Chapter 1: Signals, Systems and Signal Processing[i].

Signal:

We know that a signal can be a rather abstract notion, such as a flashing light on our bike (turn signal), or a referee's whistle indicating start, halt or end of football match, door bell, etc. Signal can be defined as a detectable physical quantity or impulse (as a voltage, current or magnetic field strength) by which messages or information can be transmitted.

A signal is a source of information generally a physical quantity which varies with respect to time, space, temperature like any independent variable. In other word, a *signal* is function of independent variables that carry some information.

Eg.$x(t) = 10t + 7t^2$. →Signal of one independent variable (i.e. time)

$s(x,y) = 3x + 2xy + 10y^2$. →Signal of two independent variables (i.e. x and y).

Speech, electrocardiogram (ECG), electroencephalogram (EEG), etc signals are examples of information-bearing signals that evolve as functions of a single independent variable, namely, time. An example of signal that is function of two independent variables is an image signal.

System:

A system may be defined as a physical device that performs an operation on a signal. More specifically, a system is something that can **manipulate, change, record, or transmit** signals. In Digital Signal Processing, a system may be defined by algorithm. For example, a filter used to reduce the noise and interference corrupting a desired information-bearing signal is called a system. In this case the filter performs some operation/s on the signal, which has the effect of reducing (filtering) the noise and interference from the desired information-bearing signal.

Signal Processing:

When we pass a signal through a system, as in filtering, we say that we have processed the signal. In this modern world we are surrounded by all kinds of signals in various forms. Some of the signals are natural, but most of the signals are manmade. Some signals are necessary (speech), some are pleasant (music), while many are unwanted or unnecessary in a given situation. In an engineering context, signals are carriers of information, both useful and unwanted. Therefore extracting or enhancing the useful information from a mix of conflicting information is a simplest form of signal processing. More generally, signal processing is an operation designed for extracting, enhancing, storing, and transmitting useful information. The distinction between useful and unwanted information is often subjective as well as objective. Hence signal processing tends to be application dependent.

Digital Signal Processing (DSP)

Digital Signal Processing refers to methods of filtering and analyzing time-varying signals based on the assumption that the signal amplitudes can be represented by a finite set of integers corresponding to the amplitude of the signal at a finite number of points in time. Digital Signal Processing is distinguished from other areas in computer science by the unique type of data it uses: *signals*. In most cases, these signals originate as sensory data from the real world: seismic vibrations, visual images, sound waves, etc. DSP is the mathematics, the algorithms, and the techniques used to manipulate these signals after they have been converted into a digital form. This includes a wide variety of goals, such as: enhancement of visual images, recognition and generation of speech, compression of data for storage and transmission, etc.

Digital signal processing is the study of signals in a digital representation and the processing methods of these signals. DSP includes subfields like: audio signal processing, control engineering, digital image processingand speech processing. RADAR Signal processing and communications signal processing are two other important subfields of DSP.
Since the goal of DSP is usually to measure or filter continuous real-world analog signals, the first step is usually to convert the signal from an analog to a digital form, by using an analog to digital converter. Often, the required output signal is another analog output signal, which requires a digital to analog converter.
The algorithms required for DSP are sometimes performed using specialized computers, which make use of specialized microprocessors called digital signal processors (also abbreviated *DSP*). These process signals in real time and are generally purpose-designed application-specific integrated circuits (ASICs). When flexibility and rapid development are more important than unit costs at high volume, DSP algorithms may also be implemented using field-programmable gate arrays (FPGAs).

Basic Elements of a Digital Signal Processing System:

The signals that we encounter in practice are mostly analog signals. These signals, which vary continuously in time and amplitude, are processed using electrical networks containing active and passive circuit elements. This approach is known as analog signal processing (ASP), for example, radio and television receivers.

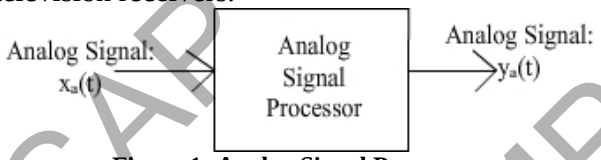


Figure 1: Analog Signal Processor.

They can also be processed using digital hardware containing adders, multipliers, and logic elements or using special-purpose microprocessors. However, one needs to convert analog signals into a form suitable for digital hardware. This form of the signal is called a digital signal. It takes of the finite number of values at specific instances in time, and hence it can be represented by binary numbers, or bits. The processing of digital signals is called DSP: in block diagram form it is represented by

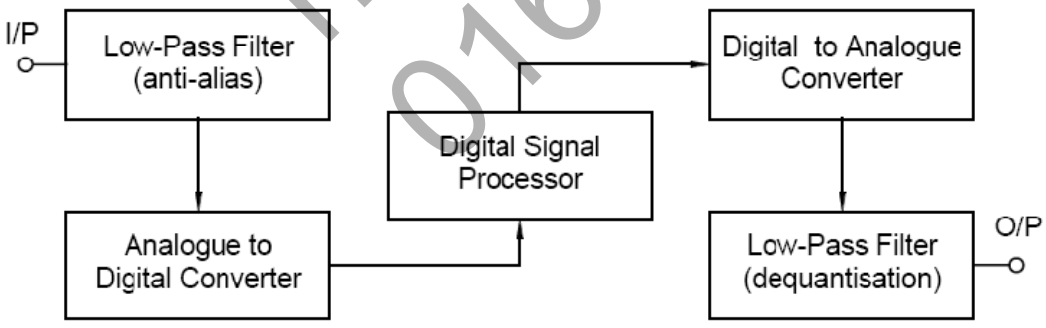


Figure 2: Basic Elements of Digital Signal Processing.

The input to and output from the systems are analog in nature. The various block elements are discussed below:

Low-Pass Filter (anti-alias): This is a prefilter or an antialiasing filter, which conditions the analog signal to prevent aliasing.

Analog to Digital Converter (ADC): ADC produces a stream of binary numbers from analog signals. It has three sub-elements as sampling, quantization and encoding.

Digital Signal Processor: This is the heart of DSP and can represent a general-purpose computer or a special-purpose processor, or digital hardware, and so on. The basic elements of DSP are adder and multiplier.

Digital to Analog Converter (DAC): This is the inverse operation to the ADC, which produces a staircase waveform from a sequence of binary numbers, a first step towards producing an analog signal.

Low-Pass Filter (dequantisation): This is a postfilter to smooth out staircase waveform into the desired analog signal. It is simply a low pass filter which smooth out (or interpolates) the sequences obtained from DAC.

Performance or these systems is usually limited by the performance (*i.e.* speed, resolution and linearity) of the analog-to-digital converter. However, when using a DSP we should never forget two facts:

- If information was not present in the sampled signal to start with, no amount of digital manipulation will extract it.
- Real signals come with noise.

Advantages of DSP over ASP:

A major drawback of ASP is its limited scope for performing complicated signal processing applications. This translates into nonflexibility in processing and complexity in system designs. All of these generally lead to expensive products. On the other hand, using DSP approach, it is possible to convert an inexpensive personal computer into a powerful signal processor. Some important advantages of DSP are:

1. Systems using the DSP approach can be developed using software running on a general purpose computer. Therefore DSP is relatively convenient to develop and test, and the software is portable.
2. DSP operations are based solely on additions and multiplications, leading to extremely stable processing capability – for example, stability independent of temperature.
3. Digital signals are easily stored on magnetic or other storable media.
4. The digital signals become transportable and can be processed off-line in a remote laboratory.
5. DSP operations can easily be modified in real time, often by simple programming changes, or by reloading of registers.
6. It is difficult to perform precise mathematical operations on signals in analog form but these same operations can be routinely implemented on a digital computer using software.
7. DSP has lower cost due to VLSI technology, which reduces costs of memories, gates, microprocessors, and so forth.

The principal disadvantage of DSP is the speed of operation of analog to digital converters and DSP, especially at very high frequencies and wide bandwidth require fast-sampling rate A/D converters and fast digital signal processors. Primarily due to the above advantages, DSP is now becoming a first choice in many technologies and applications, such as consumer electronics, communications, wireless telephones, and medical imaging.

Applications of Digital Signal Processing:

There are various application areas of digital signal processing (DSP) due to the availability of high resolution spectral analysis. It requires high speed processor to implement the Fast Fourier Transform. Some of these areas can be listed as below:

1. Speech Processing

Speech is a one dimensional signal. Digital processing of speech is applied to a wide range of speech problems such as speech spectrum analysis, channel vocoders, etc. DSP is applied to speech coding, speech enhancement, speech analysis and synthesis, speech recognition and speaker recognition.

2. Image Processing

Any two-dimensional pattern is called an image. Digital processing of images requires two-dimensional DSP tools such as Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT) algorithms and z-transforms. Processing of electrical signals extracted from images by digital techniques include image formation and recording, image compression, image restoration, image reconstruction and image enhancement.

3. Radar Signal Processing

Rader stands for "Radio Detection and Ranging". Improvement in signal processing is possible by digital technology. Development of DSP has led to greater sophistication of radar tracking algorithms. Radar systems consist of transmit-receive antenna, digital processing system and control unit.

4. Digital Communications

Application of DSP in digital communication specially telecommunications comprises of digital transmission using pulse code modulation (PCM), digital switching using Time Division Multiplexing (TDM), echo control and digital tape-recorders. DSP in telecommunication systems are found to be cost effective due to availability of medium and large scale digital ICs. These ICs have desirable properties such as small size, low cost, low power, immunity to noise and reliability.

5. Spectral Analysis

Frequency-domain analysis is easily and effectively possible in digital signal processing using fast Fourier transform (FFT) algorithms. These algorithms reduce computational complexity and also reduce the computational time.

6. Sonar Signal Processign

Sonar stands for "Sound Navigation and Ranging". Sonar is used to determine the range, velocity and direction of targets that are remote from the observer. Sonar uses sound waves at lower frequencies to detect objects under water.

7. Aviation
8. Astronomy
9. Telecommunication networks
10. Satellite communication
11. Microprocessor systems
12. Industrial noise control.

Types of Signals:

1. Continuous time Vs. Discrete-time signal

As the names suggest, this classification is determined by whether or not the time axis (x-axis) is discrete (countable) or continuous. Continuous-time signals are represented by x(t) where t denotes continuous-time and discrete-time signals or sequences are represented by x[n] where n is an integer denotes discrete-time.

2. Continuous value Vs. Discrete-value Signal

In discrete-value signals, there are finite values in y-axis. For example if we look for values between 0 to 1 V, there are infinite numbers of values (Continuous value). But if we fixed

the number of values to say 10 (0, 0.1, 0.2, …) or say 5 (0, 0.2, 0.4, 0.6, …) then there are finite numbers of values in y-axis, known as discrete-value signal.

3. Periodic Vs. Non-Periodic Signal

Periodic signals repeat with some period T, while aperiodic or nonperiodic signals do not. We can define a periodic function through the following mathematical expression, where t can be any number and T is a positive constant:

$$\boldsymbol{x(t) = x(t + T)}$$

Where T is the fundamental period (the smallest value of T that still allows above equation to be true.

For discrete time signal

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

Here, N is the time period which is an integer. That mean the signal repeats after N samples.

4. Causal Vs. Anticausal Vs. Noncausal

Causal signals are signals that are zero for all negative time, while anticausal are signals that are zero for all positive time. Noncausal signals are signals that have nonzero values in both positive and negative time.

5. Even Vs. Odd

An even signal is any signal x(t) such that x(t) = x(-t). Even signals can be easily spotted as they are symmetric around the vertical axis. An odd signal, on the other hand, is a signal that satisfy x(t) = -x(-t) (Also known as Anti-symmetric signal).

Using the definitions of even and odd signals, we can show that any signal can be written as a combination of an even and odd signal. That is, every signal has an odd-even decomposition.

$$x(t) = x_e(t) + x_o(t)\ (i.e. even\ part + odd\ part)$$

$$x_e(t) = \frac{x(t) + x(-t)}{2}\ \&\ x_o(t) = \frac{x(t) - x(-t)}{2}$$

6. Deterministic Vs. Random Signal

A deterministic signal is a signal in which each value of the signal is fixed and can be determined by a mathematical expression, rule, or table. Because of this the future values of the signal can be calculated from past values with complete confidence. On the other hand, a random signal has a lot of uncertainty about its behavior. The future values of a random signal cannot be accurately predicted and can be usually only be guessed based on the averages of sets of signals.

7. Energy Vs. Power Signal

In electrical systems, a signal may represent a voltage or current. Consider a voltage v(t) developed across a resistor R, producing a current i(t). The *instantaneous power* dissipated in this resistor is defined by

$$p(t) = \frac{v^2(t)}{R}$$

$$or\ equivalently,\ \ p(t) = Ri^2(t)$$

In both cases, the instantaneous power p(t) is proportional to the squared amplitude of the signal. Furthermore, for a resistance R of 1Ω, we see that above equations take on the same mathematical form. Accordingly, in signal analysis it is customary to define power in terms of a 1Ω resistor, so that, regardless of whether a given signal x(t) represents a voltage or a current, we may express the instantaneous power of the signal as,

$$p(t) = x^2(t)$$

Based on this convention, we define the *total energy* of the continuous-time signal x(t) as

$$E = \lim_{T\to\infty} \int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t)dt = \int_{-\infty}^{\infty} x^2(t)dt$$

And its *average power* as

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t)dt$$

In the case of a discrete-time signal x[n], the integrals are replaced by corresponding sums. Thus the total energy of x[n] is defined by

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

And its average power is defined by

$$If\ the\ signal\ is\ aperiodic, \qquad P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2$$

$$\text{If the signal is periodic,} \qquad P = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$$

The second expression is the average power in a periodic signal x[n] with fundamental period N. If x[n] is real we can take $|x[n]|^2 = x^2[n]$.

A signal is referred to as an Energy signal, if and only if the total energy of the signal satisfies the condition $0 < E < \infty$.

On the other hand, it is referred to as a Power signal, if and only if the average power of the signal satisfies the condition, $0 < P < \infty$.

The energy and power classifications of signals are mutually exclusive. In particular, an energy signal has zero average power, whereas a power signal has infinite energy. It is also interest to note that periodic signals and random signals are usually viewed as power signals, whereas signals that are both deterministic and non-periodic are energy signals.

#Find energy or power of the following signals:

$$\#\ \boldsymbol{x[n] = \left(\frac{1}{2}\right)^n u[n]}$$

The given signal is aperiodic signal and we will find energy as,

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \sum_{n=-\infty}^{\infty} \left|\left(\frac{1}{2}\right)^n u[n]\right|^2 = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{2n} = \frac{1}{1-\left(\frac{1}{2}\right)^2} = \frac{4}{3}$$

$$\#\ \boldsymbol{x[n] = \cos\left(\frac{\pi n}{4}\right)}$$

The given signal is periodic signal with period N = 8 and we will find power as,

$$P = \frac{1}{N} \sum_{n=0}^{N-1} x^2[n] = \frac{1}{8} \sum_{n=0}^{7} \cos^2\left(\frac{\pi n}{4}\right)$$

$$= \frac{1}{8} \sum_{n=0}^{7} \left[\frac{1+\cos\left(\frac{\pi n}{2}\right)}{2}\right] = \frac{1}{8} \sum_{n=0}^{7} \frac{1}{2} + \frac{1}{8} \times \frac{1}{2} \sum_{n=0}^{7} \cos\left(\frac{\pi n}{2}\right) = \frac{1}{8} \times \frac{8}{2} + \frac{1}{16} \times 0 = \frac{1}{2}$$

Continuous-Time Sinusoidal Signals

A simple harmonic oscillation is mathematically described by the following continuous-time sinusoidal signal:

$$x_a(t) = A\cos(\Omega t + \theta), \quad -\infty < t < \infty$$

This signal is completely characterized by three parameters: *amplitude* A of the sinusoid, *frequency* Ω in radians per second, and *phase* θ in radians. Instead of Ω, we often use the frequency F in cycles per second or hertz (Hz), where $\Omega = 2\pi F$.

In terms of F the sinusoid becomes: $x_a(t) = A\cos(2\pi Ft + \theta), -\infty < t < \infty$
The analog sinusoidal signal is characterized by the following properties:

1. For every fixed value of the frequency F, $x_a(t)$ is periodic: $x_a(t + T_p) = x_a(t)$, where $T_p = 1/F$ is the fundamental period of the sinusoidal signal.
2. Continuous-time sinusoidal signals with distinct frequencies are themselves distinct.
3. Increasing the frequency F results in an increase in the rate of oscillation of the signal, in the sense that more periods are included in a given time interval.

The relationships we have described for sinusoidal signals carry over to the class of complex exponential signals

$$x_a(t) = Ae^{j(\Omega t + \theta)}$$

By definition, frequency is an inherently positive physical quantity. This is obvious if we interpret frequency as the number of cycles per unit time in a periodic signal. However, in many cases, only for mathematical convenience, we need to introduce negative frequencies.

Hence the frequency range for analog sinusoids is $-\infty < F < \infty$.

Discrete-Time Sinusoidal Signals

A discrete-time sinusoidal signal may be expressed as:

$$x[n] = A\cos(\omega n + \theta), \quad -\infty < n < \infty$$

Where n is an integer variable, called the sample number, A is the *amplitude* of the sinusoid, ω is the *frequency* in radians per second, and θ is *phase* in radians. Instead of ω, we often use the frequency variable f defined by $\omega = 2\pi f$.

In terms of f the sinusoid becomes: $x[n] = A\cos(2\pi f + \theta), \quad -\infty < n < \infty$
The frequency f has dimensions of cycles per sample.
In contrast to continuous-time sinusoids, the discrete-time sinusoids are characterized by the following properties:

1. *A discrete-time sinusoid is periodic only if its frequency f is a rational number:*

By definition, a discrete-time signal x[n] is periodic with period N (>0) iff

$$x[n + N] = x[n] \qquad \text{for all } n$$

The smallest value of N for which the above equation is true is called the *fundamental period.*
Proof: For a sinusoid with frequency f_0 to be periodic, we should have

$$Cos[2\pi f_0(n + N) + \theta] = Cos[2\pi f_0 n + \theta]$$

This relation is true if and only if there exists an integer k such that

$$2\pi f_0 N = 2k\pi \Rightarrow f_0 = k/N$$

Hence, a discrete-time sinusoidal signal is periodic only if its frequency f_0 can be expressed as the ratio of two integers (i.e. f_0 is rational). To determine the fundamental period N of a periodic sinusoid, we express its frequency f_0 as the ratio of two integers and cancel common factors so that k and N are relatively prime. Then the fundamental period of the sinusoid is equal to N.

2. *Discrete-time sinusoids whose frequencies are separated by an integer multiple of 2π are identical.*

Proof: Let us consider the sinusoid $\cos[\omega_0 n + \theta]$. It easily follows that

$$\cos[(\omega_0 + 2\pi)n + \theta] = \cos[\omega_0 n + 2\pi n + \theta] = \cos[\omega_0 n + \theta]$$

As a result, all sinusoidal sequences
$x_k[n] = A\cos[\omega_k n + \theta]$ $k = 0, 1, 2, 3, \ldots\ldots$
where $\omega_k = \omega_0 + 2\pi k$ $-\pi \le \omega_0 \le \pi$
are *indistinguishable (i.e. identical).* On the other hand, the sequences of any two sinusoids with frequencies with frequencies in the range $-\pi \le \omega \le \pi$ or $-\frac{1}{2} \le f \le \frac{1}{2}$ are distinct. Consequently, discrete-time sinusoidal signals with frequencies $|\omega| \le \pi$ or $|f| \le \frac{1}{2}$ are unique. Any sequence resulting from a sinusoid with a frequency $|\omega| > \pi$, or $|f| > \frac{1}{2}$, is identical to a sequence obtained from a sinusoidal signal with frequency $|\omega| < \pi$. Thus we regard frequencies in the range $-\pi \le \omega \le \pi$ or $-\frac{1}{2} \le f \le \frac{1}{2}$ as unique and all frequencies $|\omega| > \pi$, or $|f| > \frac{1}{2}$, as aliases[1].

$x[n] = \cos(0.6\pi n) \Rightarrow f = 0.3 = 3/10$, rational number hence periodic.

$x[n] = \cos(0.8n) \Rightarrow f = 0.4/\pi$, irrational number hence aperiodic.

[1]In signal processing and related disciplines, aliasing refers to an effect that causes different signals to become indistinguishable (or aliases of one another) when sampled. It also refers to the distortion or artifact that results when the signal reconstructed from samples is different from the original continuous signal.

3. *The highest rate of oscillation in a discrete-time sinusoid is attained when $\omega = \pi$ (or $\omega = -\pi$) or, equivalently, $f = \frac{1}{2}$ (or $f = -\frac{1}{2}$).*

To illustrate this property, let us investigate the characteristics of the sinusoidal signal sequence

$$x[n] = \cos\omega_0 n$$

When the frequency varies from 0 to π. To simplify the argument, we take values of $\omega_0 = 0, \pi/8, \pi/4, \pi/2, \pi$ corresponding to f = 0, $^1/_{16}$, $^1/_8$, ¼, ½; which results in periodic sequences having periods N = ∞, 16, 8, 4, 2. We note that the period of the sinusoid decreases (or rate of oscillation increases) as the frequency increases.

To see what happens for $\pi \le \omega_0 \le 2\pi$, we consider the sinusoids with frequencies $\omega_1 = \omega_0$ and $\omega_2 = 2\pi - \omega_0$. Note that as ω_1 varies from π to 2π, ω_2 varies from π to 0. It can be easily seen that

$x_1[n] = A\cos\omega_1 n = A\cos\omega_0 n$
$x_2[n] = A\cos\omega_2 n = A\cos(2\pi - \omega_0)n = A\cos(-\omega_0 n) = x_1[n]$

Hence ω_2 is an alias to ω_1. If we had used a sine function instead of a cosine function, the result would basically be the same, except for a 180^0 phase difference between the sinusoids $x_1[n]$ and $x_2[n]$. In any case, as we increase the relative frequency ω_0 of a discrete-time sinusoid form π to 2π, its rate of oscillation decreases. For $\omega_0 = 2\pi$ the result is a constant signal, as in the case of $\omega_0 = 0$. Obviously, for $\omega_0 = \pi$ (or f = ½) we have the highest rate of oscillation.

Since discrete-time sinusoidal signals with frequencies that are separated by an integer multiple of 2π are identical, it follows that the frequencies in any interval $\omega_1 \le \omega \le \omega_1 + 2\pi$ constitute all the existing discrete-time sinusoids or complex exponentials. Hence the frequency range for discrete-time sinusoids is finite with duration 2π. Usually, we choose the range $0 \le \omega \le 2\pi$ or $-\pi \le \omega \le \pi$ ($0 \le f \le 1$, $-\frac{1}{2} \le f \le \frac{1}{2}$), which we call the *fundamental range.*

Harmonically Related Complex Exponentials

These are sets of periodic complex exponentials with fundamental frequencies that are multiple of a single positive frequency.

$$s_k(t) = e^{jk\Omega_0 t} = e^{jk2\pi F_0 t} k = 0, \pm 1, \pm 2, - - -$$

$$FundamentalPeriod = \frac{1}{kF_0} = \frac{T_p}{k}$$

Similarly for discrete-time, $s_k[n] = e^{j2\pi k f_0 n}$

Analog to Digital Convertor (ADC):

1. **Sampling**: Continuous-time signal to discrete-time signal

   x(t) ===> x(nT) ≡ x[n], T is the sampling interval.

2. **Quantization**: Discrete-time continuous value signal to discrete-time discrete-value signal.

   x[n] ===> $x_q[n]$. Quantization error, $q_e = x[n] - x_q[n]$.

3. **Coding**: In the coding process, each discrete value $x_q[n]$ is represented by a b – bit binary sequence.

Sampling of Analog Signal:

We limit our discussion to periodic or uniform sampling.

$$x(t) = Acos(2\pi Ft + \theta)$$

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

$$t = nT = \frac{n}{F_s}$$

Sampling periodically at a rate $F_s = 1/T$ samples/sec

$$x(nT) \cong x[n] = Acos(2\pi FnT + \theta) = Acos\left(2\pi\left(\frac{F}{F_s}\right)n + \theta\right)$$

If we compare with discrete-time sinusoid $x[n] = Acos(2\pi fn + \theta)$ we get,

$$f = \frac{F}{F_s}, \ \ or, \omega = \Omega T$$

Is called relative or normalized frequency. We can use f to determine the frequency F in hertz only if the sampling frequency F_s is know.

Recall,

$$-\infty < F < \infty \& -\frac{1}{2} \le f \le \frac{1}{2} cycles/samples$$

$$-\infty < \Omega < \infty \& -\pi \leq \omega \leq \pi radians/samples$$

We find that the frequency of the continuous-time sinusoid when sampled at a rate $F_s = 1/T$ must fall in the range

$$-\frac{1}{2T} = -\frac{F_s}{2} \leq F \leq \frac{F_s}{2} = \frac{1}{2T}$$

Mapping of infinite range to finite frequency range of variable f. since highest frequency in a discrete-time signal is ω = π or f = ½ with sampling rate F_s, the corresponding highest value of F is, $F_{max} = \frac{F_S}{2} = \frac{1}{2T}$

Some terms related to sampling of analog signals:

Sampling Frequency, F_s: It is the number of samples per second while converting continuous-time signal to discrete-time signal.

Nyquist Criteria, $F_s \geq 2F_{max}$: It is the criteria that has to be fulfilled for the reconstruction of signal from discrete-time signal. F_s is the sampling frequency and F_{max} is the maximum frequency contained in continuous-time signal.

Nyquist Rate, $F_N = 2F_{max}$: It is the minimum sampling frequency required for the proper reconstruction of the signal.

Folding Frequency (or Nyquist Frequency), $F_s/2$: The highest frequency that can be reconstructed or measured using discretely sampled data. It is the half of sampling frequency.

 # Consider the two analog sinusoidal signals

$$x_1(t) = \cos(2\pi(10)t) \& x_2(t) = \cos(2\pi(50)t)$$

Sample the two signals at a rate F_s = 40 Hz and find the discrete-time signals obtained.

[Hint: The two sinusoidal signals are identical & consequently indistinguishable. We say that the frequency F_2 = 50 Hz is an alias of the frequency F_1 = 10 Hz at the sampling rate of 40 Hz]

 # Consider the analog signal

$$x(t) = 3cos100\pi t$$

a. Determine the minimum sampling rate required to avoid aliasing.
b. Suppose that the signal is sampled at F_s = 200 Hz, What is the discrete-time signal obtained after sampling?
c. Suppose that the signal is sampled at the rate Fs = 75 Hz, what is the discrete-time signal obtained after sampling?
d. What is the frequency 0 < F < Fs/2 of a sinusoid that yields samples identical to those obtained in part(c) ?

 # Consider the analog signal

$$x(t) = 3\cos 2000\pi t + 5\sin 6000\pi t + 10\cos 12000\pi t$$

a. What is the Nyquist rate for this signal?
b. Assume now that we sample this signal using a sampling rate Fs = 5000 samples/s. What is the discrete-time signal obtained after sampling?

What is the analog signal y(t) we can reconstruct from the samples if we use ideal interpolation?

 # A digital communication link carries binary-coded words representing samples of an input signal,

$$x(t) = 3cos600\pi t + 2cos1800\pi t$$

The link is operated at 10,000 bits/s and each input sample is quantized into 1024 different voltage levels.

i. What is the sampling frequency & folding frequency?
ii. What is the Nyquist rate for the signal x(t) ?
iii. What are the frequencies in the resulting discrete time signal x[n] ?
iv. What is the resolution Δ ?

Solution: ➔

As the link is operated at 10,000 bits/s & each input sample is quantized into 1024 different voltage levels the each sampled value is represented by $\log_2 1024$ = 10 bits/sample

i. Then maximum sampling frequency, $F_s = \frac{10{,}000\ bits/sec}{10\ bits/sample} = 1000\ samples/sec$

Folding frequency (is the maximum frequency that can be represented uniquely by sampled signal), $\frac{F_s}{2} = 500\ samples/sec.$

ii. $x(t) = 3cos600\pi t + 2cos1800\pi t$

Here, F_1 = 300 Hz and F_2 = 900 Hz. Thus F_{max} = 900 Hz.

The Nyquist rate, F_N = 2Fmax = 1800 Hz.

iii. For F_s = 1000 Hz,

$$x[n] \cong x(nT) = x\left(\frac{n}{F_s}\right) = 3cos2\pi\left(\frac{300}{1000}\right)n + 2cos2\pi\left(\frac{900}{1000}\right)n$$

$$= 3cos2\pi\left(\frac{3}{10}\right)n + 2cos2\pi\left(\frac{9}{10}\right)n$$

$$= 3cos2\pi\left(\frac{3}{10}\right)n + 2cos2\pi\left(1 - \frac{1}{10}\right)n$$

$$= 3cos2\pi\left(\frac{3}{10}\right)n + 2cos2\pi\left(\frac{1}{10}\right)n$$

$$\therefore f_1 = \frac{3}{10} \& f_2 = \frac{1}{10}$$

$$Here\ both\ frequencies\ f_1 and\ f_2\ lies\ in\ the\ interval\ -\frac{1}{2} \le f \le \frac{1}{2}$$

iv. ADC resolution = 10 bits

Voltage resolution, $\Delta = \frac{x_{max} - x_{min}}{L-1} = \frac{5-(-5)}{1024-1} = 9.76\ mV$.

Some Trigonometric Identities:

$$\sin(2\pi + \theta) = sin\theta;\ \sin(2\pi - \theta) = -sin\theta$$

$$\sin(\pi \pm \theta) = \mp sin\theta$$

$$\cos(\pi \pm \theta) = -cos\theta$$

$$cos(2\pi \pm \theta) = cos\theta$$

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta = 2\cos^2\theta - 1 = 1 - 2\sin^2\theta$$

$$\sin(A \pm B) = sinAcosB \pm cosAsinB$$

$$\cos(A \pm B) = cosAcosB \mp sinAsinB$$

References:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 1.
2. S. Sharma, "Digital Signal Processing", Third Revised Edition, S.K. Kataria & Sons, 2007.
3. Analog Devices, "Mixed Signal and DSP Design Techniques", Prentice-hall 2000.

Chapter 2: Discrete-time Signals and Systems[i].

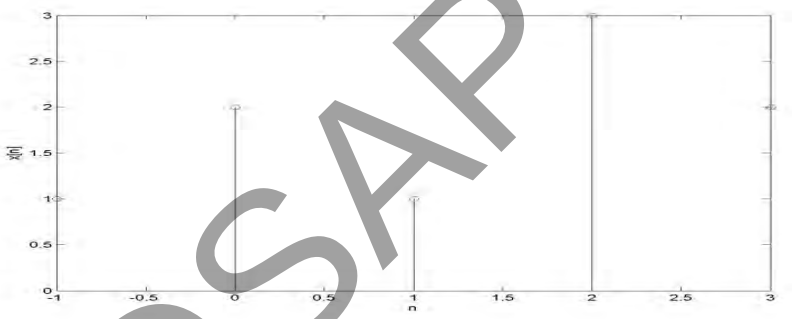
Discrete-time signals:

Digital signals are discrete in both time (the independent variable) and amplitude (the dependent variable). Signals that are discrete in time but continuous in amplitude are referred to as discrete-time signals. Discrete-time signals are data sequences. A sequence of data is denoted {x[n]} or simply x[n] when the meaning is clear. The elements of the sequence are called samples. The index n associated with each sample is an integer.

A discrete-time (D.T.) signals x[n] is a function of an independent variable that is an integer. It is important to note that a D.T. signal is not defined at instants between two successive samples. Also, it is incorrect to think that x[n] is equal to zero if n is not an integer. Simply, the signal x[n] is not defined for non-integer values of n.

Representation of D.T. Signals:

1) Graphical representation



2) Functional representation

$$x[n] = \begin{cases} 1 \ for\ n = -1, 1 \\ 2\ for\ n = 0, 3 \\ 3 \ for\ n = 2 \\ 0\ elsewhere \end{cases}$$

3) Tabular representation

n	-1	0	1	2	3
x[n]	1	2	1	3	2

4) Sequence representation

x[n] = {1, 2, 1, 3, 2} where " " *sign represents the position of n =0. The arrow is often omitted if it is clear from the context which sample is x[0].*

Sample values can either be real or complex. The terms "discrete-time signals" and "sequences" are used interchangeably.

Some Elementary D.T. signals:

1) The unit sample sequence or unit impulse is defined as δ[n] and is defined as

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

Note: The analog signal δ(t) is defined to be zero everywhere except at t=0 and has unit area.

2) The unit step signal is denoted as u[n] and is defined as

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

3) The unit ramp signal is denoted as $u_r[n]$ and is defined as

$$u_r[n] = \begin{cases} n, for\ n \geq 0 \\ 0, for\ n < 0 \end{cases}$$

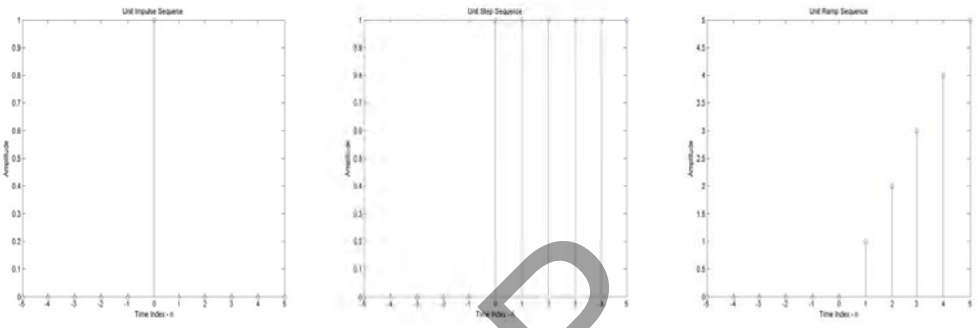


Fig: (a) Unit impulse (b) Unit Step (c) Unit Ramp sequences.

4) Sinusoidal Signal
x[n] = Acos(ωn + θ), $-\infty < n < \infty$
Where n is an integer variable, called the sample number, A is the *amplitude* of the sinusoid, ω is the frequency in radians per second, and θ is *phase* in radians.

5) Exponential Signal
The exponential signal is a sequence of the form
$x[n] = a^n$ for all n.
If the parameter a is real, then x[n] is a real signal. When the parameter a is complex valued,
$a = re^{i\theta}$ (r & θ are now parameters)
Hence, $x[n] = r^n e^{i\theta n} = r^n(\cos\theta n + j \sin\theta n)$
The real part is $x_R[n] = r^n\cos\theta n$ and the imaginary part is $x_I[n] = r^n\sin\theta n$
Alternatively, the complex signal x[n] can be represented by the amplitude function
$|x[n]| = A(n) = r^n$
/_x[n] = φ(n) = θn

Prove that

$$i.\, u[n] = \sum_{k=0}^{\infty} \delta[n-k]$$

$$ii.\, \delta[n] = u[n] - u[n-1]$$

Classification of Discrete-time signals:

1) Energy signals & Power signals:

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

If E is finite (i.e. $0 < E < \infty$), then x[n] is *energy signal.*

Many signals that possess infinite energy, have a finite average power. The average power of a periodic sequence with a period of N samples is defined as

$$P = \frac{1}{N}\sum_{n=0}^{N-1}|x[n]|^2$$

And for non-periodic sequences, it is defined in terms of the following limit if it exists:

$$P = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}|x[n]|^2$$

A signal with finite average power is called a power signal.

Find the average power of the unit step sequence u[n].

The unit step sequence is non-periodic, therefore the average power is

$$P = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}u^2[n]$$

$$= \lim_{N\to\infty}\left(\frac{N+1}{2N+1}\right) = \frac{1}{2}$$

Therefore the unit step sequence is a power signal. Note that its energy is infinite and so it is not an energy signal.

2) Periodic signals & Aperiodic signals:

The sinusoidal signal of the form

$x[n] = A\sin 2\pi f_0 n$ is periodic when f_0 is a rational number, that is if f_0 can be expressed as $f_0 = k/N$; where k and N are integers and N is the fundamental period.

3) Symmetric(even) and antisymmetric(odd) signals:

Even → $x[-n] = x[n]$ $x_e[n] = ½\{x[n] + x[-n]\}$

Odd → $x[-n] = -x[n]$ $x_o[n] = ½\{x[n] - x[-n]\}$

Transformation of the independent variable (time)

1) Shifting:

A signal x[n] may be shifted in time by replacing the independent variable n by n-k, where k is an integer. If k is positive, delay of the signal by k units of time and if k is negative, advance of the signal by k units in time.

2) Folding or a reflection of the signal about the time origin n=0:

A signal x[n] may be folded by replacing the independent variable n by –n.

Prove that the operations of folding and time shifting a signal are not commutative.

$$TD_k[x[n]] = x[n-k] \quad k>0$$

$$FD[x[n]] = x[-n]$$

Now, $TD_k[FD\{x[n]\}] = TD_k[x[-n]] = x[-n+k]$

Whereas, $FD[TD_k\{x[n]\}] = FD\{x[n-k]\} = x[-n-k]$

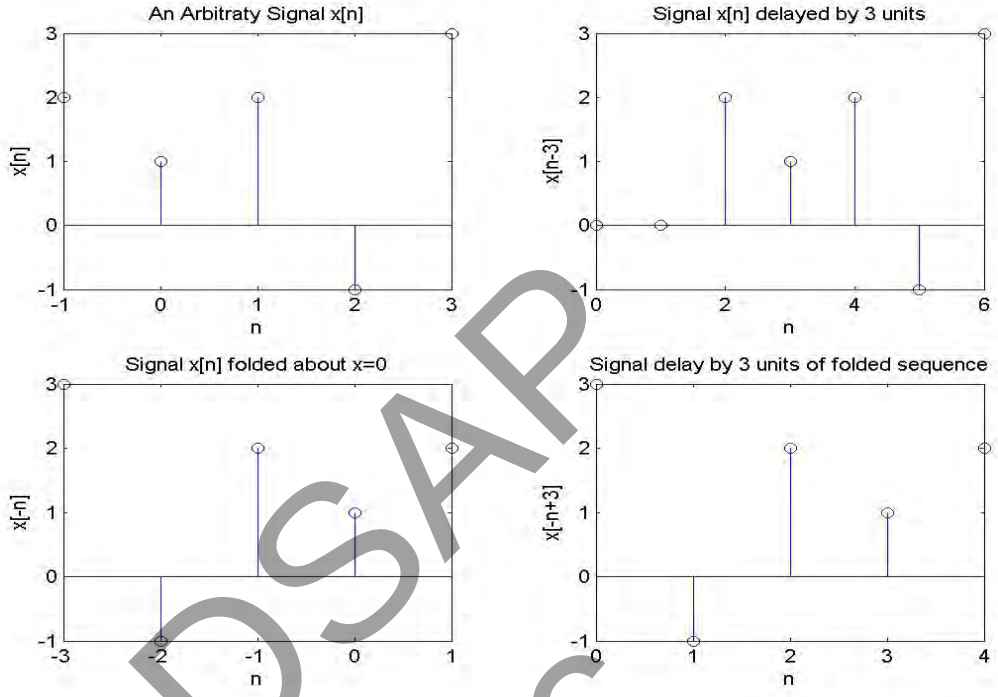
Note: Shifting of folded sequence x[-n] ---> x[-n+k]

If k positive, delay; if k negative, advance.

3) Time scaling or down-sampling:

-Replacing n by μn, where μ is an integer.

If the signal x[n] was originally obtained by sampling an analog signal $x_a(t)$, the $x[n] = x_a(nT)$, where T is the sampling interval. Now, $y[n] = x[2n] = x_a(2Tn)$. Hence the time-scaling operation is equivalent to changing the sampling rate from 1/T to 1/2T i.e. decreasing the rate by a factor of 2. This is a downsampling operation. Upsampling is not possible as we cannot obtain y[n] = x[n/2] from the signal x[n].



Note: Amplitude modification includes addition, multiplication, and scaling of D.T. signals.

Discrete-time Fourier series and properties:

In General Discrete-time Fourier series is given by,

$C_k = \frac{1}{N}\sum_{n=<N>} x[n]e^{-jr\left(\frac{2\pi}{N}\right)n}$	*Analysis equation*
$x[n] = \sum_{k=<N>} C_k\, e^{jk\omega_0 n}$	*Synthesis equation*

Where $\{C_k\}$ are spectral coefficients of x[n].

Properties of DTFS:

$$If x[n] \overset{F.S.}{\longleftrightarrow} C_k$$
$$\&y[n] \overset{F.S.}{\longleftrightarrow} D_k$$

Where x[n] and y[n] are periodic with period N & fundamental frequency $\omega_0 = 2\pi/N$. Also, C_k and D_k are periodic with period N.

There are strong similarities between the properties of discrete-time & continuous time Fourier series.

PROPERTY	PERIODIC SIGNAL	FOURIER SERIES
1. Linearity	Ax[n] + By[n]	$AC_k + BD_k$
2. Time-shifting	$x[n - n_0]$	$C_k e^{-\left(jk\frac{2\pi}{N}n_0\right)}$
3. Frequency-shifting	$e^{jM\left(\frac{2\pi}{N}\right)n}x[n]$	C_{k-M}
4. Conjugation	$x^*[n]$	C^*_{-k}
5. Time-reversal	$x[-n]$	C_k
6. Time scaling	$x_m[n] = \begin{cases} x\left[\frac{n}{m}\right] if is multiple of m \\ 0 \; if is not multiple of m \end{cases}$ (period =>mN)	$\frac{1}{m}C_k$ (period =>mN)
7. Periodic convolution	$\sum_{r=<N>} x[r]y[n-r]$	NC_kD_k
8. Multiplication	$x[n]y[n]$	$\sum_{l=<N>} C_l D_{k-l}$
9. First Difference	$x[n] - x[n-1]$	$\left\{1 - e^{-jk\left(\frac{2\pi}{N}\right)}\right\}C_k$

10. Parseval's Relation for discrete-time periodic signals:

Parseval's relation for discrete-time periodic signals (Power Density Spectrum) is given by

$$\frac{1}{N}\sum_{n=<N>}|x[n]|^2 = \sum_{k=<N>}|C_k|^2$$

Where C_k are the Fourier series coefficients of x[n] and N is period.
Parsevals relation states that, "The average power in a periodic signal is equal to the sum of average powers in all of its harmonic components".
If x[n] is real (i.e. $x[n] = x^*[n]$), then

$$C_k^* = C_{-k}$$
$$or, |C_{-k}| = |C_k| \;\; (Even\ symmetry)$$
$$\arg(C_k) = -\arg(C_{-k}) \;\; (Odd\ symmetry)$$

 # State and Prove time shifting property of DTFS
Solution: -> The DTFS is given by

$$C_k = \frac{1}{N}\sum_{n=<N>} x[n]e^{-jk\left(\frac{2\pi}{N}\right)n}$$

Time shifting property:

$$If x[n] \overset{F.S.}{\longleftrightarrow} C_k$$
$$Then x[n-l] \overset{F.S.}{\longleftrightarrow} C_k e^{-jk\left(\frac{2\pi}{N}\right)l}$$

Proof:

$$F.s.\{x[n-l]\} = \frac{1}{N}\sum_{n=<N>} x[n-l]e^{-jk\left(\frac{2\pi}{N}\right)n}$$

Let m = n – l, then

$$F.s.\{x[n-l]\} = \frac{1}{N}\sum_{m=<N>} x[m]e^{-jk\left(\frac{2\pi}{N}\right)(m+l)}$$
$$= \frac{1}{N}\sum_{m=<N>} x[m]e^{-jk\left(\frac{2\pi}{N}\right)m}e^{-jk\left(\frac{2\pi}{N}\right)l}$$

$$= e^{-jk\left(\frac{2\pi}{N}\right)l} \times \frac{1}{N} \sum_{m=<N>} x[m] e^{-jk\left(\frac{2\pi}{N}\right)m}$$

$$= e^{-jk\left(\frac{2\pi}{N}\right)l} C_k, \qquad Hence\ proved$$

When a periodic signal is shifted in time, the magnitudes of its Fourier series coefficients remain same.

Discrete-time Fourier transform and properties:

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

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

The discrete-time Fourier transform $X(e^{j\omega})$ is also known as spectrum of x[n]. The notations for Fourier transform pairs are:

$$F.T.\{x[n]\} = X(e^{j\omega})$$
$$F.T.^{-1}\{X(e^{j\omega})\} = x[n]$$
$$x[n] \overset{F.T.}{\Leftrightarrow} X(e^{j\omega})$$

Two important properties of DTFT are:

1. **Periodicity**

   The discrete-time Fourier transform $X(e^{j\omega})$ is continuous and periodic in ω with period 2π.

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

   Implication: We need only one period of $X(e^{j\omega})$ (i.e. $\omega \in [0, 2\pi], or\ [-\pi, \pi]$ for analysis and not the whole domain $-\infty < \omega < \infty$

2. **Symmetry:**

   For real valued x[n], $X(e^{j\omega})$ is conjugate symmetry.

   $$X(e^{-j\omega}) = X^*(e^{j\omega})$$
   $$|X(e^{-j\omega})| = |X(e^{j\omega})| => Even\ symmetry$$
   $$\measuredangle X(e^{-j\omega}) = -\measuredangle X(e^{j\omega}) \quad => Odd\ symmetry$$

Other properties of DTFT are:

PROPERTY	Finite Energy SIGNAL/s	FOURIER Transform
Linearity	$Ax_1[n] + Bx_2[n]$	$AX_1(e^{j\omega}) + BX_2(e^{j\omega})$
Time-shifting	$x[n - n_0]$	$e^{-(j\omega n_0)} X(e^{j\omega})$
Frequency-shifting	$e^{j\omega_0 n} x[n]$	$X(e^{j(\omega-\omega_0)})$
Conjugation	$x^*[n]$	$X^*(e^{-j\omega})$
Time-reversal	$x[-n]$	$X(e^{-j\omega})$
Differentiation	$x[n] - x[n-1]$	$(1 - e^{-j\omega}) X(e^{j\omega})$
Convolution	$x[n] * h[n]$	$X(e^{j\omega}) H(e^{j\omega})$
Accumulation	$\sum_{m=-\infty}^{\infty} x[m]$	$\frac{1}{1-e^{-j\omega}} X(e^{j\omega}) + \pi X(e^{j0}) \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$

Parseval's Relation	$\sum_{n=-\infty}^{\infty} \lvert x[n] \rvert^2 = \frac{1}{2\pi} \int_{2\pi} \lvert X(e^{j\omega}) \rvert^2 \, d\omega$

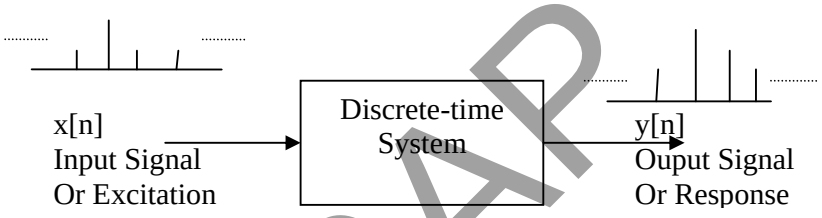
Discrete-time systems:

In many applications of digital signal processing we wish to design a device or an algorithm that performs some prescribed operation on a discrete-time signal. Such a device or algorithm is called a discrete-time system.

A discrete-time system is a device or algorithm that operates on a discrete-time signal, called the input or excitation, according to some well defined rule, to produce another discrete-time signal called the output or response of the system.

y[n] = T{x[n]}

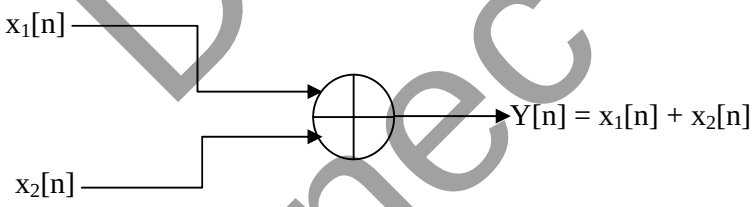
We say that the input signal x[n] is transformed by the system into a signal y[n].



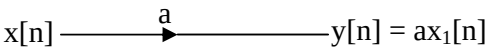
Block Diagram Representation of DT Systems:

(Basic building blocks that can be interconnected to form complex systems)

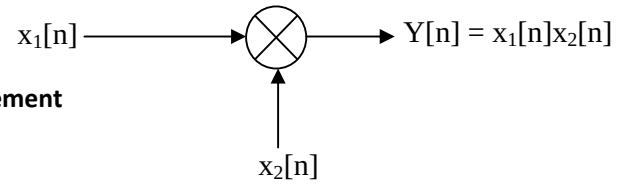
- **An Adder:**



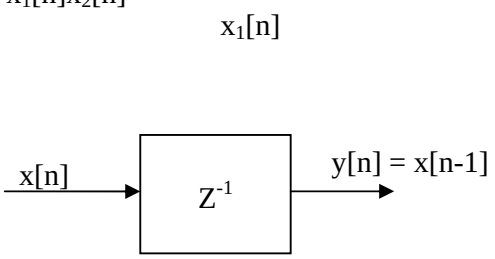
- **A Constant Multiplier**



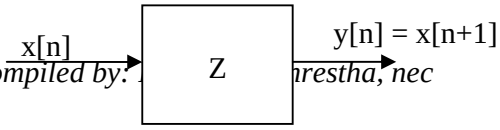
- **A Signal Multiplier**



- **A Unit delay element**



- **A Unit advance element**



#Use basic building blocks to represent following DT system described by the input-output relation:
$y[n] = 0.25y[n-1] + 0.5x[n] + 0.5x[n-1]$

Classification of Discrete-time systems:

1) Static (memoryless) vs dynamic (to have memory) systems:-

If the o/p of D.T. system at any instant n depends at most on the input sample at the same time, but not on past or future samples of the input then it is known as static system.
In any other case, the system is said to be dynamic.
$y[n] = x[n-N] \quad N \geq 0$
The system is said to have memory of duration N,
If N = 0, the system is static
$0<N<\infty$, the system is said to have finite memory.
$N = \infty$, the system is said to have infinite memory.

2) Time-invariant Vs. time-variant system:-

A system is called time-invariant if its input-output characteristics do not change with time.
Theorem:
A relaxed system T is time invariant or shift invariant if and only if $x[n] \rightarrow y[n]$
Implies $x[n-k] \rightarrow y[n-k]$
For every input signal x[n] and every time shift k.
For this we compute
$y[n, k] = T\{x[n-k]\} \rightarrow$ response of delayed input and
$y[n-k] \rightarrow$ delayed response
and check whether $y[n, k] = y[n-k]$ for all possible values of k.
if true $\rightarrow$ time invariant
& if $y[n, k] \neq y[n-k]$, even for one value of k, the system is time variant.

3) Linear Vs. nonlinear system:-

A linear system is one that satisfies the superposition theorem which states that, "A system is linear if and only if
$T\{a_1x_1[n] + a_2x_2[n]\} = a_1T\{x_1[n]\} + a_2T\{x_2[n]\}$
For any arbitrary i/p sequences $x_1[n]$ and $x_2[n]$, and any aribitrary constants a_1& a_2.

4) Causal Vs. noncausal systems:-

Theorem: "A system is said to be causal if the o/p of the system at any time n {i.e. y[n]} depends only on present and past inputs but doesnot depend on future inputs".
If a system doesnot satisfy this definition, it is called noncausal.

5) Stable Vs. unstable systems:-

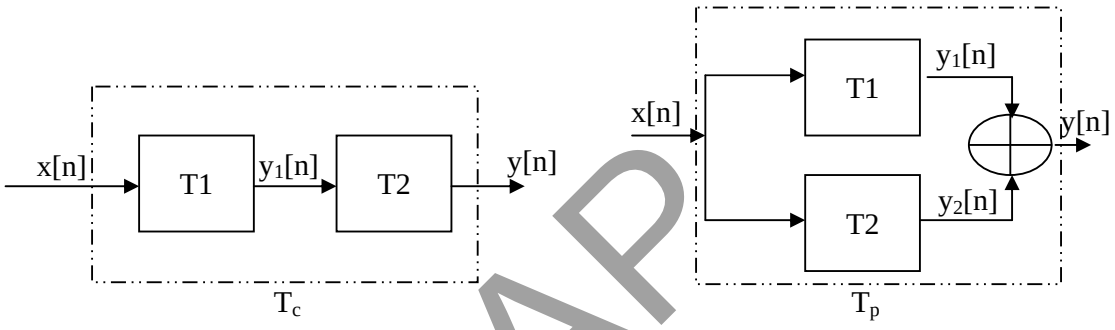
An arbitrary relaxed system is said to be bounded input bounded output (BIBO) stable if and only if every bounded input produces a bounded output.

Mathematically,
If, $|x[n]| \leq M_x < \infty$
Then there must be $|y[n]| \leq M_y < \infty$ for all n, for the system to be stable. Where, M_x and M_y are finite numbers.

Interconnection of Discrete-time systems:

Discrete time systems can be interconnected to form larger systems. There are two basics ways in which systems can be interconnected:

1. Cascade (Series)
2. Parallel



(a) Cascade Interconnection (b) Parallel Interconnection

In the cascade interconnection the output of the first systemis

$$y_1[n] = T_1\{x[n]\}$$

and the output of the second system is

$$y[n] = T_2\{y_1[n]\} = T_2[T_1\{x[n]\}]$$

We observe that system T_1 and T_2 can be combined or consolidated into a single overall system

$$T_c \equiv T_2T_1$$

We can express the output of the combined system as,

$$y[n] = T_c\{x[n]\}$$

In general, for arbitrary systems,

$$T_2T_1 \neq T_1T_2$$

However, if the systems T_1 and T_2 are linear and time-invariant, then (a) T_c is time invariant and (b) $T_2T_1 = T_1T_2$
Proof of (a)
Suppose that T_1 and T_2 are time-invariant, then
$x[n-k] \text{ — } (T_1) \rightarrow y_1[n-k]$
$y_1[n-k] \text{ —} (T_2) \rightarrow y[n-k]$
Thus, $x[n-k] \text{—} (T_c) \rightarrow y[n-k]$
And therefore, T_c is time invariant.
In the parallel interconnection, the output of the system T_1 is $y_1[n]$ and the output of the system T_2 is $y_2[n]$. Hence,
$y_3[n] = y_1[n] + y_2[n] = T_1\{x[n]\} + T_2\{x[n]\} = (T_1 + T_2)\{x[n]\} = T_p\{x[n]\}$.
Where, $T_p = T_1 + T_2$
In general, we can use parallel and cascade interconnection of systems to construct larger, more complex systems. Conversely, we can take a larger system and break it down into smaller subsystems for purposes of analysis and implementation.

Analysis of Discrete-time linear time-invariant(LTI) systems:

There are two basic methods for analyzing the behavior or response of a linear system to a given input signal.

- **Convolution method:**

  This method for analyzing the behavior of a linear system to a given input signal is first to decompose or resolve the input signal into a sum of elementary signals. Then, using the linearity property of the system, the responses of the system to the elementary signals are added to obtain the total response of the system to the given input signal.

- **Linear constant coefficient difference (LCCD) equation method:**

  This method is based on the direct solution of the input-output equation for the system, and has the form
  y[n] = F{y[n-1], y[n-1], ……, y[n-N], x[n], x[n-1], x[n-2], ……, x[n-M] }
  Specifically, for an LTI system, the general form of the input-output relationshiop is,

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

  Where $\{a_k\}$ and $\{b_k\}$ are constant parameters that specify the system and are independent of x[n] and y[n].

Resolution of a Discrete-time signal into impulses:

Suppose we have an arbitrary signal x[n] that we wish to resolve into a sum of unit sample sequences. We select the elementary signals $x_k[n]$ to be

$$x_k[n] = \delta[n-k]$$

where k represents the delay of the unit sample sequence.
Now suppose that we multiply the two sequences x[n] and δ[n-k], since δ[n-k] is zero everywhere except at n = k, where its value is unity, the result of this multiplication is another sequence that is zero everywhere except at n = k, where its value is x[k]. Thus
x[n]δ[n-k] = x[k]δ[n-k]
If we were to repeat the multiplication of x[n] with δ[n-m] where m is another delay (m≠k), the result will be a sequence that is zero everywhere except at n = m, where its value is x[m]. Hence,
x[n]δ[n-m] = x[m]δ[n-m]
Consequently, if we repeat this multiplication over all possible delays $-\infty < k < \infty$, and sum all the product sequences, the result will be a sequence equal to the sequence x[n], that is,

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

The right hand side of the equation gives the resolution of or decomposition of any arbitrary signal x[n] into a weighted (scale) sum of shifted unit samples sequences.
Response of LTI systems to arbitrary inputs: The Convolution Sum
We know,

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

First, we denote the response y[n ,k] of the system to the input unit sample sequence at n = k by the special symbol h[n, k]; $-\infty < k < \infty$, i.e.
y[n, k] = h[n, k] = T{δ[n-k]}
we note that n is the time index and k is a parameter showing the location of the input impulse.

If the impulse at the input is scaled by an amount $C_k = x[k]$, the response of the system is the corresponding scaled output, that is
$C_k h[n, k] = x[k]h[n, k]$
The response to any arbitrary input expressed as a sum of weighted impulses, as given above is
y[n] = T{x[n]} = T{

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

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

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

Where h(n,k) is the response of unit impulses δ[n-k] $-\infty < k < \infty$.
If the response of the LTI system to the unit sample sequence δ[n] is denoted by h[n], i.e.
h[n] ≡ T{δ[n]}
then by the time-invariant property, the response of the system to the delayed unit sample sequence δ[n-k] is,
h[n-k] = T{δ[n-k]}
then,

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

The formula in above equation that gives the response y[n] of the LTI system as a function of the input signal x[n] and the unit impulse h[n] is called a convolution sum.

Methods to calculate Convolution Sum:

1. Mathematically (by using direct formula of convolution sum).
2. Graphically (by using following procedure)
3. Matrix Method

Procedure to compute the convolution sum:
Suppose that we wish to compute the output of the system at some time instant say $n = n_0$, then

$$y[n_0] = \sum_{k=-\infty}^{\infty} x[k]h[n_0-k]$$

The procedures are:
Plot x[n] and h[n] as x[k] and h[k] then,

1) Folding: Fold h[k] about k = 0 to obtain h[-k].
2) Shifting: Shift h[-k] by n_0 to the right (left) if n_0 is positive (negative), to obtain $h[n_0-k]$.
3) Multiplication: Multiply x[k] by $h[n_0-k]$ to obtain the product sequence $v_{n0}(k) \equiv x[k]h[n_0-k]$.
4) Summation: Sum all the values of the product sequence $v_{n0}(k)$ to obtain the value of the output at time $n = n_0$.
5) Repetition: Repeat steps 2 through 4, for all possible time shifts $-\infty < n < \infty$ to obtain overall response.

Properties of Convolution:
To simplify the notation, we denote the convolution operations as,

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

1) Commutative Law: x[n] * h[n] = h[n] * x[n]
2) Associative Law: (x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] *h_2[n])
3) Distributive law: x[n]* (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] *h_2[n]

Causal Linear Time Invariant Systems:

Causal System: whose output at time n depends only on present and past inputs but does not depends on future inputs. i.e. Output of system x[n] at instant n = n_0 depends on values of x[n] for n ≤ n_0.

Causal LTI System:

Let us consider an LTI system having an output at time n = n_0 by

$$y[n_0] = \sum_{k=-\infty}^{\infty} h[k]x[n_0 - k]$$

Suppose that we subdivide the sum into two sets of term, one involving negative terms and other involving positive terms of values of k.

$$y[n_0] = \sum_{k=0}^{\infty} h[k]x[n_0 - k] + \sum_{k=-\infty}^{-1} h[k]x[n_0 - k]$$

={h[0]x[n_0] + h[1]x[n_0 - 1]+ h[2]x[n_0 - 2] + - - - - - - } + {h[-1]x[n_0 + 1]+ h[-2]x[n_0 +2] + - - - }

We observe that the terms in the first sum involve x[n_0], x[n_0 - 1], - - - which are the present and past values. On the other hand, the terms in the second sum involves the input signals x[n_0 + 1], x[n_0 + 2], - - -. Now, if the output at time n = n_0 is to depend only on the present and past values, then it is clear that

h[n] = 0 for n < 0

It is both a necessary and a sufficient condition for causality. Hence "*An LTI system is causal if and only if its impulse response is zero for negative values of n.*"

Thus for causal LTI system,

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

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

Stability of Linear Time Invariant Systems:

Stable System: An arbitrary relaxed system is BIBO stable if and only if its output sequence y[n] is bounded for every bounded input x[n].

For an LTI system:

Suppose we have an LTI system,

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

Taking absolute values on both sides,

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

$$\leq \sum_{k=-\infty}^{\infty} |h[k]||x[n-k]|$$

If the input is bounded, there exists a finite number M_x such that $|x[n]| \leq M_x$

$$|y[n]| \leq M_x \sum_{k=-\infty}^{\infty} |h[n]|$$

The output is bounded if the impulse response of the system satisfies the condition

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

"*A linear time-invariant system is stable if its impulse response is absolutely summable*". This is the necessary and sufficient condition for stable LTI system.

Systems with Finite-Duration and Infinite Duration Impulse Response:

Finite Duration Impulse Response System (FIR system):

h[n] = 0 for n < 0 and n ≥ M,

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

FIR has a finite memory of length M-samples.

Infinite Duration Impulse Response System (IIR System)

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

IIR has an infinite memory.

Discrete-time system described by Difference Equations:

The convolution summation formula is given by

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

Above equation suggests a means for the realization of the system. In the case of FIR systems, such a realization involves additions, multiplications, and a finite number of memory location, which is readily implemented directly.

If the system is IIR, however, its practical implementation as given by convolution is clearly impossible, since it requires an infinite number of memory locations, multiplications, and additions.

There is a practical and computationally efficient means for implementing a family of IIR systems, within the general class of IIR systems; this family of discrete-time systems is more conveniently described by difference equations.

Recursive & Non-Recursive Discrete-time systems:

If the response of any discrete-time systems depends only on the input signals (i.e. terms of the input signal) then the system is known as non-recursive discrete time system.

If we can express the output of the system not only in terms of the present and past values of the input, but also in terms of the past output values, then that system is known as recursive system.

Eg: the cumulative average of a signal x[n] in the interval $0 \leq k \leq n$ defined as,

$$y[n] = \frac{1}{n+1}\sum_{k=0}^{n} x[k] \qquad n = 0, 1, 2, \ldots..$$

The realization of above equation requires the storage of all the input samples. Since n is increasing, our memory requirements grow linearly with time.
However, y[n] can be computed more efficiently by utilizing the previous output value y[n-1]. By a simple algebraic rearrangement, we obtain,

$$(n+1)y[n] = \sum_{k=0}^{n-1} x[k] + x[n] = ny[n-1] + x[n]$$

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

Now the cumulative average y[n] can be computed recursively by multiplying the previous output value y[n-1] by n/(n+1), multiplying the present input x[n] by 1/(n+1), and adding the two products.
This is an example of recursive system. In general whose output y[n] at time n depends on any number of past output values y[n-1], y[n-2], ……, is called a recursive system.
The output of a causal and practically realizable recursive system can be expressed in general as,
y[n] = F{y[n-1], y[n-2], ……, y[n-N], x[n], x[n-1], ….., x[n], x[n-1], ….., x[n-M]}
If y[n] depends only on the present and past inputs, then
y[n] = F{ x[n], x[n-1], ….., x[n], x[n-1], ….., x[n-M]}
Such a system is called non-recursive.
In recursive system, we need to compute all the previous values y[0], y[1], y[2], …., $y[n_0 - 1]$ to compute $y[n_0]$ but in non-recursive system, we can compute the $y[n_0]$ immediately without having $y[n_0\text{-}1]$, $y[n_0\text{-}2]$, ……. This feature is desirable in some practical applications.

Difference Equations:

An LTI discrete-system can also be described by a linear coefficient difference equation of the form,

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \; a_0 \equiv 1$$

Or, equivalently,

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

If $a_N \neq 0$, then the difference equation is of order N. This equation describes a recursive approach for computing the current output, given the input values & previously computed output-values.
If the system described by difference equation has a constant coefficient (independent of time) then it is known as linear constant coefficient difference (LCCD) equation.
Consider the first order system (i.e. N = 1) & M = 0.
y[n] = ay[n-1] + x[n]
Now,
y[0] = ay[-1] + x[0]
$y[1] = ay[0] + x[1] = a^2y[-1] + ax[0] + x[1]$
$y[2] = ay[1] + x[2] = a^3y[-1] + a^2x[0] + ax[1] + x[2]$
⋮ ⋮ ⋮ ⋮
$y[n] = ay[n-1] + x[n] = a^{n+1}y[-1] + a^nx[0] + a^{n-1}x[1] + \ldots.$

$$= a^{n+1} y[-1] \sum_{k=0}^{n} a^k x[n-k] \qquad n \geq 0$$

the response contain two parts; the first part is the result of the initial condition y[-1] of the system and second part is the response of the system to the input signal x[n].

if the system is initially relaxed at time n = 0, then its memory should be zero. Hence y[-1] = 0 (state = output of the delay element).
In this case, we say that the system is at zero state and its corresponding output is called zero-state response or forced response and is denoted by y_{zs}.

$$y_{zs}[n] = \sum_{k=0}^{n} a^k x[n-k] \qquad n \geq 0$$

We note that above equation is a convolution summation involving the input signal convolved with the impulse response $h[n] = a^n u[n]$.
We obtained the result that the relaxed recursive system described by the first order difference equation is a linear time-invariant IIR system with impulse response given by $h[n] = a^n u[n]$.
Now, suppose the system is initially non-relaxed (i.e. $y[-1] \neq 0$) and the input x[n] = 0 for all n. the output of the system with zero input is called the zero-input response or natural response and is denoted by $y_{zi}[n]$.

$$y_{zi}[n] = a^{n+1} y[-1] \quad n \geq 0$$

We observe that a recursive system with nonzero initial condition is non relaxed in the sense that it can produce an output without being excited.
Linearity, time invariance and Stability of the system described by LCCD equation
A system is linear if it satisfies the following three requirements:

1. The total response is equal to the sum of the zero-input and zero-state responses. (i.e. $y[n] = y_{zi}[n] + y_{zs}[n]$)
2. The principle of superposition applies to the zero-state response.
3. The principle of superposition applies to the zero-input response.

Else the system is non-linear.
In general, recursive systems described by the constant-coefficient difference equation is linear and time-invariant. (because the coefficients a_k and b_k are constants)

References:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 2.

Chapter 3: Review of Z-Transform

Introduction:

Transform techniques are an important tool in the analysis of signals and linear time invariant (LTI) systems.

The Z-transform plays the same role in the analysis of discrete-time signals & LTI systems as the Laplace transform does in the analysis of continuous-time signals & LTI systems. Laplace transform can be developed as an extension of the continuous-time Fourier transform. This extension was motivated in part by the fact that it can be applied to a broader class of signals than the Fourier transform does not converge but the Laplace transform does. The Laplace transform allowed us, for example to perform transform analysis of unstable systems & to develop additional insights and tools for LTI system analysis.

The Direct Z-Transform:

The Z-transform of a discrete-time signal x[n] is defined as an infinite sum or infinite power series

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

where z is a continuous complex variable. This expression is generally referred to as the two-sided z-transform.

For convenience, the z-transform of a signal x[n] is denoted by

$$X(z) \equiv Z\{x[n]\}$$

Whereas the relationship between x[n] and X(z) is indicated by

$$x[n] \overset{z}{\longleftrightarrow} X(z)$$

Since the z-transform is an infinite power series, it exists only for those values of z for which this series converges. The region of convergence (ROC) of X(z) is the set of all values of z for which X(z) attains a finite value. Thus any time we cite a z-transform we should also indicate its ROC

Find z-transform of following signals:

1. $x_1[n] = \{1, 2, 1, 3, 4\}$ [Causal Signal]
2. $x_2[n] = \{1, 2, 1, 3, 4\}$ [Anti-causal signal]
3. $x_3[n] = \{1, 2, 1, 3, 4\}$ [Non-causal signal]

Z-Transform of some elementary signals:

1. x[n] = δ[n].

$$X(z) = \sum_{n=-\infty}^{\infty} \delta[n]z^{-n} = z^0 = 1 \qquad ROC: All\ z.$$

2. x[n] = δ[n-k].

$$X(z) = \sum_{n=-\infty}^{\infty} \delta[n-k]z^{-n} = z^{-k} \qquad ROC: All\ z\ except\ at\ z = 0.$$

3. x[n] = u[n]

$$X(z) = \sum_{n=-\infty}^{\infty} u[n]z^{-n}$$
$$= \sum_{n=0}^{\infty} z^{-n}$$
$$= \frac{1}{1-z^{-1}}\ if\ |z^{-1}| < 1$$
$$= \frac{1}{1-z^{-1}}, \qquad ROC: |z| > 1$$

4. x[n] = u[-n-1]

$$X(z) = \sum_{n=-\infty}^{\infty} u[-n-1]z^{-n}$$
$$= \sum_{n=-\infty}^{-1} z^{-n}$$
$$= \sum_{n=1}^{\infty} z^{n}$$
$$= \frac{z}{1-z}\ if\ |z| < 1$$
$$= -\frac{1}{1-z^{-1}}, \qquad ROC: |z| < 1$$

Analysis of ROC:

We can express the complex variable z in polar form as $z = re^{j\theta}$.
Then,

$$X(z)|_{z=re^{j\theta}} = \sum_{n=-\infty}^{\infty} x[n]\left(re^{j\theta}\right)^{-n}$$
$$= \sum_{n=-\infty}^{\infty} \{x[n]r^{-n}\}e^{-j\theta n}$$

The region of convergence (ROC) of X(z) is the set of all values of z for which X(z) attains a finite value. In the ROC of X(z),

$$|X(z)| < \infty, But$$
$$|X(z)| = \left|\sum_{n=-\infty}^{\infty} \{x[n]r^{-n}\}e^{-j\theta n}\right|$$
$$\leq \sum_{n=-\infty}^{\infty} \left|\{x[n]r^{-n}\}e^{-j\theta n}\right| \leq \sum_{n=-\infty}^{\infty} |x[n]r^{-n}|$$

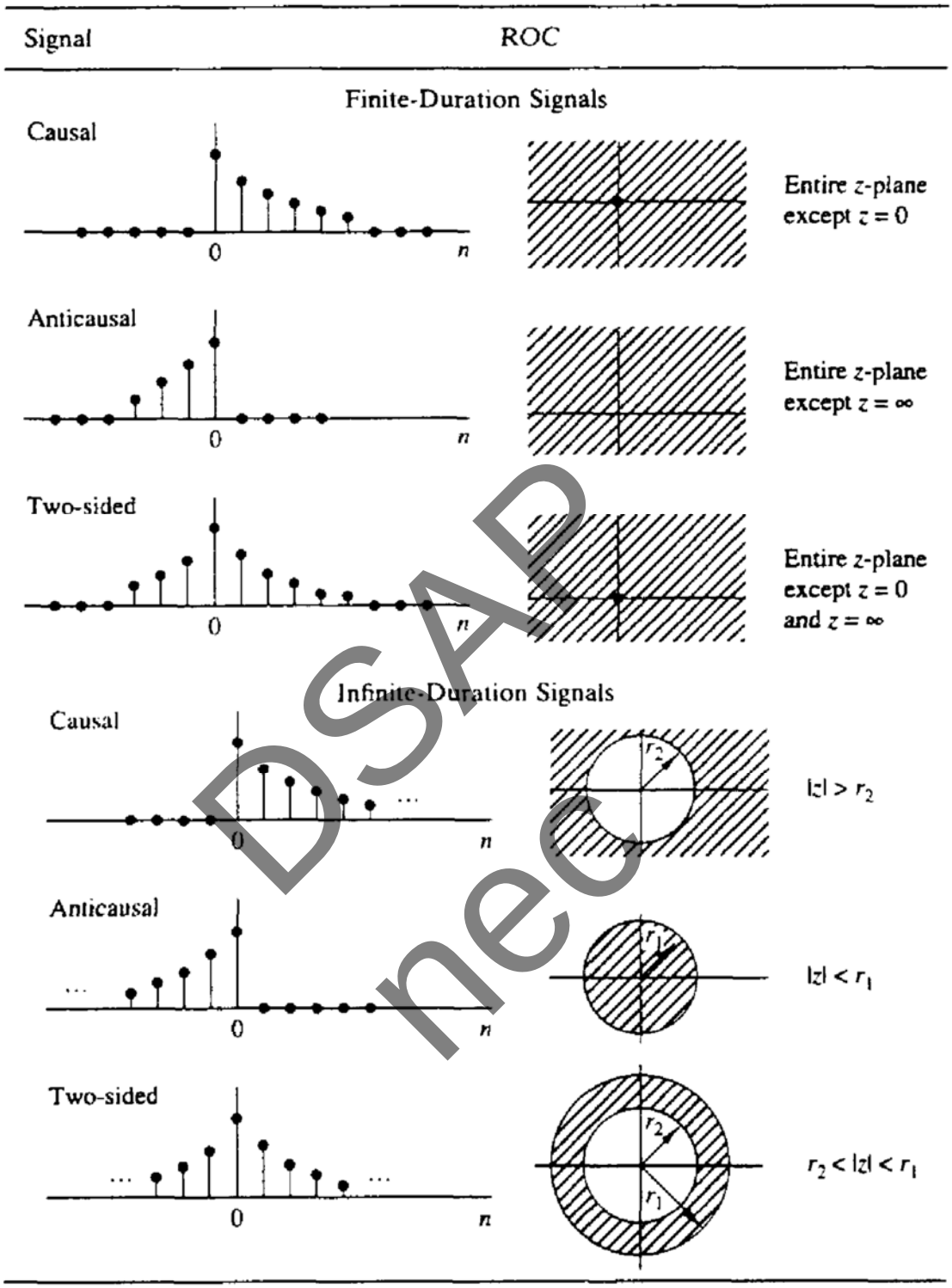


Figure : Characteristic families of signals with their corresponding ROC.

Hence $|X(z)|$ is finite if the sequence $x[n]r^{-n}$ is absolutely summable.
To elaborate again,

$$|X(z)| \leq \sum_{n=-\infty}^{-1} |x[n]r^{-n}| + \sum_{n=0}^{\infty} |x[n]r^{-n}|$$

$$\leq \sum_{n=1}^{\infty} |x[-n]r^{n}| + \sum_{n=0}^{\infty} |x[n]r^{-n}|$$

The first series, a non-causal sequence, converges for $|z| < r_2$, and the second series, a causal sequence, converges for $|z| > r_1$, resulting in an annular region of convergence.

Properties of Z-transform:

1. Linearity:

If $x_1[n] \overset{z}{\longleftrightarrow} X_1(z)$ & $x_2[n] \overset{z}{\longleftrightarrow} X_2(z)$, then

$$x[n] = ax_1[n] + bx_2[n] \overset{z}{\longleftrightarrow} aX_1(z) + bX_2(z)$$

For any constants a and b. ROC is intersection of the ROC of $X_1(z)$ and $X_2(z)$.

Find z-transform of $x[n] = (\cos\omega_0 n)u[n]$. (*hint: use Euler's identity*)

Ans:

$$X(z) = \frac{(1 - z^{-1}cos\omega_0)}{1 - 2z^{-1}cos\omega_0 + z^{-2}} \quad ROC: |z| > 1$$

2. Time Shifting:

If $x[n] \overset{z}{\longleftrightarrow} X(z)$ then $x[n-k] \overset{z}{\longleftrightarrow} z^{-k}X(z)$

ROC of $z^{-k}X(z)$ is the same as that of X(z) except for z = 0 if k>0 and z = ∞ if k<0.

The properties of linearity and time shifting are the key features that make the z-transform extremely useful for the analysis of discrete-time LTI systems.

3. Scaling in z-domain:

If $x[n] \overset{z}{\longleftrightarrow} X(z)$ ROC: $r_1 < |Z| < r_2$

Then $a^n x[n] \overset{z}{\longleftrightarrow} X(a^{-1}z)$ ROC: $|a|r_1 < |Z| < |a|r_2$

for any constant a, real or complex.

Proof:

$$Z\{a^n x[n]\} = \sum_{n=-\infty}^{\infty} a^n x[n] z^{-n} = \sum_{n=-\infty}^{\infty} x[n](a^{-1}z)^{-n} = X(a^{-1}z)$$

ROC is $r_1 < |a^{-1}z| < r_2$ ➔ $|a|r_1 < |z| < |a|r_2$

Eg:- $x[n] = a^n(cos\omega_0 n)u[n]$

$$We\ know, \quad Z\{cos\omega_0 n\, u[n]\} = \frac{1 - z^{-1}cos\omega_0}{1 - 2z^{-1}cos\omega_0 + z^{-2}}, \ ROC: |z| > 1$$

$$Then, Z\{a^n(cos\omega_0 n)\, u[n]\} = \frac{1 - az^{-1}cos\omega_0}{1 - 2az^{-1}cos\omega_0 + (az^{-1})^2}, \ ROC: |z| > |a|$$

4. Time Reversal:

If $x[n] \overset{z}{\longleftrightarrow} X(z)$ ROC: $r_1 < |Z| < r_2$

Then, $x[-n] \overset{z}{\longleftrightarrow} X(z^{-1})$ $\quad ROC: \frac{1}{r_2} < |z| < \frac{1}{r_1}$

Proof:

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

$$ROC: r_1 < |z^{-1}| < r_2 \Longrightarrow \frac{1}{r_2} < |z| < \frac{1}{r_1}$$

Eg:- $x[n] = a^{-n}u[-n]$

$$Ans: \frac{1}{1-az}, Roc: |z| < |a|$$

5. Differentiation in the z-domain:

If $x[n] \overset{z}{\longleftrightarrow} X(z)$ $\quad$ ROC: $r_1 < |Z| < r_2$

Then, $nx[n] \overset{z}{\longleftrightarrow} -z\frac{dX(z)}{dz}$ $\quad ROC: r_1 < |z| < r_2$

Proof:-

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

Differentiating both sides w.r.t. z

$$\frac{d}{dz}X(z) = \sum_{n=-\infty}^{\infty} x[n](-n)z^{-n-1} = -z^{-1}\sum_{n=-\infty}^{\infty}\{nx[n]\}z^{-n} = -z^{-1}Z\{nx[n]\}$$

Eg. Determine the signal x[n] whose z-transform is given by

$$X(z) = \log(1 + az^{-1}), \ \ |z| > |a|$$

$$Ans: nx[n] = (-1)^{n+1}a^n u[n-1]$$

$x[n] = n.a^n u[n]$

6. Convolution of two Sequences:

$$If, \ \ x_1[n] \overset{z}{\longleftrightarrow} X_1(z) \ \ \& \ \ x_1[n] \overset{z}{\longleftrightarrow} X_1(z)$$

$$then, x[n] = x_1[n] * x_2[n] \overset{z}{\longleftrightarrow} X(z) = X_1(z)X_2(z) \qquad ROC: R_{x_1} \cap R_{x_2}$$

Proof:

$$x[n] = x_1[n] * x_2[n] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[n-k]$$

$$X(z) = Z\{x_1[n] * x_2[n]\} = \sum_{n=-\infty}^{\infty} \{x_1[n] * x_2[n]\} z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x_1[k] x_2[n-k] z^{-n}$$

Upon interchanging the order of summation

$$X(z) = \sum_{k=-\infty}^{\infty} x_1[k] \sum_{n=-\infty}^{\infty} x_2[n-k] z^{-n}$$

Now, applying time shifting property

$$X(z) = \sum_{k=-\infty}^{\infty} x_1[k] z^{-k} X_2(Z) = X_1(z) X_2(z)$$

Computation of the convolution of two signals $x_1[n]$ & $x_2[n]$ using the z-transform:

(1) Find $X_1(z)$ & $X_2(z)$.
(2) Multiply $X_1(z)$ & $X_2(z)$.
(3) Find the inverse z-transform of multiplication.

This procedure is, in many cases, computationally easier than the direct evaluation of the convolution summation.

7. Correlation of two Sequences:

$$If,\ x_1[n] \overset{z}{\longleftrightarrow} X_1(z)\ \&\ x_1[n] \overset{z}{\longleftrightarrow} X_1(z)$$

$$then, r_{x_1x_2}(l) = \sum_{n=-\infty}^{\infty} x_1[n] x_2[n-l] \overset{z}{\longleftrightarrow} R_{x_1x_2}(z) = X_1(z) X_2(z^{-1})$$

$$ROC: -ROC\ of\ X_1(z) \cap ROC\ of\ X_2(z^{-1})$$

Proof:

$$r_{x_1x_2}(l) = x_1[l] * x_2[-l]$$

Using convolution property

$$R_{x_1x_2}(z) = Z\{x_1(l)\}.Z\{x_2(l)\}$$

Using time-reversal property

$$R_{x_1x_2}(z) = X_1(z) X_2(z^{-1})$$

8. Conjugate of a Complex Sequence:

$$x^*[n] \overset{z}{\longleftrightarrow} X^*(z^*)\quad ROC: ROC\ X(z)$$

9. Multiplication of two sequences:

$$If,\ x_1[n] \overset{z}{\longleftrightarrow} X_1(z)\ \&\ x_1[n] \overset{z}{\leftrightarrow} X_1(z)$$

$$then, x[n] = x_1[n]x_2[n] \overset{z}{\leftrightarrow} X(z) = \frac{1}{2\pi j}\oint_c X_1(v)X_2\left(\frac{z}{v}\right)v^{-1}dv$$

Where c is a closed contour that encloses the origin & lies within the ROC common to both $X_1(v)$ & $X_2(1/v)$. ROC: $r_{1l}r_{2l} < |z| < r_{1u}r_{2u}$

10. Parseval's Relation:

If $x_1[n]$ and $x_2[n]$ are complex-valued sequences, then

$$\sum_{n=-\infty}^{\infty} x_1[n]x_2^*[n] = \frac{1}{2\pi j}\oint_c X_1(v)X_2^*\left(\frac{1}{v^*}\right)v^{-1}dv$$

Provided that $r_{1l}r_{2l} < 1 < r_{1u}r_{2u}$, where $r_{1l} < |z| < r_{1u}$ and $r_{2l} < |z| < r_{2u}$ are the ROC of $X_1(z)$ and $X_2(z)$.

11. Initial Value Theorem:

If x[n] is causal [i.e. x[n] = 0 for n < 0], then

$$x[0] = \lim_{z \to \infty} X(z)$$

Proof:

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

$$\lim_{z \to \infty} X(z) = \lim_{z \to \infty} x[0] + x[1]z^{-1} + x[2]z^{-2} + \cdots = x[0]$$

12. Final Value Theorem:

If x[n] is causal, then

$$\lim_{n\to\infty} x[n] = \lim_{z\to\infty}(z-1)X(z)$$

Note: All the poles of X(z) should lie inside the unit circle.

Rational Z-Transforms:

Poles & Zeros:

The zeros of a z-transform X(z) are the values of z for which X(z) = 0. The poles of a z-transform are the values of z for which X(z) = ∞. If X(z) is a rational function then,

$$X(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1z^{-1} + \cdots + b_Mz^{-M}}{a_0 + a_1z^{-1} + \cdots + a_Nz^{-N}} = \frac{\sum_{k=0}^{M} b_kz^{-k}}{\sum_{k=0}^{N} a_kz^{-k}}$$

If $a_0 \neq 0$ and $b_0 \neq 0$, we can avoid the negative powers of z by factoring out the terms b_0z^{-M} & a_0z^{-N} as follows:

$$X(z) = \frac{N(z)}{D(z)} = \frac{b_0 z^{-M}}{a_0 z^{-N}} \cdot \frac{z^M + \left(\frac{b_1}{b_0}\right) z^{M-1} + \cdots + b_M/b_0}{z^N + \left(\frac{a_1}{a_0}\right) z^{M-1} + \cdots + a_M/a_0}$$

Since N(z) & D(z) are polynomials in z, they can be expressed in factored form as:

$$X(z) = \frac{N(z)}{D(z)} z^{-M+N} \cdot \frac{(z-z_1)(z-z_2)\ldots(z-z_M)}{(z-p_1)(z-p_2)\ldots(z-p_N)} = Gz^{N-M} \cdot \frac{\prod_{k=1}^{M}(z-z_k)}{\prod_{k=1}^{N}(z-p_k)}$$

Where, $G \equiv b_0/a_0$. Thus X(z) has

(a) M finite zeros at $z = z_1, z_2, z_3, \ldots, z_M$ (the roots of the numerator polynomials).

(b) M finite poles at $z = p_1, p_2, p_3, \ldots, p_N$ (the roots of the denominator polynomials).

(c) |N-M| zeros (if N > M) or poles (if N < M) at z = 0.

Note:

- A zero exists at $z = \infty$ if $X(\infty) = 0$ & a pole exists at $z = \infty$ if $X(\infty) = \infty$.
- If we count the poles and zeros at zero & infinity, we find that X(z) has exactly the same number of poles and zeros.
- For pole-zero plot we denote the location of poles by '×' (cross) sign and location of zeros by 'o' (circle) sign.

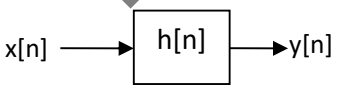
Obviously, by definition, the ROC of a z-transform should not contain any poles.

Pole location and time-domain behavior for causal signals

1. Causal real signals with simple real poles or simple complex-conjugate pairs of poles, which are inside or on the unit circle, are always bounded in amplitude.
2. A signal with a pole (or a complex-conjugate pair of poles) near the origin decays more rapidly than one associated with a pole near (but inside) the unit circle.
3. A double real pole on the unit circle results in an unbounded signal.
4. If a pole of a system is outside the unit circle, the impulse response of the system becomes unbounded and consequently, the system is unstable.

The System Function of an LTI System:

We know, for an LTI system



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

Taking z-transform, Y(z) = X(z)H(z) => H(z) = Y(z)/X(z)

If we know the x[n] and we observe the output y[n] of the system, we can determine the unit sample response. It is clear that H(z) represents the z-domain characterization of a system, whereas h[n] is the corresponding time-domain characterization of the system. The transform function H(z) is called the system function.

When the system is described by an LCCD equation,

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

Taking z-transform

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

$$Simplifying\ we\ get, \qquad H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Hence, and LTI system described by a LCCD equation has a rational system function.

Case I: If $a_k = 0$ for $1 \le k \le N$

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \frac{1}{z^M} \sum_{k=0}^{M} b_k z^{M-k}$$

In this case, H(z) contains M zeros, whose values are determined by system parameters $\{b_k\}$ & an M^{th} order pole at z=0 (trivial poles). Hence the system is called all-zero system. Such a system has finite duration impulse response & is called FIR system.

Case II: If $b_k = 0$ for $1 \le k \le M$,

$$H(z) = \frac{b_0}{\sum_{k=1}^{N} a_k z^{-k}} = \frac{b_0 z^N}{\sum_{k=1}^{N} a_k z^{N-k}},\ a_0 \cong 1$$

In this case H(z) consists of N poles, whose values are determined by the system parameters $\{a_k\}$ & and N^{th} order zero at z = 0. The corresponding system is called all-pole system. Due to presence of poles, the impulse response of such a system is infinite in duration & hence it is an IIR system.

Transfer Function Representation:

$$H(z) = b_0 z^{N-M} \left(\prod_{l=1}^{N} (z - z_l) \right) / \left(\prod_{k=1}^{N} (z - p_k) \right)$$

If the ROC of H(z) includes a unit circle ($z = e^{j\omega}$), then we can evaluate H(z) on unit circle, resulting in a frequency response function or transfer function $H(e^{j\omega})$

$$H(e^{j\omega}) = b_0 e^{j(N-M)\omega} \left(\prod_{l=1}^{N} (e^{j\omega} - z_l) \right) / \left(\prod_{k=1}^{N} (e^{j\omega} - p_k) \right)$$

Here magnitude response function,

$$|H(e^{j\omega})| = |b_0| \times \frac{|e^{j\omega} - z_1||e^{j\omega} - z_2| \ldots . |e^{j\omega} - z_M|}{|e^{j\omega} - p_1||e^{j\omega} - p_2| \ldots . |e^{j\omega} - p_N|}$$

The phase response function,

$$Arg\left(H(e^{j\omega})\right) = \{0\ or\ \pi\} + (N - M)\omega + \sum_{k=1}^{M} Arg(e^{j\omega} - z_k) - \sum_{k=1}^{N} Arg(e^{j\omega} - p_k)$$

Inversion of Z-Transform:

1. **Direct evaluation by Contour integration (not in syllabus).**
2. **Expansion into a series of terms in the variables z and z^{-1}.**
3. **Partial fraction expansion and table lookup.**

Long-Division Method:

Determine the inverse z-transform of

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}}$$

When (a) ROC: |z| > 1. (b) ROC: |z| < 0.5.

Solution: (a) Since the ROC is the exterior of a circle, we expect x[n] to be a causal signal. Thus we seek a power series expansion in negative powers of z.

$$X(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}}$$

$1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}$	1 $1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}$	$1 - \frac{3}{2}z^{-1} + \frac{7}{4}z^{-2} + \cdots$
	$\frac{3}{2}z^{-1} - \frac{1}{2}z^{-2}$ $\frac{3}{2}z^{-1} - \frac{9}{4}z^{-2} + \frac{3}{4}z^{-3}$	

$$X(z) = 1 + \frac{3}{2}z^{-1} + \frac{7}{4}z^{-2} + \frac{15}{8}z^{-3} + \cdots$$

$$\therefore x[n] = \left\{1, \underset{\uparrow}{\frac{3}{2}}, \frac{7}{4}, \frac{15}{8}, \cdots\right\}.$$

(b) In this case ROC is the interior of a circle. The signal x[n] is anticausal. So, we seek a power expension in powers of z, so we perform the long division in the following way:

$$X(z) = 2z^2 + 6z^3 + 14z^4 + \cdots$$

$$\therefore x[n] = \left\{\cdots, 30, 14, 6, 2, 0, \underset{\uparrow}{0}\right\}$$

This method is used only if one wished to determine the values of the first few samples of the signal.

The Inverse Z-Transform by Partial Fraction Expansion:

In the table lookup method, we attempted to express the function X(z) as a linear combination.

$$X(z) = \alpha_1 X_1(z) + \alpha_2 X_2(z) + \cdots + \alpha_k X_k$$

Then the inverse z-transform of X(z), can be found using the linearity property as,

$$x[n] = \alpha_1 x_1[n] + \alpha_2 x_2[n] + \cdots + \alpha_k x_k[n]$$

This approach is useful if X(z) is a rational function,

$$X(z) = \frac{N(z)}{D(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \cdots (A)$$

$$= \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}, \quad a_0 \cong 1$$

Note: If $a_0 \neq 1$, divide N(z) & D(z) by a_0.

A rational function of the form (A) is called proper, if $a_N \neq 0$ and M < N (finite zeros is less than the number of finite poles).

An improper rational function (M ≥ N) can always be written as the sum of a polynomial & a proper rational function.

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

Where $a_N \neq 0$ and M < N

Eliminating negative powers of z

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

Since N > M, the function

$$\frac{X(z)}{z} = \frac{b_0 z^{N-1} + b_1 z^{N-2} + \ - - - \ + b_M z^{N-M-1}}{z_N + a_1 z^{N-1} + \ - - - \ + a_N}$$

Is also always proper.

Now the next step is to express X(z)/z as a sum of simple fractions.

Two cases:

(I) Distinct Poles:

Suppose that the poles $p_1, p_2, ---, p_N$ are all different (distinct)

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

The problem is to determine the coefficients $A_1, A_2, ---, A_N$.

$$A_k = \frac{(z - p_k)X(z)}{z}\Big|_{z = p_k}, \quad k = 1,2,3, --, N$$

(II) Multiple order poles:

$$\frac{X(z)}{z} = \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2} + \frac{A_3}{(z - p_2)^2}$$

$$A_1 = \frac{(z - p_1)X(z)}{z}\Big|_{z = p_1}$$

$$A_3 = \frac{(z - p_2)^2 X(z)}{z}\Big|_{z = p_2}$$

$$A_2 = \frac{d}{dz}\frac{(z - p_2)^2 X(z)}{z}\Big|_{z = p_2}$$

In terms of z^{-1}

$$X(z) = \frac{A_1}{1 - p_1 z^{-1}} + \frac{A_2}{1 - p_2 z^{-1}} + \frac{A_3}{(1 - p_2 z^{-1})^2}$$

$$A_1 = (1 - p_1 z^{-1})X(z)\Big|_{z = p_1}$$

$$A_3 = (1 - p_2 z^{-1})^2 X(z)\Big|_{z = p_2}$$

$$A_2 = \frac{d}{dz}(1 - p_2 z^{-1})^2 X(z)\Big|_{z = p_2}$$

Determine the z-transform of

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}}$$

If

(a) ROC: |z| > 1

(b) ROC: |z| < 0.5

(c) ROC: 0.5 < |z| < 1.

Solution: -→

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}} = \frac{1}{1 - z^{-1} - 0.5z^{-1} + 0.5z^{-2}} = \frac{1}{(1 - z^{-1})(1 - 0.5z^{-1})}$$
$$= \frac{A}{1 - z^{-1}} + \frac{B}{1 - 0.5z^{-1}}$$

Solving, we get A = 2 and B = -1.

$$Ans\ (a): x[n] = 2u[n] - 0.5^n u[n]$$
$$Ans\ (b): x[n] = -2u[-n-1] + 0.5^n u[-n-1]$$
$$Ans\ (c): x[n] = -2u[-n-1] - 0.5^n u[n]$$

Causality and Stability:

Causality:-

We know, a causal LTI system is one whose unit sample response satisfies the condition

h[n] = 0 n<0

We also know that the ROC of the z-transform of a causal sequence is the exterior of a circle. Consequently, a linear time invariant system is a causal if and only if the ROC of the system function is the exterior of a circle of radius r<∞, including the point z=∞.

Stability:-

A necessary and sufficient condition for a LTI system to be BIBO stable is,

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

In turn, this condition implies that H(z) must contain the unit circle within its ROC.
Indeed since

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

It follows that,

$$|H(z)| \leq \sum_{n=-\infty}^{\infty} |h[n] z^{-n}|$$
$$\leq \sum_{n=-\infty}^{\infty} |h[n]||z^{-n}|$$

When evaluated on the unit circle i.e. |z| = 1

$$|H(z)| \leq \sum_{n=-\infty}^{\infty} |h[n]|$$

Hence, if the system is BIBO stable, the unit circle is contained in the ROC of H(z). The converse is also true. Therefore, a linear time invariant system is BIBO stable if and only if the ROC of the system function includes the unit circle.
A causal linear time invariant system (ROC: |z|>r<1) is BIBO stable if and only if all the poles of H(z) are inside the unit circle.

 # Example

A LTI system is characterized by the system function,

$$H(z) = \frac{3 - 4z^{-1}}{1 - 3.5z^{-1} + 1.5z^{-2}}$$

Specify the ROC of H(z) and determine h[n] for the following conditions:

(a) The system is stable.
(b) The system is causal.
(c) The system is anticausal.

Solution →

The given system is

$$H(z) = \frac{3 - 4z^{-1}}{1 - 3.5z^{-1} + 1.5z^{-2}} = \frac{1}{1 - {}^{1}/_{2}\, z^{-1}} + \frac{2}{1 - 3z^{-1}}$$

The system has poles at z = ½ and z = 3.

(a) Since the system is stable, its ROC must include the unit circle & hence it is ½<|z|<3.
Consequently h[n] is non-causal and is given as,
$h[n] = (½)^n u[n] - 2(3)^n u[-n-1]$.

(b) Since the system is causal, its ROC is |z| > 3. In this case,
$h[n] = (½)^n u[n] + 2(3)^n u[n]$.
This system is unstable.

(c) If the system is anticausal, its ROC is |z| < 0.5 hence
$h[n] = -[(½)^n + 2(3)^n] u[-n-1]$.
In this case the system is unstable.

Partial Fraction Expansion for multiple order poles:

$$\frac{X(z)}{z} = \sum_{p} \frac{A_p}{Z - Z_p} + \sum_{q} \sum_{k=1}^{M_q} \frac{B_{qk}}{(z - z_q)^k}$$

Where z_p is the p^{th} single pole and z_q is the q^{th} multiple pole of the order M_q. The constants A_p and B_{qk} are given as follows:

$$A_p = (z - z_p) X(z) \Big|_{z = z_p}$$

$$B_{qk} = \frac{1}{(M_q - k)!} \cdot \frac{d^{M_q - k}}{dz^{M_q - k}} (z - z_q)^{M_q} X(z) \Big|_{z = z_q}$$

	Signal, $x(n)$	z-Transform, $X(z)$	ROC
1	$\delta(n)$	1	All z
2	$u(n)$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
3	$a^n u(n)$	$\frac{1}{1-az^{-1}}$	$\|z\| > \|a\|$
4	$na^n u(n)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| > \|a\|$
5	$-a^n u(-n-1)$	$\frac{1}{1-az^{-1}}$	$\|z\| < \|a\|$
6	$-na^n u(-n-1)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| < \|a\|$
7	$(\cos \omega_0 n)u(n)$	$\frac{1-z^{-1}\cos\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
8	$(\sin \omega_0 n)u(n)$	$\frac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
9	$(a^n \cos \omega_0 n)u(n)$	$\frac{1-az^{-1}\cos\omega_0}{1-2az^{-1}\cos\omega_0+a^2z^{-2}}$	$\|z\| > \|a\|$
10	$(a^n \sin \omega_0 n)u(n)$	$\frac{az^{-1}\sin\omega_0}{1-2az^{-1}\cos\omega_0+a^2z^{-2}}$	$\|z\| > \|a\|$

Figure : Common Z-transform pairs.

Reference:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 3.

Chapter 4: DISCRETE FOURIER TRANSFORM (DFT)

INTRODUCTION:

Modern signal processing using computer is discrete processing moreover digital processing. So, it is necessary to represent either time domain or frequency domain signal by their samples rather than continuous function.

Frequency analysis of discrete time signal x[n] invokes Fourier transform resulting the continuous function of w i.e. $X(e^{jw})$. A difficulty encounter with the direct application of Fourier transform to a discrete time signal is that the resulting representation $X(e^{jw})$ becomes continuous function of frequency. Hence, it is unsuitable for digital processing.

The simple solution is the representation of a discrete time signal x[n] by samples of its spectrum $X(e^{jw})$. Such a frequency domain sampling leads to the Discrete Fourier Transform (DFT), powerful computational tool for performing frequency analysis of discrete time signal. A second transform domain representation that is applicable only to a finite length sequences is the DFT. The DFT is itself a sequence rather than a function of a continuous variable, and it corresponds to samples, equally spaced in frequency, of the Fourier transform of the signal. Not only its importance as a Fourier representation of sequences, the DFT plays a central role in the variety of DSP algorithms. Because there are variety of efficient algorithms for DFT computation.

Fourier analysis of different signals:

S. N.	Signal types	Fourier Analysis	Synthesis Equation	Analysis Equation
1	Continuous time periodic signal	CTFS	$x(t) = \sum_{k=-\infty}^{\infty} C_k e^{jk2\pi F_0 t}$	$C_k = \frac{1}{T}\int_{T_P} x(t)\, e^{-jk2\pi F_0 t} dt$
2	Discrete-time periodic signal	DTFS	$x[n] = \sum_{k=0}^{N-1} C_k e^{j2\pi kn/N}$	$C_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$

3	Continuous time aperiodic signal	CTFT	$x(t) = \int_{-\infty}^{\infty} X(F)\, e^{j2\pi Ft} dF$	$X(F) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi Ft} dt$
4	Discrete-time aperiodic signal	DTFT	$x[n] = \frac{1}{2\pi} \int_{0}^{2\pi} X(\omega)\, e^{j\omega n} d\omega$	$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$

Difference between DTFT and DFT

DFT	DTFT
1. Obtained by performing sampling in both the time & frequency domains.	1. Sampling is performed only in time domain
2. Discrete frequency spectrum	2. Continuous function of ω
3. The DFT can be applied only to finite length sequences.	3. The DTFT is applicable to any arbitrary sequences.

Frequency Domain Sampling & Reconstruction of Discrete-Time Signal

Let us consider a discrete-time signal (aperiodic) x[n]. It's Fourier transform is given by

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

Here $X(e^{j\omega})$ is continuous. Suppose we sample $X(\omega)$ periodically in frequency at a spacing of $\delta\omega$ radians between successive samples. Since $X(\omega)$ is periodic with period 2π, only samples in the fundamental frequency range are necessary. We take N equidistance samples in the interval $0 \leq \omega \leq 2\pi$ with spacing $\delta\omega = 2\pi/N$ as shown in figure.

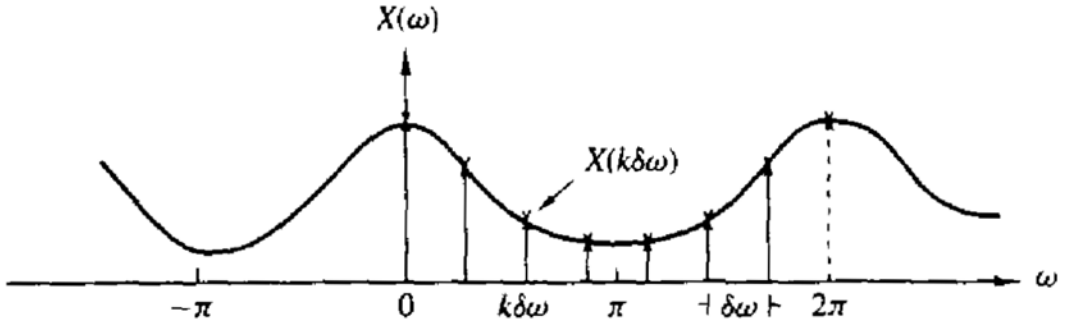


Figure 1: Frequency-domain sampling of the Fourier transform.

If we evaluate Eqn (i), at ω = 2πk/N

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\frac{2\pi kn}{N}}, k = 0,1,2,..,N-1 \qquad \textit{ii}$$

It can be subdivided into an infinite number of summations, where each sum contains N terms. Thus,

$$X\left(\frac{2\pi}{N}k\right) = \cdots + \sum_{n=-N}^{-1} x[n]e^{-j\frac{2\pi kn}{N}} + \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}} + \sum_{n=N}^{2N-1} x[n]e^{-j\frac{2\pi kn}{N}} + \cdots \qquad \textit{iii}$$

$$= \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x[n]e^{-j\frac{2\pi kn}{N}}$$

Changing the index in the inner summation from n to n-lN and by interchanging the order of summation

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1}\left[\sum_{l=-\infty}^{\infty} x[n-lN]\right]e^{-j\frac{2\pi kn}{N}}, for\ k = 0,1,2,\ldots,N-1 \qquad \textit{iv}$$

The obtained signal in the bracket of Eqn (iv) is the periodic repeatition of x[n] every N samples. Now it can be expanded in a Fourier series as,

$$x_p[n] = \sum_{k=0}^{N-1} C_k e^{j\frac{2\pi kn}{N}}, \qquad n = 0,1,2,\dots,N-1 \qquad \textit{v}$$

With Fourier coefficients

$$C_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p[n] e^{-j\frac{2\pi kn}{N}}, \qquad k = 0,1,2,\dots,N-1 \qquad \textit{vi}$$

Upon comparing Eqns (iv) & (vi)

$$C_k = \frac{1}{N} X\left(\frac{2\pi}{N}k\right), \qquad k = 0,1,2,\dots,N-1 \qquad \textit{vii}$$

Therefore,

$$x_p[n] = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j\frac{2\pi kn}{N}}, \qquad n = 0,1,2,\dots,N-1 \qquad \textit{viii}$$

Which provides the reconstruction of the periodic signal $x_p[n]$ from the samples of the spectrum X(ω).

For $0 \le n \le L\text{-}1$ & $N \ge L$, $x[n] = x_p[n]$ $\qquad 0 \le n \le N\text{-}1$

If N < L, it is not possible to recover x[n] from its periodic extension due to time-domain aliasing. Thus we conclude that the spectrum of an aperiodic discrete-time signal with finite duration L can be exactly recovered from its samples at frequencies $\omega_k = 2\pi k/N$.

The Discrete Fourier Transform (DFT)

A finite duration sequence x[n] of length L has a Fourier transform

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

Where the upper and lower indices in the summation reflect the fact that x[n] = 0 outside the range $0 \le n \le L\text{-}1$. When we sample X(ω) at equally spaced frequencies $\omega_k = 2\pi k/N$, k = 0, 1, 2, …, N-1, where $N \ge L$ the resultant samples are,

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

Since, x[n] = 0 for n ≥ L. The relation in above equation is called the discrete Fourier transform of x[n] as it is obtained by sampling the Fourier transform X(ω) at a set of N (equally spaced) discrete frequencies.

The inverse DFT, which allows us to recover the sequence x[n] from the frequency samples is given by

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

Clearly, when x[n] has length L < N, the N-point IDFT yields x[n] = 0 for L ≤ n ≤ N-1.

$$DFT: X(k) = \sum_{n=0}^{N-1} x[n]\,W_N^{kn}, \qquad k = 0,1,2,\dots,N-1$$

$$IDFT: x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn}, \qquad n = 0,1,2,\dots,N-1$$

Where by definition,

$$W_N = e^{-j\frac{2\pi}{N}}$$

Which is an N[th] root of unity.

Properties of DFT

The notation to denote the N-point DFT pair x[n] and X(k) is

$$x[n] \underset{N}{\overset{DFT}{\longleftrightarrow}} X(k)$$

The properties of the DFT are useful in the practical techniques for processing signals. The various properties are given below:

1. **Periodicity:**

   If X(k) is an N-point DFT of x[n], then

   x[n+N] = x[n] for all n

   X(k+N) = X(k) for all k

These periodicities in x[n] and X(k) follow immediately from above formulae for the DFT and IDFT respectively.

We had not viewed the DFT X(k) as a periodic sequence. In some applications it is advantageous to do this.

2. Linearity:

If $X_1(k)$ and $X_2(k)$ are the N-point DFTs of $x_1[n]$ and $x_2[n]$ and a & b are arbitrary constants (real or complex valued), then,

$$ax_1[n] + bx_2[n] \overset{DFT}{\underset{N}{\leftrightarrow}} aX_1(k) + bX_2(k)$$

3. Shifting property:

Let $x_p[n]$ is a periodic sequence with period N which is obtained by extending x[n] periodically, ie

$$x_p[n] = \sum_{l=-\infty}^{\infty} x[n - lN]$$

Now, shift the sequence $x_p[n]$ by k units to the right which is given by,

$$x'_p[n] = x_p[n-k] = \sum_{l=-\infty}^{\infty} x[n-k-lN]$$

The finite duration sequence,

$$x'[n] = \begin{cases} x'_p[n], & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

Can be obtained from x[n] by a circular shift.

The circular shift of a sequence can be represented as the index modulo N,

$$x'[n] = x(n-k, modulo\ N) = x((n-k))_N$$

If k = 2 and N = 4, we have

x'[n] = $x((n-2))_4$ which implies

$$x'(0) = x((-2))_4 = x(2)$$
$$x'(1) = x((1-2))_4 = x(3)$$
$$x'(2) = x((2-2))_4 = x(0)$$
$$x'(3) = x((3-2))_4 = x(1)$$

An N-point sequence is called circularly even if it is symmetric about the point zero on the circle, taking counterclockwise as positive direction.

This implies that,

$x[N - n] = x[n] \qquad 1 \le n \le N-1$

An N-point sequence is called circularly odd if it is antisymmetric about the point zero on the circle. This implies that

$x[N - n] = - x[n] \qquad 1 \le n \le N-1$

4. Time reversal of a sequence

The time reversal of an N-point sequence is attained by reversing its samples about the point zero on the circle. Thus the sequence $x((-n))_N$ is simply given as

$$x((-n))_N = x(N - n),\ 0 \le n \le N - 1$$

For periodic sequence $x_p[n]$,

$$\text{Even: } x_p[n] = x_p[-n] = x_p[N - n]$$

$$\text{Odd: } x_p[n] = - x_p[-n] = - x_p[N - n]$$

If $x_p[n]$ is complex valued

$$\text{Conjugate even: } x_p[n] = x^*_p[N - n]$$

$$\text{Conjugate odd: } x_p[n] = - x^*_p[N - n]$$

Thus, $x_p[n] = x_{pe}[n] + x_{po}[n]$

Where, $x_{pe}[n] = ½ (x_p[n] + x^*_p[N - n])$ and $x_{po}[n] = ½ (x_p[n] - x^*_p[N - n])$

5. Symmetry Properties of the DFTs

For complex valued x[n] (& its DFT X(k))

$x[n] = x_R[n] + jx_I[n] \qquad 0 \le n \le N - 1$

$X(k) = X_R(k) + jX_I(k) \qquad 0 \le k \le N - 1$

Now,

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

$$X_R(k) + jX_I(k) = \sum_{n=0}^{N-1} [x_R[n] + jx_I[n]] \left[cos\frac{2\pi kn}{N} - jsin\frac{2\pi kn}{N}\right]$$

$$= \sum_{n=0}^{N-1} \left[x_R[n] cos\frac{2\pi kn}{N} + x_I[n] sin\frac{2\pi kn}{N}\right] - j\sum_{n=0}^{N-1} \left[x_R[n] sin\frac{2\pi kn}{N} - x_I[n] cos\frac{2\pi kn}{N}\right]$$

Similarly of IDFT,

$$x_R[n] + jx_I[n] = \frac{1}{N}\sum_{k=0}^{N-1}\left[X_R(k)cos\frac{2\pi kn}{N} - X_I(k)sin\frac{2\pi kn}{N}\right] - j\frac{1}{N}\sum_{k=0}^{N-1}\left[X_R(k)sin\frac{2\pi kn}{N} + X_I(k)cos\frac{2\pi kn}{N}\right]$$

If x[n] is

i. Real-valued Sequences:

X(N – k) = X*(k) = X(-k) {from definition of DFT}

ii. Real and even sequences:

x[n] = x_R[n] and x[n] = x[N – n], 0 ≤ n ≤ N-1 then X_I(k) = 0. Hence DFT reduces to

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

Which is itself real valued and even. The IDFT reduces to

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

iii. Real and odd sequences:

x[n] = x_R[n] and x[n] = -x[N – n], 0 ≤ n ≤ N-1 then X_R(k) = 0. Hence DFT reduces to

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

Which is purely imaginary and odd. The IDFT reduces to

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

iv. Purely imaginary sequences:

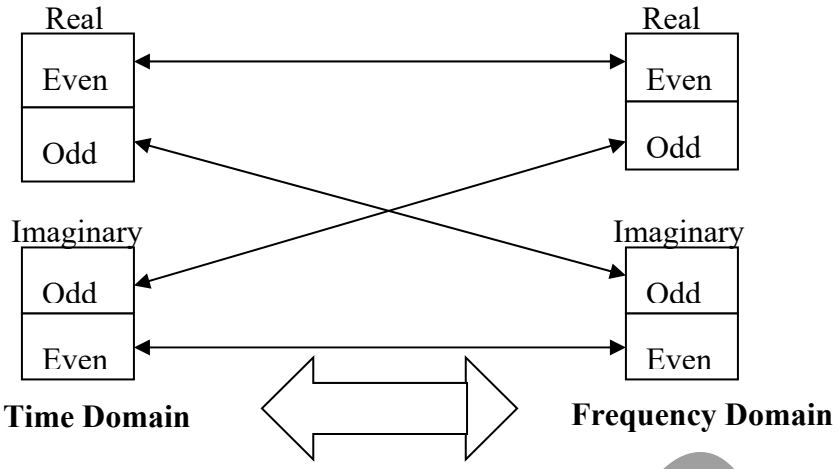
x[n] = jx_I[n], then

$$X_R(k) = \sum_{n=0}^{N-1} x_I[n]sin\frac{2\pi kn}{N} \;\&\; X_I(k) = \sum_{n=0}^{N-1} x_I[n]cos\frac{2\pi kn}{N}$$

We observe that X_R(k) is odd and X_I(k) is even.

If x_I[n] is odd, then X_I(k) = 0 and hence X(k) is purely real. On the other hand if, x_I[n] is even, then X_R(k) = 0 and hence X(k) is purely imaginary.

The symmetry properties given above may be summarized in block diagram as follows:



6. Multiplication of two DFTs and circular convolution:

If $x_1[n] \underset{N}{\overset{DFT}{\leftrightarrow}} X_1(k)$ & $x_2[n] \underset{N}{\overset{DFT}{\leftrightarrow}} X_2(k)$

Let us determine the relationship between $x_3[n]$ and the sequences $x_1[n]$ and $x_2[n]$ if we multiply the two DFTs together to obtain $X_3(k)$.

We have, $X_3(k) = X_1(k)X_2(k)$, $k = 0, 1, 2, \dots, N-1$

The IDFT of $X_3(k)$ is

$$x_3[m] = \frac{1}{N}\sum_{k=0}^{N-1} X_3(k)W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1} X_1(k)X_2(k)W_N^{-kn}, \quad m = 0,1,2,\dots,N-1$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1} x_1[n]W_N^{kn}\right]\left[\sum_{l=0}^{N-1} x_2[l]W_N^{kl}\right]W_N^{-kn}$$

Rearranging the order of summation,

$$x_3[m] = \frac{1}{N}\sum_{n=0}^{N-1} x_1[n]\sum_{l=0}^{N-1} x_2[l]\left[\sum_{k=0}^{N-1} W_N^{k(n+l-m)}\right] - - - (A)$$

{We have to obtain the form of $x_3[n] = x_1[n]$ operator $x_2[n]$}

The inner sum in bracket has the form,

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & if\ a = 1 \\ \dfrac{1-a^N}{1-a}, & if\ a \neq 1 \end{cases} - - - (B)$$

Where a is defined as,

$$a = W_N^{(n+l-m)} = e^{-j\frac{2\pi}{N}(n+l-m)}$$

As $e^{-j2\pi p} = 1$, where p is an integer (positive or negative)

a = 1, if p = (n + l – m)/N

or, n + l – m = pN, i.e. n + l – m is a multiple integer of N.

On the other hand, $a^N = e^{-j2\pi(n+l-m)} = 1$ for any value of a ≠ 0 (as n, l, m are integers)

Consequently (B) reduces to

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & if\ l = m - n + pN = ((m-n))_N, p\ is\ integer \\ 0, & otherwise \end{cases} ---(C)$$

If we substitute (C) into (A),

$$x_3[m] = \frac{1}{N}\sum_{n=0}^{N-1} x_1[n]\left[x_2\big((m-n)\big)_N . N\right]$$

$$or, x_3[m] = \sum_{n=0}^{N-1} x_1[n]\, x_2\big((m-n)\big)_N,\ \ m = 0,1,2,---,N-1$$

The expression above has the form of a convolution sum. However, it is not the ordinary linear convolution that we discussed/encountered earlier. Instead, the above convolution sum involves the index $((m-n))_N$, circular shift instead of linear shift, and is called **circular convolution.**

Thus we conclude that multiplication of the DFTs of two sequences is equivalent to the circular convolution of the two sequences in the time domain.

The circular convolution is denoted by $x_3[n] = x_1[n]$ (N) $x_2[n]$

Find circular convolution of following signals:

$x_1[n] = \{2, 1, 2, 1\}$ & $x_2[n] = \{1, 2, 3, 4\}$ (Ans: $x_3[n] = \{14, 16, 14, 16\}$)

Calculation of circular convolution:

There are several methods to calculate circular convolution of two sequences. Three of them are discussed below:

a. Graphical Method:

In this method the sequences are plot in graph in the form of circles. The procedure or steps are given below:

i. Graph each sequences as points on a circle, taking counterclockwise (CCW) as positive direction.

ii. **Folding:** Fold any one sequence on a circle. The folded sequence is simply graphed in a clockwise direction. Eg: fold $x_2[n]$ to obtain $x_2((-n))_N$.

iii. **Shifting:** shift the folded sequence i.e. $x_2((-n))_N$ by l to obtain $x_2((l - n))_N$, take l = 0, 1, 2, 3, … , N-1. $x_2((l - n))_N$ is simply the sequence $x_2((-n))_N$ rotated CCW by l units in time.

iv. **Multiplication:** Multiply $x_1[n]$ and $x_2((l - n))_N$ to yield the product sequence.

v. **Summation:** Sum the values in the product sequence to obtain $x_3[m]$ at m = l.

vi. **Repetition:** Repeat step (iii) to (v) to obtain $x_3[m]$ for all values of n.

The basic difference between linear and circular method is that, in circular convolution, the folding and shifting operation are performed in a circular fashion by computing the index of one of the sequences modulo N. In linear convolution, there is not modulo N operation.

b. DFT and IDFT method:

In this method, the DFTs of the sequences are calculated, the two DFTs are multiplied together and finally the IDFT of multiplied sequences is calculated to obtain the circular convolution.

c. Matrix Multiplication method:

In this method, the circular convolution of two sequences $x_1[n]$ and $x_2[n]$ can be obtained by representing the sequences in the matrix form as shown below:

$$\begin{bmatrix} x_2(0) & x_2(N-1) & x_2(N-2) & \cdots & x_2(2) & x_2(1) \\ x_2(1) & x_2(0) & x_2(N-1) & \cdots & x_2(3) & x_2(2) \\ x_2(2) & x_2(1) & x_2(0) & \cdots & x_2(4) & x_2(3) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_2(N-2) & x_2(N-3) & x_2(N-4) & \cdots & x_2(0) & x_2(N-1) \\ x_2(N-1) & x_2(N-2) & x_2(N-3) & \cdots & x_2(1) & x_2(0) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ \vdots \\ x_1(N-2) \\ x_1(N-1) \end{bmatrix} = \overrightarrow{x_3}[n]$$

The sequence $x_2[n]$ is repeated via circular shift of samples and represented in N×N matrix form. The sequence $x_1[n]$ is represented as column matrix. The multiplication of these two matrices gives the sequence $x_3[n]$.

7. Time reversal of a sequence:

If $x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k)$, then

$$x((-n))_N = x[N-n] \overset{DFT}{\underset{N}{\leftrightarrow}} X((-k))_N = X(N-k)$$

8. Circular time shift of a sequence:

If $x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k)$, then

$$x\big((n-l)\big)_N \overset{DFT}{\underset{N}{\leftrightarrow}} X(k)e^{-j\frac{2\pi kl}{N}}$$

9. Circular frequency shift:

If $x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k)$, then

$$x[n]e^{j\frac{2\pi ln}{N}} \overset{DFT}{\underset{N}{\leftrightarrow}} X((k-l))_N$$

This is the dual to the circular time shifting property.

10. Complex-conjugate properties:

If $x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k)$, then

$$x^*[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X^*\big((-k)\big)_N = X^*(N-k)$$

And the IDFT of $X^*(k)$ is

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

Therefore,

$$x^*((-n))_N = x^*(N-n) \overset{DFT}{\underset{N}{\leftrightarrow}} X^*(k)$$

11. Circular Correlation:

In general, for complex valued sequences x[n] and y[n], if

$$x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k) \;\&\; y[n] \overset{DFT}{\underset{N}{\leftrightarrow}} Y(k)$$

Then,

$$\tilde{\gamma}_{xy}(l) \overset{DFT}{\underset{N}{\leftrightarrow}} \tilde{R}_{xy} = X(k)Y^*(k)$$

Where$\tilde{\gamma}_{xy}$, is the (un-normalized) circular cross-correlation sequence defined as,

$$\tilde{\gamma}_{xy}(l) = \sum_{n=0}^{N-1} x[n]y^*((n-l))_N$$

Proof:

$$\tilde{\gamma}_{xy} = x(l)(N)y^*(-l)$$

$$Then, \quad \tilde{R}_{xy} = X(k)Y^*(k)$$

In the special case when y[n] = x[n], we have the corresponding expression for the circular auto-correlation of x[n] is

$$\tilde{\gamma}_{xx}(l) \overset{DFT}{\underset{N}{\leftrightarrow}} \tilde{R}_{xx} = |X(k)|^2$$

12. Multiplication of two sequences,

$$x_1[n]x_2[n] \overset{DFT}{\underset{N}{\leftrightarrow}} \frac{1}{N} X_1(k)(N)X_2(k)$$

13. Parseval's Theorem:

For complex valued sequences x[n] & y[n], in general, if

$$x[n] \overset{DFT}{\underset{N}{\leftrightarrow}} X(k) \;\&\; y[n] \overset{DFT}{\underset{N}{\leftrightarrow}} Y(k)$$

$$then, \sum_{n=0}^{N-1} x[n]y^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$$

Proof:

From Circular Correlation

$$\tilde{\gamma}_{xy}(l) = \sum_{n=0}^{N-1} x[n]y^*\big((n-l)\big)_N$$

$$for\ l = 0, \qquad \tilde{\gamma}_{xy}(0) = \sum_{n=0}^{N-1} x[n]y^*\big((n)\big)_N = \sum_{n=0}^{N-1} x[n]y^*[n]$$

$$and, \qquad \tilde{\gamma}_{xy}(l) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{R}_{xy}(k)e^{j\frac{2\pi kl}{N}}$$

$$or, \qquad \tilde{\gamma}_{xy}(l) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)e^{j\frac{2\pi kl}{N}}$$

$$for\ l = 0, \qquad \tilde{\gamma}_{xy}(0) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$$

When y[n] = x[n],

$$then, \qquad \sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

This expression relates the energy in the finite duration sequence x[n] to the power in the frequency components X(k).

Applications of DFT

The DFT has seen wide usage across a large number of fields:

- Spectral analysis
- Data compression
- Partial differential equations
- Multiplication of large integers
- Outline of DFT polynomial multiplication algorithm.

Fast Fourier Transform (FFT)

Efficient computation of the DFT Algorithms

The FFT is simply an algorithm to speed up the DFT calculation by reducing the number of multiplications and additions required.

Applications of FFT algorithms

Though discrete Fourier transforms can convert time domain data into frequency domain accurately, it has an unavoidable drawback. It is the inefficiency of the algorithm which makes the processing much time consuming. As a solution to this problem, FFT is invented. FFT is much faster & efficient than DFT.

DFT plays an important role in discrete-time signal analysis and processing where it can perform lots of operations like spectral analysis, correlation analysis, linear convolution of two sequences which is a key digital filtering operation and lots more.

The FFT is simply a fast (computationally efficient) way to calculate DFT. The idea behind the FFT algorithms is the "Divide & Conquer" approach, to split the DFT into a series of lower DFT and exploiting the symmetry & periodicity properties of complex exponential. Less computation is required to evaluate and combine the lower order DFTs than to evaluate original N-point DFT.

Computational complexity of the DFT

There are many ways to measure the complexity and efficient of an implementation or algorithms, and a final assessment depends on both the available technology and the intended application. We will use the number of arithmetic multiplications and additions as a measure of computational complexity. This measure is simple to apply, and the number of multiplications and additions is directly related to the computational speed when algorithms are implemented on general purpose digital computers or special purpose microprocessors.

The N-point DFT of sequence x[n] is given by

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

The direct computation of n-point DFT requires N^2 complex multiplication and N(N-1) additions.

For a complex-valued x[n]

$$X_R(k) = \sum_{n=0}^{N-1} [x_R[n] cos\frac{2\pi kn}{N} + x_I[n] sin\frac{2\pi kn}{N}]$$

$$X_I(k) = -\sum_{n=0}^{N-1} [x_R[n] sin\frac{2\pi kn}{N} - x_I[n] cos\frac{2\pi kn}{N}]$$

The direct computation of above DFT requires,

1. $2N^2$ evaluations of trigonometric functions.

2. $4N^2$ real multiplications.
3. 4N(N-1) real additions.
4. A number of indexing & addressing operations.

Divide & Conquer Approach:

This approach is based on the decomposition of an N-point DFT into successively smaller DFTs.

Let us consider the computation of an N-point DFT, where N can be factored as a product of two integers

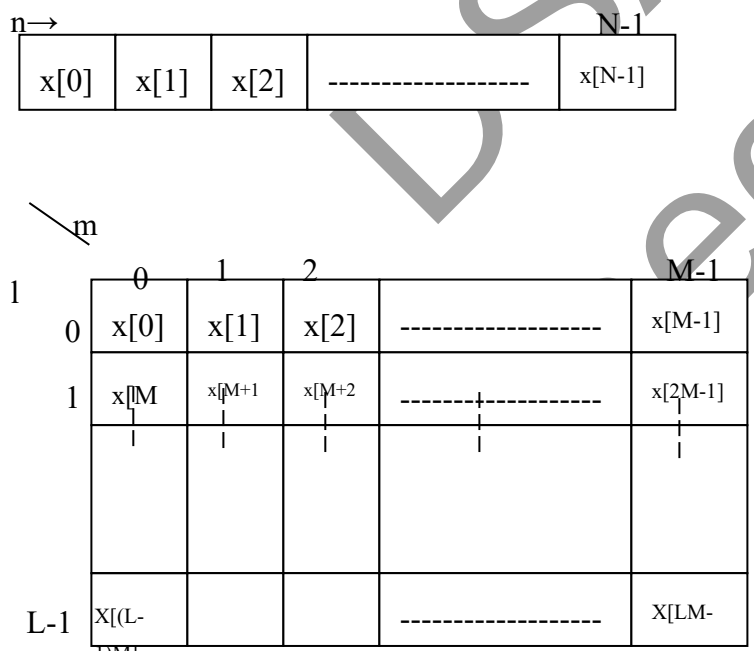
i.e. N = LM

if N is prime number, then padding with zeros is done. Now the sequence x[n], $0 \leq n \leq N-1$, can be stored in either a one-dimensional array indexed by n or as a two-dimensional array indexed by l & m, where $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$. Note that l is the row index and m is the column index. Thus sequence x[n] can be mapped in a rectangular array which depends on indexes (l, m).

Suppose we select the mapping

n = Ml + m

This leads to an arrangement in which the first row consists of the first M elements of x[n], the second row consists of the next M elements of x[n] and so on as shown in figure below:



Row-wise mapping n = Ml + m

On the other hand, the mapping

n = l + mL

stores the first L elements of x[n] in the first column, the next L elements in the second column, and so on.

A similar arrangement can be used to store the computed DFT values

K = PQ

k = Mp + q → row-wise mapping

k = qL + p → column-wise mapping

Now suppose that x[n] is mapped into the rectangular array x[l, m] and X(k) is mapped into a corresponding rectangular array X(p, q). Then the DFT can be expressed as a double sum over the elements of the rectangular array multiplied by the corresponding phase factors.

To be specific, Let us adopt a column-wise mapping for x[n] and the row-wise mapping for the DFT, then

$$X(p,q) = \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x(l,m)W_N^{(Mp+q)(mL+l)}$$

But $W_N^{(Mp+q)(mL+l)} = W_N^{MLmp}W_N^{Mpl}W_N^{Lmq}W_N^{ql}$

However,

$W_N^{Nmp} = 1$, $W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq}$

and $W_N^{mpL} = W_{N/M}^{pl} = W_L^{pl}$

$$\therefore X(p,q) = \sum_{l=0}^{L-1}\{W_N^{lq}[\sum_{m=0}^{M-1} x(l,m)W_M^{mq}]\}W_L^{lp}$$

Above expression involves the computation of DFTs of length M and length L

1. First, we compute the M-point DFTs

$$F(l,q) = \sum_{m=0}^{M-1} x(l,m)W_M^{mq}, \quad 0 \le q \le M-1$$

   For each rows l = 0, 1, 2, ----, L-1
2. Second, we compute a new rectangular array G(l, q)

$$G(l,q) = W_N^{lq}F(l,q), \quad 0 \le l \le L-1, \quad 0 \le q \le M-1$$

3. Finally, we compute the L-point DFTs.

$$X(p,q) = \sum_{l=0}^{L-1} G(l,q)W_L^{lp}, \quad for\ each\ column, 0 \le q \le M-1\ of\ the\ array\ G(l,q)$$

Computational Improvement of Divide and Conquer Approach:

Computational Complexity	Divide and Conquer	Direct DFT
Complex Multiplication	N(M + L + 1)	N^2

Complex Additions	N(M + L – 2)	N(N-1)
Eg: N = 10,000 we select L = 10 and M = 1000	N(M + L + 1) = 1,01,10,000 N(M + L – 2) = 1,00,80,000	N^2 = 10,00,00,000 N(N-1) = 9,99,90,000

When N is highly composite number,

$N = r_1r_2r_3\text{---}r_v$ (products of prime numbers)

Radix-2 FFT:

If N is factored as $N = r_1r_2r_3\text{---}r_L$, where $r_1 = r_2 = r_3 = \text{----} = r_L = r$ then $N = r^L$. Therefore, the DFT will be of size 'r' where the number 'r' is called the radix of the FFT algorithm.

If r = 2, then it is known as radix-2 FFT which is most widely used algorithm.

Decimation-in-Time Algorithm (DTA)

In this case, we assume that x[n] represents a sequence of N values, where N is an integer power of 2 i.e. $N = 2^L$. The given sequence is decimated (broken) into two N/2 point sequences which consists of the even and odd numbered values of x[n].

The N-point DFT of sequence x[n] is expressed as

$$X(k) = \sum_{n=0}^{N-1} x[n]W_N^{nk}, \qquad 0 \le k \le N-1$$

Splitting x[n] into its even & odd numbered values, we get

$$X(k) = \sum_{n=0,n-even}^{N-1} x[n]W_N^{nk} + \sum_{n=0,n-odd}^{N-1} x[n]W_N^{nk}, \qquad 0 \le k \le N-1$$

Now, putting n = 2r for even n and n = 2r+1 for odd n, we get

$$X(k) = \sum_{r=0}^{N/2-1} x[2r]W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1]W_N^{(2r+1)k}, \qquad 0 \le k \le N-1$$

$$Since\ W_N^2 = e^{-j\frac{2\pi}{N}\times 2} = e^{-j\frac{2\pi}{(N/2)}} = W_{N/2}$$

$$X(k) = \sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{rk}, \qquad 0 \le k \le N-1$$

$$or, X(k) = G(k) + W_N^k H(k), \qquad k = 0,1,2,---,\frac{N}{2}-1$$

Here, G(k) and H(k) are the N/2-points DFTs of the even and odd number sequences repectively. Here, we have computed each of the sums for $0 \le k \le N/2-1$ because G(k) and H(k) are considered periodic with period N/2. For k, greater than N/2 for X(k), using periodic property,

$X(k) = G(k) + W_N^k H(k), \qquad 0 \le k \le N/2-1$

& $X(k + N/2) = G(k + N/2) + W_N^{(k+\frac{N}{2})} H(k + N/2)$

Making the use of the symmetry property of W_N, G(k) & H(k)

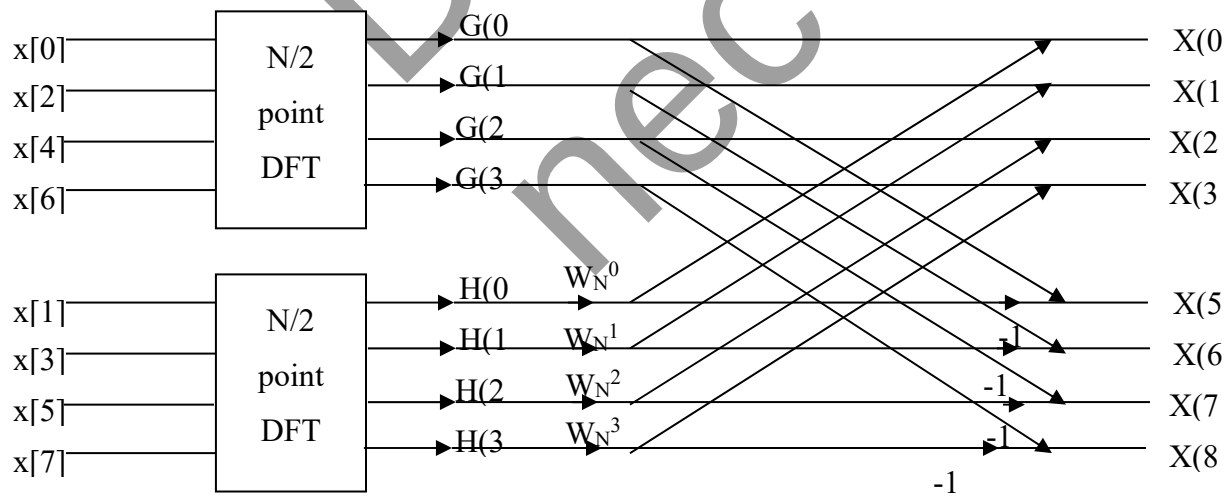
i.e. $W_N^{k+\frac{N}{2}} = -W_N^k, G\left(k + \frac{N}{2}\right) = G(k) \& H\left(k + \frac{N}{2}\right) = H(k)$

we obtain,

$X(k) = G(k) + W_N^k H(k), \qquad 0 \le k \le N/2-1$

& $X(k + N/2) = G(k) - W_N^k H(k)$

The flow graph of the decimation-in-time decomposition of an 8-Point (N=8) DFT computation into two 4-points DFT computations is shown in figure (a).



Fig(a): Illustration of flow graph of the first stage Decimation in time FFT algorithm for N=8

$Again, \qquad G(k) = \sum_{r=0}^{N/2-1} g[r] W_{N/2}^{rk}, \qquad 0 \le k \le N/2-1$

Or, $G(k) = \sum_{l=0}^{N/4-1} g[2l]W_{\frac{N}{2}}^{2lk} + \sum_{l=0}^{N/4-1} g[2l+1]W_{\frac{N}{2}}^{(2l+1)k},\ 0 \le k \le \frac{N}{2} - 1$

$$or, G(k) = \sum_{l=0}^{N/4-1} a[l]W_{\frac{N}{4}}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} b[l]W_{N/4}^{lk},$$

Or, G(k) = A(k) +W_N^{2k}B(k), $0 \le k \le N/4-1$

Here, A(k) and B(k) are N/4-point DFTs, while G(k) is N/2-point DFT.

For k, greater than N/4 the relation for G(k) is

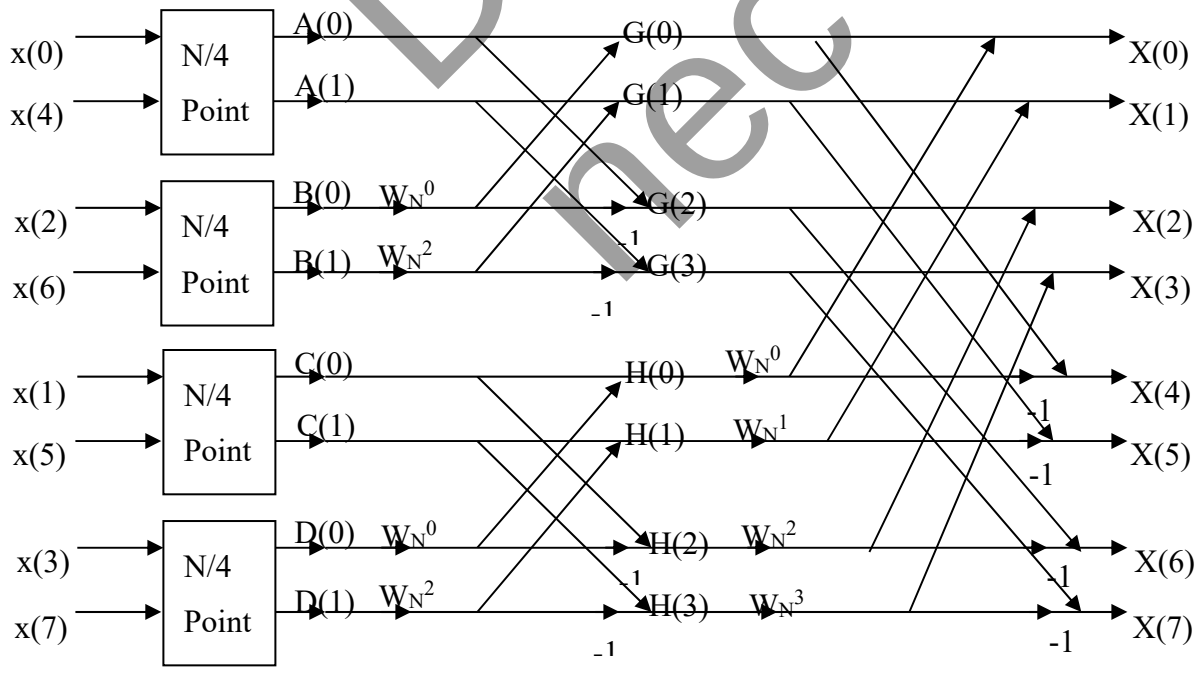
G(k + N/2) = A(k + N/4) +$W_N^{2(k+\frac{N}{4})}B$(k + N/4), $0 \le k \le N/4 - 1$

= A(k) – W_N^{2k}B(k)

In a similar manner,

H(k) = C(k) + W_N^{2k}D(k), $0 \le k \le N/4 - 1$

H(k + N/4) = C(k) - W_N^{2k}D(k),



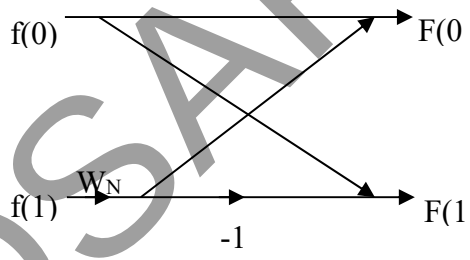
Fig(b): Illustration of flow graph of second stage decimation in time FFT algorithm for N=8

This process of reducing an L-point DFT (L is a power of 2) to an L/2 points DFTs, may be continued until we are left with 2-point DFTs, or there are L (= $\log_2 N$) stages to be evaluated.

A 2-points DFTs, F(k), k = 0, 1 can be evaluated as under

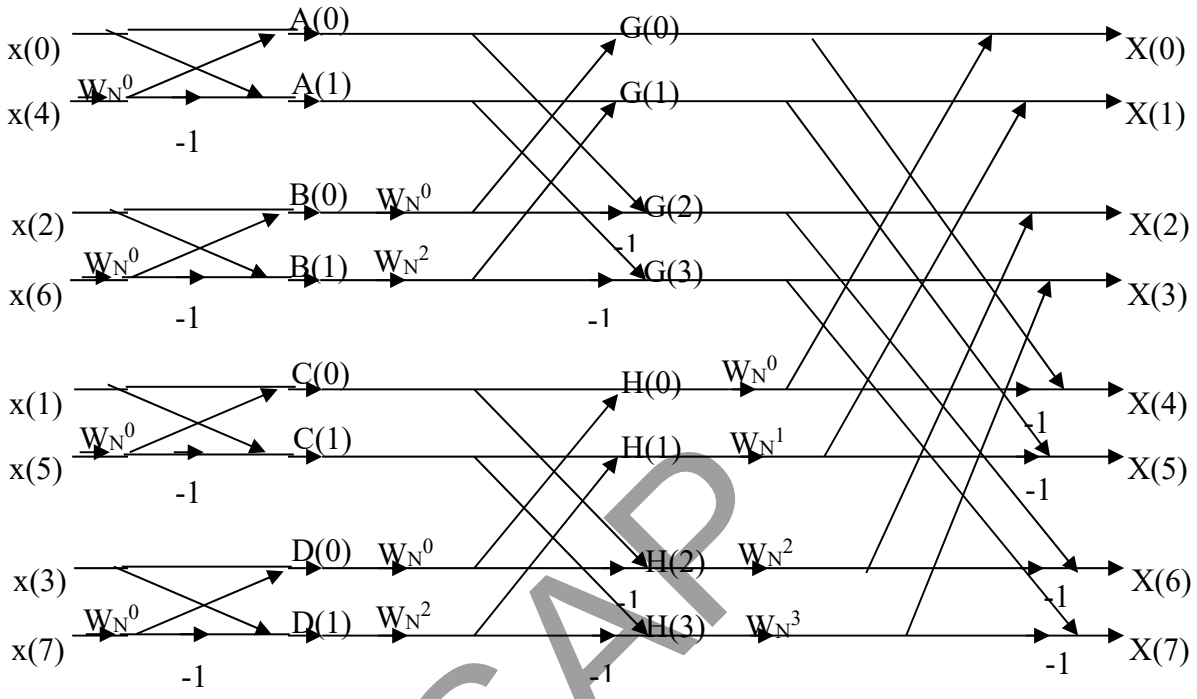
$$F(k) = \sum_{n=0}^{N-1} f[n] W_N^{nk}, \qquad k = 0,1$$
$$= f(0)W_N^0 + f(1)W_N^k$$
$$\therefore F(0) = f(0) + W_N^0 f(1)$$
$$\&F(1) = f(0) + W_N^1 f(1)$$
$$= f(0) - W_N^0 f(1)$$

Therefore in flow graph, the 2-point DFT can be calculated as



Fig(c): FFT Butterfly for 2-point DFT.

The complete flow graph of the decimation in time FFT algorithms for N=8, which is having of three stages is shown in figure(d). The first stage compute the four 2-points DFTs, the second stage computes the two 4-points DFTs, whereas the third stage computes the desired 8-point DFT.



Fig(d): Final flow graph of a Decimation in time FFT algorithm for N=8.

In order to understand the basic concepts of the FFT and its derivation, note that the DFT expansion can be greatly simplified by taking advantage of the symmetry and periodicity of the twiddle factors (W_N^k). If the equations are rearranged and factored, the result is the Fast Fourier Transform (FFT) which requires only (N/2) $\log_2(N)$ complex multiplications. The computational efficiency of the FFT versus the DFT becomes highly significant when the FFT point size increases to several thousand. However, notice that the FFT computes *all* the output frequency components (either all or none!). If only a few spectral points need to be calculated, the DFT may actually be more efficient. Calculation of a single spectral output using the DFT requires only N complex multiplications.

The radix-2 FFT algorithm breaks the entire DFT calculation down into a number of 2-point DFTs. Each 2-point DFT consists of a multiply-and-accumulate operation called a *butterfly*, as shown in Figure (c).

The 8-point decimation-in-time (DIT) FFT algorithm computes the final output in three stages as shown in Figure (D). The eight input time samples are first divided (or *decimated*) into four groups of 2-point DFTs. The four 2-point DFTs are then combined into two 4-point DFTs. The two 4-point DFTs are then combined to produce the final output X(k). Note that the basic two-point DFT butterfly operation forms the basis for all computation. The computation is done in three stages. After the first stage computation is complete, there is no need to store any previous results. The first stage outputs can be stored in the same registers which originally held the time samples x(n). Similarly, when the second

stage computation is completed, the results of the first stage computation can be deleted. In this way, *in-place* computation proceeds to the final stage.

Note that in order for the algorithm to work properly, the order of the input time samples, x(n), must be properly re-ordered using a *bit reversal* algorithm.

The decimal index, n, is converted to its binary equivalent. The binary bits are then placed in reverse order, and converted back to a decimal number. Bit reversing is often performed in DSP hardware in the data address generator (DAG), thereby simplifying the software, reducing overhead, and speeding up the computations.

The computation of the FFT using *decimation-in-frequency* (DIF) is shown in Figures (F). This method requires that the bit reversal algorithm be applied to the output X(k). Note that the butterfly for the DIF algorithm differs slightly from the decimation-in-time butterfly as shown in Figure (E). The use of decimation-in-time versus decimation-in-frequency algorithms is largely a matter of preference, as either yields the same result. System constraints may make one of the two a more optimal solution.

It should be noted that the algorithms required to compute the inverse FFT are nearly identical to those required to compute the FFT, assuming complex FFTs are used. In fact, a useful method for verifying a complex FFT algorithm consists of first taking the FFT of the x(n) time samples and then taking the inverse FFT of the X(k). At the end of this process, the original time samples, Re x(n), should be obtained and the imaginary part, Im x(n), should be zero (within the limits of the mathematical round off errors).

Computational Efficiency of an N-Point FFT:

The process to calculate N-point spectrum from the even N/2 point spectrum and the odd N/2 point spectrum can be progressively applied until it reaches the stage to calculate 2-point spectra. The 2^{nd} stage calculate N/2 point spectra from N/4 point spectra. The 3^{rd} stage does N/4 point spectra from N/8 point spectra, and so forth. Thus, it is already obvious that this process requires a less number of calculations, because the number of stages for N point FFT calculation is p, when $N = 2^p$, in terms of N itself, the number of the stages in $\log_2 N$. The total number of N calculations involving one addition & one multiplication must be carried out for each stage. Therefore, the total number of calculations for N-point FFT algorithm is $N\log_2 N$. The conventional method, on the other hand, requires N^2 calculations.

The total number of complex multiplications is reduced to $(N/2)\log_2 N$ and the number of complex additions is $N\log_2 N$.

Table 1: Computational Improvement of Radix-2 FFT Algorithm.

DFT: N^2 Complex Multiplications		**FFT: $(N/2)\log_2(N)$ Complex Multiplications**	
N	**DFT Multiplications**	**FFT Multiplications**	**FFT Efficiency**
256	**65,536**	**1,024**	**64 : 1**
512	**262,144**	**2,304**	**114 : 1**
1,024	**1,048,576**	**5,120**	**205 : 1**
2,048	**4,194,304**	**11,264**	**372 : 1**
4,096	**16,777,216**	**24,576**	**683 : 1**

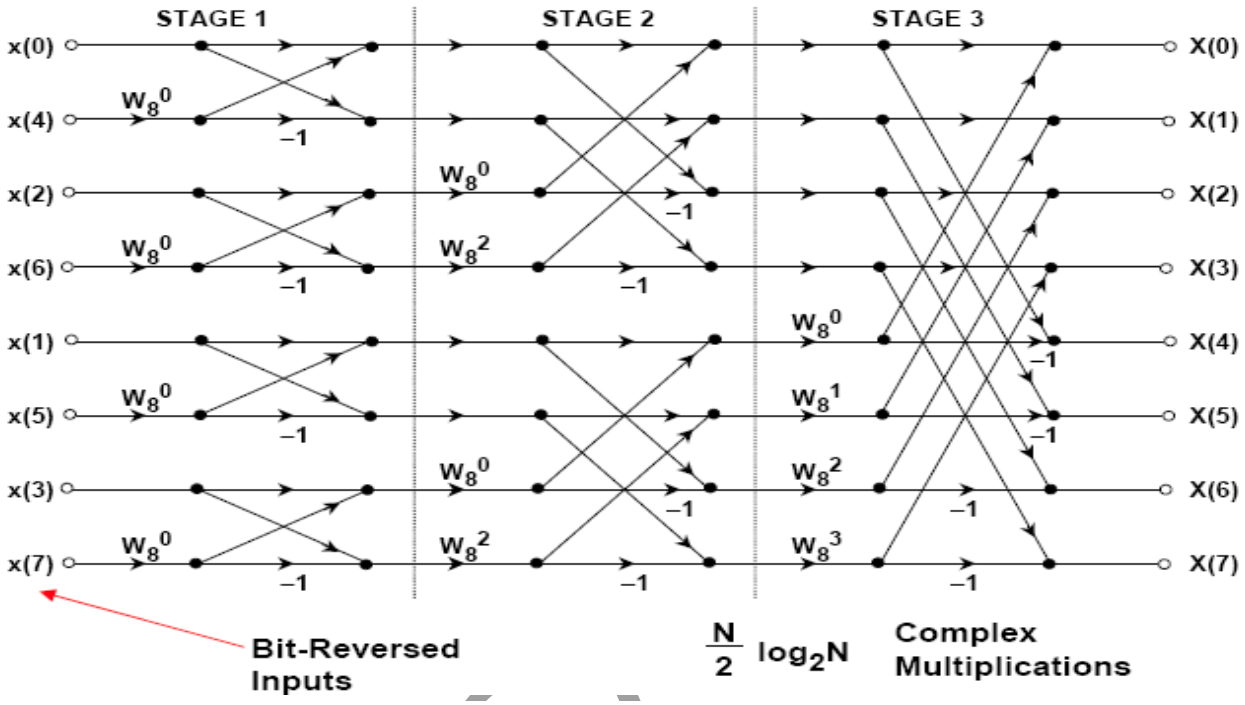
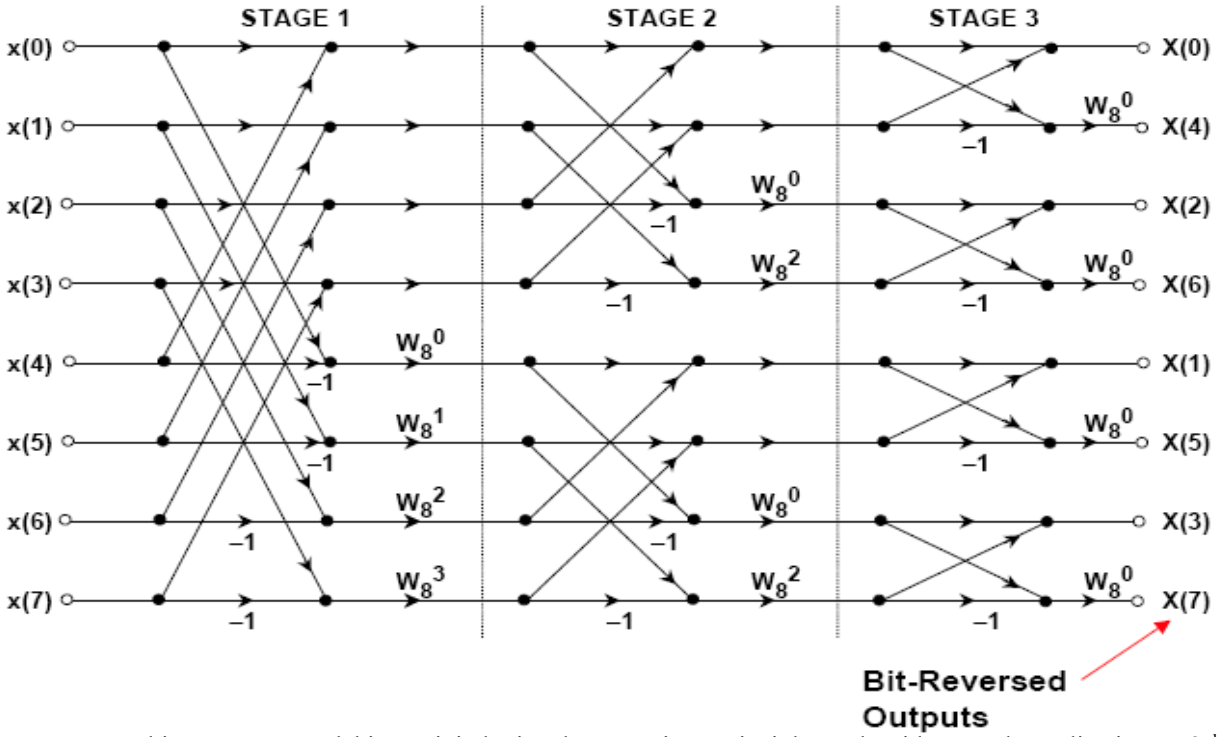


Figure: 8- point Decimation in time(above) and Decimation in Frequency(below) FFT algorithm



1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 7 & 8.

Chapter 5: Discrete Filter Structure

- Realization of LTI-DT systems in either software or hardware (FIR and IIR)
- Cascade, parallel & lattice structures are of particular importance which exhibit robustness in finite-word length implementation.
- Quantization effects in the implementation of digital filters using finite precision arithmetic.

Structures for the realization of Discrete-time systems:

An LTI DT system are characterized by general LCCD equation

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

Taking z-transform and characterizing in the rational system function,

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

In this characterization, we obtain the zeros and poles of the system function, which depend on the choice of the system parameters $\{b_k\}$ and $\{a_k\}$ & which determine the frequency response characteristics of the system.

Major factors that influence the choice of specific realization

- Computational complexity (no. of arithmetic operations)
- Memory requirements (no. of locations required to store system parameters, i/p, o/p, etc)
- Finite-word-length effects in the computations (quantization effects)

A. Structure for FIR systems:

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

Or, the system function

$$H(z)=\sum_{k=0}^{M-1} b_k z^{-k}$$

The unit sample response of the FIR system is identical to the coefficients (b_k}, that is,

$$h[n] = \begin{cases} b_n, & 0 \le n \le M-1 \\ 0, & otherwise \end{cases}$$

1. Direct Form Structure:

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

$$or, y[n] = \sum_{k=0}^{M-1} b_k x[n-k]$$

This structure requires

- M-1 memory locations
- M multiplications & M-1 additions per output point.

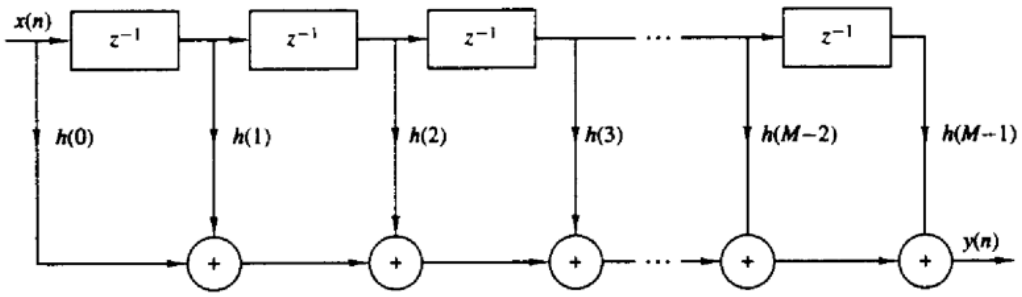


Figure 1: Direct Form Structure of FIR system.

This structure is often called a *transversal* or *tapped-delay line filter*.

2. Cascade-Form Structure:

The cascade realization follows from difference equation,

$$H(z)=\sum_{k=0}^{M-1} b_k z^{-k}$$

Factorizing in 2nd order FIR systems so that,

$$H(z)=\prod_{k=1}^{K} H_k(z), \qquad k = 1,2,3,\ldots.,K$$

$$H_k(z) = b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}$$

and K is the integer part of $^{(M+1)}/_{2}$

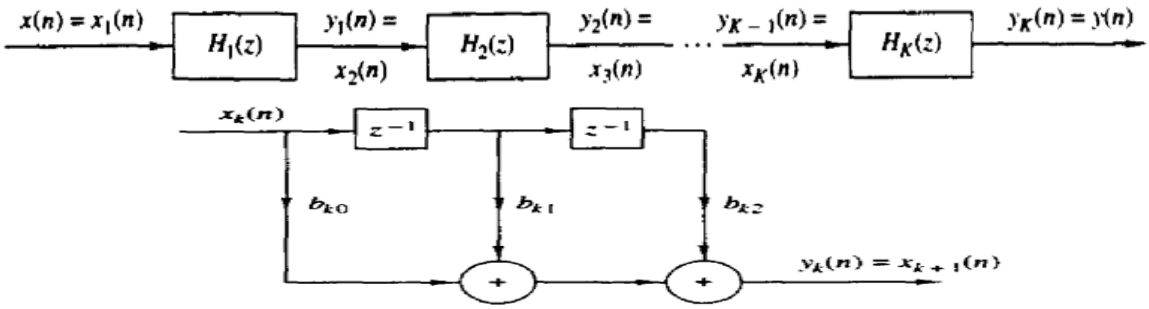


Figure 2: Cascade Realization of an FIR System

3. Frequency-Sampling Structures:

In frequency sampling realization, the parameters that characterize the filter are the values of the desired frequency response instead of the impulse response h[n]. To derive the frequency-sampling structure, we specify the desired frequency response at a set equally spaced frequencies,

$$\omega_{,} = \frac{2\pi}{M}(k+\alpha), \qquad k = 0, 1, 2, \ldots, \frac{M-1}{2}, \qquad M\ odd$$

$$k = 0, 1, 2, \ldots, \frac{M}{2} - 1, \qquad M\ even, \qquad \alpha = 0\ or \frac{1}{2}$$

And solve for the unit sample response h[n] from these equally spaced frequency specifications. Thus we can write the frequency response as

$$H(\omega) = \sum_{n=0}^{M-1} h[n]e^{-j\omega n}$$

And the values of H(ω) at frequencies $\omega_k = (2\pi/M)(k + \alpha)$ are simply

$$H(k+\alpha) = H\left(\frac{2\pi}{M}(k+\alpha)\right) = \sum_{n=0}^{M-1} h[n]e^{-j\frac{2\pi}{M}(k+\alpha)n}, \qquad k = 0, 1, 2, \ldots, M-1$$

The set of values {H(k + α)} are called the frequency samples of H(ω). In the case where α = 0, {H(k)} corresponds to the M-point DFT of {h[n]}.

It is a simple matter to invert above equation and express h[n] in terms of the frequency samples.

$$h[n] = \sum_{k=0}^{M-1} H(k+\alpha)e^{j\frac{2\pi}{M}(k+\alpha)n}, \qquad n = 0, 1, 2, \ldots, M-1$$

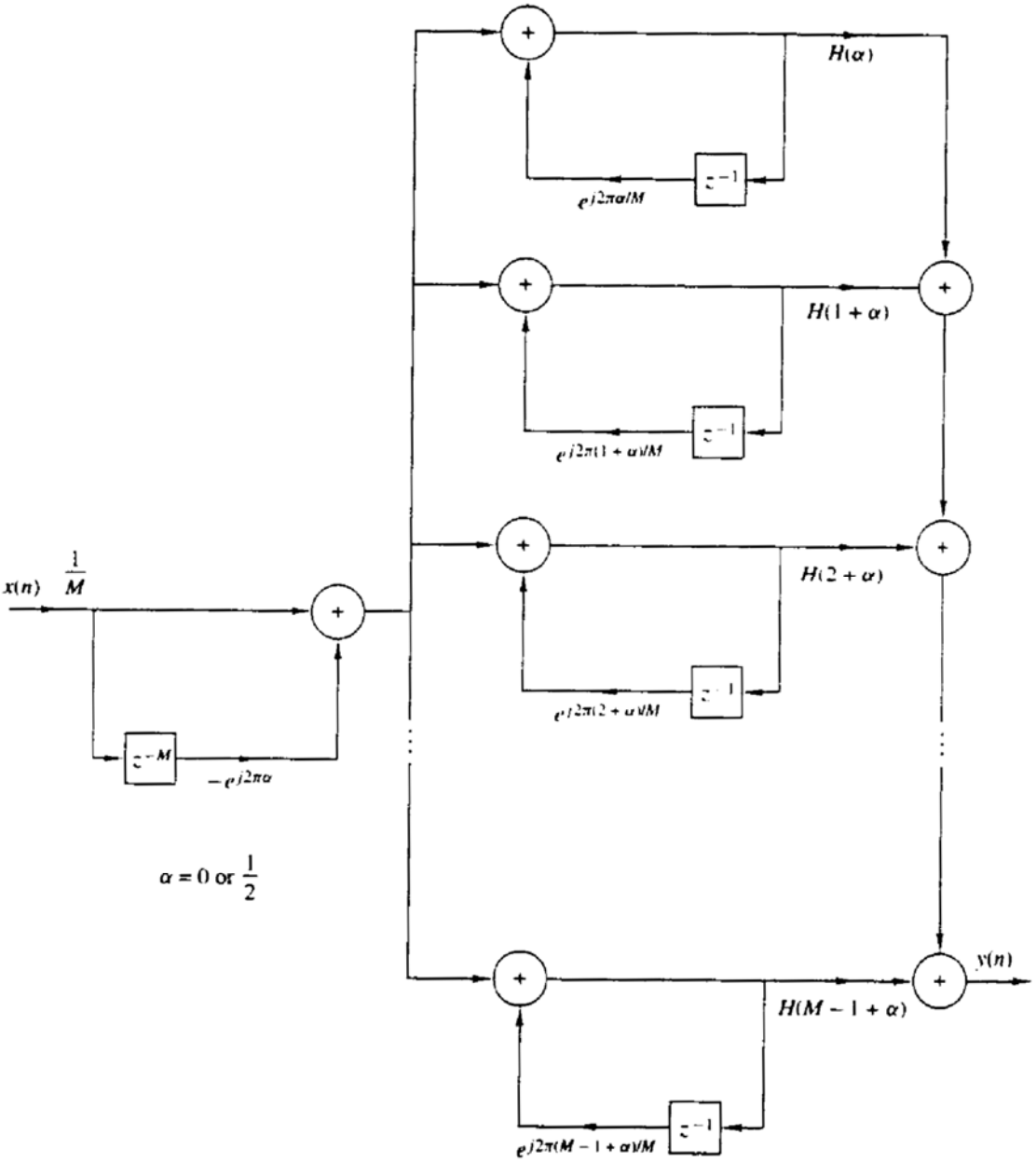
when α = 0, it is simply the IDFT of {H(k)}. Now taking z-transform, we have

$$H(z) = \sum_{n=0}^{M-1} h[n]z^{-n} = \sum_{n=0}^{M-1}\left[\sum_{k=0}^{M-1} H(k+\alpha)e^{j\frac{2\pi}{M}(k+\alpha)n}\right]z^{-n}$$

By interchanging the order of the two summations and performing the summation over the index n, we obtain

$$H(z) = \sum_{k=0}^{M-1} H(k+\alpha)\left[\frac{1}{M}\sum_{n=0}^{M-1}\left(e^{j\frac{2\pi}{M}(k+\alpha)}z^{-1}\right)^n\right] = \frac{1-z^{-M}e^{j2\pi\alpha}}{M}\sum_{k=0}^{M-1}\frac{H(k+\alpha)}{1-e^{\frac{j2\pi(k+\alpha)}{M}}z^{-1}}$$

Thus the system function is characterized by the set of frequency samples {H(k + α)} instead of {h[n]}. The realization is illustrated in figure below:



4. Lattice structure of FIR System:

Lattice filters are used extensively in digital speech processing and in the implementation of adaptive filters.

Let us consider FIR system functions

$$H(z) = A_m(z), \qquad m = 0, 1, 2, \ldots\ldots, M-1 \quad ---\text{I}$$

Where, $A_m(z)$ is the polynomial,

$$A_m(z) = \sum_{k=0}^{m} \alpha_m(k) z^{-k}, \qquad m \geq 1 \quad ---\text{II}$$

The subscript m on the polynomial $A_m(z)$ denotes the degree of the polynomial. For mathematical convenience, we define $\alpha_m(0) = 1$. If {x[n]} is input sequence to the filter $A_m(z)$ & {y[n]} is the output sequence, we have,

$$y[n] = x[n] + \sum_{k=1}^{m} \alpha_m(k) x[n-k] \quad ---\text{III}$$

Now, suppose that we have a filter of order 1 i.e. m=1, then

$$y(n) = x(n) + \alpha_1(1) x(n-1) ----IV$$

This output can be obtained from a 1st order or single-stage lattice filter, as in figure below:

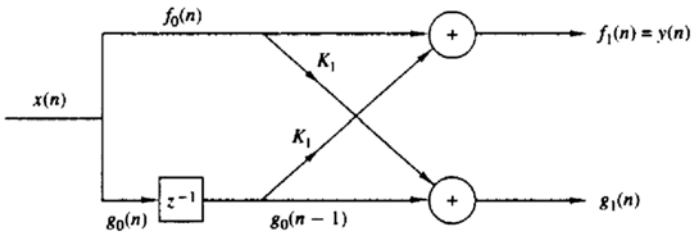


Figure 3: Single Stage Lattice Fiter.

Where $k_1=\alpha_1(1)$ is called a reflection coefficient. From Figure 3,

$$f_0(n) = g_0(n) = x[n]$$
$$f_1(n) = f_0(n) + k_1 g_0(n-1) = x(n) + k_1 x(n-1)$$
$$g_1(n) = k_1 f_0(n) + g_0(n-1) = k_1 x(n) + x(n-1) ----V$$

Now, for m=2,

$$y(n) = x(n) + \alpha_2(1)x(n-1) + \alpha_2(2)x(n-2) - - - VI$$

By cascading two lattice stages as shown in Figure 4, It is possible to obtain the output y(n)

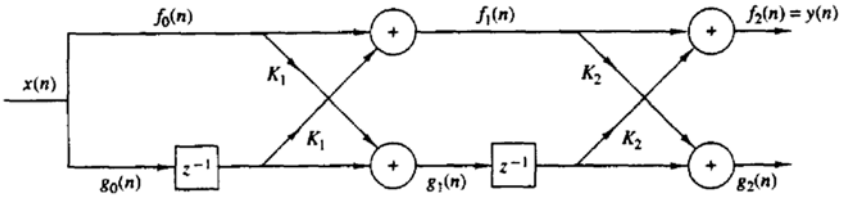


Figure 4: Two stage-lattice filter.

Output from 2nd stage is

$$f_2(n) = f_1(n) + k_2 g_1(n-1)$$
$$g_2(n) = k_2 f_1(n) + g_1(n-1) - - - VII$$

From equations V and VII,

$$f_2(n) = x(n) + k_1 x(n-1) + k_2\{k_1 x(n-1) + x(n-2)\}$$
$$i.e.\, y(n) = x(n) + k_1(1+k_2)x(n-1) + k_2 x(n-2) - - - VIII$$

Now equating equations VI and VIII,

$$\alpha_2(1) = k_1(1+k_2);\ \alpha_2(2) = k_2$$
$$i.e.\, k_1 = \frac{\alpha_2(1)}{1+k_2} = \frac{\alpha_2(1)}{1+\alpha_2(2)} - - - IX$$

Now by method of induction,

$$f_m(n) = f_{m-1}(n) + k_m g_{m-1}(n-1)$$
$$g_m(n) = k_m f_{m-1}(n) + g_{m-1}(n-1) - - - - X$$

Hence, (M-1) stage lattice filter is,

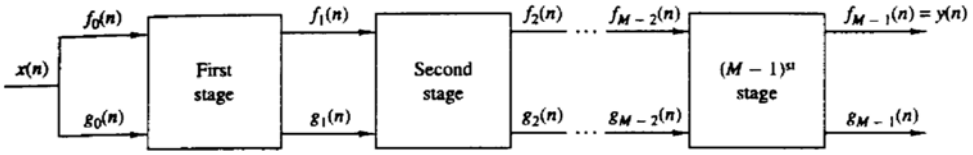


Figure 5: (M-1) stage lattice

Then, the output of the (M-1) stage filter corresponds to the output of an (M-1) order FIR , i.e. $y[n] = f_{M-1}[n]$

As a consequence of the equivalence between an FIR filter & a lattice filter, the output $f_m(n)$ of an m-stage lattice filter can be expressed as,

$$f_m(n) = \sum_{k=0}^{m} \alpha_m(k)x(n-k),\ \alpha_m(0) = 1 - - - XI$$

which is convolution sum. In z-domain,

$$F_m(z) = A_m(z)X(z)$$
$$or, A_m(z) = \frac{F_m(z)}{X(z)} = \frac{F_m(z)}{F_0(z)} - - - XII$$

The other output component from the lattice, namely $g_m(n)$, can also be expressed in the form of a convolution sum as,

$$g_m(n) = \sum_{k=0}^{m} \beta_m(k)x(n-k), \; - - - XIII$$

where, the filter coefficients $\{\beta_m(k)\}$ are associated with a filter that produces $f_m(n) = y(n)$ but operates in reverse order.

i.e $\beta_m(k) = \alpha_m(m - k)$, k = 0, 1, 2, -----, m with $\beta_m(m) = 1$.

In z-domain,

$$B_m(z) = \sum_{k=0}^{m} \beta_m(k)\, z^{-k} = \sum_{k=0}^{m} \alpha_m(m-k)\, z^{-k}$$

Taking m – k = l,

$$B_m(z) = \sum_{l=m}^{0} \alpha_m(l)\, z^{-(m-l)} = z^{-m} A_m(z^{-1})$$

This shows that the zeros of $B_m(z)$ are simply the reciprocals of zeros of $A_m(z)$. i.e. $B_m(z)$ is reverse polynomials of $A_m(z)$.

Let us write the lattice equation in z-domain,

$$F_0(z) = G_0(z) = X(z)$$
$$F_m(z) = F_{m-1}(Z) + k_m z^{-1} G_{m-1}(z)$$
$$G_m(z) = k_m F_{m-1}(z) + z^{-1} G_{m-1}(z)$$

If we divide each equation by X(z),

$$A_0(z) = B_0(z) = 1$$
$$A_m(z) = A_{m-1}(Z) + k_m z^{-1} B_{m-1}(z)$$
$$B_m(z) = k_m A_{m-1}(z) + z^{-1} B_{m-1}(z)$$

- Characterization of class-m FIR filters in direct form requires m(m+1)/2 filter coefficients, $\{\alpha_m(k)\}$, the lattice-form characterization requires only the m reflection coefficients $\{k_i\}$.
- The addition of stages to the lattice does not alter the parameters of the previous stages.

I. Conversion of lattice coefficient to direct form filter coefficients:

The direct form FIR filter coefficients $\{\alpha_m(k)\}$ can be obtained from the lattice coefficients $\{k_i\}$ by using the following relations:

$$A_0(z) = B_0(z) = 1$$
$$A_m(z) = A_{m-1}(Z) + k_m z^{-1} B_{m-1}(z)$$
$$B_m(z) = k_m A_{m-1}(z) + z^{-1} B_{m-1}(z)$$
$$= z^{-m} A_m(z^{-1}), m = 1,2,3, - - -, M-1$$
$$A_m(z) = 1 + \sum_{k=1}^{m} \alpha_m(k)\, z^{-k}$$

II. Conversion of direct form FIR coefficient to lattice coefficients

Suppose, we are given the polynomial $A_m(z)$ and we wish to determine the corresponding lattice filter parameters $\{k_i\}$. For the m-stage lattice we immediately obtain the parameter $k_m = \alpha_m(m)$. To obtain k_{m-1} we need the polynomial $A_{m-1}(z)$.

We have

$$A_m(z) = A_{m-1}(Z) + k_m z^{-1} B_{m-1}(z) - - - \text{i}$$
$$B_m(z) = k_m A_{m-1}(z) + z^{-1} B_{m-1}(z) - - - ii$$

From (i) and (ii)

$$A_m(z) = A_{m-1}(Z) + k_m z^{-1}\{B_m(z) - k_m A_{m-1}(z)\}/z^{-1}$$

After some calculation

$$A_{m-1}(Z) = \frac{A_m(z) - k_m B_m(z)}{1 - k_m^2}$$

Alternative formula is given by

$$k_m = \alpha_m(m)$$
$$\alpha_{m-1}(k) = \frac{\alpha_m(k) - \alpha_m(m)\alpha_m(m-k)}{1 - \alpha_m^2(m)}, m = M-1, M-1, \ldots, 1$$

B. Structures for IIR systems:

$$\text{y[n]} = -\sum_{\text{k=1}}^{\text{N}} \text{a}_\text{k}\text{y[n-k]} + \sum_{\text{k=0}}^{\text{M}} \text{b}_\text{k}\text{x[n-k]}$$

In Z-domain,

$$\text{H(z)} = \frac{\sum_{\text{k=0}}^{\text{M}} \text{b}_\text{k}\text{z}^{-\text{k}}}{1 + \sum_{\text{k=1}}^{\text{N}} \text{a}_\text{k}\text{z}^{-\text{k}}} = H_1 H_2$$

$$H_1(z) = \sum_{\text{k=0}}^{\text{M}} \text{b}_\text{k}\text{z}^{-\text{k}} \longrightarrow All - zero\ FIR\ System$$

$$H_2(z) = \frac{\sum_{\text{k=0}}^{\text{M}} \text{b}_\text{k}\text{z}^{-\text{k}}}{1 + \sum_{\text{k=1}}^{\text{N}} \text{a}_\text{k}\text{z}^{-\text{k}}} \longrightarrow All - pole\ IIR\ System$$

1. Direct form structure

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

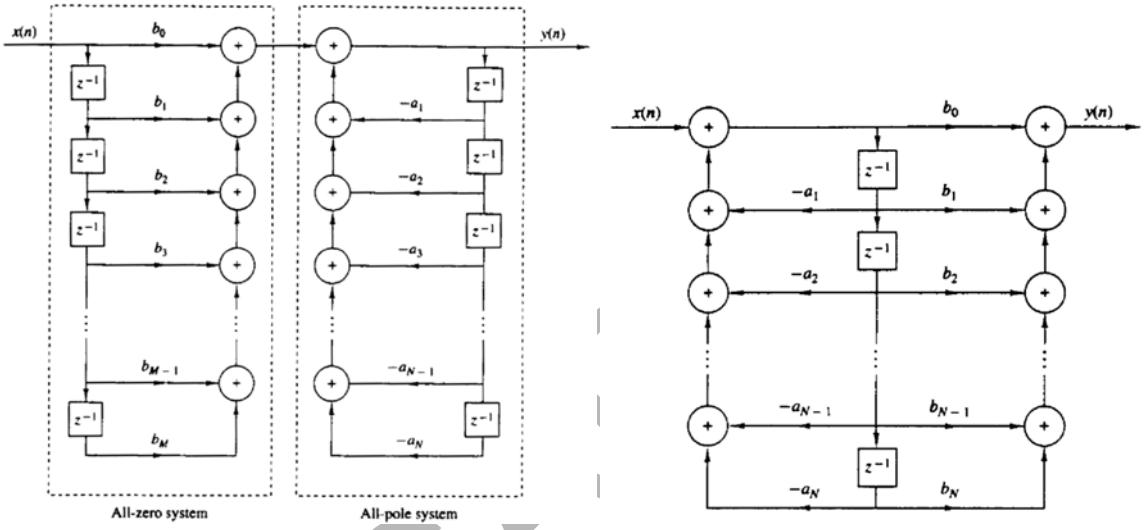


Figure : (a): Direct form I (b) Direct form II (N=M)

2. Cascade form structure:

The system can be factored into a cascade of 2nd order subsystems, such that H(z) can be expressed as,

$$H(z) = \prod_{k=1}^{L} H_k(z) \quad (assume\ N \geq M)$$

Where L is the integer part of (N+1)/2.

$H_k(z)$ has the general form,

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}$$

The coefficients $\{a_{ki}\}$ and $\{b_{ki}\}$ in the 2nd order subsystems are real. This implies that in forming the second order subsystems we should group together a pair of complex-conjugate poles and we should group together a pair of complex-conjugate zeros.

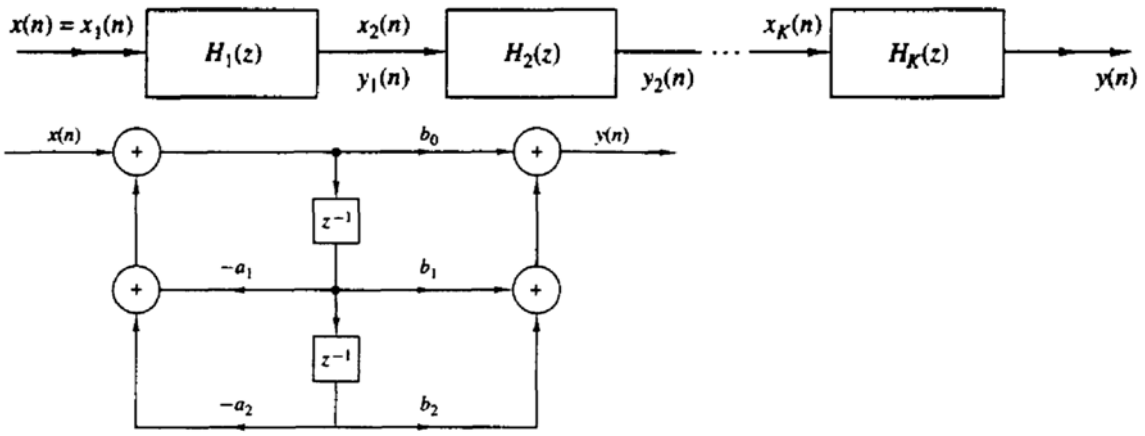


Figure 6: (a) Cascade of 2nd -order subsystem (b) Realization of each 2nd-order section.

3. Parallel form structure

A parallel form realization of an IIR system can be obtained by performing a partial-fraction expansion of H(z).

$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}$$

Where $\{p_k\}$ are the poles, $\{A_k\}$ are the coefficients (or residues) in the partial fraction expansion, and the constant $C = b_N/a_N$.

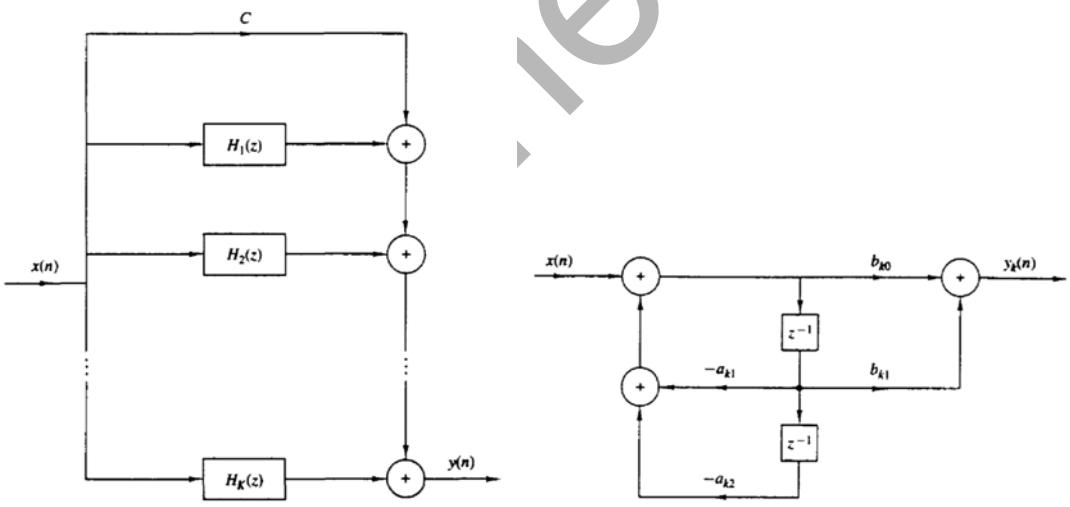


Figure 7: (a) Parallel structure of IIR system. (b) Structure of 2nd –order section in a parallel IIR system realization.

In general, some of the poles of H(z) may be complex valued. In such a case, the corresponding coefficients A_k are also complex valued. To avoid multiplications by complex numbers, we can combine pairs of complex-conjugate poles to form two-pole subsystems. In addition, we can combine, in an arbitrary manner, pairs of real-valued poles to form two-pole subsystems. Each of these sybsystems has the form

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}$$

Where the coefficients $\{b_{ki}\}$ and $\{a_{ki}\}$ are real-valued system parameters. The overall function can now be expressed as

$$H(z) = C + \sum_{k=1}^{K} H_k(z)$$

Where K is the integer part of (N+1)/2. When N is odd, one of the $H_k(z)$ is really a single-pole system (i.e. $b_{k1}=a_{k2}=0$).

Obtain the direct, the cascade and the parallel form realizations of the following IIR filter

$$H(z) = \frac{3(2z^2 + 5z + 4)}{(2z + 1)(z + 2)}$$

4. Lattice and Lattice-Ladder Structures for IIR Systems:

- All pole IIR System → Lattice Structure
- Pole-zero IIR System → Lattice-Ladder Structure

Lattice Figure of IIR System:

Let us begin with an all pole system with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^{N-1} a_k z^{-k}} = \frac{1}{A_N(z)}$$

The difference equation for this IIR system is

$$y[n] = x[n] - \sum_{k=1}^{N-1} a_N(k) y[n-k]$$

If we interchange the roles of input and output, we obtain

$$x[n] = y[n] - \sum_{k=1}^{N-1} a_N(k) x[n-k]$$

$$Or, y[n] = x[n] + \sum_{k=1}^{N-1} a_N(k)x[n-k]$$

Which describes an FIR system having the system function $H(z) = A_N(z)$.

Now, we take the all-zero lattice filter and redefine the input as $x[n] = f_N(n)$ and output as $y[n] = f_0(n)$.

These are exactly the opposite of the definitions for the all-zero lattice filters. These definitions dictate that the quantities $\{f_m(n)\}$ can be computed in descending order. This computation can be accomplished by rearranging the recursive equation and thus solving for $f_{m-1}(n)$ in terms of $f_m(n)$.

$$i.e.\, f_{m-1}(n) = f_m(n) - k_m g_{m-1}(n-1), \qquad m = N, N-1, \dots, 2, 1$$

The equation for $g_m(n)$remains unchanged

$$i.e.\, g_m(n) = k_m f_{m-1}(n) + g_{m-1}(n-1),$$

The result of these changes in the set of equation

$$\begin{aligned} &f_N(n) = x(n) \\ &f_{m-1}(n) = f_m(n) - k_m g_{m-1}(n-1), \qquad m = N, N-1, \dots, 2, 1 \\ &g_m(n) = k_m f_{m-1}(n) + g_{m-1}(n-1) \\ &y[n] = f_0(n) = g_0(n) \end{aligned}$$

Which correspond to the structure.

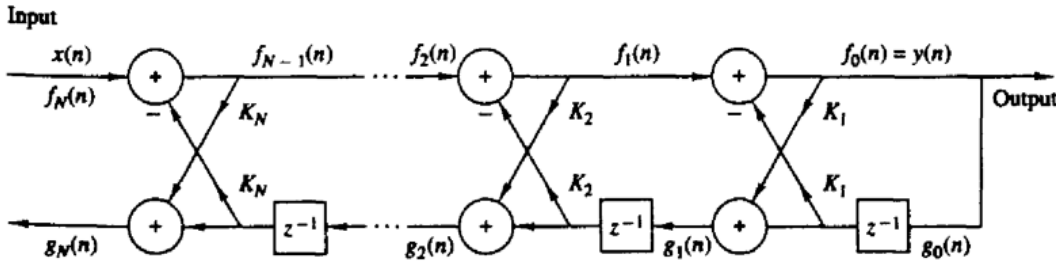


Figure: Lattice structure for an all pole IIR sytem.

In practical applications the all-pole lattice structure has been used to model the human voice tract and a stratified (def: deposited or arranged in horizontal layers) earth. In such cases the lattice parameters, $\{k_m\}$ have the physical significance of being identical to reflection coefficients in the physical medium. This is the reason that the lattice parameters are often called reflection coefficients. In such applications, a stable model of the medium requires that the reflection coefficients, obtained by performing measurements on output signals from the medium, be less than unity.

C. Lattice Ladder Structures:

Let us consider an IIR system with system function

$$H(z)=\frac{\sum_{k=0}^{M} c_M(k)z^{-k}}{1+\sum_{k=1}^{N} a_N(k)z^{-k}}=\frac{C_M(z)}{A_N(z)}----(I)$$

Where, the notation for the numerator polynomial has been changed to avoid confusion with our previous development. We assume that N ≥ M.

The direct form II structure is,

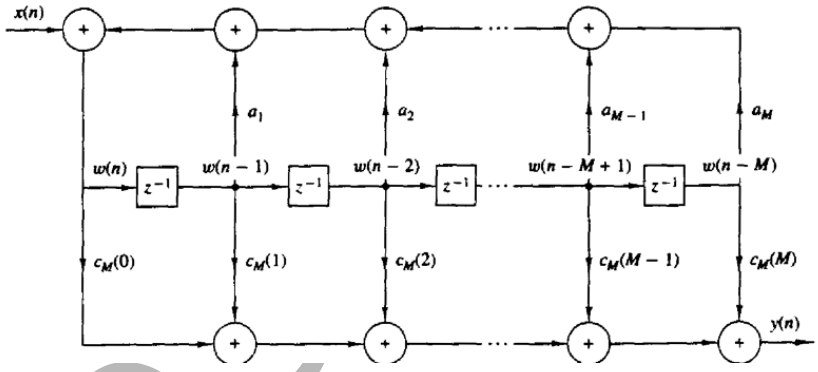


Figure : Direct form-II realization of IIR system.

$$y[n]=-\sum_{k=1}^{N} a_N(k)y[n-k]+\sum_{k=0}^{M} c_M(k)x[n-k]---(II)$$

$$w[n]=-\sum_{k=1}^{N} a_N(k)y[n-k]+x[n]-----(III)$$

$$y[n]=\sum_{k=0}^{M} c_M[k]w[n-k]------(IV)$$

Note that eqn (III) is the input-output of an all-pole IIR system and that eqn (IV) is the input-output of an all-zero system. Furthermore, we observe that the output of the all-zero system is simply a linear combination of delayed outputs from the all-pole system.

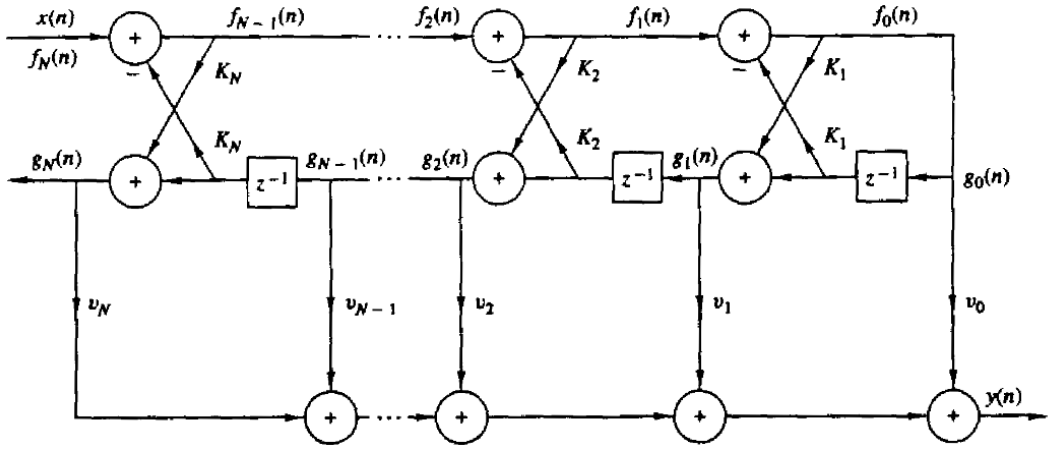


Figure: Lattice-Ladder structure for the realization of a pole-zero system.

$$y[n] = \sum_{m=0}^{M} v_m g_m(n) \quad ----(V)$$

Where $\{v_m\}$ are the parameters that determine the zeros of the system. The system function corresponding to eqn (V) is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} v_m G_m(z)}{X(z)} \quad -----(VI)$$

Since $X(z) = F_N(z)$ & $F_0(z) = G_0(z)$

$$H(z) = \sum_{m=0}^{M} v_m \frac{G_m(z)}{G_0(z)} \frac{F_0(z)}{F_N(z)}$$
$$= \sum_{m=0}^{M} v_m \frac{B_m(z)}{A_N(z)}$$
$$= \frac{\sum_{m=0}^{M} v_m B_m(z)}{A_N(z)} \quad ----(VII)$$

If we compare eqn (I) with (VII), we conclude that

$$C_M(z) = \sum_{m=0}^{M} v_m B_m(z) \quad ----(VIII)$$

This is the desired relationship that can be used to determine the weighting coefficients $\{v_m\}$. Thus, we have demonstrated that the coefficients of the numerator polynomial $C_M(z)$ determine the ladder parameters $\{v_m\}$, whereas the coefficients in the denominator polynomial $A_N(z)$ determine the lattice parameters $(k_m\}$.

The ladder parameters are determined from (III) which can be expressed as,

$$C_m(z) = \sum_{k=0}^{m-1} v_k B_k(z) + v_m B_m(z)$$

$$or\ C_m(z) = C_{m-1}(z) + v_m B_m(z)$$

Thus $C_m(z)$ can be computed recursively from the reverse polynomials $\mathbf{B_m}(z)$, m = 1, 2, 3, ….., M. Since $\beta_m(m) = 1$ for all m, the parameters v_m, m = 0, 1, …., M can be determined by first noting that,

$$v_m = c_m(m), \qquad m = 0,1,2,\ldots\ldots,M$$

$$\therefore C_{m-1}(z) = C_m(z) - v_m B_m(z)$$

Quantization of filter coefficients and effects on location of Pole and Zeros:

In the realization of FIR and IIR filters in hardware or in software on a general purpose computer, the accuracy with which filter coefficients can be specified is limited by the word length of the computer or the register provided to store the coefficients. Since, the coefficients used in implementing a given filter are not exact, the poles and zeros of system function will, in general, be different from the desired poles and zeros. Consequently, we obtain a filter having a frequency response of the filter with unqantized coefficients.

The sensitivity of the filter frequency response characteristics to quantization of the filter coefficients is minimized by realizing a filter having a large number of poles and zeros as an interconnection of second order filter sections. This leads to the parallel form and cascade form realizations in which the basic building blocks are second order filter sections.

Pole Perturbation:

Consider a general IIR filter with system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

The direct form realization of IIR filter with quantized coefficients has the system function

$$\overline{H}(z) = \frac{\sum_{k=0}^{M} \overline{b}_k z^{-k}}{1 + \sum_{k=1}^{N} \overline{a}_k z^{-k}}$$

Where, the quantized coefficients $\{\overline{b}_k\}$ and $\{\overline{a}_k\}$ can be related to the unquantized coefficients $\{b_k\}$ and $\{a_k\}$ by the relation

$$\overline{a}_k = a_k + \Delta a_k \;\; k = 1,2,3, \dots, N$$
$$\overline{b}_k = b_k + \Delta b_k \;\; k = 1,2,3, \dots, M$$

$$\{\Delta a_k\} and \; \{\Delta b_k\} represent \; quantization \; errors.$$

The denominator of H(z) may be expressed in the form

$$D(z) = 1 + \sum_{k=1}^{N} a_k z^{-k} = \prod_{k=1}^{N} (1 - p_k z^{-1})$$

Where $\{p_k\}$ are the poles of H(z). Similarly, we can express denominator of $\overline{H}(z)$as,

$$\overline{D}(z) = \prod_{k=1}^{N} (1 - \overline{p}_k z^{-1})$$

Where $\overline{p}_k = p_k + \Delta p_k, k = 0,1,2, \dots \dots, N$ and Δp_k is the error or **perturbation** resulting from the quantization of filter coefficient.

The perturbation error Δp_i can be expressed as,

$$\Delta p_i = \sum_{k=1}^{N} \frac{\partial p_i}{\partial a_k} \Delta a_k$$

Where $\frac{\partial p_i}{\partial a_k}$, the partial derivative of p_i with respect to a_k represents the incremental change in the pole p_i due to change in the coefficient a_k. Thus, the total error Δp_i is expressed as a sum of the incremental errors due to changes in each of the coefficients $\{a_k\}$.

The partial derivatives $\frac{\partial p_i}{\partial a_k}, k = 1,2, \dots, N$ can be obtained by differentiating D(z) with respect to each of $\{a_k\}$. We have,

$$\left(\frac{\partial D(z)}{\partial a_k}\right)_{z=p_i} = \left(\frac{\partial D(z)}{\partial z}\right)_{z=p_i} \left(\frac{\partial p_i}{\partial a_k}\right)$$

Then

$$\frac{\partial p_i}{\partial a_k} = \frac{\left(\frac{\partial D(z)}{\partial a_k}\right)_{z=p_i}}{\left(\frac{\partial D(z)}{\partial z}\right)_{z=p_i}}$$

$$\left(\frac{\partial D(z)}{\partial a_k}\right)_{z=p_i} = -z^{-k}|_{z=p_i} = -p_i^{-k}$$

$$\left(\frac{\partial D(z)}{\partial z}\right)_{z=p_i} = \left\{\frac{\partial}{\partial z}\left[\prod_{l=1}^{N}(1-p_l z^{-1})\right]\right\}_{z=p_i}$$

$$= \left\{\sum_{k=1}^{N}\frac{p_k}{z^2}\prod_{\substack{l=1\\l\neq k}}^{N}(1-p_l z^{-1})\right\}_{z=p_i} = \frac{1}{p_i^N}\prod_{\substack{l=1\\l\neq k}}^{N}(p_i-p_l)$$

$$\frac{\partial p_i}{\partial a_k} = -\frac{p_i^{N-k}}{\prod_{\substack{l=1\\l\neq k}}^{N}(p_i-p_l)}$$

$$Thus, \Delta p_i = -\sum_{k=1}^{N}\frac{p_i^{N-k}}{\prod_{\substack{l=1\\l\neq k}}^{N}(p_i-p_l)}\Delta a_k$$

This expression provides a measure of sensitivity of the i[th] pole to changes in the coefficients $\{a_k\}$.

References:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 9.

Design of Digital Filters

Introduction:

Filters are a particularly important class of linear time-invariant systems. In the design of frequency selective filters, the desired filter characteristics are specified in the frequency domain in terms of the desired magnitude and phase response of the filter. In the filter design process, we determine the coefficients of a causal FIR or IIR filter that closely approximate the desired frequency response specifications.

In practice, FIR filters are employed in filtering problems where there is a requirement for a linear-phase characteristic within the passband of the filters. As a general rule, an IIR filter has lower sidelobes. In the stopband than an FIR filter having the same number of parameters. For this reason, if some phase distortion is either tolerable or unimportant, an IIR filter is preferable, primarily because its implementation involves fewer parameter, require less memory & has lower computational complexity.

Digital Filter Design:

IIR filter and FIR filter are the two kinds of digital filters. The former one is commonly referred to as Recursive and the later as non-recursive.

Digital filters or digital signal processors are small special purpose digital computers designed to implement an algorithm that converts an input sequence x[n] into a desired output sequence y[n]. Such filters employ as hardwares, the devices such as adders, multipliers, shift registers, and delay elements. On the other hand, an analog filter employs resistors, capacitors and op-amps. As a result of this, digital processors are unaffected by factors such as component accuracy, temperature stability, long term drift, etc that affect the analog filters.

However, digital filter designs have to take into account such things as finite word size, round-off errors, aliasing, and other factors.

The difference equations that represent the IIR and FIR filters, in general, can be described as under:

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

Where a_k and b_k are suitable constants, x[n] and y[n] represent the input and output sequences.

In short, the procedure of implementation of a digital filter has the following steps in order:

1. Selection of filters
2. Specification of the frequency response characteristics of the filter
3. Phase response specifications
4. Filter Design
5. Filter realization
6. Filter implementation

Comparison between analog and digital filters:

	Analog Filters		Digital Filters
1.	Both inputs and outputs are continuous-time signals.	1.	Both inputs and outputs are discrete-time signals.
2.	Implementation of such filters is carried out using passive components such as resistor, capacitor, inductors and active components such as transistors, op-amps.	2.	Digital filters are implemented on a microcontroller using DSP integrated circuits. Three basic elements such as adder, multiplier and delay elements are utilized.
3.	Analog filters operate in infinite frequency range theoretically but limited in practice by the finite maximum operating frequency of the semiconductor devices used. Eg. Op-amp functions upto 100 MHz and higher frequency are handled by microwave devices.	3.	Frequency range is restricted to half of the sampling rate. It is also restricted by maximum computational speed available in particular application. In fact, is drawback of a digital filter.
4.	Main disadvantages of analog filters are its higher noise sensitivity, non-linearities, dynamic range limitations, lack of flexibility in designing and reproductivity, errors generated due to drift and variations in the value of active and passive components used in circuits.	4.	Main advantages of digital filters are that these are insensitive to noise, higher linearities, unlimited dynamic range, flexibility in software design, high accuracy, highly reliable.
5.	Analog filters have higher frequency range as well as they can interact directly with the real analog world.	5.	Digital filters require additional ADC and DAC converters section for connecting to the physical analog world.

Comparison between IIR and FIR digital filters:

	IIR Digital Filter		FIR Digital Filter
1.	IIR digital filters are characterized by rational system function: $H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}$	1.	FIR digital filters are characterized by system function which are not rational as: $H(z)=\sum_{k=0}^{M} b_k z^{-k}$
2.	Impulse response of these digital filters are computed for infinite number of samples i.e. $h[n] \neq 0$ for $0 \leq n \leq \infty$	2.	Impulse response of these filters is computed for finite no. of samples i.e. $h[n] \neq 0$ for $0 \leq n \leq M-1$ & 0 elsewhere.
3.	These filters do not have linear phase & these are used where some phase distortion is tolerable.	3.	These filters have linear phase characteristics. These filters are used in speech processing, it eliminates the adverse effects of frequency dispersion due to non-linearity of phase.
4.	Theoretically, these filters are stable. After truncation their coefficients, becomes unstable.	4.	These filters are realized by direct convolution, that is why these are stable.
5.	These filters has less flexibility for	5.	These filters have greater flexibility to control

	obtaining non –standard frequency response or for which analog filter design techniques are not available.		the shape of their magnitude response and realization efficiently.
6.	These filters are usually realized by recursive method. The present output of these filters also depends on previous outputs as well as past and present inputs. It is a feedback system.	6.	These filters are generally realize non-recursively or by direct convolution. These are not feedback systems. They are not dependent on previous outputs.
7.	They are more susceptible to round-off noise associated with finite precision arithmetic, quantization error and coefficient inaccuracies.	7.	These effects are less severe in FIR digital filters.
8.	Short-time delay.	8.	Time delay increase with increase in order.
9.	These require lesser number of arithmetic operations and these have lower computational complexity and smaller memory requirements.	9.	For sharp amplitude response, we require higher order FIR digital filter. This is a main drawback of an FIR filters.
10	IIR filters have resemblance with analog filters. The common method for IIR filter design is to design an IIR analog filter followed by analog to digital transformation methods.	10	FIR filters are unique to discrete-time domain. These cannot be derived from analog filters.
11	It requires less approximation parameter to design so, it is simple to design.	11	It requires more approximation parameter so, design procedure is complex.

Filters:

The term filter is commonly used to describe a device that discriminate, according to some attribute of the objects applied at its input, what passes through it.
Filtering in electrical world is a process by which the frequency spectrum of a signal can be modified, reshaped, or manipulated to achieve some desired objectives.
Filters allow some frequencies to pass while completely blocks other frequencies.

Why filters:

- Noise reduction
- For demodulation
- To separate distinct singals
- To limit the bandwidth of signals, etc

Types:

Filters are usually classified according to their frequency-domain characteristics as **Low-pass, High-pass, Band-pass, Band-stop or Band-elimination and Notch filters**.

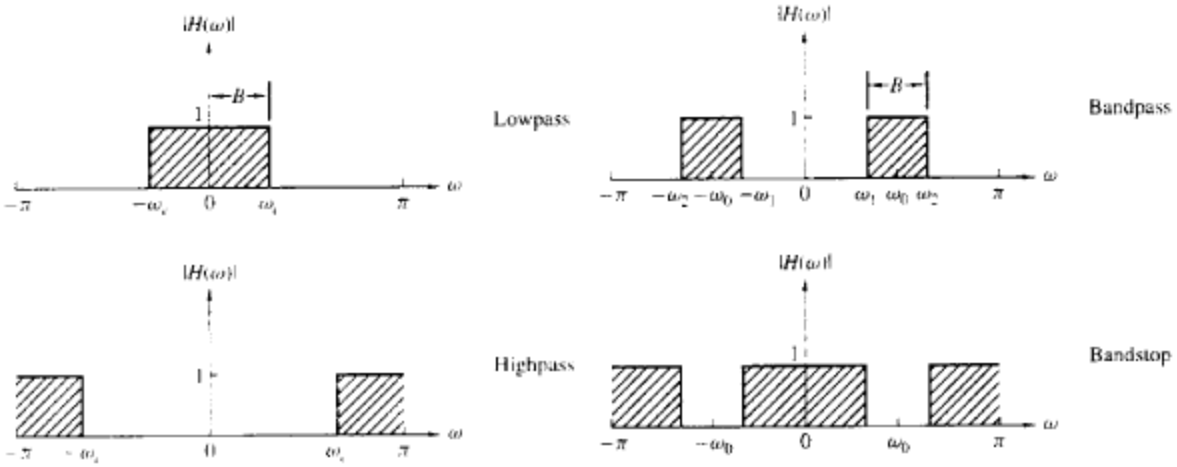


Figure: Magnitude responses for some ideal frequency-selective discrete-time filters.

Notch Filter

A notch filter is a filter that contains one or more deep notches or, ideally, perfect nulls in its frequency response characteristic. Figure below illustrates the frequency response characteristic of a notch filter with nulls at frequencies ω_0 and ω_1.

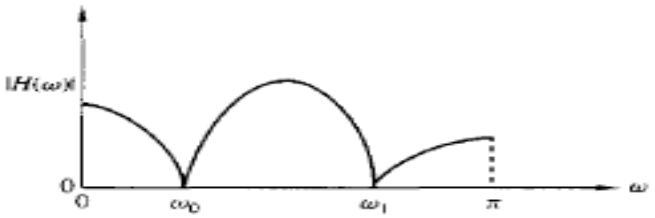


Figure: Frequency response characteristics of Notch filter

To create a null in the frequency response of a filter at frequency ω_0, we simply introduce a pair of complex conjugate zeros on the unit circle at an angle ω_0.

$$i.e. z_{1,2} = e^{\pm j\omega_0}$$

Thus system function for an FIR notch filter is

$$H(z) = b_0\left(1 - e^{j\omega_0}z^{-1}\right)(1 - e^{-j\omega_0}z^{-1})$$
$$= b_0(1 - 2cos\omega_0 z^{-1} + z^{-2})$$

The problem with the FIR notch filter is that the notch has relatively larger bandwidth, which means that other frequency components around the desired null are severely attenuated.

Suppose we place a pair of complex conjugate poles at ,

$$p_{1,2} = re^{\pm j\omega_0}$$

The effect of the poles is to introduce a resonance in the vicinity of the null and thus reduce the bandwidth of the notch. The system function for the resulting filter is

$$H(z) = \frac{b_0(1 - 2cos\omega_0 z^{-1} + z^{-2})}{1 - 2rcos\omega_0 z^{-1} + r^2 z^{-2}}$$

Notch filters are useful in many applications where specific frequency components must be eliminated. For example, instrumentation and recording systems require that the power-line frequency of 60 Hz and its harmonics be eliminated.

Advantages of using digital filters:

1. A digital filter is programmable, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the circuitry (hardware). An analog filter can be changed by redesigning the filter circuit.
2. Digital filters are easily designed, tested and implemented on a general-purpose computer or workstation.
3. The characteristics of analog filter circuits (particularly those containing active components) are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely stable with respect both to time and temperature.
4. Unlike their analog counterparts, digital filters can be handle low frequency signals accurately. As the speed of DSP technology continues to increase, digital filters are being applied to high frequency signals in the RF (radio frequency) domain, which in the past was the exclusive preserve of analog technology.
5. Digital filters are very much more versatile in their ability to process signals in a variety of ways: this includes the ability of some types of digital filter to adapt to changes in the characteristics of the signal.
6. Fast DSP processors can handle complex combinations of filters in parallel or cascade (series), making the hardware requirements relatively simple and compact in comparison with the equivalent analog circuitry.

Characteristics of Practical Frequency-Selective Filters:

Ideal Low-pass filter have sharp cut-off from passband to stopband. There is no gap between passband and stopband in ideal lowpass filter. In practice there are some ripples tolerable in passband and stopband and some gap between passband and stopband is also observed. The practical lowpass filter is shown in figure below:

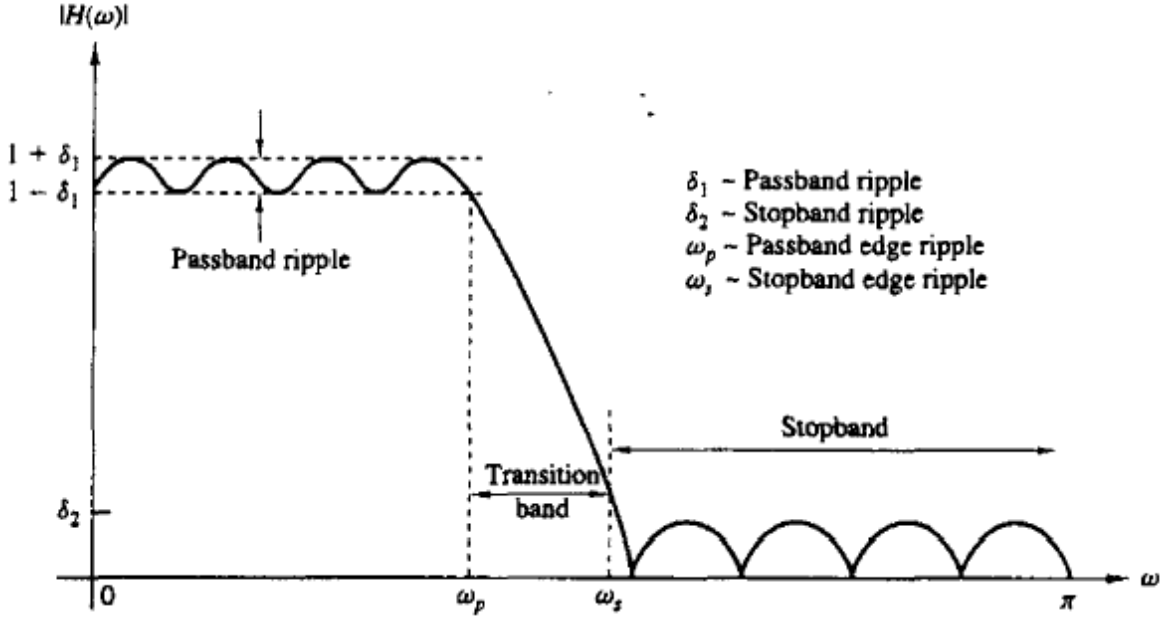


Figure: Magnitude Characteristics of Physically realizable lowpass filters.

Pass-band:

The frequency range over which a filter passes signal energy. At passband, the filters' frequency response is equal to or greater than -3dB.

Stop-band:

The frequency range over which a filter eliminates or reject signal energy. It is the band of frequency attenuated by digital filters. At stopband, the filters' frequency response is less than -3dB.

Transition band:

The frequency range between the passband and stopband.

Ripple:

It refers to fluctuations in passband or stopband of a filters' frequency versus magnitude curve. A small amount of ripple in the passband and stopband is usually tolerable.

Specifications of Analog Filters:

Gain, $G = 20\log_{10}|H(j\omega)|$ dB

Attenuation, $\alpha = -20\log_{10}|H(j\omega)|$ dB

- The width of the transition band is $\omega_s-\omega_p$
- The width of the passband is usually called the bandwidth of the filter
- The ripple in the passband is $20\log_{10}\delta_1$ decibels (dB)
- The ripple in the stopband is $20\log_{10}\delta_2$ decibels (dB)

In any filter design problem we can specify

1. The Maxmimum tolerable passband ripple
2. The maximum tolerable stopband ripple
3. The passband edge frequency, ω_p
4. The stopband edge frequency, ω_s.

Based on these specification, we can select the parameters $\{a_k\}$ and $\{b_k\}$ in the frequency response characteristics, $H(j\omega)$ which best approximates the desired specifications.

Design Procedure:

-Firslty lowpass filter is designed
-Then we change LPF to any other types
LPF ---(frequency transformation)→ Other types (BP, BS, HP)

Digital filter types:

1) **FIR (Non-Recursive)**
2) **IIR (Recursive) (We will study IIR filter in chapter 7)**

Design of Digital FIR filters:

- If a linear phase characteristics is required then FIR design, else FIR or IIR filter design.
- As a general rule, an IIR filter has lower sidelobes in the stopband than an FIR filter having the same number of parameters.
- IIR implementation involves a fewer parameters, require less memory and has lower computational complexity.

Why FIR filters?

In many digital signal processing applications, FIR filters are preferred over their IIR counterparts. The main advantages of the FIR filter designs over their IIR equivalents are the following:

i. They can have an exact linear phase.
ii. There exist computationally efficient realizations for implementing FIR filters. These include both nonrecursive and recursive realizations.
iii. FIR filters realized nonrecursively are inherently stable and free of limit cycle oscillations when implemented on a finite-wordlength digital system.
iv. The design methods are generally linear.
v. They can be realized efficiently in hardware.
vi. The filter start up transients have finite duration.
vii. The output noise due to multiplication roundoff errors in an FIR filter is usually very low and the sensitivity to variations in the filter coefficients is also low.

The main disadvantage of conventional FIR filter designs is that they require, especially in applications demanding narrow transition bands, considerably more arithmetic operations and hardware components, such as multiplier, adders, and delay elements than do comparable IIR filters.

Causality and its implications:

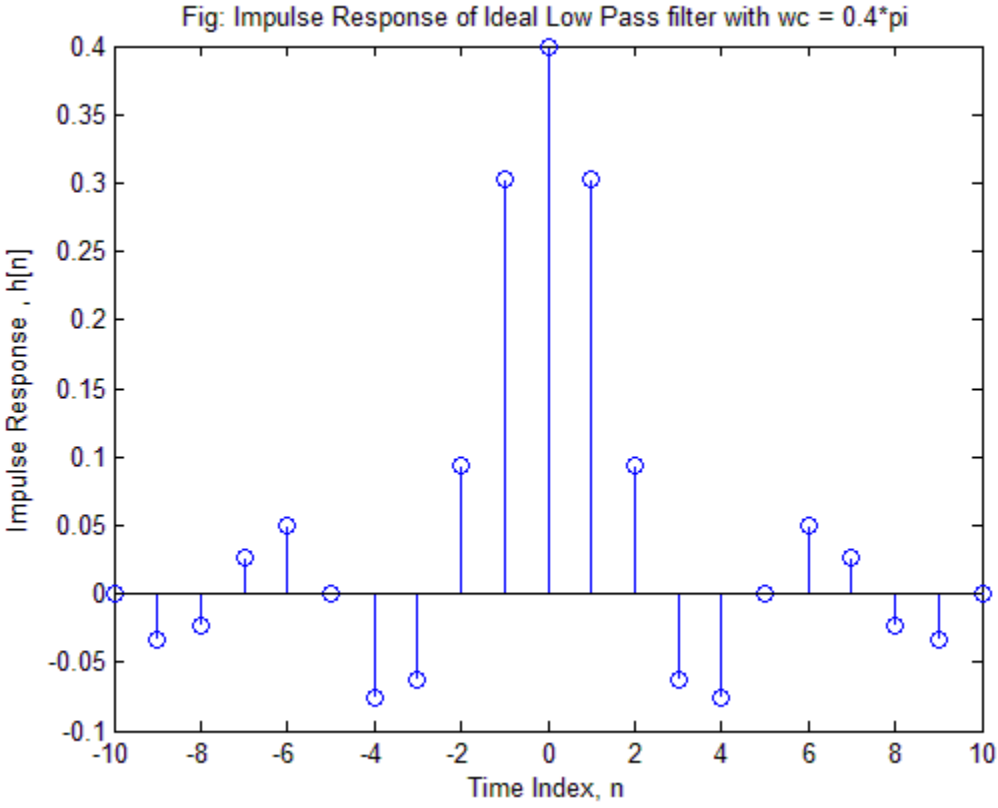
Let us examine the impulse response h[n] of an ideal lowpass filter with frequency response characteristics:

$$H(\omega) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < \omega \le \pi \end{cases}$$

The impulse response of this filter is

$$h[n] = \begin{cases} \omega_c/\pi, & n = 0 \\ \frac{\omega_c}{\pi}\frac{sin\omega_c n}{\omega_c n}, & n \neq 0 \end{cases}$$

Which is a Sinc function and it is clear that the ideal lowpass filter is noncausal and hence it cannot be realized in practice.



One possible solution is to introduce a large delay n_0 in h[n] and arbitrarily to set h[n] = 0 for n < n_0. However, the resulting system no longer has an ideal frequency response characteristics. If we set h[n] = 0 for n < n_0 , the Fourier series expansion of H(ω) results in the Gibbs Phenomenon.

Magnitude Response and Phase Response of Digital Filters:

The discrete-time Fourier transform of a finite sequence impulse response h[n] is given by

$$H(e^{j\omega}) = \sum_{n=0}^{M-1} h[n]e^{-j\omega n} = |H(e^{j\omega})|e^{j\varphi(\omega)}$$

The magnitude and phase responses are given by:

$$M(\omega) = |H(e^{j\omega})| = \sqrt{(H_R(e^{j\omega})^2 + H_I(e^{j\omega})^2)}$$

$$\varphi(\omega) = \tan^{-1} H_I(e^{j\omega}) \Big/ H_R(e^{j\omega})$$

Where $H_R(e^{j\omega}) = \text{Re}\{H(e^{j\omega})\}$ & $H_I(e^{j\omega}) = \text{Im}\{H(e^{j\omega})\}$

Filters can have a linear or non-linear phase, depending upon the delay function, namely the phase delay and group delay. The phase and group delays of the filter are given by

$$\tau_p = -\frac{\varphi(\omega)}{\omega} \qquad \& \qquad \tau_g = -\frac{d\varphi(\omega)}{d\omega}$$

Linea phase filters are those filters in which the phase delay and group delay are constants i.e. independent of frequency. Linear phase filters are also called constant time delay filters. For the phase response to be linear

$$\frac{\varphi(\omega)}{\omega} = -\tau, \qquad -\pi \le \omega \le \pi$$
$$\therefore \ \varphi(\omega) = -\omega\tau$$

Where τ is a constant phase delay expressed in number of samples.

$$\varphi(\omega) = \tan^{-1}\frac{H_I(e^{j\omega})}{H_R(e^{j\omega})} = -\omega\tau$$
$$or, -\omega\tau = \tan^{-1}\frac{\sum_{n=0}^{M-1} h[n]\sin\omega n}{\sum_{n=0}^{M-1} h[n]\cos\omega n}$$
$$or, \tan(\omega\tau) = \frac{\sum_{n=0}^{M-1} h[n]\sin\omega n}{\sum_{n=0}^{M-1} h[n]\cos\omega n}$$

Simplifying, we get

$$\sum_{n=0}^{M-1} h[n]\sin(\omega\tau - \omega n) = 0$$

And the solution is given by,

$$\tau = \frac{M-1}{2} \quad \& \, h[n] = h[M-1-n] \; for \; 0 < n < M-1$$

If these conditions are satisfied, then the FIR filter will have constant phase and group delays and thus the phase of the filter will be linear. The phase and group delays of the linear phase FIR filter are equal and constant over the frequency band. Whenever a constant group delay alone is preferred, the impulse response will be of the form,

$$h[n] = -h[M-1-n] \longrightarrow Antisymmetric\ impulse\ response\ sequence.$$

FIR filter design based on windowed Fourier series:

FIR filters are described by a transfer function that is a polynomial in z^{-1} and require different approaches for their design.

A direct and straightforward method is based on truncating the Fourier series representation of the prescribed frequency response.

The second method is based on the observation that for a length of N FIR digital filter, N distinct equally spaced frequency samples of its frequency response constitute the N-point DFT of its impulse response and hence, the impulse response sequence can be readily computed by applying an inverse DFT to these frequency samples.

Impulse Response of Ideal Filter:

Ideal lowpass filter has zero-phase frequency response

$$H_{LP}(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

The corresponding impulse response,

$$h_{LP}(n) = \begin{cases} \omega_c/\pi, & n = 0 \\ \frac{\omega_c}{\pi}\frac{\sin \omega_c n}{\omega_c n}, & n \neq 0 \end{cases} = \frac{\sin \omega_c n}{\pi n}, \qquad -\infty \le n \le \infty$$

The impulse response of an ideal lowpass filter is doubly infinite, not absolutely summable, and therefore unrealizable. By setting all impulse response coefficients outside the range –M ≤ n ≤ M equal to zero, we arrive at a finite length non-causal approximation of length N = 2M +1, which is shifted to right yield the coefficients of a causal FIR lowpass filter:

$$\hat{h}_{LP}(n) = \begin{cases} \frac{\sin \omega_c (n-m)}{\pi (n-m)}, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

FIR filter design by windowing:

If $H_d(\omega)$ is desired frequency response and $h_d[n]$ be its corresponding unit sample response. We begin with $H_d(\omega)$ & determine the $h_d[n]$. The relations are given by Fourier transform.

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d[n] e^{-j\omega n} - - - (i)$$

Where,

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

Here, the unit sample response from (i) is infinite duration and it is also non-causal and unrealizable. In order to obtain a realizable filter it must be truncated at some point. Let n = M-1 to yield the FIR filter of length M. i.e.

$$h[n] = \begin{cases} h_d[n], & 0 \le n \le M-1 \\ 0, & otherwise \end{cases}$$

Now, frequency response corresponding to the finite duration sequence is,

$$H(\omega) = \sum_{n=0}^{M-1} h_d[n]e^{-j\omega n} \quad ---(iii)$$

The process of obtaining (iii) from (i) is **windowing**. That is, truncating $h_d[n]$ to length M is equivalent to multiplying $h_d[n]$ by a Rectangular window, defined as

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

So, the unit sample response of FIR filter becomes,

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

note that the multiplication of $h_d[n]$ and w[n] results that the convolution of $H_d(\omega)$ and $W(\omega)$ will be the frequency domain representation of designed FIR filter. i.e.

$$H(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(v)W(\omega - v)dv$$

Gibbs Phenomenon:

The causal FIR filters, obtained by simply truncating the impulse response coefficients of the ideal filters given in the previous section exhibit an *oscillating behavior*, in their magnitude response is known as Gibb's Phenomenon.

The oscillation behavior of the magnitude response on both sides of the cut-off frequency is clearly visible in both cases. Moreover, as the length of the filter increased, the number of ripples in both passband and stopband increases with a corresponding decrease in the width of the ripples. However, the heights of the largest ripples, which occurs in both sides of the cut-off frequency remains same independent of the filter length and are approximately 11 percent of the difference between the passband and stopband magnitude of the ideal filters.

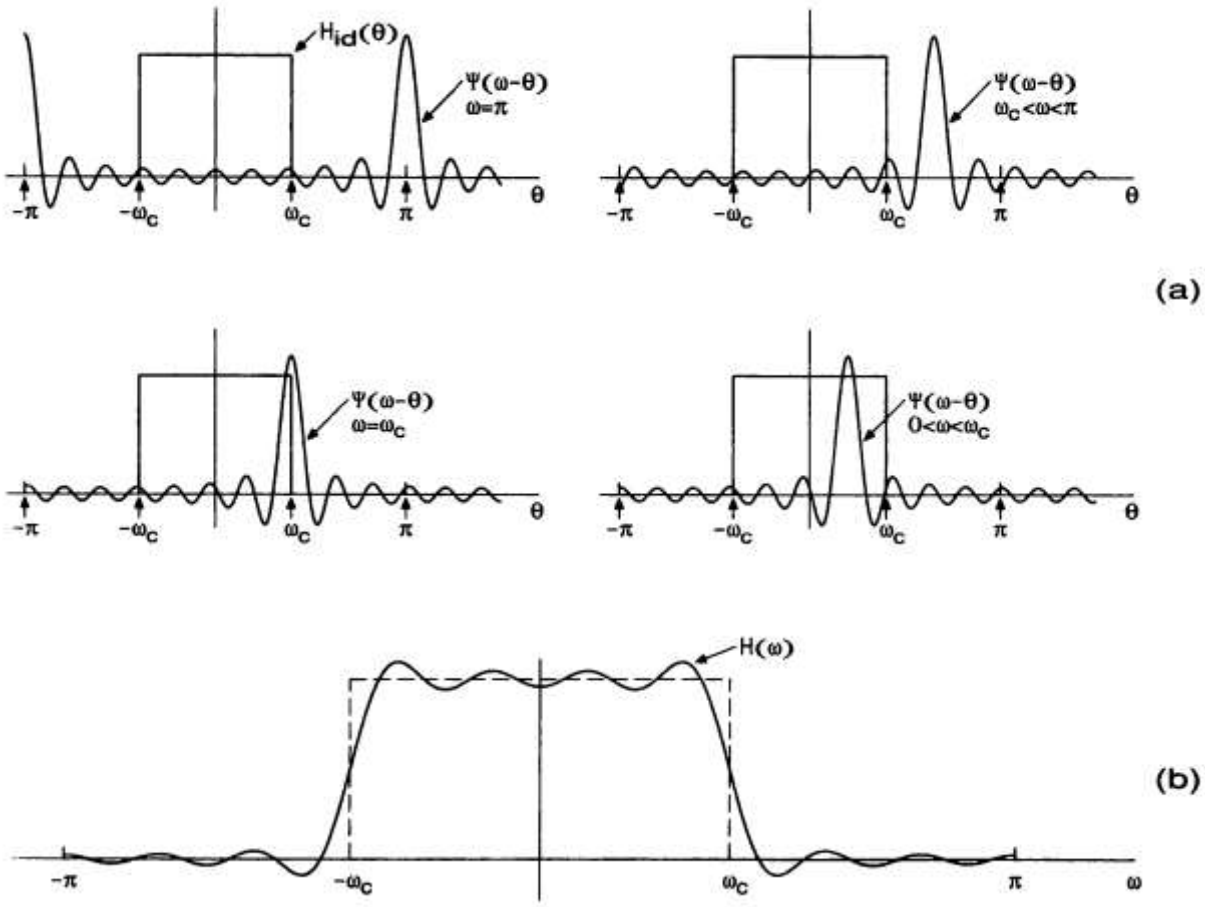


Figure: Illustration of effect of windowing in the frequency domain & Gibbs phenomenon.

The reason behind the Gibb's phenomenon can be explained by considering the truncation operations as multiplication by a finite-length window sequence ω(n) and by examining the windowing process in the frequency domain.

Rectangular window:

The rectangular window is defined as,

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

Then its Fourier transform is,

$$W(\omega) = \sum_{n=0}^{M-1} w[n]\, e^{-j\omega n}$$

On expansion we get,

$$W(\omega) = e^{-j\omega(M-1)/2}\frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

This window function has a magnitude response

$$|W(\omega)| = \left|\frac{\sin(\omega M/2)}{\sin(\omega/2)}\right|, \qquad -\pi \le \omega \le \pi$$

Here, the width of the main lobe is 4π/M. As the value of M increases the main lobe becomes narrower. However the sidelobes of |W(ω)| remain unaffected by value of M.

Note: (i) when M is increased, the width of main lobe is decreased and the transition band is reduced. (ii) Attenuation in side lobes is independent of M but it depends upon types of windows. (iii) A window function with minimum stop band, attenuation has maximum main lobe width. So proper window should be chosen in order to achieve a desire stop band attenuation.

Types of Windows:

1. Fixed Window Function:

Type	w[n] for $0 \le n \le M$	Transition width of main lobes	Minimum stopband attenuation
Rectangular	1	$\frac{4\pi}{M+1}$	-21 dB
Bartlett	$1-\frac{\lvert n\rvert}{M+1}$	$\frac{8\pi}{M}$	-25 dB
Hanning	$0.5\left(1-\cos\frac{2\pi n}{M-1}\right)$	$\frac{8\pi}{M}$	-44 dB
Hamming	$0.54-0.46\cos\left(\frac{2\pi n}{M-1}\right)$	$\frac{8\pi}{M}$	-53 dB
Blackman	$0.42-0.5\cos\frac{2\pi n}{M-1}+0.08\cos\frac{4\pi n}{M-1}$	$\frac{12\pi}{M}$	-74 dB
Blackman	$0.42-0.5\cos\frac{2\pi n}{M-1}+0.08\cos\frac{4\pi n}{M-1}$	$\frac{12\pi}{M}$	-74 dB

Note:

a. The $H_d(\omega)$ of an ideal LP filter is

$$H_d(\omega) = \begin{cases} e^{-j\omega\left(\frac{M-1}{2}\right)}, & for\ |\omega| \le \omega_c \\ 0, & elsewhere \end{cases}$$

b. The order of the filter is

$$N = k\left[\frac{2\pi}{\omega_s - \omega_p}\right]$$

The value of k can be obtained from the width of the main lobes.

c. The width of the main lobe $= k\left(\frac{2\pi}{M}\right)$

d. The phase delay $\tau = \frac{M-1}{2}$

2. Adaptive Window

a. Kaiser window:

$$w[n] = \frac{I_0\left[\beta\left(\sqrt{1-\left({}^{n}/_{M}\right)^2}\right)\right]}{I_0(\beta)} \; for -M \le n \le M$$

β = adjustable parameter

$I_0(x)$ = zeroth order Bessel function

$$I_0(x) = 1 + \sum_{r=1}^{\infty}\left[\frac{\left({}^{x}/_{2}\right)^r}{r!}\right]^2$$

β controls the minimum attenuation $\alpha_s = -20\log_{10}\delta_s$, in the stopband of the windowed filter response.

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7) & \text{for } \alpha_s > 50 \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21) & \text{for } 21 < \alpha_s \le 50 \\ 0 & for\ \alpha_s < 21 \end{cases}$$

The order of filter N is estimated by using the formula

$$N = \begin{cases} \dfrac{\alpha_s - 7.95}{14.36\Delta f} + 1 \text{ for } \alpha_s > 21 \\ \dfrac{0.9222}{\Delta f} + 1 \text{ for } \alpha_s \le 21 \end{cases}$$

$$\Delta f = f_s - f_p = \frac{\omega_s - \omega_p}{2\pi} \; (transition\ width)$$

FIR filters design by Frequency Sampling method:

In this method

- A set of samples is found from the desired frequency response, which is considered as DFT coefficients.
- Then IDFT of these samples are calculated which gives filter coefficients.

$$H_d(\omega) \xrightarrow{sample} H(k) \xrightarrow{IDFT} h(n)$$

The $H_d(\omega)$ is sampled at

$$\omega_k = \frac{2\pi}{M}(k+\alpha), \qquad k = 0,1,2,---,K$$

$$K = \begin{cases} \dfrac{M-1}{2} & if\ M\ odd \\ \dfrac{M}{2}-1 & if\ M\ even \end{cases}$$

If α = 0 , Type I design, else if α = ½ , Type II design.

Type I Design of frequency sampling method:

Consider the design of the FIR filter whose desired frequency response is denoted by $H_d(\omega)$. This frequency response is sampled uniformly at M points. Such frequency samples are given at,

$$\omega_k = \frac{2\pi}{M}k, \quad k = 0,1,2,3,\dots,M-1$$

Such sampled desired frequency response is a Discrete Fourier Transform, it can be denoted by

$$H(k) = H_d(\omega)\big|_{\omega=\omega_k}, \quad k = 0,1,2,\dots,M-1$$

$$or\ H(k) = H_d\left(\frac{2\pi k}{M}\right), \quad k = 0,1,2,\dots,M-1$$

Hence H(k) is M-point DFT. By taking inverse DFT of H(k), we get h[n]. This h[n] is unit sampled response of FIR filter.

$$h[n] = \frac{1}{M}\sum_{k=0}^{M-1} H(k)e^{\frac{j2\pi kn}{M}}, \quad n = 0,1,2,3,\dots,M-1$$

hence, the unit sample response of FIR filter of length M is obtained using frequency sampling method. The above impulse response can be simplified for linear phase condition as,

$$h[n] = \frac{1}{M}\left[H(0) + 2\sum_{k=1}^{K} Re\left\{H(k)e^{\frac{j2\pi kn}{M}}\right\}\right]$$

$$K = \begin{cases} \dfrac{M-1}{2} & if\ M\ odd \\ \dfrac{M}{2}-1 & if\ M\ even \end{cases}$$

Derive an expression for system function if the unit sample response h[n] is obtained using frequency sampling method.

Solution: ➔

The system function H(z) is given as z-transform of h[n] i.e.

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

Substituting for h[n]

$$H(z) = \sum_{n=0}^{M-1} \left[\frac{1}{M} \sum_{k=0}^{M-1} H(k) e^{\frac{j2\pi kn}{M}} \right] z^{-n}, n = 0, 1, 2, 3, \dots, M-1$$

Interchanging the order of summation in above equation, we get

$$H(z) = \frac{1}{M} \sum_{k=0}^{M-1} \left[H(k) \sum_{n=0}^{M-1} e^{\frac{j2\pi kn}{M}} z^{-n} \right]$$

The inner summation is in the form of,

$$\sum_{n=N_1}^{N_2} a^n = \frac{a^{N_1} - a^{N_2+1}}{1-a}$$

$$e^{\frac{j2\pi kn}{M}} z^{-n} = \left(e^{\frac{j2\pi k}{M}} z^{-1} \right)^n$$

Now applying above result to the summation,

$$\sum_{n=0}^{M-1} \left(e^{\frac{j2\pi k}{M}} z^{-1} \right)^n = \frac{1 - \left(e^{\frac{j2\pi k}{M}} z^{-1} \right)^M}{1 - e^{\frac{j2\pi k}{M}} z^{-1}} = \frac{1 - e^{j2\pi k} z^{-M}}{1 - e^{\frac{j2\pi k}{M}} z^{-1}} = \frac{1 - z^{-M}}{1 - e^{\frac{j2\pi k}{M}} z^{-1}}$$

$$\therefore H(z) = \frac{1 - z^{-M}}{M} \sum_{k=0}^{M-1} \left[\frac{H(k)}{1 - e^{\frac{j2\pi k}{M}} z^{-1}} \right]$$

This equation gives the system function H(z).

Design a LP FIR filter using frequency sampling method having cutoff frequency of π/2 rad/samples. The filter should have linear phase & length of 17.

Solution: →

i) To determine the desired frequency response:

The desired frequency response is $H_d(\omega)$. For the linear phase FIR lowpass filter, it is given as under:

$$H_d(\omega) = \begin{cases} e^{-j\omega\left(\frac{M-1}{2}\right)}, & for\ 0 \le \omega \le \omega_c \\ 0\ , & \omega_c \le \omega \le \pi \end{cases}$$

The length of the filter is M = 17 and the cutoff frequency is $\omega_c = \pi/2$ rad/samples.

$$H_d(\omega) = \begin{cases} e^{-j\omega 8}, & for\ 0 \le \omega \le \dfrac{\pi}{2} \\ 0\ , & for\ \dfrac{\pi}{2} \le \omega \le \pi \end{cases}$$

This is the desired frequency response of required lowpass filter.

ii) To obtain $H_d(\omega)$

$$H(k) = H_d(\omega)|_{\omega=\omega_k}$$

$$\omega_k = \frac{2\pi}{M}k,\ \ k = 0,1,2,3,\dots,M-1$$

$$H(k) = \begin{cases} e^{-j\frac{2\pi}{17}k8}, & for\ 0 \le \dfrac{2\pi}{17}k \le \dfrac{\pi}{2} \\ 0\ , & for\ \dfrac{\pi}{2} \le \dfrac{2\pi}{17}k \le \pi \end{cases}$$

Now for $\frac{2\pi}{17}k = 0, k = 0$ & $\frac{2\pi}{17}k = \frac{\pi}{2}, k = 17/4$

$$H(k) = \begin{cases} e^{-j\frac{16\pi k}{17}}, & for\ 0 \le k \le \dfrac{17}{4} == 0 \le k \le 4.25 \approx 4 \\ 0\ , & for\ \dfrac{17}{4} \le k \le \dfrac{17}{2} == 4.25 \approx 5 \le k \le 8 \end{cases}$$

The above equation gives sampled version of $H_d(\omega)$ i.e. H(k)

iii) To obtain h[n]

For odd value of M, h[n] is given as under

$$h[n] = \frac{1}{M}\left[H(0) + 2\sum_{k=1}^{\frac{M-1}{2}} Re\left\{H(k)e^{\frac{j2\pi kn}{M}}\right\}\right]$$

$$= \frac{1}{17}\left[1 + 2\sum_{k=1}^{8} Re\left\{H(k)e^{\frac{j2\pi kn}{17}}\right\}\right] = \frac{1}{17}\left[1 + 2\sum_{k=1}^{4} Re\left\{e^{-j\frac{16\pi k}{17}}e^{\frac{j2\pi kn}{17}}\right\}\right]$$

Solving,

$$h[n] = \frac{1}{17}\left[1 + 2\sum_{k=1}^{4}\cos\left(\frac{2\pi k(8-n)}{17}\right)\right],\ \ n = 0,1,2,3,\dots,16$$

This is the unit sample response of the FIR filter obtained using frequency sampling method.

Design of Optimum Equiripple Linear-Phase FIR filters:

The windowing method and frequency sampling method of FIR design are simple however they lack precise control in ω_p and ω_s.

This method is viewed as an optimum design criterion in the sense that the weighted approximation error between the desired frequency response and the actual frequency response is spread evenly across the passband and stopband of the filter minimizing the maximum error. The resulting filter designs have ripples in both the passband and the stopband.

To describe the design procedure, let us consider the design of lowpass filter with passband edge frequency ω_p and stopband edge frequency ω_s, then the filter frequency response satisfies the condition

$$1 - \delta_1 \leq H_r(\omega) \leq 1 + \delta_1 \quad |\omega| \leq \omega_p$$

Similarly, in the stopband, the filter frequency response is specified to fall between the limits $\pm\delta_2$

$$-\delta_2 \leq H_r(\omega) \leq \delta_2 \quad |\omega| > \omega_s$$

Thus δ_1 represents the ripple in the passband and δ_2 represents the attenuation or ripple in the stopband. The remaining filter parameter is M, the filter length or the number of filter coefficients.

Let us focus on the four different cases that result in a linear phase FIR:

Case 1: Symmetric unit sample response & M odd

$$h[n] = h[M - 1 - n]$$

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right)$$

If we let k = (M-1)/2 – n & define a new set of filter parameters {a(k)}

$$a(k) = \begin{cases} h\left(\frac{M-1}{2}\right) & k = 0 \\ 2h\left(\frac{M-1}{2} - k\right) & k = 1,2,3,---,\frac{M-1}{2} \end{cases}$$

Then

$$H_r(\omega) = \sum_{k=0}^{(M-1)/2} a(k)\cos\omega k$$

Case 2: Symmetric unit sample response & M even

$$h[n] = h[M-1-n]$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\cos\omega\left(\frac{M-1}{2}-n\right)$$

Again, we change the summation index from n to k = M/2 – n & define a new set of filter parameters {b(k)} as

$$b(k) = 2h\left(\frac{M}{2}-k\right), \qquad k = 1,2,3,--,\frac{M}{2}$$

Then

$$H_r(\omega) = \sum_{k=1}^{M/2} b(k)\cos\omega\left(k-\frac{1}{2}\right)$$

$$= \cos\frac{\omega}{2}\sum_{k=0}^{\frac{M}{2}-1}\hat{b}(k)\cos\omega k$$

Where,

$$\hat{b}(0) = \frac{1}{2}b(1)$$

$$\hat{b}(k) = 2b(k) - \hat{b}(k-1) \;\; k = 1,2,---,\frac{M}{2}-2$$

$$\hat{b}\left(\frac{M}{2}-1\right) = 2b\left(\frac{M}{2}\right)$$

Case 3: Antisymmetric unit sample response and M odd:

$$h[n] = -h[M-1-n]$$

$$H_r(\omega) = 2\sum_{n=0}^{(M-3)/2} h(n)\sin\omega\left(\frac{M-1}{2}-n\right)$$

Suppose k = (M-1)/2 - n

$$c(k) = 2h\left(\frac{M-1}{2}-k\right) \;\; k = 1,2,3,---,\frac{M-1}{2}$$

Then

$$H_r(\omega) = \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k$$
$$= \sin \omega \sum_{k=0}^{(M-3)/2} \hat{c}(k) \cos \omega k$$

Where,

$$\hat{c}\left(\frac{M-3}{2}\right) = c\left(\frac{M-1}{2}\right)$$
$$\hat{c}\left(\frac{M-5}{2}\right) = 2c\left(\frac{M-3}{2}\right)$$

$$\hat{c}(k-1) + \hat{c}(k+1) = 2c(k) \; 2 \le k \le \frac{M-5}{2}$$
$$\hat{c}(0) + \frac{1}{2}\hat{c}(2) = c(1)$$

Case 4: Antisymmetric unit sample response & M even:

$$h[n] = -h[M-1-n]$$

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left(\frac{M-1}{2} - n\right)$$

Suppose k = (M/2) - n

$$d(k) = 2h\left(\frac{M}{2} - k\right) \quad k = 1,2,3,---,\frac{M}{2}$$

Then

$$H_r(\omega) = \sum_{k=1}^{M/2} d(k) \sin \omega \left(k - \frac{1}{2}\right)$$
$$= \sin \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \hat{d}(k) \cos \omega k$$

Where,

$$\hat{d}\left(\frac{M}{2} - 1\right) = 2d\left(\frac{M}{2}\right)$$

$$\hat{d}(k-1) - \hat{d}(k) = 2d(k)\ 2 \le k \le \frac{M}{2} - 1$$

$$\hat{d}(0) - \frac{1}{2}\hat{d}(1) = d(1)$$

Now, to summarize, we know

$$H_r(\omega) = Q(\omega)P(\omega)$$

Where

$$Q(\omega) = \begin{cases} 1 & case\ 1 \\ \cos\frac{\omega}{2} & case\ 2 \\ \sin\omega & case\ 3 \\ \sin\frac{\omega}{2} & case\ 4 \end{cases}$$

And the P(ω) has the common form

$$P(\omega) = \sum_{k=0}^{L} \alpha(k)\cos\omega k$$

With {α(k)} representing the parameters of the filter which are linearly related to the unit response h(n) of the FIR filter. Suppose $H_{dr}(\omega)$ be the real valued desired frequency response and weighting function be W(ω). The weighting function on the approximation error allows us to choose the relative size of the errors in the different frequency bands.

It is convenient to normalize W(ω) to unity in stopband. So,

$$W(\omega) = \begin{cases} \delta_2/\delta_1 & : Passband \\ 1 & : Stopband \end{cases}$$

Then weighting approximation error is given as

$E(\omega) = W(\omega)[H_{dr}(\omega) - H_r(\omega)]$

$= W(\omega)\ [H_{dr}(\omega) - Q(\omega)P(\omega)]$

$= W(\omega)Q(\omega)[\ H_{dr}(\omega)/Q(\omega) - P(\omega)]$

$= \widehat{W}(\omega)[\widehat{H}_{dr}(\omega) - P(\omega)$

$\widehat{W}(\omega)$= modified weighting function

$\widehat{H}_{dr}(\omega)$ = modified desired frequency response

Given, the error function E(ω), the chebyshev approximation problem is to determine the filter parameters {α(k)} then minimize the maximum value of E(ω) over particular frequency band.

i.e.

$$\underset{over\{\alpha(k)\}}{min}\left[\underset{\omega \in s}{max}|E(\omega)|\right] = \underset{over\{\alpha(k)\}}{min}\left[\underset{\omega \in S}{max}\left|\widehat{W}(\omega)\left[\widehat{H}_{dr}(\omega) - \sum_{k=0}^{L}\alpha(k)\cos\omega k\right]\right|\right]$$

Where S is the set of frequency bands over which the optimization is to be performed.

Solution to the above problem is given by alternation theorem.

Alternation theorem:

Let S be a compact subset of the interval [0, π). A necessary and sufficient condition for

$$P(\omega) = \sum_{k=0}^{L}\alpha(k)\cos\omega k$$

To be unique, nest weighted chebyshev approximation to $\widehat{H}_{dr}(\omega)$ in S, is that the error function E(ω) exhibit at least L + 2 extremal frequencies in S. That is there must exist at least L + 2 frequencies $\{\omega_i\}$ in S such that $\omega_1 < \omega_2 < -- < \omega_{L+2}$, $E(\omega_i) = E(\omega_{i+1}$, and

$$|E(\omega_i)| = \underset{\omega \in S}{max}|E(\omega)| \;\; i = 1,2,---,L+2$$

We note that the error function E(ω) alternates in sign between two successive extremal frequencies. Hence the theorem is called the alternation theorem.

Now for the unique solution for the chebyshev optimization problem, at the desired extemal frequencies $\{\omega_n\}$, we have the set of equations,

$$\widehat{W}(\omega_n)\left[\widehat{H}_{dr}(\omega_n) - P(\omega_n)\right] = (-1)^n\delta \quad n = 0,1,2,--,L+2$$

Where δ = maximum value of E(ω). ($\delta = \delta_2$)

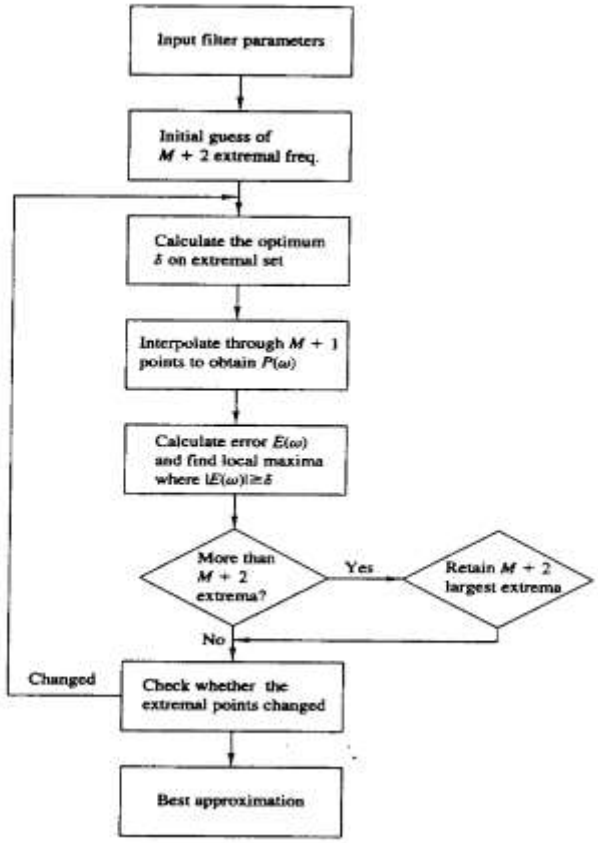
$$P(\omega_n) + \frac{(-1)^n\delta}{\widehat{W}(\omega_n)} = \widehat{H}_{dr}(\omega_n) \quad n = 0,1,2,--,L+2$$

$$\sum_{k=0}^{L}\alpha(k)\cos\omega_n k + \frac{(-1)^n\delta}{\widehat{W}(\omega_n)} = \widehat{H}_{dr}(\omega_n) \quad n = 0,1,2,--,L+2$$

We should find α(k) & δ. So can be expressed in matrix form.

$$\begin{bmatrix} 1 & \cos\omega_0 & \cos 2\omega_0 & \cdots & \cos L\omega_0 & 1/\widehat{W}(\omega_0) \\ 1 & \cos\omega_1 & \cos 2\omega_1 & & \cos L\omega_1 & -1/\widehat{W}(\omega_1) \\ & \vdots & & \ddots & \vdots & \\ 1 & \cos\omega_{L+1} & \cos 2\omega_{L+1} & \cdots & \cos L\omega_{L+1} & (-1)^{L+1}/\widehat{W}(\omega_{L+1}) \end{bmatrix} \begin{bmatrix} \alpha(0) \\ \alpha(1) \\ \vdots \\ \vdots \\ \alpha(L) \\ \delta \end{bmatrix} = \begin{bmatrix} \widehat{H}_{dr}(\omega_0) \\ \widehat{H}_{dr}(\omega_1) \\ \vdots \\ \vdots \\ \widehat{H}_{dr}(\omega_{L+1}) \end{bmatrix}$$

Initially, we know neither the set of extermal frequency $\{\omega_n\}$ nor the parameter $\{\alpha(k)\}$. To solve for the parameter, we use an iterative algorithm, called REMEZ EXCHANGE ALGORITHM.

Remez Exchange Algorithm:	**Flow chart for R Remez Exchange Algorithm**
1. Guess L+2 extremal frequencies. 2. Determine P(ω) and δ. 3. Compute the error function E(ω). 4. If $\|E(\omega)\| \geq \delta$, determine the another set of L+2 extremal frequencies. 5. Repeat 1 to 4 untill it converges to the optimal set of extremal frequencies. 6. Obtain the best approximation.	

Chapter 7: Design of IIR Digital Filters[i].

Characteristics of Commonly used Analog Filters

IIR digital filters can easily be obtained by beginning with an analog filter and then using a mapping to transform the s-plane to the z-plane. Thus the design of a digital filter is reduced to designing an appropriate analog filter and then performing the conversion from H(s) to H(z), in such a way so as to preserve as much as possible, the desired characteristics of the analog filter.

Analog filter is a well-developed field and many books have been written on the subject. Analog filters are good, robust and have been started since centuries.

We will be describing the importance characteristics of commonly used analog filter and introduce the relevant filter parameters. Our discussion is limited to lowpass filters.

SOME PRELIMINARIES

We discuss two preliminary issues in this section. First, we consider the magnitude squared response specifications, which are more typical of analog (and hence of IIR) filters. These specifications are given on the relative linear scale. Second, we study the properties of the magnitude-squared response.

Let $H_a(j\Omega)$ be the frequency response of an analog filter. Then the lowpass filter specifications on the magnitude-squared response are given by

$$\frac{1}{1+\varepsilon^2} \le |H_a(j\Omega)|^2 \le 1, |\Omega| \le \Omega_p$$
$$0 \le |H_a(j\Omega)|^2 \le \delta_2^2, |\Omega| \ge \Omega_s$$

Where ε is a passband ripple parameter, Ω_p is the passband cutoff frequency in rad/sec, δ_2 is a stopband attenuation parameter, and Ω_s is the stopband cutoff in rad/sec. These specifications are shown in figure 1, from which we observe that $|H_a(j\Omega)|^2$ must satisfy

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2} \; at \; \Omega = \Omega_p$$
$$|H_a(j\Omega)|^2 = \delta_2^2 \; at \; \Omega = \Omega_s$$

The manner in which specifications of a lowpass filter are given to the engineer is illustrated in figure 1(b).

The attenuation α, can be given as,

$\alpha(\Omega) = -20\log|H_a(j\Omega)|$ measured in dB which makes,

$|H_a(j\Omega)| = 10^{-\alpha(\Omega)/20}$

For the passband extending from $\Omega = 0$ to $\Omega = \Omega_p$ the attenuation should not be larger than α_{max}. from Ω_p to Ω_s we have a transition band. Then the specifications indicate that from Ω_s on for all higher frequencies the attenuation should not be less than α_{min}.

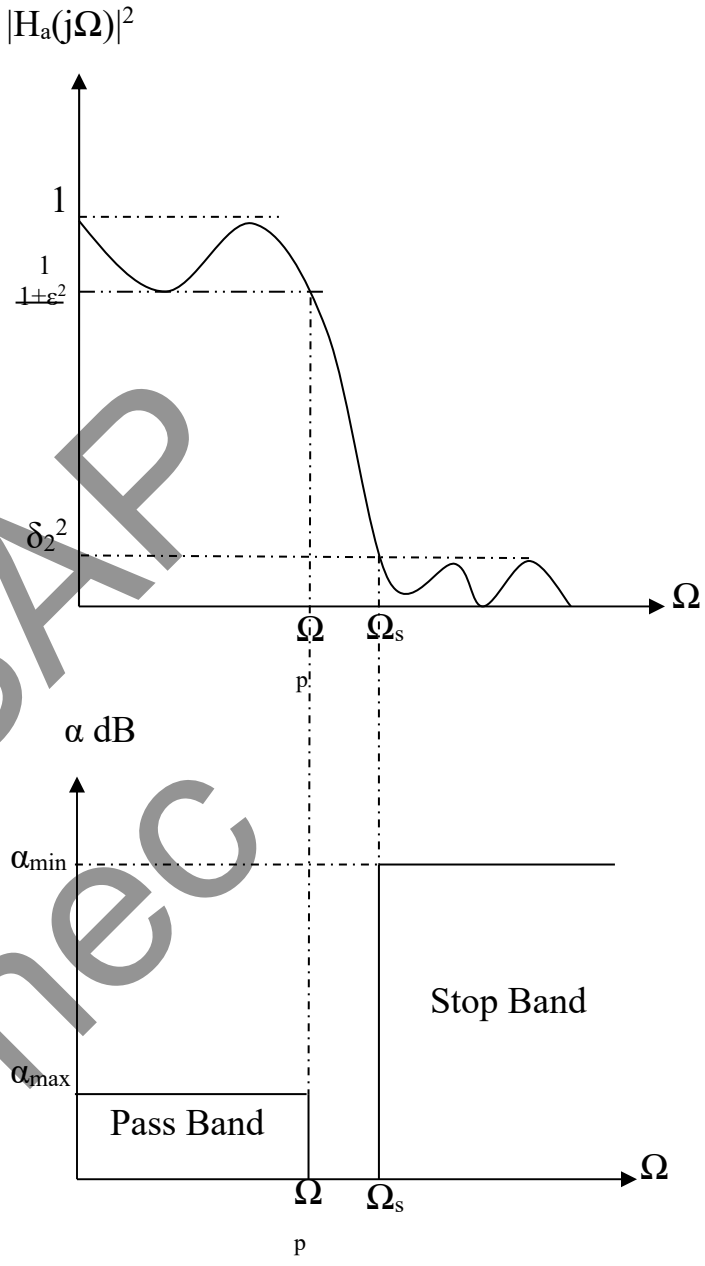


Figure (a) Magnitude squared response of analog filter.
(b) Attenuation curve

The parameters ε and δ_2 are related to parameters α_{max} and α_{min} respectively, of the dB scale. These relations are given by,

$$\alpha_{max} = -20\log\left(\sqrt{\frac{1}{1+\varepsilon^2}}\right) \Rightarrow \varepsilon = \sqrt{10^{0.1\alpha_{max}} - 1}$$

$$\& \ \alpha_{min} = -20log\delta_2 \Rightarrow \delta_2 = 10^{-\frac{\alpha_{min}}{20}}$$

$$\delta = \sqrt{10^{0.1\alpha_{min}-1}}, where\ \delta^2 = \frac{1}{\delta_2^2} - 1$$

Butterworth lowpass filter (Maximally flat response)

This filter is characterized by the property that its magnitude response is flat in both passband and stopband. The magnitude-squared response for an N^{th} order lowpass filter is given by:

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} - - - (i)$$

Where N is the order of the filter and Ω_c is the cutoff frequency in rad/sec. The plot of the magnitude-squared response is shown below:

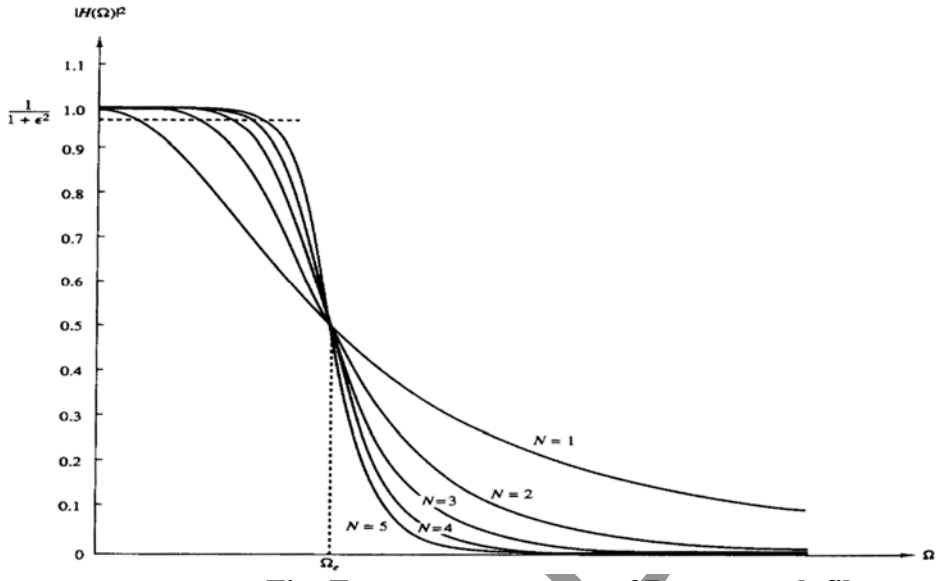


Fig: Frequency response of Butterworth filters.

From this plot we can observe the following properties:

- At Ω = 0, $|H_a(j0)|^2 = 1$ for all N.
- At Ω = Ω_c, $|H_a(j\Omega_c)|^2 = 0.5$ for all N, which implies a 3 dB attenuation at Ω_c.
- $|H_a(j\Omega)|^2$ is a monotonically decreasing function of Ω.
- $|H_a(j\Omega)|^2$ approaches an ideal lowpass filter as N tends to infinity(∞).
- $|H_a(j\Omega)|^2$ is maximally flat at Ω = 0 since derivatives of all orders exist and are equal to zero.

Now,

$$|\mathrm{H_a(j\Omega)}|^2 = \frac{1}{1+\left(\Omega/\Omega_c\right)^{2N}} = \frac{1}{1+\varepsilon^2\left(\Omega/\Omega_p\right)^{2N}}$$

Where, Ω is the normalized frequency.

The order of the filter required to meet an attenuation δ_2 at a specified frequency Ω_s is easily determined from above equation. Thus at Ω = Ω_s we have

$$\frac{1}{1+\varepsilon^2\left(\Omega_s/\Omega_p\right)^{2N}} = \delta_2^2$$

Simplifying above equation we get

$$N = \frac{log_{10}\left(\delta/\varepsilon\right)}{log_{10}\left(\Omega_s/\Omega_p\right)}$$

Where,

$$\varepsilon^2 = 10^{0.1\alpha_{max}} - 1 \; and \; \delta^2 = 10^{0.1\alpha_{min}} - 1 = \frac{1}{{\delta_2}^2} - 1$$

Thus the butterworth filter is completely characterized by the parameters N, δ_2, ε and the ratio Ω_s/Ω_p. The transfer function with Butterworth response is given by

$$H_a(s) = \frac{1}{B_N(s)}$$

Where,

$$B_N(s) = 1 \; or \; (s+1) \times \prod (s^2 + 2cos\psi_k s + 1)$$

Two simple rules permit us to determine ψ_k

1. a) If N is odd, then there is a pole at $\psi=0^0$;
   b) If N is even, then there are poles at $\psi=\pm 90^0/N$
2. Poles are separated by $\psi=180^0/N$

Example: for N=2, poles are at $\psi=\pm 90^0/N=\pm 45^0$

$$B_2(s) = (s^2 + 2cos45s + 1) = s^2 + 1.414s + 1$$

The table of the transfer function for different order of the Butterworth filter is given as:

Order of Filter N	**Transfer function H(s) = 1/A(s) where A(s)**
1	(s+1)
2	(s^2+1.414s+1)
3	(s^2+s+1)(s+1)
4	(s^2+0.766s+1)(s^2+1.848s+1)
5	(s^2+0.618s+1)(s^2+1.618s+1)(s+1)
6	(s^2+0.518s+1)(s^2+4.414s+1)(s^2+1.932s+1)
7	(s^2+1.802s+1)(s^2+1.247s+1)(s^2+0.445s+1)(s+1)

If -3 dB frequency, or cut-off or center frequency is given we use the following formula,

$$N = \frac{log_{10}\left(1/\delta_2^2 - 1\right)}{2log_{10}\left(\Omega_s/\Omega_c\right)} = \frac{\log(10^{0.1\alpha_{min}} - 1)}{2log_{10}\left(\Omega_s/\Omega_c\right)}$$

To find Ω_c from N, we have,

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\Omega/\Omega_c\right)^{2N}}$$

Simplifying above equation we get,

$$\frac{\Omega_s}{\Omega_c} = (10^{0.1\alpha_{min}} - 1)^{1/2N} = \delta^{1/N}$$

$$\therefore \Omega_c = \frac{\Omega_s}{\delta^{1/N}}$$

Matching the frequency response exactly at stopband.

If we were instead to match the frequency response at passband, we would obtain

$$\Omega_c = \frac{\Omega_p}{\varepsilon^{1/N}}$$

In principle, any value of the critical frequency that satisfies

$$\frac{\Omega_p}{\varepsilon^{1/N}} \le \Omega_c \le \frac{\Omega_s}{\delta^{1/N}}$$

would be valid.

Chebyshev Lowpass filters (Equi-ripple Magnitude response)

There are two types of Chebyshev filters. The Chebyshev-I filters have equiripple response in the passband, while the Chebyshev-II filters have equiripple response in the stopband. Butterworth filters have monotone response in both bands. It is noted that by choosing a filter that has an equiripple rather than a monotonic behavior, we can obtain lower-order filter. Therefore Chebyshev filters provide lower order than Butterworth filters for the same specifications.

The magnitude-squared response of a Chebyshev-I filter is

$$|\mathrm{H_a(j\Omega)}|^2 = \frac{1}{1+\epsilon^2\, T_N^2\left(\Omega/\Omega_c\right)}$$

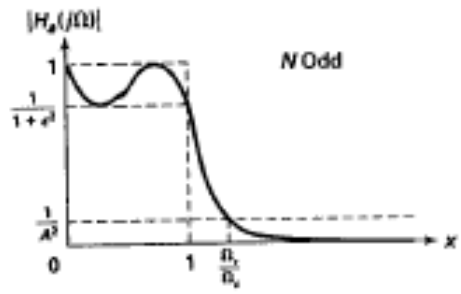
Where N is the order of the filter, ε is the passband ripple factor, and $T_N(x)$ is the N^{th} order Chebyshev polynomial given by

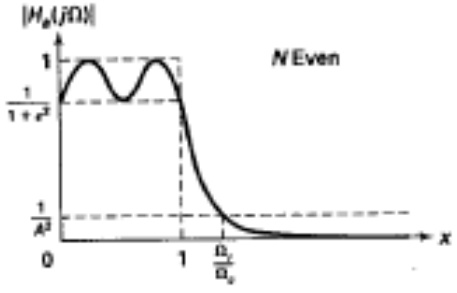
$$T_N(x) = \begin{cases} \cos\left(N\cos^{-1}(x)\right), & 0 \le x \le 1 \\ \cosh\left(\cosh^{-1}(x)\right), & 1 < x < \infty \end{cases}$$

Where $x = \Omega/\Omega_c$.

The equiripple response of the Chebyshev filters is due to this polynomial $T_N(x)$. Its key properties are (a) for $0 < x < 1$, $T_N(x)$ oscillates between -1 and 1, and (b) for $1 < x < \infty$, $T_N(x)$ increases monotonically to ∞.

There are two possible shapes of $|\mathrm{H_a(j\Omega)}|^2$, one for N odd and one for N even as shown below. Note that $x = \Omega/\Omega_c$ is the normalized frequency.





From the above two response plots we observe the following properties:

- At $x = 0$ (or $\Omega = 0$):

$$|\mathrm{H_a(j0)}|^2 = 1, \quad \text{for N odd.}$$

$$|\mathrm{H_a(j0)}|^2 = \frac{1}{1+\epsilon^2} \quad for\ N\ even.$$

- At $x = 1$ (or $\Omega = \Omega_c$):

$$|H_a(j1)|^2 = \frac{1}{1+\epsilon^2} \quad for\ all\ N.$$

- For $0 \le x \le 1$ (or $0 \le \Omega \le \Omega_c$), $|H_a(j\Omega)|^2$ oscillates between 1 and $\frac{1}{1+\epsilon^2}$.
- For $x > 1$ (or $\Omega > \Omega_c$), $|H_a(j\Omega)|^2$ decreases monotonically to 0.

At $x=\Omega_r$, $|H_a(jx)|^2 = \delta_2{}^2$.
Design Equations:
Given Ω_p, Ω_s, α_{max}, α_{min}; three parameters are required to determine a chebyshev-filter:
ε, Ω_c & N

$$N = \frac{cosh^{-1}\left(\delta/\varepsilon\right)}{cosh^{-1}\left(\Omega_s/\Omega_p\right)}$$

Elliptic filters:

Elliptic (or Cauer) filters exhibit equiripple behavior in both the passband and the stopband, as illustrated in figure below for N odd and N even. This class of filters contains both poles and zeros and is characterized by the magnitude squared frequency response

$$|H(\Omega)|^2 = \frac{1}{1+\varepsilon^2 U_N\left(\frac{\Omega}{\Omega_p}\right)}$$

Design of IIR Filters:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

IIR filters can be obtained from analog filters by using mapping function. They have infinite impulse response and contain poles, so they are unstable.

Firstly Analog filters is designed from the desired specifications then digital filters is obtained by mapping s-domain to z-domain.

H(s) ---(mapping)→ H(z)

Analog filter design is a mature and well developed field, so it is not surprising that we begin the design of a digital filter in the analog domain and then convert the design into the digital domain.

An analog filter can be described by its system function,

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k}$$

Where $\{\alpha_k\}$ and $\{\beta_k\}$ are the filter coefficients, or by its impulse response, which is related to $H_a(s)$ by the Laplace transform

$$H_a(s) = \int_{-\infty}^{\infty} h(t) e^{-st} dt$$

Alternatively, the analog filter having the rational system function H(s) can be described by the linear constant-coefficient differential equation

$$\sum_{k=0}^{N} \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} \beta_k \frac{d^k x(t)}{dt^k}$$

Where x(t) denotes the input signal and y(t) denotes the output of the filter.

Each of these three equivalent characterizations of an analog filter leads to alternative methods for converting the filter into the digital domain. We recall that an analog linear time-invariant system with system function H(s) is stable if all its poles lie in the left half of the s-plane. Consequently, if the conversion technique is to be effective, it should possess the following desrirable properties:

1. The jΩ axis in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship between the two frequency variables in the two domains.
2. The left-half plane (LHP) of the s-plane should map into the inside of the unit circle in the z-plane. Thus a stable analog filter will be converted to a stable digital filter.

IIR Filter Design:

H(s) –(inverse Laplace transform)→ **$h_a(t)$** –(sampling)→ **h[n]** –(Z-transform)→ **H(z)**

Techniques:

1. By approximation of derivative method (**not needed in our course)**
2. By Impulse invariance method. (**imp.**)
3. By Bilinear transformation method. (**imp.**)
4. By Matched z-Transform method (**only for short notes**)

Impulse Invariance Method:

In the impulse invariance method, the objective is to design an IIR filter having a unit sample response h[n] that is the sampled version of the impulse response of the analog filter. That is

$$h[n] \equiv h(nT) \;\; n = 0,1,2,3, \ldots$$

where T is the sampling interval.

We know when a continuous time signal $x_a(t)$ with spectrum $X_a(F)$ is sampled at a rate $F_s = 1/T$ samples per second, the spectrum of the sampled signal is the periodic repetition of the scaled spectrum $F_s X_a(F)$ with period F_s. Specifically, the relation is

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a[(f-k)F_s] \; --- (1)$$

Where $f = F/F_s$ is the normalized frequency. Aliasing occurs if the sampling rate F_s is less than twice the higest frequency contained in $X_a(F)$.

Expressed in the context of sampling the impulse response of an analog filter with frequency response $H_a(F)$, the digital filter with unit sample response $h(n) = h_a(nT)$ has the frequency response

$$H(f) = F_s \sum_{k=-\infty}^{\infty} H_a[(f-k)F_s]$$

Or, equivalentely,

$$H(\omega) = F_s \sum_{k=-\infty}^{\infty} H_a[(\omega - 2\pi k)F_s] --- (2)$$

Or,

$$H(\Omega T) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(\Omega - \frac{2\pi k}{T}\right)$$

Mapping relates the z-transform of h[n] to the Laplace transform of $h_a(t)$ by (generalization of above equation):

$$H(z)|_{z=e^{sT}} = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(s - j\frac{2\pi k}{T}\right) - - - (3)$$

Where

$$H(z) = \sum_{n=0}^{\infty} h[n]\, z^{-n} - - - (4)$$

$$H(z)|_{z=e^{sT}} = \sum_{n=0}^{\infty} h[n]\, e^{-sTn} - - - (5)$$

Note that when s = jΩ, eq^{n}() reduces to (), where the factor of j in $H_a(\Omega)$ is suppressed in our notation.

The Mapping Relation is given by:

$$z = e^{sT}$$

If we substitute s =

$$z = e^{sT} \tag{8.3.24}$$

If we substitute $s = \sigma + j\Omega$ and express the complex variable z in polar form as $z = re^{j\omega}$, (8.3.24) becomes

$$re^{j\omega} = e^{\sigma T} e^{j\Omega T}$$

Clearly, we must have

$$\begin{aligned} r &= e^{\sigma T} \\ \omega &= \Omega T \end{aligned} \tag{8.3.25}$$

Consequently, $\sigma < 0$ implies that $0 < r < 1$ and $\sigma > 0$ implies that $r > 1$. When $\sigma = 0$, we have $r = 1$. Therefore, the LHP in s is mapped inside the unit circle in z and the RHP in s is mapped outside the unit circle in z.

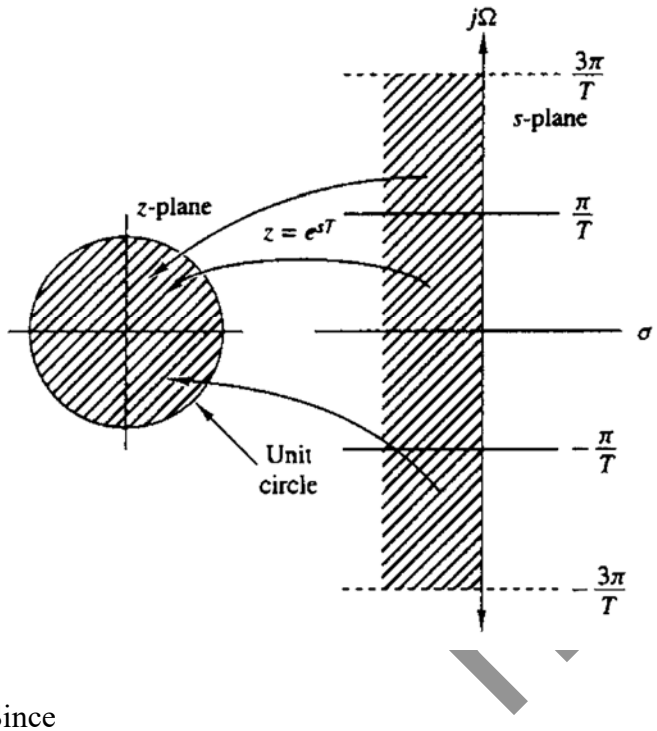


Figure 8.32 The mapping of $z = e^{sT}$ maps strips of width $2\pi/T$ (for $\sigma < 0$) in the s-plane into points in the unit circle in the z-plane.

Since

$\omega = \Omega T$, $(2k-1)\pi/T \leq \Omega \leq (2k+1)\pi/T$ maps into the interval

$-\pi \leq \omega \leq \pi$, where k is an integer.

Thus the mapping from the analog frequency Ω to the frequency variable ω in the digital domain in many-to-one, which simply reflects the effects of aliasing due to sampling. The Impulse invariance method is in-appropriate for designing highpass filters due to the spectrum aliasing that results from the sampling process.

To explore further the effect of the impulse invariance design method on the characteristics of the resulting filter, let us express the system function of the analog filter in partial-fraction form. On the assumption that the poles of the analog filter are distinct, we can write

$$H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s - p_k} \tag{8.3.26}$$

where $\{p_k\}$ are the poles of the analog filter and $\{c_k\}$ are the coefficients in the partial-fraction expansion. Consequently,

$$h_a(t) = \sum_{k=1}^{N} c_k e^{p_k t} \qquad t \geq 0 \tag{8.3.27}$$

If we sample $h_a(t)$ periodically at $t = nT$, we have

$$\begin{aligned} h(n) &= h_a(nT) \\ &= \sum_{k=1}^{N} c_k e^{p_k T n} \end{aligned} \tag{8.3.28}$$

Now, with the substitution of (8.3.28), the system function of the resulting digital IIR filter becomes

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n) z^{-n} \\ &= \sum_{n=0}^{\infty} \left(\sum_{k=1}^{N} c_k e^{p_k T n} \right) z^{-n} \\ &= \sum_{k=1}^{N} c_k \sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n \end{aligned} \tag{8.3.29}$$

The inner sum in (8.3.29) converges because $p_k < 0$ and yields

$$\sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n = \frac{1}{1 - e^{p_k T} z^{-1}} \tag{8.3.30}$$

Therefore, the system function of the digital filter is

$$H(z) = \sum_{k=1}^{N} \frac{c_k}{1 - e^{p_k T} z^{-1}} \tag{8.3.31}$$

We observe that the digital filter has poles at

$$z_k = e^{p_k T} \qquad k = 1, 2, \ldots, N \tag{8.3.32}$$

Mapping

Analog Domain H(s)	Digital Domain H(z)
$\frac{1}{s-p_k}$	$\frac{1}{1-e^{p_kT}z^{-1}}$
$\frac{s+a}{(s+a)^2+b^2}$	$\frac{1-e^{-aT}cos(bT)z^{-1}}{1-2e^{-aT}cos(bT)z^{-1}+e^{-2aT}z^{-2}}$
$\frac{b}{(s+a)^2+b^2}$	$\frac{e^{-aT}sin(bT)z^{-1}}{1-2e^{-aT}cos(bT)z^{-1}+e^{-2aT}z^{-2}}$

Example:

Convert the analog filter with system function

$$H_a(s) = \frac{s+0.1}{(s+0.1)^2+9}$$

into a digital IIR filter by means of the impulse invariance method.

Solution We note that the analog filter has a zero at $s = -0.1$ and a pair of complex-conjugate poles at

$$p_k = -0.1 \pm j3$$

as illustrated in Fig. 8.33.

We do not have to determine the impulse response $h_a(t)$ in order to design the digital IIR filter based on the method of impulse invariance. Instead, we directly determine $H(z)$, as given by (8.2.31), from the partial-fraction expansion of $H_a(s)$. Thus we have

$$H(s) = \frac{\frac{1}{2}}{s+0.1-j3} + \frac{\frac{1}{2}}{s+0.1+j3}$$

Then

$$H(z) = \frac{\frac{1}{2}}{1-e^{-0.1T}e^{j3T}z^{-1}} + \frac{\frac{1}{2}}{1-e^{-0.1T}e^{-j3T}z^{-1}}$$

IIR Filter Design by the Bilinear Transformation:

Drawbacks of Impulse Invariance methods are:

- Many to one mapping
- Due to which aliasing effect occurs
- Appropriate for lowpass and limited class of bandpass filters

The bilinear transformation is a conformal mapping that transforms the jΩ-axis into the unit circle in the Z-plane only once, thus avoiding aliasing of frequency components.
Let us consider an analog filter with system function,

$$H(s) = \frac{b}{s+a}$$ ------- (i)

This system is also characterized by the differential equation:

$$\frac{dy(t)}{dt} + ay(t) = bx(t)$$ -------- (ii)

We integraten the derivative and approximate the integral by trapezoidal formula,

$$y(t) = \int_{t_0}^{t} y'(\tau)d\tau + y(t_0)$$ ------- (iii)

Where, y'(t) denotes the derivative of y(t)
The approximation of the integral by the trapezoidal formula at t = nT and t_0 = nT-T yields,

$$y(nT) = \frac{T}{2}[y'(nT) + y'(nT - T)] + y(nT - T)$$ ----- (iv)

But from (ii)

$$y'(nT) = -ay(nT) + bx(nT)$$ ----- (v)

Now substituting y'(nT) in equation (iv)
y[n] = T/2{-ay[n] + bx[n] + (-ay[n-1] + bx[n-1])} + y[n-1]
where x(nT) = x[n] and y(nT) = y[n]. Simplifying above equation:

$$\left(1 + \frac{aT}{2}\right) y(n) - \left(1 - \frac{aT}{2}\right) y(n-1) = \frac{bT}{2}[x(n) + x(n-1)]$$

Taking z-transform

$$\left(1 + \frac{aT}{2}\right) Y(z) - \left(1 - \frac{aT}{2}\right) z^{-1}Y(z) = \frac{bT}{2}(1 + z^{-1})X(z)$$

Consequently the equivalent digital filter is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(bT/2)(1 + z^{-1})}{1 + aT/2 - (1 - aT/2)z^{-1}}$$

$$H(z) = \frac{b}{\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) + a}$$

Or equivalently, -------- (vi)

Clearly, the mapping from the s-plane to the Z-plane is

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

This is called Bilinear Transformation.

To investigate the characteristics of the bilinear transformation, let

$$z = re^{j\omega}$$

$$s = \sigma + j\Omega$$

Then (8.3.40) can be expressed as

$$\begin{aligned} s &= \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{re^{j\omega}-1}{re^{j\omega}+1} \\ &= \frac{2}{T}\left(\frac{r^2-1}{1+r^2+2r\cos\omega} + j\frac{2r\sin\omega}{1+r^2+2r\cos\omega}\right) \end{aligned}$$

Consequently,

$$\sigma = \frac{2}{T}\frac{r^2-1}{1+r^2+2r\cos\omega} \tag{8.3.41}$$

$$\Omega = \frac{2}{T}\frac{2r\sin\omega}{1+r^2+2r\cos\omega} \tag{8.3.42}$$

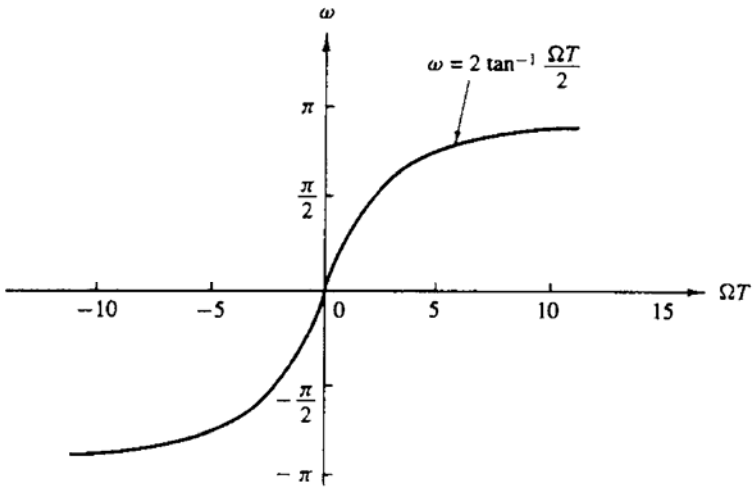


Figure 8.36 Mapping between the frequency variables ω and Ω resulting from the bilinear transformation.

First, we note that if $r < 1$, then $\sigma < 0$, and if $r > 1$, then $\sigma > 0$. Consequently, the LHP in s maps into the inside of the unit circle in the z-plane and the RHP in s maps into the outside of the unit circle. When $r = 1$, then $\sigma = 0$ and

$$\Omega = \frac{2}{T}\frac{\sin\omega}{1+\cos\omega}$$
$$= \frac{2}{T}\tan\frac{\omega}{2} \tag{8.3.43}$$

or, equivalently,

$$\omega = 2\tan^{-1}\frac{\Omega T}{2} \tag{8.3.44}$$

The relationship in (8.3.44) between the frequency variables in the two domains is illustrated in Fig. 8.36. We observe that the entire range in Ω is mapped only once into the range $-\pi \le \omega \le \pi$. However, the mapping is highly nonlinear. We observe a frequency compression or *frequency warping*, as it is usually called, due to the nonlinearity of the arctangent function.

Example of Bilinear Transformation

Design a single-pole lowpass digital filter with a 3-dB bandwidth of 0.2π, using the bilinear transformation applied to the analog filter

$$H(s) = \frac{\Omega_c}{s+\Omega_c}$$

where Ω_c is the 3-dB bandwidth of the analog filter.

Solution The digital filter is specified to have its -3-dB gain at $\omega_c = 0.2\pi$. In the frequency domain of the analog filter $\omega_c = 0.2\pi$ corresponds to

$$\Omega_c = \frac{2}{T}\tan 0.1\pi$$
$$= \frac{0.65}{T}$$

Thus the analog filter has the system function

$$H(s) = \frac{0.65/T}{s+0.65/T}$$

This represents our filter design in the analog domain.

Now, we apply the bilinear transformation given by (8.3.40) to convert the analog filter into the desired digital filter. Thus we obtain

$$H(z) = \frac{0.245(1+z^{-1})}{1-0.509z^{-1}}$$

where the parameter T has been divided out.

The frequency response of the digital filter is

$$H(\omega) = \frac{0.245(1+e^{-j\omega})}{1-0.509e^{-j\omega}}$$

At $\omega = 0$, $H(0) = 1$, and at $\omega = 0.2\pi$, we have $|H(0.2\pi)| = 0.707$, which is the desired response.

Frequency Transformation

In the Analog Domain

Suppose we have a lowpass filter with passband edge frequency Ω_p and we wish to convert it to another filter with passband edge frequency Ω_p',

Type of transformation	Transformation	Band edge frequencies of new filter
Lowpass	$s \rightarrow \frac{\Omega_p}{\Omega_p'} s$	Ω_p'
Highpass	$s \rightarrow \frac{\Omega_p \Omega_p'}{s}$	Ω_p'
Bandpass	$s \rightarrow \Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}$	Ω_l, Ω_u
Bandstop	$s \rightarrow \Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_l \Omega_u}$	Ω_l, Ω_u

In the Digital Domain

Type of transformation	Transformation	Parameters
Lowpass	$z^{-1} \longrightarrow \frac{z^{-1} - a}{1 - az^{-1}}$	ω_p' = band edge frequency of new filter $a = \frac{\sin[(\omega_p - \omega_p')/2]}{\sin[(\omega_p + \omega_p')/2]}$
Highpass	$z^{-1} \longrightarrow -\frac{z^{-1} + a}{1 + az^{-1}}$	ω_p' = band edge frequency new filter $a = -\frac{\cos[(\omega_p + \omega_p')/2]}{\cos[(\omega_p - \omega_p')/2]}$
Bandpass	$z^{-1} \longrightarrow -\frac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = -2\alpha K/(K+1)$ $a_2 = (K-1)/(K+1)$ $\alpha = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \cot\frac{\omega_u - \omega_l}{2} \tan\frac{\omega_p}{2}$
Bandstop	$z^{-1} \longrightarrow -\frac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = -2\alpha/(K+1)$ $a_2 = (1-K)/(1+K)$ $\alpha = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \tan\frac{\omega_u - \omega_l}{2} \tan\frac{\omega_p}{2}$

References:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000.
2. S. Sharma, "Digital Signal Processing", Third Revised Edition, S.K. Kataria & Sons, 2007.

[i] Compiled By: Rupesh Dahi Shrestha, Assistant Professor, Dept. of Electrical and Electronics, nec, 2011