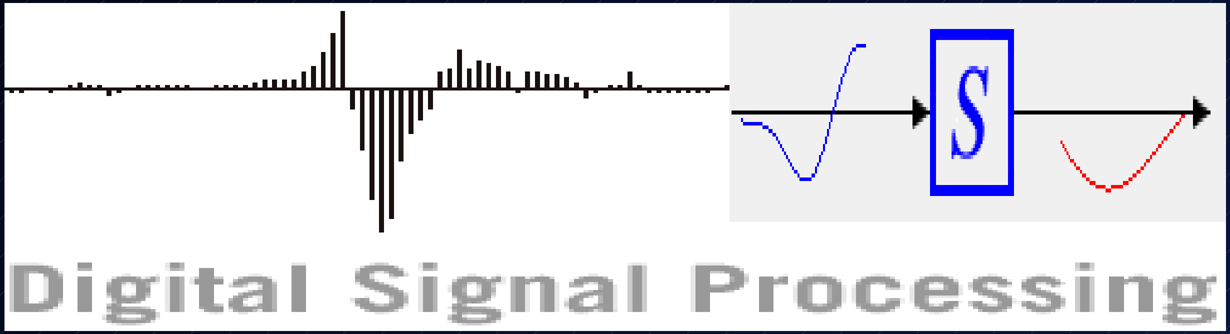
Digital Signal Processing



By: Prof. M.R.Asharif

University of the Ryukyus

Department of Information Engineering

Introduction (1)

DSP is different from other areas in computer science by the type of data it uses: *signals.*

DSP is the mathematics and algorithms that used to manipulate these signals. But first it should be converted into digital form.

Introduction (2)

Digital Signal Processing (DSP) Deals with the transformation of signals that are discrete in both amplitude and time

- Is area of science and engineering developed rapidly over the last 30 years.
- DSP is a result of significant advances in:
  - Digital computer technology
  - Integrated circuit fabrication
- DSP involves time and amplitude quantization of signals and relies on the theory of discrete time signals and systems.

History of DSP

The roots of DSP are in the 1960s and 1970s when digital computers first became available.

With generation of personal computers in 1980s and 1990s, DSP has been found to have more applications.

Applications (2)

Most DSP applications deal with analogue signals.

-the analogue signal has to be converted to digital form

-Information is lost in converting from analogue to digital

-When the signal is converted to digital form, the precision is limited by the number of bits available.

Early applications of DSP

radar & sonar, where national security was at risk;

oil exploration, where large amounts of money could be made;

space exploration, where the

The Scientist and Engineer's Guide to Digital Signal Processing

New Applications

After PC has been developed enough (1980-1990), new commercial applications were expanded for DSP, such products as:

Mobile telephones, compact disc players, and electronic voice mail.

Interdisciplinary DSP

DSP is very interdisciplinary and it has fuzzy and overlapping borders with many other

areas of science, engineering and mathematics

Such as:

Communication Theory, Numerical Analysis,

Probability and Statistics, Analog Signal Processing, Decision Theory, Digital Electronics,

Analog Electronics

Area of DSP Applications

DSP Application is mainly for two purposes:

1- To Design a Digital Filter

2- For Spectral analysis

Telecommunications: Multiplexing, Compression, Echo control

Audio Processing: Music, Speech generation, Speech recognition

Echo Location: Radar, Sonar, Reflection seismology

Image Processing: Medical, Space, Commercial Imaging Products

ADC and DAC

Signals that encountered in Engineering are usually Continuous, such as: changes of light intensity with distance; voltage that varies over time etc. Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) are the processes that allow digital computers to interact with these everyday signals. Digital Signal is referred as signal that is sampled and quantized. By sampling frequenc and number of bits for quantization, one can decide how much information contains in digital signal.

From Continuous to Digital (1)

Two processes are made in digitization, ADC:
2) Sample and Hold (S/H) and (2) Quantization
S/H changes time variable from continuous to discrete.
Quantization assigns an integer value to each discrete flat region. That is converting voltage from continuous to discrete.
Sampling without quantization is used in switched capacitor filters.

From Continuous to Digital (2)

Quantization: A process in which the continuous range of values of an analog signal is sampled and divided into A discrete time discrete valued (Digital) signal.

Two types of Quantization method

1. Truncation (cutting decimal points) example 126.66=126
2. Rounding off (Rounding to the upper or lower value) example 126.7=127 and 126.3=126.

Quantization Error

Maximum error in any digitized sample is: **±½ LSB.** The quantization error appears very much like *random noise* with uniform Distribution between +-1/2 LSB with zero mean 1/ root(12) LSB as standard deviation.

For example:

With 8 bits: added rms noise is:

1/ root(12)*1/256 = 1/900

With 12 bits: added rms noise =1/14000

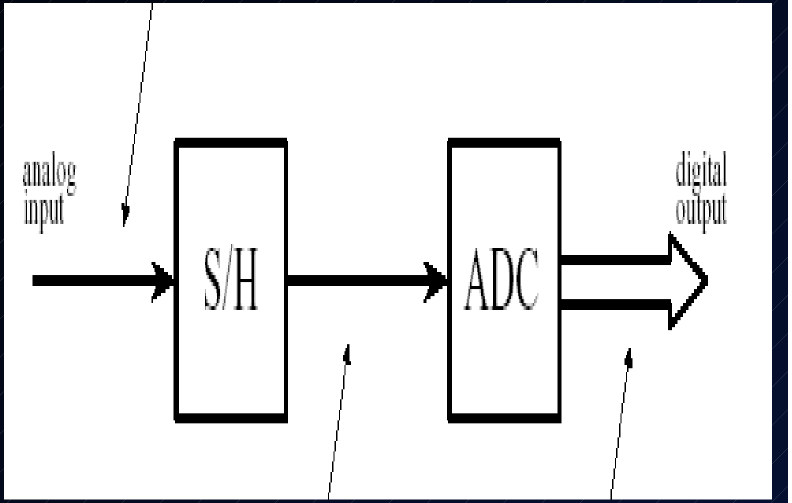
With 16 bits: added rms noise =1/227000

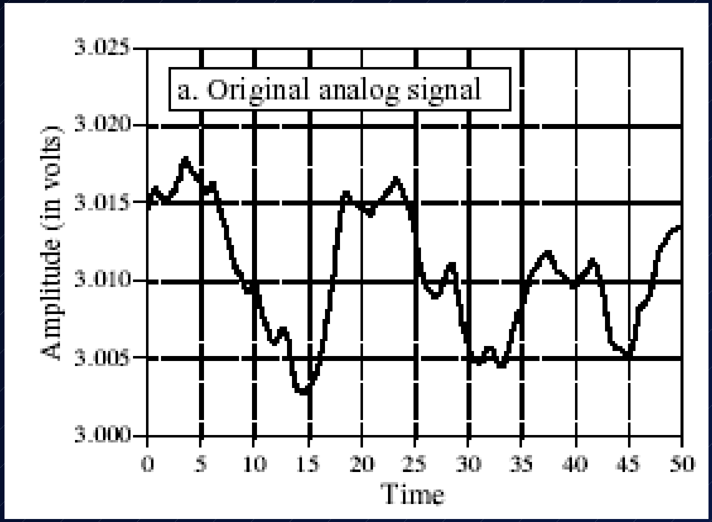
Notes: The more bit you use the less quantization noise (error) you get.

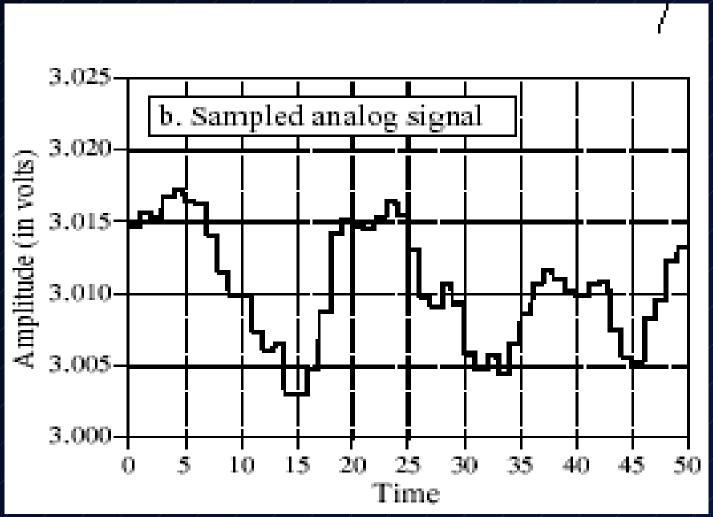
For example: For speech use at least 16 bit.

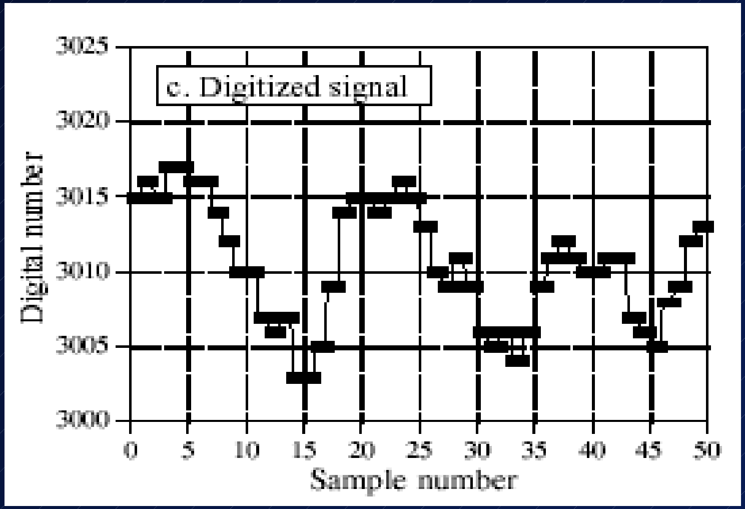
For image use at least 8 bit

Processes of ADC







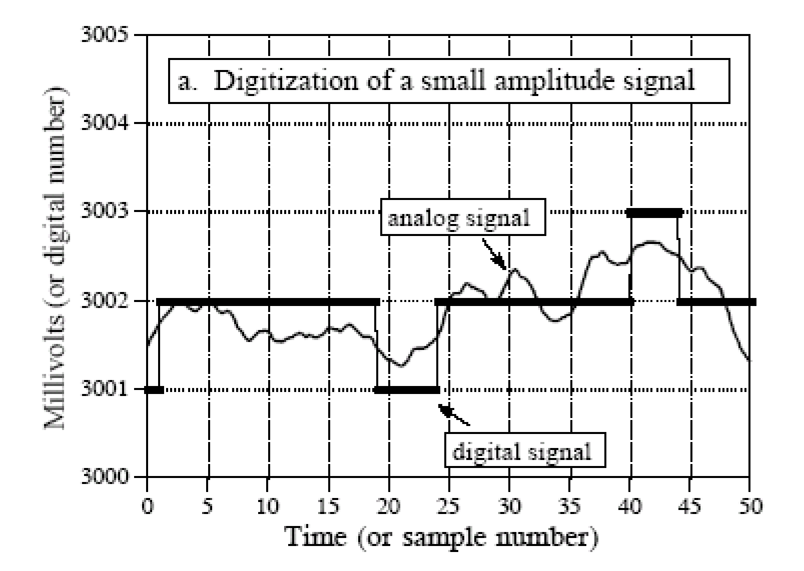


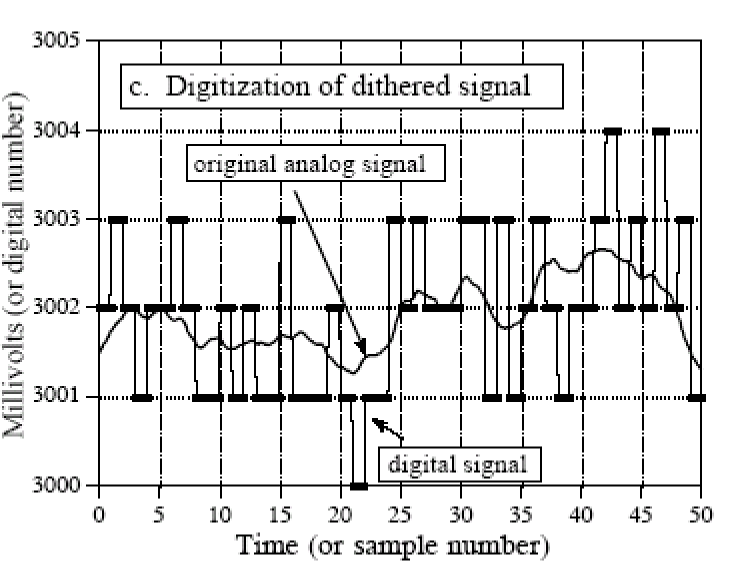
Dithering

- **Dithering** is a common technique for improving the digitization of these

  slowly varying signals.

This is quite a strange situation: *adding noise provides more information*.





The Sampling Theorem

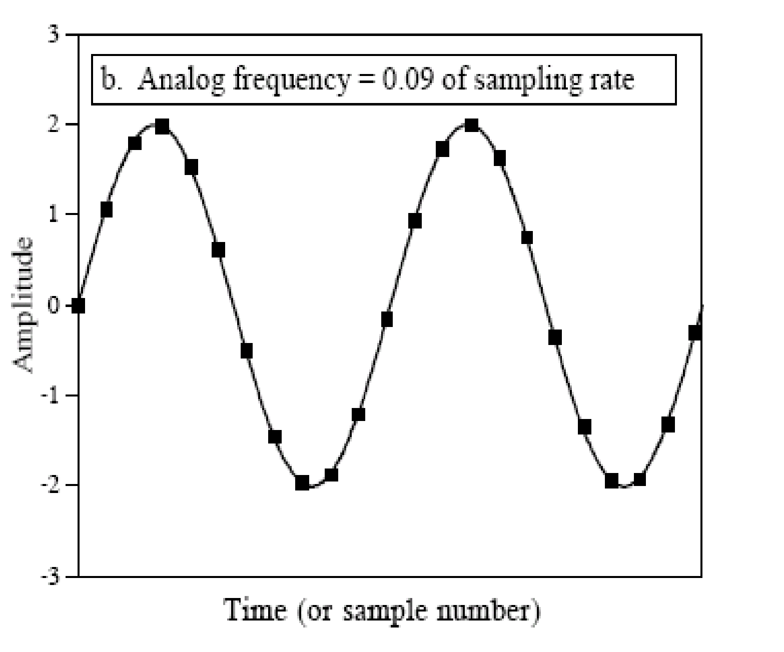
- If you can exactly *reconstruct* the analog signal from the samples, you must have done the sampling *properly*.

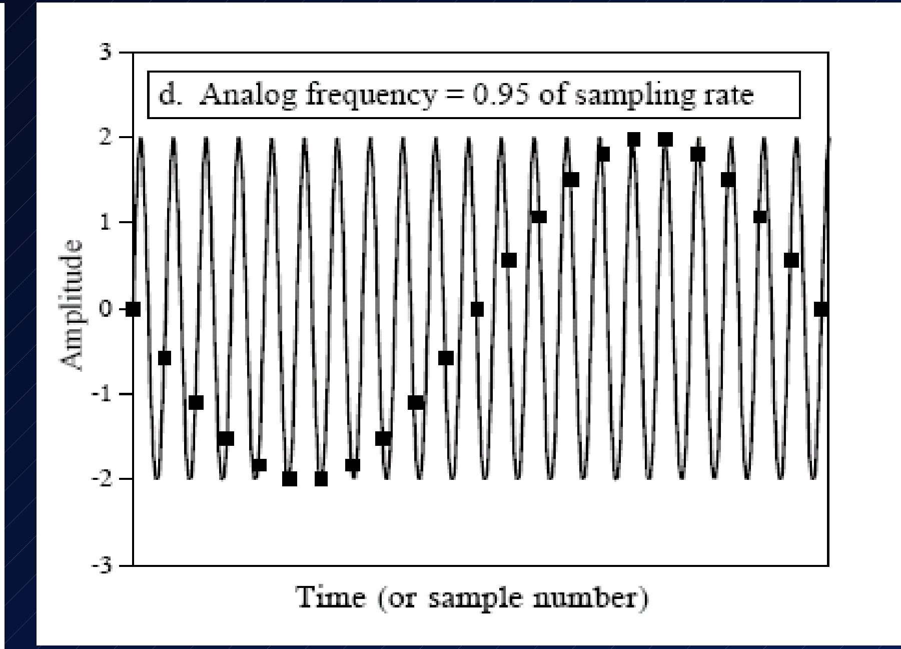
A continuous signal can be properly sampled, *only if it does not contain frequency components above one-half of the sampling rate*.

a sampling rate of 2,000 samples/second is good for up to 1000 Hz analog signal

Two examples for sampling

good irreversible





Advantage of Digital Signal Processing Over Analogue Signal Processing

1. DSP is programmable:

   -In DSP it is possible to change the program of the hardware with out changing the hardware design.

2. Off line processing (i.e. possible to design and test in simulation using PCs etc.)
3. Possible to recover the signal (i.e. Robust Noise)
4. Possible for having time sharing.

   Possible to use it for very large scale Integrated (VLSI) technology

Disadvantages

1. Large Bandwidth and CPU demand.
2. Distortion when converting Analogue to Digital is not perfect reconstruction of the original analogue signal.
3. DSP designed can be expensive
4. The design of DSP systems can be extremely time consuming and a high complex and specialized activity.

What is a Signal?

A signal is defined as a function of one or more variables, that conveys information on the nature of the physical phenomenon

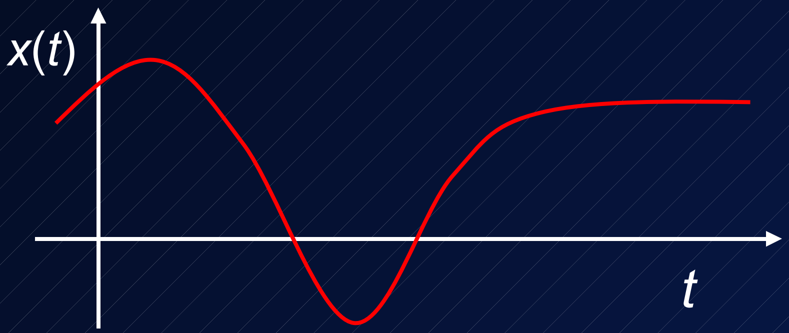
- A signal is a pattern of variation of some form
- Signals are variables that carry information

Examples of signal include:

- Electrical signals
  - Voltages and currents in a circuit
- Acoustic signals
  - Acoustic pressure (sound) over time
- Mechanical signals
  - Velocity of a car over time
- Video signals
  - Intensity level of a pixel (camera, video) over time

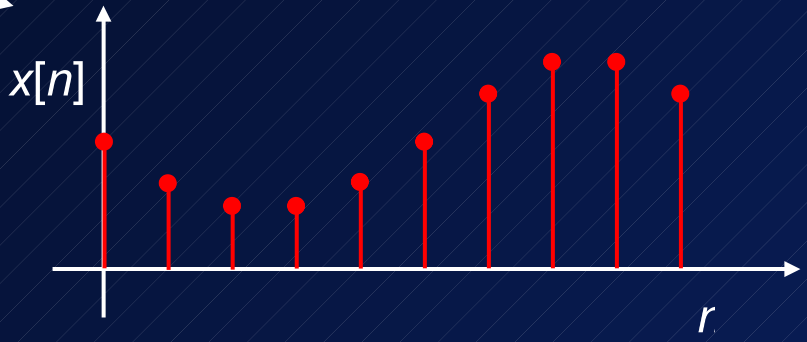
Continuous & Discrete-Time Signals

Continuous-Time Signals (Analogue)



- Are defined along a continuum of time and thus represented by a continuous independent variable and referred as analogue signal.
- E.g. voltage, velocity,

Discrete-Time Signals



- Are defined at discrete time and thus independent variable has discrete value i.e. represented mathematically as a sequence of numbers.
- Denote by $x[n]$, where n is an integer value that varies discretely

Discrete-Time Signals

- Exists only at discrete points in time.
- Often obtained by sampling on analogue signal i.e. measuring it’s value a distinct points in time.
- Sampling points usually separated by equal intervals of time.
- Given an analogue signal f(t), let f(n) be a value

of f(t) when t=nT can be written as:

$$f(t)=\sin\ (wt)=\sin(2\pi ft)$$

→ for Analogue

when t=nT

where n= integer

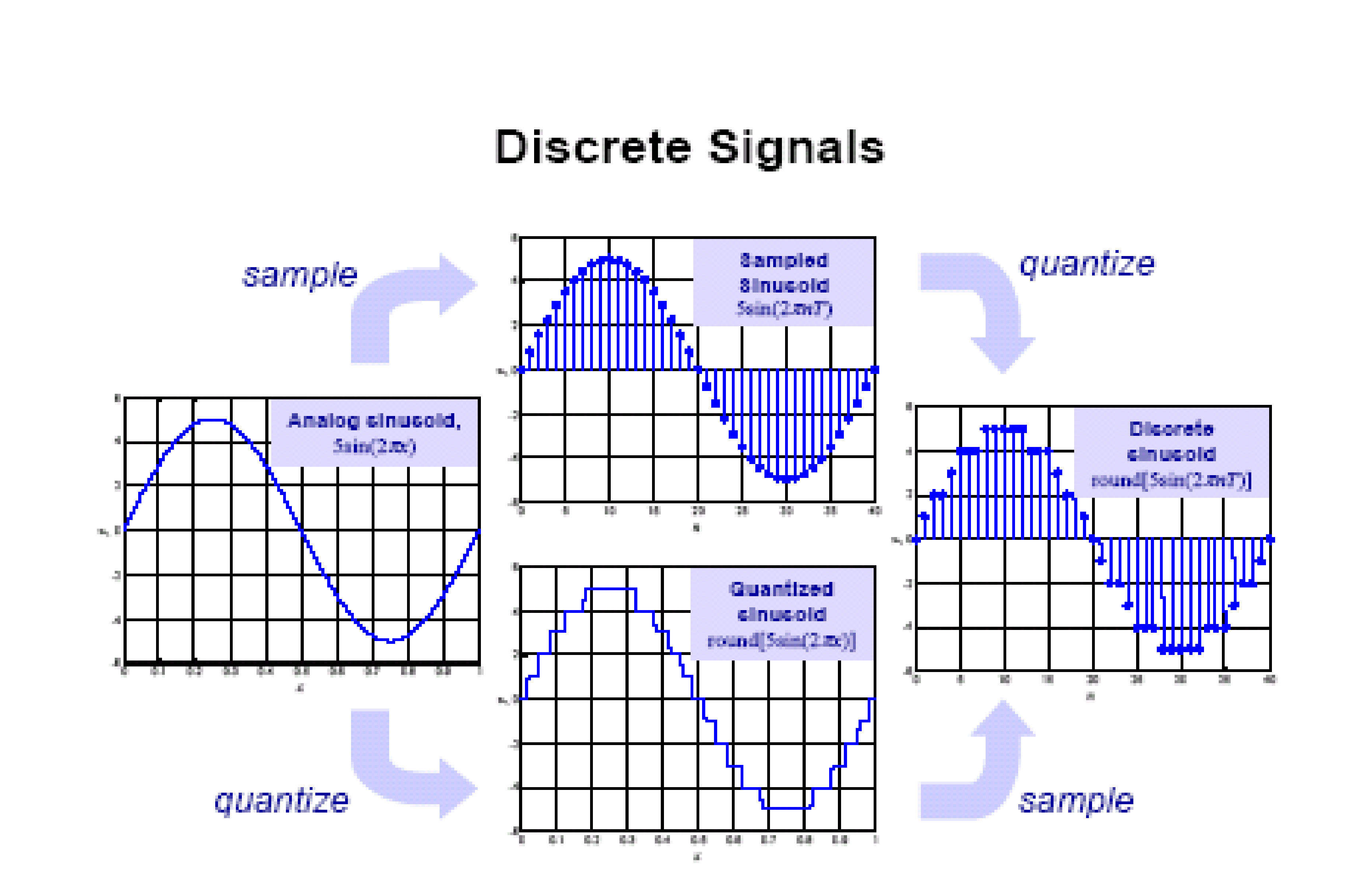
$$f(nT)=\sin(2\pi fnT\ \pi)$$

→ Discrete time

Continuous valued versus Discrete-valued signals.

- A signal is continuous valued if it takes on all possible values on a finite or an infinite range.
- A signal is discrete valued if it takes on values from a finite set of possible values, usually equidistant values.
- A discrete time signal having a set of discrete values is called a digital signal.
- In order for a signal to be processed digitally it must be discrete in time and its value must be discrete.
- If a signal to be processed is in Analogue form, it should be converted to a digital signal by:
  - Sampling the analogue signal at a discrete instants of time, obtaining a discreet time signal.
  - Quantizing it’s continuous values to a set of discrete values (approximation process)

Discrete-valued signals.



Discrete Signals

sample

Sampled Sinusoid
5sin(2πnT)

quantize

Analog sinusoid,
5sin(2πx)

Quantized sinusoid
round[5sin(2πx)]

Discrete sinusoid
round[5sin(2πnT)]

quantize

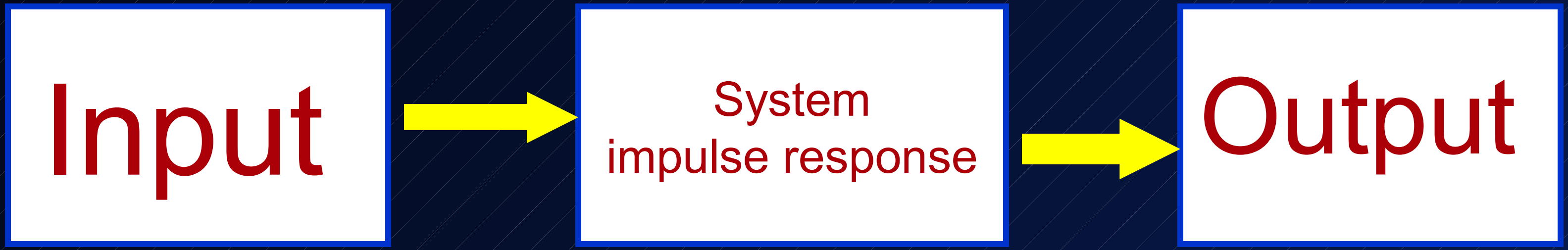
sample

What is a System?

- Systems is any process that produce an out put signal in response to input signal.

How is a System Represented?

- A system takes a signal as an input and transforms it into another signal.

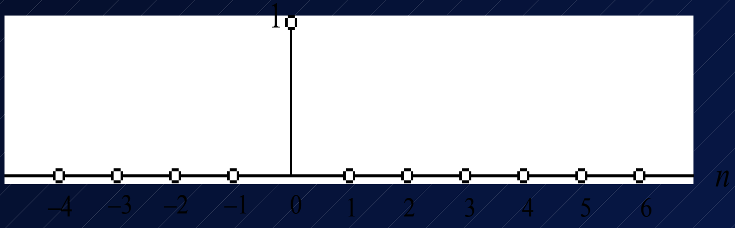


Input, Impulse Response and Output in systems

Delta function (unit impulse) is a normalized impulse, that is sample number zero has a value of one, while the other samples have a value of zero. The Greek letter tau is used to identify the delta function.

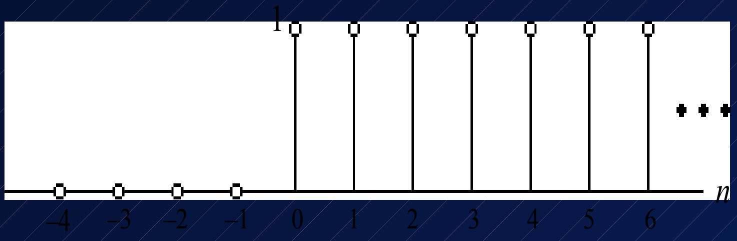
- Unit impulse for discrete.

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



- Unit step -

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



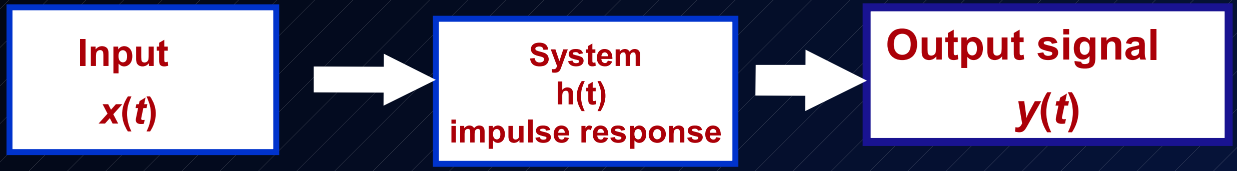
Impulse: is a signal composed of all zeros, except a single non zero points.

Impulse Response: is a signal that exists a system when the delta function (unit impulse is) the input.

Continuous & Discrete-Time Systems

Continuous-Time Systems

- Input and out put continues signal such as in analogue electronics.
  - E.g. circuit, car velocity



Input

$x(t)$

System

h(t)

impulse response

Output signal

$y(t)$

Fourier Transform (FT)

Is a mathematical technique capable of converting a time domain signal to a frequency domain signal and vice versa.

For analogue system f(t) transforming it using Fourier transform

$$f(t) \xrightarrow{FT} f(w) \quad \text{where: } w=2\pi f$$

where: e^{-jwt} is furrier kernel

$$f(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt$$

e=2.71

j= $\sqrt{-1}$

Continuous & Discrete-Time Systems

- **For any system in linear if the input is impulse □ (t) the out put of any system is the impulse response of h (t).** i.e.

$$g(t)=h(t)$$

$$g(t)=h\ (t)= h(t)\ \square\ \square(t)$$

$h(t)= \square\square(\eta)\ .\ h(t-\mu)d\mu$ **convolution**

Convolution with μ(t) results the same function:

$g(t)= \square h(\eta)\ .\ f(t-\mu)d\mu= h(t)\ \square\ f(t)$

Taking the Fourier transform both sides:

$G(w)=H(W).\ F(w)$ where $f(t) \longleftrightarrow F(w)$

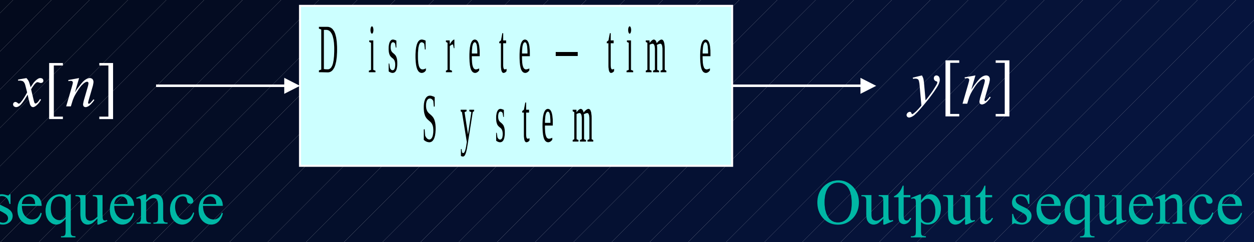
$$h(t) \longleftrightarrow H(w)$$

$$g(t) \longleftrightarrow G(w)$$

Continuous & Discrete-Time Systems

Discrete-Time Systems

A discrete-time system processes a given **input sequence** *x*[*n*] to generates an **output sequence** *y*[*n*] with more desirable properties



The behavior of a linear, time-invariant discrete-time system with input signal *x[n]* and output signal *y[n]* is described by the *convolution sum*

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

The signal *h[n]*, assumed known, is the response of the system to a unit-pulse input.

Convolution

Is a mathematical way of combining two signals to form a third signal. It is a convolution is an integral that expresses the amount of overlap of one function *g* as it is shifted over another function *f*.

Analogue convolution

for input f(t), impulse response g(t) the out put of the system is equal to the input signal convolved with the impulse response that i.e.

h(t)= f(t) ▯ g(t)

and also equal to

$$f * g \equiv \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau = \int_{-\infty}^{\infty} g(\tau)f(t-\tau)\,d\tau$$

where the symbol ▯ (occasionally also written as f g) denotes convolution of *f* and *g*. Convolution is more often taken over an infinite range,

Convolution

Convolution Sum discrete system

- The summation

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

is called the **convolution sum** of the sequences *x*[*n*] and *h*[*n*] and represented compactly as.

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

Convolution

Convolution Sum

Properties -

- Commutative property:

$$x[n]*h[n] = h[n]*x[n]$$

- Associative property :

$$(x[n]*h_1[n])*h_2[n] = x[n]*(h_1[n]*h_2[n])$$

- Distributive property :

$$x[n]*(h_1[n] + h_2[n]) = x[n]*h_1[n] + x[n]*h_2[n]$$

Digital Filter

Digital filter uses a signal processor to perform numerical calculations on sampled values of the signal.

Digital filters used for two general purposes:

3. Separation of signals that have been combined.
4. Restoration of signals that have been distorted in some way.

Advantage:

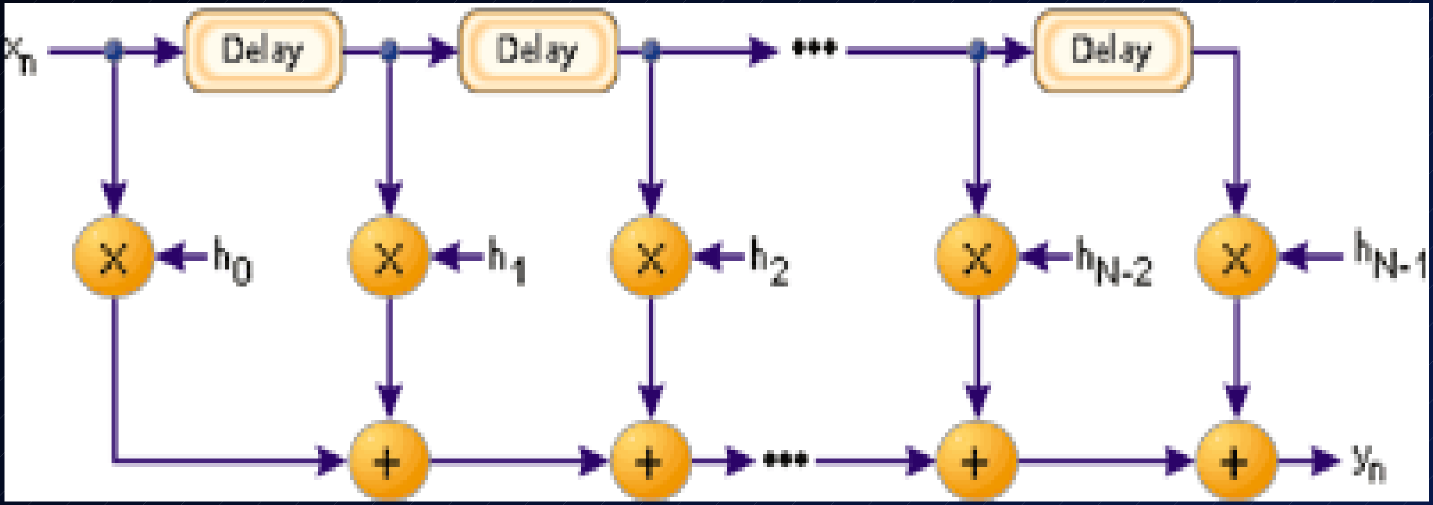
- ❖ Can handle low frequency signals accurately compared to analogue system.
- ❖ It is programmable
- ❖ Easily designed tested and implemented on a pc
- ❖ Very much versatile in their ability to process signals in a variety of ways

Limitations:

- ❖ Information may lost when analogue signals is converted to digital signal because of:
  - ❖ Inaccuracies in measurement.
  - ❖ Uncertainty in time.

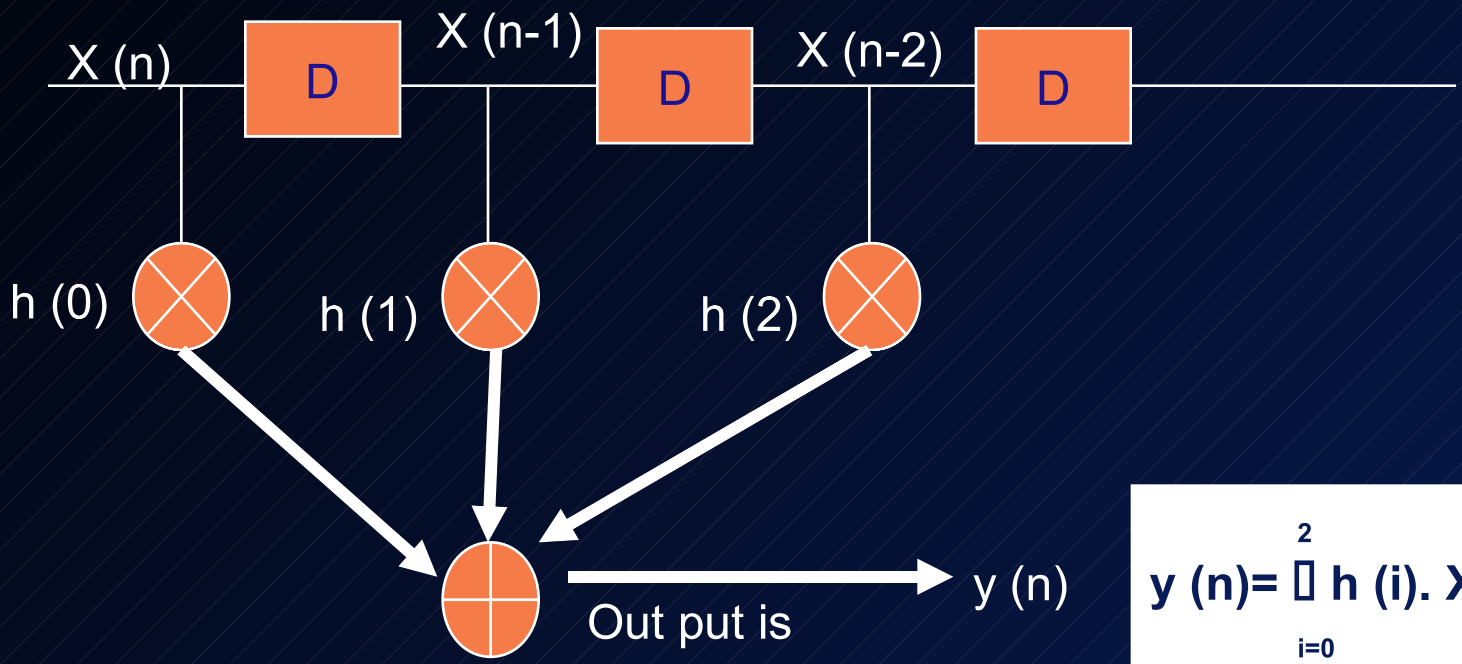
Finite Impulse Response (FIR)

FIR is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR is usually implemented by using a series of delays multipliers and adders to create the filter outputs.



Sample Finite Impulse Response Filter Design

Finite Impulse Response Filters



$$y(n)=\sum_{i=0}^{2} h(i).\,X(n-i)$$

Structure of a FIR Filter for input X (n) **impulse response** h (n) **and out put** y (n). Where :

D =Delay ⊗ =Multiplier

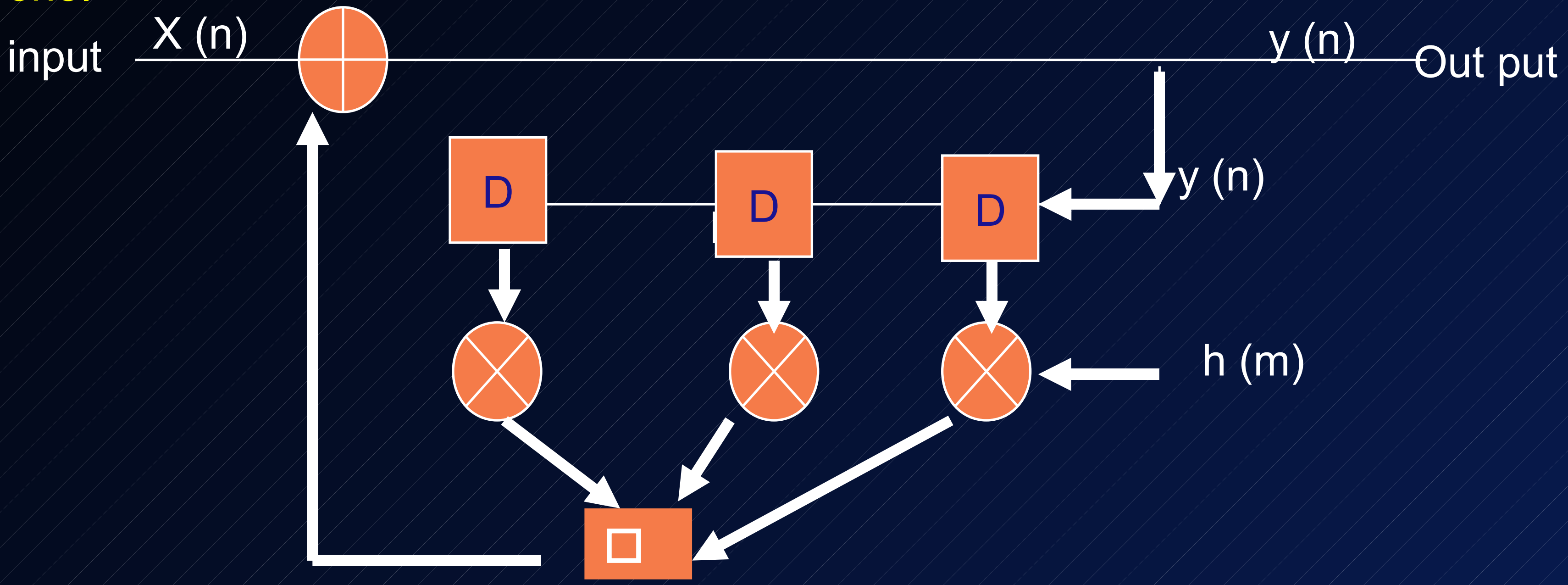
⊕ =Adder

Discrete Convolution equation for FIR filter

$$y(n)=\sum_{m=-\infty}^{+\infty} h(m).\,X(n-m)$$

Infinite Impulse Response (IIR)

The Infinite Impulse Response refers to the ability of the filter to have an infinite impulse response and does not imply that it necessarily will have one.



Structure of a IIR Filter for input X (n) **impulse response** h (1) **and out put** y (n). Where :

D =Delay

⊗ =Multiplier

⊕ =Adder

Discrete Convolution equation for IIR filter

$$y\ (n)= x\ (n)+ \sum_{m=-\infty}^{+\infty} h\ (m).\ y\ (n-m)$$

Discrete-time signals and systems (1)

Discrete-time signals

A discrete-time signal is represented as a sequence of numbers:

$$X=\{x\,[n]\}, \qquad -¥<n<¥$$

Here n is an integer, and $x[n]$ is the nth sample in the sequence.

Discrete-time signals are often obtained by sampling continuous-time signals.

In this case the nth sample of the sequence is equal to the value of the analogue

signal $x_a(t)$ at time $t=nT$:

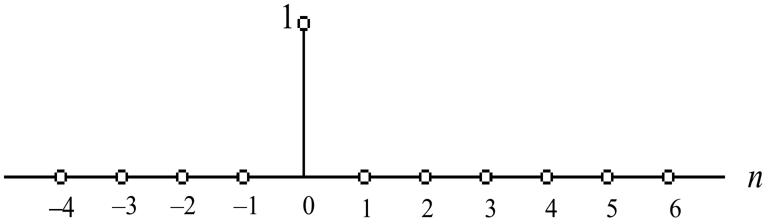
$$x[n] = x_a(nT)\ , \quad -¥<n<¥$$

Discrete-time signals and systems (2)

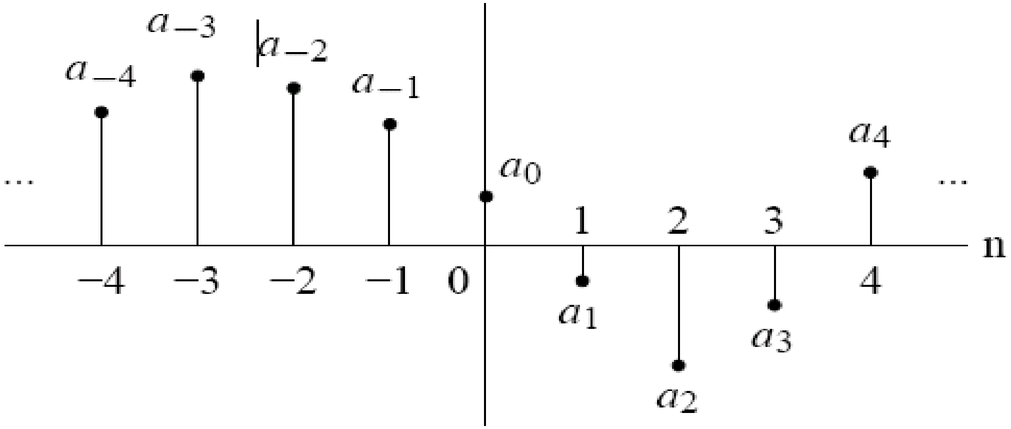
unit sample sequence

This sequence is often referred to as a **discrete-time impulse**, or just **impulse**. It plays the same role for discrete-time signals as the Dirac delta function does for continuous-time signals

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



An important aspect of the impulse sequence is that an arbitrary sequence can be represented as a sum of scaled, delayed impulses. For example, the sequence can be represented by unit sample as:



Discrete-time signals and systems (3)

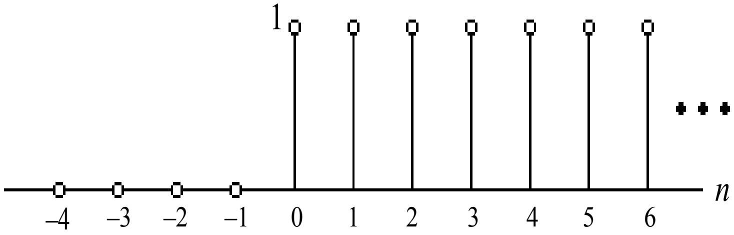
$$x[n] = a_{-4}\delta[n+4] + a_{-3}\delta[n+3] + a_{-2}\delta[n+2] + a_{-1}\delta[n+1] + a_0\delta[n]$$
$$+ a_1\delta[n-1] + a_2\delta[n-2] + a_3\delta[n-3] + a_4\delta[n-4].$$

In general, any sequence can be expressed as

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

The **unit step sequence**

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



Discrete-time signals and systems (4)

Mathematical Relationships between unit step u(n) and unit impulse function □(n).

The unit step is related to the impulse by

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

Alternatively, this can be expressed as

$$u[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \cdots = \sum_{k=0}^{\infty} \delta[n-k].$$

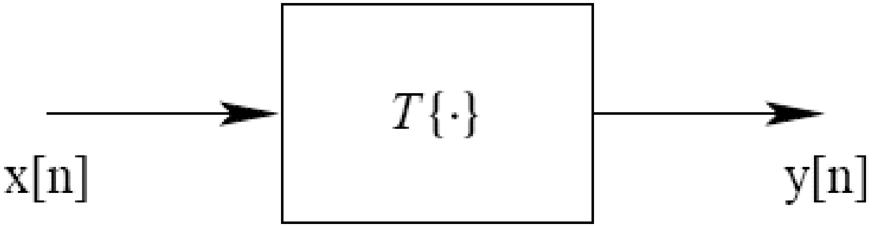
Conversely, the unit sample sequence can be expressed as the first backward difference of the unit step sequence

$$\delta[n] |= u[n] - u[n-1].$$

Discrete-time systems (1)

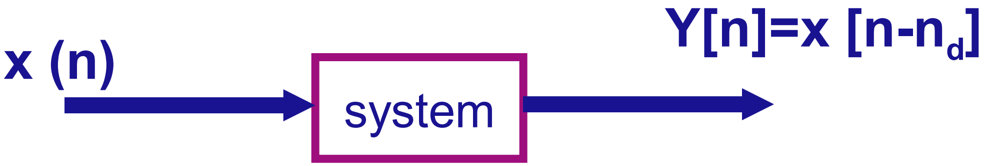
A discrete-time system is defined as a transformation or mapping operator that maps an input signal *x*[*n*] to an output signal *y*[*n*]. This can be denoted as:

$$y[n] = T\{x[n]\}.$$



Discrete-time systems (2)

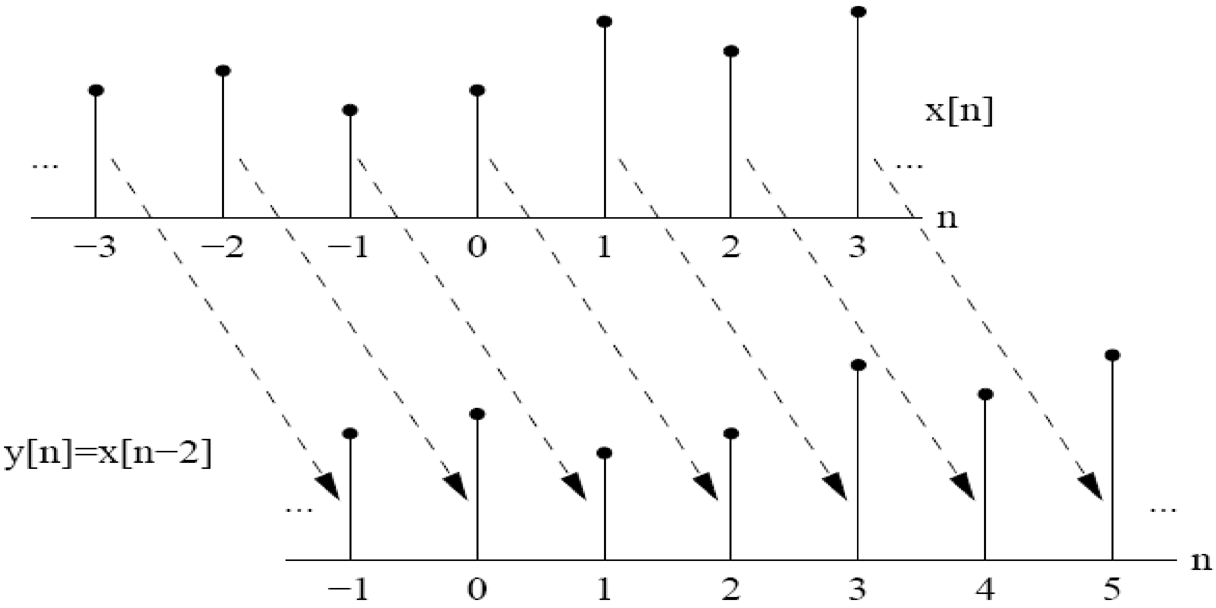
Example : Ideal delay



where:

n$_d$=number of delay

$y[n] = x[n - n_d]$:



This operation shifts input sequence later by n_d samples.

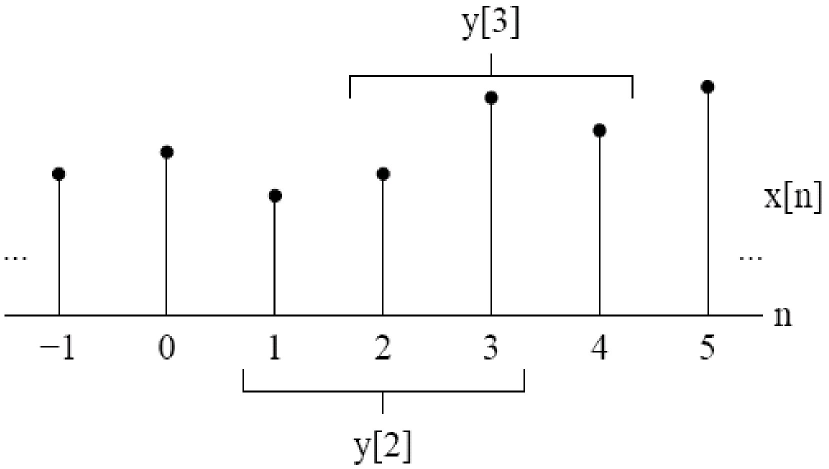
Discrete-time systems (3)

Example: Moving Average

The general moving average system is defined by equation:

$$y[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n-k]$$

For $M_1 = 1$ and $M_2 = 1$, the input sequence



Yields an out put with:

$$y[2] = \frac{1}{3}(x[1] + x[2] + x[3])$$

$$y[3] = \frac{1}{3}(x[2] + x[3] + x[4])|$$

In general, systems can be classified by placing constraints on the Transformation *T{ .}*.

Discrete-time systems (4)

1.Memoryless systems

A system is referred as memory less if the output $y[n]$ at every value of n depends only on the input value $x[n]$ at the same value of n.
For example, $y[n] =(x[n])2$ is memoryless, but the ideal delay $y[n] = x[n -d]$
Is not unless d=0.

$$\boxed{\mathbf{Y[n] =5\, x[n]\ is\ memoryless}}$$

2. Linear systems

A system is linear if the principle of superposition applies. Thus if $y1[n]$ is the response of the system to the input $x1[n]$, and $y2[n]$ the response to $x2[n]$, then linearity implies

Additivity:

$$T\{x1[n] + x2[n]\} = T\{x1[n]\} + T\{x2[n]\} = y1[n] + y2[n]$$

Homogeneity**:**

Discrete-time systems (5)

3. Time invariant system

A system is time invariant if a time shift or delay of the input sequence causes a corresponding shift in the output sequence. That is, if *y*[*n*] is the response to *x*[*n*], then *y*[*n*-d] is the response of x[n-d]

For example, the accumulator system: $y[n] = \sum_{k=-\infty}^{n} x[k]$ Is Time invariant

4. Causality

A system is causal if the output at *n* depends only on the input *at n and Earlier input.*

For example, the backward difference system $y[n] = x[n] - x[n-1]$ is causal, but the forward difference system $y[n] = x[n+1] - x[n]$ **is not causal.**

Discrete-time systems (6)

5. Stability

A system is stable if every bounded input sequence produces a bounded output sequence:

Bounded input |*x*[*n*]| ≤ *Bx* < ¥
Bounded output: |*y*[*n*]| ≤ *By* < ¥

For example, the accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$ is an example of an *unbounded* system, since its response to the unit step *u*[*n*] is:

$$y[n] = \sum_{k=-\infty}^{n} u[n] = \begin{cases} 0 & n < 0 \\ n+1 & n \geq 0, \end{cases}$$

which has no finite upper bound.

Discrete-time systems (7)

6. Accumulator

The system cumulatively adds, i.e., it accumulates all the input sample values.

If the input is □ (n) for the system the out put is u(n)

□ (n)

Accumulator

u(n)

$$y(n)= \underset{k=-¥}{\overset{n}{\square}} x(k)$$

$$\mathbf{u(n)}= \underset{k=-¥}{\overset{n}{\square}} \square(k)$$

Linear time-invariant systems (LTI)

If the linearity property is combined with the representation of a general sequence as a linear combination of delayed impulses, then it follows that a linear time-invariant (LTI) system can be completely characterized by its impulse response.

Suppose *h* [*n*] is the response of a linear system to the impulse δ[*n*-k] at n=k. Since

$$y[n] = T\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\}$$

If the system is additionally time invariant, then the response to δ[*n*-k] is h[n-k] the above equation becomes:

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

This expression is called the **convolution sum**. Therefore, a LTI system has the property that given *h*[*n*], we can find *y*[*n*] for *any* input *x*[*n*]. Alternatively, *y*[*n*] is the **convolution** of *x*[*n*] with *h*[*n*], denoted as follows:

$$y[n] = x[n] * h[n].$$

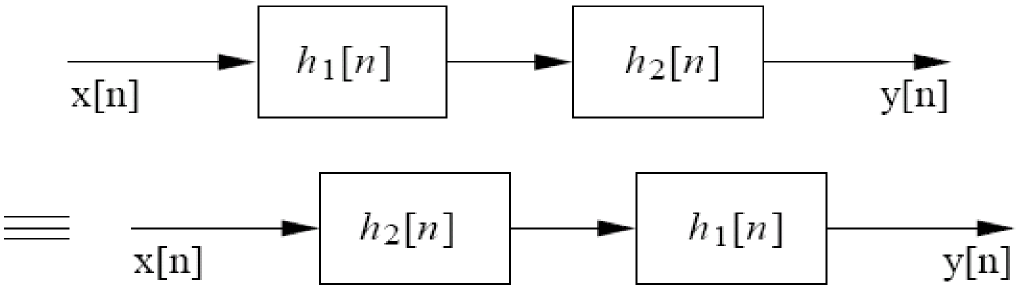
Properties of LTI systems (1)

All LTI systems are described by the convolution sum

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

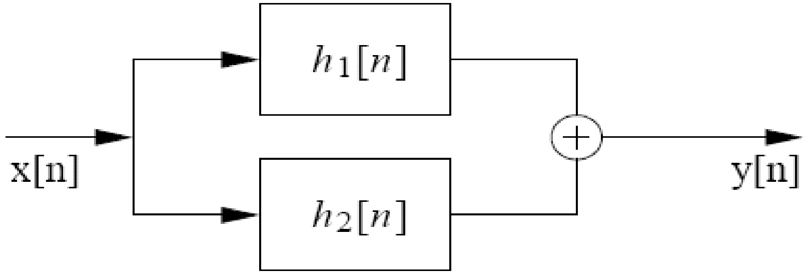
Some properties of LTI systems can therefore be found by considering the properties of the convolution operation: like commutative, cascade and parallel connection

Cascade connection:



$$y[n] = h[n] * x[n] = h_1[n] * h_2[n] * x[n] = h_2[n] * h_1[n] * x[n].$$

Parallel connection:



Properties of LTI systems (2)

A LTI system is **stable** if and only if *S*= □k-¥ | h[k] | < ¥

The Ideal delay system *h*[*n*] =□ [n-d] is stable since S=1< ¥

The **moving average** system:

$$h[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} \delta[n-k]$$

$$= \begin{cases} \frac{1}{M_1+M_2+1} & -M_1 \leq n \leq M_2 \\ 0 & \text{otherwise,} \end{cases}$$

the **Forward difference** system *h*[*n*] = □[*n*+1] - □[n] and the **Backward difference** system *h*[*n*] = □[*n*] - □[*n*-1] are stable since *S* is the sum of a finite number of finite samples, and is therefore less than ¥.

But the **Accumulator** System is unstable.

Difference Equation

The difference between the input and out put for DSP can be calculated using Deference equation unlike differential in Analogue.

1. Ideal delay

$h(n)=\square(n-m)$

2. Moving average

$$h[n]=\frac{1}{M_1+M_2+1}\sum_{k=-M_1}^{M_2}\delta[n-k]$$

$$=\begin{cases}\frac{1}{M_1+M_2+1} & -M_1\le n\le M_2\\ 0 & \text{otherwise,}\end{cases}$$

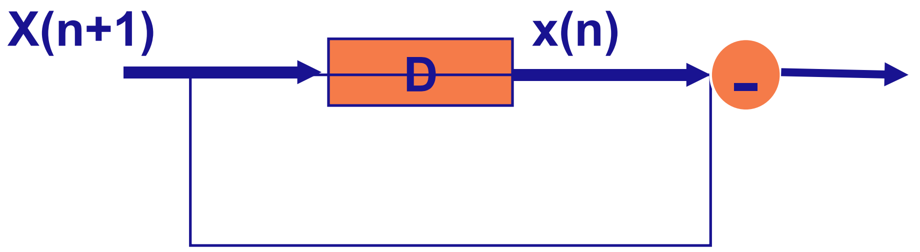
3. Accumulator

$$h(n)=\overset{n}{\underset{k=-\text{¥}}{\square}}\ \square(k)\ =\begin{cases}1\ ,\ n\ge 0\\ 0\ ,\ n<0\end{cases}$$

Or $h(n)=u(n)$

Difference Equation

4. Forward difference

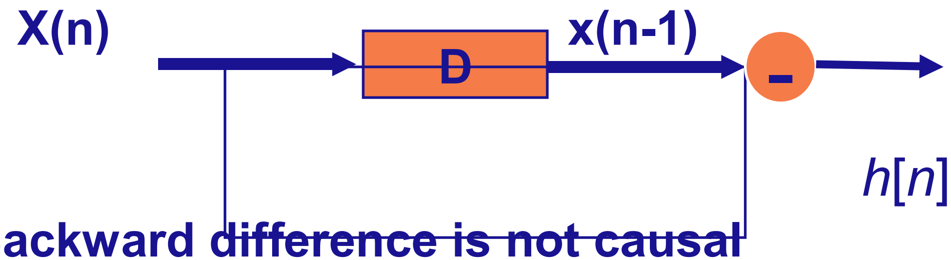


y(n)= x(n+1)-x(n)

$h[n]$ = □[n+1] - □[n]

Forward difference is not causal but it is Stable.

5. Backward difference system $h[n]$ = □[n] - □[n-1] are stable since S.



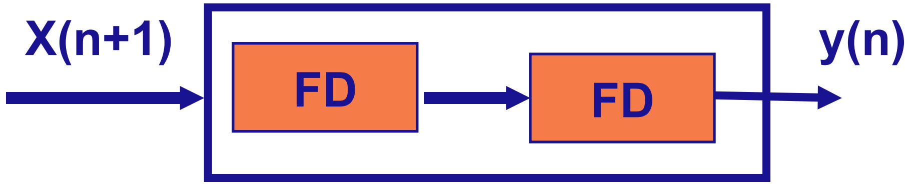
y(n)= x(n)-x(n-1)

$h[n]$ = □[n] - □[n-1]

Backward difference is not causal

Difference Equation

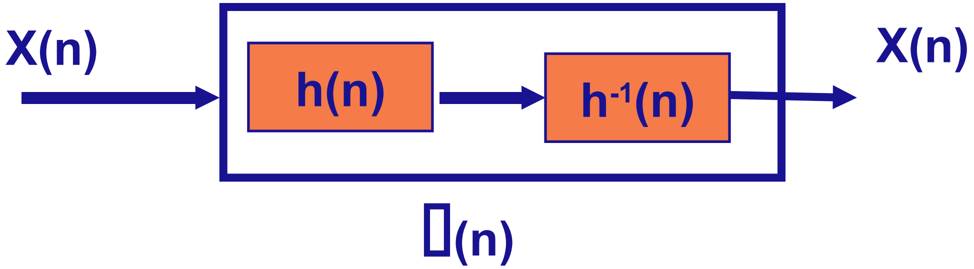
A Forward difference Followed by ideal delay is equal to backward difference.



$$h[n] = [\,\square[n+1] - \square[n]\,] \;\square\; \square[n-1]$$

$$\boxed{\boldsymbol{h[n] = \square[n] - \square[n-1]}}$$

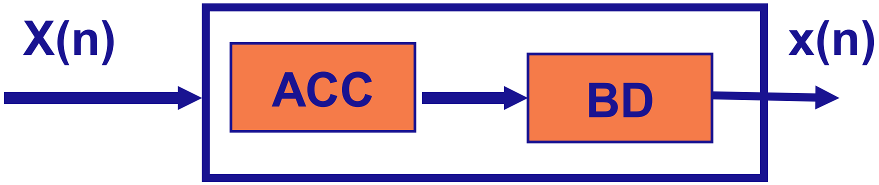
Inverse System



Difference Equation

Example:

Backward difference is the inverse of Accumulator.



$h[n] = u(n) * \left[\, \delta[n] - \delta[n-1] \right]$ and equals to

$$u[n] - u[n-1] = \delta[n]$$

The inverse of any system is very important in all inverse engineering. For example for TV channel broadcast you don’t know, and if you take it as input And inverse it by backward difference, then you will get The input, because the out put is the input.

Frequency Domain Representation Of Discreet Time Signal systems.

General Fourier Equation for frequency representation of sample or signal is:

$$H(e^{jw}) = \sum_{k=-\infty}^{+\infty} h(k)e^{-jwk}$$

Where e is:

$$e^{-jwk} = \cos wk - j\sin wk$$

$$H(e^{jw}) = \frac{\sin(\frac{wn}{2})e^{j(N-1)\frac{w}{2}}}{\sin(\frac{w}{2})}$$

Frequency Domain Representation Of Discreet Time Signal systems.

The Inverse of frequency representation of Fourier transform

$$h(n) = \frac{1}{2}\int_{-\pi}^{\pi} H(e^{jw})e^{jwn}\, dw$$

If we know the Impulse response we can use the above equation to design filter using ideal low pass filter where the h(n) is:

$$h(n) = \frac{1}{2}\pi \int_{-w_{co}}^{w_{co}} e^{jwn} dw$$

- ❖Ideal low pass is not casual and stable
- ❖It is possible to design filter using low pass. But we need an infinite taps. And if the number of delay tap) is many it is possible to make it sharp and finite.

General definition for even and odd sequence.

Even sequence:

$X_e(n)=X_e(-n)$

Odd sequence:

$X_o(n)=-X_0(-n)$

1. For any sequence in time domain

$$x(n) = x_e(n) + x_o(n)$$
$$x(-n) = x_e(n) - x_o(n)$$

$$\begin{cases} x_e(n) = \frac{1}{2}\left[x(n) + x^*(-n)\right] \\ x_o(n) = \frac{1}{2}\left[x(n) - x^*(-n)\right] \end{cases}$$

Even sequence:

Odd sequence:

General definition for even and odd sequence.

1. For Fourier transform in frequency domain

$$X(e^{jw}) = X_e(e^{jw}) + (e^{jw})$$

Even Part is: $x_e(e^{jw}) = \frac{1}{2}\left[x(e^{jw}) + x^*(e^{-jw})\right]$

Odd Part is: $x_o(e^{jw}) = \frac{1}{2}\left[x(e^{jw}) - x^*(e^{-jw})\right]$

$$\left.\begin{array}{ll} \text{Even} & X_e(e^{jw}) = X_e{}^*(e^{-jw}) \\ \text{Odd} & X_o(e^{jw}) = -X_e{}^*(e^{-jw}) \end{array}\right\}$$

Changing jw to -wj is the same because of complex conjugate

Relationship Between Time Domain and Frequency Domain.

where the real part Re of the signal is even part of the sequence

$$F\left[R_e\left[x[n]\right]\right] = \frac{1}{2}\left[x(e^{-jw}) + *x(e^{-jw)}) = x_e(e^{jw})\right]$$

where: R_e =Real part

F =Fourier transform

Transforming Even Part by Fourier Transform is equal to Real part

$$F_1\left[X_e(n)\right] = \frac{1}{2}\left[x(e^{jw}) + x*(e^{jw})\right]$$

$$= R_e\left[x(e^{jw})\right]$$

$$X*(n) \xrightarrow{F_1} X*(e^{-jw})$$

$$X*(-n) \xrightarrow{F_1} X*(e^{jw})$$

The Real part Sequence transformed by Fourier transform equal to even part

$$R_e\left[x(n)\right] \xrightarrow{F_1} X_e(e^{jw})$$

Properties of Fourier Transform In Discrete

Linearity

$$A_1x_1(n)+a_2x_2(n) \xrightarrow{F_1} a_1x_1(e^{jw})+a_2x_2(e^{jw})$$

Time Shift and Frequency Shift

$$\boxed{x\,(n-D)= \xrightarrow{F_1} e^{-jwnD}.x(e^{jw})}$$ → Time Shift

For a shift in time the out put is the multiplication in frequency.

$$e^{jwn}.x(n) \xrightarrow{F_1} X(e^{j(w-w_0)}$$ → Frequency Shift

Frequency shift is important in communication engineering because shift in frequency does not make any overlap.

Properties of Fourier Transform In Discrete

Time Reversal

$X(-n) \xrightarrow{F_1} X(e^{-jw})$

X(n) is real signal: $X(-n) \xrightarrow{F_1} X^*(e^{jw})$

Differentiation in Frequency Domain

$$n.x(n) \xrightarrow{F_1} \frac{j^{\cdot d(x(e^{jw}))}}{d\omega}$$

Convolution

$$y(n) = \sum x(k)h(n-k) = x(n) \otimes h(n)$$ Time Domain.

$$y(e^{jw}) = X(e^{jw}).H(e^{jw})$$ Frequency Domain

Properties of Fourier Transform In Discrete

Parseval's Theory

$$E = \sum_{-\infty}^{+\infty} \left| x(n) \right|^{2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| x(e^{-jw}) \right|^{2} d\omega$$

Where E=energy

The Modulation or Windowing Domain

$$y(n) = w(n).x(n)$$

Multiplication in Time Domain but convolution in frequency domain

$$y(e^{jw}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} x(e^{jw}).w(e^{j(w-\theta)})\ d\theta$$

$$h_1 \otimes h_2 \longrightarrow H_1.H_2$$

Discrete convolution

$$h_1.h_2 \longrightarrow H_1 \otimes H_2$$

Integration convolution

Chapter- 4
Z-Transform

Defn.

For time sequence x(n) the discrete Z-transform is defined as:

$$x(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}$$

Where Z is complex and can be found any where

❖Z is a complex number, but in Fourier transform e^{jw} is represented in unit circle. And it is similar to laplace transform which is analogue.

❖Z is infinite but it is possible to make it convergent.

Z-Transform

Example 1

For sequence x(n)=a^{n} where n≤0 transformed by Z-transform

$$x(z) = \sum_{n=0}^{\infty} a^{n} z^{-n}$$ Geometric Series

$$x(z) = \sum_{n=0}^{\infty} (az^{-1})^{n} = 1 + az^{-1} + (az^{-1})^{2} + \ldots$$

if $\left|az^{-1}\right| < 1,$ **then** $x[z] = \dfrac{1}{1 - az^{-1}}$ **where**

$|z| < |a|$ **The condition for convergence**

Z-Transform

Example 2

For sequence x(n)=a$^{|n|}$ where -∞<n<n transformed by Z-transform

$$x(z)=\sum_{n=\infty}^{-1} a^{-n}z^{-n}+\sum_{n=0}^{\infty} a^{n}z^{n}$$ **Making a summation into parts**

$$=\sum_{n=1}^{\infty} a^{n}z^{n}+\sum_{n=0}^{\infty} a^{n}z^{-n}$$

The left part $\sum_{n=1}^{\infty} a^{n}z^{n}$ **is convergent when:** $\begin{cases} |az|<1 \\ |z|<\dfrac{1}{|a|} \end{cases}$

The right part $\sum_{n=0}^{\infty} a^{n}z^{-n}$ **is convergent when:** $\begin{cases} |az^{-1}|<1 \\ |z|>|a| \end{cases}$

Properties of Z-Transform

Linearity Let x(n) and y(n) be any two functions and let X(z) and Y(z) be their respective transforms. Then for any consonants a and b

$$ax(n) + by(n) \xrightarrow{Z} ax(z) + by(z)$$

Shifting

$$x(n + k) \xrightarrow{Z} z^{k} x(z)$$

convolution If w(n)=x(n)*y(n) then

$$w(z) = X(z)Y(z)$$

$$y(n) = \sum h(k)x(n - k) \xrightarrow{Z} y(z) = H(z).X(z)$$

Properties of Z-Transform

Relationships between Fourier transform and Z-transform

$$x(z) = \sum x(n) z^{-n}$$

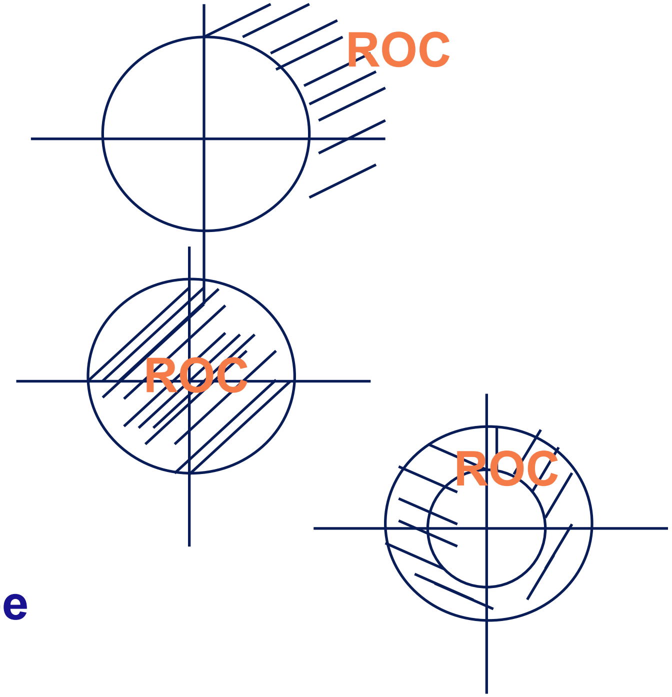
$$x(e^{jw}) = \sum x(n) e^{-jwn} \qquad Z = re^{jw}$$

Range of convergence (ROC) for Z-transform

If the sequence is right sided then ROC is:

If the sequence is left sided then ROC is:

If the sequence is both sided then the pole is outside or inside the ring. It is Like donat shape



Inverse of Z-Transform

Method 1

$$x(n) = \frac{1}{2\pi j} \oint_c x(z).z^n \frac{dz}{z}$$

Method 2

$$x(z) = \frac{1}{1 - 0.5z^{-1}}$$

Z_p=0.5 that means we have the pole inside circle

Inverse of Z-Transform

Inspection method

$$a^n a(n) \xrightarrow{Z} \frac{1}{1-az^{-1}} \qquad |z| > |a|$$

Right Sided

$$-a^n a(-n-1) \xrightarrow{Z} \frac{1}{1-az^{-1}} \qquad |z| < |a|$$

Left Sided

Partial fraction Expansion method

$$x(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \qquad x(z) = \frac{b_0 \prod_{k=1}^{M} (1-(kz^{-1})}{a_0 \prod_{k=1}^{N} (1-(d_k z^{-1})}$$

$$x(z) = \sum_{k=1}^{N} \frac{A_k}{1-d_k z^{-1}}$$

For M<N where

$$A_k = (1-d_k z^{-1})x(z)$$

Inverse of Z-Transform

Power series Expansion

$$x(z) = \sum x(n) z^{-n} = + x(-2) z^{-2} + x(-1) z^{-1} + x(0) + x(1) z^{-1} \ldots$$

If you have $x(z) = \log(1 + a z^{-1})$ Which is the result of function or series

$$\log(1 + x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} \, x(n)}{n} = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} a^n z^{-n}}{n}$$

$$x(n) = (-1)^{n+1} \frac{a^n}{n}$$ For $n \geq 1$

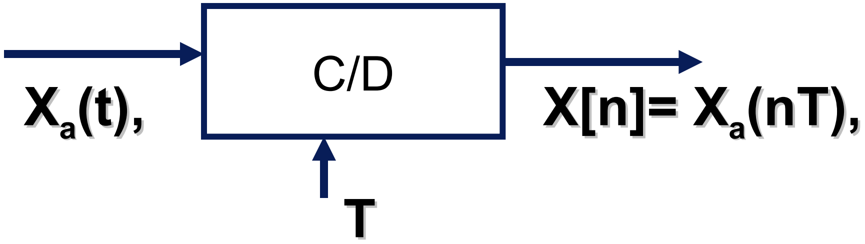
This was the function which is the result of the above function

Chapter 3
Sampling of Continuous-Time Signals

Sampling:

To use digital signal processing methods on analogue signal, it is necessary to represent the signal as a sequence of numbers. this is commonly done by sampling the analogue signal, denoted $X_a(t)$, periodically to produce the sequence

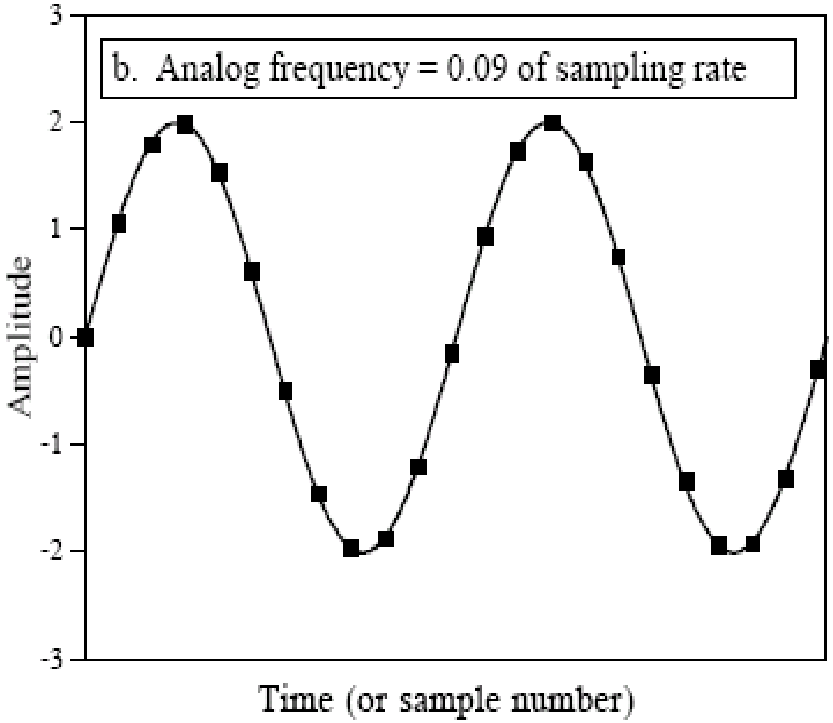
$X[n]= X_a(nT)$, $-\infty<n<\infty$ where: T=sampling period.



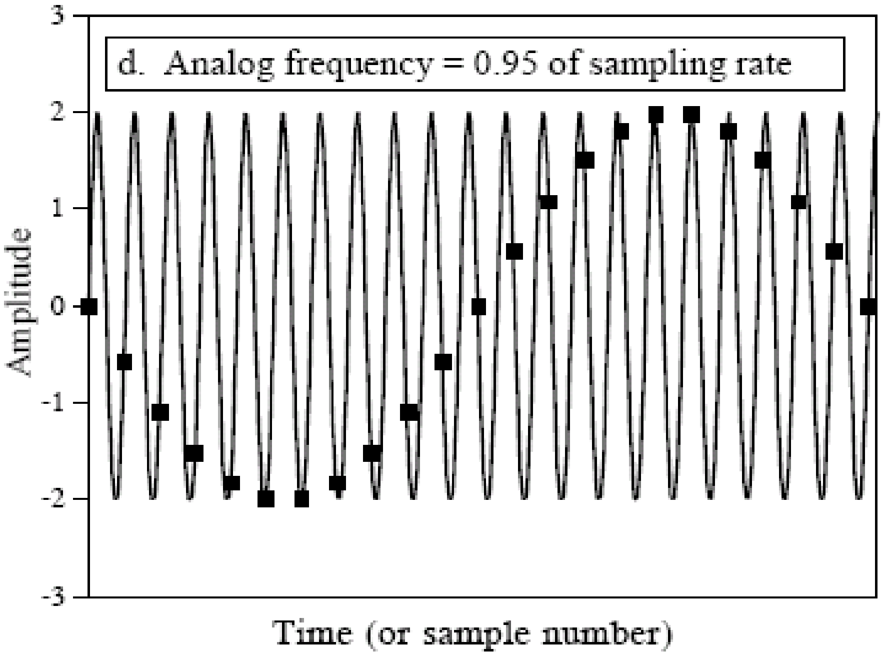
Block diagram representation of an ideal continuous-to-discrete (C/D) converter.

Two examples for sampling

good



irreversible



Sampling of Continuous-Time Signals

Fourier transform in analogue :

$$\left\{\begin{array}{l} x_a(i\Omega) = \displaystyle\int_{-\infty}^{+\infty} x_a(t)e^{-j\Omega t}\,dt \\[2ex] x_a(t) = \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{+\infty} x_a(j\Omega)e^{j\Omega t}\,d\Omega \end{array}\right.$$

Where:Ω =frequency

$$\left\{\begin{array}{l} x(n) = x_a(nT) = \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{+\infty} x_a(j\Omega)e^{j\Omega nT}\,d\Omega \\[2ex] x(n) = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{+\pi} x\,(e^{j\omega})e^{j\omega n}\,d\omega \end{array}\right.$$

→ Not periodic

→ periodic (between -Π to Π)

Sampling of Continuous-Time Signals

<u>Sampling Theorem:</u> If a signal $X_a(t)$ has a band limited Fourier transform $X_a(j\ \ )$, such that $X_a(j^{\Omega})=0$ for $^{\Omega}$ ≥2 ΠF_N, then $X_a(t)$ can be uniquely constructed from equally sampled spaces
$X_a(t)$, -∞<n< ∞ if 1/ΩT >2F$_N$.

If the Fourier transform of Not periodic $X_a(t)$ is defined as:

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

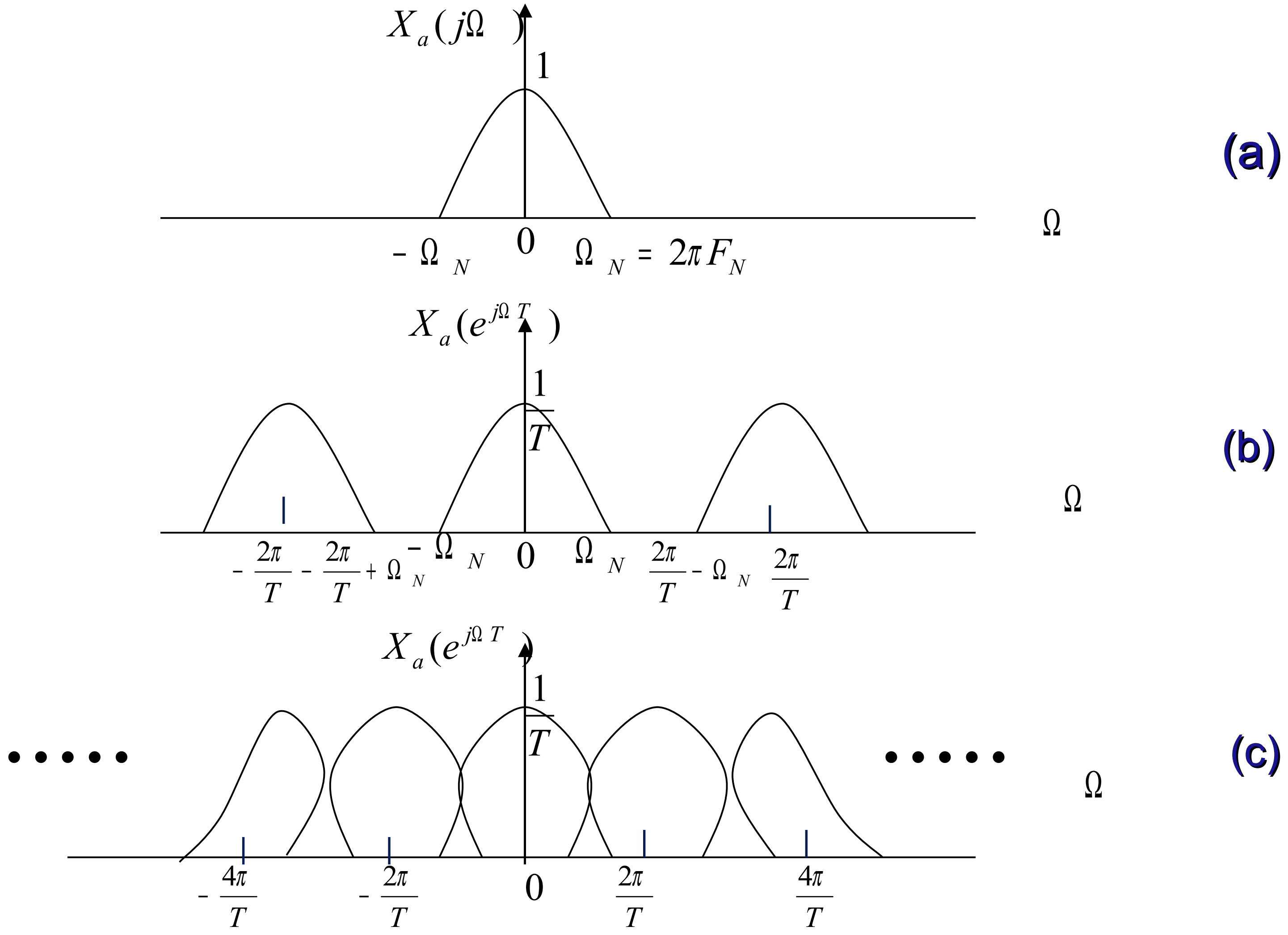
then If $(e^{j\omega})$ is evaluated for frequency $\omega = \Omega T$ then, $X(e^{j\Omega T})$ is related to $X_a(j\Omega)$ by:

$$X(e^{j\Omega t}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(j\Omega + j\frac{2\pi}{T}k)$$

where: K=integer

Recovery of Analogue Signal From Sampling (1)

Illustration of Sampling:



Recovery of Analogue Signal From Sampling (2)

In the above figures:

Figure (a) assume that $X_a(j\Omega) = 0\_for\_|\Omega| > \Omega_N = 2\pi F_N$ the Frequency F_N is called **Niguist Frequency.**

Figure (b) depicts the case when 1/T>2F_N so that the image of transform don't overlap into the base band $|\Omega| < 2\pi F_N$

Figure (c) on the other hand shows the case 1/T>2$F_{N.}$ In this case the image centered at 2 Π/T overlaps into the base band. This condition, where a high frequency seemingly takes on the identity of the lower frequency is called aliasing.

Aliasing can be avoided only if the Fourier transform is band limited and The sampling frequency (1/T) is equal to at least twice the Niguist Frequency

(1/T>2F_N)

Recovery of Analogue Signal From Sampling (3)

Under the condition 1/T > 2F$_N$ the Fourier transform of the sequence of sample is proportional to the analogue signal in the base band; i.e.,

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega)$$

Using this result, it can be shown that the original signal can be related to the sequence of samples by **Interpolation** Formula

$$|\Omega| < \frac{\pi}{T}$$

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

Interpolation Formula

If the samples of a band limited analogue signal taken at a rate of at least twice the Niguist Frequency, it is possible to reconstruct the orignal analogue signal using the above equation.

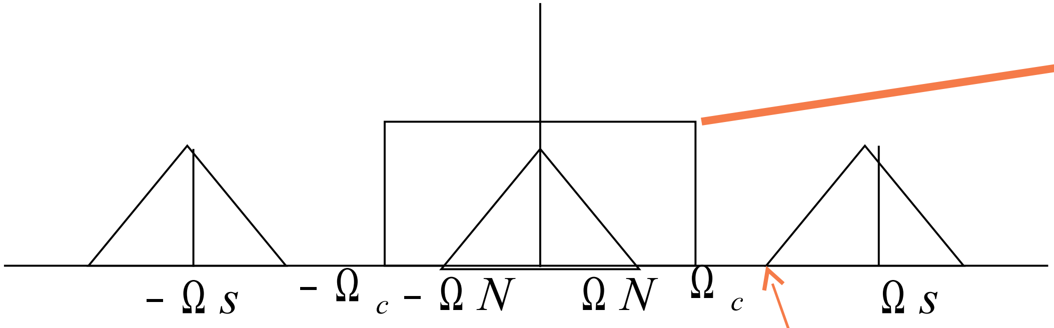
Recovery of Analogue Signal From Sampling (4)

General formula for **Interpolation.**

$$X_a(t) = \sum_{k=-\infty}^{\infty} C_K \phi_k(t)$$

Where: C_k = **Sampling Value**

ϕ_k = **Sink**



Low Pass Filter Band width

Where: Ω_c **is Cutoff frequency**

$\Omega s - \Omega N$

If $\Omega_N < \Omega_c < \Omega_s - \Omega_N$ **then it is possible to recover. If the above condition is not set, then aliasing will be produced.**

Recovery of Analogue Signal From Sampling (5)

order to avoid aliasing (distortion)

$$\Omega_s - \Omega_N > \Omega_N$$ **Condition for anti aliasing**

$$\Omega s > 2\Omega_N$$ **Nyquist Frequency Rate**

Since $\Omega_s = \frac{2\pi}{T}$ **we can write the above equation as:**

$\Omega_N < \frac{\pi}{T}$ **or** $F_N < \frac{F_s}{2}$ **or** $\Omega_s > 2\Omega_N$

Recovery of Analogue Signal From Sampling (5)

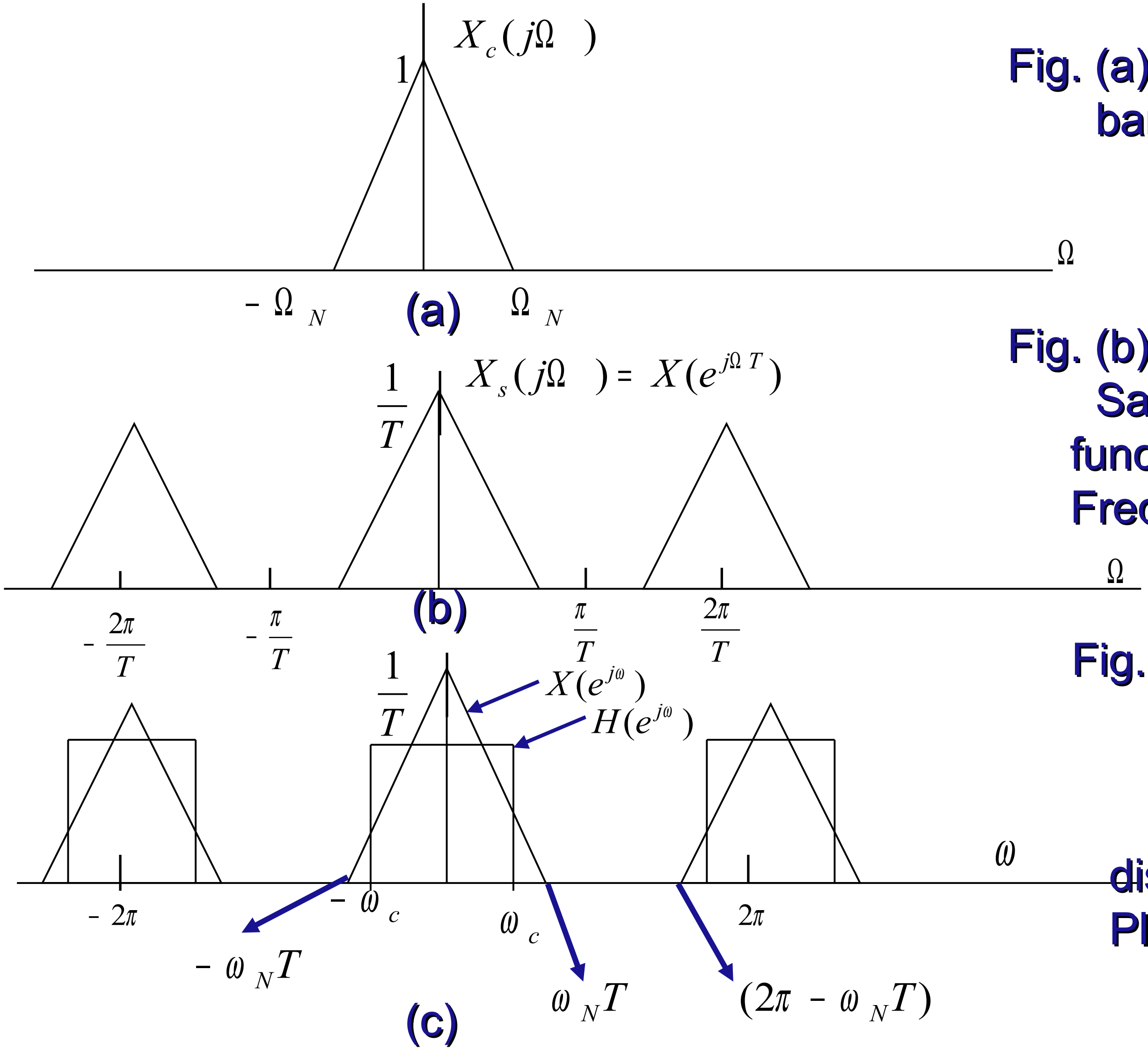


Fig. (a) Fourier transform of band limited input signal

Fig. (b) Fourier transform of Sampled input Plotted as function of continuous-time Frequency Ω

Fig. (c) Fourier transform $X(e^{j\omega})$ of sequence Samples and Frequency Response $H(e^{j\omega})$ of discrete time system Plotted Vs. ω

Changing The Sampling Rate Using Discrete-Time Processing

Sampling Rate Reduction by an Integer Factor

Decimation/Downsampling/Compressor

The process of sampling rate reduction is called Decimation.

The sampling rate of a sequence can be reduced by "sampling it, i.e., by defining a new sequence

$$x_d[n] = x[nM] = x_c(nMT)$$

can be obtained directly from $x_c(t)$ by sampling with period $T' = MT$ .Furthermore, if $x_c(j\Omega) = 0\_for\_|\Omega| > \Omega_N$ then $x_d[n]$

Is an exact representation of $x_c(t)$ if $\pi / T' = \pi / (MT) > \Omega_N$

That is, the sampling rate can be reduced by a factor of M with out aliasing

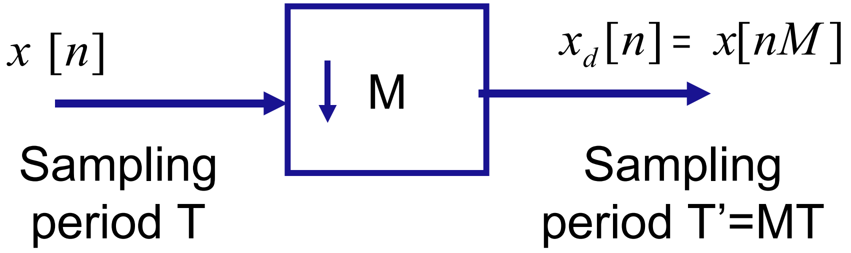
If the original sampling rate was at least M times the Nyquist rate.

The operation of reducing the sampling rate is called **downsampling or decimation.**

Changing The Sampling Rate Using Discrete-Time Processing

Sampling Rate Reduction by an Integer Factor

Representation of Downsampler or discrete-time sampler



$x\,[n]$

$\downarrow$ M

$x_d[n] = x[nM]$

Sampling period T

Sampling period T'=MT

We discussed that the discreet-time Furrier Transform of $x[n] = x_c(nT)$ **is**

$$x\,(e^{jw}) = \frac{1}{T}\sum_{k=-\infty}^{+\infty} x_c\left(j\frac{\omega}{T} - j\frac{2\pi k}{T}\right)$$

Changing The Sampling Rate Using Discrete-Time Processing

Sampling Rate Reduction by an Integer Factor

Similarly, the discreet-time Furrier Transform of

$x_d[n] = x[nM] = x_c(nMT)$ **With** T'=MT is

$$x_d(e^{jw}) = \frac{1}{T'}\sum_{k=-\infty}^{+\infty} x_c\left(j\frac{\omega}{T'} - j\frac{2\pi k}{T'}\right)$$

Now Since T'=MT , we can write the above Equation as

$$x_d(e^{jw}) = \frac{1}{MT}\sum_{k=-\infty}^{+\infty} x_c\left(j\frac{\omega}{MT} - j\frac{2\pi r}{T}\right)$$

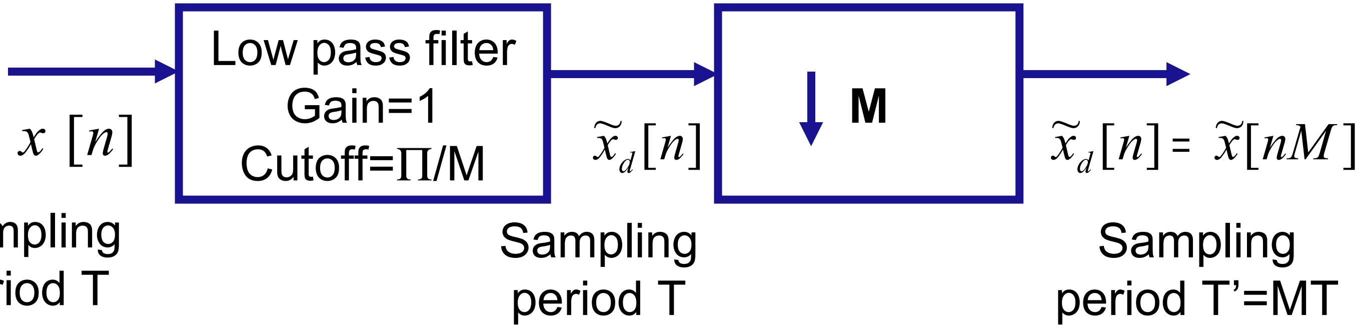
Changing The Sampling Rate Using Discrete-Time Processing

Sampling Rate Reduction by an Integer Factor

Equation for Decimation in Frequency Domain when r in the above equation is expressed as r =i + KM

$$x_d(e^{jw}) = \frac{1}{M}\sum_{i=0}^{M-1} x\ (e^{j(\omega/M - 2\pi i/M)})$$

General System for Sampling rate reduction by M



$x\ [n]$

Sampling period T

Low pass filter
Gain=1
Cutoff=Π/M

$\tilde{x}_d[n]$

Sampling period T

$\downarrow$ M

$\tilde{x}_d[n] = \tilde{x}[nM]$

Sampling period T’=MT

Changing The Sampling Rate Using Discrete-Time Processing

Frequency-domain illustration of downsampling

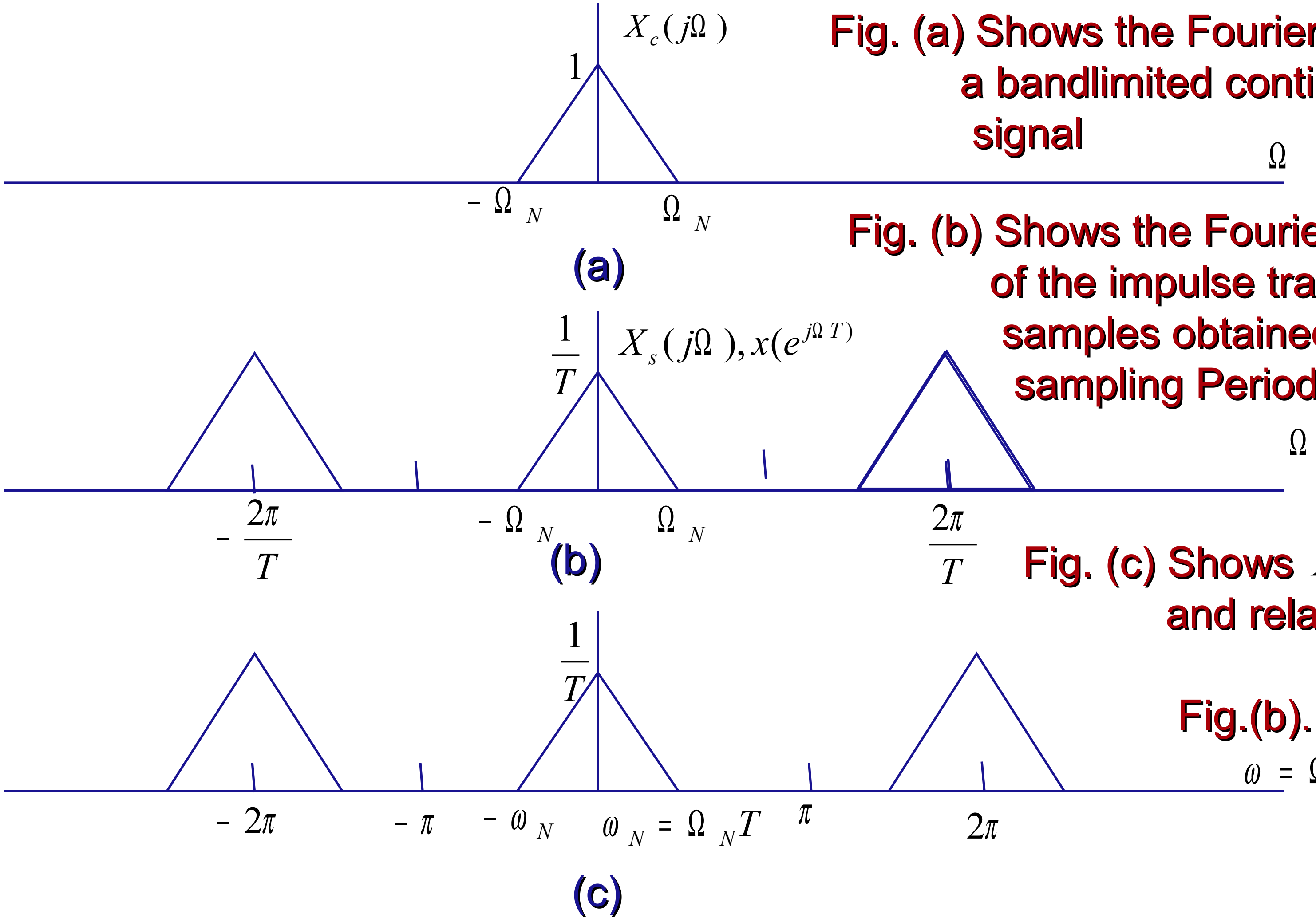


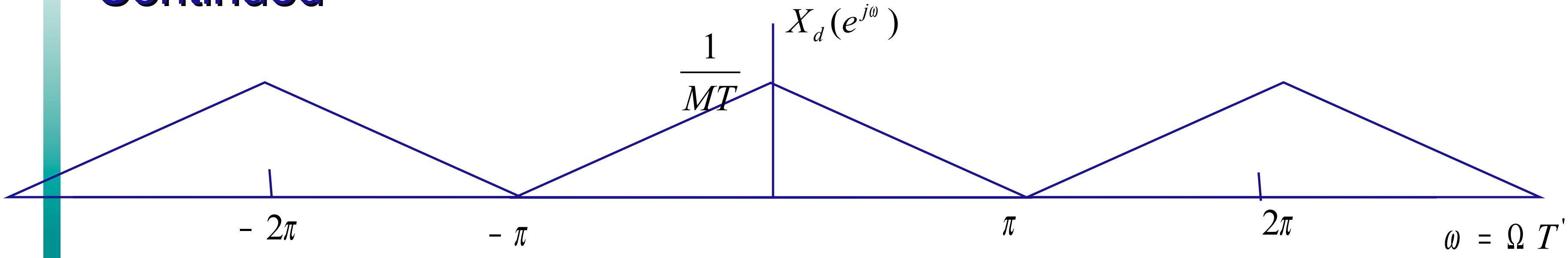
Fig. (a) Shows the Fourier transform a bandlimited continuous time signal

Fig. (b) Shows the Fourier transform of the impulse train of samples obtained with sampling Period T.

Fig. (c) Shows $X(e^{j\omega})$ and related to Fig.(b).

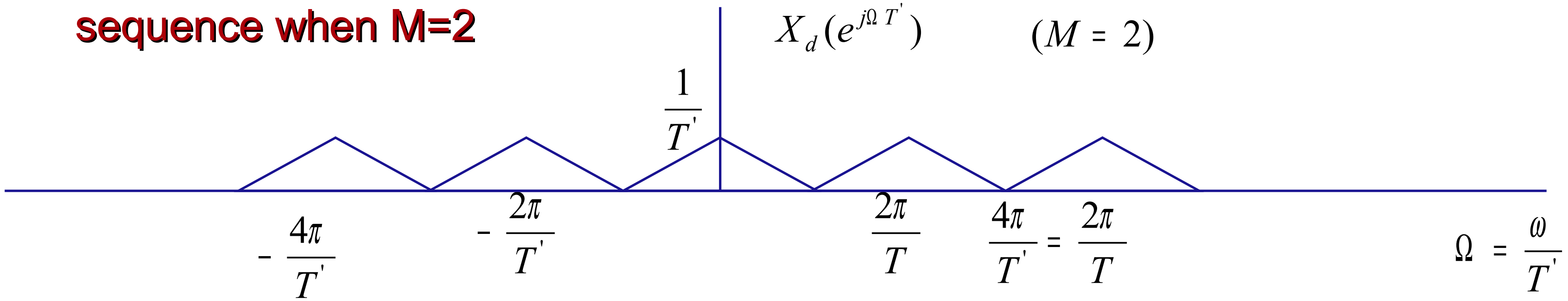
Changing The Sampling Rate Using Discrete-Time Processing

Continued



(D)

Fig. (d) Shows the discrete-time Fourier transform of downsampled sequence when M=2



(e)

Fig. (e) Shows the discrete-time Fourier transform of the downsampled sequence plotted as a function of the continuous time frequency variable Ω

Changing The Sampling Rate Using Discrete-Time Processing

Increasing the Sampling Rate by an Integer Factor

Interpolation

Increasing the sampling rate involves operations analogous to D/C Conversion. To see this consider a signal $x[n]$ whose sampling rate wish to increase by an integer factor of L. If we consider the underlying Continuous-time signal $x_c(t)$, the objective is to obtain samples

$$x_i(n) = x_c(nT')$$

Where $T' = \frac{T}{L}$, from the sequence of samples

$$x[n] = x_c(nT)$$

The operation of increasing the sampling rate is referred as

Upsampling.

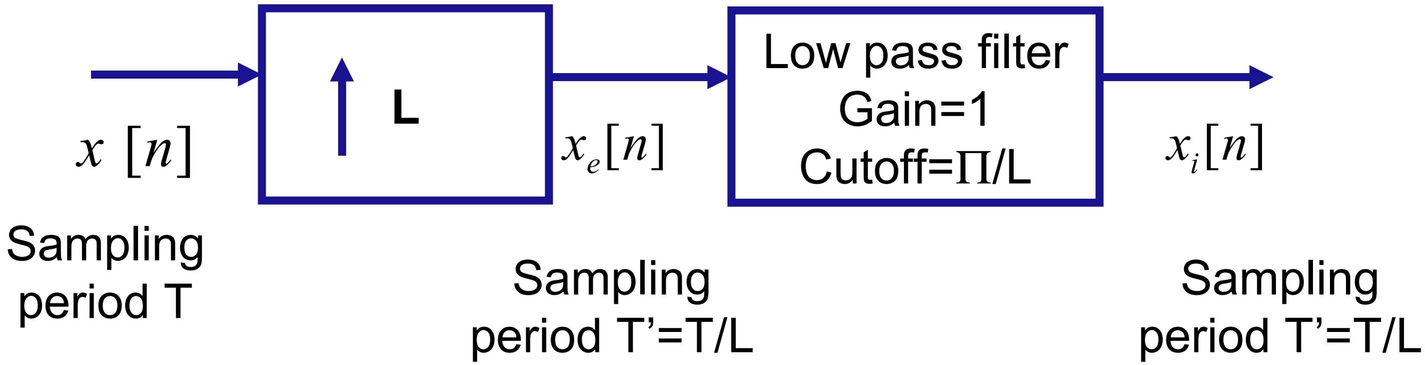
It is clear from the above two equation that

$$x_i[n] = x[n/L] = x_c(nT/L), \qquad n = 0, \pm L, \pm 2L, \ldots.$$

Changing The Sampling Rate Using Discrete-Time Processing

Increasing the Sampling Rate by an Integer Factor

General System for Sampling rate Increased L



The above figure show a system for obtaining $x_i[n]$ from $x[n]$ using only discreet-time processing.

The System on the left is called **Sampling rate Expander** or simply an **expander**. It's out put is

Changing The Sampling Rate Using Discrete-Time Processing

Increasing the Sampling Rate by an Integer Factor

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

Or equivalently,

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

The system on the right (above figure) is a lowpass discrete-time system with cutoff frequency on Or equivalently, Π/L and gain L. This system plays a role similar to the ideal D/C converter.

Changing The Sampling Rate Using Discrete-Time Processing

Increasing the Sampling Rate by an Integer Factor

The Fourier transform of can be expressed as

$$x_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-KL]\right)e^{-j\omega n}$$

$$= \overset{x_e[n]}{\sum_{k=-\infty}^{\infty}} x[k]e^{-j\omega Lk} = X(e^{j\omega L})$$

Thus the Fourier transform of the out put of the expander Is a frequency scaled version of the Fourier transform of the input, i.e., Ω is replaced by ΩL so that ω is Normalized by $\omega = \Omega T'$

Changing The Sampling Rate Using Discrete-Time Processing

Frequency-domain illustration of interplation

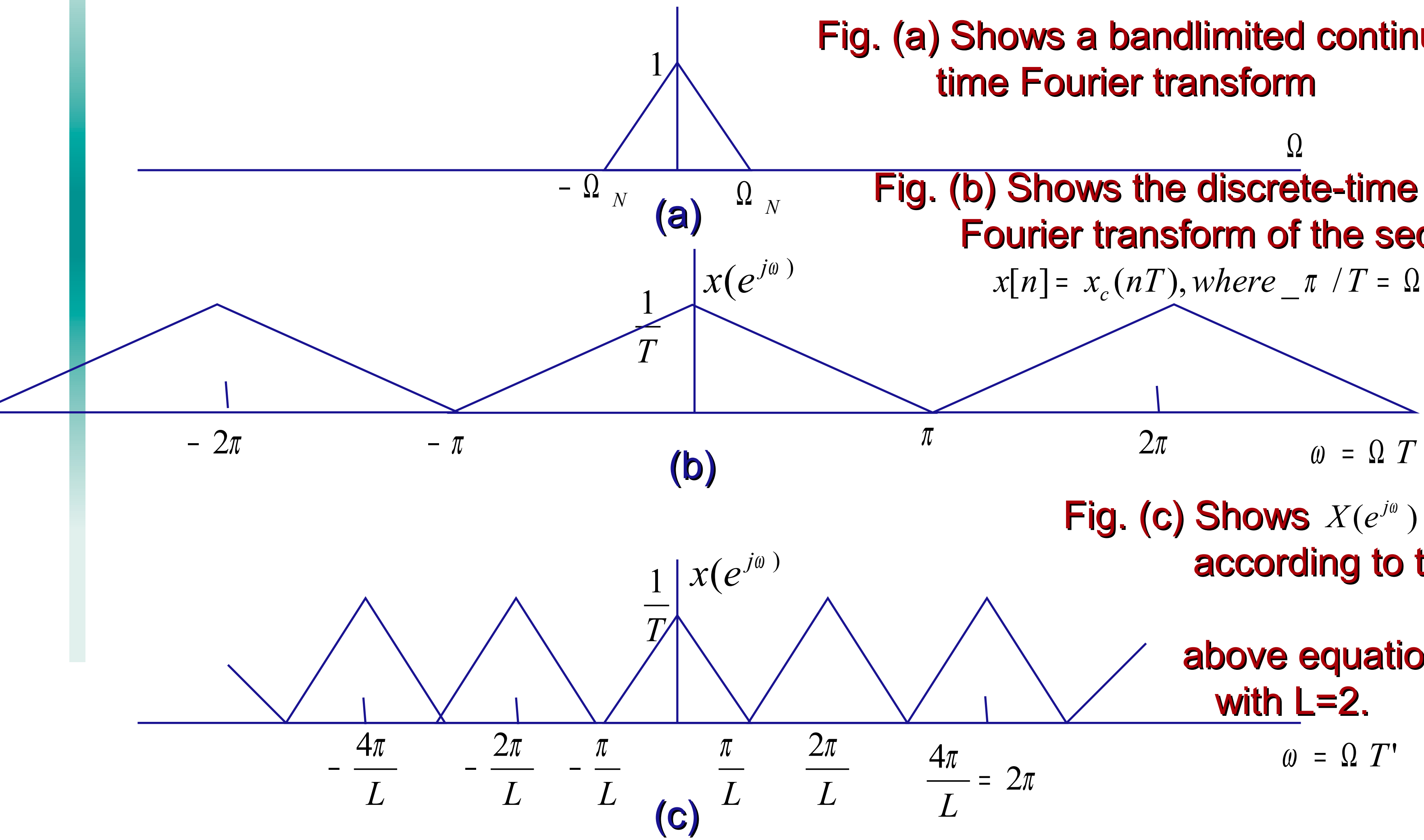


Fig. (a) Shows a bandlimited continuous time Fourier transform

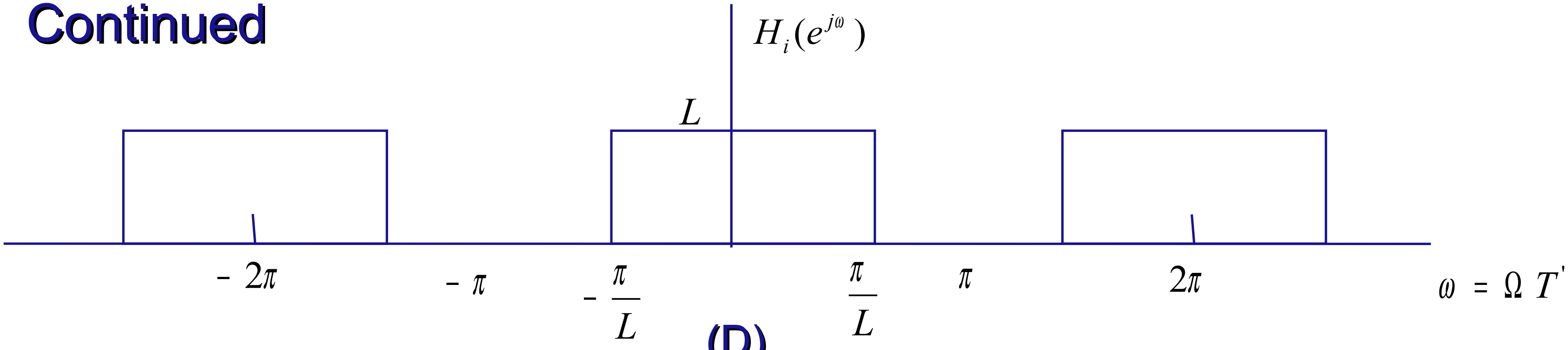
Fig. (b) Shows the discrete-time Fourier transform of the sequence

$x[n] = x_c(nT), where\_\pi / T = \Omega_N$

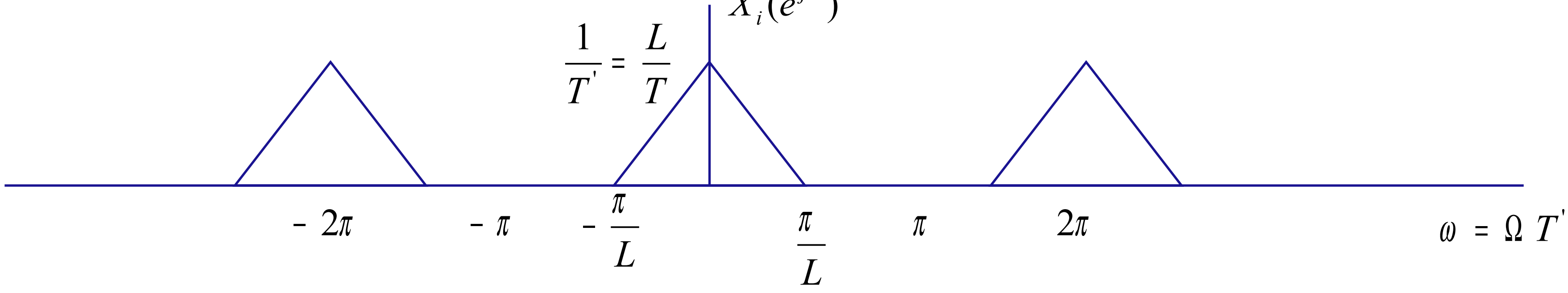
Fig. (c) Shows $X(e^{j\omega})$ according to the above equation, with L=2.

Changing The Sampling Rate Using Discrete-Time Processing

Continued



(D)



(e)

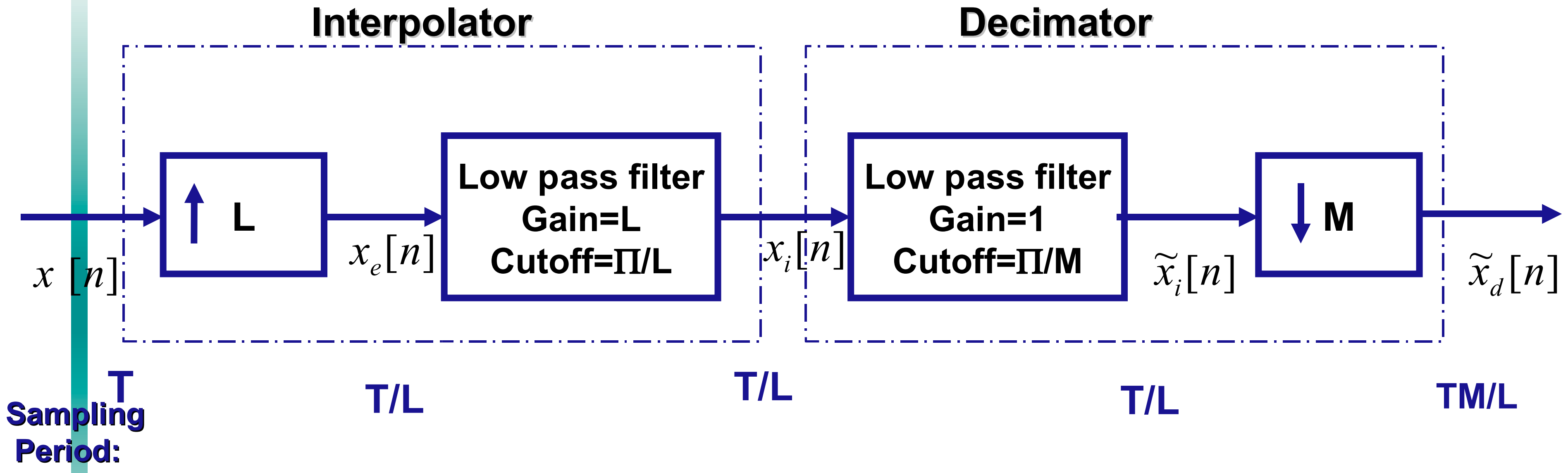
Fig. (e) Shows the the Fourier transform of the desired signal $x_i[n]$ We see that $X_i(e^{j\omega})$ Can be obtained from $X_e(e^{j\omega})$ by correcting the amplitude scale from 1/T to 1/T' and by removing all the frequency-scaled images of $X_e(e^{j\omega})$ except at integer multiple of 2Π.

Changing The Sampling Rate By a Noninteger Factor

- By combining decimation and interpolation it is possible to change the sampling rate by noninteger factor.

- An interpolator which decrease the sampling period from T to T/L, Followed by Decimator which increase the sampling period by M, produce an output sequence $\tilde{x}_d[n]$ that has an effective sampling period of T'=TM/L. (See the following fig.)

Changing The Sampling Rate By a Noninteger Factor



Interpolator

Decimator

$x[n]$

↑ L

$x_e[n]$

Low pass filter
Gain=L
Cutoff=Π/L

$x_i[n]$

Low pass filter
Gain=1
Cutoff=Π/M

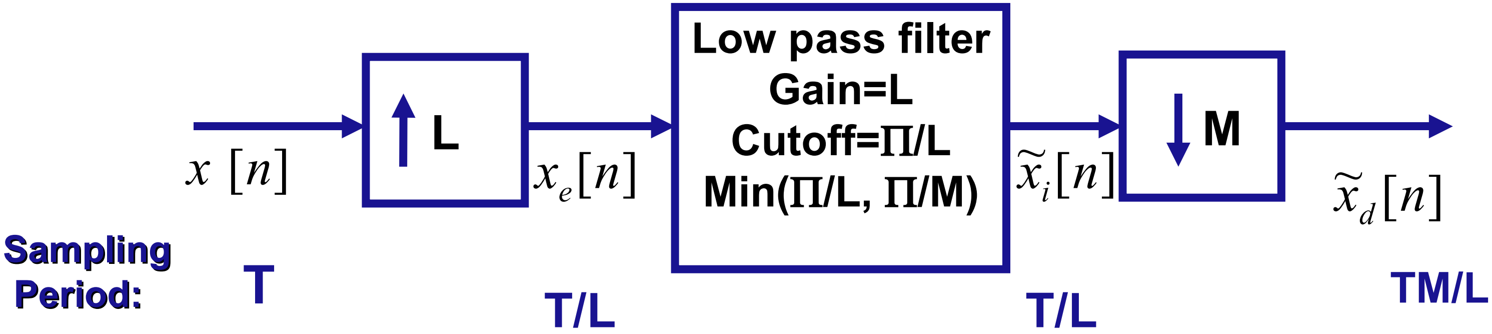
$\tilde{x}_i[n]$

↓ M

$\tilde{x}_d[n]$

Sampling Period: T | T/L | T/L | T/L | TM/L

System for changing the sampling rate by a noninteger factor.



$x[n]$

↑ L

$x_e[n]$

Low pass filter
Gain=L
Cutoff=Π/L
Min(Π/L, Π/M)

$\tilde{x}_i[n]$

↓ M

$\tilde{x}_d[n]$

Sampling Period: T | T/L | T/L | TM/L

Simplified system in which the decimation and interpolation filters are combined.

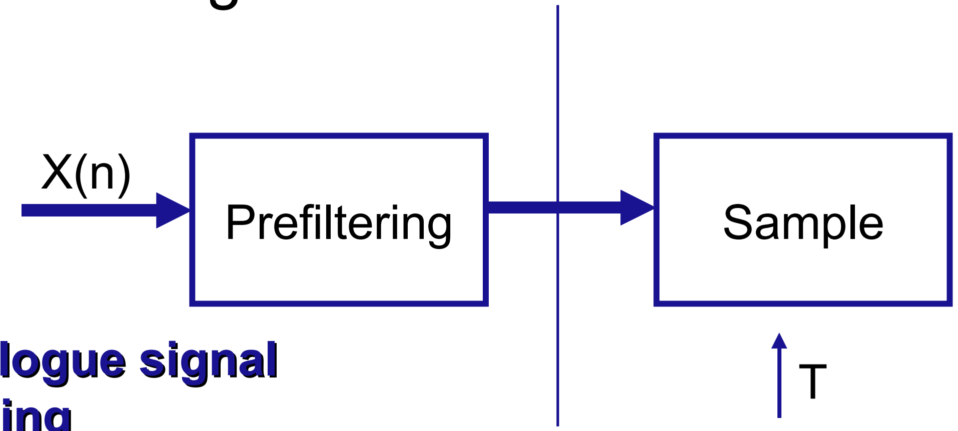
Practical Consideration in AD/DA conversion

Practical Problems:

- Continuous-time signals are not band limited
- Ideal lowpass filter is impossible to be realized

Prefiltering to Avoid Aliasing

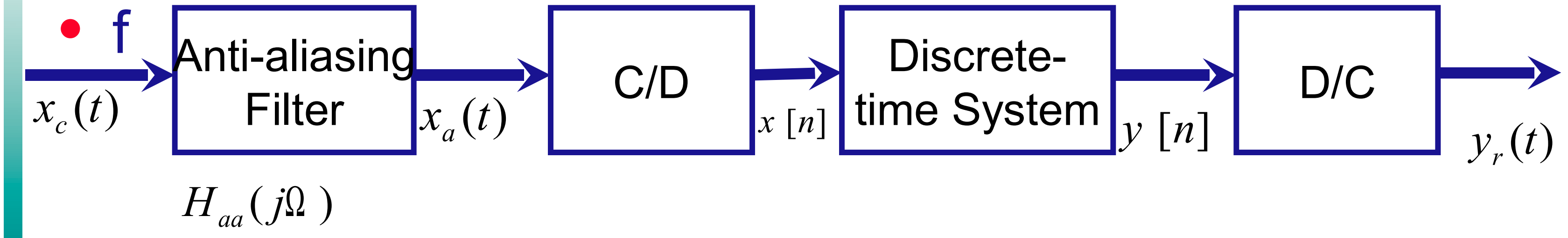
When processing analogue system if the input signal is not band limited or if the Nyquist frequency of the input is to high, prefiltering is often used to avoid aliasing.



Prefiltering the analogue signal to reduce anti-aliasing

Continued.

- f



$x_c(t)$

Anti-aliasing Filter

$H_{aa}(j\Omega)$

$x_a(t)$

C/D

$x[n]$

Discrete-time System

$y[n]$

D/C

$y_r(t)$

Fig. Use of Prefiltering to avoid aliasing

For an ideal lowpass anti-aliasing filter (above fig.) behaves as a linear time-invariant system with frequency response given by the following equation even When $x_c(j\Omega)$ is not bandlimited.

$$H_{eff}(j\Omega) = \begin{cases} H(e^{j\Omega t}) & |\Omega| < \pi/T, \\ 0, & |\Omega| > \pi/T. \end{cases}$$

In practice, the frequency response $H_{aa}(j\Omega)$ can not be ideally bandlimited, but $H_{aa}(j\Omega)$ Can be made small for $|\Omega| < \pi/T$ so that the aliasing is minimized. In this case, the Overall frequency response of the system in the above fig. would be approximately

$$H_{eff}(j\Omega) \approx H_{aa}(j\Omega)H(e^{j\Omega T})$$

Chapter-7
Digital Filter Design

Filter can be defined as a system that modifies certain frequencies relative to others.

Digital filter is a linear shift invariance system (LIS).

The designing filter involves the following stages:

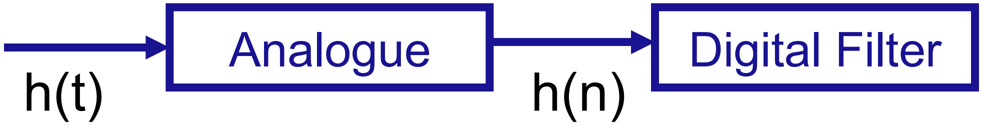
4) Desired characteristics (Specification) of the system.
5) Approximation of the specification using a casual discrete-time system.
6) The realization of the system (building the filter by finite arithmetic computation.

Design of Discrete-Time IIR Filters From Continuous-Time Filters

The traditional approach to the design of discrete-time IIR filters involves the transformation of continuous-time filter into a discrete-time filter meeting prescribed specification.

1. Filter Design by Impulse Invariance

Analogue filter can be changed to digital filter by sampling the impulse response h(t) of analogue. (concept of impulse invariance)



In the impulse invariance design procedure the impulse response of the discrete-time filter is chosen as equally spaced samples of the impulse response of the continuous-time filter; i.e.

$$h[n] = T_d h_c(nT_d)$$

Where: T_d=sampling interval

Note: Impulse invariance techniques have problem of aliasing

Continued..

To develop the transformation (from continuous-time to discrete-time), let us consider the simple function of the continuous time filter expressed in terms of partial fraction expression, as:

$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k}$$

The corresponding impulse response is

$$h_c(t) = \begin{cases} \sum_{k=1}^{N} A_k e^{skt} & t \geq 0, \\ 0, & t < 0. \end{cases}$$

The Impulse response of the discrete-time filter obtained by sampling $T_d h_c(t)$ **is**

$$h[n] = T_d h_c(nT_d) = \sum_{k=1}^{N} T_d A_k e^{s_k n T_d} u[n]$$

$$= \sum_{k=1}^{N} T_d A_k e^{skT_d} u[n]$$

Continued..

The system function H (z) of the discrete-time filter is therefore given by

$$H(z)=\sum_{k=1}^{N}\frac{T_d A_k}{1-e^{s_k T_d}z^{-1}}$$

2. Bilinear Transformation

- This technique have in distortion of frequency axis.
- Avoid the problem of aliasing.

With $H_c(s)$ denoting the continuous-time system function and H(z) the discrete-time System function, the bilinear transformation corresponds to replacing s by

$$s=\frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right),$$

That is,

$$H(z)=H_c\left[\frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right].$$

FIR Design by Window

IIR filter design are based on transformation of continuous-time IIR system in to Discrete time system.
In contrast, **FIR filters** are almost entirely restricted to discrete-time implementations.
The design technique for FIR filters are based on directly approximating the desired frequency response of the discrete-time system,

The simplest method of FIR filter design is called the Window method.

This method generally begins with an ideal desired frequency response that can be represented as:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n]e^{-j\omega n},$$

Continued…

Where $H_d[n]$ is the corresponding impulse response sequence, which can be expressed in terms of $H_d(e^{jw})$ as

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

To obtain a casual FIR filter from $H_d[n]$ is to define a new system with impulse response h[n] given by

$$h\,[n] = \begin{cases} h_d[n], & 0 \le n \le M, \\ 0, & otherwise. \end{cases}$$

Continued…

More generally we can represent **h[n]** as the product of desired impulse response and a finite-duration "window" **w[n]**; i.e.'

$$h[n] = h_d[n]w[n],$$

Where for simple truncation as in above equation the window is the rectangular window

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

It follows from the modulation or windowing theorem that

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

Properties of commonly used windows

Some commonly used windows are defined by the following equations:

Rectangular

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

Bartlett (triangular)

$$w[n] = \begin{cases} 2n/M, & 0 \le n \le M/2 \\ 2 - 2n/M, & M/2 < n \le M, \\ 0, & otherwise \end{cases}$$

Hanning

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/M), & 0 \le n \le M, \\ 0, & otherwise. \end{cases}$$

Hamming

$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/M), & 0 \le n \le M, \\ 0, & otherwise. \end{cases}$$

The Kaiser Window Filter Design Method

The Kaiser window is defined as

$$w[n] = \begin{cases} \dfrac{I_o[\beta(1-[(n-\alpha)/\alpha]^2)^{1/2}]}{I_o(\beta)} & 0 \le n \le M, \\ 0, & otherwise. \end{cases}$$

Where α=M/2, and $I_o(\cdot)$ represents the zero-order modified Bessel function of the first kind.

Properties of Linear phase FIR Filter

The Shape of the impulse response defined by Equation.

$$H(z) = \sum_{n=0}^{M-1} h(n) = Z^{-n} = Z^{-(m-1)} \sum_{n=0}^{M-1} h(n) Z^{M-1-n}$$

The frequency response of the above system is

$$H(e^{jw}) = \sum_{n=0}^{M-1} h(n) e^{j\omega n} \quad -\pi \le \omega \le \pi$$

Taking $\angle H(e^{j\omega}) = -\alpha\,\omega$ **where** $\alpha = \frac{M-1}{2}$ **in linear phase and with symmetry condition** $h(n) = h(M-1-n)$

1) **Case 1 when M is odd**

$$\alpha = \frac{M-1}{2}$$ **integer**

Continued..

1) Case 2 when M is even

$$\alpha = \frac{M-1}{2}$$ Non integer

For other linear phase $\angle H(e^{j\omega}) = \beta - \alpha\omega$ and anti symmetric condition $h(n) = -h(m-1-n)$ which is opposite of symmetry.

1) Case 1 when M is odd

$$\alpha = \frac{M-1}{2}$$ integer

1) Case 2 when M is even

$$\alpha = \frac{M-1}{2}$$ Non integer

Algorithmic Procedure For The Design Of FIR Filters With Generalized Linear Phase

Type I

In designing a causal Type I linear phase FIR filter, it is convenient first to consider the design of a zero-phase filter, i.e., one for which

$$h_e[n] = h_e[-n]$$

and then to insert sufficient delay to make it causal.

Type I For Linear phase and symmetry

M: odd, β=0, $\alpha = \frac{M-1}{2}$

$$h(n) = h(M-1-n)$$ Symmetry condition

$$H(e^{j\omega}) = \left[\sum_{n=0}^{M-1/2} a(n)\cos w.n\right] e^{-j\omega\frac{M-1}{2}}$$

$$a(0) = h(\frac{M-1}{2})$$ Is the middle sample

Algorithmic Procedure For The Design Of FIR Filters With Generalized Linear Phase

Type II

A Type II causal filter is one for which h[n]=0 outside the range $0 \le n \le M$, with filter length (M+1) even, I.e.

M: even, and with the symmetry property

$$h(n) = h(M-1-n)$$

$\beta = 0$ and $\alpha = \dfrac{M-1}{2}$ Not integer

The frequency response $H(e^{j\omega})$ can be expressed in the form

$$H(e^{j\omega}) = \left\{ \sum_{n=1}^{M/2} b(n)\cos(w(n-1/2)) \right\} e^{-j\omega\frac{M-1}{2}}$$

When $b(n) = 2h(\dfrac{M}{2} - n), n = 1,2,\ldots(M+1/2),$

Algorithmic Procedure For The Design Of FIR Filters With Generalized Linear Phase

Type III

For Linear Anti-symmetric

When $\beta = \pi / 2$ and $\alpha = \dfrac{M - 1}{2}$ integer

$$h(n) = -h(M - 1 - n)$$ Anti-symmetric

The frequency response $H(e^{j\omega})$ can be expressed in the form

$$H(e^{j\omega}) = \left[\sum_{n=1}^{\frac{M-1}{2}} c(n)\sin(\omega n)\right] e^{j\left[\frac{\pi}{2} - (\frac{M-1}{2})\omega\right]}$$

Where $c(n) = 2h(\dfrac{M-1}{2})$

at $H_r(\omega) = 0\_for\_\omega = 0\_\omega = \pi$

Algorithmic Procedure For The Design Of FIR Filters With Generalized Linear Phase

Type IV

For Linear Anti-symmetric

When $\beta = \pi / 2$ and $\alpha = \frac{M - 1}{2}$ not integer

$h(n) = - h(M - 1 - n)$ Anti-symmetric

The frequency response $H(e^{j\omega})$ can be expressed in the form

$$H(e^{j\omega}) = \left[\sum_{n=1}^{\frac{M}{2}} d(n) \sin[\omega (n - \frac{1}{2})] \right] e^{j\left[\frac{\pi}{2} - \omega (\frac{M-1}{2})\right]}$$

Where $d(n) = 2h(\frac{M}{2} - n)$

at $\omega = 0, H_r(0) = 0$ $\angle H(0) = e^{j.\pi/2} = j$

Thank you