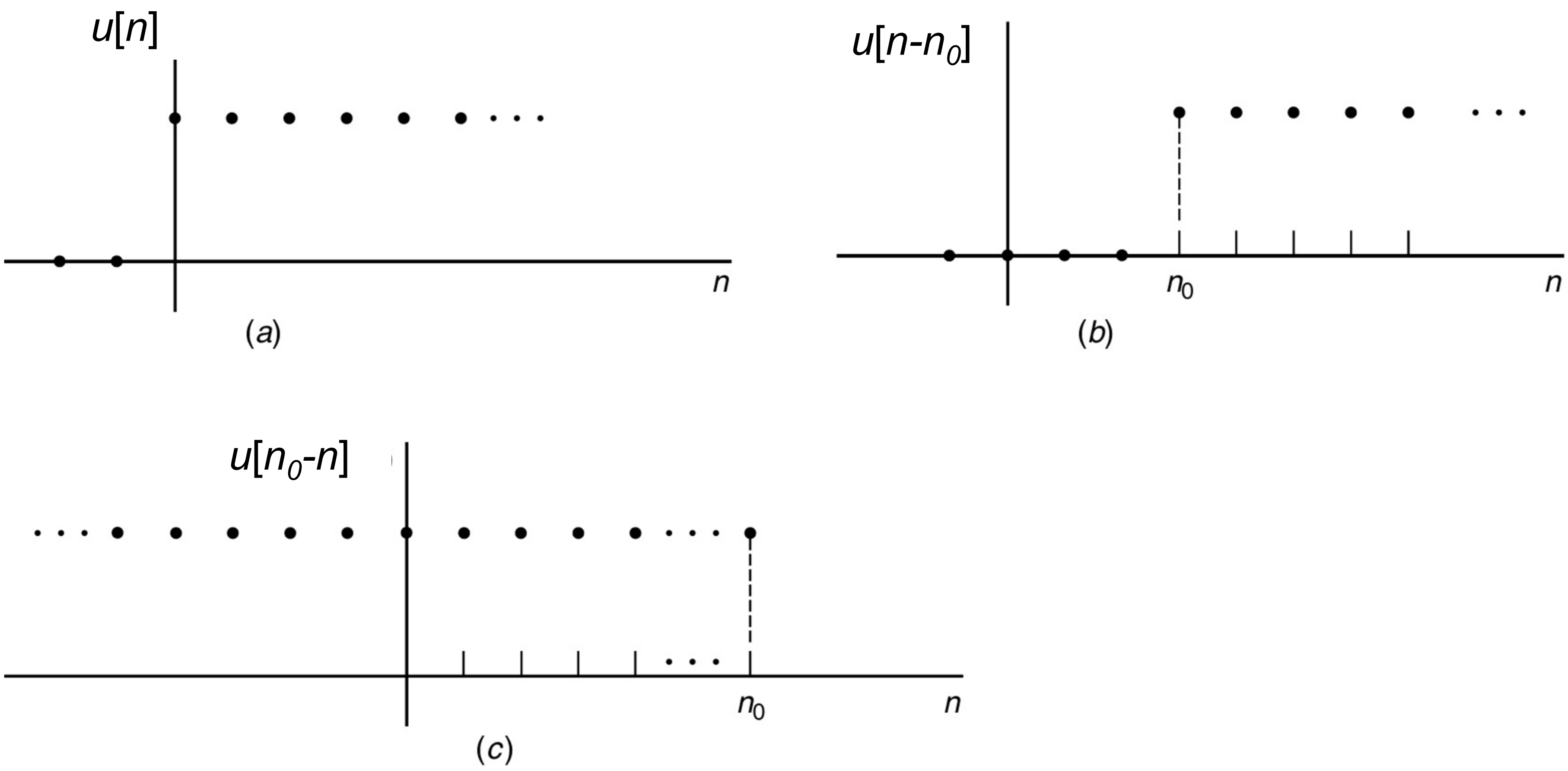


(d)

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

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

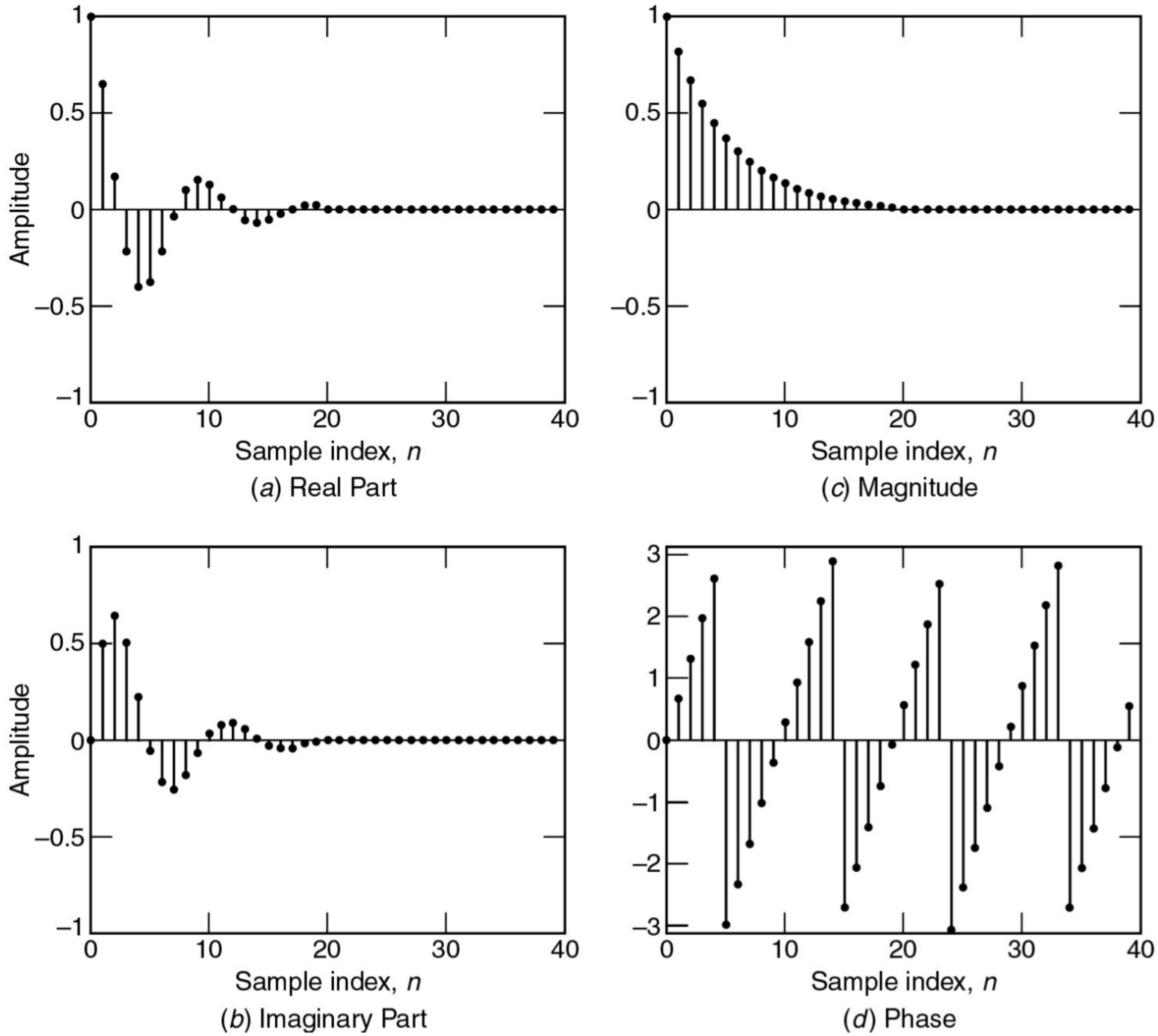
Variants on Discrete-Time Step Function



$$n \rightarrow -n \;\; \Leftrightarrow \;\; \text{signal flips around 0}$$

Complex Signal

$$x[n] = (0.65 + 0.5j)^n \; u[n]$$

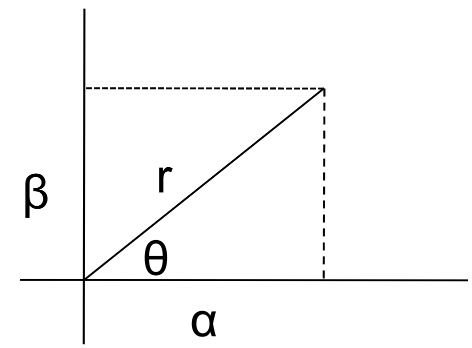


Complex Signal

$$x[n] = (\alpha + j\beta)^n\, u[n] = (re^{j\theta})^n\, u[n]$$

$$r = \sqrt{\alpha^2 + \beta^2}$$

$$\theta = \tan^{-1}(\beta / \alpha)$$



$$x[n] = r^n\, e^{j\theta n}\, u[n]$$

r^n is a dying exponential

$e^{j\theta n}$ is a linear phase term

Complex DT Sinusoid

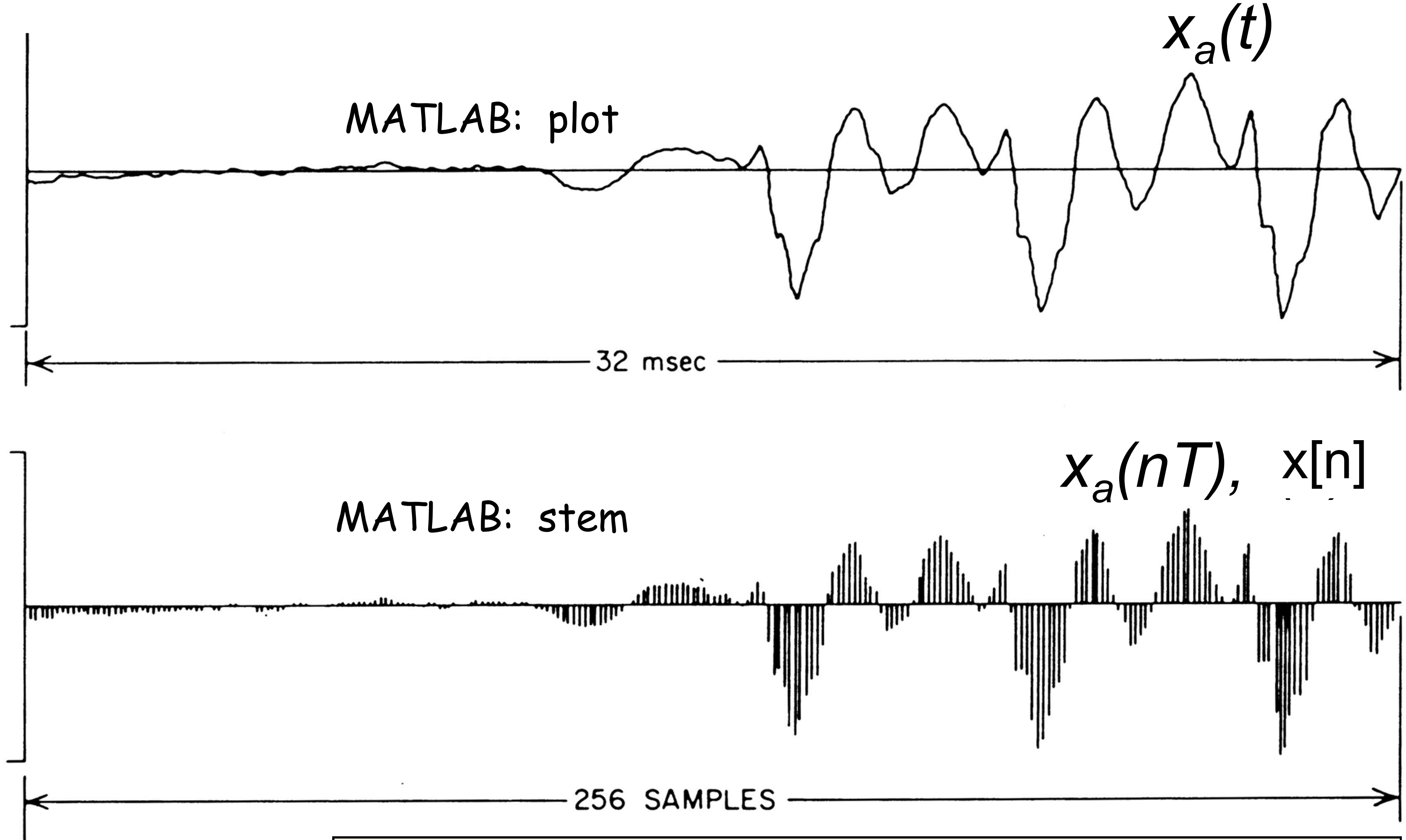
$$x[n] = Ae^{j\omega n}$$

- Frequency ω is in radians (per sample), or just radians
  - once sampled, $x[n]$ is a sequence that **relates to time only through the sampling period** T
- Important property: periodic in ω with period 2π:

$$Ae^{j\omega_0 n} = Ae^{j(\omega_0 + 2\pi r)n}$$

  - Only unique frequencies are 0 to 2π (or $-\pi$ to $+\pi$)
  - Same applies to real sinusoids

Sampled Speech Waveform

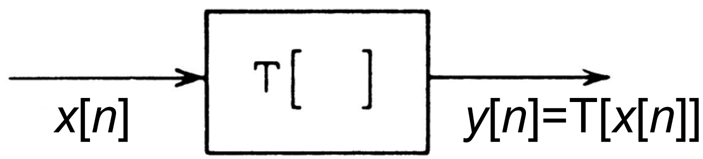


T=0.125 msec, f_S=8 kHz

Trap #1: loss of sampling time index

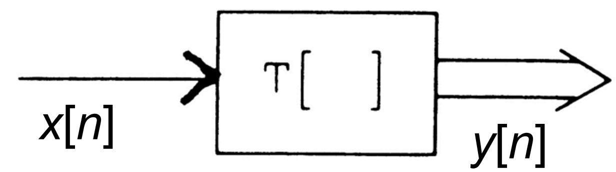
Signal Processing

- Transform digital signal into more desirable form



(a)

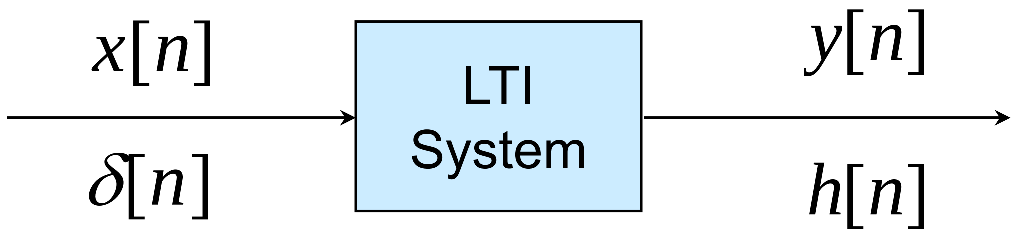
single input—single output



(b)

single input—multiple output, e.g., filter bank analysis, sinusoidal sum 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):

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

- 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

Example:

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

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

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

$$x_1[n] + x_2[n] \leftrightarrow y_3[n] = x_1[n] + x_2[n] + 2x_1[n+1] + 2x_2[n+1] + 3$$

$\neq y_1[n] + y_2[n] \Rightarrow$ 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?

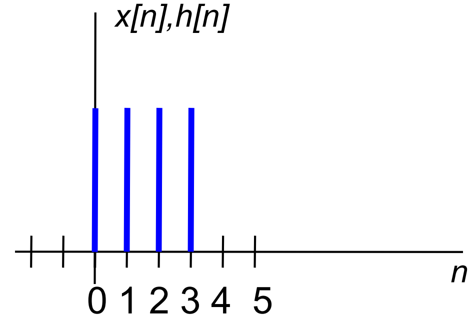
$y[n]$ depends on $x[n+1]$, a sample in the future

$\Rightarrow$ System is not causal!

Convolution Example

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

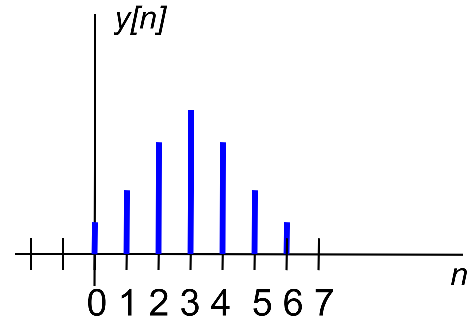
What is $y[n]$ for this system?

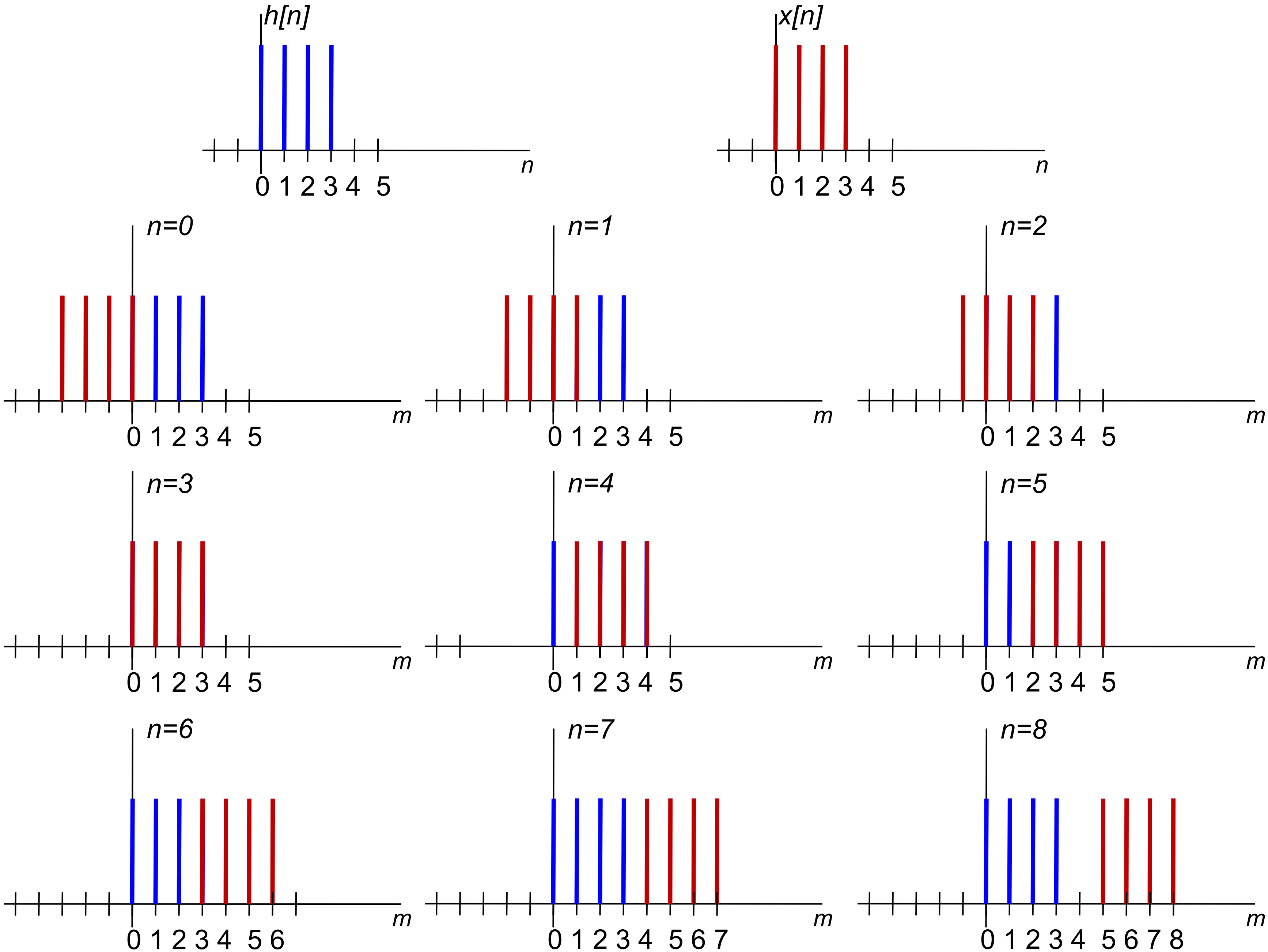


Solution :

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

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




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

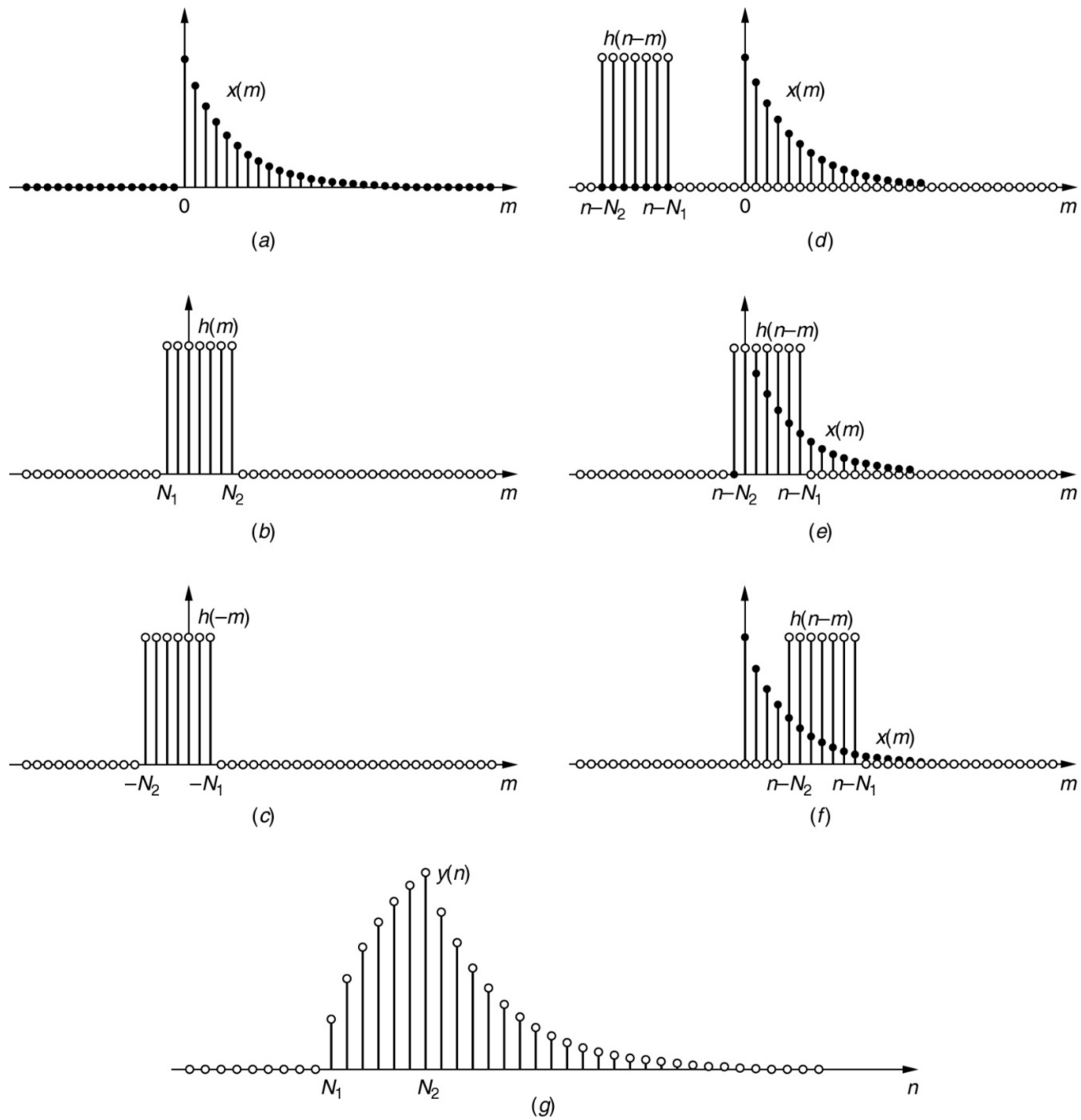
SOLUTION:

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

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

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

Convolution Example



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**

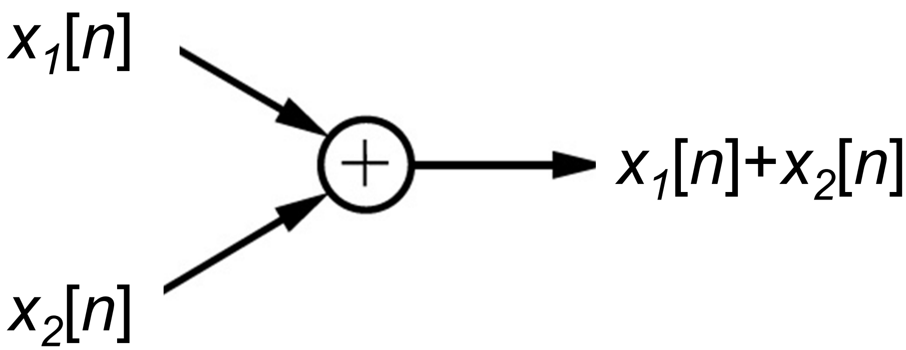
$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

$$y[n] = h[n] * x[n], \quad \text{where } * \text{ denotes discrete convolution}$$

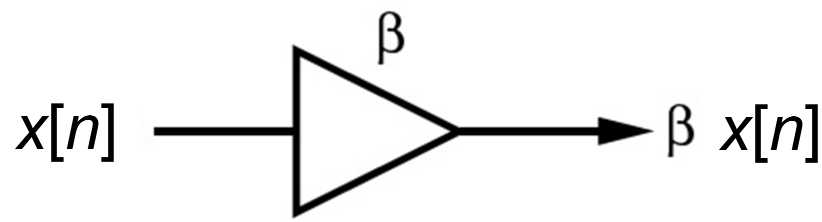
$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]=x[n] * h[n]$$

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

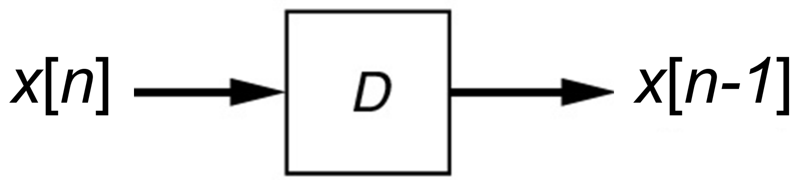
Signal Processing Operations



(a)



(b)

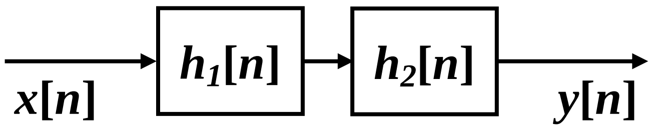


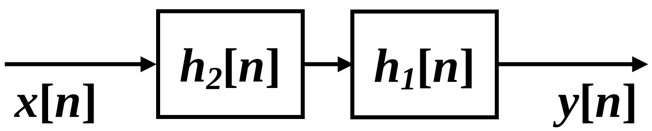
(c)

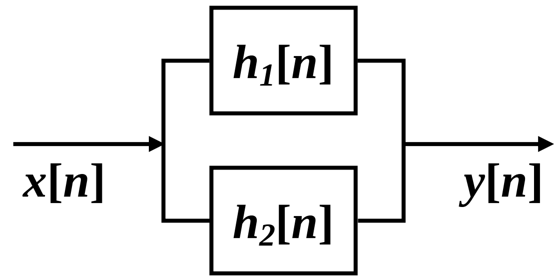
D is a delay of 1-sample

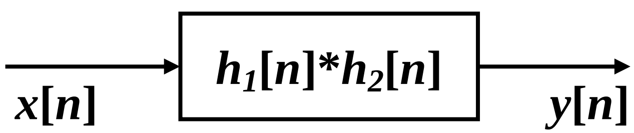
Can replace D with delay element z^{-1}

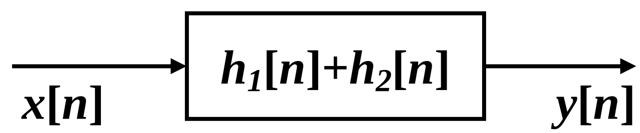
Equivalent LTI Systems







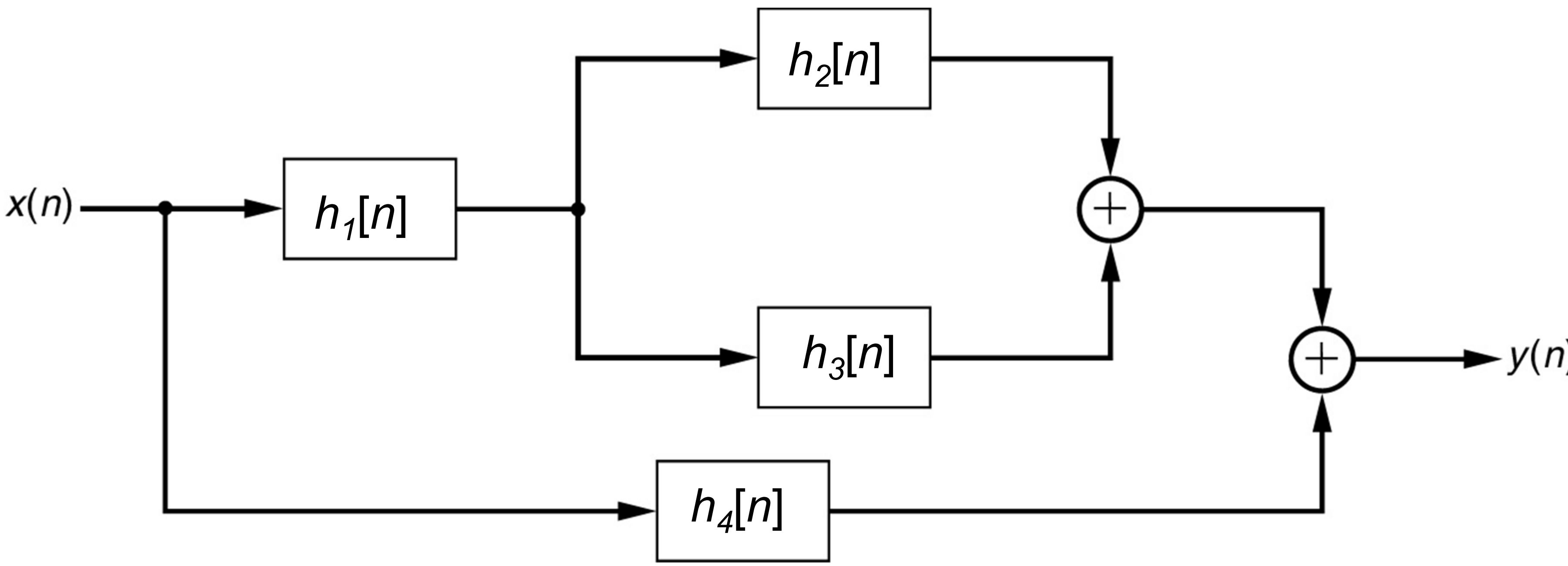




$$h_1[n]*h_2[n]= h_2[n]*h_1[n]$$

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

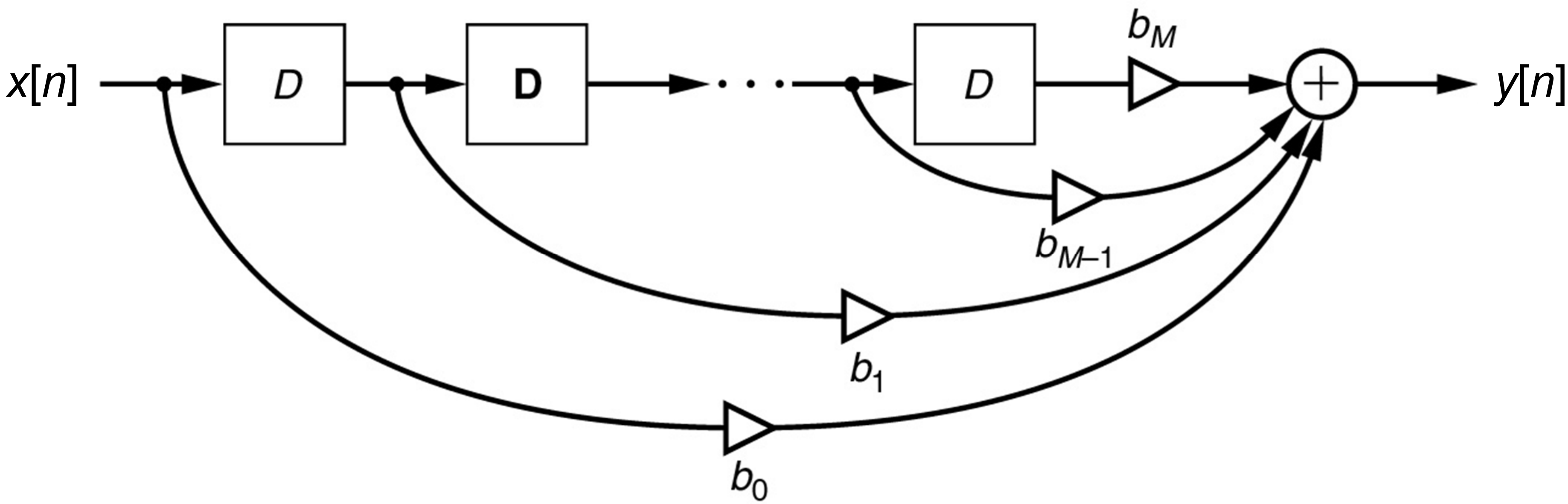
More Complex Filter Interconnections



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

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

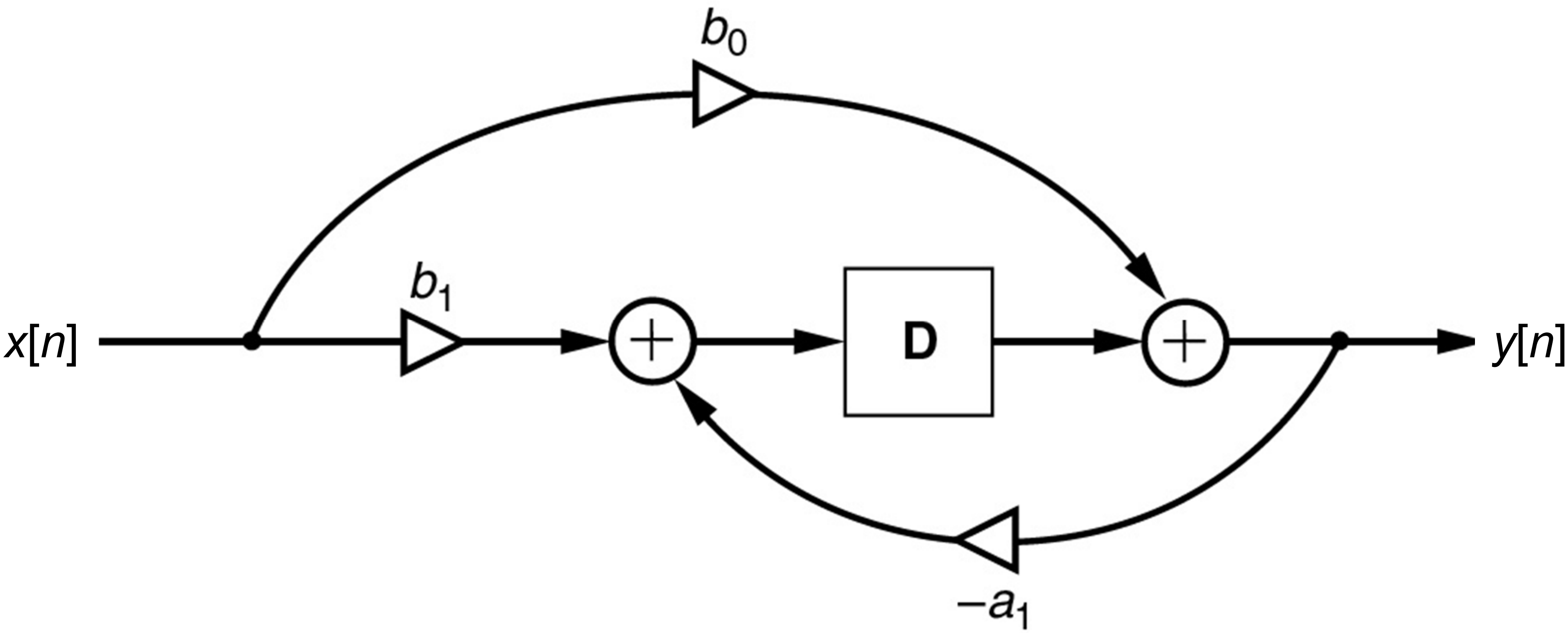
Network View of Filtering (FIR Filter)



$$D\,(\text{Delay Element}) \Leftrightarrow z^{-1}$$

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

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:

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

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

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

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

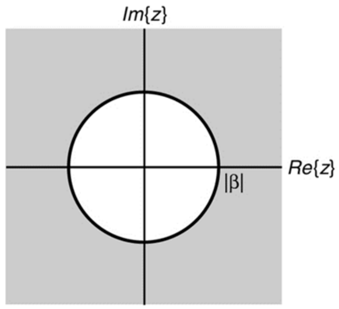
- direct evaluation using residue theorem
- partial fraction expansion of *X(z)*
- long division
- power series expansion

- ***X(z)* converges (is finite) only for certain values of *z*:**

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

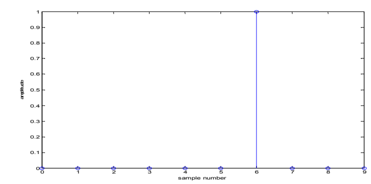
- sufficient condition for convergence

- **region of convergence: $R_1 < |z| < R_2$**



Examples of Convergence Regions

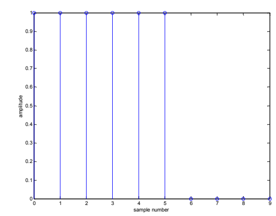
1. $x[n]=\delta[n-n_0]$ -- delayed impulse



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

$|z|<\infty,\ n_0<0;\ \forall z,\ n_0=0$

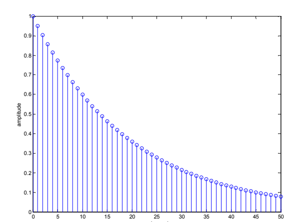
2. $x[n]=u[n]-u[n-N]$ -- box pulse



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

- all finite length sequences converge in the region $0<|z|<\infty$

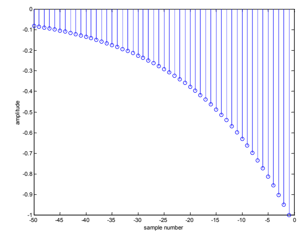
3. $x[n]=a^n u[n]\quad (a<1)$



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

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

Examples of Convergence Regions



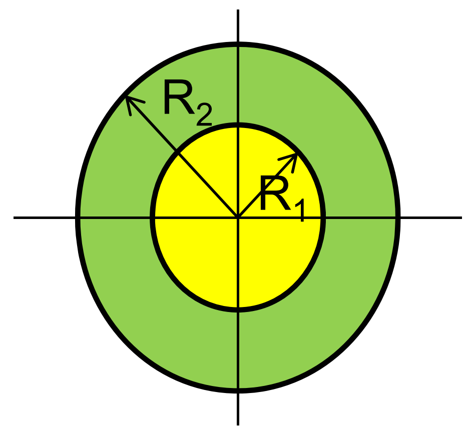
4. $x[n] = -b^n u[-n-1]$

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

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

5. $x[n]$ non-zero for $-\infty < n < \infty$ 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 $n \geq 0$ => $|z| > R_1$
- sub-sequence for $n < 0$ => $|z| < R_2$
- total sequence => $R_1 < |z| < R_2$



Example

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:

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

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

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

ROC: $|a| r_i < |z| < |a| r_o$

z-Transform Property

The sequence $x[n]$ has z-transform $X(z)$.

Show that the sequence $nx[n]$ has z-transform

$$-z\frac{dX(z)}{dz}.$$

Solution:

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

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

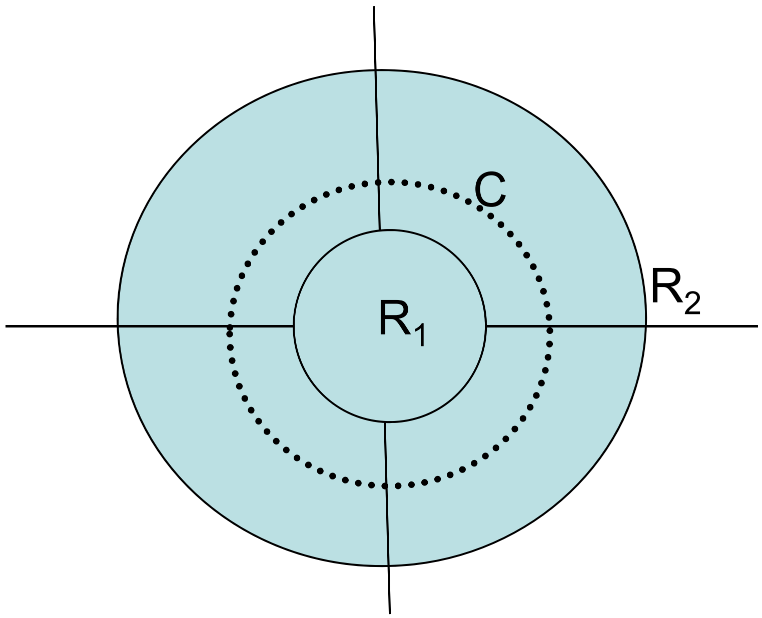
$$= -\frac{1}{z}\sum_{n=-\infty}^{\infty} nx[n]z^{-n}$$

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

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, can use a partial fraction expansion for finding inverse transforms

Partial Fraction Expansion

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

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

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

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

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

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	$n\,x[n]$	$-z\,dX(z)/dz$
Time Reversal	$x[-n]$ non-causal, need $x[N_0-n]$ to be causal for finite length sequence	$X(z^{-1})$
Convolution	$x[n] * h[n]$	$X(z)\,H(z)$
Multiplication of Sequences	$x[n]\,w[n]$	$\frac{1}{2\pi j}\oint_C X(v)W(z/v)v^{-1}dv$ circular convolution in the frequency domain

Discrete-Time Fourier Transform (DTFT)

Discrete-Time Fourier Transform

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

$$z = e^{j\omega};\ |z| = 1,\ \arg(z) = j\omega$$

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

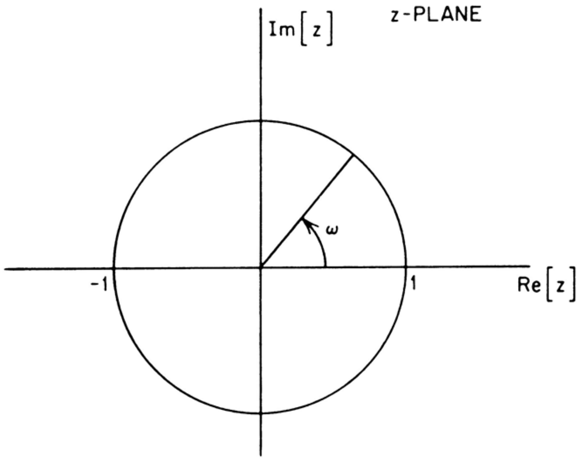


Fig. 2.4 The unit circle of the *z*-plane.

- **evaluation of *X(z)* on the unit circle in the *z*-plane**
- **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 $$x[n] = \delta[n], \qquad X(e^{j\omega}) = 1$$

Delayed impulse $$x[n] = \delta[n - n_0], \quad X(e^{j\omega}) = e^{-j\omega n_0}$$

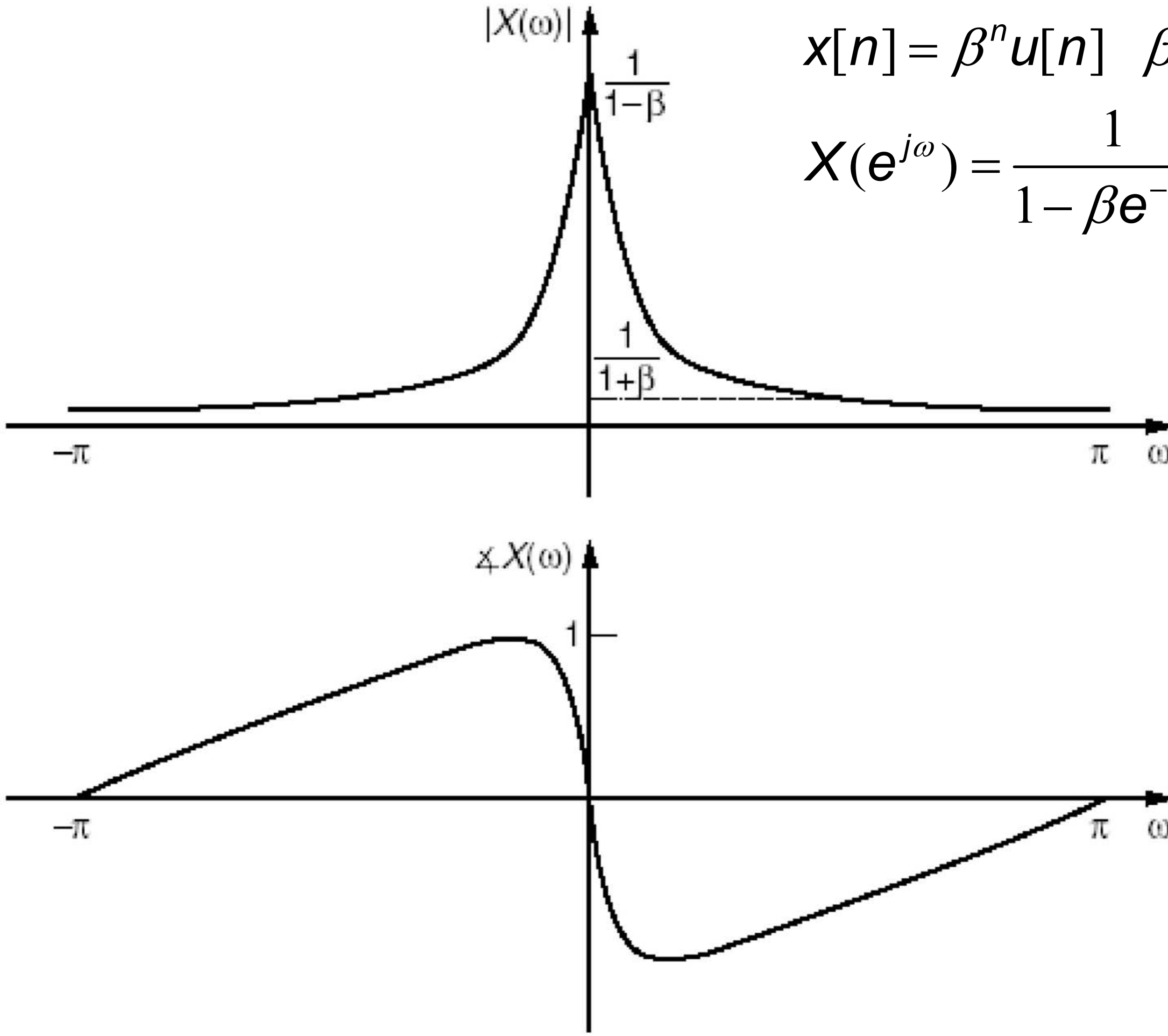
Step function $$x[n] = u[n], \qquad X(e^{j\omega}) = \frac{1}{1 - e^{-j\omega}}$$

Rectangular window $$x[n] = u[n] - u[n - N],\ X(e^{j\omega}) = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$$

Exponential $$x[n] = a^n\ u[n], \qquad X(e^{j\omega}) = \frac{1}{1 - ae^{-j\omega}},\ a < 1$$

Backward exponential $$x[n] = -b^n\ u[-n - 1],\ X(e^{j\omega}) = \frac{1}{1 - be^{-j\omega}},\ b > 1$$

DTFT Examples



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

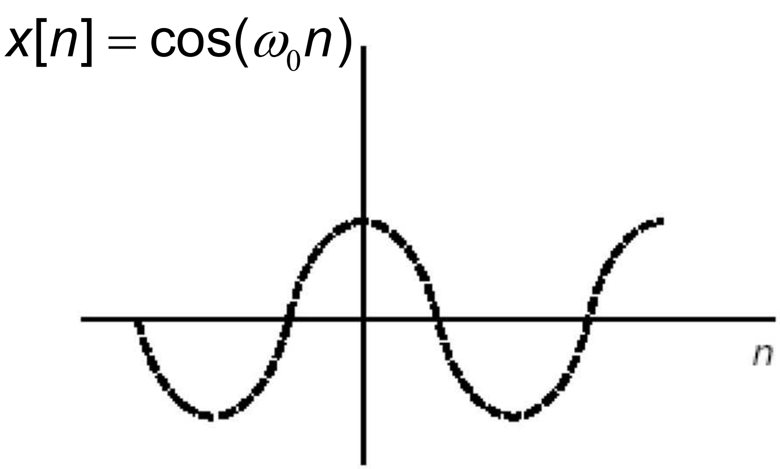
$$X(e^{j\omega}) = \frac{1}{1 - \beta e^{-j\omega}} \quad |\beta| < 1$$

DTFT Examples

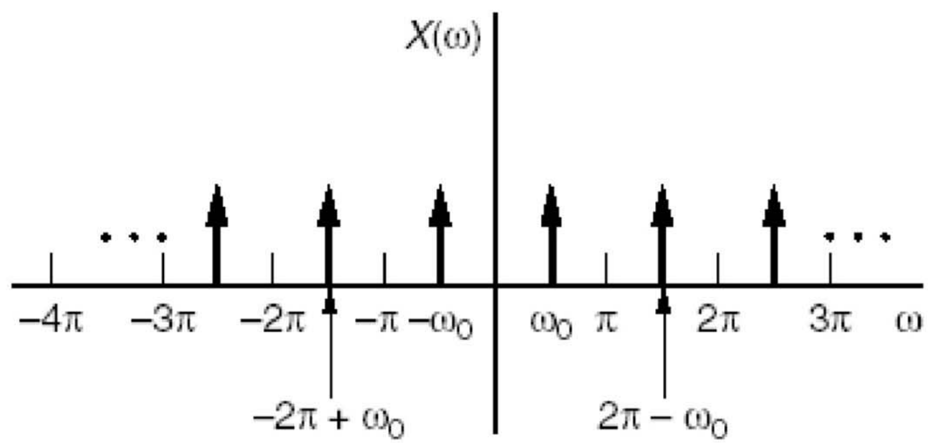
$$x[n] = \cos(\omega_0 n), \qquad -\infty < n < \infty$$

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

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



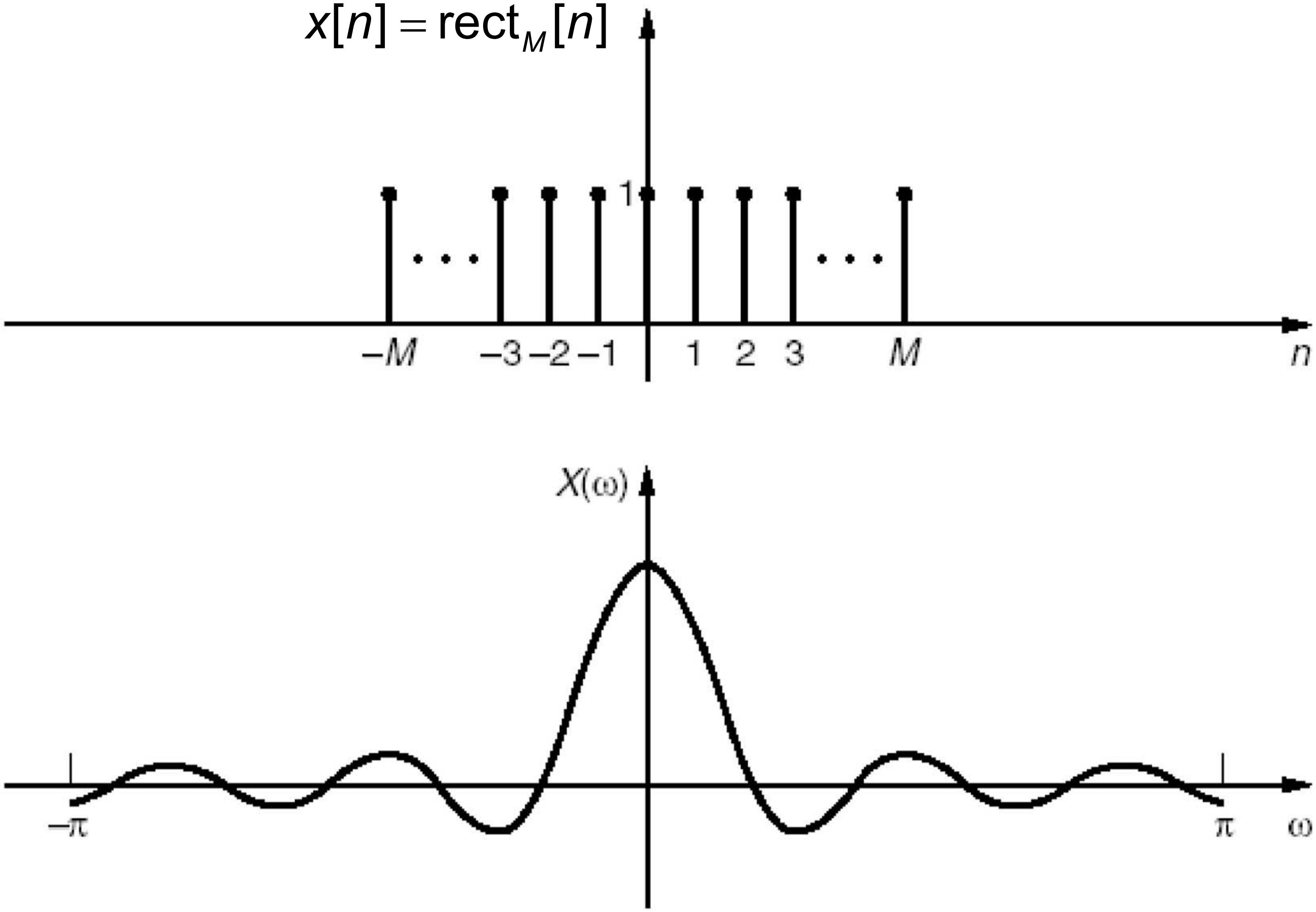
(a)



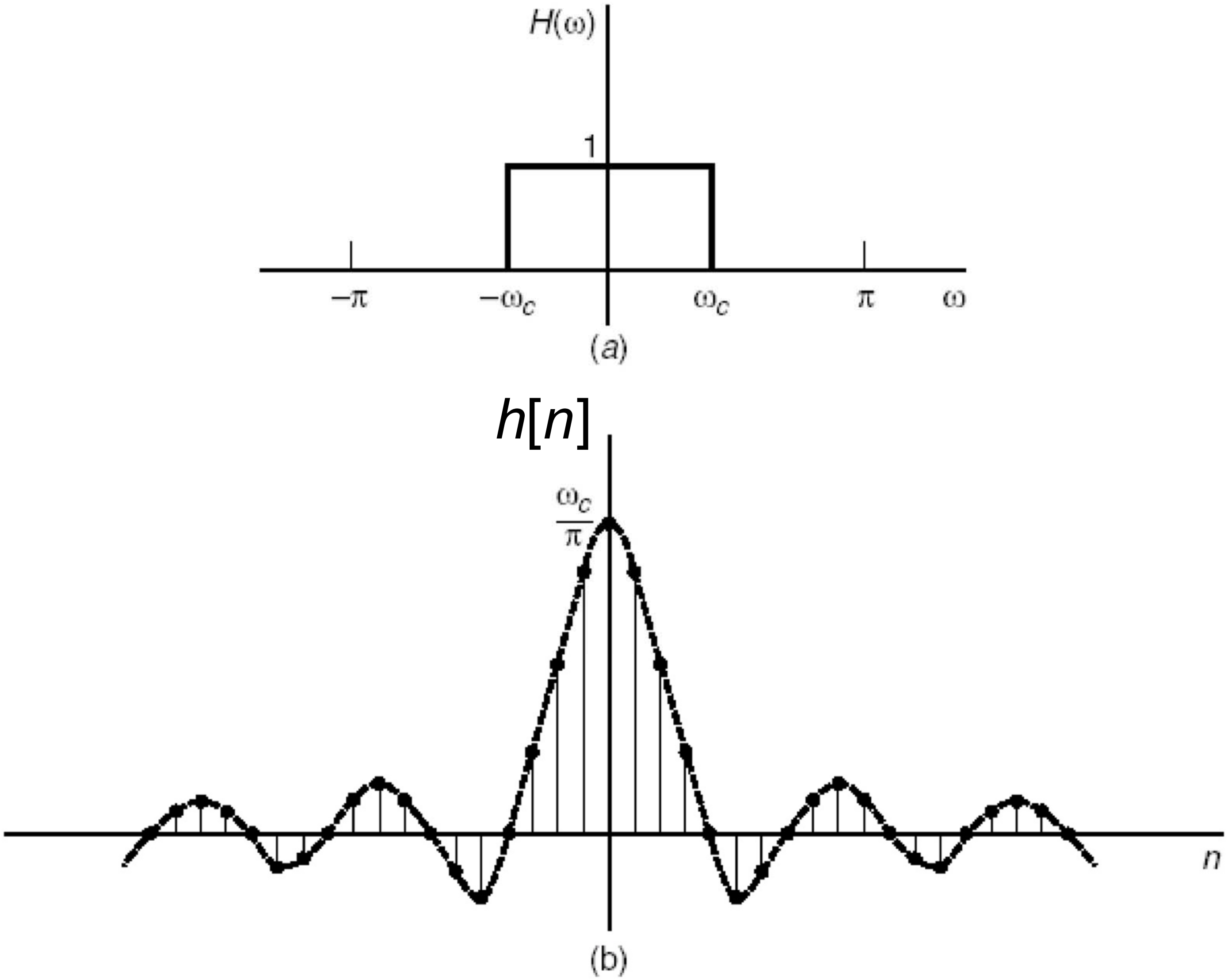
(b)

DTFT Examples

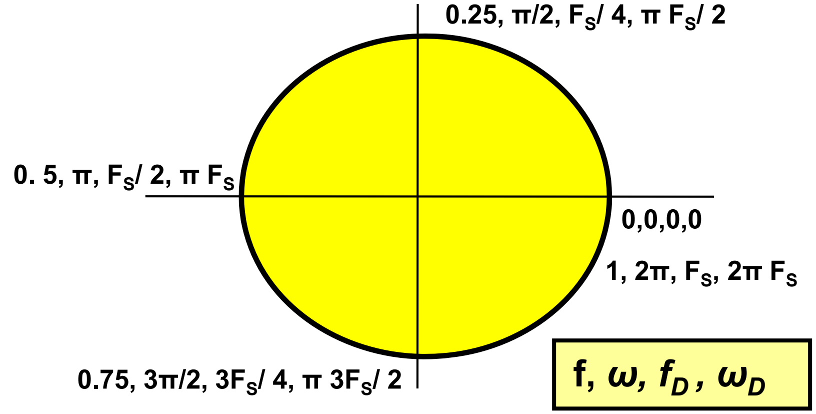
$$x[n] = \mathrm{rect}_M[n]$$



DTFT Examples



Fourier Transform Properties



- periodicity in ω

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

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

Units of Frequency (Digital Domain) (Trap #2 - loss of F_S)

- normalized frequency: f, 0→0.5→1 (independent of F_S)
- normalized radian frequency: ω, 0→ π→2 π (independent of F_S)
- digital frequency: $f_D = f * F_S$, 0→F_S/2→F_S
- digital radian frequency: $\omega_D = \omega * F_S$, 0→ πF_S→2πF_S

Periodic DT Signals

- A signal is periodic with period N if $x[n] = x[n+N]$ for all n
- For the complex exponential this condition becomes

$$Ae^{j\omega_0 n} = Ae^{j\omega_0 (n+N)} = Ae^{j(\omega_0 n + \omega_0 N)}$$

  which requires $\omega_0 N = 2\pi k$ for some integer k
- Thus, not all DT **sinusoids** are periodic!
- Consequence: there are N distinguishable frequencies with period N
  - *e.g.*, $\omega_k = 2\pi k/N, \quad k=0,1,\ldots,N\text{-}1$

Periodic DT Signals

Example 1:

$F_s = 10,000$ Hz

Is the signal $x[n] = \cos(2\pi \cdot 100n / F_s)$ a periodic signal?

If so, what is the period.

Solution:

If the signal is periodic with period N, then we have:

$$x[n] = x[n+N]$$

$$\cos(2\pi \cdot 100n / F_s) = \cos(2\pi \cdot 100(n+N) / F_s)$$

$$\frac{2\pi \cdot 100N}{F_s} = 2\pi \cdot k \ (k \text{ an integer})$$

$$k = \frac{100N}{F_s} = \frac{100N}{10,000} = \frac{N}{100}$$

For k an integer we get $N = 100k = 100$ (for $k = 1$)

Thus $x[n]$ is periodic of period 100 samples.

Periodic DT Signals

Example 2:

$F_s = 11059$ Hz; Is the signal

$x[n] = \cos(2\pi \cdot 100n / F_s)$ periodic? If so, what is the period.

Solution:

If the signal is periodic with period N, then we have:

$x[n] = x[n+N]$

$\cos(2\pi \cdot 100n / F_s) = \cos(2\pi \cdot 100(n+N) / F_s)$

$\dfrac{2\pi \cdot 100N}{F_s} = 2\pi \cdot k$ (k an integer)

$k = \dfrac{100N}{F_s} = \dfrac{100N}{11{,}059}$

For k an integer we get $N = \dfrac{11059}{100}k$ which is not an integer

Thus $x[n]$ is not periodic at this sampling rate.

Periodic DT Signals

Example 3:

$F_s = 10,000$ Hz

Is the signal $x[n] = \cos(2\pi \cdot 101n / F_s)$ a periodic signal?

If so, what is the period.

Solution:

If the signal is periodic with period N, then we have:

$$x[n] = x[n+N]$$

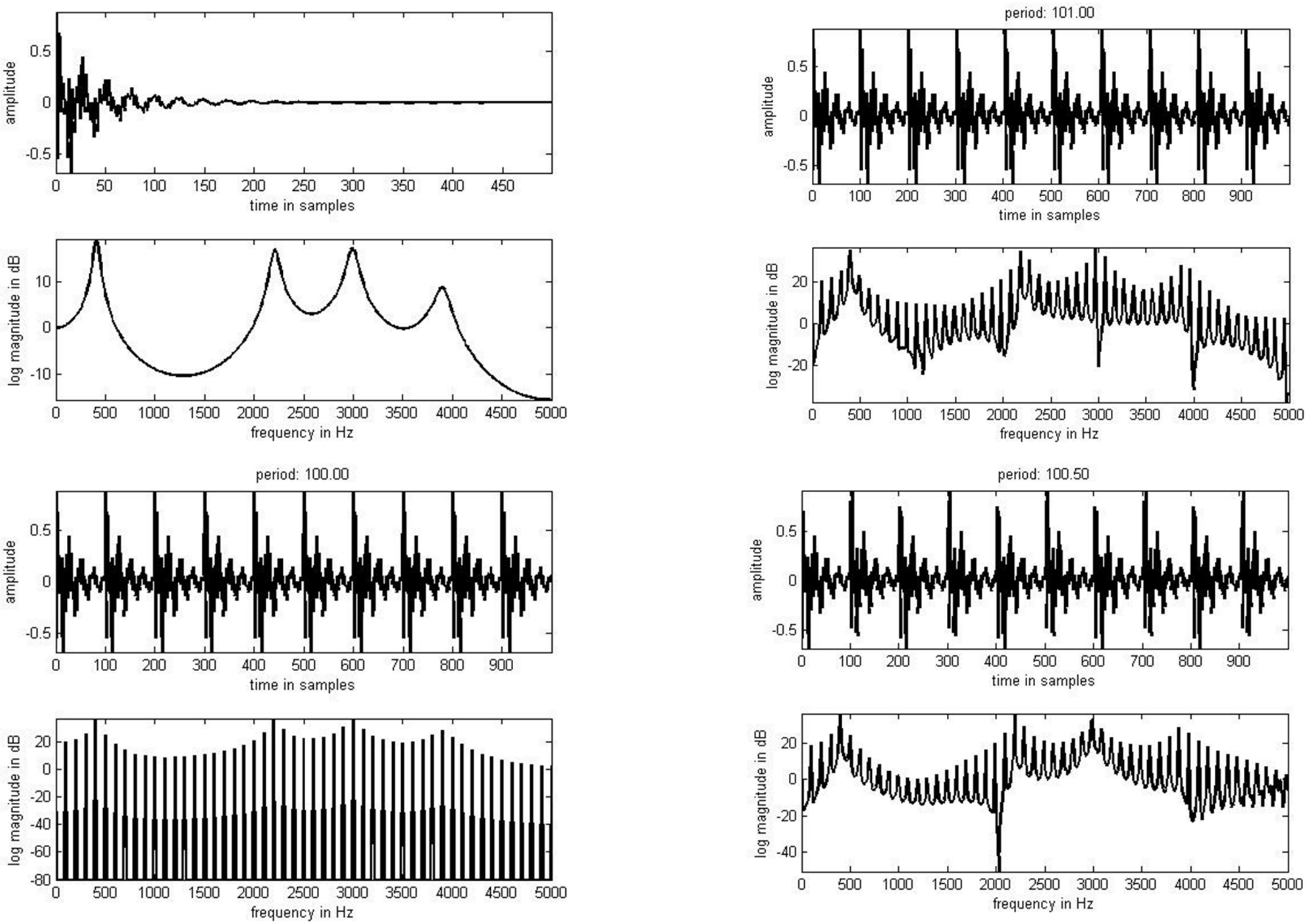
$$\cos(2\pi \cdot 101n / F_s) = \cos(2\pi \cdot 101(n+N) / F_s)$$

$$\frac{2\pi \cdot 101N}{F_s} = 2\pi \cdot k \ (k \text{ an integer})$$

$$k = \frac{101N}{F_s} = \frac{101N}{10,000} \text{ which is not an integer}$$

Thus $x[n]$ is not periodic at this sampling rate.

Periodic Sequences??



The DFT – Discrete Fourier Transform

Discrete Fourier Transform

- consider a periodic signal with period N (samples)

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

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

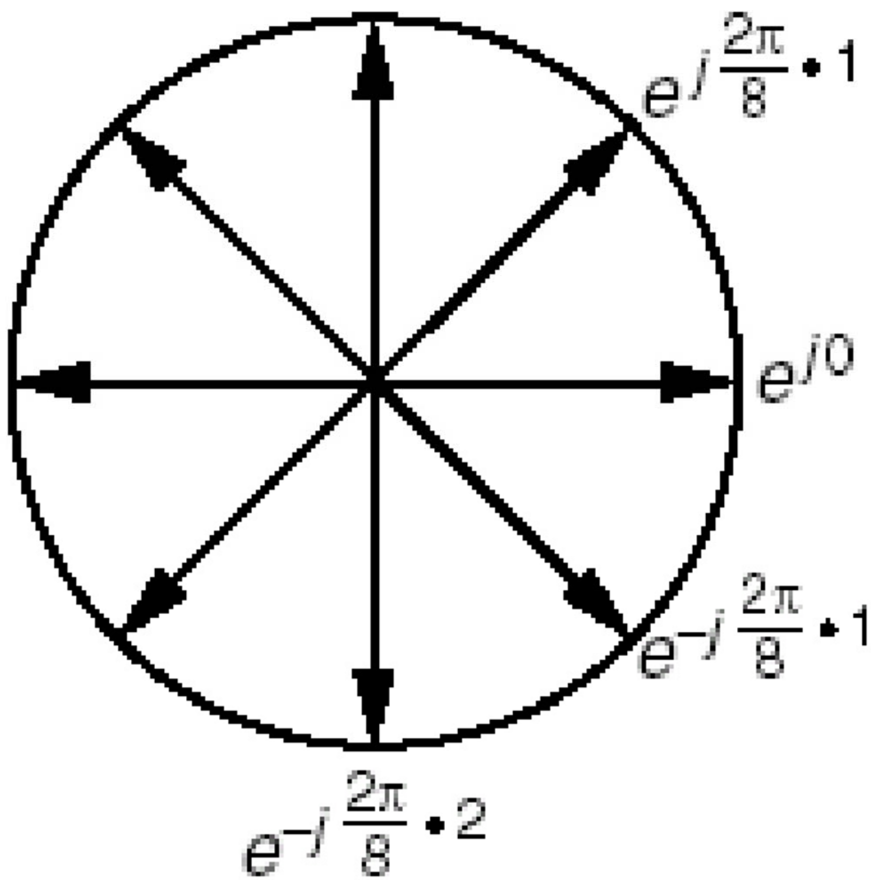
$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi kn/N}$$

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

- N sequence values
- N DFT coefficients

- exact representation of the discrete periodic sequence

DFT Unit Vectors (N=8)



$$k = 0;\ e^{-j2\pi k/8} = 1$$

$$k = 1;\ e^{-j2\pi k/8} = \frac{\sqrt{2}}{2}(1 - j)$$

$$k = 2;\ e^{-j2\pi k/8} = -j$$

$$k = 3;\ e^{-j2\pi k/8} = \frac{\sqrt{2}}{2}(-1 - j)$$

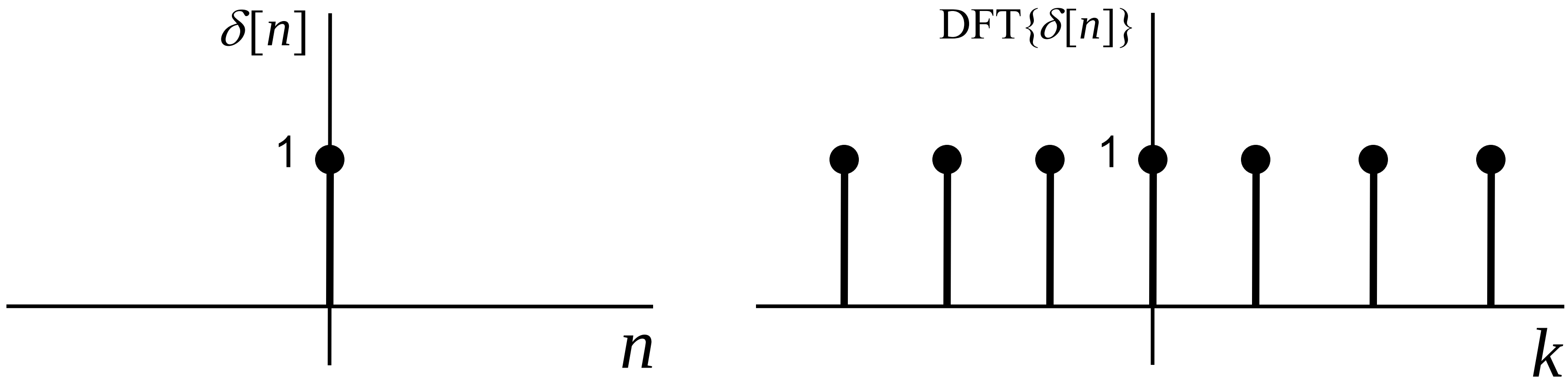
$$k = 4;\ e^{-j2\pi k/8} = -1$$

$$k = 5;\ e^{-j2\pi k/8} = \frac{\sqrt{2}}{2}(-1 + j)$$

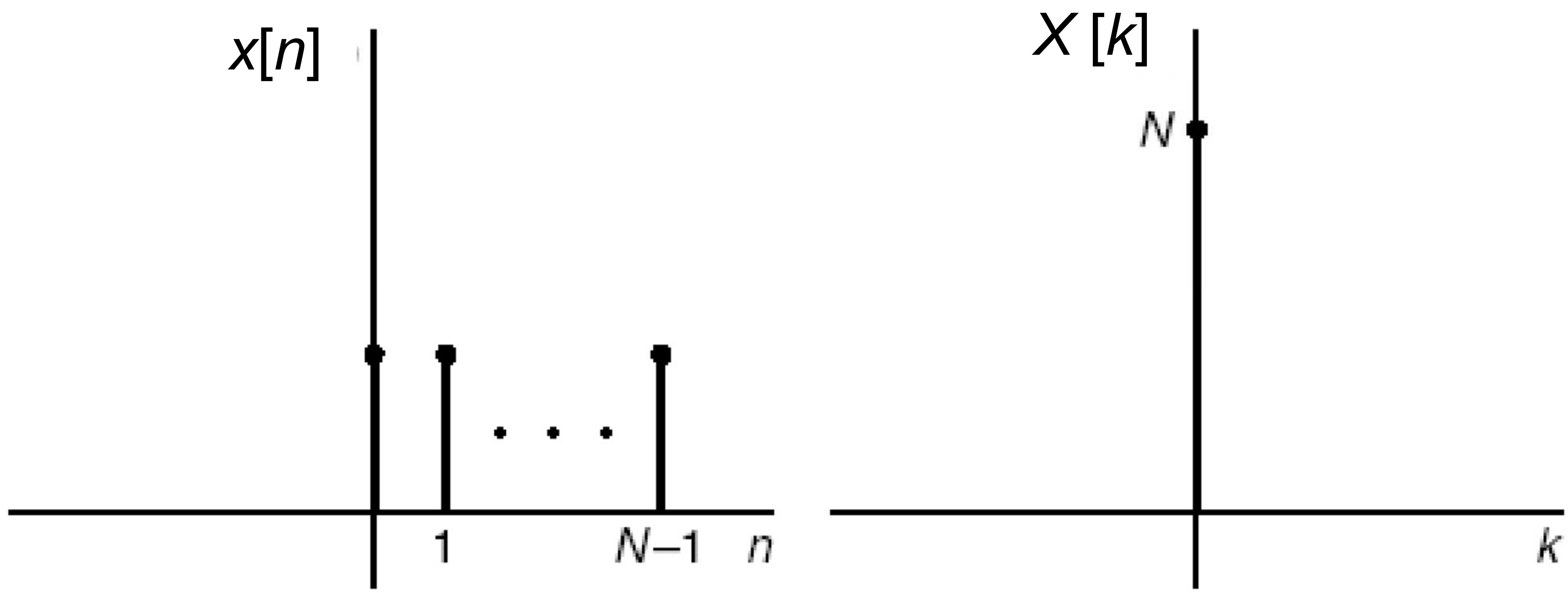
$$k = 6;\ e^{-j2\pi k/8} = j$$

$$k = 7;\ e^{-j2\pi k/8} = \frac{\sqrt{2}}{2}(1 + j)$$

DFT Examples

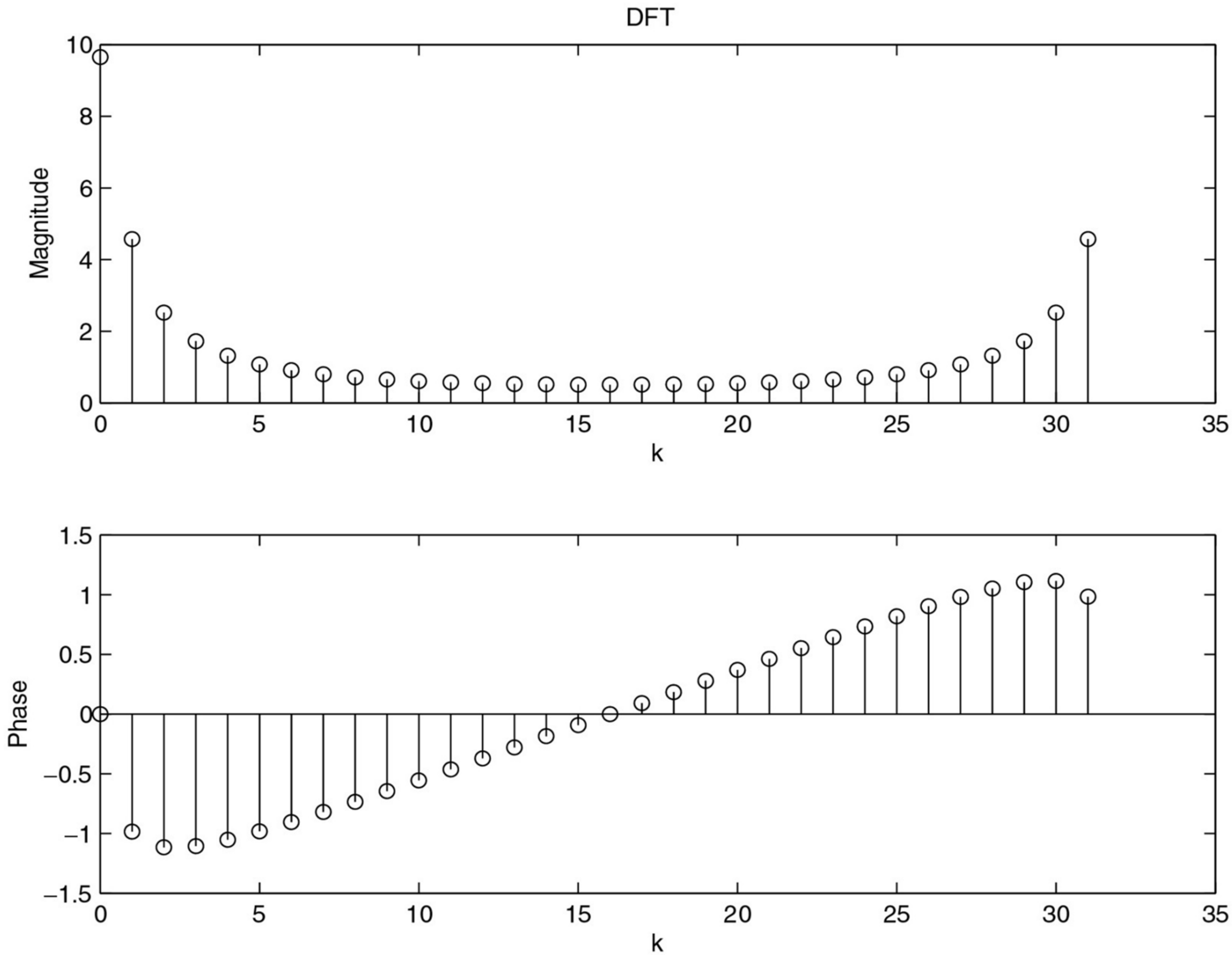


DFT Examples

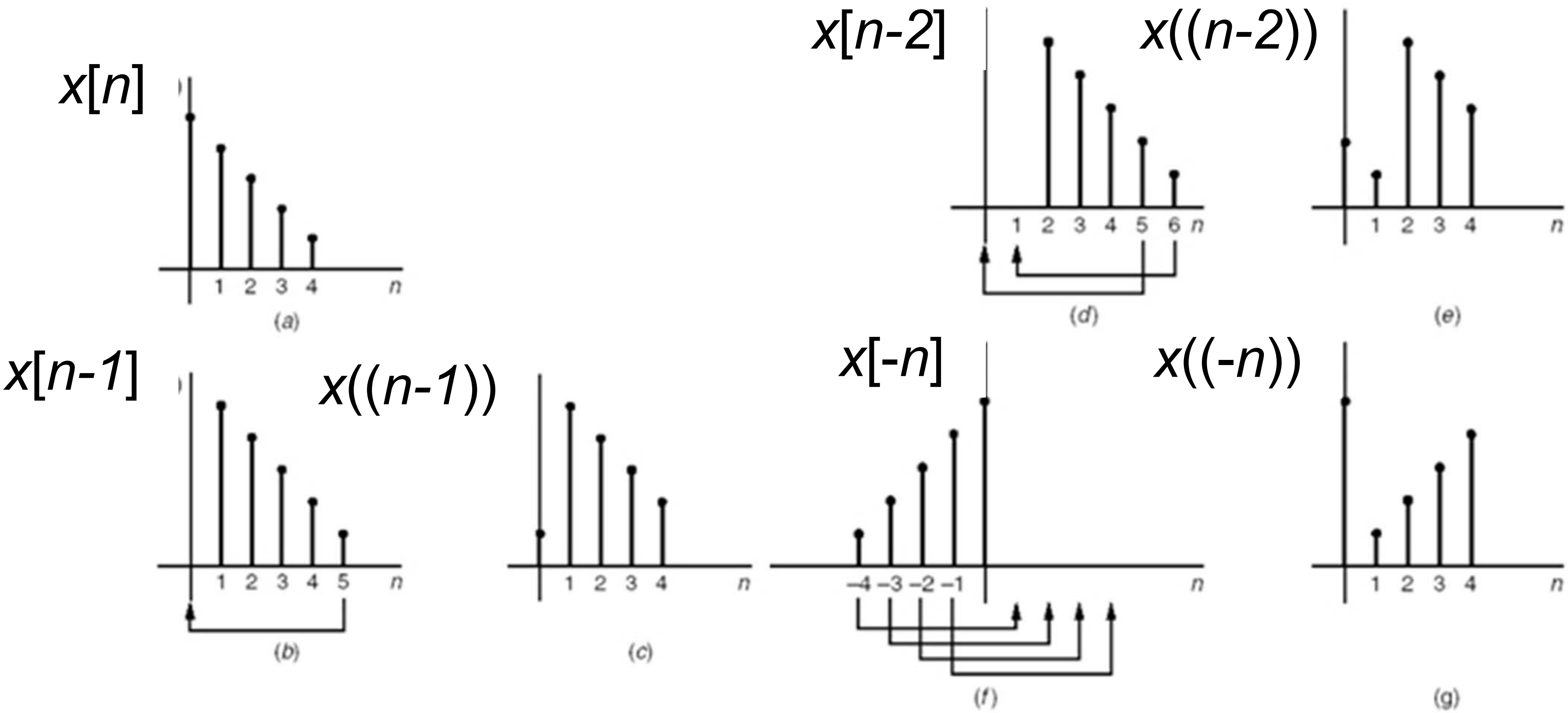


DFT Examples

$$\tilde{x}[n] = (0.9)^n \quad 0 \le n \le 31 \ (N = 32)$$



Circularly Shifting Sequences



Review

- DTFT of sequence $\{x[n], -\infty < n < \infty\}$

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

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

- DFT of periodic sequence $\{\tilde{x}[n], 0 \le n \le N-1\}$

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

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

DFT for Finite Length Sequences

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 N equally spaced points on the unit circle,

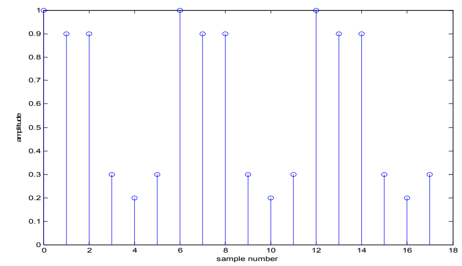
$$z_k = e^{j2\pi k/N}, \; k = 0, 1, \ldots, N-1$$

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

--looks like DFT of periodic sequence!

Relation to Periodic Sequence

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



$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x[n + r\,N]$$

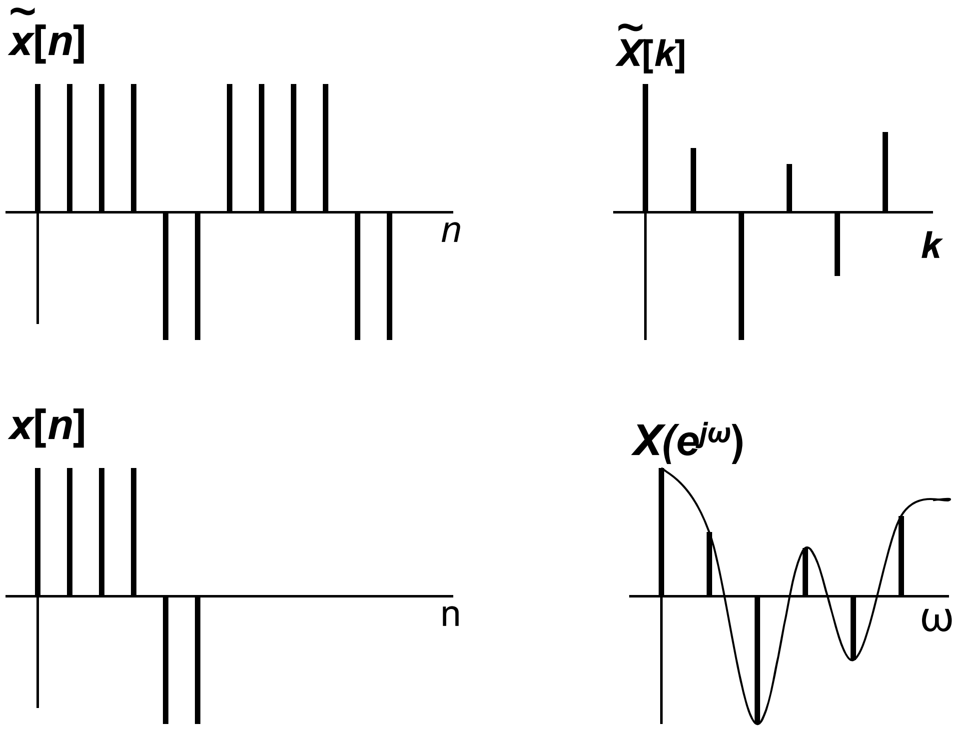
- the Fourier coefficients, $\tilde{X}[k]$, are then <u>identical</u> to the values of $X(e^{j2\pi k/N})$ for the finite duration sequence => <u>a sequence of length N can be exactly represented by a DFT representation</u> of the form:

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

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

Works for both finite sequence and for periodic sequence

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

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

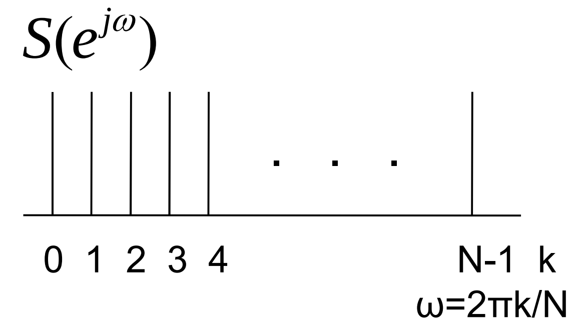
Thus we form the spectral sequence

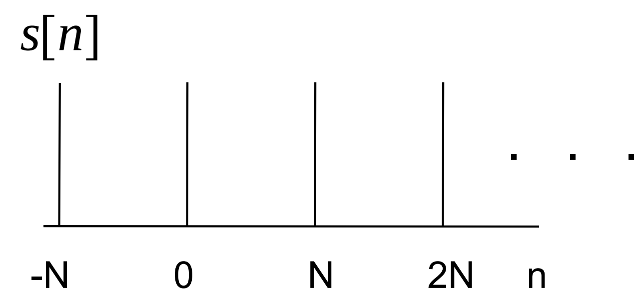
$$\tilde{X}(e^{j\omega}) = X(e^{j\omega}) \cdot S(e^{j\omega})$$

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.,

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

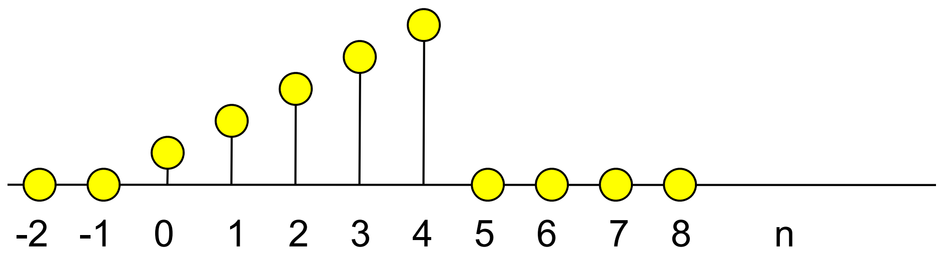
If $N < L,$ i.e., the number of frequency samples is smaller than the duration of the finite duration signal, then there is time domain aliasing and the resulting aliased signal (over the interval $0 \leq n \leq L-1$) satisfies the aliasing relation:

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

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 $n=0,1,\ldots,N-1$
DFT defined for all k	DTFT defined for all ω

- when using DFT representation, all sequences behave as if they were infinitely periodic => DFT is really the representation of the extended periodic function, $\tilde{x}[n]=\sum_{r=-\infty}^{\infty} x[n+rN]$
- alternative (equivalent) view is that all sequence indices must be interpreted modulo N

$$\tilde{x}[n]=\sum_{r=-\infty}^{\infty} x[n+rN]=x[n \text{ modulo } N]=x(\,[n]\,)_N$$

DFT Properties for Finite Sequences

- $X[k]$, the DFT of the finite sequence $x[n]$, can be viewed as a **<u>sampled version</u>** 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
1. Linearity	$ax_1[n]+bx_2[n]$	$aX_1[k]+bX_2[k]$
2. Shift	$x([n-n_0])_N$	$e^{-j2\pi kn_0/N}X[k]$
3. Time Reversal	$x([-n])_N$	$X^*[k]$
4. Convolution	$\sum_{m=0}^{N-1} x[m]\, h([n-m])_N$	$X[k]H[k]$
5. Multiplication	$x[n]\, w[n]$	$\frac{1}{N}\sum_{r=0}^{N-1} X[r]W([k-r])_N$

Key Transform Properties

$$y[n] = x_1[n] * x_2[n] \quad \Leftrightarrow \quad Y(e^{j\omega}) = X_1(e^{j\omega}) \cdot X_2(e^{j\omega})$$

convolution multiplication

$$y[n] = x_1[n] \cdot x_2[n] \quad \Leftrightarrow \quad Y(e^{j\omega}) = X_1(e^{j\omega}) \otimes X_2(e^{j\omega})$$

multiplication circular convolution

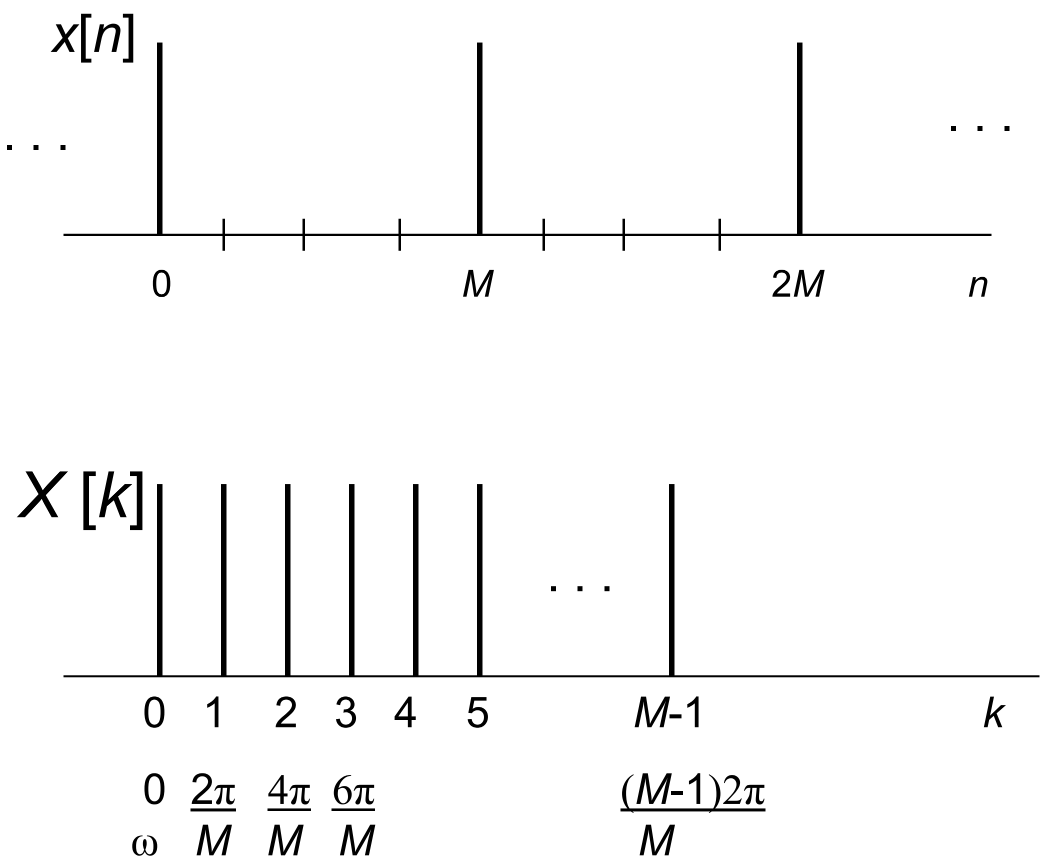
Special Case: $x_2[n] =$ impulse train of period M samples

$$x_2[n] = \sum_{k=-\infty}^{\infty} \delta[k - nM]$$

$$X_2[k] = \sum_{n=0}^{M-1} \delta[n] e^{-j2\pi nk/M} = 1, \quad k = 0,1,\ldots,M-1$$

$$x_2[n] = \frac{1}{M}\sum_{k=0}^{M-1} X_2[k] e^{j2\pi nk/M} = \frac{1}{M}\sum_{k=0}^{M-1} e^{j2\pi nk/M}$$ sampling function

Sampling Function



Summary of DSP-Part 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

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 with $H(e^{j\omega})$

as the complex frequency response

$$H(e^{j\omega}) = H_r(e^{j\omega}) + jH_i(e^{j\omega})$$ real, imaginary representation

$$H(e^{j\omega}) = |H(e^{j\omega})|\, e^{j \arg|H(e^{j\omega})|}$$ magnitude, phase representation

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

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

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

Digital Filters

- causal linear shift-invariant => $h[n]=0$ for $n<0$
- stable system => every bounded input produces a bounded output => a necessary and sufficient condition for stability and for the existence of $H(e^{j\omega})$

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

Digital Filter Implementation

- input and output satisfy linear difference equation 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:

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

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

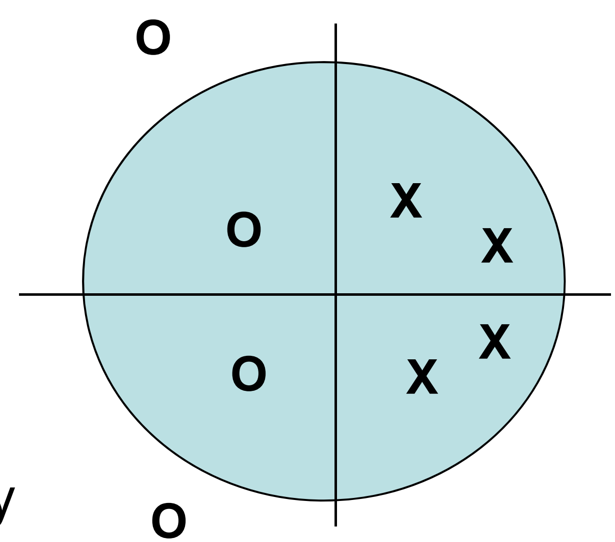
$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{r=0}^{M}b_r z^{-r}}{1-\sum_{k=1}^{N}a_k z^{-k}}$$

canonic form showing poles and zeros

Digital Filters

- *H(z)* is a rational function in z^{-1}

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

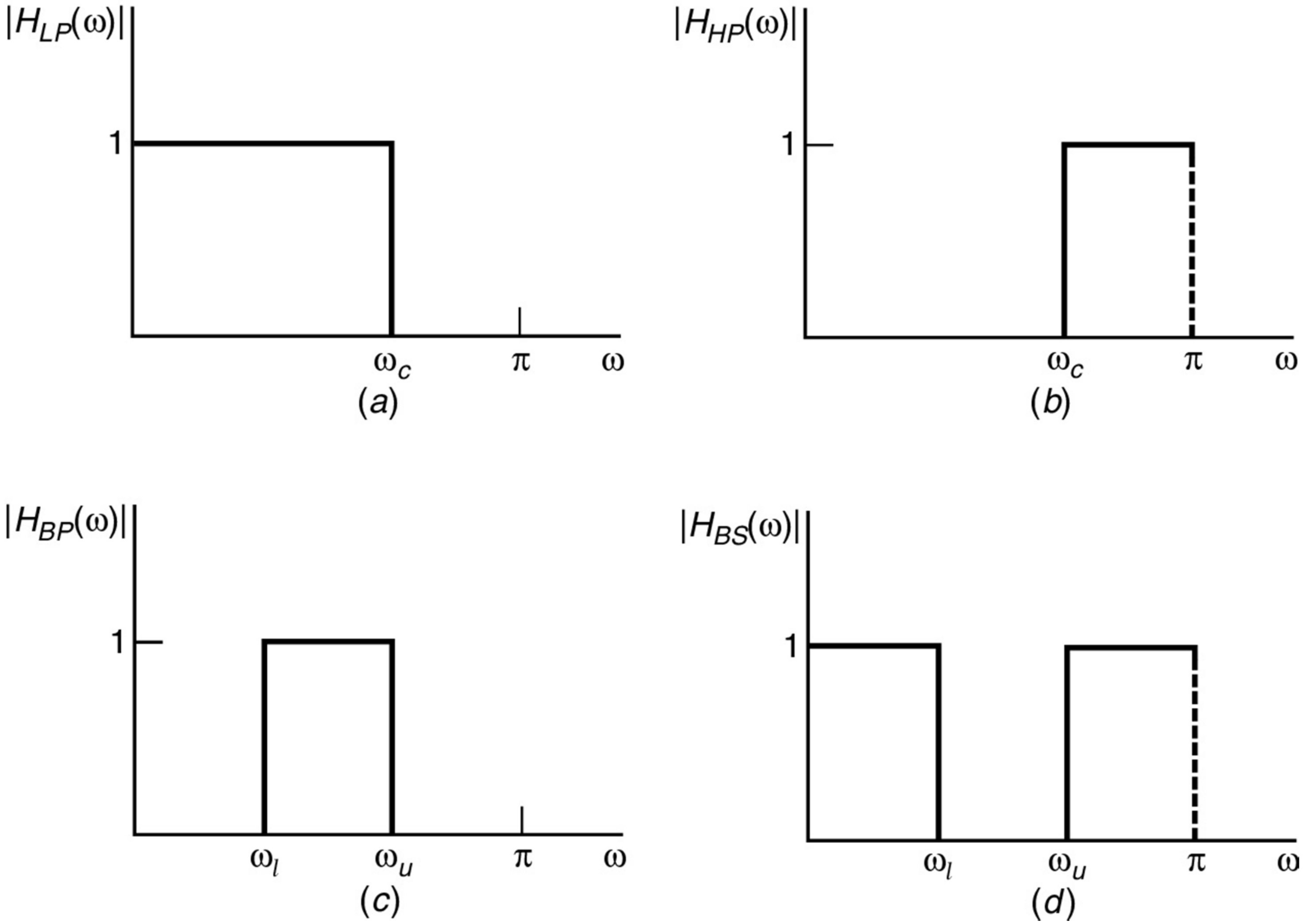


- converges for $|z| > R_1$, with $R_1 < 1$ for stability

=>

all poles of $H(z)$ inside the unit circle for a stable, causal system

Ideal Filter Responses



FIR Systems

- if a_k=0, all k, then

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

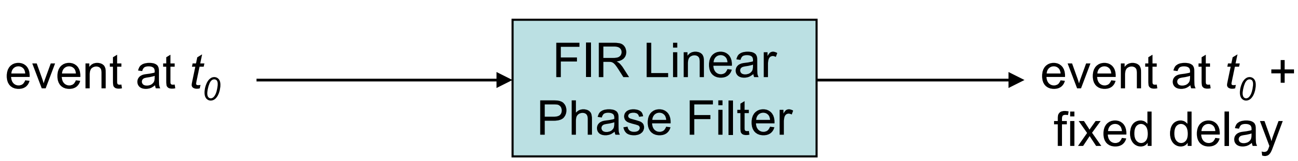
$$1.\ \ h[n] = b_n \quad 0 \le n \le M$$
$$= 0 \quad \text{otherwise}$$

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

$$3.\ \text{if } h[n] = \pm\, h[M-n] \text{ (symmetric, antisymmetric)}$$

$$H(e^{j\omega}) = A(e^{j\omega}) e^{-j\omega M/2}, \ \ A(e^{j\omega}) = \text{ real (symmetric), imaginary (anti-symmetric)}$$

- <u>linear phase filter</u> => no signal dispersion 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 design => analytical, closed form method
  - frequency sampling => optimization method
  - minimax error design => optimal method

Window Designed Filters

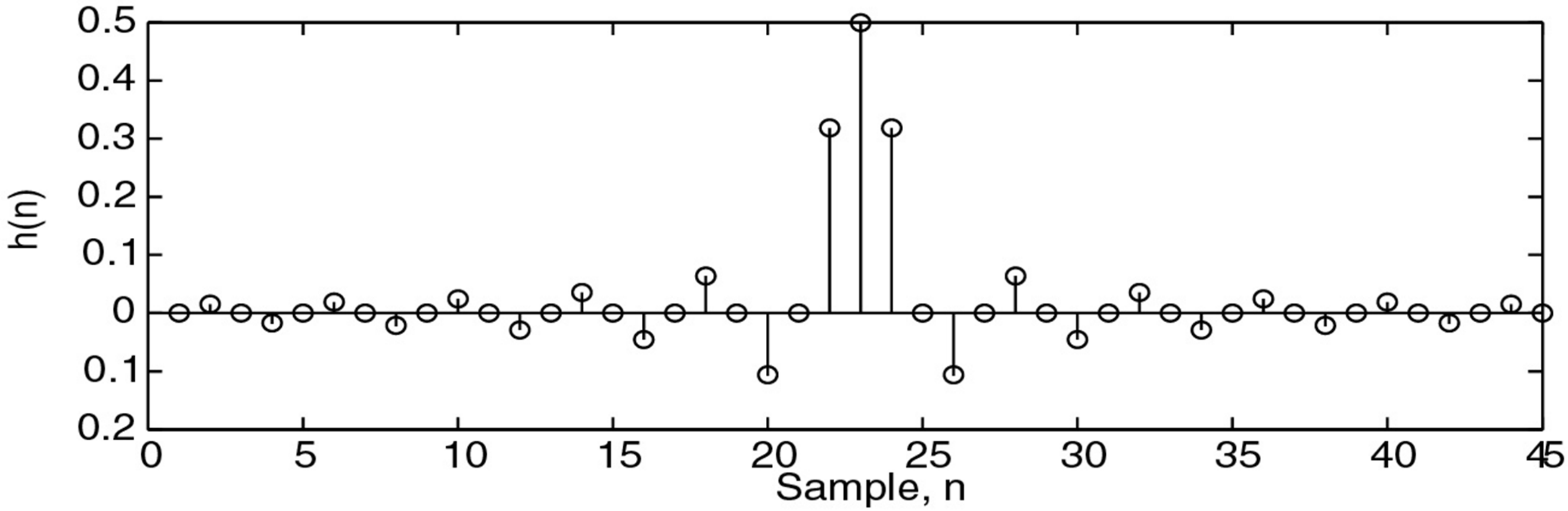
Windowed impulse response

$$h[n] = h_l[n] \cdot w[n]$$

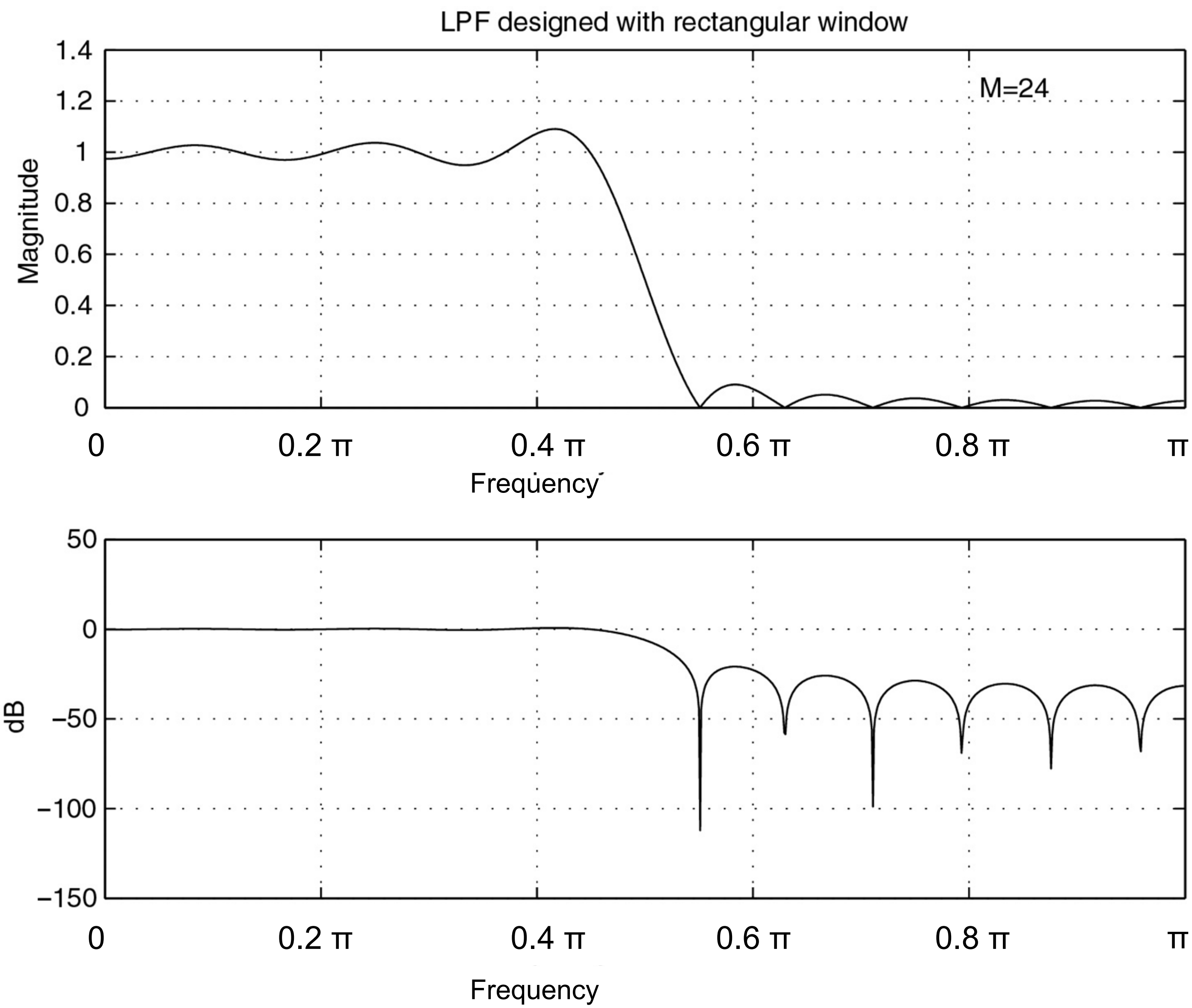
In the frequency domain we get

$$H(e^{j\omega}) = H_l(e^{j\omega}) * W(e^{j\omega})$$

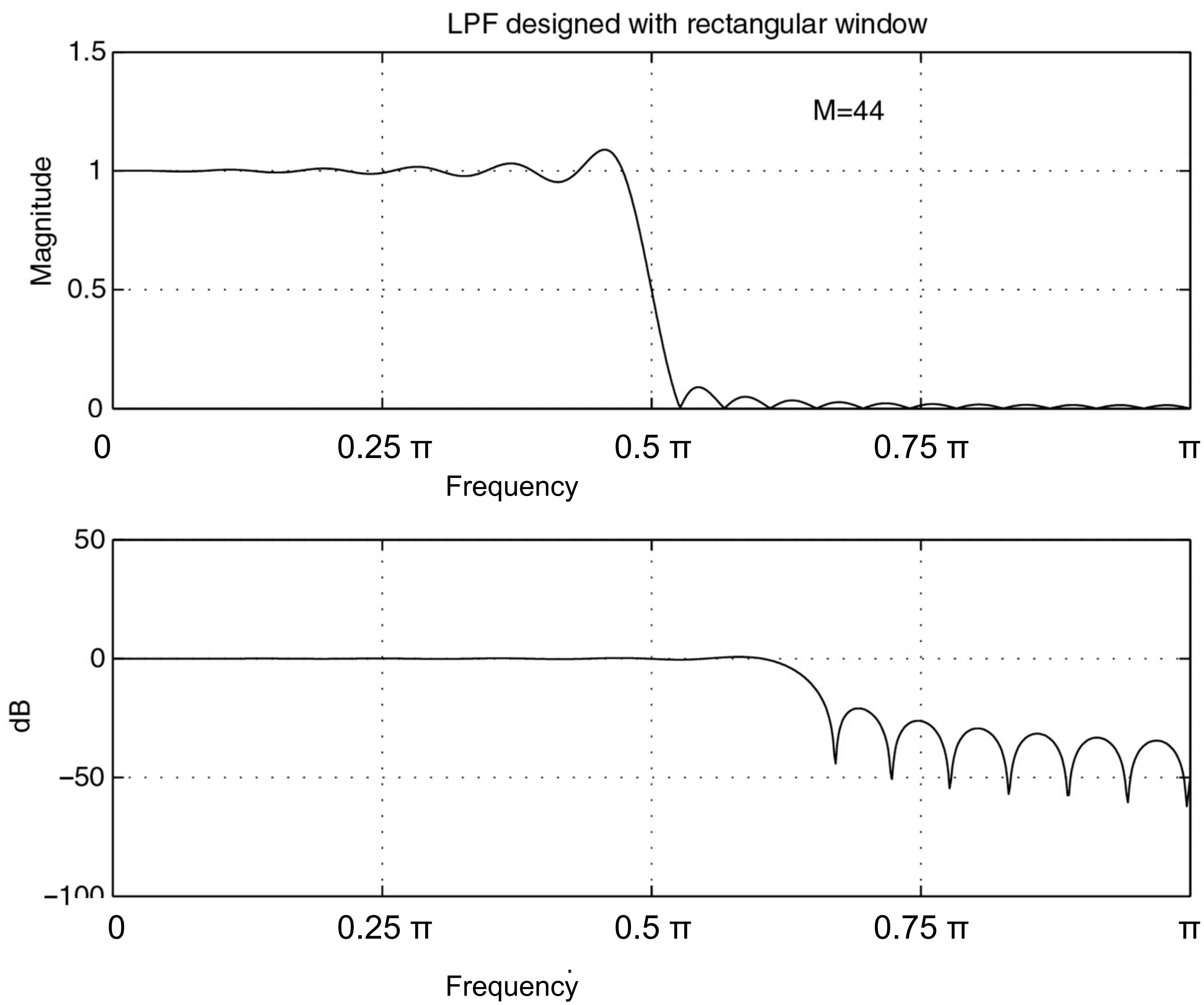
LPF Example Using RW



LPF Example Using RW



LPF Example Using RW



Common Windows (Time)

1. Rectangular $w[n]=\begin{cases}1 & 0\le n\le M\\ 0 & otherwise\end{cases}$

2. Bartlett $w[n]=1-\dfrac{2\,|\,n-M/2\,|}{M}$

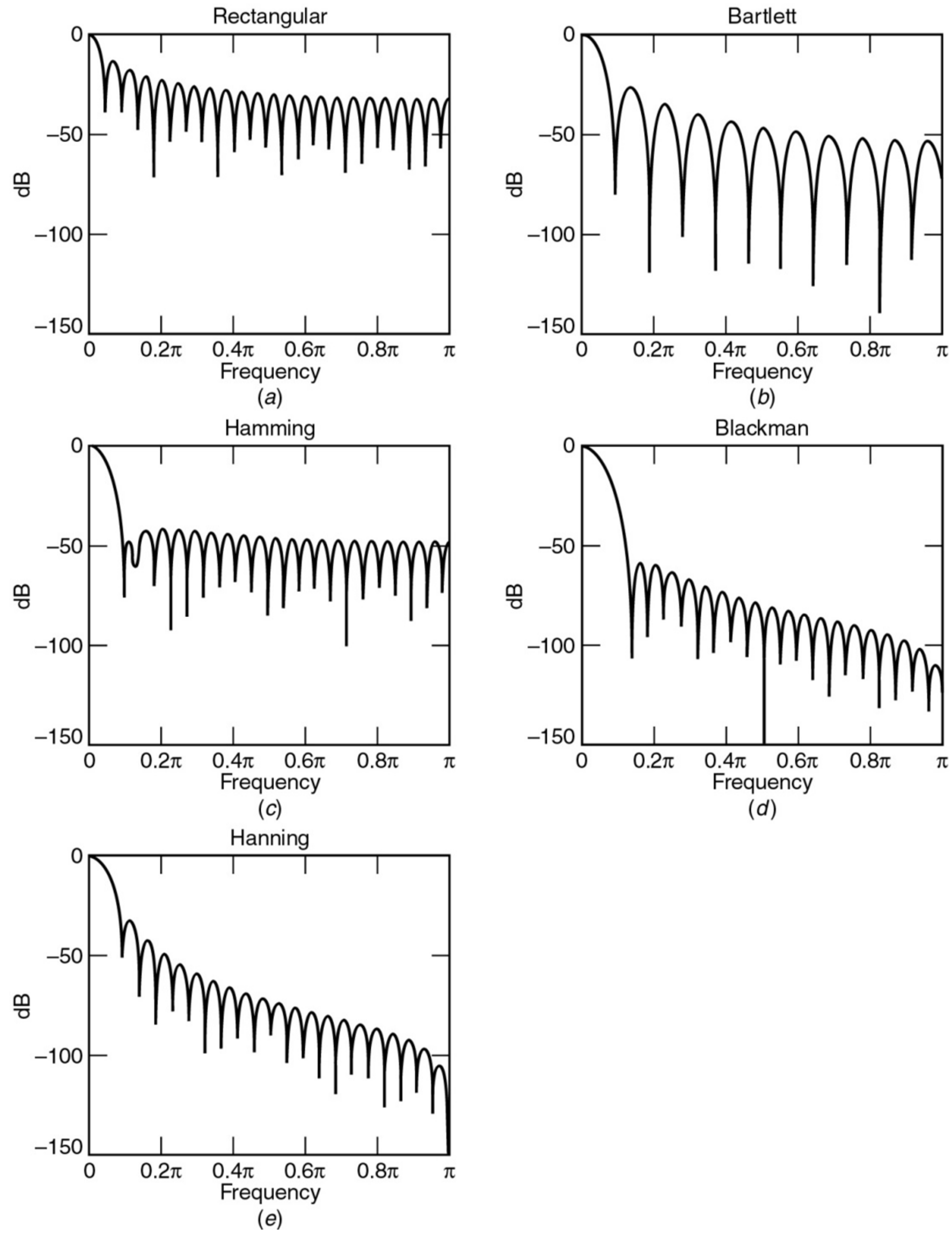
3. Blackman $w[n]=0.42-0.5\cos\left(\dfrac{2\pi n}{M}\right)+0.08\cos\left(\dfrac{4\pi n}{M}\right)$

4. Hamming $w[n]=0.54-0.46\cos\left(\dfrac{2\pi n}{M}\right)$

5. Hanning $w[n]=0.5-0.5\cos\left(\dfrac{2\pi n}{M}\right)$

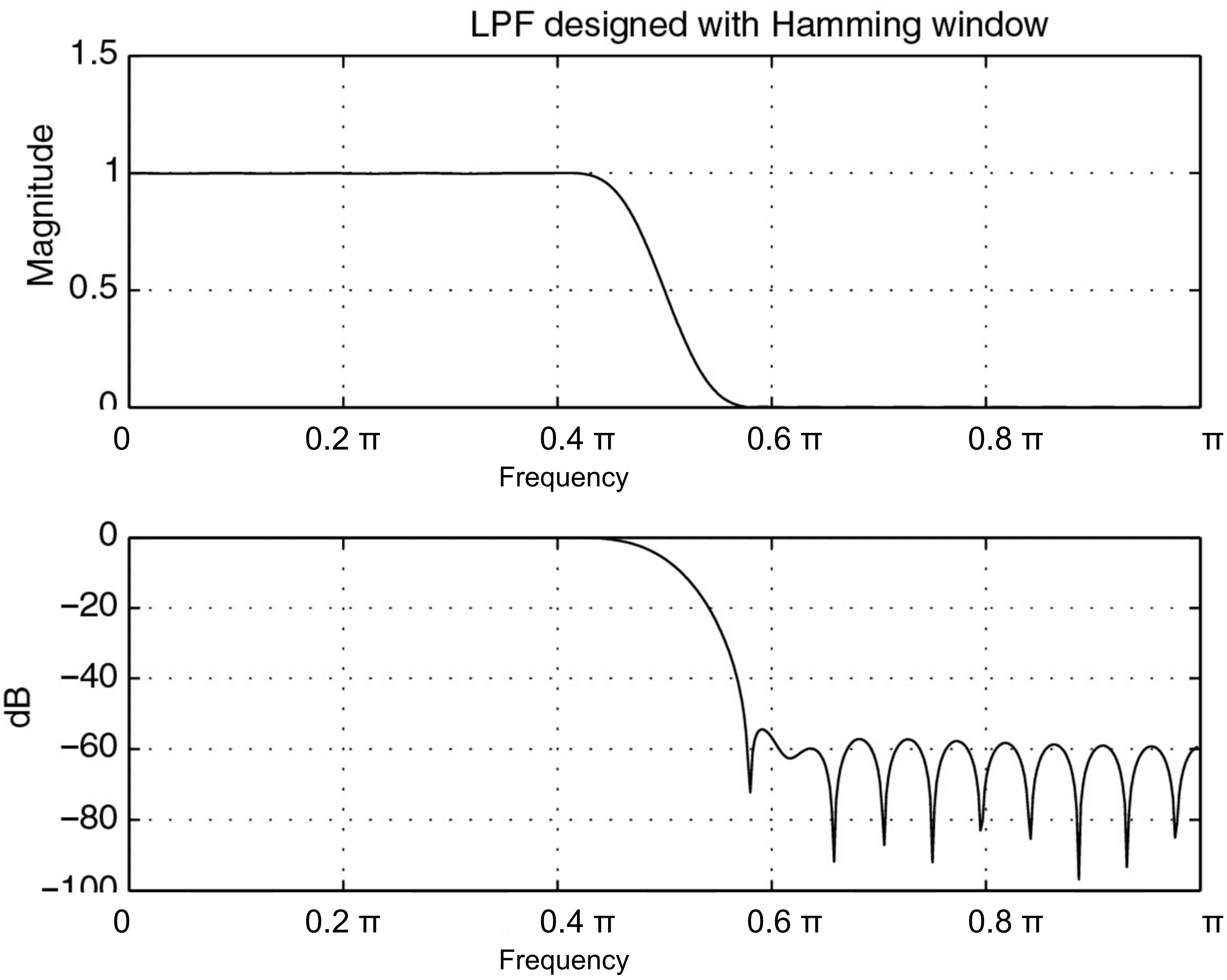
6. Kaiser $w[n]=\dfrac{I_0\left\{\beta\sqrt{1-((n-M/2)/(M/2))^2}\right\}}{I_0\{\beta\}}$

Common Windows (Frequency)

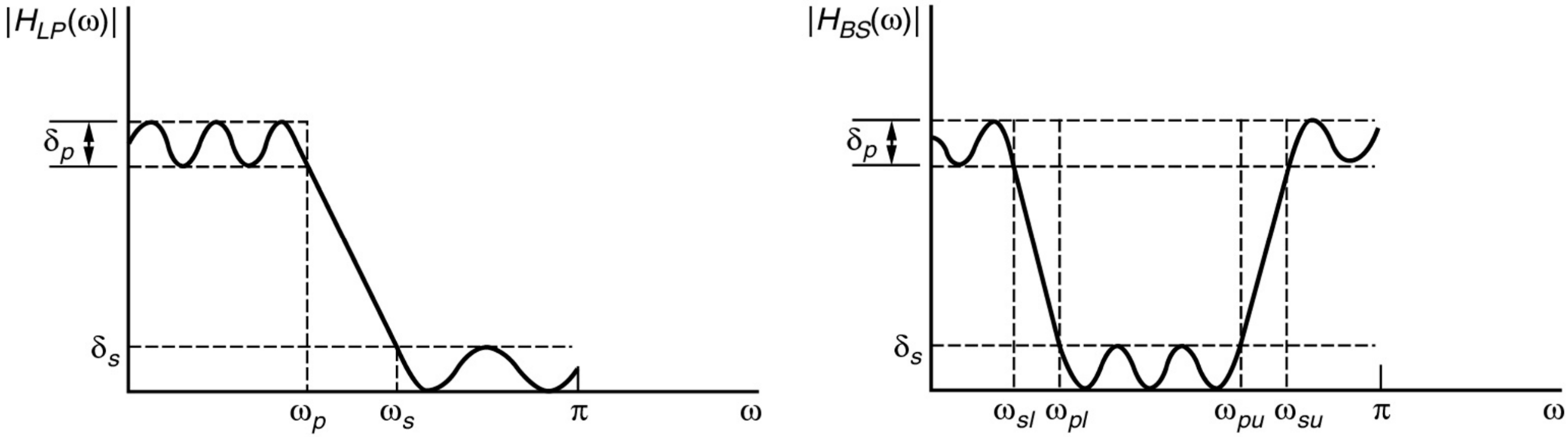


Window	Mainlobe Width	Sidelobe Attenuation
Rectangular	$4\pi / M$	-13 dB
Bartlett	$8\pi/M$	-27 dB
Hanning	$8\pi/M$	-32 dB
Hamming	$8\pi/M$	-43 dB
Blackman	$12\pi/M$	-58 dB

Window LPF Example



Equiripple Design Specifications



(*a*) (*b*)

$\omega_p =$ normalized edge of passband frequency

$\omega_s =$ normalized edge of stopband frequency

$\delta_p =$ peak ripple in passband

$\delta_s =$ peak ripple in stopband

$\Delta\omega = \omega_s - \omega_p =$ normalized transition bandwidth

Optimal FIR Filter Design

- Equiripple in each defined band (passband and stopband for lowpass filter, high and low stopband and passband for bandpass filter, etc.)
- Optimal in sense that the cost function

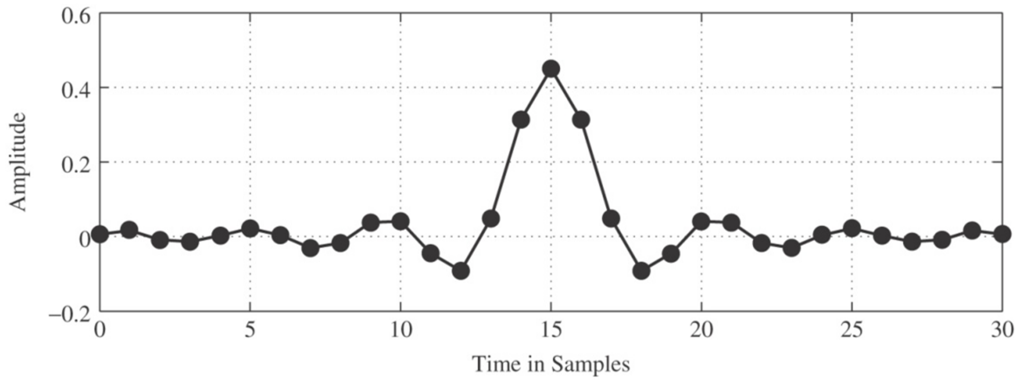
$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \beta(\omega) \,|\, H_d(\omega) - H(\omega) \,|^2 d\omega$$

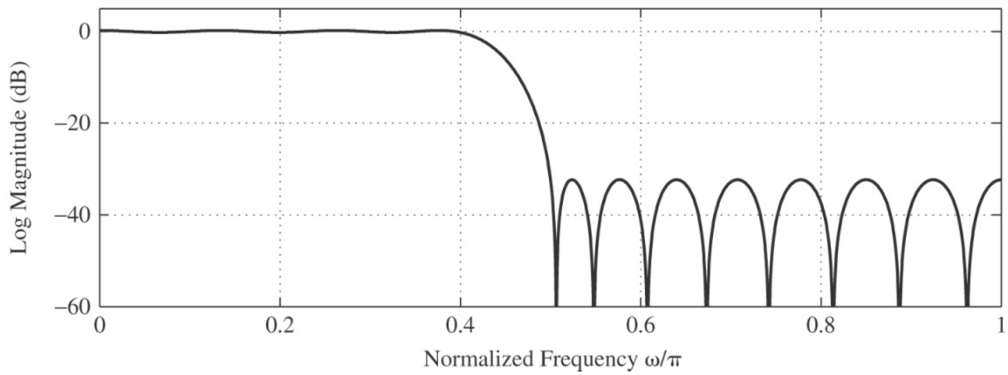
is minimized. Solution via well known iterative algorithm based on the alternation theorem of Chebyshev approximation.

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)))

Remez Lowpass Filter Design

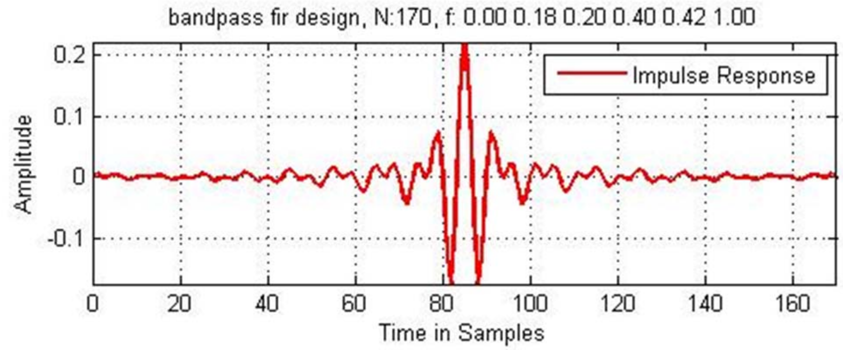


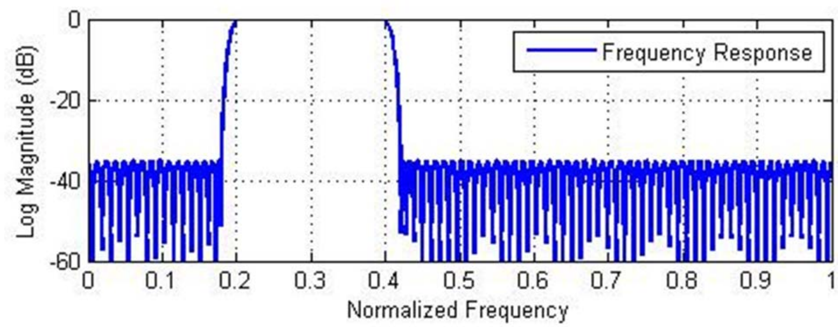



```
N=30
F=[0 0.4 0.5 1];
A=[1 1 0 0];
B=firpm(N,F,A)

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

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

Remez Bandpass Filter Design

```
% bandpass_filter_design
   N=input('Filter Length in Samples:');
   F=[0 0.18 .2 .4 .42 1];
   A=[0 0 1 1 0 0];
   B=firpm(N,F,A);
   NF=1024;
   [H,W]=freqz(B,1,NF);

figure,orient landscape;
stitle=sprintf('bandpass fir design,
N:%d,f: %4.2f %4.2f %4.2f %4.2f %4.2f
%4.2f',N,F);
  n=0:N;
  subplot(211),plot(n,B,'r','LineWidth',2);
  axis tight,grid on,title(stitle);
  xlabel('Time in Samples'),ylabel('Amplitude');
  legend('Impulse Response');

subplot(212),plot(W/pi,20*log10(abs(H)),'b','LineWidth',2);
 axis ([0 1 -60 0]), grid on;
 xlabel('Normalized Frequency'),ylabel('Log Magnitude (dB)');
 legend('Frequency Response');
```





FIR Implementation

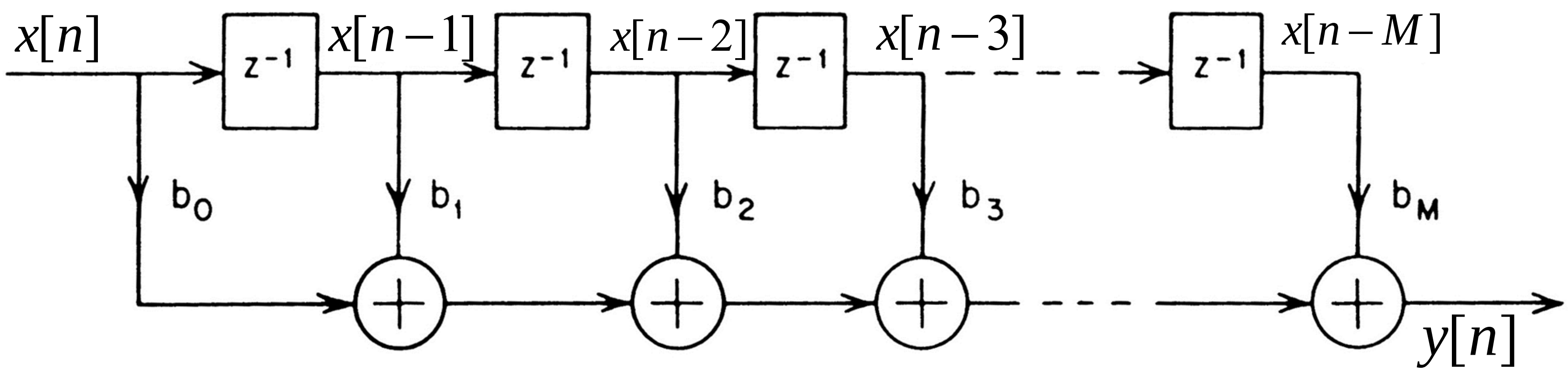


Fig. 2.5 Digital network for FIR system.

• 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 $y[n-1],y[n-2],...,y[n-N]$ as well as $x[n],x[n-1],...,x[n-M]$
- 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

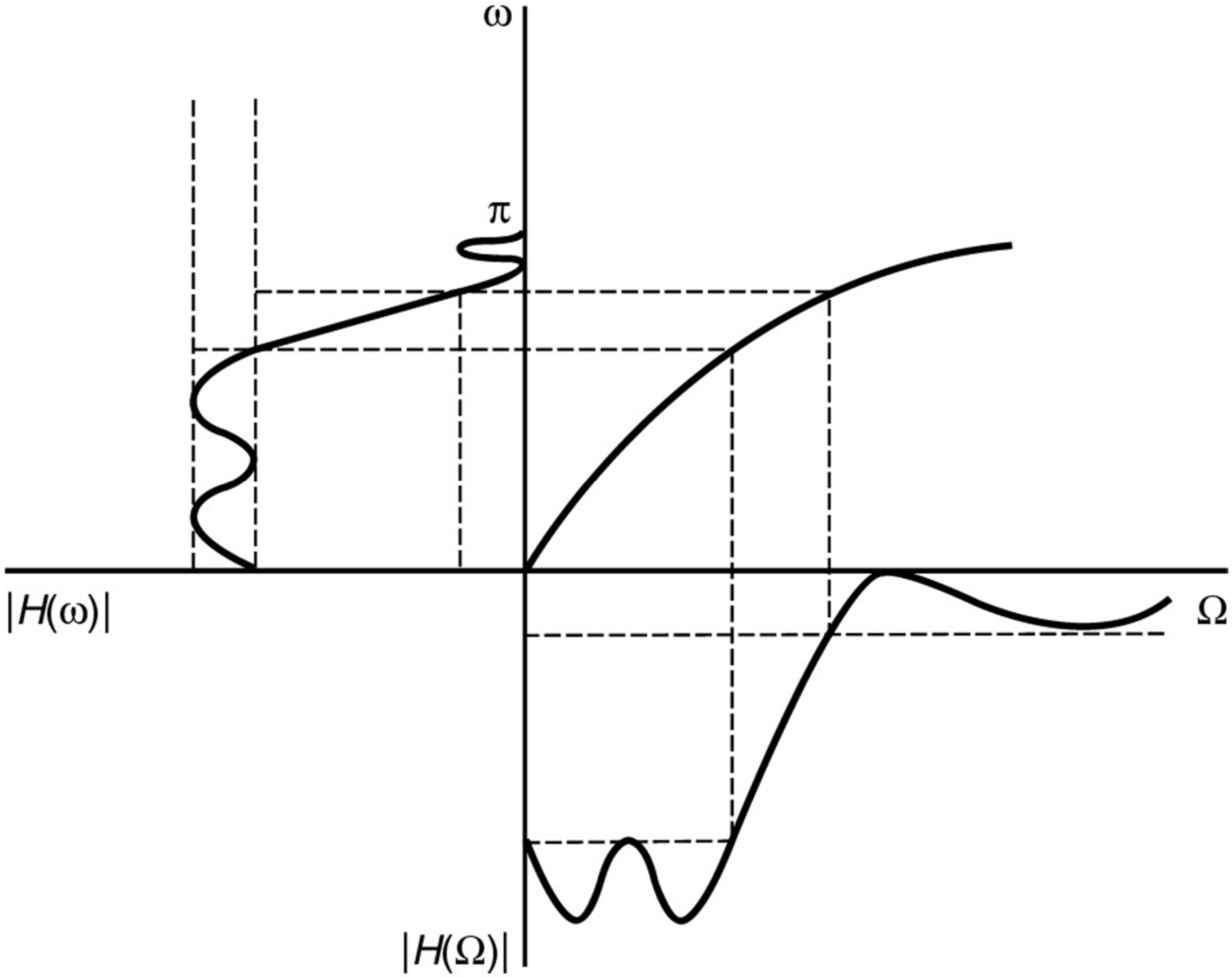
$$h[n]=\sum_{k=1}^{N}A_k(d_k)^n u[n]$$ - for causal systems

$h[n]$ is an <u>infinite duration impulse response</u>

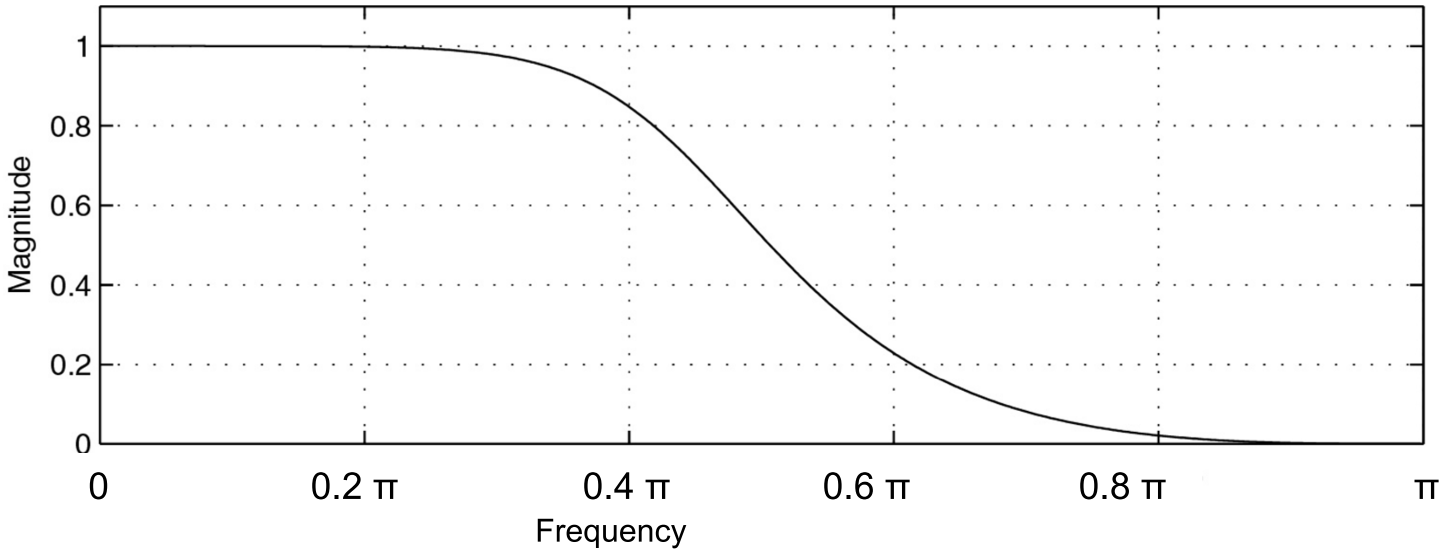
IIR Design Methods

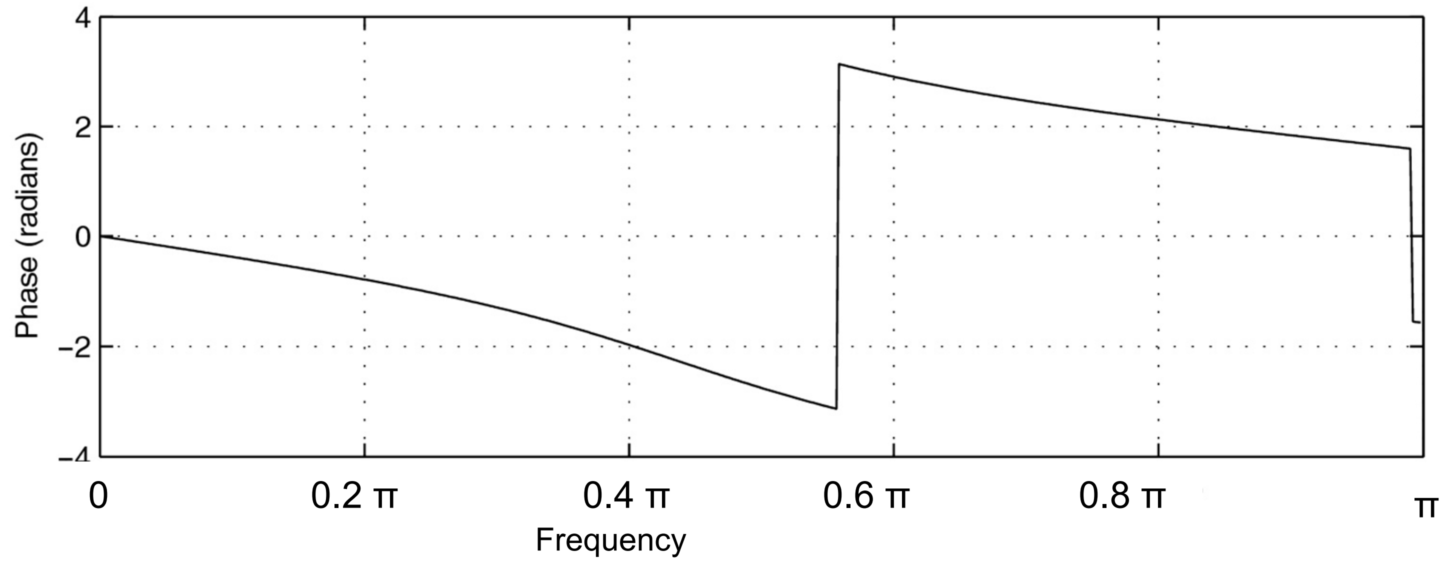
- **Impulse invariant transformation** – match the analog impulse response by sampling; resulting frequency response is aliased version of analog frequency response
- **Bilinear transformation** – use a transformation to map an analog filter to a digital filter by warping the analog frequency scale (0 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)

IIR Filter Design

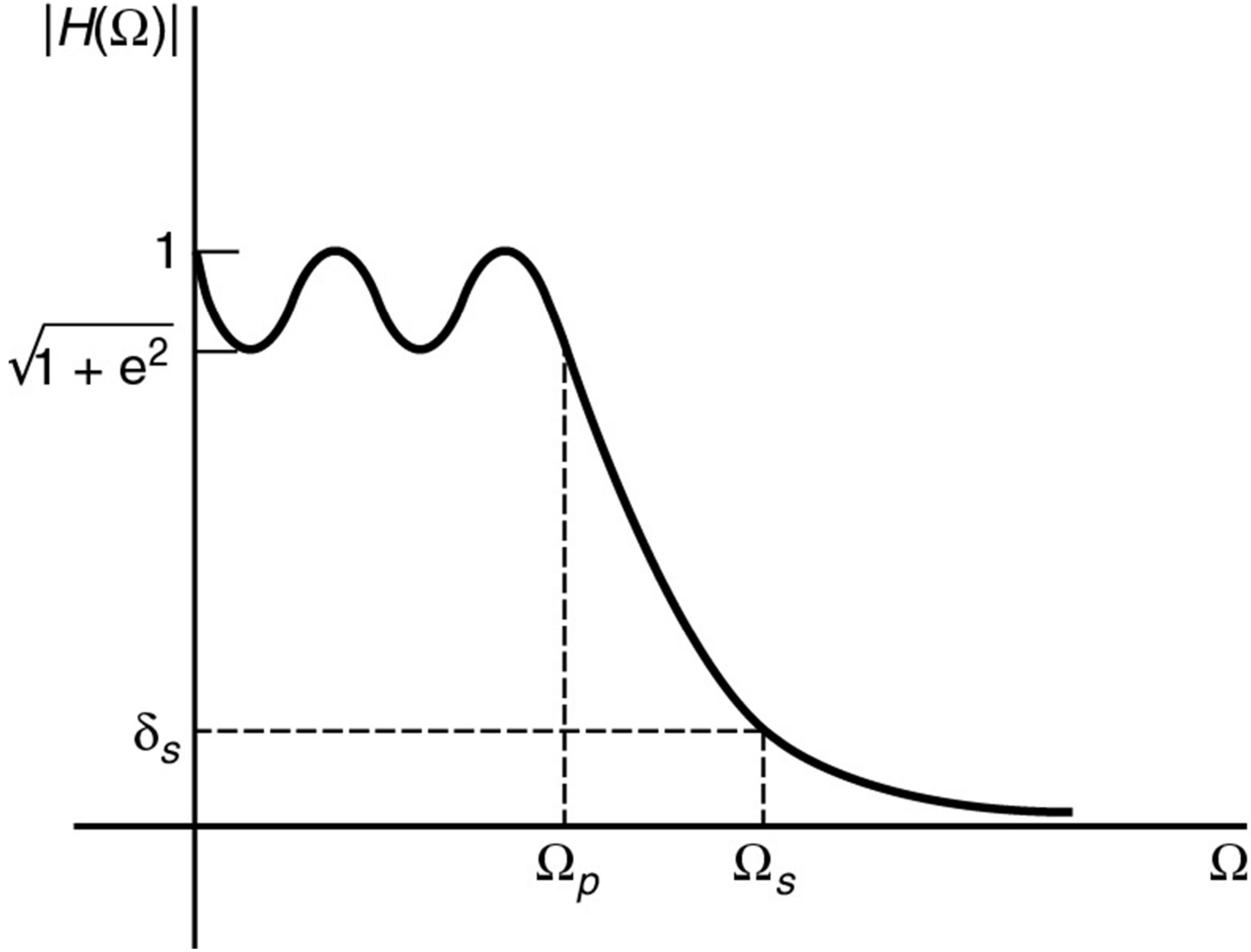


Butterworth Design

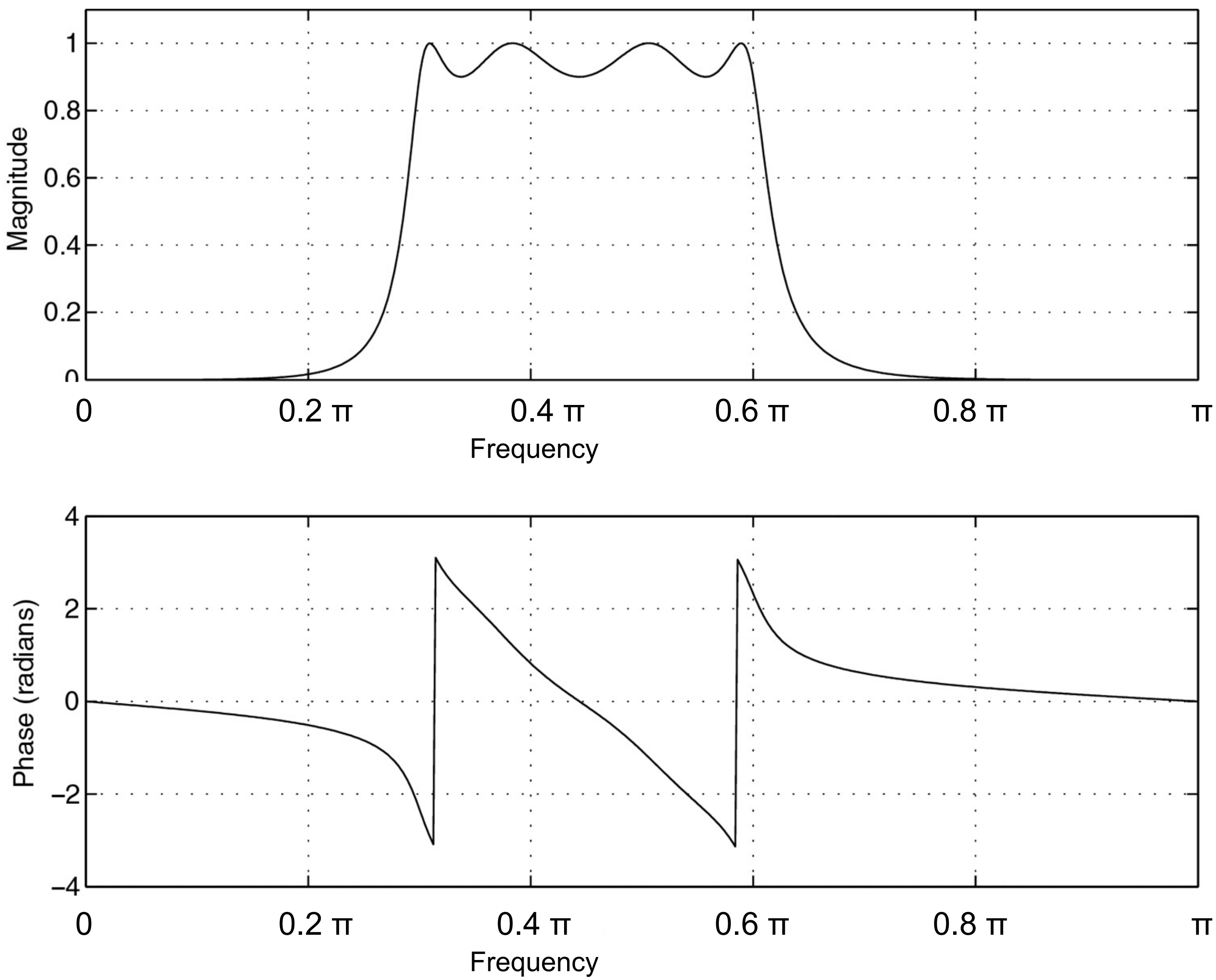




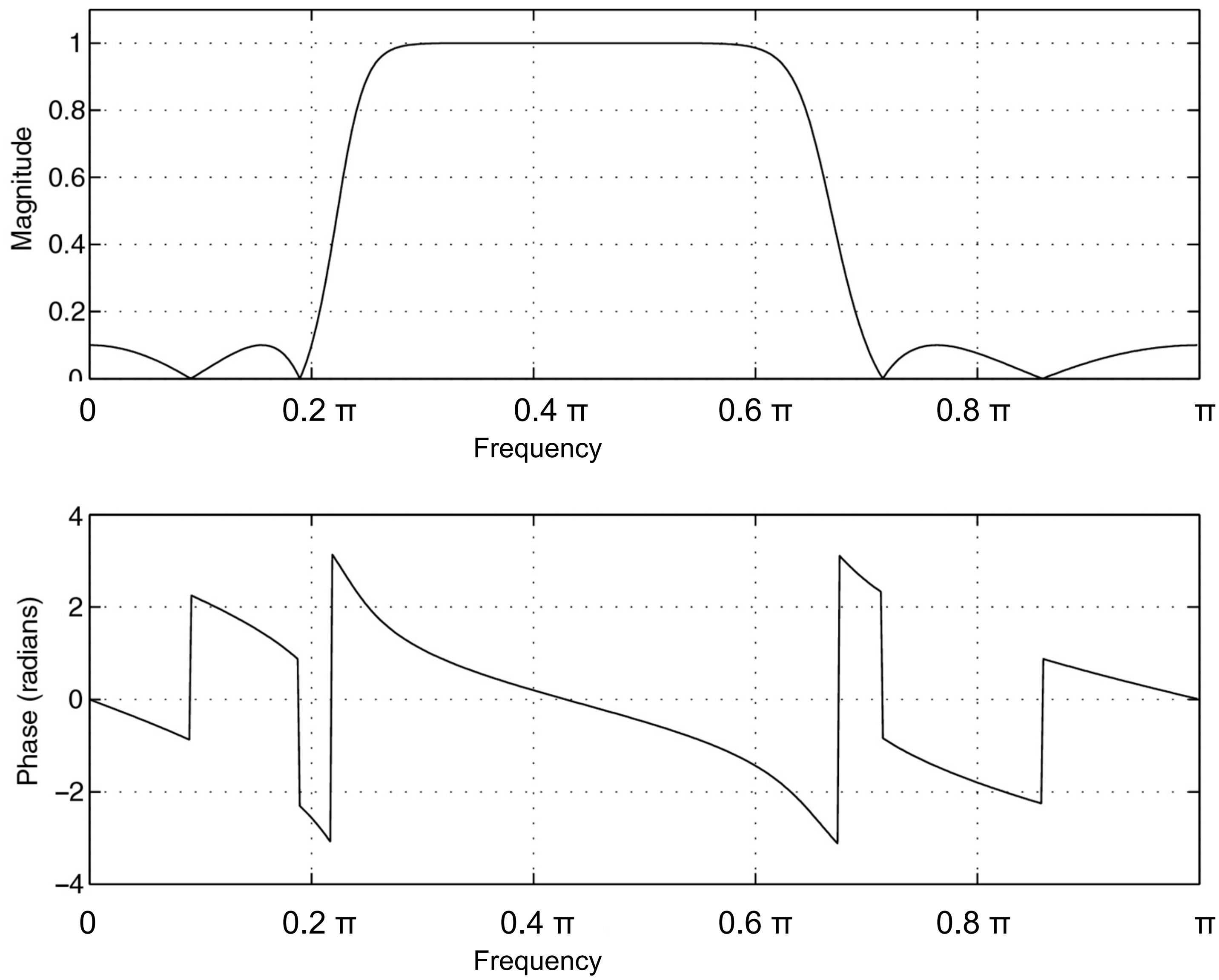
Chebyshev Type I Design



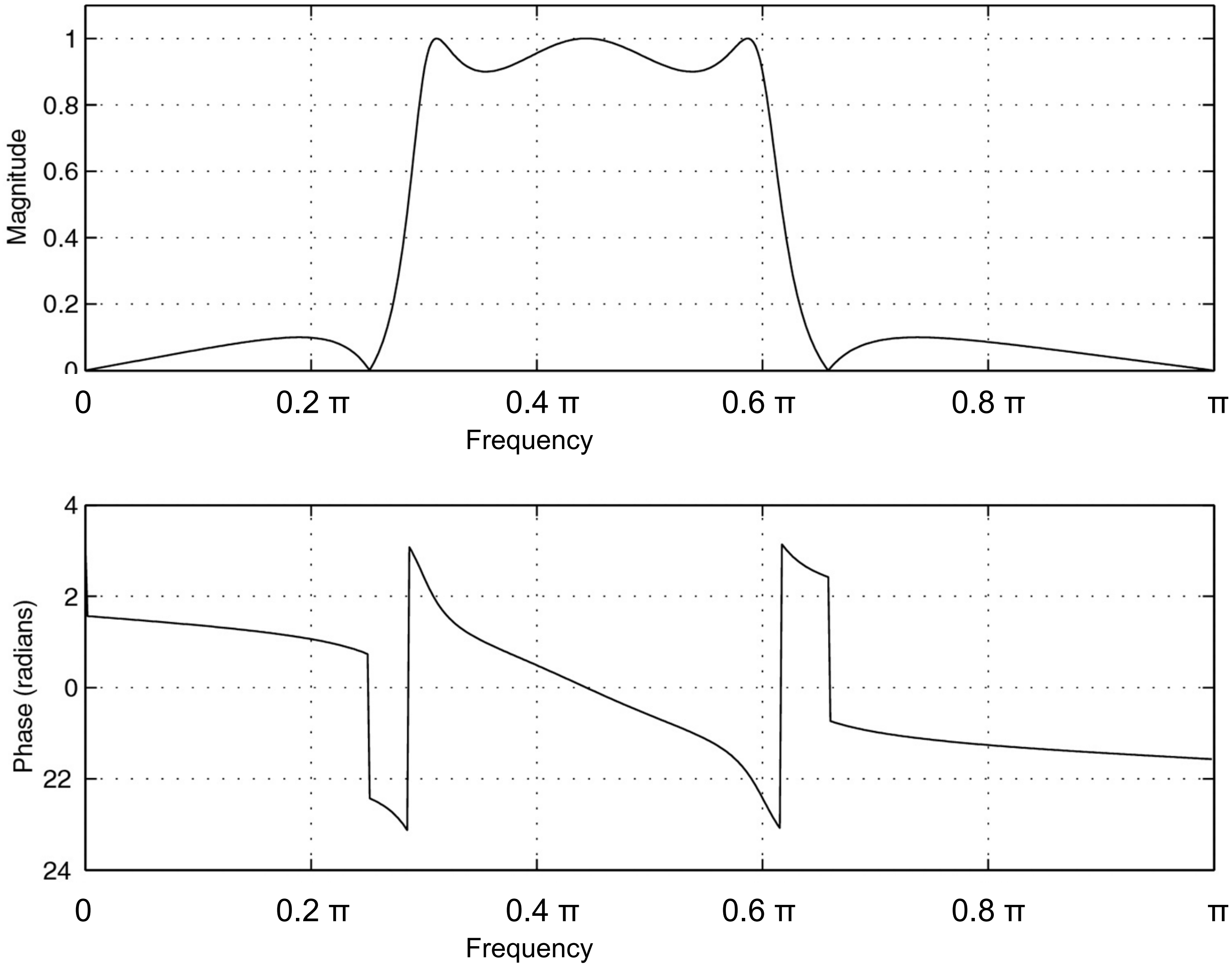
Chebyshev BPF Design



Chebyshev Type II Design



Elliptic BPF Design



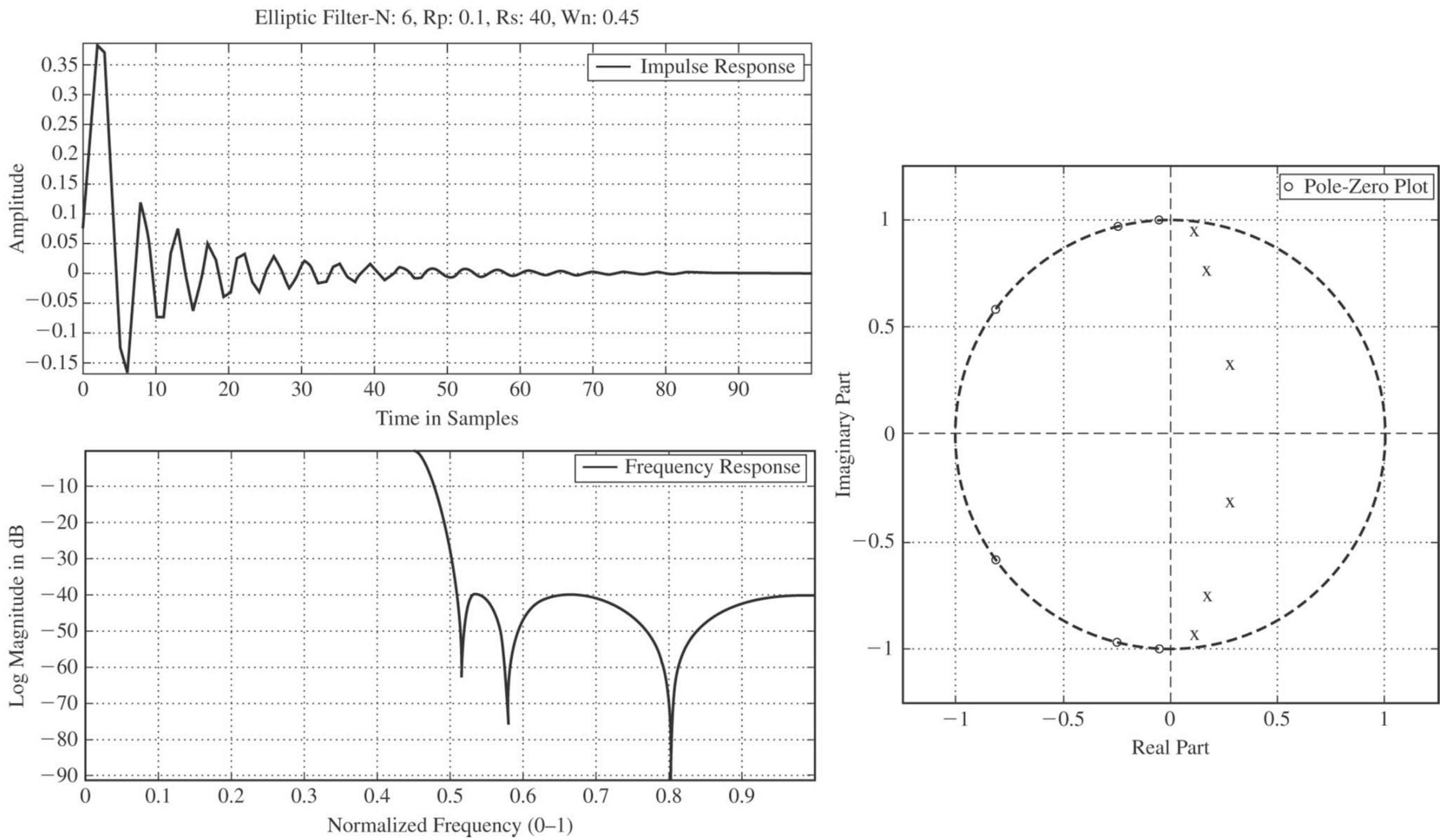
IIR Filters

- IIR filter issues:
  - efficient implementations in terms of computations
  - can approximate any desired magnitude response with **arbitrarily small error**
  - non-linear phase => **time dispersion of waveform**
- IIR design methods
  - 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

Matlab Elliptic Filter Design

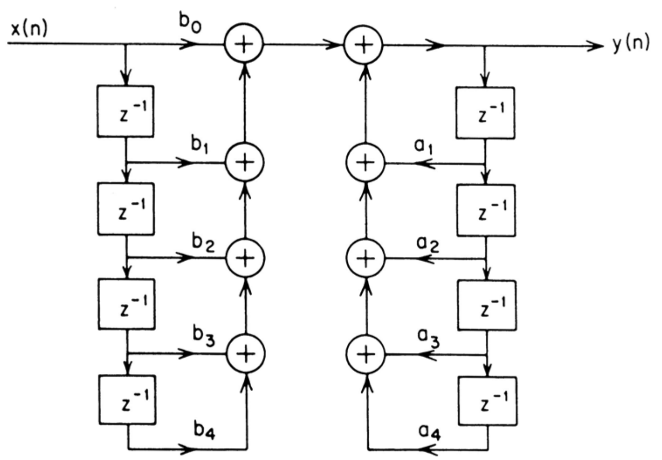
- use **ellip** to design elliptic filter
  - [B,A]=ellip(N,Rp,Rs,Wn)
  - B=numerator polynomial—N+1 coefficients
  - A=denominator polynomial—N+1 coefficients
  - N=order of polynomial for both numerator and denominator
  - Rp=maximum in-band (passband) approximation error (dB)
  - Rs=out-of-band (stopband) 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 Lowpass Filter

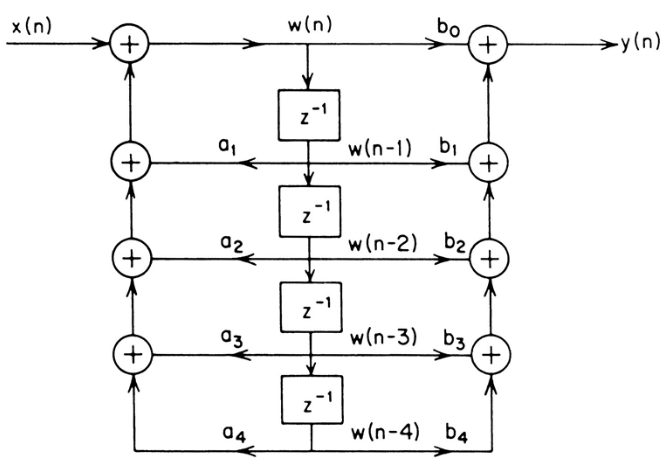


[b,a]=ellip(6,0.1,40,0.45); [h,w]=freqz(b,a,512); x=[1,zeros(1,511)]; y=filter(b,a,x); zplane(b,a);
appropriate plotting commands;

IIR Filter Implementation



$M=N=4$



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

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

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

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

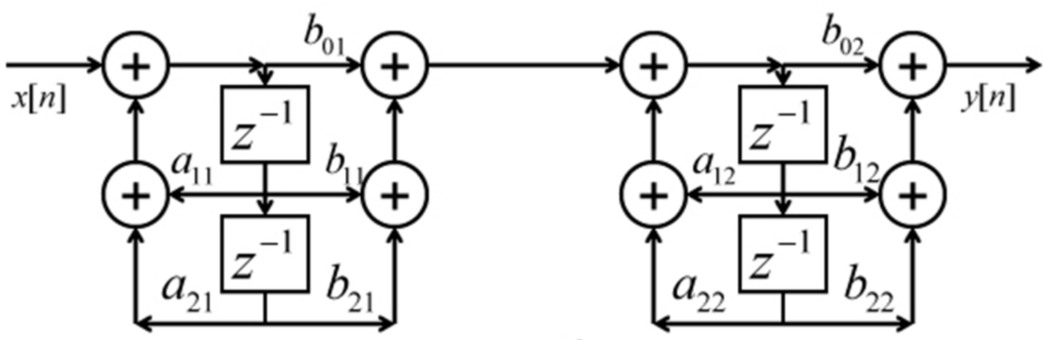
IIR Filter Implementations

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

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

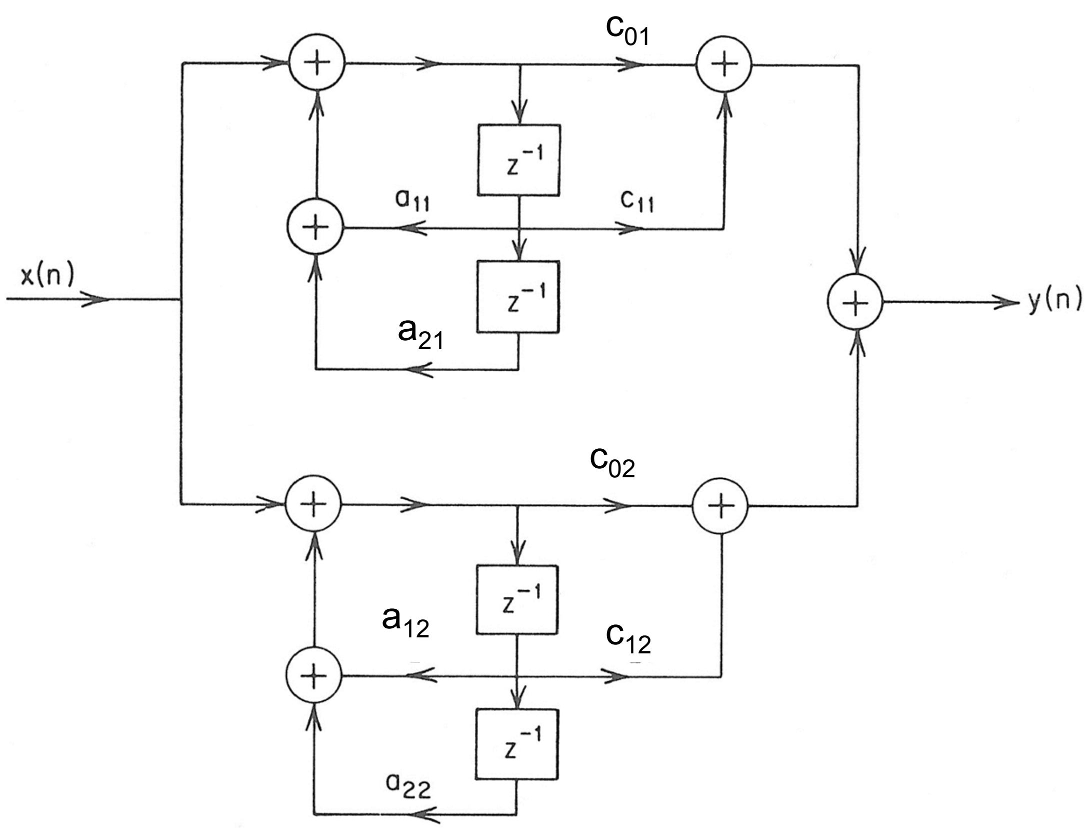
- cascade of second order systems



Used in formant synthesis systems based on ABS methods

IIR Filter Implementations

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

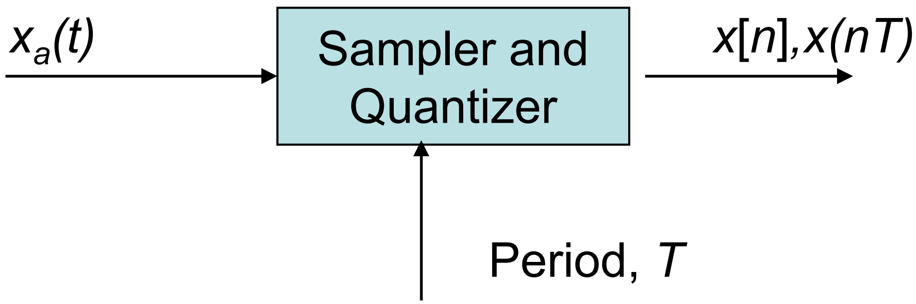


Common form for speech synthesizer implementation

DSP in Speech Processing

- ***filtering*** — speech coding, post filters, pre-filters, noise reduction
- ***spectral analysis*** — vocoding, speech synthesis, speech recognition, speaker recognition, speech enhancement
- ***implementation structures*** — speech synthesis, analysis-synthesis systems, audio encoding/decoding for MP3 and AAC
- ***sampling rate conversion*** — audio, speech
  - DAT — 48 kHz
  - CD — 44.06 kHz
  - Speech — 6, 8, 10, 16 kHz
  - Cellular — TDMA, GSM, CDMA transcoding

Sampling of Waveforms



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

$T = 1/8000$ sec $= 125\,\mu\text{sec}$ for 8kHz sampling rate

$T = 1/10000$ sec $= 100\;\mu\text{sec}$ for 10 kHz sampling rate

$T = 1/16000$ sec $= 67\;\mu\text{sec}$ for 16 kHz sampling rate

$T = 1/20000$ sec $= 50\;\mu\text{sec}$ for 20 kHz sampling rate

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

$1/T \geq 2\,F_N$ $(F_S \geq 2F_N)$ (A-D or C/D converter)

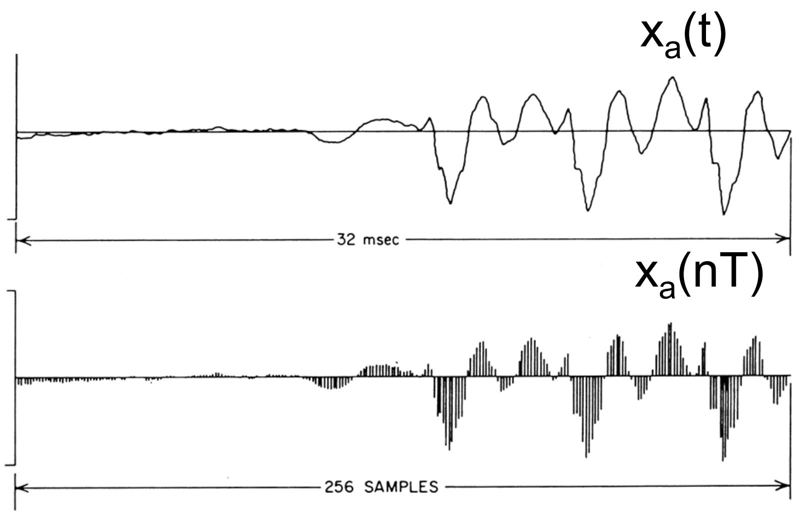
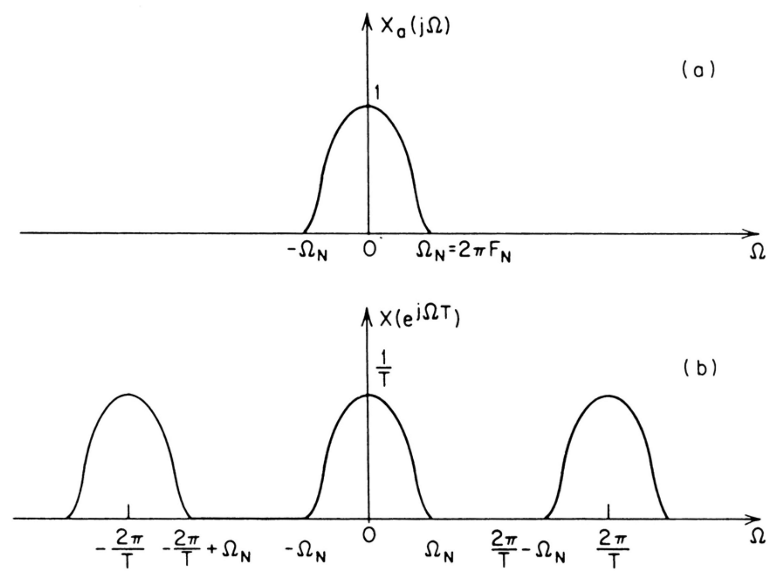


Fig. 2.1 Representations of a speech signal.



$x_a(nT) = x_a(t)\, u_T(nT)$, where $u_T(nT)$ is a periodic pulse train of period T, with periodic spectrum of period $2\pi/T$

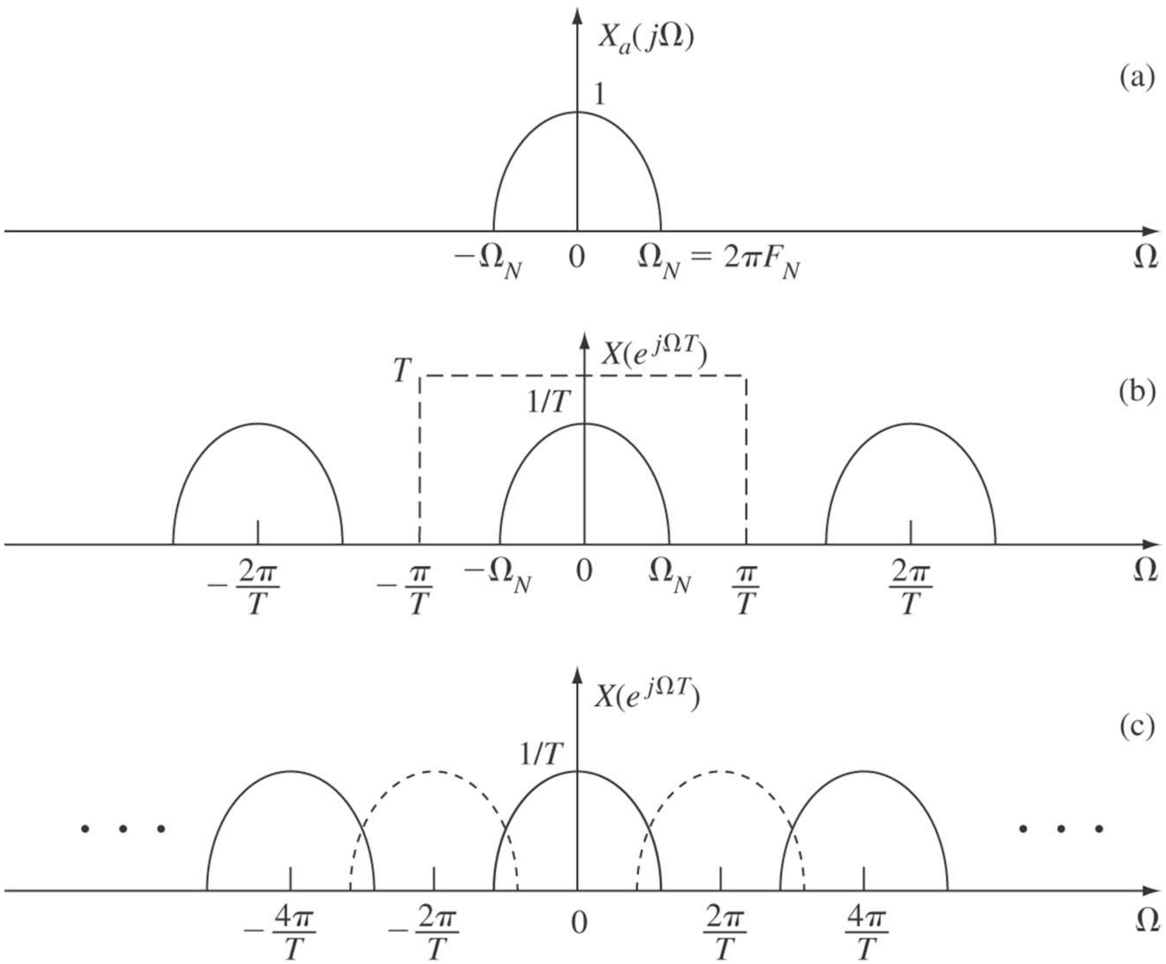
Sampling Theorem Equations

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

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

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

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 at least twice the Nyquist frequency to prevent aliasing (frequency overlap)
  - telephone speech (300-3200 Hz) => F_S=6400 Hz
  - wideband speech (100-7200 Hz) => F_S=14400 Hz
  - audio signal (50-21000 Hz) => F_S=42000 Hz
  - AM broadcast (100-7500 Hz) => 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.,

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

- 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.,

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

bandlimited sample interpolation—perfect at every sample point, perfect in-between samples via interpolation

- digital-to-analog converter

Decimation and Interpolation of Sampled Waveforms

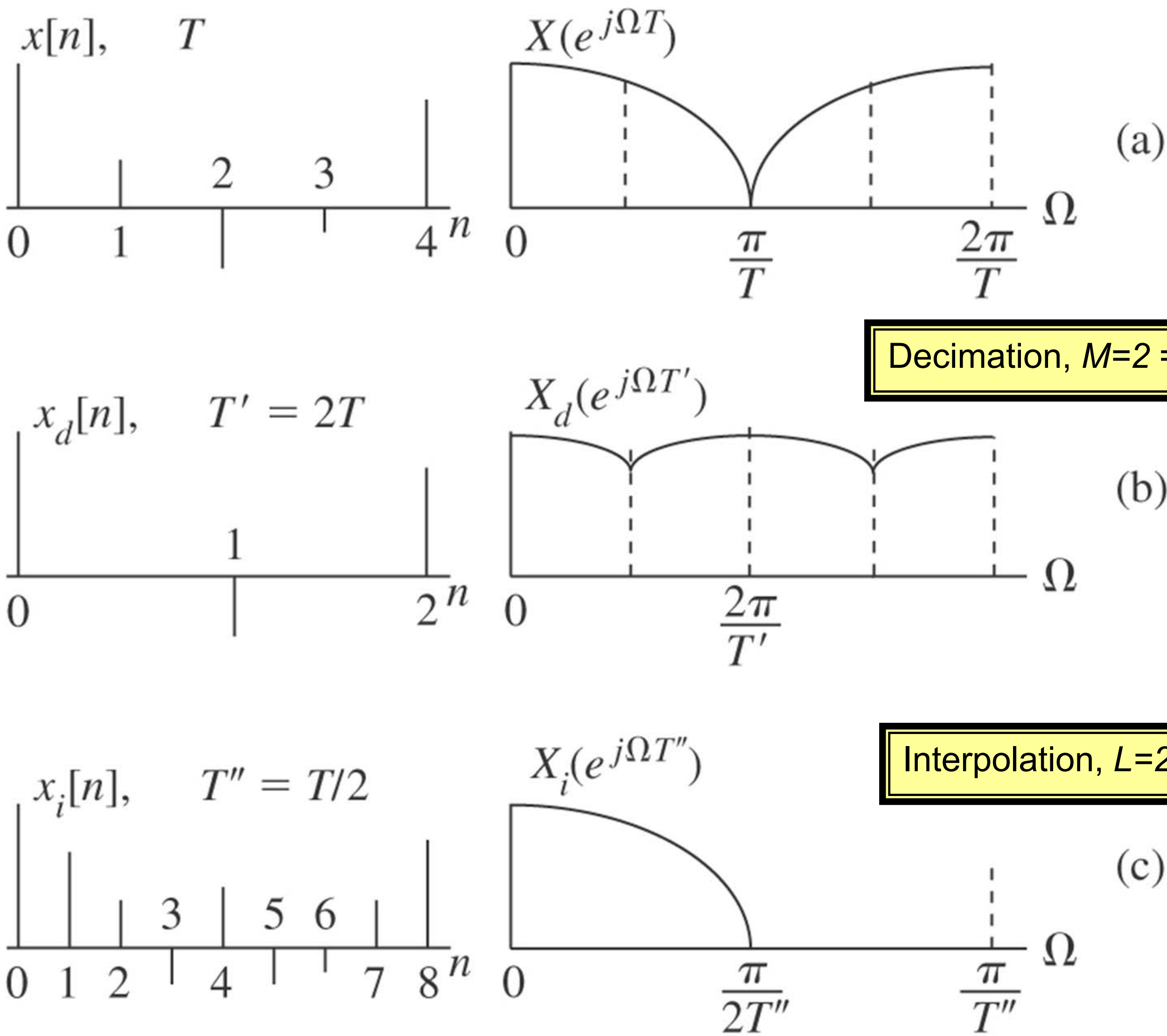
- CD rate (44.06 kHz) to DAT rate (48 kHz)—media conversion
- Wideband (16 kHz) to narrowband speech rates (8kHz, 6.67 kHz)—storage
- oversampled to correctly sampled rates--coding

$$x[n] = x_a(nT), \quad X_a(j\Omega) = 0 \text{ for } |\Omega| > 2\pi F_N$$

if $1/T > 2F_N$ (adequate sampling) then

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

Decimation and Interpolation



Decimation, $M=2 \Rightarrow T'=2T$

Interpolation, $L=2 \Rightarrow T'=T/2$

Decimation

 Standard Sampling: begin with digitized signal:

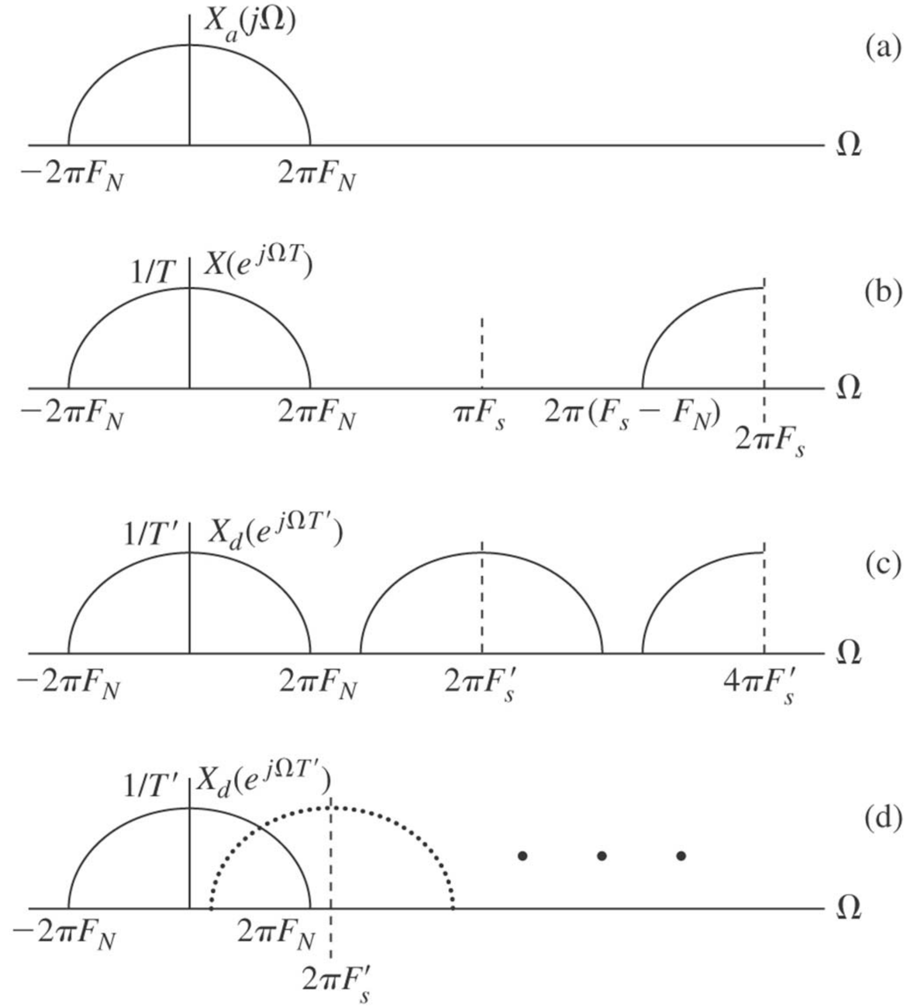
$$x[n] = x_a(nT) \leftrightarrow X_a(j\Omega) = 0, \quad |\Omega| \geq 2\pi F_N \quad (a)$$

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

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

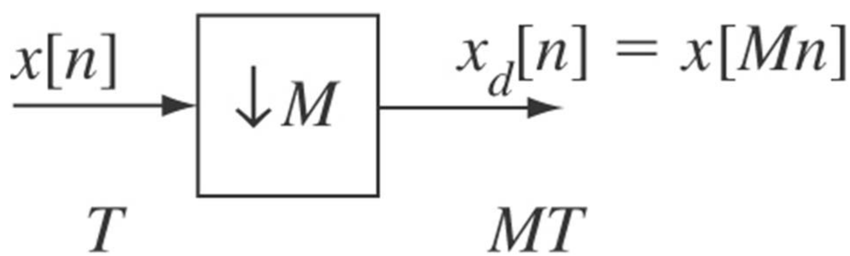
$$X(e^{j\Omega T}) = 0, \; 2\pi F_N \leq |\Omega| \leq 2\pi(F_s - F_N)$$

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

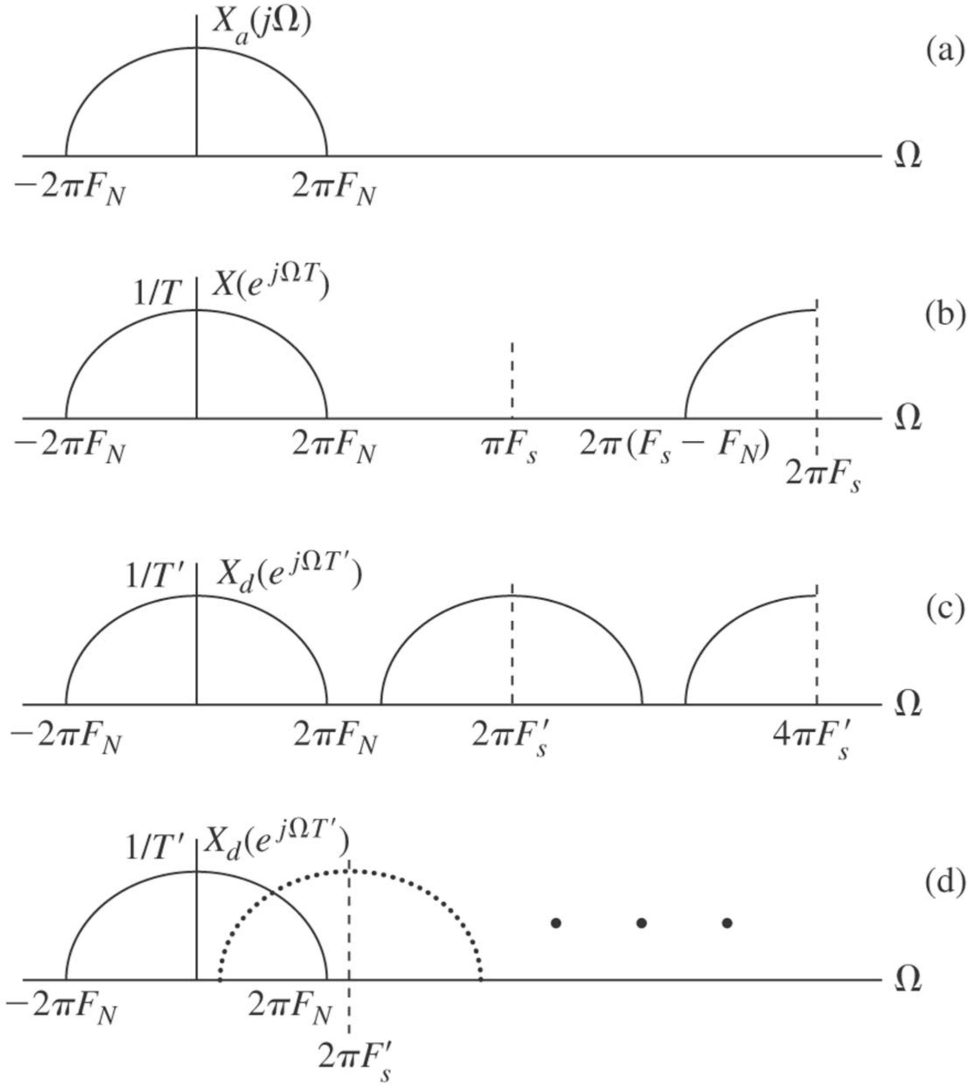


Decimation

- want to reduce sampling rate of sampled signal by factor of $M \geq 2$
- want 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

$$F_s' \geq 2F_N$$

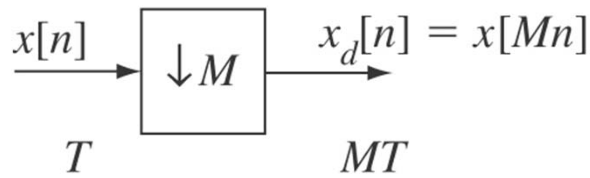
to avoid aliasing for $M = 2$ $\quad (c)$

- when

$$F_s' < 2F_N$$

we get aliasing for $M = 2$ $\quad (d)$

Decimation



- DTFTs of $x[n]$ and $x_d[n]$ related by aliasing relationship:

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(\omega-2\pi k)/M})$$

- or equivalently (in terms of analog frequency):

$$X_d(e^{j\Omega T'}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(\Omega T'-2\pi k)/M})$$

- assuming $F_s' = \frac{1}{MT} \geq 2F_N$, (i.e., no aliasing) we get:

$$X_d(e^{j\Omega T'}) = \frac{1}{M} X(e^{j\Omega T'/M}) = \frac{1}{M} X(e^{j\Omega T}) = \frac{1}{M}\frac{1}{T} X_a(j\Omega)$$

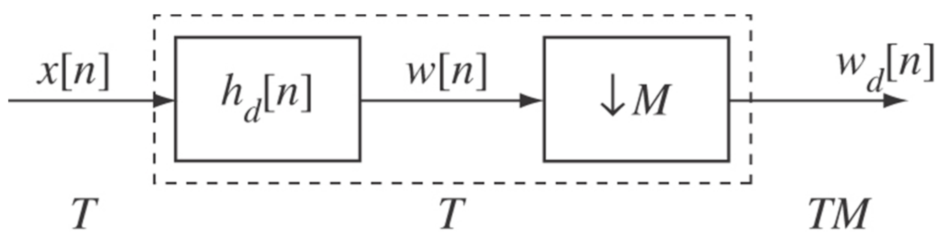
$$= \frac{1}{T'} X_a(j\Omega), \qquad -\frac{\pi}{T'} < \Omega < \frac{\pi}{T'}$$

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 down-sample the reuslting lowpass-filtered signal by a factor of M without aliasing

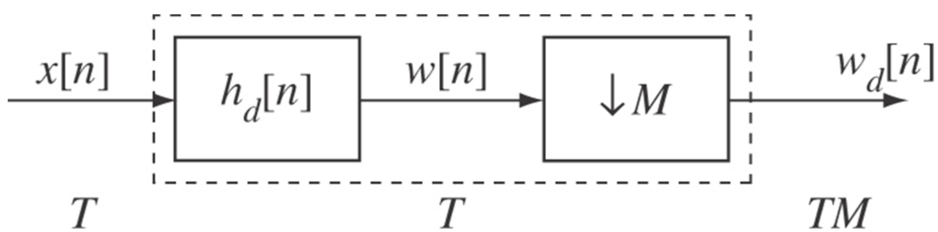


Decimation

- using a lowpass filter gives:

$$W_d(e^{j\Omega T'}) = \frac{1}{T'} H_d(e^{j\Omega T}) X_a(j\Omega), \quad -\frac{\pi}{T'} < \Omega < \frac{\pi}{T'}$$

- if filter is used, the down-sampled signal, $w_d[n]$, no longer represents the original analog signal, $x_a(t)$, but instead the lowpass filtered version of $x_a(t)$
- the combined operations of lowpass filtering and downsampling are called *decimation*.



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

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

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

Interpolation

- implementing the previous equation by filtering the upsampled sequence

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

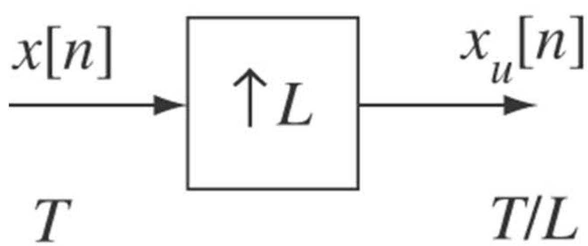
- $x_u[n]$ has the correct samples for $n = 0, \pm L, \pm 2L, \ldots,$ but it has zero-valued samples in between (from the upsampling operation)
- The Fourier transform of $x_u[n]$ is simply:

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

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

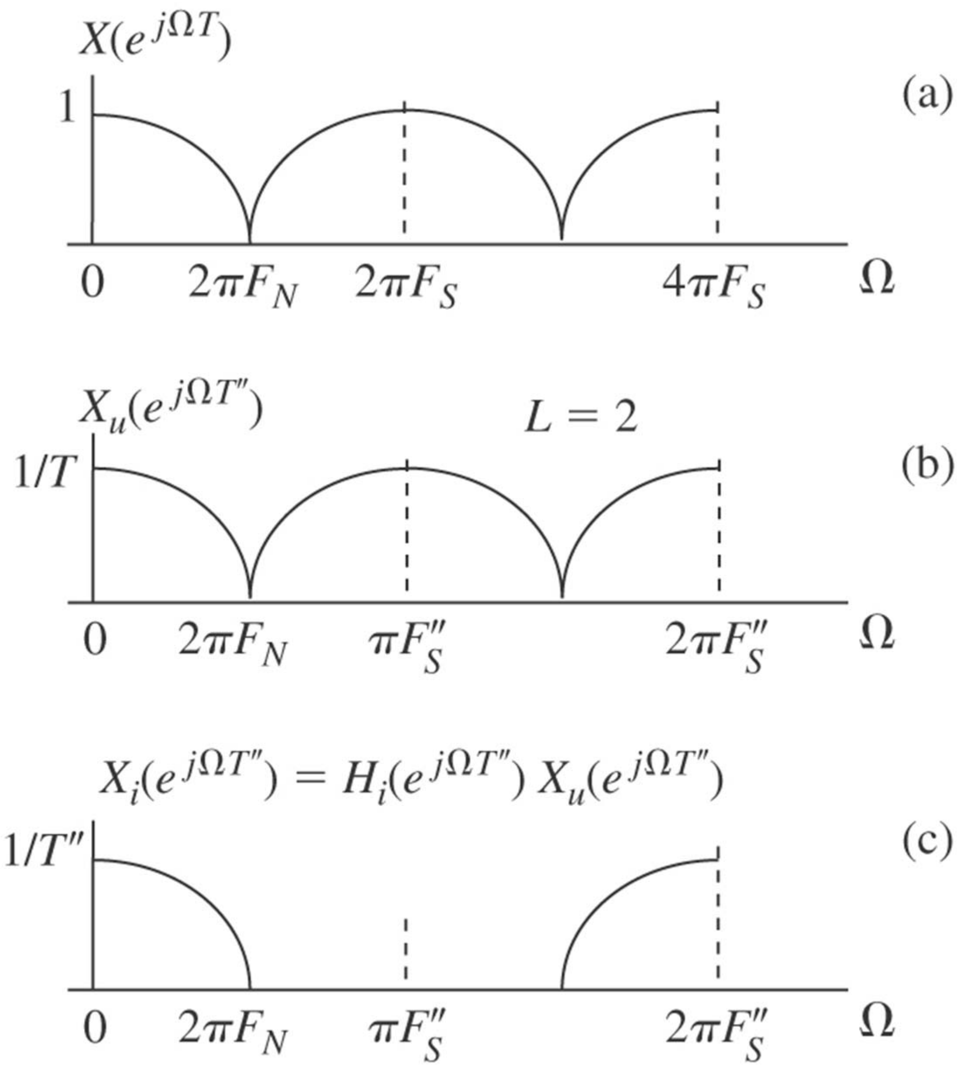
- Thus $X_u(e^{j\Omega T''})$ is periodic with two periods, namely

with period $2\pi / L$, due to upsampling) and 2π due to being a digital signal



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

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:

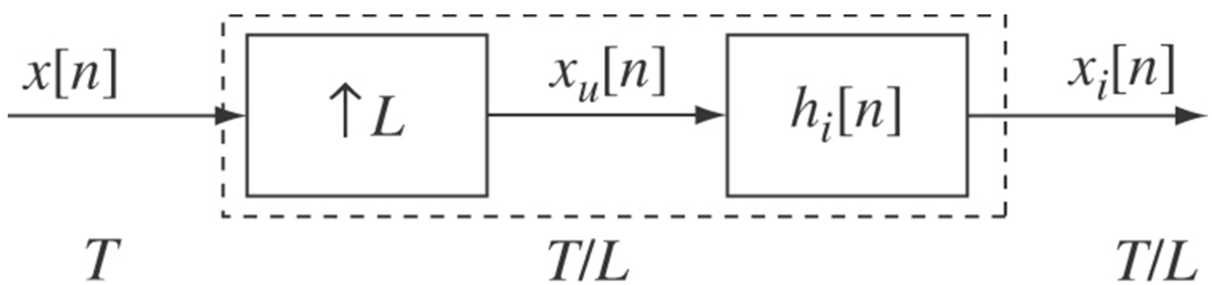
$$X_i(e^{j\omega}) = \begin{cases} (1/T'')\,X(e^{j\omega L}) & 0 \le |\omega| < \pi/L \\ 0 & \pi/L \le |\omega| \le \pi \end{cases}$$

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

Interpolation

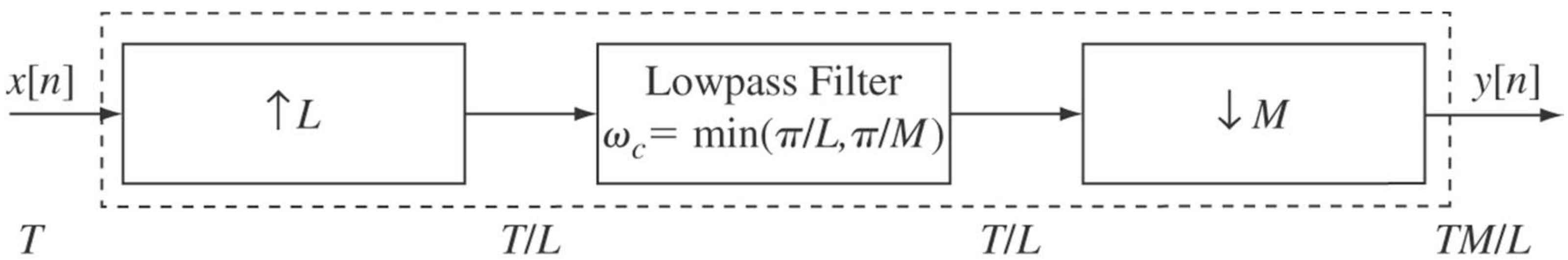
- Original signal, $x[n]$, at sampling period, T, is first upsampled to give signal $x_u[n]$ with sampling period $T'' = T / L$
- lowpass filter removes images of original spectrum giving:

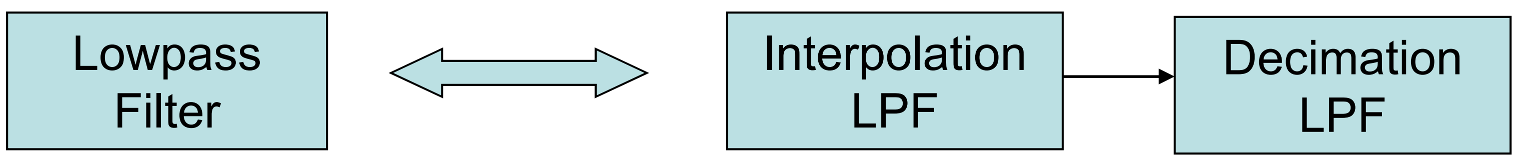
$$x_i[n] = x_a(nT'') = x_a(nT / L)$$



SR Conversion by Non-Integer Factors

- $T' = MT/L$ => 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 of DSP-Part II

- 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