520.445/645 – Audio Signal Processing
☞ Topic 3 ☜

Review of Digital Signal Processing

1

1

What is Digital Signal Processing (DSP)?

Input signal → Analog-to-digital converter → Computer → Digital-to-analog converter → Output signal

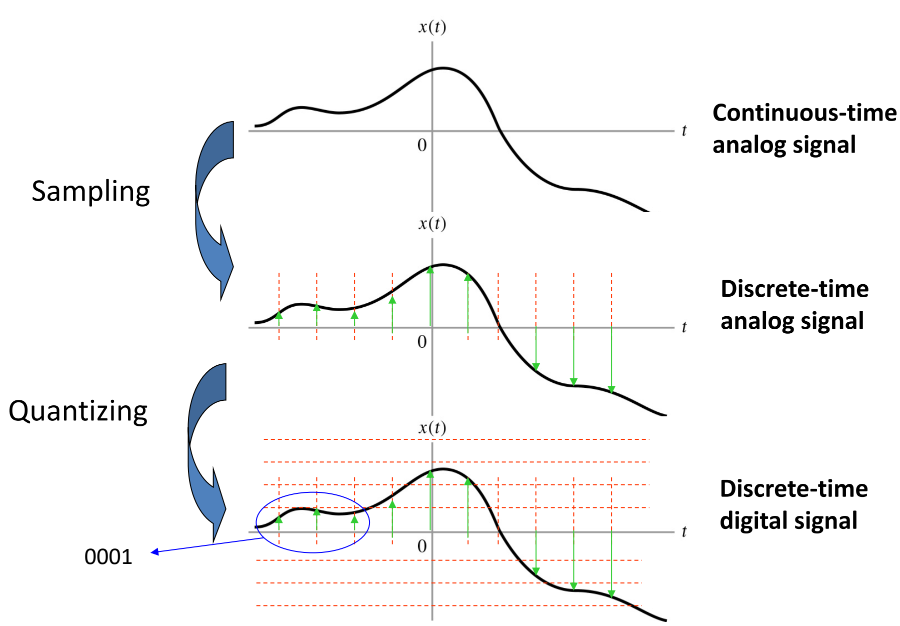
- **Digital**
  - Method to represent a quantity, a phenomenon or an event
- **Signal**
  - A signal is a function of independent variable(s) that carry information
  - It is a detectable physical quantity (voltage, current, magnetic field, sound, photo, gesture) In this class, interested in sound and speech
- **Processing**
  - Transformation, analysis
  - Filtering, recognition, synthesis, coding, detection, estimation

2

2

Digital processing of analog signals

- **A-to-D conversion:** bandwidth control, sampling and quantization
- **Computational processing:** implemented on computers or customized ICs with finite-precision arithmetic
  - **basic numerical processing:** add, subtract, multiply (scaling, amplification, attenuation), mute, …
  - **algorithmic numerical processing:** convolution or linear filtering, non-linear filtering, difference equations, DFT, …
- **D-to-A conversion:** re-quantification and filtering (or interpolation) for reconstruction



CONTINUOUS-TIME

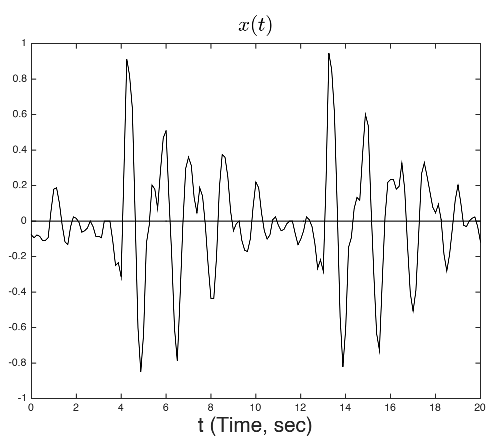
5

5

Continuous-time signal

A quantity of interest that changes over time:

- ○Quantity of interest $x(.)$ => sound pressure, $x(.) \in \mathbb{R}$
- ○Depends on independent variable t => time, $t \in \mathbb{R}$



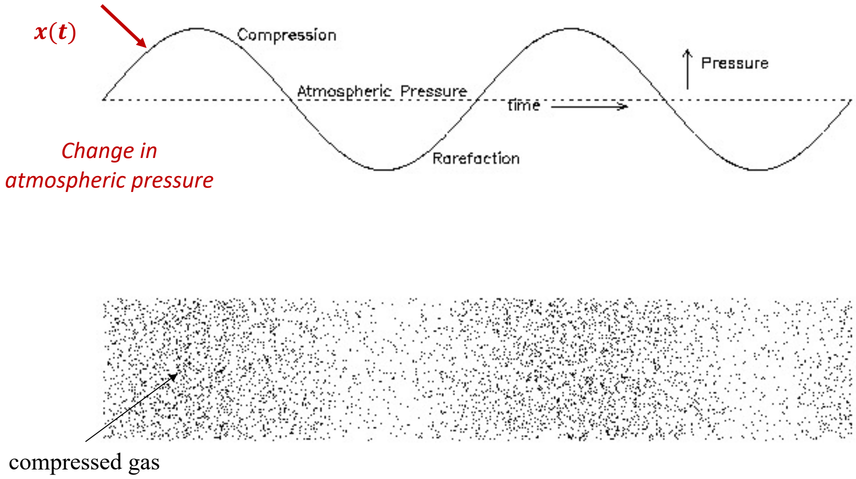
6

6

Sound as a signal

Time signal

$x(t)$



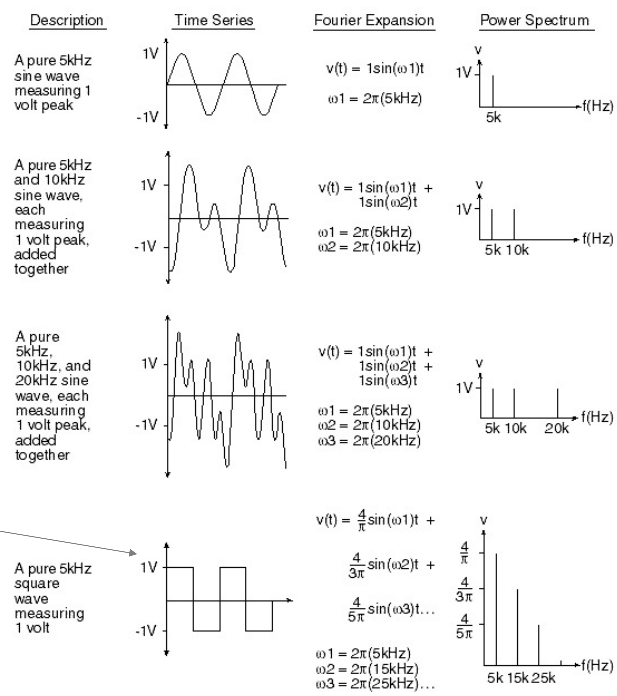
Change in atmospheric pressure

7

7

Signal decomposition

Sound is converted into an electrical signal



8

Fourier series

- Each <u>periodic</u> signal can be expressed as linear combination of weighted exponentials at integer multiples of the fundamental:

$$x_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{j\frac{2\pi kt}{T_0}}$$

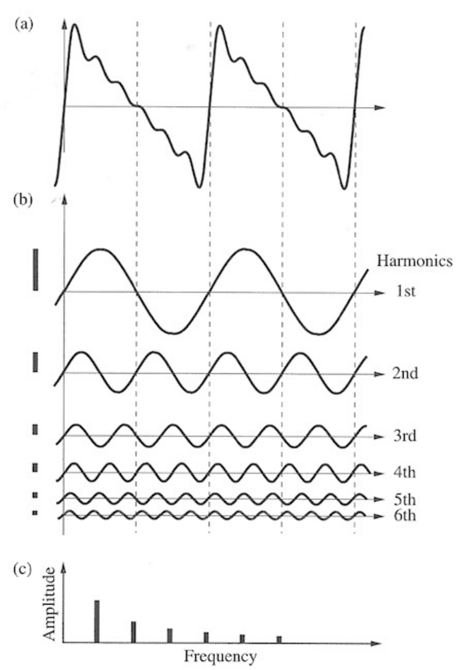
where T is the signal period and c_k are the Fourier coefficients. They are defined as:

$$c_k = \int_0^{T_0} x_p(t) e^{-j\frac{2\pi kt}{T}} dt$$

9

9

Fourier series example



10

Fourier transform

- The principle of expressing time signals as sums of weighted sinusoids can be extended to aperiodic signals.
  - [Note: practically, we are talking about finite duration signals, so they cannot be periodic]
- The Fourier transform of a general signal is:

$$\boxed{X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt}$$

- The Fourier transform is an invertible operation

$$\boxed{x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df}$$

- No information is lost because this is a 1:1 reversible mapping

11

11

Fourier transform

$$\boxed{X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt}$$

- The variable f has units of Hertz (i.e. cycles per second).
- Equivalently, we can define a new variable

$\Omega = 2\pi f$ in units of radians/second.

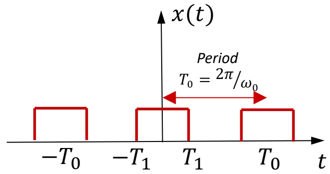
- So, we can write:

$$\boxed{X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt}$$

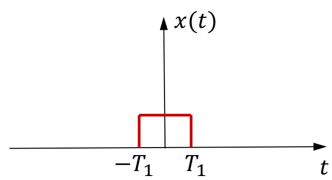
12

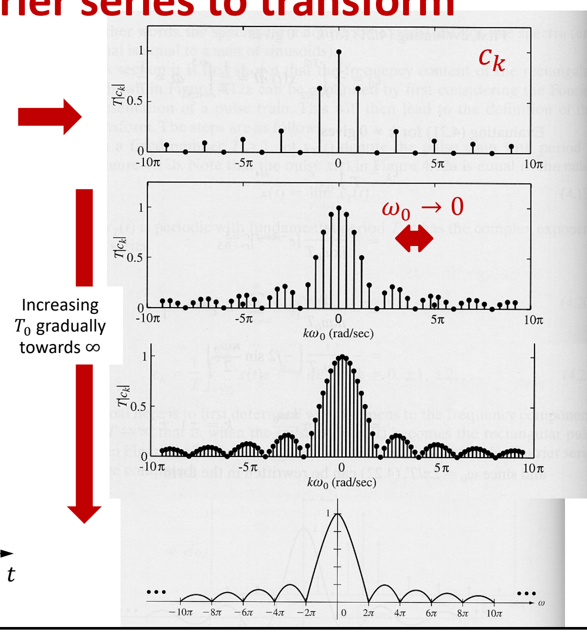
12

From Fourier series to transform



Let $T_0 \to \infty$





13

Important

	Time-Domain	Frequency-Domain
Continuous-time Fourier Series CTFS	Continuous Periodic	Discrete Aperiodic
Continuous-time Fourier Transform CTFT	Continuous Aperiodic	Continuous Aperiodic

14

14

DISCRETE-TIME

15

15

Discrete-time signals

- A sequence of numbers
- Mathematical representation:
    - $X = \{\mathrm{x[n]}\}, \quad -\infty < n < \infty$
  - Sampled from an analog signal $x_a(t)$, at times nT:
    - $x[n] = x_a(nT), \quad -\infty < n < \infty$
- T is called the sampling period, and its reciprocal $F_s = {}^1/_T$ the sampling frequency

16

16

Sampling rate

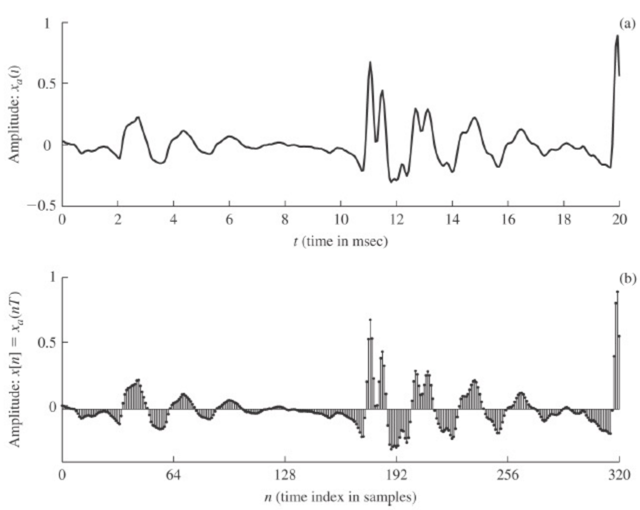
- Changing the sampling rate controls the sampling period (how often a sample is drawn from the signal)

$$F_S = 8000 \text{ Hz} \quad \leftrightarrow \quad T = 1/8000 = 125\ \mu\text{sec}$$
$$F_S = 10000 \text{ Hz} \quad \leftrightarrow \quad T = 1/10000 = 100\ \mu\text{sec}$$
$$F_S = 16000 \text{ Hz} \quad \leftrightarrow \quad T = 1/16000 = 62.5\ \mu\text{sec}$$
$$F_S = 20000 \text{Hz} \quad \leftrightarrow \quad T = 1/20000 = 50\ \mu\text{sec}$$

17

17

Discrete-time signals



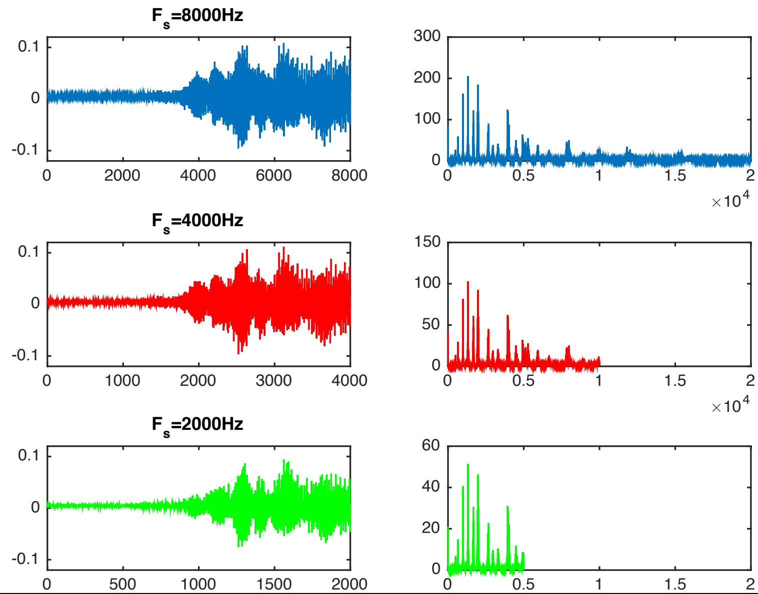
plot( );

stem( );

18

18

Changing the sampling rate

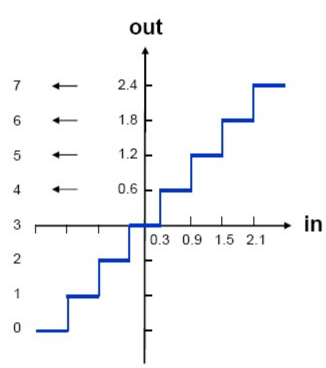


19

19

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.



A 3-bit uniform quantizer

Quantization:

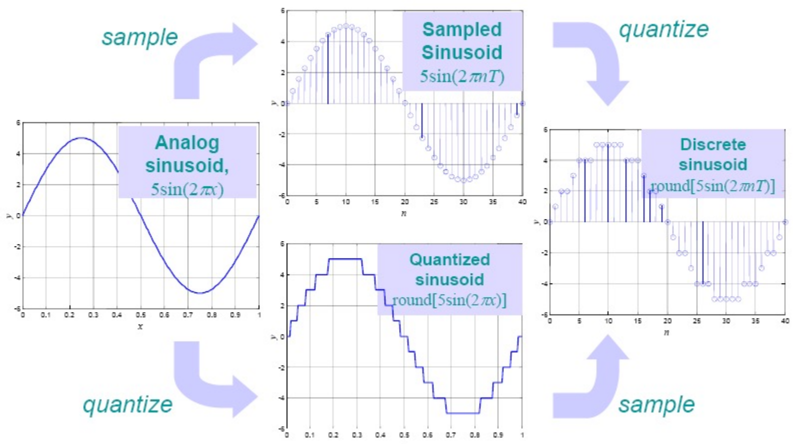
- Transforming a continuously-valued input into a representation that assumes one of a finite set of values
- The finite set of output values is indexed; e.g., the value 1.8 has an index of 6 (or 110) in binary representation
- Storage or transmission uses binary representation; a quantization table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

20

20

Discrete signals

- Confusion: discrete-time vs. discrete signals!

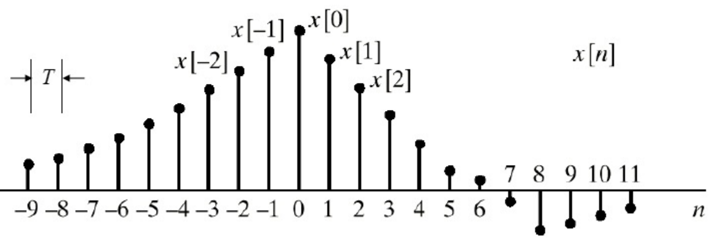


Issues with discrete signals

- What sampling rate is appropriate?
  - 6.4 kHz (telephone bandwidth)
  - 8 kHz (extended telephone BW)
  - 10 kHz (extended bandwidth)
  - 16 kHz (hi-fidelity speech)
  - 44.1 KHz (CD quality music)
- How many quantization levels are necessary at each bit rate (bits/sample)
  - 16, 12, 8.. => ultimately determines the S/N ratio of the signal
  - Speech coding/compression is concerned with answering this question in an optimal manner

Discrete-time signals

- Discrete-time signals are sequences



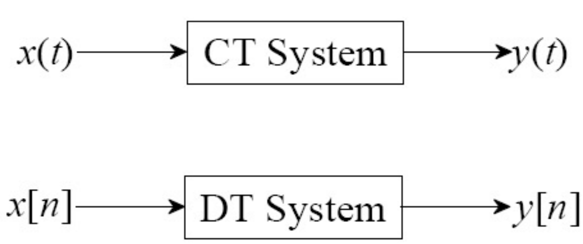
- $x[n]$ denotes the “sequence value at ‘time’ n”

23

23

Signal processing

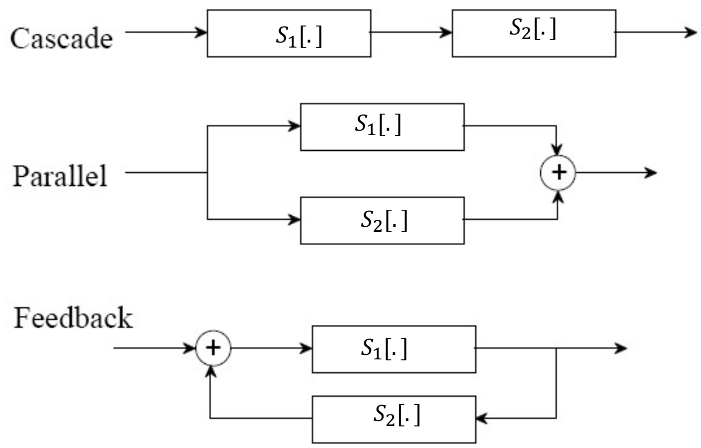
- Transform digital signal into more desirable form
- A system is an entity that manipulates one or more signals to accomplish a function, thereby yielding new signals.



24

24

System interconnections



25

25

System properties

- System properties apply to both continuous-time and discrete-time systems
  - Causality
  - Linearity
  - Time invariance
  - Invertibility

26

26

System properties

- Causality
  - A system is said to be causal if the present value of the output signal depends only on the present or past values of the input signal.

27

27

System properties

- Stability
  - A system is said to be BIBO stable (i.e. **B**ounded-**I**nput, **B**ounded-**O**utput stable) if a bounded-input yields a bounded output.

28

28

System properties

- Linearity
  – A system is said to be linear in terms of the system input x(t) and the system output y(t) if it satisfies the following two properties of superposition and homogeneity.

Superposition :

$$x_1(t) \rightarrow \blacksquare \rightarrow y_1(t) \qquad x_2(t) \rightarrow \blacksquare \rightarrow y_2(t)$$

$$\Rightarrow \quad x_1(t)+x_2(t) \rightarrow \blacksquare \rightarrow y_1(t)+y_2(t)$$

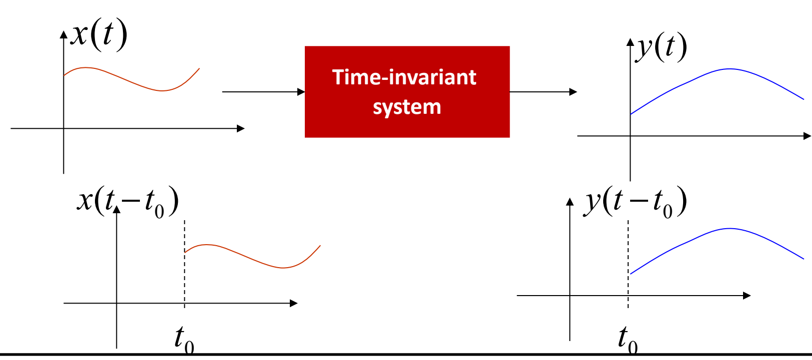
Homogeneity :

$$x_1(t) \rightarrow \blacksquare \rightarrow y_1(t) \quad \Rightarrow \quad ax_1(t) \rightarrow \blacksquare \rightarrow ay_1(t)$$

29

System properties

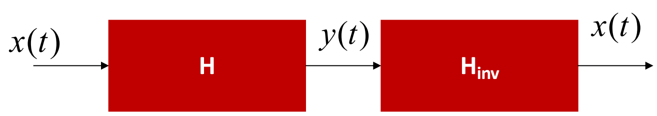
- Time invariance
  – A system is said to be time invariant if a time delay or time advance of the input signal leads to an identical time shift in the output signal.



30

System properties

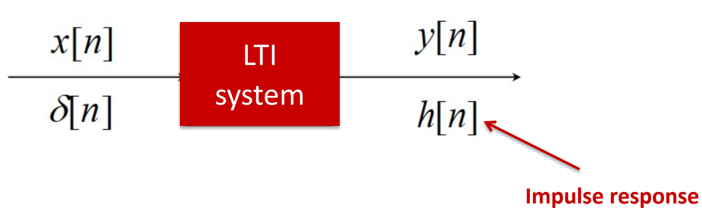
- Invertibility
  - A system is said to be Invertible if the input of the system can be recovered from the output.



LTI systems

- A system that is <u>BOTH</u> linear and time-invariant is said to be LTI.
- The output can be determined via a convolution sum

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



LTI systems

- Easiest to understand/manipulate
- Powerful processing capabilities
- Characterized completely by their response to unit sample, h[n], via convolution relationship
- Basis for linear filtering
- Used as models for speech production (source convolved with system)

33

33

Eigen property of LTI systems

- Consider the complex exponential: $x[n] = e^{j\omega_0 n}$
- The output of a stable LTI system is

$$
\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] \\
&= \sum_{k=-\infty}^{\infty} h[k]e^{j\omega_0 (n-k)} \\
&= \left(\sum_{k=-\infty}^{\infty} h[k]e^{-j\omega_0 k}\right) e^{j\omega_0 n} \\
&= H(\omega_0)e^{j\omega_0 n}
\end{aligned}
$$

  - i.e A complex exponential input to an LTI system results in the same complex exponential at the output scaled by $H(\omega_0)$.
  - Therefore, the complex exponentials are called eigenfunctions of LTI systems, and $H(\omega_0)$ is the associated eigenvalue.

34

34

Transform representations

- Z-transform
- Discrete-time Fourier transform
- Discrete Fourier transform

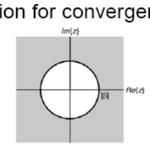
Z-transform

- The Z-transform is a general mapping from time-domain to another domain, defined by: $$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$
  - X(z) is an infinite power (geometric) series in the complex variable, $z = re^{j\omega}$
    - so the sum does not always converge.

- ***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: $R_1 < |z| < R_2$**



Rational Functions

- We are often interested in a special class of functions of the form:

$$X(z) = \frac{P(z)}{Q(z)}$$

  - The roots of the Polynomial P(z) are called *zeros* (values of z for which P(z)=0).
  - The roots of the Polynomial Q(z) are called *poles* (values of z for which Q(z)=0).
    - ✓The zeros of X(z) are values for which X(z)=0, and poles are values for which X(z) goes to infinity.

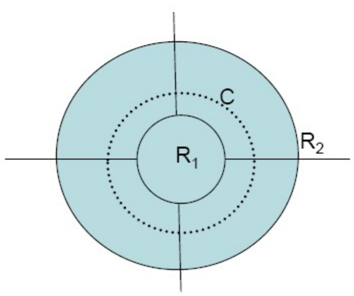
37

37

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

38

38

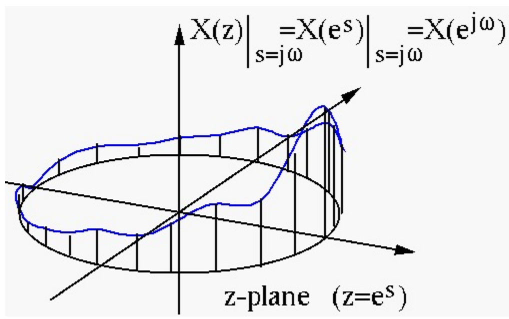
Discrete-Time Fourier Transform (DTFT)

- If the ROC includes the unit circle, then

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

Abuse of notation/ Used interchangeably

  – is called the discrete-time Fourier transform (DTFT) of x[n]



41

41

Discrete-Time Fourier Transform (DTFT)

- If the ROC includes the unit circle, then

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

Abuse of notation/ Used interchangeably

  – is called the discrete-time Fourier transform (DTFT) of x[n]

- The 'synthesis equation' is defined as:

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

$X(\omega)$: periodic with period 2π

- The pair of equation is known as the discrete-time Fourier transform pair representation of a sequence.

42

42

Discrete-Time Fourier Transform (DTFT)

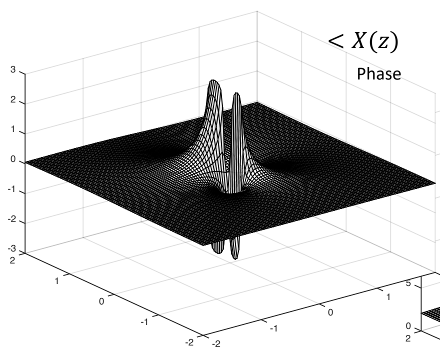
- Note that $X(\omega)$ is an infinite sum
  - It may not converge
  - We need $|X(\omega)| < \infty \quad \forall \omega$
  - $|X(\omega)| = \left|\sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega}\right| \le \sum_{n=-\infty}^{\infty} |x[n]||e^{-jn\omega}| \le \sum_{n=-\infty}^{\infty} |x[n]| < \infty$
  - This inequality $\sum_{n=-\infty}^{\infty} |x[n]| < \infty$ is called absolute summability
  - It is sufficient condition for existence of DTFT

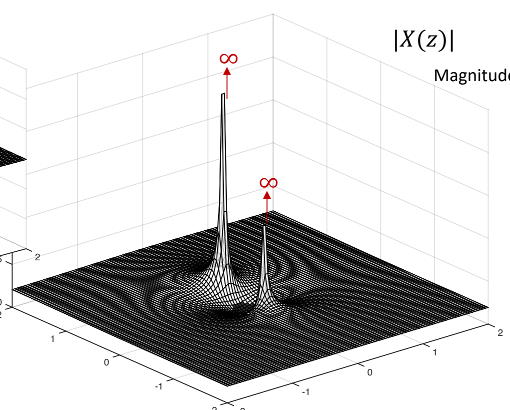
43

43

From Z-transform to Fourier transform

$$\left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n] \quad \xrightarrow{z} \quad \frac{1}{1-\frac{1}{2}z^{-1}} + \frac{1}{1+\frac{1}{3}z^{-1}} \quad , |z| > \frac{1}{2}$$



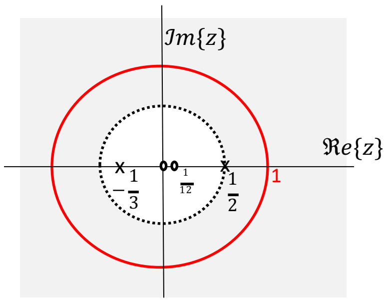


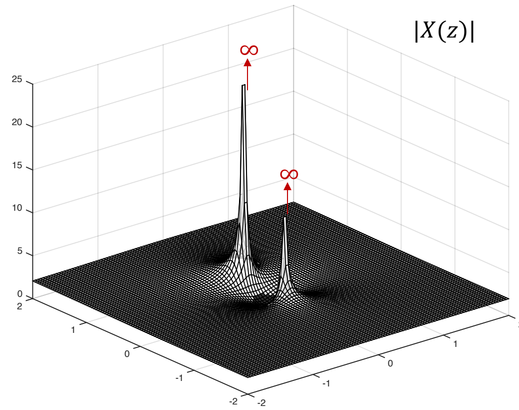
44

44

From Z-transform to Fourier transform

$$\left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n] \quad \xrightarrow{z} \quad \frac{1}{1-\frac{1}{2}z^{-1}} + \frac{1}{1+\frac{1}{3}z^{-1}} \quad , |z| > \frac{1}{2}$$

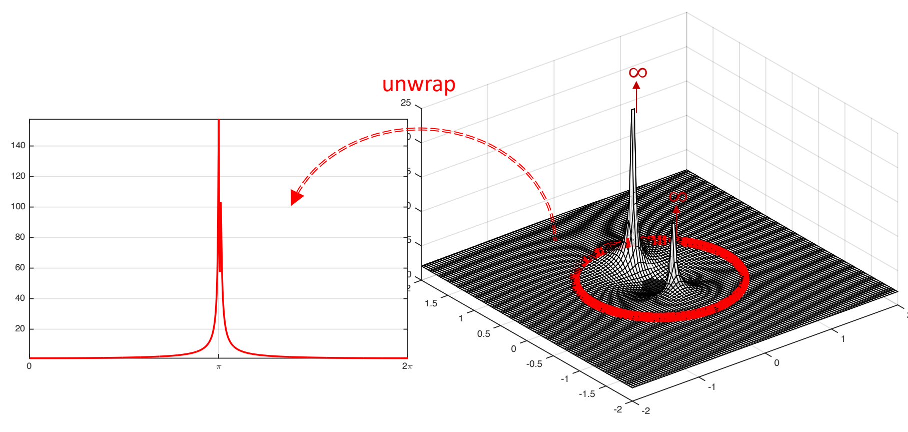




45

From Z-transform to Fourier transform

$$\left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n] \quad \xrightarrow{F} \quad \frac{1}{1-\frac{1}{2}e^{-j\omega}} + \frac{1}{1+\frac{1}{3}e^{-j\omega}}$$

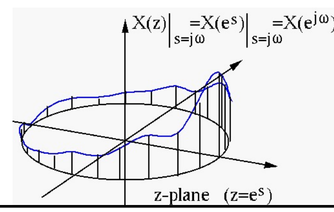


46

DTFT properties

- Polar Form
$$\begin{aligned} X(\omega) &= X_r(\omega) + jX_i(\omega) \\ &= |X(\omega)| e^{j<X(\omega)} \end{aligned}$$

- Periodicity
  - The Fourier transform is periodic, with period 2π:
$$X(\omega + 2\pi) = X(\omega)$$
  - So, it is sufficient to represent a discrete-time signal over the frequency range $[-\pi,\pi]$.



Discrete Fourier Transform

- The Fourier transform of a discrete-time signal is a continuous function of frequency. In practice, digital computers cannot work with continuous frequency. So, we need to sample the Fourier transform.

- For sequences of length N, we define a discrete-Fourier transform (DFT) as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \qquad 0 \le k \le N-1$$
$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn} \qquad 0 \le n \le N-1$$

Discrete Fourier Transform

- DFT of a finite length sequence (length N) is basically a discrete sequence X[k] of length N which correspond to X(z) {or X(ω)} evaluated at N equally-spaced points on the unit circle.

$$X[k]=X(e^{j\frac{2\pi k}{N}})=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi kn}{N}}\ ,\ \ k=0,1,\ldots,N-1$$

49

49

Important

	Time-Domain	**Frequency-Domain**
Continuous-time Fourier Series (CTFS) or (FS)	Continuous Periodic	Discrete Aperiodic
Continuous-time Fourier Transform (CTFT)	Continuous Aperiodic	Continuous Aperiodic
Discrete-Time Fourier Transform (DTFT)	Discrete Aperiodic	Continuous Periodic
Discrete Fourier Transform DFT	Discrete Periodic	Discrete Periodic

50

50

SAMPLING

51

51

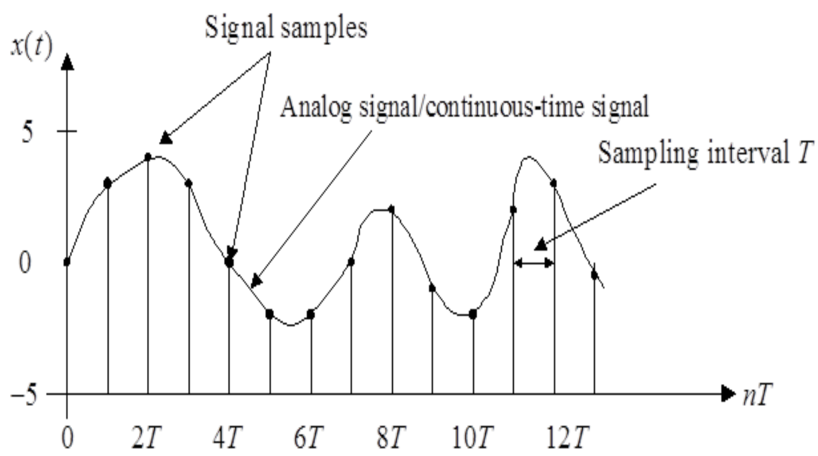
The sampling process

- Sampling is equivalent to taking snapshots of $x(t)$ every T_s seconds
- Each snapshot is called a *sample*
- T_s is the sampling interval, i.e., the time interval between each sample (second/sample)
  - Typically regularly spaced samples, though not necessary
- The sequence of samples is given by

$$x[n] = x(nT_s), \quad n \in I$$

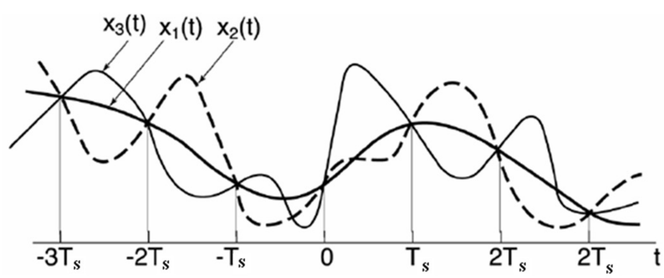
  - $F_S = \dfrac{1}{T_s}$ is the sampling frequency or rate (units of samples/second = hertz)

52



53

Sampling Interval Selection



- Given a set of sampling points, under what conditions can we uniquely reconstruct the original CT signals from which those samples came?
- Need to consider the frequency content of the CT signal being sampled → Fourier Transform

54

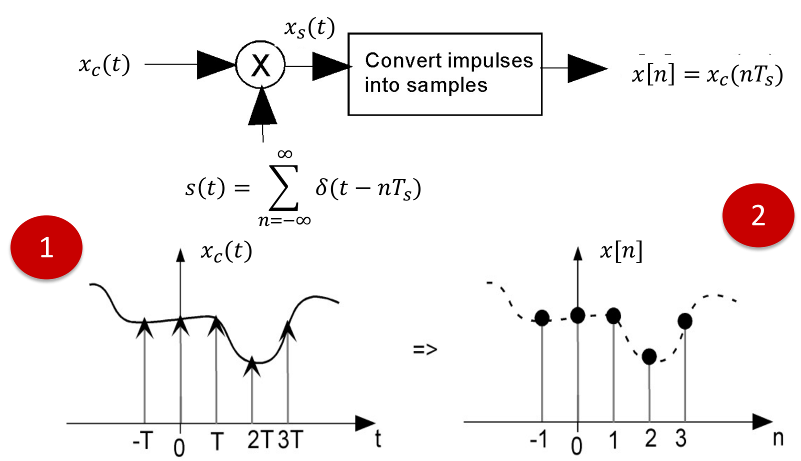
Sampling CT signals

- Obtain $x[n]$ from $x_c(t)$ in 2 steps:
  1. Sample $x_c(t)$ at uniform intervals T_s to obtain $x_s(t)$
  2. Convert sampled signal $x_s(t)$ into discrete-time signal $x[n]$

55

55

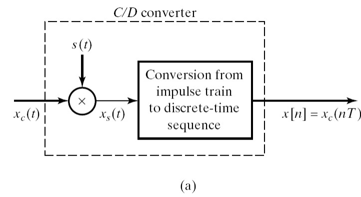
Sampling CT signals



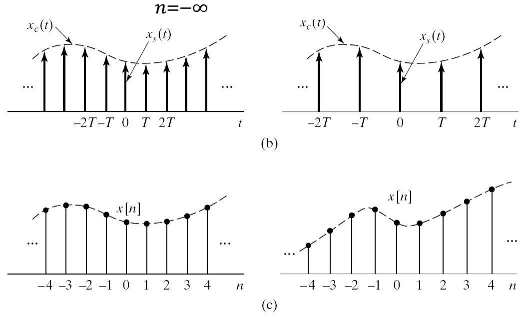
56

56

Sampling CT signal



$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$



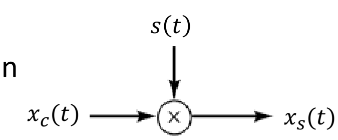
Two sampling rates

57

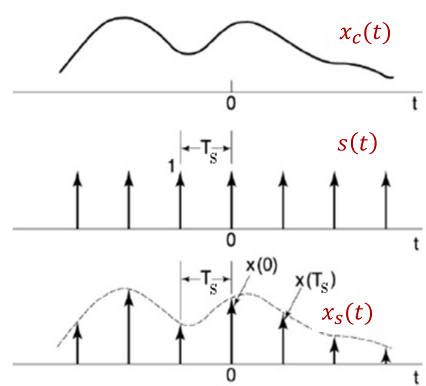
Step 1: Impulse sampling

- Multiplying x(t) by a sampling function

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$



$$\begin{aligned} x_s(t) &= x_c(t).s(t) \\ &= \sum_{n=-\infty}^{\infty} x_c(t)\delta(t - nT_s) \\ &= \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s) \end{aligned}$$



58

Step 1: Impulse sampling

- Multiplication in time is convolution in Freq $x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s)$

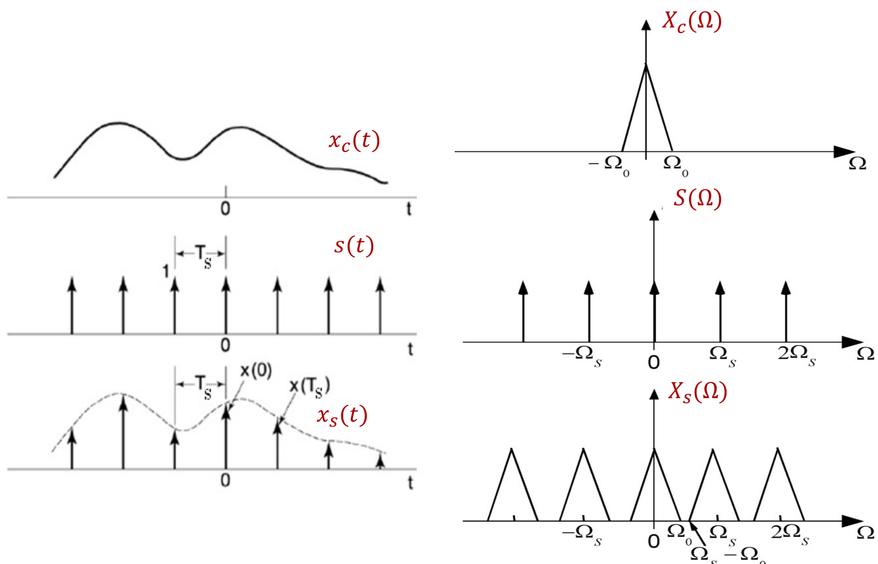
$$
\begin{aligned}
x_c(t) &\mapsto X_c(\Omega)\\
\sum_{n=-\infty}^{\infty} \delta(t - nT_s) &\mapsto \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s)\\
x_c(t).\sum_{n=-\infty}^{\infty} \delta(t - nT_s) &\mapsto X_c(\Omega) * \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s)\\
&= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c(\Omega - k\Omega_s)
\end{aligned}
$$

The Fourier transform of a delta comb is also a delta comb

Sampling Freq: $\Omega_s = 2\pi f_s = 2\pi \frac{1}{T_s}$

59

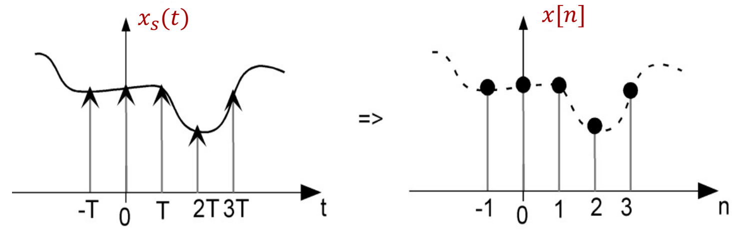
Step 1: Impulse sampling



60

Step 2: Discrete samples

- Step 2 involves conversion into discrete samples



Step 2: Discrete samples

- We know that:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s)$$ *Sampled signal*

$$x[n] = x_c(nT_s)$$ *Discrete-time signal*

- In freq. domain

$$\delta(t - t_0) \mapsto e^{-j\Omega t_0}$$

$$\delta(t - nT_s) \mapsto e^{-j\Omega nT_s}$$

$$x_c(nT_s)\delta(t - nT_s) \mapsto x_c(nT_s)e^{-j\Omega nT_s}$$

$$\underbrace{\sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s)}_{x_s(t)} \mapsto \underbrace{\sum_{n=-\infty}^{\infty} x_c(nT_s)e^{-j\Omega nT_s} \mapsto \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega nT_s}}_{X_s(\Omega)}$$

$$\boxed{X(\omega) = X_s(\Omega)|_{\Omega = \frac{\omega}{T_s}}}$$

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

C/D in Fourier Domain

- We have: Step1:

$$X_s(\Omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c(\Omega - k\Omega_s)$$

Step 2:

$$X(\omega) = X_s\,(\Omega)\Big|_{\Omega=\frac{\omega}{T_s}} = X_s\left(\frac{\omega}{T_s}\right)$$

- Therefore:

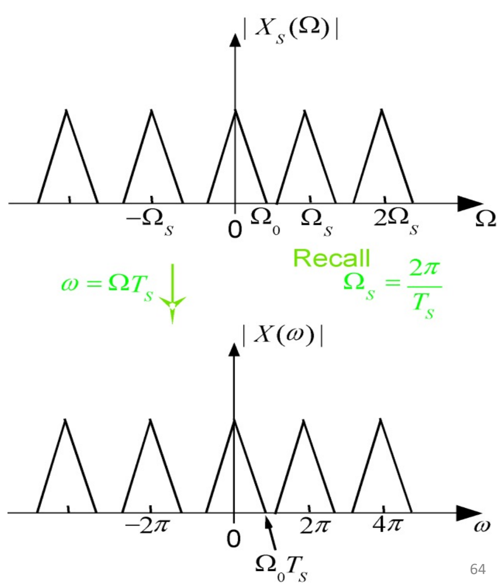
$$X(\omega) = X_s\left(\frac{\omega}{T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{T_s} - k\Omega_s\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{T_s} - k\frac{2\pi}{T_s}\right)$$

63

63

In other words…

- Make infinite number of duplicates of the spectrum of the continuous signal
- Scale the frequency axis by multiplying it by in order to obtain new axis
  - Makes periodic with a period (since is discrete)



64

64

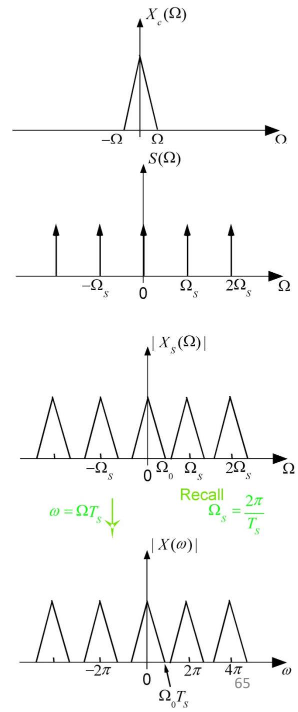
All together..

- Step1: $X_s(\Omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c(\Omega - k\Omega_s)$

  Step 2: $X(\omega) = X_s(\Omega)\Big|_{\Omega=\frac{\omega}{T_s}} = X_s\left(\frac{\omega}{T_s}\right)$

- Therefore:

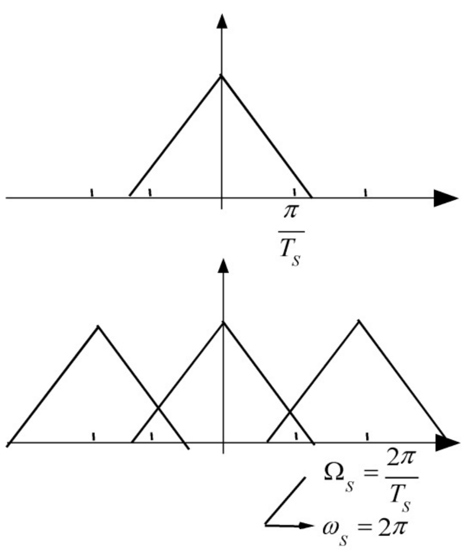
$$X(\omega) = X_s\left(\frac{\omega}{T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{T_s} - k\Omega_s\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{T_s} - k\frac{2\pi}{T_s}\right)$$



65

65

Aliasing



- Largest bandwidth allowed is π

66

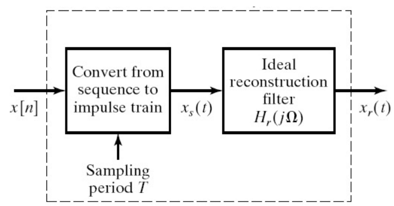
66

D/C Conversion

- The reconstruction of continuous-time signal involves 2 steps:
  - **Step1:** convert samples of into a sequence of impulses:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_s)$$

  - **Step2:** filter $x_s(t)$ with ideal low pass filter (reconstruction filter) with frequency response:



$$H_r(\Omega) = \begin{cases} T_s & , \quad |\Omega| \le \dfrac{\pi}{T_s} \\ 0 & , \quad |\Omega| > \dfrac{\pi}{T_s} \end{cases}$$

or

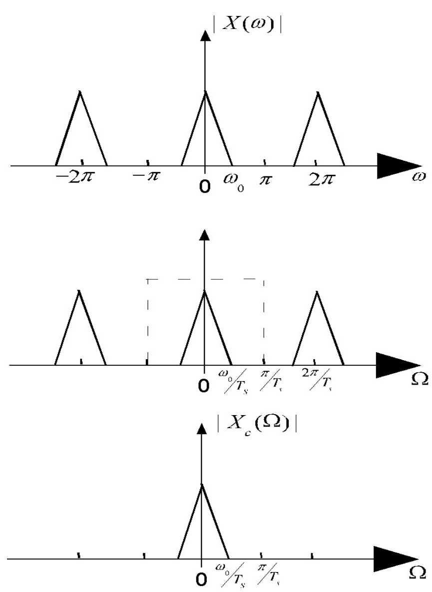
$$h_r(t) = \frac{\sin\left(\frac{\pi t}{T_s}\right)}{\frac{\pi t}{T_s}}$$

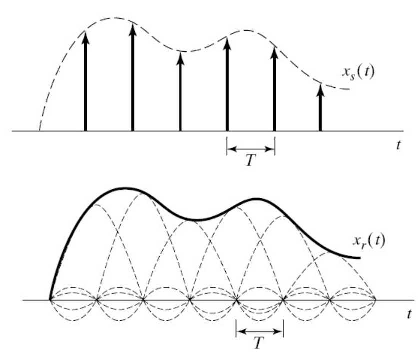
$$X_c(\Omega) = H_r(\Omega)X(\Omega T_s)$$

$$\begin{aligned} x_c(t) &= x_s(t) * h_r(t) \\ &= \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT_s) \end{aligned}$$

67

D/C Conversion





68

68

Nyquist Sampling Theorem

- Let $x(t)$ be a band-limited signal such that

$$X(\Omega) = 0, \qquad |\Omega| > \Omega_M$$

Then, $x(t)$ is uniquely determined by its samples $x(nt), n \in \mathbb{Z}$ if

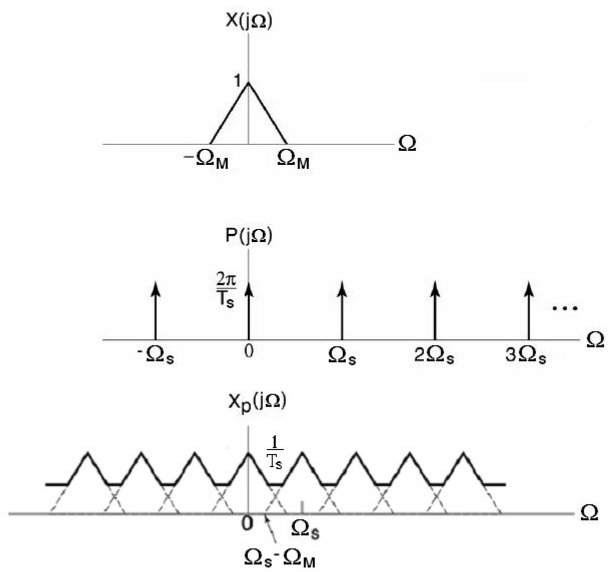
$$\boldsymbol{\Omega_s > 2\Omega_M}$$

where $\Omega_s = \frac{2\pi}{T_s}$ is the sampling frequency

and Ω_M is called the Nyquist frequency

69

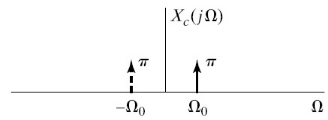
Aliasing Effect

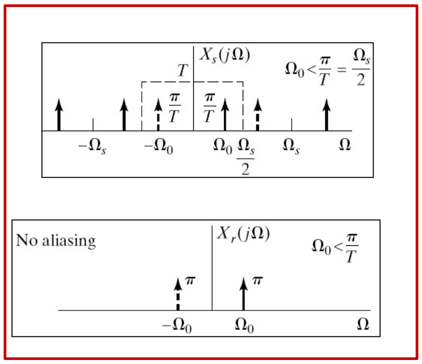


70

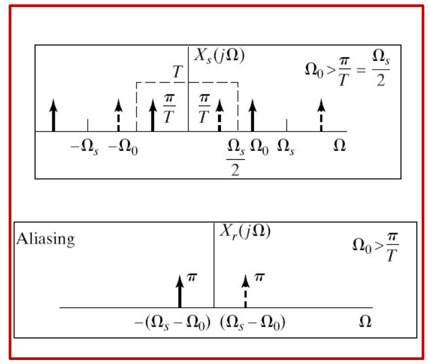
Aliasing example: cos signal

- Suppose $x_c(t) = \cos(\Omega_0 t)$





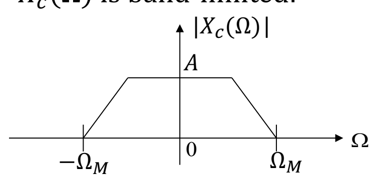
No Aliasing



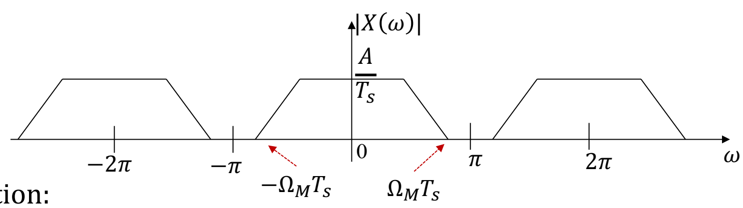
Aliasing

Oversampling

- Suppose that $x_c(t) \leftrightarrow X_c(\Omega)$ is band-limited:



- Then if T_s is sufficiently small, $X(\omega)$ appears as:

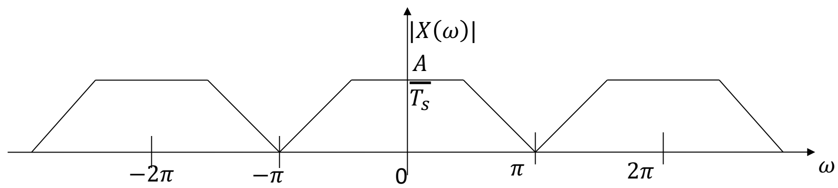


- Condition:

$\boxed{\Omega_s > 2\Omega_M}$ or $\boxed{\Omega_M T_s < \pi}$

Critically sampling

Critically sampled: $\boxed{\Omega_s = 2\Omega_M}$ or $\boxed{\Omega_M T_s = \pi}$

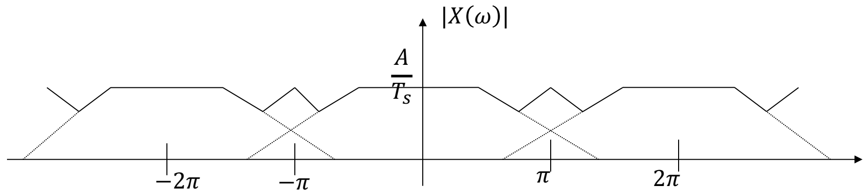


- According to the Sampling Theorem, in general the signal cannot be reconstructed from samples at the rate $T_s = \frac{\pi}{\Omega_M}$
- This is because of errors will occur if $X_c(\Omega_M) \neq 0$, the folded frequencies will add at $\omega = \pi$

73

Undersampling (aliasing)

If sampling theorem condition is **not** satisfied $\boxed{\Omega_s = 2\Omega_M}$ or $\boxed{\Omega_M T_s = \pi}$



- The frequencies are **folded** and **summed**
  - This **changes** the shape of the spectrum.
  - The process is not reversible.
- If a signal is not **strictly band-limited**, sampling can still be done at twice the **effective band-limited**.

74

Sampling rate conversion

- i.e. Resampling in the digital domain
- **Case 1:** reducing by an integer
  - Goal: Reduce sampling rate by factor of M
    - (instead of 10000 samples per second, use only 5000 samples per second)

75

Downsampling

- Goal: Reduce sampling rate by factor of M
  - New sampling rate: $T' = MT_s$
  - Resampled signal $x_d[n] = x_c(nT') = x_c(MT_s) = x[nM]$ ($x_d[.]$ for decimated)
  - So, we can reduce the rate by taking every Mth sample of $x[n]$
    - System called <u>downsampler</u>

$$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow x_d[n]$$

  - Dowsampling often results in aliasing

76

Downsampling

- Recall:

$$X(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{T_s} - k\frac{2\pi}{T_s}\right)$$

Similarly:

$$X_d(\omega) = \frac{1}{MT_s} \sum_{l=-\infty}^{\infty} X_c\left(\frac{\omega}{MT_s} - l\frac{2\pi}{MT_s}\right)$$

- We can show that:

$$\boxed{X_d(\omega) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\omega}{M} - k\frac{2\pi}{M}\right)}$$

77

77

Downsampling

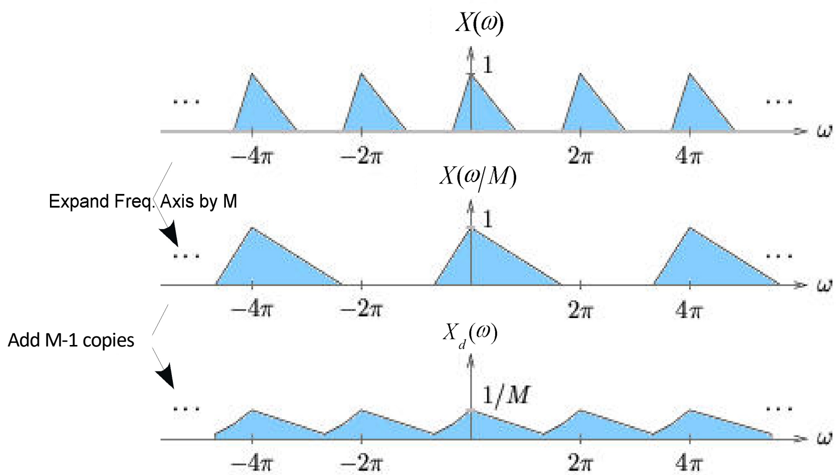
$$\boxed{X_d(\omega) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\omega}{M} - k\frac{2\pi}{M}\right)}$$

- The equation above means:
  - Expand frequency axis by factor of M
  - Make M copies within interval of 2π
  - Divide spectrum by M

78

78

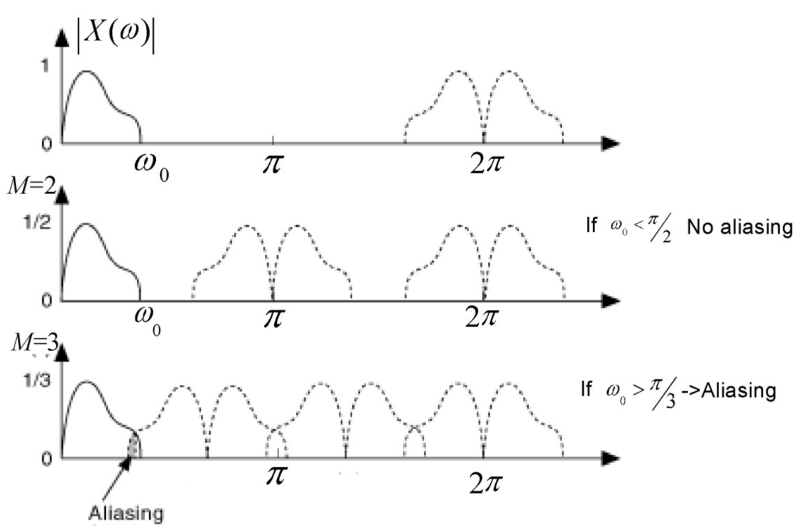
Downsampling (Case M=2)



79

79

Another example

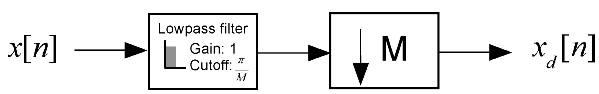


80

80

Dealing with aliasing

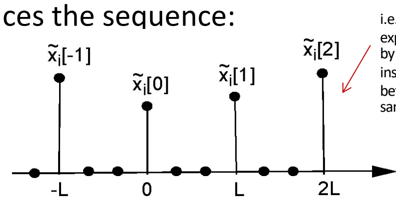
- In order to prevent aliasing, x[n] should have its highest frequency limited to $\frac{\pi}{M}$
  - should be filtered by a lowpass filter with cutoff $\frac{\pi}{M}$
- The cascade of low-pass filter and downsampler is called a decimator



Case 2: Increasing rate by integer

- Steps:
  - Increase by factor L
  - New sampling rate: $T' = \frac{T_s}{L}$
  - So, $x_i[n] = x_c\left(n\frac{T_s}{L}\right)$ (interpolator)
  - Samples of $x_i[n]$ for values of n that are integer multiple of L: $x_i[nL] = x[n]$
  - So, an upsampler produces the sequence:

$$\breve{x}_i[n] = \begin{cases} x\left[\frac{n}{L}\right] & , \quad n = 0, \pm L, \pm 2L, \dots \\ 0 & , \quad otherwise \end{cases}$$



i.e. Upsampler expands time scale by factor of L by inserting L-1 zeros between each sample of x[n]

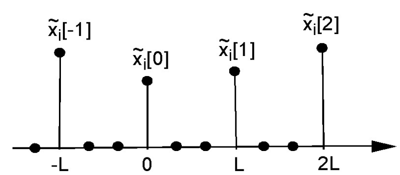
Upsampling in Frequency domain

$$
\begin{aligned}
\breve{X}_i(\omega) &= \sum_{n=-\infty}^{\infty} \breve{x}_i[n]\, e^{-jn\omega} \\
&= \sum_{n=-\infty}^{\infty} x[n]\, e^{-jnL\omega} \\
&= X(L\omega)
\end{aligned}
$$

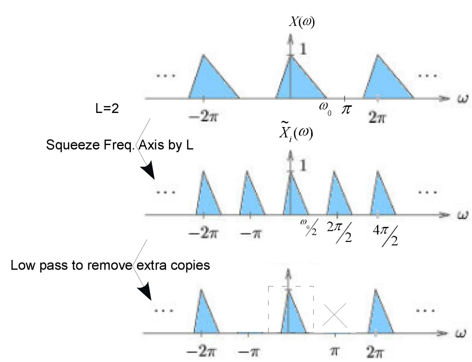
- Upsampling is simply a scaling in frequency

Upsampling (cont.)

- Now.. We need to fill the zeros

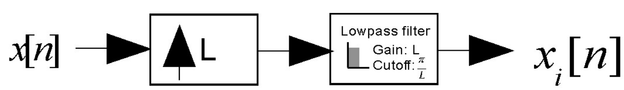


  – Done by filtering $\breve{x}_i[n]$ with a lowpass with cutoff $\frac{\pi}{L}$ and gain L



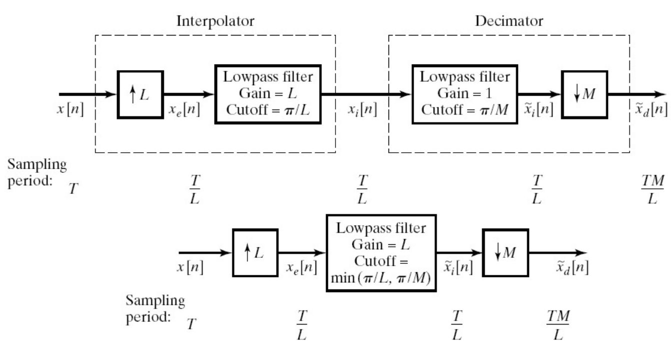
Interpolator

- Cascade of upsampler and lowpass filter is called interpolator



Sampling rate conversion by rational factor

- If we have a rational factor (say 2.5), it corresponds to a change of sampling rate of 5/2; i.e. $^L/_M$
  - Downsampling by M can create aliasing
  - Upsampling by L is safe
  - Start with upsampling, then downsampling



DISCRETE-TIME FILTERS

87

87

Digital Filters

- A digital filter is a discrete-time linear time-invariant system that performs a filtering operation
  - **Filtering**: refers to extracting a desired signal from noise

88

88

Discrete-time Filters

- Filters are a special case of LTI systems. They are defined by frequency response:
- It has two parts:

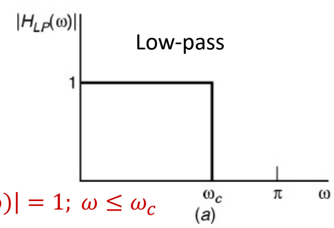
$$H(\omega) = \underbrace{|H(\omega)|}_{\text{Magnitude response}} \underbrace{e^{j\varphi(\omega)}}_{\text{Phase response}}$$

89

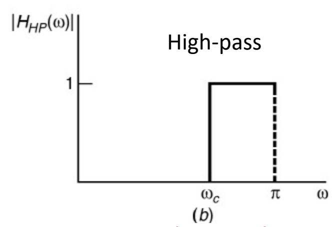
89

Ideal Magnitude Response

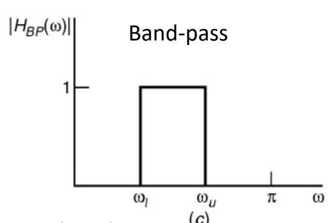
Recall Eigenfunction property of LTI systems



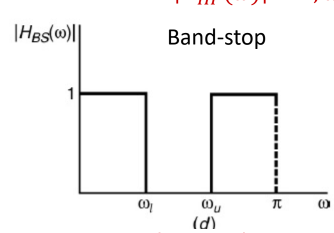
$|H_{LP}(\omega)| = 1;\ \omega \le \omega_c$



$|H_{HP}(\omega)| = 1; \omega_c \le \omega < \pi$



$|H_{BP}(\omega)| = 1; \omega_L \le \omega \le \omega_U$



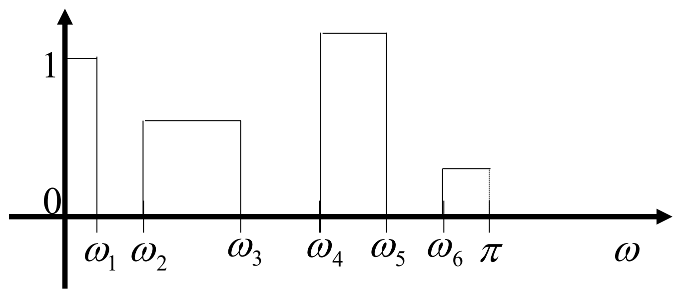
$|H_{BS}(\omega)| = 0; \omega_L \le \omega \le \omega_U$

90

90

Multiband filters

- You can combine multiple simple filters into a multi-band filter



91

91

Ideal filters

- The phase response $< H(\omega)$ of an ideal filter is linear

$$< H(\omega) = -t_0 \omega$$

Linear phase is often crucial for sound processing applications

  – The group delay function (derivative of phase function) is a constant

$$\tau(\omega) = -\frac{d < H(\omega)}{d\omega} = t_0$$

92

92

Difference Equation

- The filters of interest often satisfy a difference equation that relates the input and output:

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

- Evaluating the z-transform of this equation:

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

Rational function in z^{-1}

Canonical form showing poles and zeros

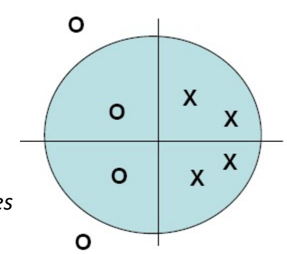
93

93

Difference Equation

- In general, we can factorize H(z)

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



M zeroes, N poles

– This system is:
  - Causal ($h[n] = 0, n < 0$) if Region of Convergence: $|z| > R_1$
  - Stable if ROC contains unit circle; so $R_1 < 1$

  $\Rightarrow$ Stable and causal system if all poles inside unit circle

  $\Rightarrow$ If all zeros (in addition to poles) are inside unit circle, system is minimum-phase
    - Guarantees system and its inverse are causal and stable
    - Has minimum group delay

94

94

Types of filters

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

If current output depends on past/present inputs <u>and</u> outputs
$(N, M \neq 0)$

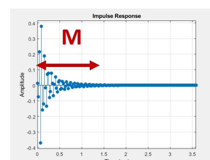
IIR system
(**I**nfinite **I**mpulse **R**esp)

System is not guaranteed to be stable (recursion/feedback)

If current output depends only on past/present inputs
$(N = 0)$

FIR system
(**F**inite **I**mpulse **R**esp)

$$h[n] = \begin{cases} b_n & , \quad 0 \leq n \leq M \\ 0 & , \quad otherwise \end{cases}$$



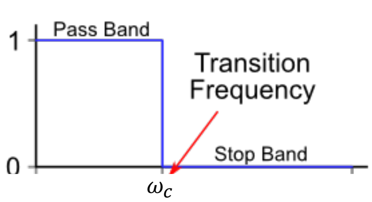
95

FIR filter design

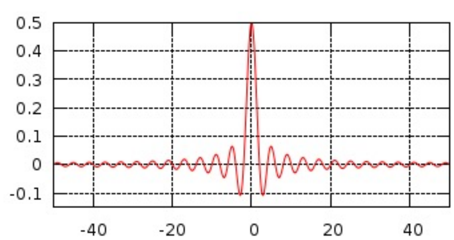
$$y[n] = \sum_{l=0}^{M} b_l x[n-l]$$

Non-recursive equation

ideal lowpass filter



In time-domain



Sinc function $h[n] = \dfrac{\sin(2\pi\omega_c n)}{\pi n}$

96

96

FIR filter design

- **Goal:** Design a FIR filter that corresponds to sinc function impulse response
  - Option: Truncate $h[n]$ to some finite-duration M
    - Corresponds to multiplying $h[n]$ by a 'window'

☞ Called: Windowing design method
  - Convenient but not optimal

97

97

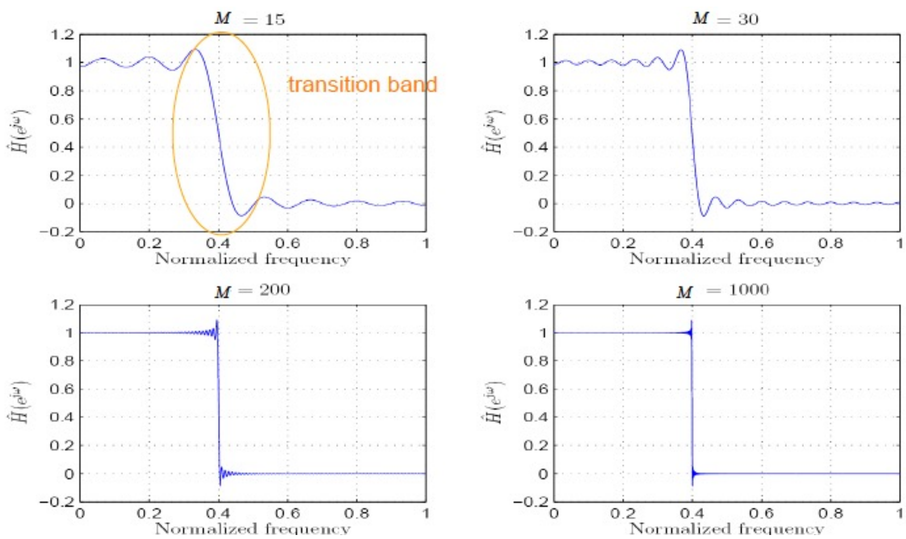
FIR Window design

- There are 2 issues with window design methods:
  1. Need to truncate infinite impulse response
     - Multiply by a window / convolution by frequency response
     - Choice of window & frequency response?
  2. The impulse response (sinc function) is non-casual
     - Introduce a shift in impulse response
     - Resulting filter is causal

98

98

Truncated Low-pass FIR



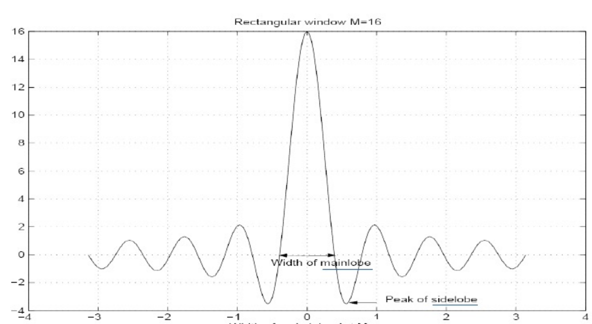
Transition band gets narrower, but ripples remain

99

99

Window design method

- Ripples are caused by rectangular window
- Note: Fourier transform of rectangular window

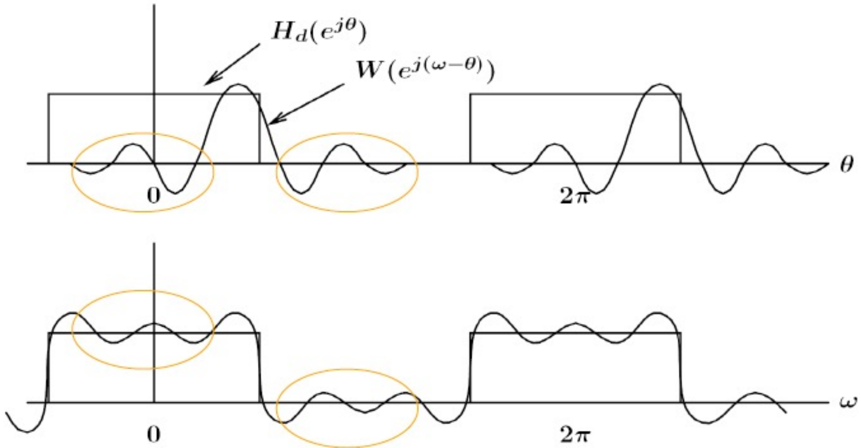


- Recall: $h[n]$ .w[n] $\Longleftrightarrow$ $H(\omega) * W(\omega)$

100

100

Truncated filter in Fourier domain



101

101

FIR filter design

- Choice of window: Ideally, want:
  - M small (impulse response as short as possible)
    - ✓Fewest number of computations possible
  - $W(\omega)$ as close as possible to a dirac delta function
    - ✓Match as close as possible ideal low-pass filter

◆ These 2 requirements are at odds!

102

102

FIR filter design

- Choice of window:
  - Use windows with no abrupt discontinuity in their time-domain response and consequently low side-lobes in their frequency response.
  - In this case, the reduced ripple comes at the expense of a wider transition region but can be compensated for by increasing the length of the filter.

103

103

Common windows

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

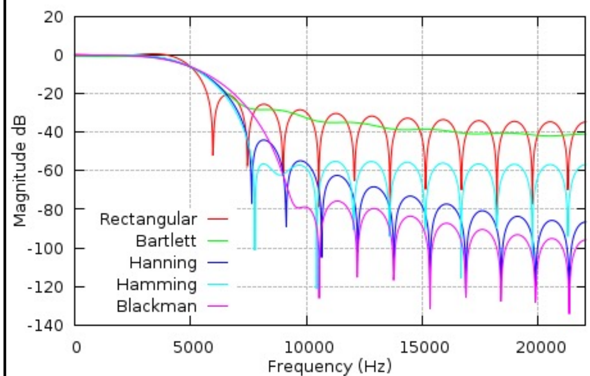
5. Hanning $w[n] = 0.5 - 0.5\cos\left(\dfrac{2\pi n}{M}\right)$

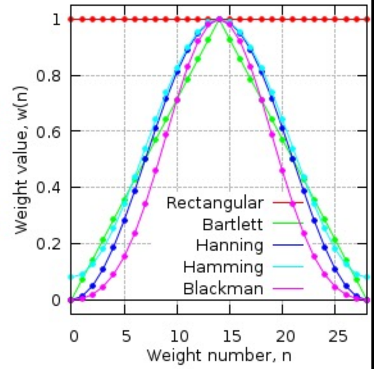
6. Kaiser $w[n] = \dfrac{I_0\left\{\beta\sqrt{1-((n-M/2)/(M/2))^2}\right\}}{I_0\{\beta\}}$

104

104

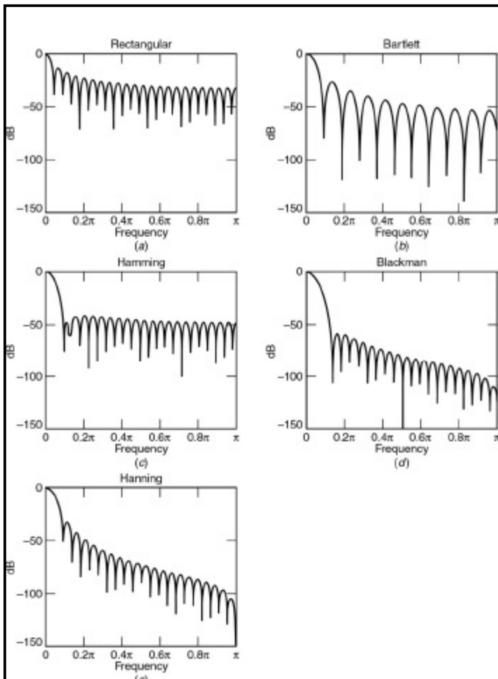
Choice of window

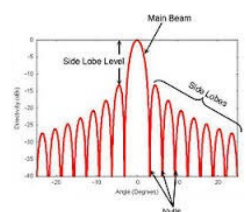




105

105

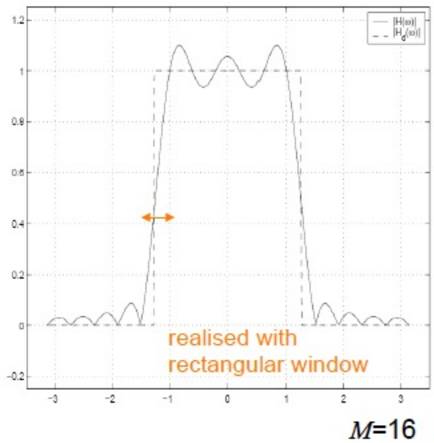


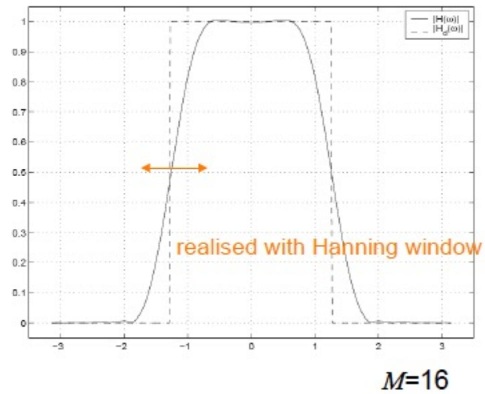


Window	Mainlobe Width	Sidelobe Attenuation
Rectangular	$4\pi / M$	-13 dB
Bartlett	$8\pi/M$	-27 dB
Hanning	$8\pi/M$	-32 dB
Hamming	$8\pi/M$	-43 dB
Blackman	$12\pi/M$	-58 dB

106

Filter realized with different windows

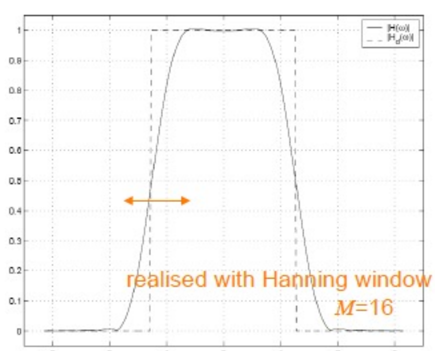


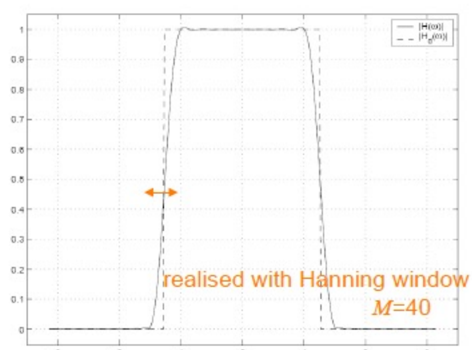


Note: There are much less ripples for the Hanning window but the transition width has increased

107

Filter realized with Hanning window





Transition width can be improved by increasing the size of the Hanning window to M = 40

108

Window characteristics

- Fundamental trade-off between main-lobe width and side-lobe amplitude
- As window becomes smoother, peak side-lobe decreases, but the main-lobe width increases.
- Need to increase window length to achieve same transition bandwidth.

109

109

FIR Filters in MATLAB

- MATLAB FIR1: window-based standard lowpass, bandpass, highpass, bandstop, and multiband configurations
- MATLAB FIR2: Designs a FIR digital filter any arbitrary frequency response
- MATLAB FIRPM: Designs a FIR digital filter with arbitrary constraints by finding optimal filter between desired and achieved response
- The function Freqz: Evaluates the Filter Frequency Response

110

110

IIR filter design

- 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 ∞) to the digital frequency scale (0 to π); use frequency pre-warping to preserve critical frequencies of transformation (i.e., filter cutoff frequencies)

112

112

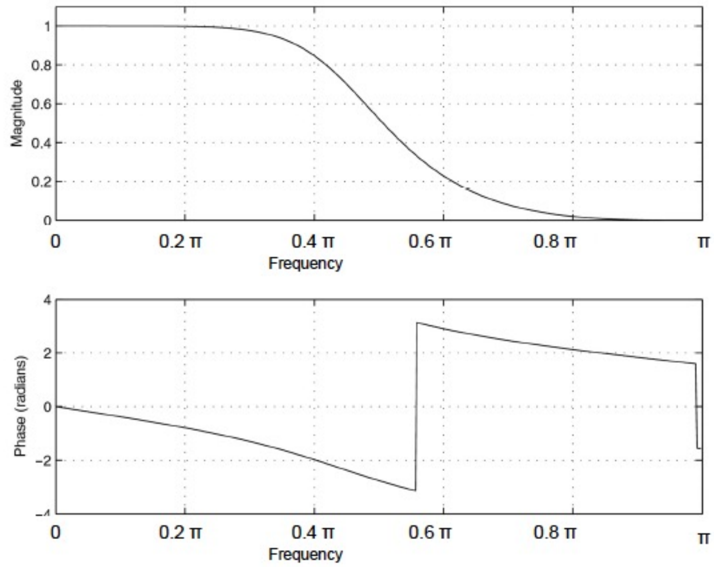
IIR Filters

- 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
  – *Chebyshev* designs-equi-ripple in either passband **or** Stopband
  – *Elliptic* designs-equi-ripple in both passband **and** stopband

113

113

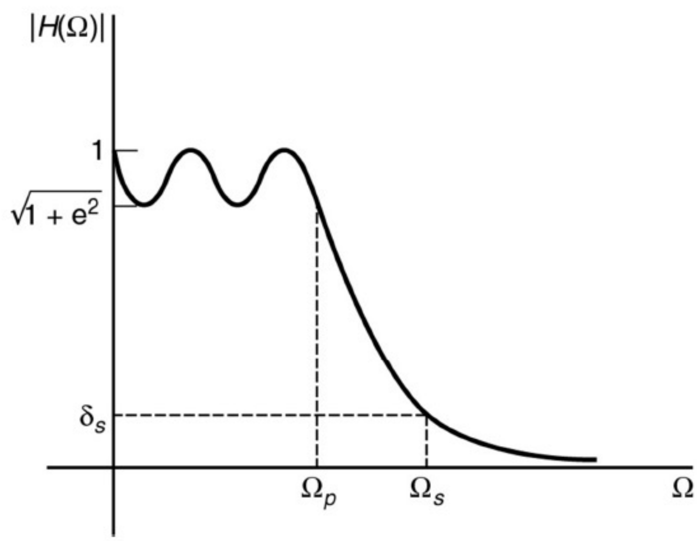
Butterworth design



114

114

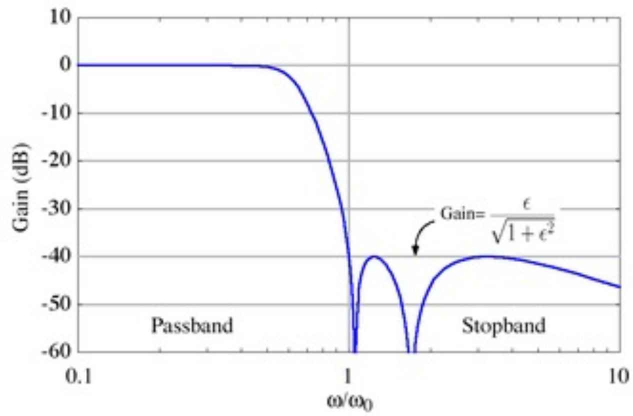
Chebyshev Type I Design



115

115

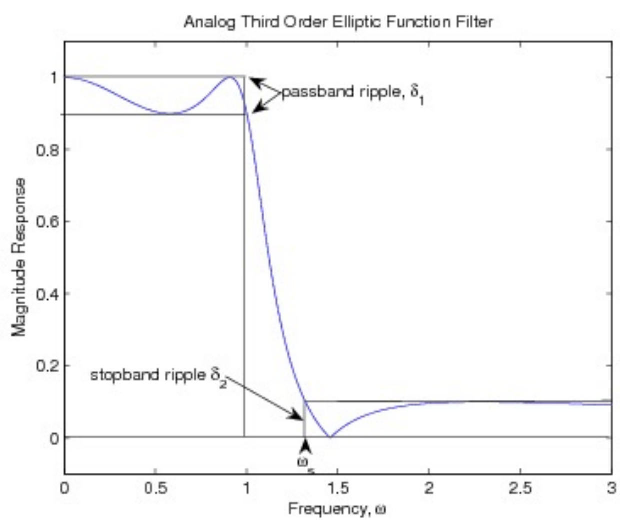
Chebyshev II Design



116

116

Elliptic Filter Design

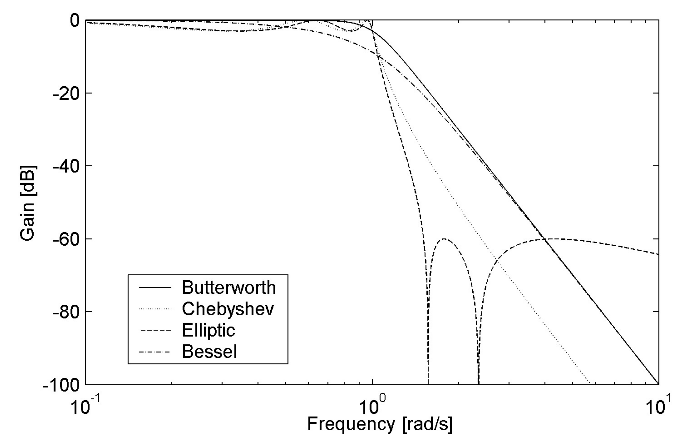


117

117

Which filter?

Choice of filter is a tradeoff between simplicity, sharp transitions, oscillations (ripples in passband and stopband)



118

118

Example: MATLAB

- 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)

119

119