



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

**SCHOOL OF BIO AND CHEMICAL ENGINEERING
DEPARTMENT OF BIOMEDICAL ENGINEERING**

UNIT – I – Digital Signal Processing & Its Applications – SEC1324

INTRODUCTION TO DISCRETE TIME SIGNALS AND SYSTEMS

A signal is a function of independent variables such as time, distance, position, temperature and pressure. A signal carries information, and the objective of signal processing is to extract useful information carried by the signal. Signal processing is concerned with the mathematical representation of the signal and the algorithmic operation carried out on it to extract the information present. For most purposes of description and analysis, a signal can be defined simply as a mathematical function, y where x is the independent variable .

$$y = f(x)$$

signal e.g.: $y = \sin(\omega t)$ is a function of a variable in the time domain and is thus a time signal

$X(\omega) = 1/(-m\omega^2 + ic\omega + k)$ is a frequency domain signal; An image $I(x,y)$ is in the spatial domain

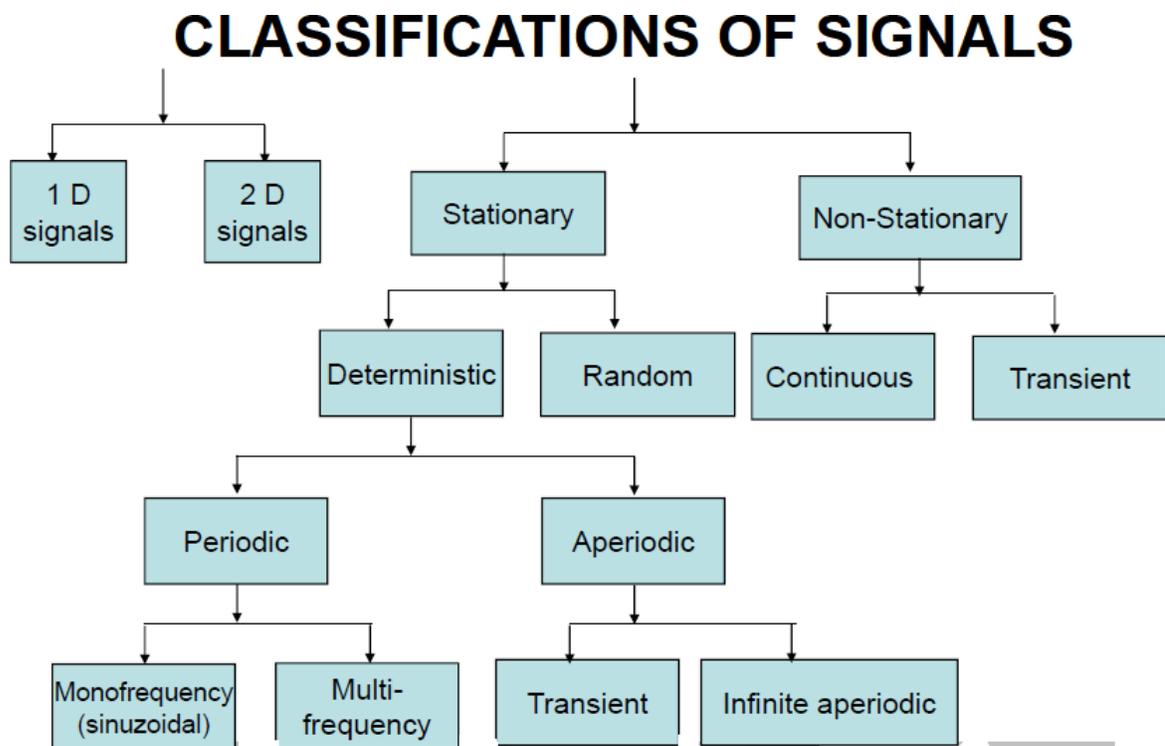


Fig.1:Classification

at $t=0$, will have the same motions at all time. There is no place for uncertainty here. If we can uniquely specify the value of θ for all time, *i.e.*, we know the underlying functional relationship between t and θ , the motion is **deterministic** or predictable. In other words, a signal that can be uniquely determined by a well defined process such as a mathematical expression or rule is called a **deterministic signal**. The opposite situation occurs if we know all the physics there is to know, but still cannot say what the signal will be at the next time instant-then the signal is **random** or **probabilistic**. In other words, a signal that is generated in a random fashion and can not be predicted ahead of time is called a **random signal**.

1.2 EXAMPLES OF SIGNALS

For a simple pendulum as shown, basic definition is: where θ_m is the peak amplitude of the motion and $\omega = \sqrt{l/g}$ with l the length of the pendulum and g the acceleration due to gravity. As the system has a constant amplitude (we assume no damping for now), a constant frequency (dictated by physics) and an initial condition ($\theta=0$ when $t=0$), we know the value of $\theta(t)$ for all time

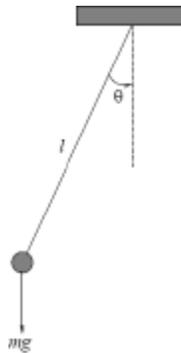
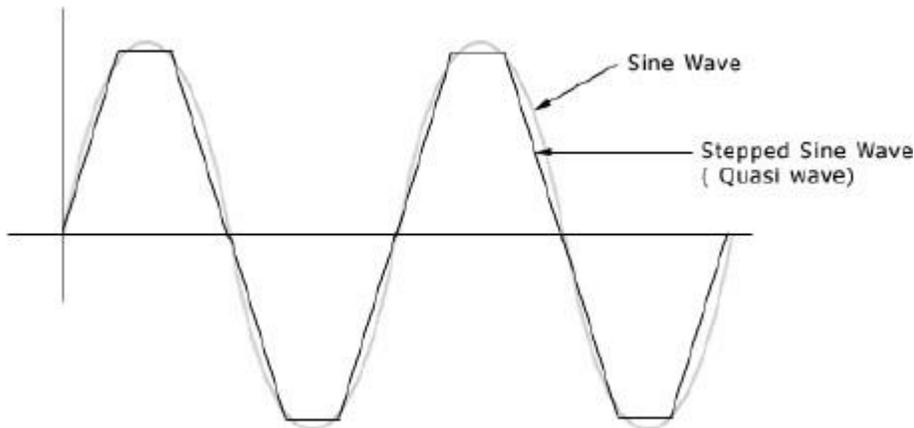


Fig. 2: Typical examples to deterministic signals are sine chirp and digital stepped sine.



1.3 Random signals are characterized by having many frequency components present over a wide range of frequencies. The amplitude versus time appears to vary rapidly and unsteadily

with time. The ‘shhhh’ sound is a good example that is rather easy to observe using a microphone and oscilloscope. If the sound intensity is constant with time, the random signal is stationary, while if the sound intensity varies with time the signal is nonstationary. One can easily see and hear this variation while making the ‘shhhh’ sound.

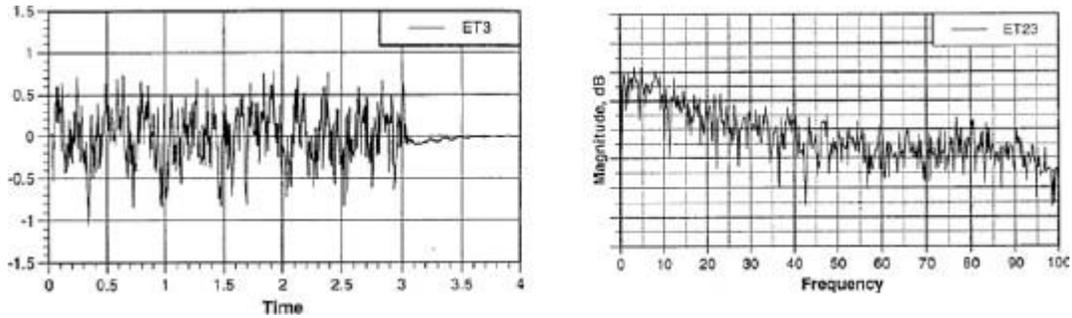


Fig. 3: Random signal

Random signals are characterized by analyzing the statistical characteristics across an ensemble of records. Then, if the process is ergodic, the time (temporal) statistical characteristics are the same as the ensemble statistical characteristics. The word temporal means that a time average definition is used in place of an ensemble statistical definition

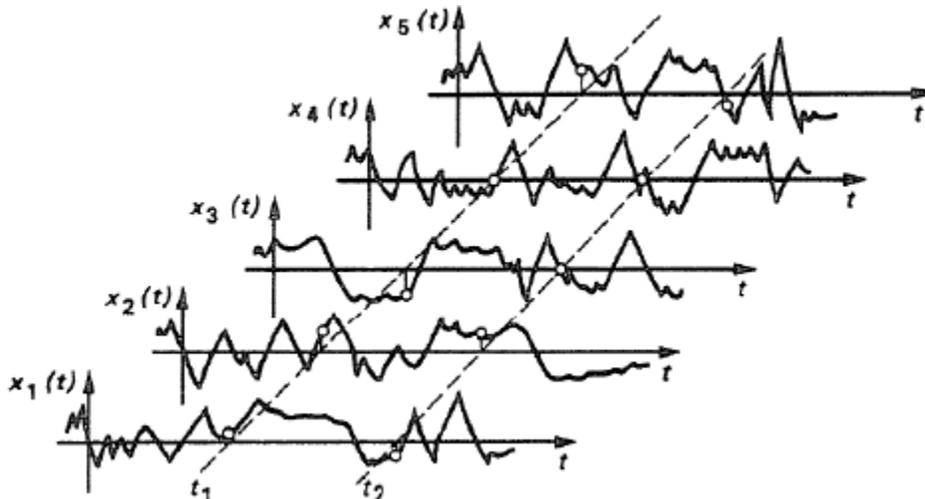


Fig. 4: Transient signal

1.4. Transient signals

may be defined as signals that exist for a finite range of time as shown in the figure. Typical examples are hammer excitation of systems, explosion and shock loading etc. It should be noted that periodicity does not necessarily mean a sinusoidal signal as shown in the figure.

$$S(t) = \sum_{i=1}^{\infty} s_i \sin\left(\frac{2\pi i}{T} \cdot t\right)$$

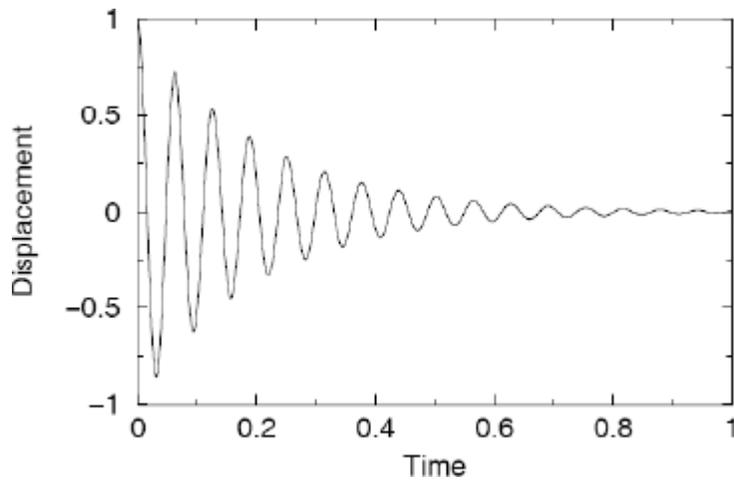
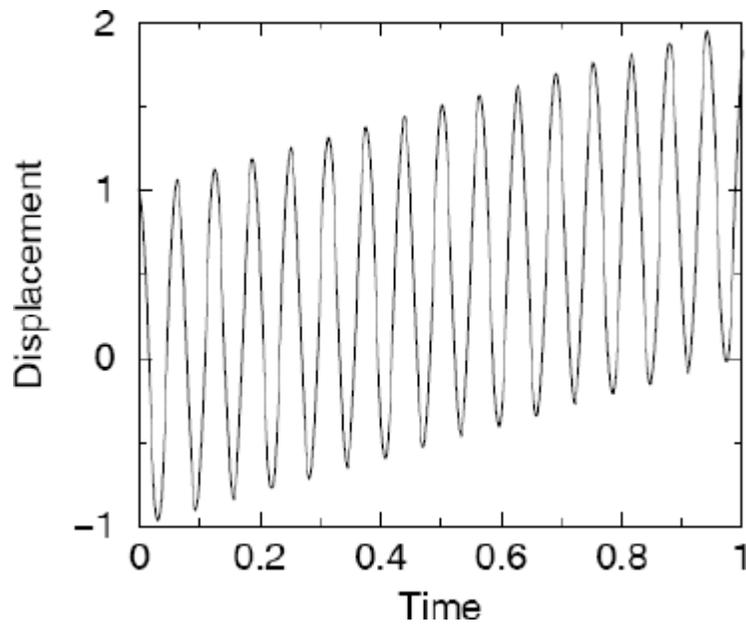
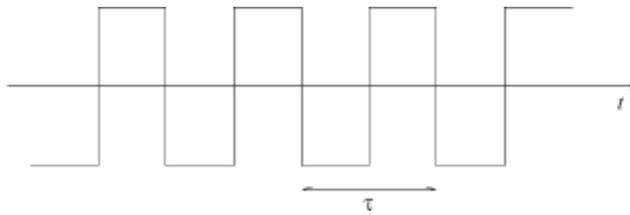


Fig. 6: A signal with a time varying mean is an **aperiodic** signal





For a simple pendulum as shown, if we define the period τ by , then for the pendulum, and such signals are defined as periodic.

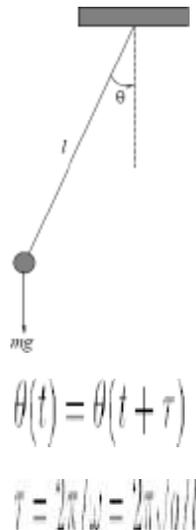


Fig. 7: Pendulum

A periodic signal is one that repeats itself in time and is a reasonable model for many real processes, especially those associated with constant speed machinery.

Stationary signals are those whose average properties do not change with time. **Stationary signals** have constant parameters to describe their behaviour.

Nonstationary signals have time dependent parameters. In an engine excited vibration where the engines speed varies with time; the fundamental period changes with time as well as with the corresponding dynamic loads that cause vibration.

1.5 Deterministic Vs Random Signal:

The signals can be further classified as **monofrequency** (sinusoidal) signals and **multifrequency** signals such as the square wave which has a functional form made up of an infinite superposition of different sine waves with periods $\tau, \tau/2, \tau/3, \dots$

1 D signals are a function of a single independent variable. The speech signal is an example of a 1 D signal where the independent variable is time.

2D signals are a function of two independent variables. An image signal such as a photograph is an example of a 2D signal where the two independent variables are the two spatial variables

1.6 CONTINUOUS VERSUS DISCRETE SIGNALS

The value of a signal at a specific value of the independent variable is called its **amplitude**.

- The variation of the amplitude as a function of the independent variable is called its **waveform**.
- For a 1 D signal, the independent variable is usually labelled as time. If the independent variable is continuous, the signal is called a **continuous-time signal**. A continuous time signal is defined at every instant of time.
- If the independent variable is discrete, the signal is called a **discrete-time signal**. A discrete time signal takes certain numerical values at specified discrete instants of time, and between these specified instants of time, the signal is not defined. Hence, a discrete time signal is basically a sequence of numbers.

1.7 ANALOG VERSUS DIGITAL SIGNALS

A continuous-time signal with a continuous amplitude is usually called an **analog signal**. A speech signal is an example of an analog signal.

A discrete time signal with discrete valued amplitudes represented by a finite number of digits is referred to as a **digital signal**

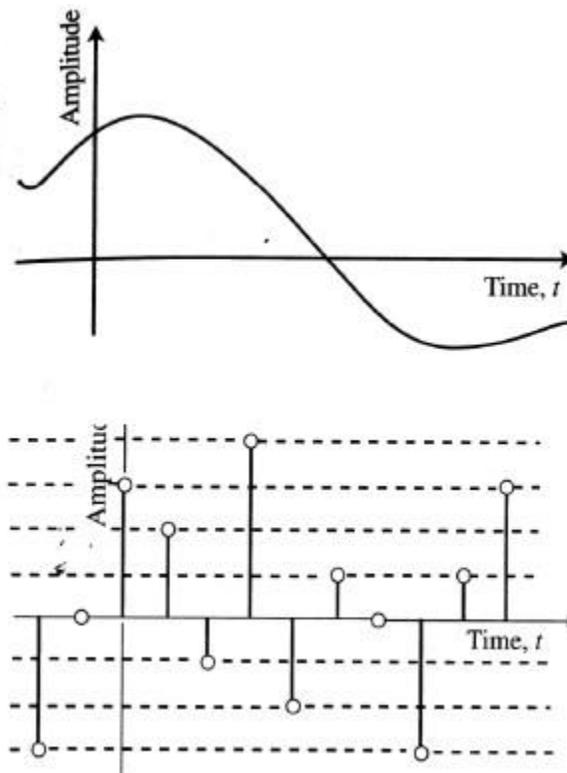


Fig. 8: ADC Conversion

1.8 CONVOLUTIONS

The convolution of f and g is written $f * g$, using an [asterisk](#) or star. It is defined as the integral of the product of the two functions after one is reversed and shifted. As such, it is a particular kind

of [integral transform](#):

$$\begin{aligned}(f * g)(t) &\stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau \\ &= \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau.\end{aligned}$$

While the symbol t is used above, it need not represent the time domain. But in that context, the convolution formula can be described as a weighted average of the function $f(\tau)$ at the moment t where the weighting is given by $g(-\tau)$ simply shifted by amount t . As t changes, the weighting function emphasizes different parts of the input function.

For functions f, g [supported](#) on only $[0, \infty)$ (i.e., zero for negative arguments), the integration limits can be truncated, resulting in

$$(f * g)(t) = \int_0^t f(\tau) g(t - \tau) d\tau \quad \text{for } f, g : [0, \infty) \rightarrow \mathbb{R}$$

In this case, the [Laplace transform](#) is more appropriate than the [Fourier transform](#) below and boundary terms become relevant.

1.8.1 Circular convolution

When a function gT is periodic, with period T , then for functions, f , such that $f * gT$

$$(f * g_T)(t) \equiv \int_{t_0}^{t_0+T} \left[\sum_{k=-\infty}^{\infty} f(\tau + kT) \right] g_T(t - \tau) d\tau,$$

exists, the convolution is also periodic and identical to:

where t_0 is an arbitrary choice. The summation is called a [periodic summation](#) of the function f .

When gT is a [periodic summation](#) of another function, g , then $f * gT$ is known as a *circular* or *cyclic* convolution of f and g .

And if the periodic summation above is replaced by fT , the operation is called a *periodic* convolution of fT and gT

1.9 SAMPLING AND QUANTIZATION

Nearly all data acquisition systems sample data with uniform time intervals. For evenly sampled data, time can be expressed as:

$$T = (N - 1)\Delta t$$

where N is the sampling index which is the number of equally spaced samples. For most Fourier analyzers N is restricted to a power of 2.

- The sample rate or the sampling frequency is:

$$f_s = 1 / (N - 1)\Delta t$$

Sampling frequency is the reciprocal of the time elapsed Δt from one sample to the next.

- The unit of the sampling frequency is cycles per second or Hertz (Hz), if the sampling period is in seconds.

- The sampling theorem asserts that the uniformly spaced discrete samples are a complete representation of the signal if the bandwidth f_{max} is less than half the sampling

rate. The sufficient condition for exact reconstructability from samples at a uniform sampling rate f_s (in samples per unit time) ($f_s \geq 2f_{max}$).

1.9.1 Aliasing

One problem encountered in A/D conversion is that a high frequency signal can be falsely confused as a low frequency signal when sufficient precautions have been avoided.

- This happens when the sample rate is not fast enough for the signal and one speaks of **aliasing**.
- Unfortunately, this problem can not always be resolved by just sampling faster, the signal's frequency content must also be limited.

- Furthermore, the costs involved with postprocessing and data analysis increase with the quantity of data obtained. Data acquisition systems have finite memory, speed and data storage capabilities. Highly oversampling a signal can necessitate shorter sample lengths, longer time on test, more storage medium and increased database management and archiving requirements. The central concept to avoid aliasing is that the sample rate must be at least twice the highest frequency component of the signal ($f_s \geq 2f_{max}$).

We define the Nyquist or cut-off frequency

- The concept behind the cut-off frequency is often referred to as $2\Delta t$

Shannon's sampling criterion. Signal components with frequency content above the cut-off frequency are aliased and can not be distinguished from the frequency components below the cut-off frequency. Conversion of analog frequency into digital frequency during sampling is shown in the figure. Continuous signals with a frequency less **than** one-half of the sampling rate are directly converted into the corresponding digital frequency. Above one-half of the sampling rate, aliasing takes place, resulting in the frequency being misrepresented in the digital data. Aliasing always changes a higher frequency into a lower frequency between 0 and 0.5. In addition, aliasing may also change the phase of the signal by 180

degrees.

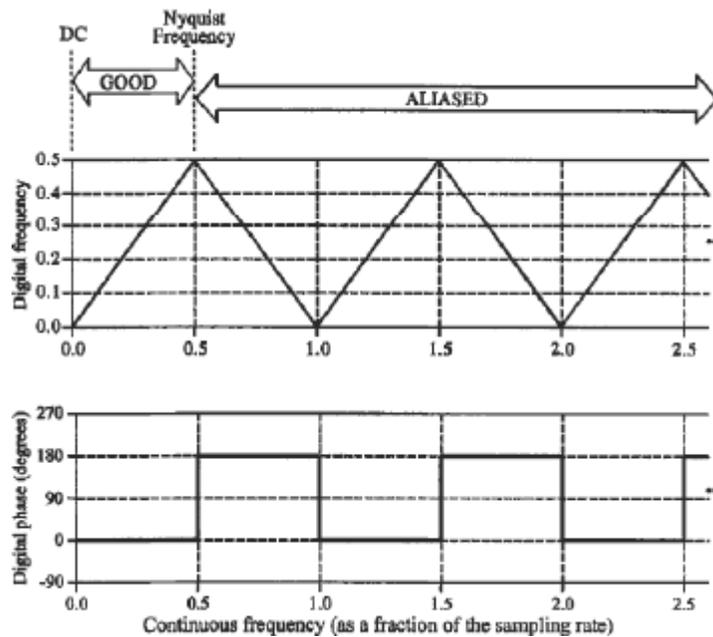
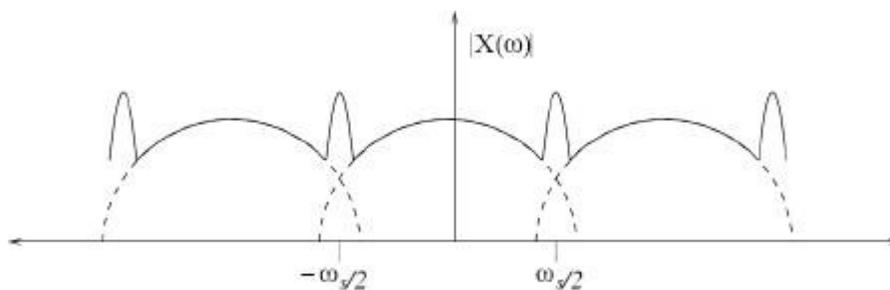


Fig. 9: Aliasing

If any energy in the original signal extends beyond the Nyquist frequency, it is folded back into the Nyquist interval in the spectrum of the sampled signal. This folding is called aliasing.



Spectrum of sampled signal $x_s(t)$ with overlap - aliasing.

1.9.2 Quantization

Quantization is involved to some degree in nearly all digital signal processing, as the process of representing a signal in digital form ordinarily involves rounding. Quantization also forms the core of essentially all [lossy compression](#) algorithms. The difference between an input value and its quantized value (such as [round-off error](#)) is referred to as **quantization error**. A device or [algorithmic function](#) that performs quantization is called a **quantizer**. An [analog-to-digital converter](#) is an example of a quantizer.

Because quantization is a many-to-few mapping, it is an inherently [non-linear](#) and irreversible

process (i.e., because the same output value is shared by multiple input values, it is impossible in general to recover the exact input value when given only the output value).

The set of possible input values may be infinitely large, and may possibly be continuous and therefore **uncountable** (such as the set of all **real numbers**, or all real numbers within some limited range). The set of possible output values may be **finite** or **countably infinite**. The input and output sets involved in quantization can be defined in a rather general way. For example, **vector quantization** is the application of quantization to multi-dimensional (vector-valued) input data

1.10 CONCEPTS OF SIGNAL PROCESSING

In the case of **analog signals**, most signal processing operations are usually carried out in the **time domain**.

- In the case of **discrete time signals**, both **time domain** and **frequency domain** applications are employed.
- In either case, the desired operations are implemented by a combination of some **elementary operations** such as:
 - Simple time domain operations
 - Filtering
 - Amplitude modulation

The three most basic time-domain signal operations are:

- **Scaling**
- **Delay**
- **Addition**

Scaling is simply the multiplication of a signal by a positive or a negative constant. In the case of analog signals, this operation is usually called **amplification** if the magnitude of the multiplying constant, called **gain**, is greater than one. If the magnitude of the multiplying constant is less than one, the operation is called **attenuation**. Thus, if $x(t)$ is an analog signal, the scaling operation generates a signal $y(t)=\alpha x(t)$, where α is the multiplying constant.

Delay operation generates a signal that is delayed replica of the original signal. For an analog signal $x(t)$, $y(t)=x(t-t_0)$ is the signal obtained by delaying $x(t)$ by the amount t_0 , which is assumed to be a positive number. If t_0 is negative, then it is an **advance** operation

Addition operation generates a new signal by the addition of signals. For instance, $y(t)=x_1(t)+x_2(t)-x_3(t)$ is the signal generated by the addition of the three analog signals $x_1(t)$, $x_2(t)$ and $x_3(t)$.

1.11 TYPICAL APPLICATIONS

The main applications of DSP are

audio signal processing,

sometimes referred to as audio processing, is the intentional alteration of **auditory signals**, or **sound**, often through an audio effect **effects unit**. As audio signals may be electronically

represented in either **digital** or **analog** format, **signal processing** may occur in either domain. Analog processors operate directly on the electrical signal, while digital processors operate mathematically on the digital representation of that signal.

audio compression

bit-rate reduction involves **encoding information** using fewer **bits** than the original representation.^[2] Compression can be either **lossy** or **lossless**. **Lossless compression** reduces bits by identifying and eliminating **statistical redundancy**. No information is lost in lossless compression. **Lossy compression** reduces bits by identifying unnecessary information and removing it.^[3] The process of reducing the size of a data file is referred to as data compression. In the context of data transmission, it is called source coding (encoding done at the source of the data before it is stored or transmitted) in opposition to channel coding.^[4]

digital image processing,

is the use of computer **algorithms** to perform **image processing** on **digital images**. As a subcategory or field of **digital signal processing**, digital image processing has many advantages over **analog image processing**. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be model in the form of **multidimensional systems**

speech processing

s the study of **speech signals** and the processing methods of these signals. The signals are usually processed in a **digital** representation, so speech processing can be regarded as a special case of **digital signal processing**, applied to **speech signal**. Aspects of speech processing includes the acquisition, manipulation, storage, transfer and output of **speech signals**.

speech recognition,

is the **inter-disciplinary** sub-field of **computational linguistics** which incorporates knowledge and research in the **linguistics**, computer science, and electrical engineering fields to develop methodologies and technologies that enables the recognition and **translation** of spoken language into text by computers and computerized devices such as those categorized as **Smart Technologies** and **robotics**. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT).

digital communications, radar, sonar, financial signal processing seismology and biomedicine. Specific examples are speech compression and transmission in digitalmobile phones, room correction of sound in hi-fi and sound reinforcement applications, weather forecasting,economic forecasting, seismic data processing, analysis and control of industrial processes, medical imagingsuch as CAT scans and MRI, MP3 compression, computer graphics, image

manipulation, hi- fi loudspeakerscrossovers and equalization, and audio effects for use with electric guitar amplifiers

1.12 ADVANTAGES OF DIGITAL SIGNAL PROCESSING COMPARED WITH ANALOG SIGNAL PROCESSING

Accuracy

Implementation of sophisticated algorithms

Storage

Noise reduction

1.13 APPLICATIONS OF SIGNAL PROCESSING IN BIOMEDICAL ENGINEERING

- audio signal processing – for electrical signals representing sound, such as speech or music
- Speech signal processing – for processing and interpreting spoken words
- Image processing – in digital cameras, computers and various imaging systems
- Video processing – for interpreting moving pictures
- Wireless communication - waveform generations, demodulation, filtering, equalization
- Control systems
- Array processing – for processing signals from arrays of sensors
- Seismology
- Financial signal processing – analyzing financial data using signal processing techniques, especially for prediction purposes.
- Feature extraction, such as image understanding and speech recognition.
- Quality improvement, such as noise reduction, image enhancement, and echo cancellation.
- (Source coding), including audio compression, image compression, and video compression



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

**SCHOOL OF BIO AND CHEMICAL ENGINEERING
DEPARTMENT OF BIOMEDICAL ENGINEERING**

UNIT – II – Digital Signal Processing & Its Applications – SEC1324

FREQUENCY ANALYSIS OF THE SIGNALS

1.1 Discrete Fourier Transform

the **discrete Fourier transform (DFT)** converts a finite sequence of equally spaced samples of a function into the list of coefficients of a finite combination of complex sinusoids, ordered by their frequencies, that has those same sample values. It can be said to convert the sampled function from its original domain (often time or position along a line) to the frequency domain.

The input samples are complex numbers (in practice, usually real numbers), and the output coefficients are complex as well. The frequencies of the output sinusoids are integer multiples of a fundamental frequency, whose corresponding period is the length of the sampling interval. The combination of sinusoids obtained through the DFT is therefore periodic with that same period. The DFT differs from the discrete-time Fourier transform (DTFT) in that its input **and** output sequences are both finite; it is therefore said to be the Fourier analysis of finite-domain (or periodic) discrete-time functions

The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications.^[1] In digital signal processing, the function is any quantity or signal that varies over time, such as the pressure of a sound wave, a radio signal, or daily temperature readings, sampled over a finite time interval (often defined by a window function^[2]). In image processing, the samples can be the values of pixels along a row or column of a raster image. The DFT is also used to efficiently solve partial differential equations, and to perform other operations such as convolutions or multiplying large integers.

1.2 Computation of DFT

he sequence of N complex numbers x_0, x_1, \dots, x_{N-1} is transformed into an N -periodic sequence of complex numbers:

$$X_k \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i k n / N}, \quad k \in \mathbb{Z}$$
$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{2\pi i k n / N}, \quad n \in \mathbb{Z}$$

which is also N -periodic. In the domain $n \in [0, N-1]$, this is the **inverse transform** In this interpretation, each X_k is a complex number that encodes both amplitude and phase of a sinusoidal component $(e^{2\pi i k n / N})$ of function x_n . The sinusoid's frequency is k cycles per N samples. Its amplitude and phase are:

$$|X_k|/N = \sqrt{\text{Re}(X_k)^2 + \text{Im}(X_k)^2}/N$$

$$\arg(X_k) = \text{atan2}(\text{Im}(X_k), \text{Re}(X_k)) = -i \ln \left(\frac{X_k}{|X_k|} \right),$$

1.3 Fourier Transform:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot \left(\cos\left(-2\pi k \frac{n}{N}\right) + i \sin\left(-2\pi k \frac{n}{N}\right) \right), \quad k \in \mathbb{Z}$$

1.4 Inverse Fourier Transform:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot \left(\cos\left(2\pi k \frac{n}{N}\right) + i \sin\left(2\pi k \frac{n}{N}\right) \right), \quad n \in \mathbb{Z}$$

N = number of time samples we have

n = current sample we're considering (0, ..., $N-1$)

x_n = value of the signal at time n

k = current frequency we're considering (0 Hertz up to $N-1$ Hertz)

X_k = amount of frequency k in the signal (Amplitude and Phase, a complex number)

1.5 PROPERTIES

The periodicity can be shown directly from the definition:

$$X_{k+N} \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}(k+N)n} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \underbrace{e^{-2\pi in}}_1 = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} = X_k.$$

Similarly, it can be shown that the IDFT formula leads to a periodic extension.

1.5.1 Shift theorem

Multiplying x_n by a *linear phase* $e^{\frac{2\pi i}{N}nm}$ for some integer m corresponds to a *circular shift* of the output X_k : X_k is replaced by X_{k-m} , where the subscript is interpreted modulo N (i.e., periodically). Similarly, a circular shift of the input x_n corresponds to multiplying the output X_k by a linear phase. Mathematically, if $\{x_n\}$ represents the vector \mathbf{x} then

$$\text{if } \mathcal{F}(\{x_n\})_k = X_k$$

$$\text{then } \mathcal{F}(\{x_n \cdot e^{\frac{2\pi i}{N}nm}\})_k = X_{k-m}$$

$$\text{and } \mathcal{F}(\{x_{n-m}\})_k = X_k \cdot e^{-\frac{2\pi i}{N}km}$$

he convolution theorem for the discrete-time Fourier transform indicates that a convolution of two infinite sequences can be obtained as the inverse transform of the product of the individual transforms. An important simplification occurs when the sequences are of finite length, N . In terms of the DFT and inverse DFT, it can be written as follows:

$$\mathcal{F}^{-1} \{ \mathbf{X} \cdot \mathbf{Y} \}_n = \sum_{l=0}^{N-1} x_l \cdot (y_N)_{n-l} \stackrel{\text{def}}{=} (\mathbf{X} * \mathbf{Y}_N)_n ,$$

which is the convolution of the \mathbf{X} sequence with a \mathbf{Y} sequence extended by periodic summation:

$$(\mathbf{Y}_N)_n \stackrel{\text{def}}{=} \sum_{p=-\infty}^{\infty} y_{(n-pN)} = y_{n(\text{mod}N)} .$$

Similarly, the cross-correlation of \mathbf{X} and \mathbf{Y}_N is given by:

$$\mathcal{F}^{-1} \{ \mathbf{X}^* \cdot \mathbf{Y} \}_n = \sum_{l=0}^{N-1} x_l^* \cdot (y_N)_{n+l} \stackrel{\text{def}}{=} (\mathbf{X} \star \mathbf{Y}_N)_n .$$

1.6 Radix 2 FFT algorithms

An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the most important difference is that an FFT is much faster. (In the presence of round-off error, many FFT algorithms are also much more accurate than evaluating the DFT definition directly, as discussed below.)

Let x_0, \dots, x_{N-1} be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \quad k = 0, \dots, N-1 .$$

Evaluating this definition directly requires $O(N^2)$ operations: there are N outputs X_k , and each output requires a sum of N terms. An FFT is any method to compute the same results in $O(N \log N)$ operations. More precisely, all known FFT algorithms require $\Theta(N \log N)$ operations (technically, O only denotes an upper bound), although there is no known proof that a lower complexity score is impossible. (Johnson and Frigo, 2007)

To illustrate the savings of an FFT, consider the count of complex multiplications and additions. Evaluating the DFT's sums directly involves N^2 complex multiplications and $N(N-1)$ complex additions [of which $O(N)$ operations can be saved by eliminating trivial operations such as

multiplications by 1]. The well-known radix-2 Cooley–Tukey algorithm, for N a power of 2, can compute the same result with only $(N/2)\log_2(N)$ complex multiplications (again, ignoring simplifications of multiplications by 1 and similar) and $N\log_2(N)$ complex additions. In practice, actual performance on modern computers is usually dominated by factors other than the speed of arithmetic operations and the analysis is a complicated subject (see, e.g., Frigo & Johnson, 2005), but the overall improvement from $O(N^2)$ to $O(N \log N)$ remains.

1.6.1 DIF & DIT algorithms

In the context of fast Fourier transform algorithms, a **butterfly** is a portion of the computation that combines the results of smaller discrete Fourier transforms (DFTs) into a larger DFT, or vice versa (breaking a larger DFT up into subtransforms). The name "butterfly" comes from the shape of the data-flow diagram in the radix-2 case, as described below.^[1] The earliest occurrence in print of the term is thought to be in a 1969 MIT technical report.^{[2][3]} The same structure can also be found in the Viterbi algorithm, used for finding the most likely sequence of hidden states.

Most commonly, the term "butterfly" appears in the context of the Cooley–Tukey FFT algorithm, which recursively breaks down a DFT of composite size $n = rm$ into r smaller transforms of size m where r is the "radix" of the transform. These smaller DFTs are then combined via size- r butterflies, which themselves are DFTs of size r (performed m times on corresponding outputs of the sub-transforms) pre-multiplied by roots of unity (known as twiddle factors). (This is the "decimation in time" case; one can also perform the steps in reverse, known as "decimation in frequency", where the butterflies come first and are post-multiplied by twiddle factors. See also the Cooley–Tukey FFT article.)

In the case of the radix-2 Cooley–Tukey algorithm, the butterfly is simply a DFT of size-2 that takes two inputs (x_0, x_1) (corresponding outputs of the two sub-transforms) and gives two outputs (y_0, y_1) by the formula (not including twiddle factors):

$$y_0 = x_0 + x_1$$

$$y_1 = x_0 - x_1.$$

If one draws the data-flow diagram for this pair of operations, the (x_0, x_1) to (y_0, y_1) lines cross and resemble the wings of a butterfly, hence the name (see also the illustration at right).

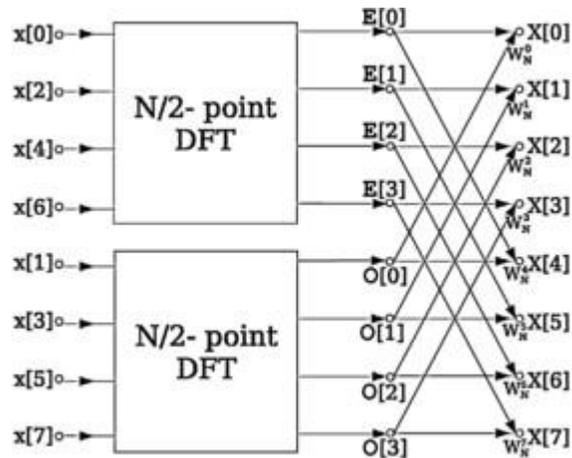


Fig. 1: Block schematic of DFT computation

A decimation-in-time radix-2 FFT breaks a length- N DFT into two length- $N/2$ DFTs followed by a combining stage consisting of many butterfly operations.

More specifically, a radix-2 decimation-in-time FFT algorithm on $n = 2^p$ inputs with respect to a primitive n -th root of unity $\omega_n^k = e^{-\frac{2\pi i k}{n}}$ relies on $O(n \log n)$ butterflies of the form:

$$\begin{aligned} y_0 &= x_0 + x_1 \omega_n^k \\ y_1 &= x_0 - x_1 \omega_n^k, \end{aligned}$$

where k is an integer depending on the part of the transform being computed. Whereas the corresponding inverse transform can mathematically be performed by replacing ω with ω^{-1} (and possibly multiplying by an overall scale factor, depending on the normalization convention), one may also directly invert the butterflies:

$$\begin{aligned} x_0 &= \frac{1}{2}(y_0 + y_1) \\ x_1 &= \frac{\omega_n^{-k}}{2}(y_0 - y_1), \end{aligned}$$

corresponding to a decimation-in-frequency FFT algorithm.



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

**SCHOOL OF BIO AND CHEMICAL ENGINEERING
DEPARTMENT OF BIOMEDICAL ENGINEERING**

UNIT – III – Digital Signal Processing & Its Applications – SEC1324

FIR FILTER DESIGNING

3.1 DIGITAL FILTERS

In signal processing, a **digital filter** is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. This is in contrast to the other major type of electronic filter, the analog filter, which is an electronic circuit operating on continuous-time analog signals.

A digital filter system usually consists of an analog-to-digital converter to sample the input signal, followed by a microprocessor and some peripheral components such as memory to store data and filter coefficients etc. Finally a digital-to-analog converter to complete the output stage. Program Instructions (software) running on the microprocessor implement the digital filter by performing the necessary mathematical operations on the numbers received from the ADC. In some high performance applications, an FPGA or ASIC is used instead of a general purpose microprocessor, or a specialized DSP with specific paralleled architecture for expediting operations such as filtering.

Digital filters may be more expensive than an equivalent analog filter due to their increased complexity, but they make practical many designs that are impractical or impossible as analog filters. When used in the context of real-time analog systems, digital filters sometimes have problematic latency (the difference in time between the input and the response) due to the associated analog-to-digital and digital-to-analog conversions and anti-aliasing filters, or due to other delays in their implementation.

A digital filter is characterized by its transfer function, or equivalently, its difference equation. Mathematical analysis of the transfer function can describe how it will respond to any input. As such, designing a filter consists of developing specifications appropriate to the problem (for example, a second-order low pass filter with a specific cut-off frequency), and then producing a transfer function which meets the specifications.

The transfer function for a linear, time-invariant, digital filter can be expressed as a transfer function in the Z-domain; if it is causal, then it has the form:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}$$

where the order of the filter is the greater of N or M . See Z-transform's LCCD equation for further discussion of this transfer function.

This is the form for a recursive filter with both the inputs (Numerator) and outputs (Denominator), which typically leads to an IIR infinite impulse response behaviour, but if the

denominator is made equal to unity i.e. no feedback, then this becomes an FIR or finite impulse response filter.

3.2 ANALYSIS TECHNIQUES

A variety of mathematical techniques may be employed to analyze the behaviour of a given digital filter. Many of these analysis techniques may also be employed in designs, and often form the basis of a filter specification.

Typically, one characterizes filters by calculating how they will respond to a simple input such as an impulse. One can then extend this information to compute the filter's response to more complex signals.

3.3 IMPULSE RESPONSE

The impulse response, often denoted $h[k]$ or h_k , is a measurement of how a filter will respond to the Kronecker delta function. For example, given a difference equation, one would set $x_0 = 1$ and $x_k = 0$ for $k \neq 0$ and evaluate. The impulse response is a characterization of the filter's behaviour. Digital filters are typically considered in two categories: infinite impulse response (IIR) and finite impulse response (FIR). In the case of linear time-invariant FIR filters, the impulse response is exactly equal to the sequence of filter coefficients:

$$y_n = \sum_{k=0}^N h_k x_{n-k}$$

IIR filters on the other hand are recursive, with the output depending on both current and previous inputs as well as previous outputs. The general form of an IIR filter is thus:

$$\sum_{m=0}^M a_m y_{n-m} = \sum_{k=0}^N b_k x_{n-k}$$

Plotting the impulse response will reveal how a filter will respond to a sudden, momentary disturbance.

3.4 DIFFERENCE EQUATION

In discrete-time systems, the digital filter is often implemented by converting the transfer function to a linear constant-coefficient difference equation (LCCD) via the Z-transform. The discrete frequency-domain transfer function is written as the ratio of two polynomials. For example:

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

This is expanded:

$$H(z) = \frac{z^2 + 2z + 1}{z^2 + \frac{1}{4}z - \frac{3}{8}}$$

and to make the corresponding filter causal, the numerator and denominator are divided by the highest order of z

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2}} = \frac{Y(z)}{X(z)}$$

The coefficients of the denominator, a_k are the 'feed-backward' coefficients and the coefficients of the numerator are the 'feed-forward' coefficients, b_k . The resultant linear difference equation is:

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

or, for the example above:

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

rearranging terms:

$$\Rightarrow (1 + \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2})Y(z) = (1 + 2z^{-1} + z^{-2})X(z)$$

then by taking the inverse z -transform:

$$\Rightarrow y[n] + \frac{1}{4}y[n - 1] - \frac{3}{8}y[n - 2] = x[n] + 2x[n - 1] + x[n - 2]$$

and finally, by solving for $y[n]$:

$$y[n] = -\frac{1}{4}y[n - 1] + \frac{3}{8}y[n - 2] + x[n] + 2x[n - 1] + x[n - 2]$$

This equation shows how to compute the next output sample, $y[n]$, in terms of the past outputs, $y[n - p]$, the present input, $x[n]$, and the past inputs, $x[n - p]$. Applying the filter to an input

in this form is equivalent to a Direct Form I or II realization, depending on the exact order of evaluation.

3.5 FIR FILTERS & ITS DESIGNING

In signal processing, a **finite impulse response (FIR)** filter is a filter whose impulse response (or response to any finite length input) is of *finite* duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying).

The impulse response (that is, the output in response to a Kronecker delta input) of an N th-order discrete-time FIR filter lasts exactly $N + 1$ samples (from first nonzero element through last nonzero element) before it then settles to zero.

FIR filters can be discrete-time or continuous-time, and digital or analog.

For a causal discrete-time FIR filter of order N , each value of the output sequence is a weighted sum of the most recent input values:

$$\begin{aligned} y[n] &= b_0x[n] + b_1x[n - 1] + \cdots + b_Nx[n - N] \\ &= \sum_{i=0}^N b_i \cdot x[n - i], \end{aligned}$$

where:

- $x[n]$ is the input signal,
- $y[n]$ is the output signal,
- N is the filter order; an N th-order filter has $(N + 1)$ terms on the right-hand side
- b_i is the value of the impulse response at the i 'th instant for $0 \leq i \leq N$ of an N th-order FIR filter. If the filter is a direct form FIR filter then b_i is also a coefficient of the filter .

This computation is also known as discrete convolution.

The $x[n-i]$ in these terms are commonly referred to as *taps*, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a *5th order/6-tap filter*, for instance.

The impulse response of the filter as defined is nonzero over a finite duration. Including zeros, the impulse response is the infinite sequence:

$$h[n] = \sum_{i=0}^N b_i \cdot \delta[n - i] = \begin{cases} b_n & 0 \leq n \leq N \\ 0 & \text{otherwise.} \end{cases}$$

If an FIR filter is non-causal, the range of nonzero values in its impulse response can start before $n = 0$, with the defining formula appropriately generalized.

An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter. FIR filters:

Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.

Are inherently stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than $\sum |b_i|$ times the largest value appearing in the input.

Can easily be designed to be linear phase by making the coefficient sequence symmetric. This property is sometimes desired for phase-sensitive applications, for example data communications, seismology, crossover filters, and mastering.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed. However many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

An FIR filter is designed by finding the coefficients and filter order that meet certain specifications, which can be in the time-domain (e.g. a matched filter) and/or the frequency domain (most common). Matched filters perform a cross-correlation between the input signal and a known pulse-shape. The FIR convolution is a cross-correlation between the input signal and a time-reversed copy of the impulse-response. Therefore, the matched-filter's impulse response is "designed" by sampling the known pulse-shape and using those samples in reverse order as the coefficients of the filter.^[1]

When a particular frequency response is desired, several different design methods are common:

1. Window design method
2. Frequency Sampling method
3. Weighted least squares design
4. Parks-McClellan method (also known as the Equiripple, Optimal, or Minimax method).
The Remez exchange algorithm is commonly used to find an optimal equiripple set of coefficients. Here the user specifies a desired frequency response, a weighting function for errors from this response, and a filter order N . The algorithm then finds the set of $(N + 1)$ coefficients that minimize the maximum deviation from the ideal. Intuitively, this finds the filter that is as close as you can get to the desired response given that you can use only $(N + 1)$ coefficients. This method is particularly easy in practice since at least one text^[2] includes a program that takes the desired filter and N , and returns the optimum coefficients.

5. Equiripple FIR filters can be designed using the FFT algorithms as well.^[3] The algorithm is iterative in nature. You simply compute the DFT of an initial filter design that you have using the FFT algorithm (if you don't have an initial estimate you can start with $h[n]=\delta[n]$). In the Fourier domain or FFT domain you correct the frequency response according to your desired specs and compute the inverse FFT. In time-domain you retain only N of the coefficients (force the other coefficients to zero). Compute the FFT once again. Correct the frequency response according to specs.

Software packages like MATLAB, GNU Octave, Scilab, and SciPy provide convenient ways to apply these different methods.

In the window design method, one first designs an ideal IIR filter and then truncates the infinite impulse response by multiplying it with a finite length window function. The result is a finite impulse response filter whose frequency response is modified from that of the IIR filter. Multiplying the infinite impulse by the window function in the time domain results in the frequency response of the IIR being convolved with the Fourier transform (or DTFT) of the window function. If the window's main lobe is narrow, the composite frequency response remains close to that of the ideal IIR filter.

The ideal response is usually rectangular, and the corresponding IIR is a sinc function. The result of the frequency domain convolution is that the edges of the rectangle are tapered, and ripples appear in the passband and stopband. Working backward, one can specify the slope (or width) of the tapered region (*transition band*) and the height of the ripples, and thereby derive the frequency domain parameters of an appropriate window function. Continuing backward to an impulse response can be done by iterating a filter design program to find the minimum filter order. Another method is to restrict the solution set to the parametric family of Kaiser windows, which provides closed form relationships between the time-domain and frequency domain parameters. In general, that method will not achieve the minimum possible filter order, but it is particularly convenient for automated applications that require dynamic, on-the-fly, filter design.

The window design method is also advantageous for creating efficient half-band filters, because the corresponding sinc function is zero at every other sample point (except the center one). The product with the window function does not alter the zeros, so almost half of the coefficients of the final impulse response are zero. An appropriate implementation of the FIR calculations can exploit that property to double the filter's efficiency.

A moving average filter is a very simple FIR filter. It is sometimes called a boxcar filter, especially when followed by decimation. The filter coefficients, b_0, \dots, b_N , are found via the following equation:

$$b_i = \frac{1}{N + 1}$$

To provide a more specific example, we select the filter order:

$$N = 2$$

The impulse response of the resulting filter is:

$$h[n] = \frac{1}{3}\delta[n] + \frac{1}{3}\delta[n - 1] + \frac{1}{3}\delta[n - 2]$$

The Fig. (a) on the right shows the block diagram of a 2nd-order moving-average filter discussed below. The transfer function is:

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

Fig. (b) on the right shows the corresponding pole-zero diagram. Zero frequency (DC) corresponds to (1,0), positive frequencies advancing counterclockwise around the circle to the Nyquist frequency at (-1,0). Two poles are located at the origin, and two zeros are located at $z_1 = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$, $z_2 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$.

The frequency response, in terms of normalized frequency ω , is:

$$H(e^{j\omega}) = \frac{1}{3} + \frac{1}{3}e^{-j\omega} + \frac{1}{3}e^{-j2\omega}.$$

Fig. (c) on the right shows the magnitude and phase components of $H(e^{j\omega})$. But plots like these can also be generated by doing a discrete Fourier transform (DFT) of the impulse response.^[note 2] And because of symmetry, filter design or viewing software often displays only the $[0, \pi]$ region. The magnitude plot indicates that the moving-average filter passes low frequencies with a gain near 1 and attenuates high frequencies, and is thus a crude low-pass filter. The phase plot is linear except for discontinuities at the two frequencies where the magnitude goes to zero. The size of the discontinuities is π , representing a sign reversal. They do not affect the property of linear phase. That fact is illustrated in Fig. (d).

The frequency response of an ideal low pass filter is shown in the image below. The frequency axis is normalised with respect to the sampling frequency. The cut-off, or transition frequency (f_t) is always between 0 and 0.5, as 0.5 represents the Nyquist frequency. As you would expect from a low pass filter, all frequencies below f_t are passed, where-as all those above are stopped.

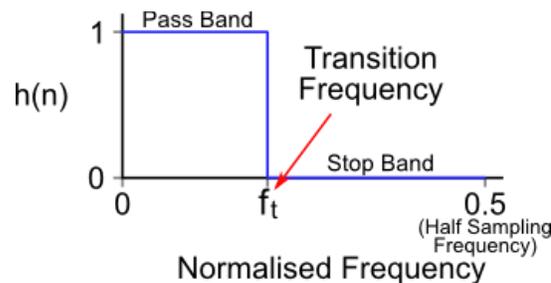


Fig. 1: Transition Frequency

The impulse response of this ideal low pass filter is shown below, it is a sinc function. The equation is shown next to the plot. If we could create a filter with this impulse response we would have an ideal low pass filter like that shown above. A set of Gnuplot commands are also given for recreating this graph.

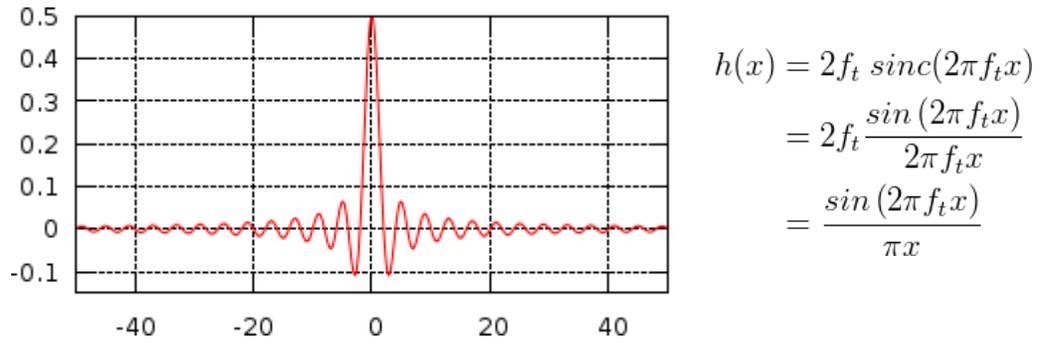


Fig. 2: Impulse response of ideal LPF (ft=0.25)

Below is a plot of impulse responses for different values of f_t .

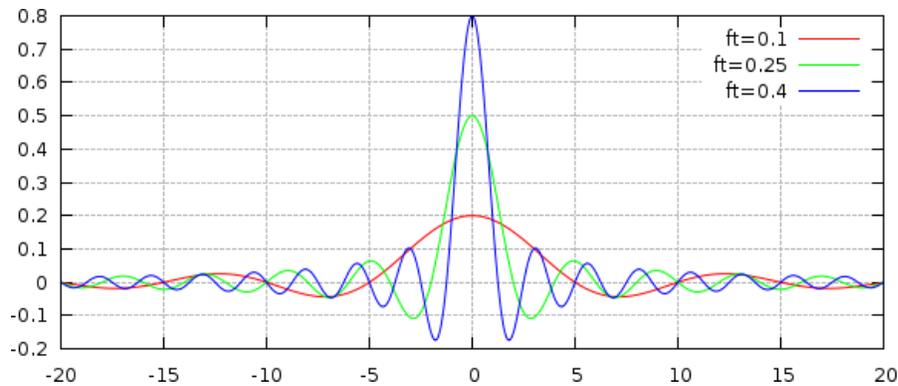


Fig. 3: Sinc function for different values of normalised transition frequency

Unfortunately it is not as easy as that. Given the non-recursive filter structure like that shown below, there are two problems with creating this ideal impulse response.

- First, the sinc function is infinite in the x direction, the ripples keep on going in both directions. However, the FIR filter only allows us to create finite impulse responses, the number of filter taps must be finite.
- Second, the impulse response is non-casual, this means an implementation would require samples from the future.

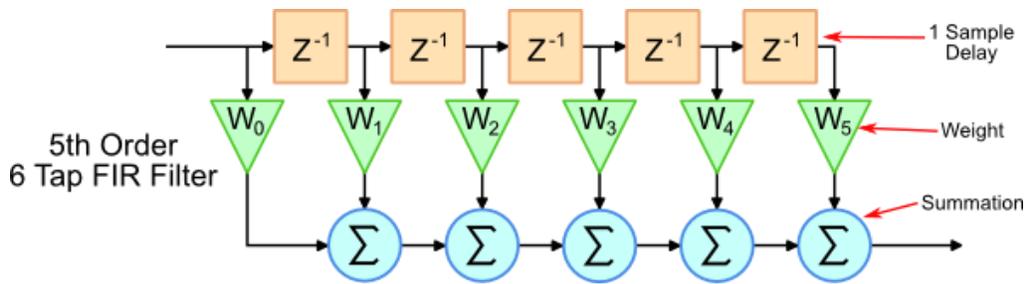


Fig. 4: Non-Recursive, Finite Impulse Response Filter

Luckily, the solutions to these issues are quite simple. First, a window is applied to the sinc function such that only a portion of the impulse response is actually used. Secondly, the impulse response is shifted such that the filter only operates on available samples (those from the past). These techniques are demonstrated in the the following example.

3.6 LOW PASS FILTER EXAMPLE

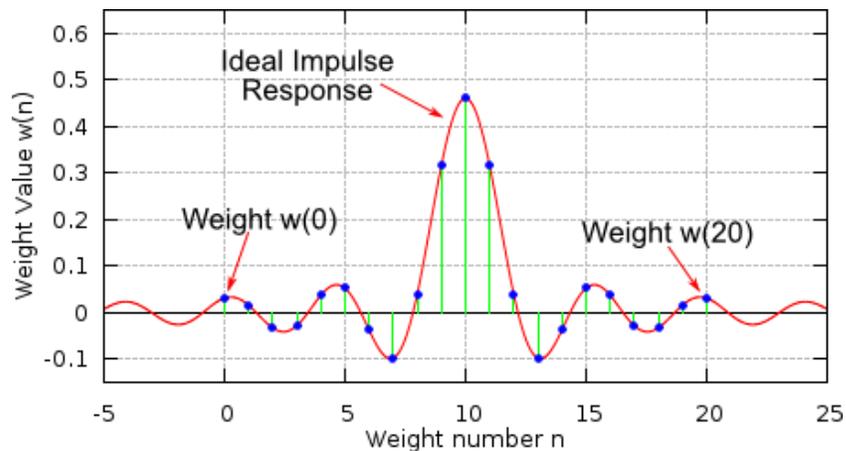


Fig. 4: Filter weights $M=20$, $f_t=0.23$

Consider the filter with the properties given below.

- **Filter Type:** Low Pass
- **Sampling Frequency:** 2000 Hz
- **Cut off Frequency:** 460 Hz
- **Filter Length (# weights):** 21

The plot to the right shows the filter weights that have been calculated using the equations below.

- M - This is the filter order, it is always equal to the number of taps minus 1

- f_t - This is the normalised transition frequency.

There are 21 weights that fit on the ideal impulse response curve. It can be seen how the impulse response is effectively cropped by only using 21 weights.

Note: When calculating weight values with an odd number of weights, a divide by zero will occur at $n=M/2$. Therefore, based on l'Hôpital's rule, the value of $2f_t$ is used.

$$M = \text{filter length} - 1 = 20$$

$$f_t = \frac{\text{Cut off frequency}}{\text{Sampling Frequency}} = \frac{460}{2000} = 0.23$$

$$w_{lpf}(n) = \begin{cases} \frac{\sin[2\pi f_t(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} & n \neq \frac{M}{2} \\ 2f_t & n = \frac{M}{2} \end{cases}$$

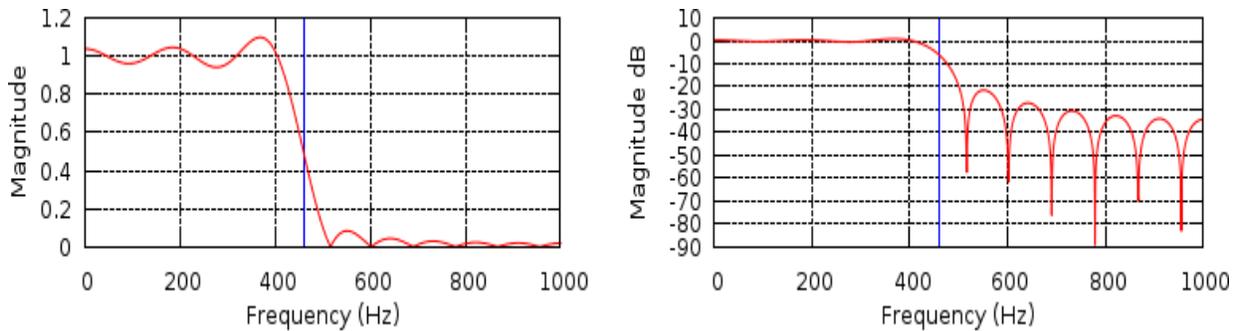


Fig. 5: The large amount of ripple visible on the non-dB plot is due to the rather crude approach of truncating the infinite ideal impulse response. The approach that has just been used is called applying a **Rectangular Window**. The next section describes different window types that can decrease the ripple and improve the attenuation of the stop band.

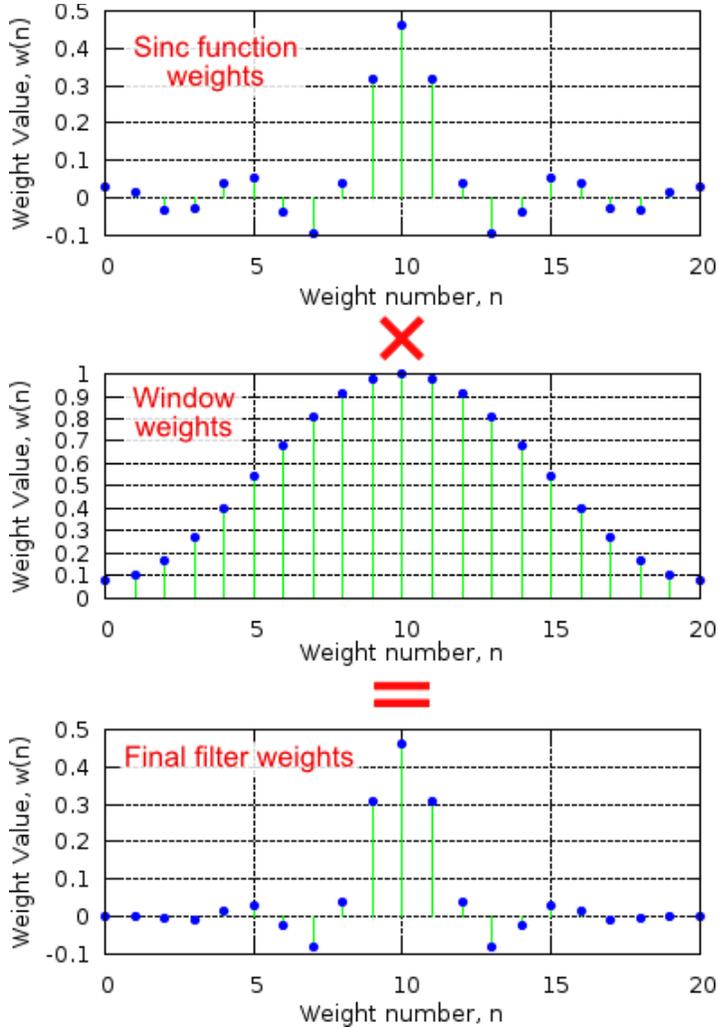


Fig. 6: Windowing

Applying a window to the sinc function weights provides extra control over the characteristics of the filter. The image to the right illustrates the process.

First, the normal sinc weights are calculated as described above. Then the window weights are calculated, in this case a Hamming Window has been used, the equation is below. The two sets of weights are multiplied together to create the final set of filter weights.

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

3.7 Hamming Windowing Equation

Once again, M is the order of the filter, which is equal to the filter length - 1.

The plots below show the effect on the filter's frequency response before applying the Hamming Window (green) and after (red). The trick is to select the window type and filter length that will give a filter with the correct rate of roll-off and level of attenuation in the stop band.

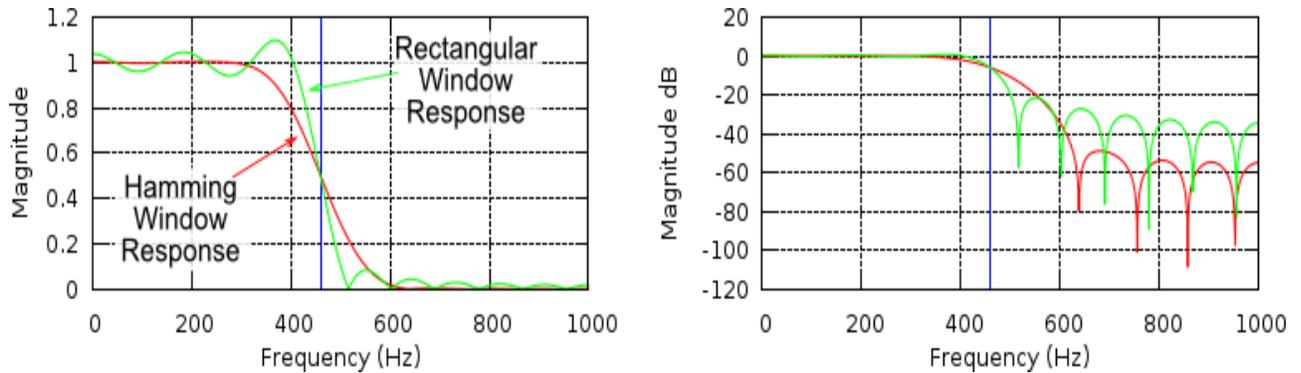


Fig. 7: Different Windows

The table below gives the equations for different window types.

3.8 Window Type Weight Equation

Rectangular $w(n) = 1$

Bartlett $w(n) = 1 - \frac{2|n - \frac{M}{2}|}{M}$

Hanning $w(n) = 0.5 - 0.5 \cos\left(\frac{2\pi n}{M}\right)$

Hamming $w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M}\right)$

Blackman $w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{M}\right) + 0.08 \cos\left(\frac{4\pi n}{M}\right)$

The image below shows the effect of different windows on the frequency response of a 28th Order (29 weights) low pass filter, with a cut-off frequency of 5000Hz and sampling frequency of 44100Hz.

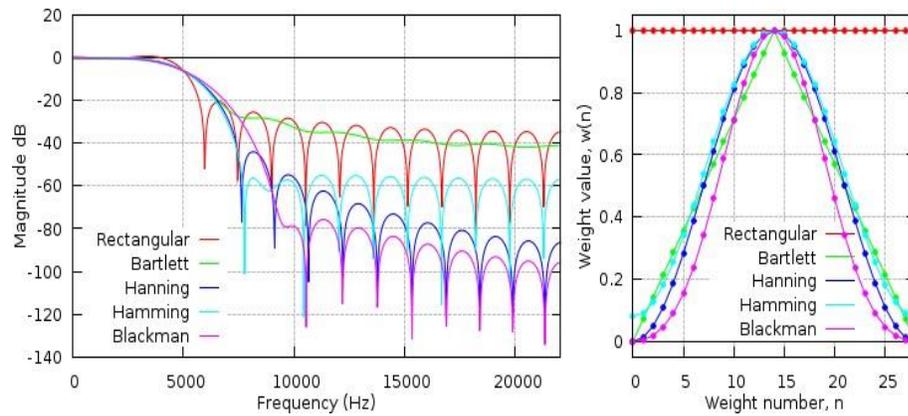


Fig. 8: Frequency Response and Weight Values of different windows types

3.9 HIGH PASS, BAND PASS AND BAND STOP FILTERS

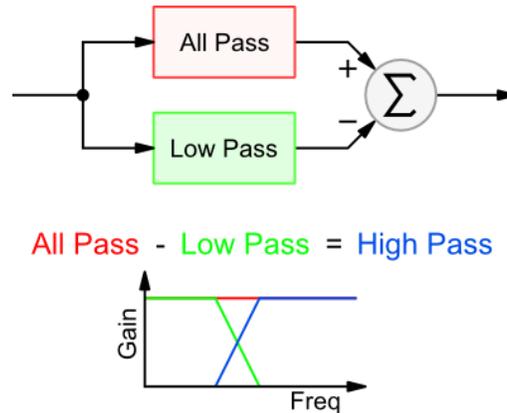


Fig. 9: The high pass filter is made up from a low pass and an all pass filter. The image to the right demonstrates how this works. If you take an all pass filter and subtract the output of the low pass, you are left with a high pass filter.

The all pass filter is of the same order as the low pass filter. All the weight values are 0.0 apart from the centre weight which has a value 1.0. **Note:** This places the constraint that when creating a high pass filter in this way, the order must be even (an odd number of taps).

The equation for calculating the weights (before windowing) is shown below. Comparing this equation with the low pass filter it is easy to see the subtraction and the all pass filter's single 1.0 weight applied in the case of $n=M/2$. Windows are applied in exactly the same way as with the low pass filter.

$$w_{hpf}(n) = \begin{cases} -\frac{\sin[2\pi f_t(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} & n \neq \frac{M}{2} \\ 1 - 2f_t & n = \frac{M}{2} \end{cases}$$

Below is a Low Pass and High Pass filter frequency response with the same transition frequency.

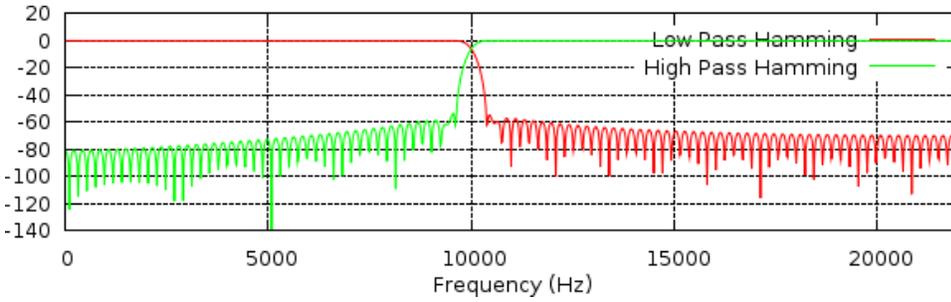
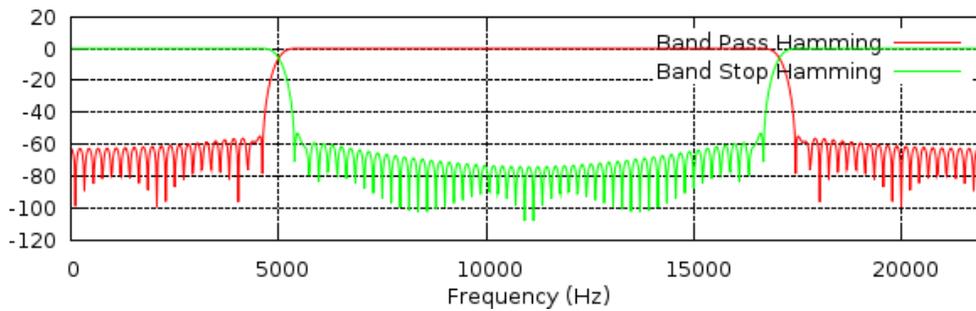


Fig. 10: Low pass and high pass hamming

The band stop and band pass are achieved in a similar way. The equations for calculating the weights are shown below. For both band pass and band stop, the filter order needs to be even (an odd filter length). Once again, windows are applied across the weights as before.

$$w_{bp}(n) = \begin{cases} \frac{\sin[2\pi f_{t2}(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} - \frac{\sin[2\pi f_{t1}(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} & n \neq \frac{M}{2} \\ 2(f_{t2} - f_{t1}) & n = \frac{M}{2} \end{cases}$$

$$w_{bs}(n) = \begin{cases} \frac{\sin[2\pi f_{t1}(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} - \frac{\sin[2\pi f_{t2}(n-\frac{M}{2})]}{\pi(n-\frac{M}{2})} & n \neq \frac{M}{2} \\ 1 - 2(f_{t2} - f_{t1}) & n = \frac{M}{2} \end{cases}$$



3.10 THE KAISER WINDOW

The Kaiser Window is specified differently to the previous windows. Rather than specifying the filter order, the amount of ripple and the transition band width are specified. The image below shows the meaning of these two parameters.

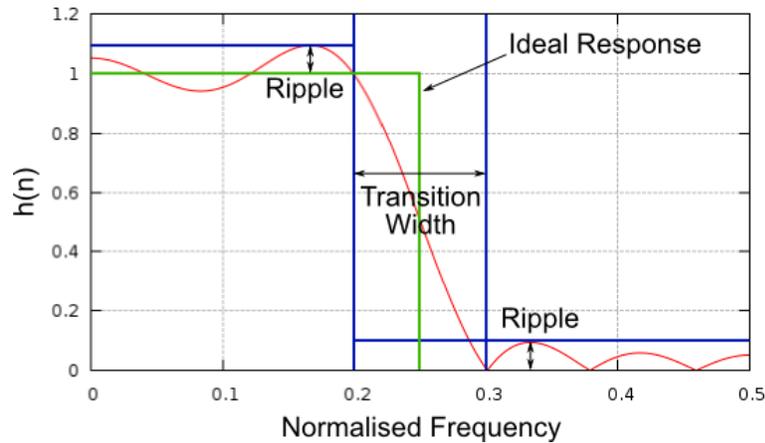


Fig. 11: Once you have decided on the amount of ripple and transition width, using the equations below you can calculate the values for A and tw . These values can be used to calculate the filter order, M and a further parameter, β . In these equations the transition width and sampling frequency are in Hz.

$$A = -20.0 \log_{10}(r) \quad M = \begin{cases} \text{ceil} \left(\frac{A-7.95}{2.285*tw} \right) & A > \\ \text{ceil} \left(\frac{5.79}{tw} \right) & A \leq \end{cases}$$

$$tw = 2\pi \frac{\text{Transition}}{\text{Sampling Freq}} \quad \beta = \begin{cases} 0.0 & \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21) & \\ 0.1102(A - 8.7) & \end{cases}$$

Once you have values for M and β , you can finally calculate the actual window weights. The Kaiser window equation makes use of another function I_0 , this is a Zeroth Order Modified Bessel

Function. Although it expands to infinity, the denominator quickly becomes very large, therefore you only really need to calculate $I_0(x)$ up to around $i=20$.

$$w(n) = \frac{I_0 \left(\beta \sqrt{1 - \left(\frac{2n}{M} - 1 \right)^2} \right)}{I_0(\beta)} \quad I_0(x) = \sum_{i=0}^{\infty} \frac{\left(\frac{x}{2} \right)^{2i}}{(i!)^2}$$

$$= 1 + \frac{\left(\frac{x}{2} \right)^2}{(1!)^2} + \frac{\left(\frac{x}{2} \right)^4}{(2!)^2} + \frac{\left(\frac{x}{2} \right)^6}{(3!)^2}$$

Below are some examples weights for the following Kaiser Window based low pass filter:

- **Ripple: 0.01**

- **Cut-off Frequency:** 250Hz
- **Transition Window:** 100Hz
- **Sampling Frequency:** 1000Hz
- **Filter Order, M :** 23
- **Beta:** 3.395321

The filter's frequency response is show below, for comparison a Blackman window based filter with the same number of weights is also shown. It can be seen that the transition frequency of 250Hz is correct, as is the transition width of 100Hz.

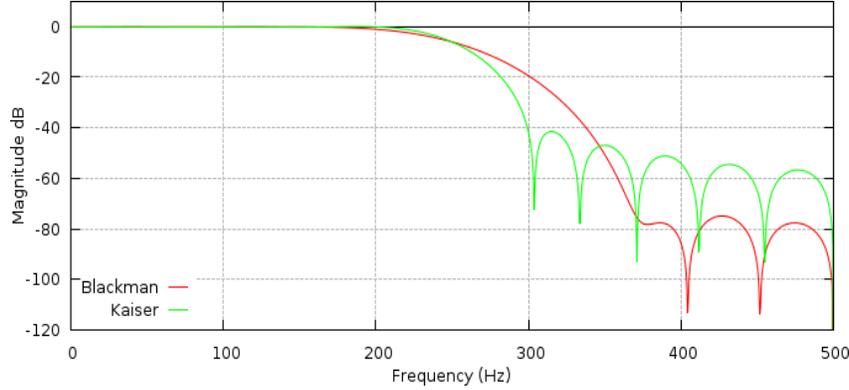


Fig. 12: Alternate Kaiser Window

Below is a rearrangement of the Kaiser Window equation.

$$\begin{aligned}
 \sqrt{1 - \left(\frac{2n}{M} - 1\right)^2} &= \sqrt{1 - \left(\frac{2n}{M} - 1\right) \left(\frac{2n}{M} - 1\right)} \\
 &= \sqrt{1 - \left(\frac{4n^2}{M^2} - \frac{4n}{M} + 1\right)} \\
 &= \sqrt{\frac{4n}{M} - \frac{4n^2}{M^2}} \\
 &= \sqrt{\frac{4nM - 4n^2}{M^2}} \\
 &= \frac{2}{M} \sqrt{nM - n^2} \\
 &= \frac{2}{M} \sqrt{n(M - n)}
 \end{aligned}$$

therefore
$$\frac{I_0\left(\beta \sqrt{1 - \left(\frac{2n}{M} - 1\right)^2}\right)}{I_0(\beta)} = \frac{I_0\left(\frac{2\beta}{M} \sqrt{n(M - n)}\right)}{I_0(\beta)}$$



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

**SCHOOL OF BIO AND CHEMICAL ENGINEERING
DEPARTMENT OF BIOMEDICAL ENGINEERING**

UNIT – IV – Digital Signal Processing & Its Applications – SEC1324

IIR FILTER DESIGNING

4.1 BUTTERWORTH AND CHEBYSHEV FILTERS

The **Butterworth filter** is a type of signal processing filter designed to have as flat a frequency response as possible in the passband. It is also referred to as a **maximally flat magnitude filter**. It was first described in 1930 by the British engineer and physicist Stephen Butterworth in his paper entitled "On the Theory of Filter Amplifiers".^[1]

Butterworth had a reputation for solving "impossible" mathematical problems. At the time, filter design required a considerable amount of designer experience due to limitations of the theory then in use. The filter was not in common use for over 30 years after its publication. Butterworth stated that:

"An ideal electrical filter should not only completely reject the unwanted frequencies but should also have uniform sensitivity for the wanted frequencies".

Such an ideal filter cannot be achieved but Butterworth showed that successively closer approximations were obtained with increasing numbers of filter elements of the right values. At the time, filters generated substantial ripple in the passband, and the choice of component values was highly interactive. Butterworth showed that a low pass filter could be designed whose cutoff frequency was normalized to 1 radian per second and whose frequency response (gain) was

$$G(\omega) = \sqrt{\frac{1}{1 + \omega^{2n}}}$$

where ω is the angular frequency in radians per second and n is the number of poles in the filter—equal to the number of reactive elements in a passive filter. If $\omega = 1$, the amplitude response of this type of filter in the passband is $1/\sqrt{2} \approx 0.707$, which is half power or -3 dB. Butterworth only dealt with filters with an even number of poles in his paper. He may have been unaware that such filters could be designed with an odd number of poles. He built his higher order filters from 2-pole filters separated by vacuum tube amplifiers. His plot of the frequency response of 2, 4, 6, 8, and 10 pole filters is shown as A, B, C, D, and E in his original graph.

Butterworth solved the equations for two- and four-pole filters, showing how the latter could be cascaded when separated by vacuum tube amplifiers and so enabling the construction of higher-order filters despite inductor losses. In 1930, low-loss core materials such as molypermalloy had not been discovered and air-cored audio inductors were rather lossy. Butterworth discovered that

it was possible to adjust the component values of the filter to compensate for the winding resistance of the inductors.

He used coil forms of 1.25" diameter and 3" length with plug-in terminals. Associated capacitors and resistors were contained inside the wound coil form. The coil formed part of the plate load resistor. Two poles were used per vacuum tube and RC coupling was used to the grid of the following tube.

Butterworth also showed that his basic low-pass filter could be modified to give low-pass, high-pass, band-pass and band-stop functionality.

4.2 OVERVIEW

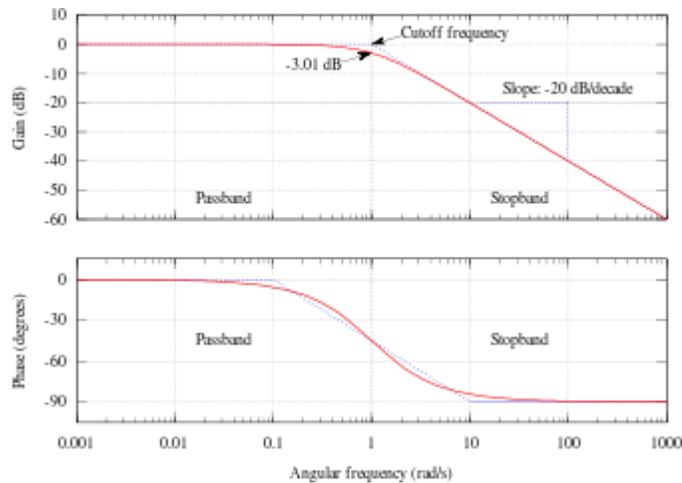


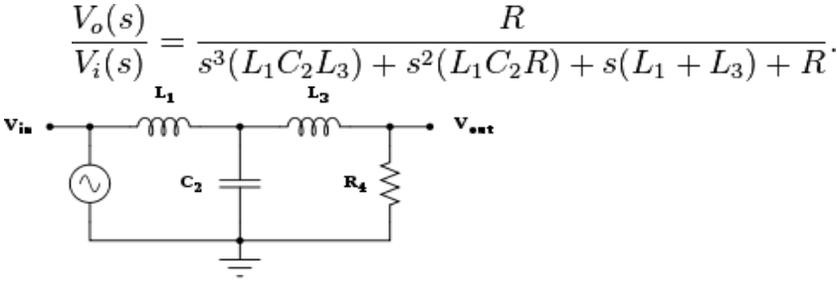
Fig. 1: The Bode plot of a first-order Butterworth low-pass filter

The frequency response of the Butterworth filter is maximally flat (i.e. has no ripples) in the passband and rolls off towards zero in the stopband.^[2] When viewed on a logarithmic Bode plot, the response slopes off linearly towards negative infinity. A first-order filter's response rolls off at -6 dB per octave (-20 dB per decade) (all first-order lowpass filters have the same normalized frequency response). A second-order filter decreases at -12 dB per octave, a third-order at -18 dB and so on. Butterworth filters have a monotonically changing magnitude function with ω , unlike other filter types that have non-monotonic ripple in the passband and/or the stopband.

Compared with a Chebyshev Type I/Type II filter or an elliptic filter, the Butterworth filter has a slower roll-off, and thus will require a higher order to implement a particular stopband specification, but Butterworth filters have a more linear phase response in the pass-band than Chebyshev Type I/Type II and elliptic filters can achieve.

Example

A transfer function of a third-order low-pass Butterworth filter design shown in the figure on the right looks like this:



A third-order low-pass filter (Cauer topology). The filter becomes a Butterworth filter with cutoff frequency $\omega_c=1$ when (for example) $C_2=4/3$ farad, $R_4=1$ ohm, $L_1=3/2$ henry and $L_3=1/2$ henry.

A simple example of a Butterworth filter is the third-order low-pass design shown in the figure on the right, with $C_2 = 4/3$ F, $R_4 = 1 \Omega$, $L_1 = 3/2$ H, and $L_3 = 1/2$ H.^[3] Taking the impedance of the capacitors C to be $1/(Cs)$ and the impedance of the inductors L to be Ls , where $s = \sigma + j\omega$ is the complex frequency, the circuit equations yield the transfer function for this device:

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{1}{1 + 2s + 2s^2 + s^3}$$

The magnitude of the frequency response (gain) $G(\omega)$ is given by

$$G(\omega) = |H(j\omega)| = \frac{1}{\sqrt{1 + \omega^6}},$$

obtained from

$$G^2(\omega) = |H(j\omega)|^2 = H(j\omega)H^*(j\omega) = \frac{1}{1 + \omega^6},$$

and the phase is given by

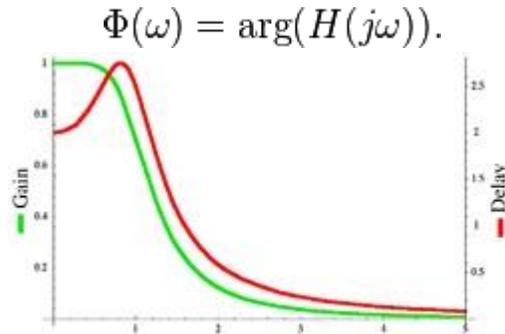


Fig. 2: Gain and group delay of the third-order Butterworth filter with $\omega_c=1$

The group delay is defined as the derivative of the phase with respect to angular frequency and is a measure of the distortion in the signal introduced by phase differences for different frequencies. The gain and the delay for this filter are plotted in the graph on the left. It can be seen that there are no ripples in the gain curve in either the passband or the stop band.

The log of the absolute value of the transfer function $H(s)$ is plotted in complex frequency space in the second graph on the right. The function is defined by the three poles in the left half of the complex frequency plane.

Log density plot of the transfer function $H(s)$ in complex frequency space for the third-order Butterworth filter with $\omega_c=1$. The three poles lie on a circle of unit radius in the left half-plane.

These are arranged on a circle of radius unity, symmetrical about the real s axis. The gain function will have three more poles on the right half plane to complete the circle.

By replacing each inductor with a capacitor and each capacitor with an inductor, a high-pass Butterworth filter is obtained.

A band-pass Butterworth filter is obtained by placing a capacitor in series with each inductor and an inductor in parallel with each capacitor to form resonant circuits. The value of each new component must be selected to resonate with the old component at the frequency of interest.

A band-stop Butterworth filter is obtained by placing a capacitor in parallel with each inductor and an inductor in series with each capacitor to form resonant circuits. The value of each new component must be selected to resonate with the old component at the frequency to be rejected.

4.3 TRANSFER FUNCTION

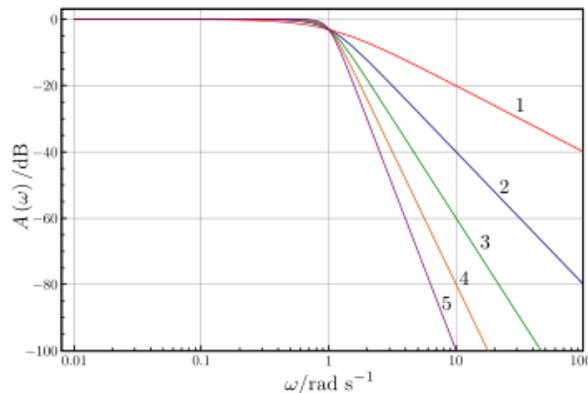


Fig. 3: Plot of the gain of Butterworth low-pass filters of orders 1 through 5, with cutoff frequency $\omega_0 = 1$. Note that the slope is $20n$ dB/decade where n is the filter order.

Like all filters, the typical prototype is the low-pass filter, which can be modified into a high-pass filter, or placed in series with others to form band-pass and band-stop filters, and higher order versions of these.

The gain $G(\omega)$ of an n -order Butterworth low pass filter is given in terms of the transfer function $H(s)$ as

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

where

- n = order of filter
- ω_c = cutoff frequency (approximately the -3dB frequency)
- G_0 is the DC gain (gain at zero frequency)

It can be seen that as n approaches infinity, the gain becomes a rectangle function and frequencies below ω_c will be passed with gain G_0 , while frequencies above ω_c will be suppressed. For smaller values of n , the cutoff will be less sharp.

We wish to determine the transfer function $H(s)$ where $s = \sigma + j\omega$ (from Laplace transform). Because $|H(s)|^2 = H(s)\overline{H(s)}$ and, as a general property of Laplace transforms at $s = j\omega$, $H(-j\omega) = \overline{H(j\omega)}$, if we select $H(s)$ such that:

$$H(s)H(-s) = \frac{G_0^2}{1 + \left(\frac{-s^2}{\omega_c^2}\right)^n},$$

then, for imaginary inputs, $s = j\omega$, we have the frequency response of the Butterworth filter.

The n poles of this expression occur on a circle of radius ω_c at equally-spaced points, and symmetric around the imaginary axis. For stability, the transfer function, $H(s)$, is therefore chosen such that it contains only the poles in the negative real half-plane of s . The k -th pole is specified by

$$-\frac{s_k^2}{\omega_c^2} = (-1)^{\frac{1}{n}} = e^{\frac{j(2k-1)\pi}{n}} \quad k = 1, 2, 3, \dots, n$$

and hence;

$$s_k = \omega_c e^{\frac{j(2k+n-1)\pi}{2n}} \quad k = 1, 2, 3, \dots, n.$$

The transfer(or system) function may be written in terms of these poles as

$$H(s) = \frac{G_0}{\prod_{k=1}^n (s - s_k) / \omega_c}$$

The denominator is a Butterworth polynomial in s .

4.4 NORMALIZED BUTTERWORTH POLYNOMIALS

The Butterworth polynomials may be written in complex form as above, but are usually written with real coefficients by multiplying that pole pairs that are complex conjugates, such as s_1 and s_n . The polynomials are normalized by setting $\omega_c = 1$. The normalized Butterworth polynomials then have the general form

$$B_n(s) = \prod_{k=1}^{\frac{n}{2}} \left[s^2 - 2s \cos \left(\frac{2k + n - 1}{2n} \pi \right) + 1 \right] \quad n = \text{even}$$

$$B_n(s) = (s + 1) \prod_{k=1}^{\frac{n-1}{2}} \left[s^2 - 2s \cos \left(\frac{2k + n - 1}{2n} \pi \right) + 1 \right] \quad n = \text{odd.}$$

To four decimal places, they are

n Factors of Polynomial $B_n(s)$

1 $(s + 1)$

2 $s^2 + 1.4142s + 1$

3 $(s + 1)(s^2 + s + 1)$

4 $(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)$

5 $(s + 1)(s^2 + 0.6180s + 1)(s^2 + 1.6180s + 1)$

6 $(s^2 + 0.5176s + 1)(s^2 + 1.4142s + 1)(s^2 + 1.9319s + 1)$

7 $(s + 1)(s^2 + 0.4450s + 1)(s^2 + 1.2470s + 1)(s^2 + 1.8019s + 1)$

8 $(s^2 + 0.3902s + 1)(s^2 + 1.1111s + 1)(s^2 + 1.6629s + 1)(s^2 + 1.9616s + 1)$

The normalized Butterworth polynomials can be used to determine the transfer function for any low-pass filter cut-off frequency ω_c , as follows

$$H(s) = \frac{G_0}{B_n(a)}, \text{ where } a = \frac{s}{\omega_c}$$

Transformation to other bandforms are also possible, see prototype filter.

4.4.1 MAXIMAL FLATNESS

Assuming $\omega_c = 1$ and $G_0 = 1$, the derivative of the gain with respect to frequency can be shown to be

$$\frac{dG}{d\omega} = -nG^3\omega^{2n-1}$$

which is monotonically decreasing for all ω since the gain G is always positive. The gain function of the Butterworth filter therefore has no ripple. Furthermore, the series expansion of the gain is given by

$$G(\omega) = 1 - \frac{1}{2}\omega^{2n} + \frac{3}{8}\omega^{4n} + \dots$$

In other words, all derivatives of the gain up to but not including the $2n$ -th derivative are zero at $\omega = 0$, resulting in "maximal flatness". If the requirement to be monotonic is limited to the passband only and ripples are allowed in the stopband, then it is possible to design a filter of the same order, such as the inverse Chebyshev filter, that is flatter in the passband than the "maximally flat" Butterworth.

4.4.2 HIGH-FREQUENCY ROLL-OFF

Again assuming $\omega_c = 1$, the slope of the log of the gain for large ω is

$$\lim_{\omega \rightarrow \infty} \frac{d \log(G)}{d \log(\omega)} = -n.$$

In decibels, the high-frequency roll-off is therefore $20n$ dB/decade, or $6n$ dB/octave (the factor of 20 is used because the power is proportional to the square of the voltage gain; see 20 log rule.)

4.5 CHEBYSHEV FILTERS

are analog or digital filters having a steeper roll-off and more passband ripple (type I) or stopband ripple (type II) than Butterworth filters. Chebyshev filters have the property that they minimize the error between the idealized and the actual filter characteristic over the range of the filter,^[citation needed] but with ripples in the passband. This type of filter is named after Pafnuty Chebyshev because its mathematical characteristics are derived from Chebyshev polynomials.

Because of the passband ripple inherent in Chebyshev filters, the ones that have a smoother response in the passband but a more irregular response in the stopband are preferred for some applications.

4.5.1 TYPE I CHEBYSHEV FILTERS

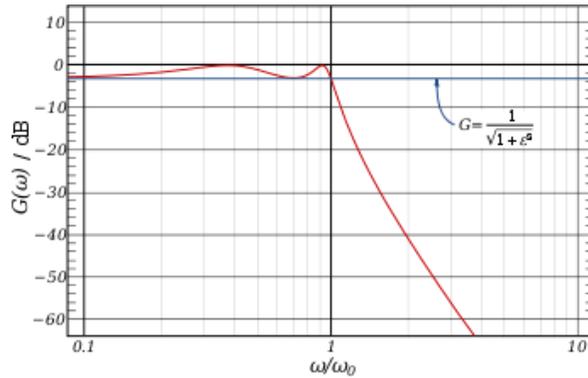


Fig. 4: The frequency response of a fourth-order type I Chebyshev low-pass filter with $\varepsilon = 1$

Type I Chebyshev filters are the most common types of Chebyshev filters. The gain (or amplitude) response as a function of angular frequency ω of the n th-order low-pass filter is equal to the absolute value of the transfer function $H_n(j\omega)$:

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

where ε is the ripple factor, ω_0 is the cutoff frequency and T_n is a Chebyshev polynomial of the n th order.

The passband exhibits equiripple behavior, with the ripple determined by the ripple factor ε . In the passband, the Chebyshev polynomial alternates between -1 and 1 so the filter gain alternates between maxima at $G = 1$ and minima at $G = 1/\sqrt{1 + \varepsilon^2}$. At the cutoff frequency ω_0 the gain again has the value $1/\sqrt{1 + \varepsilon^2}$ but continues to drop into the stopband as the frequency increases. This behavior is shown in the diagram on the right. The common practice of defining the cutoff frequency at -3 dB is usually not applied to Chebyshev filters; instead the cutoff is taken as the point at which the gain falls to the value of the ripple for the final time.

The order of a Chebyshev filter is equal to the number of reactive components (for example, inductors) needed to realize the filter using analog electronics.

The ripple is often given in dB:

$$\text{Ripple in dB} = 10 \log_{10}(1 + \varepsilon^2)$$

so that a ripple amplitude of 3 dB results from $\varepsilon = 1$.

An even steeper roll-off can be obtained if ripple is allowed in the stopband, by allowing zeroes on the $j\omega$ -axis in the complex plane. However, this results in less suppression in the stopband. The result is called an elliptic filter, also known as Cauer filter.

Poles and zeroes

Log of the absolute value of the gain of an 8th order Chebyshev type I filter in complex frequency space ($s = \sigma + j\omega$) with $\varepsilon = 0.1$ and $\omega_0 = 1$. The white spots are poles and are arranged on an ellipse with a semi-axis of 0.3836... in σ and 1.071... in ω . The transfer function poles are those poles in the left half plane. Black corresponds to a gain of 0.05 or less, white corresponds to a gain of 20 or more.

For simplicity, it is assumed that the cutoff frequency is equal to unity. The poles (ω_{pm}) of the gain function of the Chebyshev filter are the zeroes of the denominator of the gain function. Using the complex frequency s , these occur when:

$$1 + \varepsilon^2 T_n^2(-js) = 0.$$

Defining $-js = \cos(\theta)$ and using the trigonometric definition of the Chebyshev polynomials yields:

$$1 + \varepsilon^2 T_n^2(\cos(\theta)) = 1 + \varepsilon^2 \cos^2(n\theta) = 0.$$

Solving for θ

$$\theta = \frac{1}{n} \arccos\left(\frac{\pm j}{\varepsilon}\right) + \frac{m\pi}{n}$$

where the multiple values of the arc cosine function are made explicit using the integer index m . The poles of the Chebyshev gain function are then:

$$\begin{aligned} s_{pm} &= j \cos(\theta) \\ &= j \cos\left(\frac{1}{n} \arccos\left(\frac{\pm j}{\varepsilon}\right) + \frac{m\pi}{n}\right). \end{aligned}$$

Using the properties of the trigonometric and hyperbolic functions, this may be written in explicitly complex form:

$$s_{pm}^{\pm} = \pm \sinh\left(\frac{1}{n} \operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right) \sin(\theta_m)$$

$$+j \cosh\left(\frac{1}{n} \operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right) \cos(\theta_m)$$

where $m = 1, 2, \dots, n$ and

$$\theta_m = \frac{\pi}{2} \frac{2m-1}{n}.$$

This may be viewed as an equation parametric in θ_m and it demonstrates that the poles lie on an ellipse in s -space centered at $s = 0$ with a real semi-axis of length $\sinh(\operatorname{arsinh}(1/\varepsilon)/n)$ and an imaginary semi-axis of length of $\cosh(\operatorname{arsinh}(1/\varepsilon)/n)$.

The transfer function

The above expression yields the poles of the gain G . For each complex pole, there is another which is the complex conjugate, and for each conjugate pair there are two more that are the negatives of the pair. The transfer function must be stable, so that its poles are those of the gain that have negative real parts and therefore lie in the left half plane of complex frequency space.

The transfer function is then given by

$$H(s) = \frac{1}{2^{n-1}\varepsilon} \prod_{m=1}^n \frac{1}{(s - s_{pm}^-)}$$

where s_{pm}^- are only those poles with a negative sign in front of the real term in the above equation for the poles.

The group delay

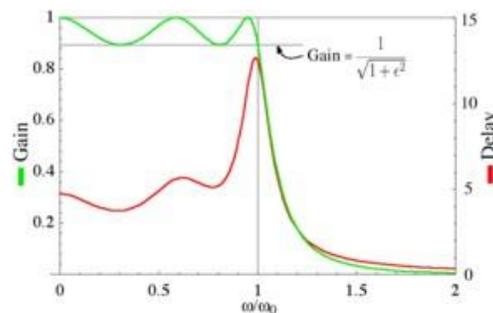


Fig. 5: Gain and group delay of a fifth-order type I Chebyshev filter with $\varepsilon = 0.5$.

The group delay is defined as the derivative of the phase with respect to angular frequency and is a measure of the distortion in the signal introduced by phase differences for different frequencies.

$$\tau_g = -\frac{d}{d\omega} \arg(H(j\omega))$$

The gain and the group delay for a fifth-order type I Chebyshev filter with $\epsilon=0.5$ are plotted in the graph on the left. It can be seen that there are ripples in the gain and the group delay in the passband but not in the stopband.

4.5.2 TYPE II CHEBYSHEV FILTERS

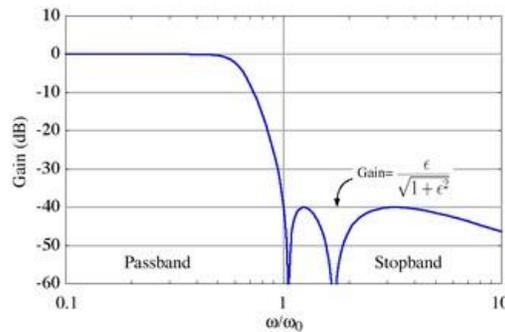


Fig. 6: The frequency response of a fifth-order type II Chebyshev low-pass filter with $\epsilon = 0.01$

Also known as inverse Chebyshev filters, the Type II Chebyshev filter type is less common because it does not roll off as fast as Type I, and requires more components. It has no ripple in the passband, but does have equiripple in the stopband. The gain is:

$$G_n(\omega, \omega_0) = \frac{1}{\sqrt{1 + \frac{1}{\epsilon^2 T_n^2(\omega_0/\omega)}}}$$

In the stopband, the Chebyshev polynomial oscillates between -1 and 1 so that the gain will oscillate between zero and

$$\frac{1}{\sqrt{1 + \frac{1}{\epsilon^2}}}$$

and the smallest frequency at which this maximum is attained is the cutoff frequency ω_c . The parameter ϵ is thus related to the stopband attenuation γ in decibels by:

$$\epsilon = \frac{1}{\sqrt{10^{0.1\gamma} - 1}}$$

For a stopband attenuation of 5 dB, $\varepsilon = 0.6801$; for an attenuation of 10 dB, $\varepsilon = 0.3333$. The frequency $f_0 = \omega_0/2\pi$ is the cutoff frequency. The 3 dB frequency f_H is related to f_0 by:

$$f_H = \frac{f_0}{\cosh\left(\frac{1}{n} \cosh^{-1} \frac{1}{\varepsilon}\right)}.$$

Poles and zeroes

Log of the absolute value of the gain of an 8th order Chebyshev type II filter in complex frequency space ($s=\sigma+j\omega$) with $\varepsilon = 0.1$ and $\omega_0 = 1$. The white spots are poles and the black spots are zeroes. All 16 poles are shown. Each zero has multiplicity of two, and 12 zeroes are shown and four are located outside the picture, two on the positive ω axis, and two on the negative. The poles of the transfer function are poles on the left half plane and the zeroes of the transfer function are the zeroes, but with multiplicity 1. Black corresponds to a gain of 0.05 or less, white corresponds to a gain of 20 or more.

Assuming that the cutoff frequency is equal to unity, the poles (ω_{pm}) of the gain of the Chebyshev filter are the zeroes of the denominator of the gain:

$$1 + \varepsilon^2 T_n^2(-1/j s_{pm}) = 0.$$

The poles of gain of the type II Chebyshev filter are the inverse of the poles of the type I filter:

$$\frac{1}{s_{pm}^\pm} = \pm \sinh\left(\frac{1}{n} \operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right) \sin(\theta_m) + j \cosh\left(\frac{1}{n} \operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right) \cos(\theta_m)$$

where $m = 1, 2, \dots, n$. The zeroes (ω_{zm}) of the type II Chebyshev filter are the zeroes of the numerator of the gain:

$$\varepsilon^2 T_n^2(-1/j s_{zm}) = 0.$$

The zeroes of the type II Chebyshev filter are therefore the inverse of the zeroes of the Chebyshev polynomial.

$$1/s_{zm} = -j \cos\left(\frac{\pi}{2} \frac{2m-1}{n}\right)$$

for $m = 1, 2, \dots, n$.

The transfer function

The transfer function is given by the poles in the left half plane of the gain function, and has the same zeroes but these zeroes are single rather than double zeroes.

The group delay

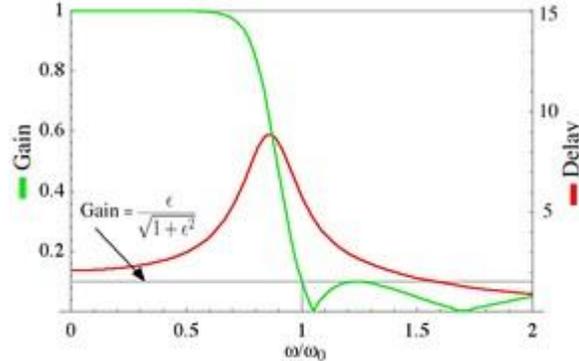


Fig. 7: Gain and group delay of a fifth-order type II Chebyshev filter with $\epsilon = 0.1$.

The gain and the group delay for a fifth-order type II Chebyshev filter with $\epsilon=0.1$ are plotted in the graph on the left. It can be seen that there are ripples in the gain in the stopband but not in the pass band.

4.6 IMPLEMENTATION CAUER TOPOLOGY

A passive LC Chebyshev low-pass filter may be realized using a Cauer topology. The inductor or capacitor values of a n th-order Chebyshev prototype filter may be calculated from the following equations:^[1]

$$\begin{aligned}
 G_0 &= 1 \\
 G_1 &= \frac{2A_1}{\gamma} \\
 G_k &= \frac{4A_{k-1}A_k}{B_{k-1}G_{k-1}}, \quad k = 2, 3, 4, \dots, n \\
 G_{n+1} &= \begin{cases} 1 & \text{if } n \text{ odd} \\ \coth^2\left(\frac{\beta}{4}\right) & \text{if } n \text{ even} \end{cases}
 \end{aligned}$$

G_1, G_k are the capacitor or inductor element values. f_H , the 3 dB frequency is calculated with:

$$f_H = f_0 \cosh\left(\frac{1}{n} \cosh^{-1} \frac{1}{\epsilon}\right)$$

The coefficients A , γ , β , A_k , and B_k may be calculated from the following equations:

$$\begin{aligned}\gamma &= \sinh\left(\frac{\beta}{2n}\right) \\ \beta &= \ln\left[\coth\left(\frac{R_{db}}{17.37}\right)\right] \\ A_k &= \sin\frac{(2k-1)\pi}{2n}, \quad k = 1, 2, 3, \dots, n \\ B_k &= \gamma^2 + \sin^2\left(\frac{k\pi}{n}\right), \quad k = 1, 2, 3, \dots, n\end{aligned}$$

where R_{dB} is the passband ripple in decibels.

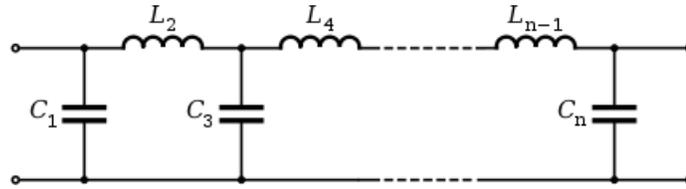


Fig. 8: Low-pass filter using Cauer topology

The calculated G_k values may then be converted into shunt capacitors and series inductors as shown on the right, or they may be converted into series capacitors and shunt inductors. For example,

- $C_1 \text{ shunt} = G_1, L_2 \text{ series} = G_2, \dots$

or

- $L_1 \text{ shunt} = G_1, C_1 \text{ series} = G_2, \dots$

Note that when G_1 is a shunt capacitor or series inductor, G_0 corresponds to the input resistance or conductance, respectively. The same relationship holds for G_{n+1} and G_n . The resulting circuit is a normalized low-pass filter. Using frequency transformations and impedance scaling, the normalized low-pass filter may be transformed into high-pass, band-pass, and band-stop filters of any desired cutoff frequency or bandwidth.

Digital

As with most analog filters, the Chebyshev may be converted to a digital (discrete-time) recursive form via the bilinear transform. However, as digital filters have a finite bandwidth, the response shape of the transformed Chebyshev is warped. Alternatively, the Matched Z-transform method may be used, which does not warp the response.

4.7 COMPARISON WITH OTHER LINEAR FILTERS

The following illustration shows the Chebyshev filters next to other common filter types obtained with the same number of coefficients (fifth order):

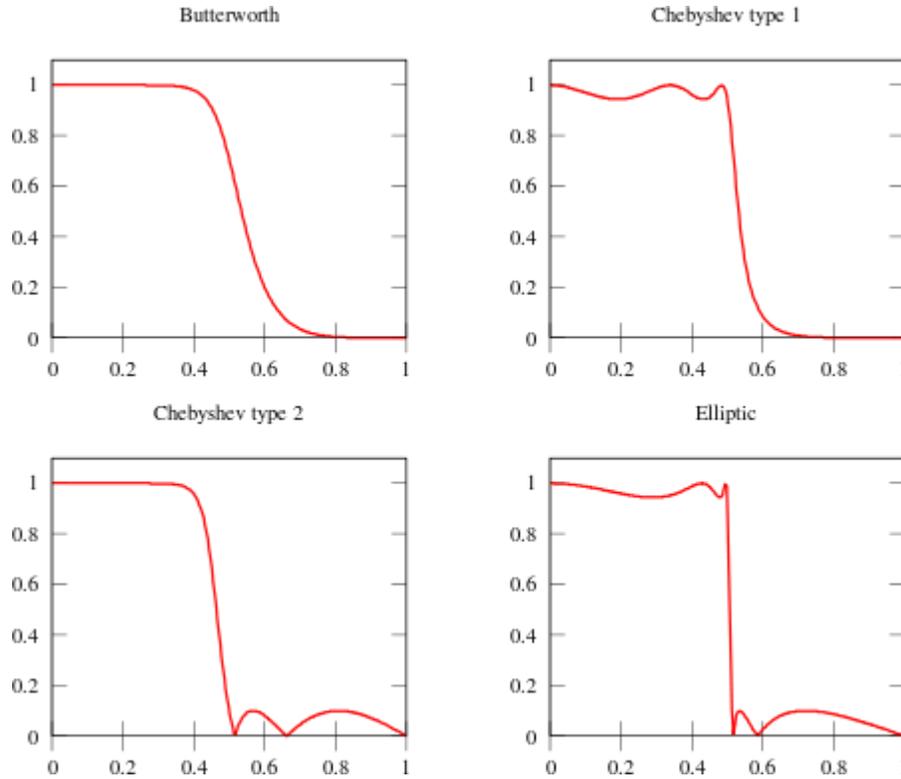


Fig. 9: Chebyshev filters are sharper than the Butterworth filter; they are not as sharp as the elliptic one, but they show fewer ripples over the bandwidth.

frequency transformation in analog and digital domain

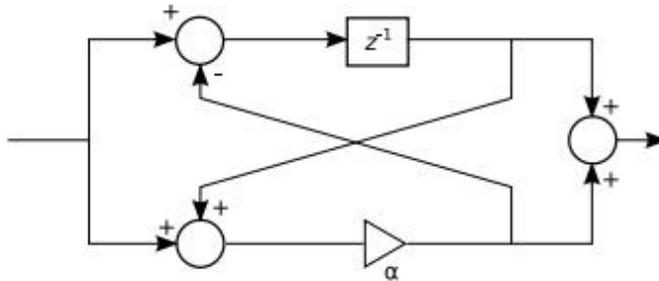
There are two approaches to the design of digital filters of bandpass, highpass, and bandstop types. Approach 1: Design an analog lowpass filter, apply the frequency band transformations in analog domain, and then map the relevant filter to a digital filter. Disadvantage Due to aliasing problem inherent in the use of impulse invariant technique a bandpass or highpass filter cannot be transformed. Approach 2: Design an analog lowpass filter, map it to a digital filter, and then apply frequency band transformations in digital domain to obtain the desired digital filter. In this handout we introduce the second approach by defining several transformations to map a LPF to a LPF, BPF, and HPF with given specifications. Define a mapping from the z -plane to the \tilde{z} -plane of the form $z - 1 = f(\tilde{z} - 1)$ (1) such that the transfer function $H(z - 1)$ mapped $\rightarrow G(\tilde{z} - 1)$ (2) Conditions for this mapping are a $f(\cdot)$ is real and rational. b $f(\cdot)$ must produce stable $G(\tilde{z} - 1)$ from stable $H(z - 1)$ i.e. the interior of the unit circle in the z -plane must be mapped to the interior of the unit circle in the \tilde{z} -plane. c $|f(\tilde{z} - 1)| = 1, |z - 1| = 1$ d The inverse mapping exists

i.e. $\tilde{z}^{-1} = f^{-1}(z^{-1})$ A class of transformation for mapping a LPF to another LPF with different frequency characteristics or to a HPF, has a general form of $f(\tilde{z}^{-1}) = a_0 + a_1 \tilde{z}^{-1} + b_1 \tilde{z}^{-1}$

Lowpass to Lowpass Mapping

The transformation

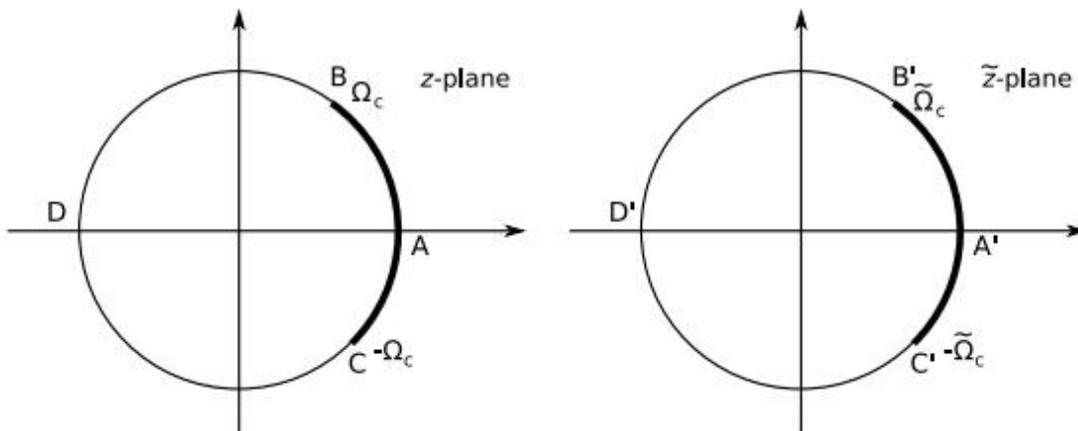
$$z^{-1} = f(\tilde{z}^{-1}) = \frac{\tilde{z}^{-1} - \alpha}{1 - \alpha \tilde{z}^{-1}}$$



which satisfies the above conditions can be used to map a LPF to another LPF. Let $z = e^{j\Omega T}$ and $\tilde{z} = e^{j\tilde{\Omega} T}$. Then the relationship between the frequencies Ω and $\tilde{\Omega}$ is

$$\Omega = \frac{2}{T} \tan^{-1}\left(K \tan \frac{\tilde{\Omega} T}{2}\right), \quad K = \frac{1 + \alpha}{1 - \alpha}$$

where α can be obtained based upon the cutoff frequency of the transformed filter.

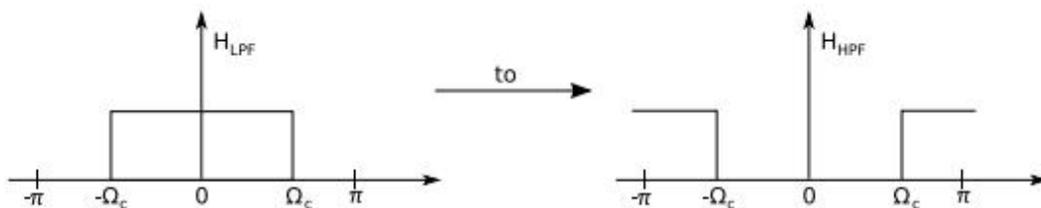


(a) Passband and stopband of the original LPF on the unit circle

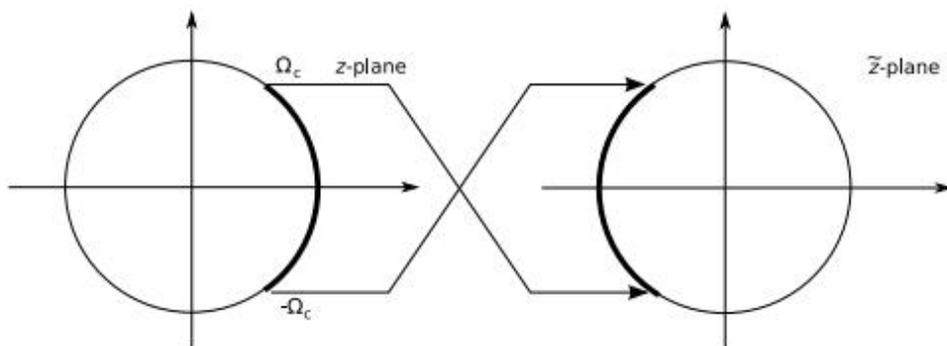
(b) Passband and stopband of the mapped LPF on the unit circle

Lowpass to Highpass Mapping

We desire to map



On the unit circle we must rotate the frequency band.



Thus, the mapping is

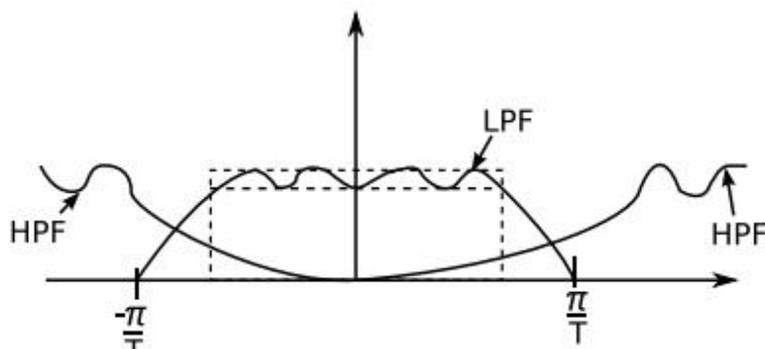
$$z^{-1} \longrightarrow -\tilde{z}^{-1},$$

or in general

$$z^{-1} \longrightarrow \frac{-(\tilde{z}^{-1} + \alpha)}{1 + \alpha\tilde{z}^{-1}}$$

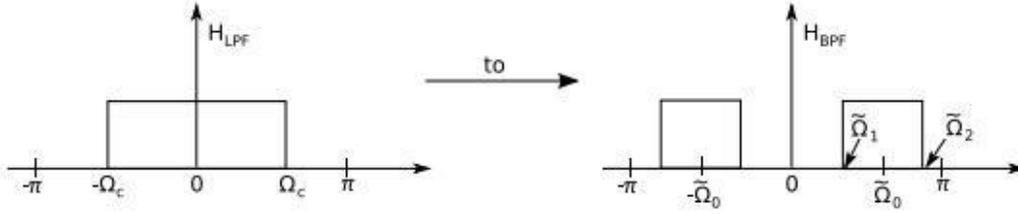
The cutoff frequency of the HPF is related to that of LPF by

$$\Omega_{c_{HP}} + \Omega_{c_{LP}} = \frac{\pi}{T} \text{ (normalized case)}$$



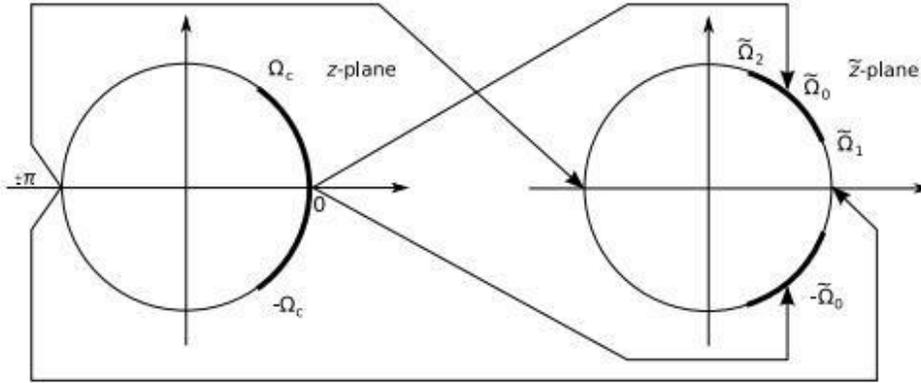
Lowpass to Bandpass Mapping

We desire to map



Note that since every point in LPF characteristics is mapped to two points in that of the BPF, we need a 2nd order mapping i.e.

$$z^{-1} = f(\tilde{z}^{-1}) = \frac{a_0 + a_1\tilde{z}^{-1} + a_2\tilde{z}^{-2}}{b_0 + b_1\tilde{z}^{-1} + b_2\tilde{z}^{-2}}$$



It can be shown that the mapping which satisfies the above mentioned conditions will have a form

$$z^{-1} = f(\tilde{z}^{-1}) = - \left[\frac{a_0 + a_1\tilde{z}^{-1} + a_2\tilde{z}^{-2}}{a_2 + a_1\tilde{z}^{-1} + a_0\tilde{z}^{-2}} \right]$$

or in a more useful form

$$z^{-1} = f(\tilde{z}^{-1}) = - \left[\frac{\tilde{z}^{-2} - \frac{2\alpha K}{K+1}\tilde{z}^{-1} + \frac{K-1}{K+1}}{\frac{K-1}{K+1}\tilde{z}^{-2} - \frac{2\alpha K}{K+1}\tilde{z}^{-1} + 1} \right]$$

where

$$\begin{aligned} K &= \frac{a_0 + a_2}{a_0 - a_2} \\ &= \tan \frac{\Omega_c T}{2} \left[\cot \left[\left(\frac{\tilde{\Omega}_2 - \tilde{\Omega}_1}{2} \right) T \right] \right] \end{aligned} \quad (12)$$

and $\alpha = \cos \tilde{\Omega}_0$, $T = -a_1/(a_0 + a_2)$, and $\tan \left(\frac{(\tilde{\Omega}_1 - \tilde{\Omega}_2)T}{2} \right) = \tan \left(\frac{\Omega_c T}{2} \right)$

4.8 REALIZATION OF DIGITAL FILTERS DIRECT FORM I DIRECT FORM II

In signal processing, a **digital filter** is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. This is in contrast to the other major type of electronic filter, the analog filter, which is an electronic circuit operating on continuous-time analog signals.

A digital filter system usually consists of an analog-to-digital converter to sample the input signal, followed by a microprocessor and some peripheral components such as memory to store data and filter coefficients etc. Finally a digital-to-analog converter to complete the output stage. Program Instructions (software) running on the microprocessor implement the digital filter by performing the necessary mathematical operations on the numbers received from the ADC. In some high performance applications, an FPGA or ASIC is used instead of a general purpose microprocessor, or a specialized DSP with specific parallel architecture for expediting operations such as filtering.

Digital filters may be more expensive than an equivalent analog filter due to their increased complexity, but they make practical many designs that are impractical or impossible as analog filters. When used in the context of real-time analog systems, digital filters sometimes have problematic latency (the difference in time between the input and the response) due to the associated analog-to-digital and digital-to-analog conversions and anti-aliasing filters, or due to other delays in their implementation.

Characterization

A digital filter is characterized by its transfer function, or equivalently, its difference equation. Mathematical analysis of the transfer function can describe how it will respond to any input. As such, designing a filter consists of developing specifications appropriate to the problem (for example, a second-order low pass filter with a specific cut-off frequency), and then producing a transfer function which meets the specifications.

The transfer function for a linear, time-invariant, digital filter can be expressed as a transfer function in the Z-domain; if it is causal, then it has the form:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}$$

where the order of the filter is the greater of N or M . See Z-transform's LCCD equation for further discussion of this transfer function.

This is the form for a recursive filter with both the inputs (Numerator) and outputs (Denominator), which typically leads to an IIR infinite impulse response behaviour, but if

the denominator is made equal to unity i.e. no feedback, then this becomes an FIR or finite impulse response filter.

Analysis techniques

A variety of mathematical techniques may be employed to analyze the behaviour of a given digital filter. Many of these analysis techniques may also be employed in designs, and often form the basis of a filter specification.

Typically, one characterizes filters by calculating how they will respond to a simple input such as an impulse. One can then extend this information to compute the filter's response to more complex signals.

Impulse response

The impulse response, often denoted $h[k]$ or h_k , is a measurement of how a filter will respond to the Kronecker delta function. For example, given a difference equation, one would set $x_0 = 1$ and $x_k = 0$ for $k \neq 0$ and evaluate. The impulse response is a characterization of the filter's behaviour. Digital filters are typically considered in two categories: infinite impulse response (IIR) and finite impulse response (FIR). In the case of linear time-invariant FIR filters, the impulse response is exactly equal to the sequence of filter coefficients:

$$y_n = \sum_{k=0}^N h_k x_{n-k}$$

IIR filters on the other hand are recursive, with the output depending on both current and previous inputs as well as previous outputs. The general form of an IIR filter is thus:

$$\sum_{m=0}^M a_m y_{n-m} = \sum_{k=0}^N b_k x_{n-k}$$

Plotting the impulse response will reveal how a filter will respond to a sudden, momentary disturbance.

Difference equation

In discrete-time systems, the digital filter is often implemented by converting the transfer function to a linear constant-coefficient difference equation (LCCD) via the Z-transform. The discrete frequency-domain transfer function is written as the ratio of two polynomials. For example:

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

This is expanded:

$$H(z) = \frac{z^2 + 2z + 1}{z^2 + \frac{1}{4}z - \frac{3}{8}}$$

and to make the corresponding filter causal, the numerator and denominator are divided by the highest order of z

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2}} = \frac{Y(z)}{X(z)}$$

The coefficients of the denominator, a_k are the 'feed-backward' coefficients and the coefficients of the numerator are the 'feed-forward' coefficients, b_k . The resultant linear difference equation is:

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

or, for the example above:

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

rearranging terms:

$$\Rightarrow (1 + \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2})Y(z) = (1 + 2z^{-1} + z^{-2})X(z)$$

then by taking the inverse z -transform:

$$\Rightarrow y[n] + \frac{1}{4}y[n - 1] - \frac{3}{8}y[n - 2] = x[n] + 2x[n - 1] + x[n - 2]$$

and finally, by solving for $y[n]$:

$$y[n] = -\frac{1}{4}y[n - 1] + \frac{3}{8}y[n - 2] + x[n] + 2x[n - 1] + x[n - 2]$$

This equation shows how to compute the next output sample, $y[n]$, in terms of the past outputs, $y[n - p]$, the present input, $x[n]$, and the past inputs, $x[n - p]$. Applying the filter to an input in this form is equivalent to a Direct Form I or II realization, depending on the exact order of evaluation.

Filter design[\[edit\]](#)

The design of digital filters is a deceptively complex topic. Although filters are easily understood and calculated, the practical challenges of their design and implementation are significant and are the subject of much advanced research.

There are two categories of digital filter: the recursive filter and the nonrecursive filter. These are often referred to as infinite impulse response (IIR) filters and finite impulse response (FIR) filters, respectively

Filter realization

After a filter is designed, it must be *realized* by developing a signal flow diagram that describes the filter in terms of operations on sample sequences.

A given transfer function may be realized in manyways. Consider how a simple expression such as $ax + bx + c$ could be evaluated – one could also compute the equivalent $x(a + b) + c$. In the same way, all realizations may be seen as "factorizations" of the same transfer function, but different realizations will have different numerical properties. Specifically, some realizations are more efficient in terms of the number of operations or storage elements required for their implementation, and others provide advantages such as improved numerical stability and reduced round-off error. Some structures are better for fixed-point arithmetic and others may be better for floating-point arithmetic.

4.8.1 DIRECT FORM I

A straightforward approach for IIR filter realization is Direct Form I, where the difference equation is evaluated directly. This form is practical for small filters, but may be inefficient and impractical (numerically unstable) for complex designs. In general, this form requires $2N$ delay elements (for both input and output signals) for a filter of order N .

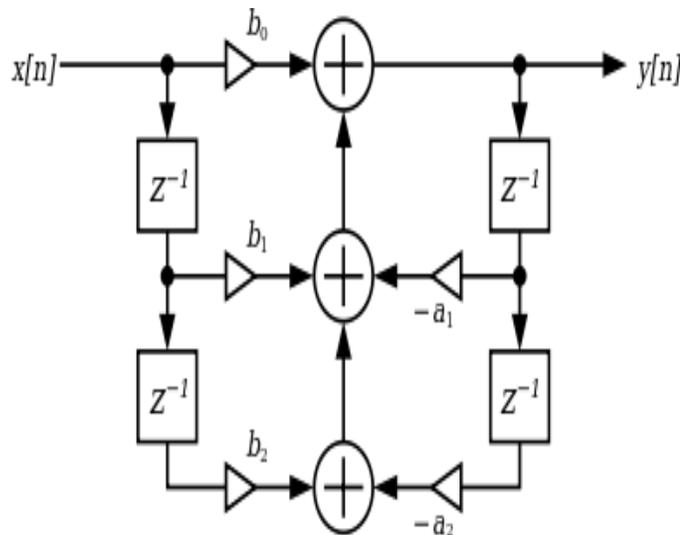


Fig. 10: Realization Structure

4.8.2 DIRECT FORM II

The alternate Direct Form II only needs N delay units, where N is the order of the filter – potentially half as much as Direct Form I. This structure is obtained by reversing the order of the numerator and denominator sections of Direct Form I, since they are in fact two linear systems, and the commutativity property applies. Then, one will notice that there are two columns of delays (z^{-1}) that tap off the center net, and these can be combined since they are redundant, yielding the implementation as shown below.

The disadvantage is that Direct Form II increases the possibility of arithmetic overflow for filters of high Q or resonance. It has been shown that as Q increases, the round-off noise of both direct form topologies increases without bounds.^[5] This is because, conceptually, the signal is first passed through an all-pole filter (which normally boosts gain at the resonant frequencies) before the result of that is saturated, then passed through an all-zero filter (which often attenuates much of what the all-pole half amplifies).

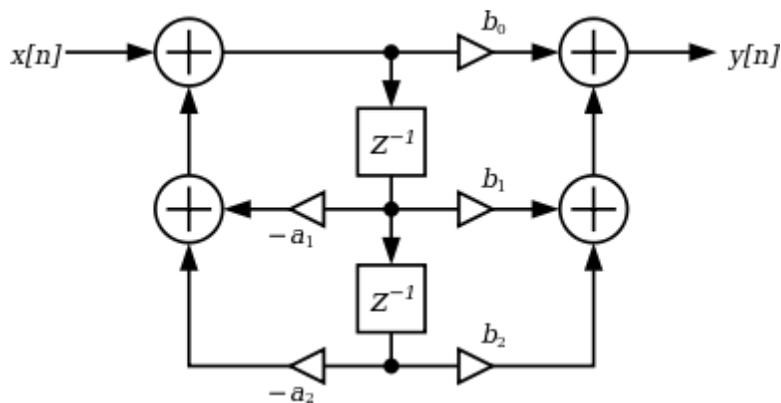


Fig. 11: Direct form II

Cascaded second-order sections

A common strategy is to realize a higher-order (greater than 2) digital filter as a cascaded series of second-order "biquadratic" (or "biquad") sections (see digital biquad filter). The advantage of this strategy is that the coefficient range is limited. Cascading direct form II sections results in N delay elements for filters of order N . Cascading direct form I sections results in $N+2$ delay elements since the delay elements of the input of any section (except the first section) are redundant with the delay elements of the output of the preceding section.



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

**SCHOOL OF BIO AND CHEMICAL ENGINEERING
DEPARTMENT OF BIOMEDICAL ENGINEERING**

UNIT – V – Digital Signal Processing & Its Applications – SEC1324

DSP APPLICATIONS

5.1 MULTIRATE SIGNAL PROCESSING

The processing of a discrete time signal at different sampling rates in different parts of a system is called **multirate DSP**. The discrete time systems that employ sampling rate conversion while processing the discrete time signals are called **multirate DSP systems**. The process of converting a signal from one sampling rate to another sampling rate is called **sampling rate conversion**.

There are two general methods for sampling rate conversion. In the first method, the discrete signal is converted to analog signal using a D/A converter and then the analog signal is resampled at the desired rate using an A/D converter. The advantage in this method is that the new sampling rate need not have any relation to the old sampling rate. The disadvantage in this method is signal distortion during D/A and A/D process.

In the second method, the sampling rate conversion is entirely performed in the digital domain, using interpolators and decimators. The advantage in rate conversion in the digital domain is that the signal distortion in D/A and A/D process are avoided or eliminated.

There are two ways for sampling rate conversion in the digital domain. They are,

1. Downsampling or decimation
2. Upsampling or interpolation

Downsampling or **decimation** is the process of reducing the sampling rate by an integer factor D. **Upsampling** or **interpolation** is the process of increasing the sampling rate by an integer factor I.

5.2 APPLICATIONS OF MULTIRATE DSP SYSTEM

Multirate signal processing is employed in the following systems.

- * Sub-band coding of speech signals and image compression
- * QMF (Quadrature Mirror Filters) for realizing alias-free LTI multirate systems
- * Narrowband FIR and IIR filters for various applications
- * Digital transmultiplexers for converting TDM (Time Division Multiplexed) signals to FDM (Frequency Division Multiplexed) signals and vice versa
- * Oversampling A/D (Analog-to-Digital) and D/A (Digital-to-Analog) converters for high quality digital audio systems and data loggers (or digital storage systems)
- * In digital audio systems the sampling rates of broadcasted signal, CD (Compact Disc), MPEG (Motion Picture Expert Group) standard CD, etc., are different. Hence to access signals from all these devices, sampling rate converters are needed in digital audio systems
- * In video broadcasting the American standard NTSC (National Television System Committee) and European standard PAL (Phase Alternating Line) employ different sampling rates. Hence to receive both the signals sampling rate converters are needed in video receivers

5.3 MERITS OF MULTIRATE PROCESSING

The advantages of multirate processing of discrete time signals are given below.

- * The reduction in number of computations
- * The reduction in memory requirement (or storage) for filter coefficients and intermediate results.
- * The reduction in the order of the system
- * The finite word length effects are reduced

5.4 DOWN SAMPLING OR DECIMATION

Downsampling (or **decimation**) is the process of reducing the samples of the discrete time signal.

Let, $x(n)$ = Discrete time signal

D = Sampling rate reduction factor (and D is an integer)

Now, $x(Dn)$ = Downsampled version of $x(n)$

The device which performs the process of downsampling is called a **downsampler** (or **decimator**). Symbolically, the downsampler can be represented as shown in fig 9.1.



Fig 1 : Decimator.

When the input signal to the decimator is not bandlimited then the spectrum of decimated signal has aliasing. In order to avoid aliasing the input signal should be bandlimited to $\frac{\pi}{D}$ for decimation by a factor D . Hence the input signal is passed through a lowpass filter with a bandwidth of $\frac{\pi}{D}$ before decimation. Since this lowpass filter is designed to avoid aliasing in the output spectrum of decimator, it is called **anti-aliasing filter**.

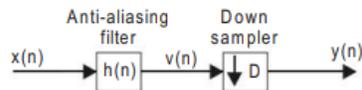


Fig 2 : Decimation by a factor D without aliasing.

5.5 UPSAMPLING OR INTERPOLATION

The **upsampling** (or **interpolation**) is the process of increasing the samples of the discrete time signal.

Let, $x(n)$ = Discrete time signal

I = Sampling rate multiplication factor (and I is an integer).

Now, $x\left(\frac{n}{I}\right)$ = Upsampled version of $x(n)$.

The device which perform the process of upsampling is called **upsampler** (or **interpolator**). Symbolically, the upsampler can be represented as shown in fig 9.5.

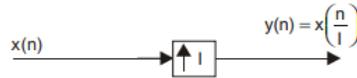


Fig 3 : Interpolator.

The output spectrum of interpolator is compressed version of the input spectrum,

Therefore, the spectrum of upsampled signal has multiple images in a period of 2π . When upsampled by a factor of I , the output spectrum will have I images in a period of 2π , with each image band limited to π/I . Since the frequency spectrum in the range 0 to $\frac{\pi}{I}$ are unique, we have to filter the other images. Hence the output of upsampler is passed through a lowpass filter with a bandwidth of $\frac{\pi}{I}$. Since this lowpass filter is designed to avoid multiple images in the output spectrum, it is called **anti-imaging filter**.

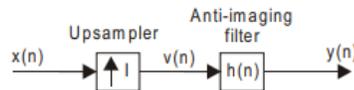


Fig 4 : Interpolation by a factor I without anti-imaging.

5.6 IMPLEMENTATION OF SAMPLING RATE CONVERSION USING DECIMATOR IN FIR FILTER

The transfer function of FIR filter is given by,

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

We know that, $H(z) = \frac{Y(z)}{X(z)}$

$$\therefore \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h(n) z^{-n}$$

On expanding the equation we get,

$$\begin{aligned} \frac{Y(z)}{X(z)} &= h(0) + h(1) z^{-1} + h(2) z^{-2} + h(3) z^{-3} + \dots + h(N-1) z^{-(N-1)} \\ \therefore Y(z) &= h(0) X(z) + h(1) z^{-1} X(z) + h(2) z^{-2} X(z) + h(3) z^{-3} X(z) + \\ &\quad \dots + h(N-1) z^{-(N-1)} X(z) \end{aligned}$$

Using the equation the direct form FIR Filter structure is drawn as shown in fig

Let us decimate the output of FIR Filter by introducing a decimator at the output of FIR filter as shown

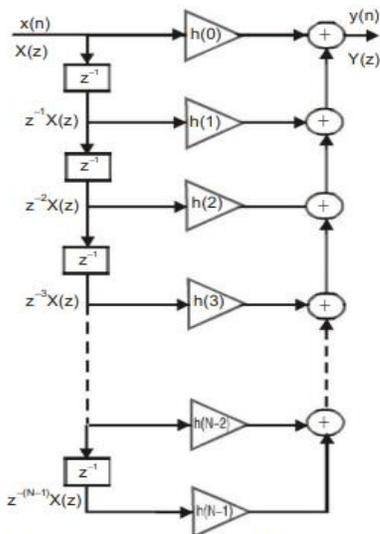


Fig 5 : Direct form FIR filter structure.

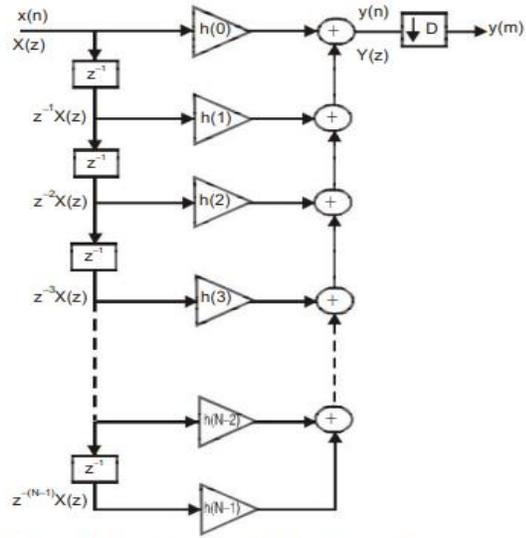


Fig 6 : Direct form FIR filter structure with decimator at the output.

In the filter structure of fig 6 we require only one out of every D samples of the output signal. But actually the system computes all the samples and discard $D - 1$ samples in every D samples and retain only one sample in every D samples. Hence this structure is an inefficient structure. To overcome the inefficiency in the calculations, (i.e., to avoid unwanted calculations, the decimator can be embedded in the filter structure itself. Therefore the decimator can be introduced before multipliers as shown in fig.7. so that the multiplications and additions can be performed at lower sampling rate (or performed only for the samples to be retained).

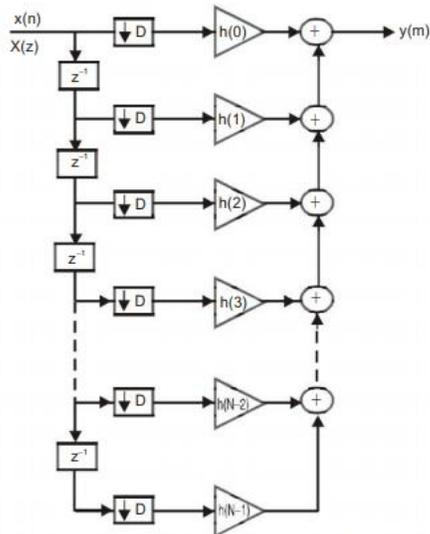


Fig 7 : Efficient structure for direct form FIR filter with decimator.

5.7 IMPLEMENTATION OF SAMPLING RATE CONVERSION USING INTERPOLATOR IN FIR FILTER

Consider the direct form FIR filter structure as shown in fig 5

Let us interpolate the input of FIR filter by introducing an upsampler at the input of FIR filter as shown in fig 8

In the filter structure of fig 8 the upsampler introduces $I-1$ zeros inbetween any two samples of $x(n)$ and so a large number of zeros are introduced in the upsampled input signal. Therefore most of the multiplications and additions will be product or sum of zeros. The multiplications and additions of zeros can be eliminated if the interpolator is embedded in the filter structure.

In order to embed the interpolator inside the filter structure, the direct form structure is transposed as shown in fig 9. and then the interpolators/upsamplers are introduced after multipliers as shown in fig 10. so that zeros are inserted after multiplications, which results in large savings in calculations.

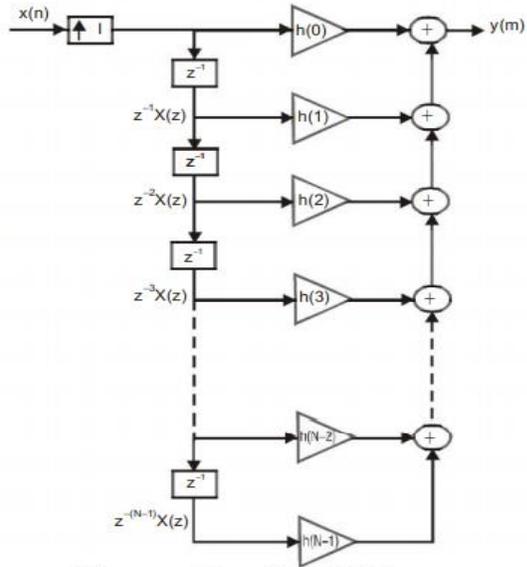


Fig 8 : Direct form FIR filter structure with interpolator at the input.

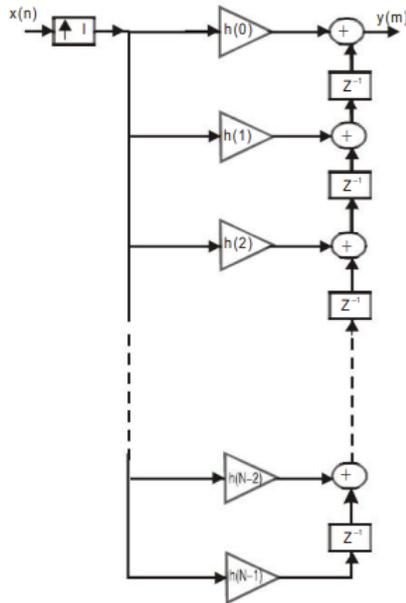


Fig 9 : Transposed direct form, FIR filter structure with interpolator at the input.

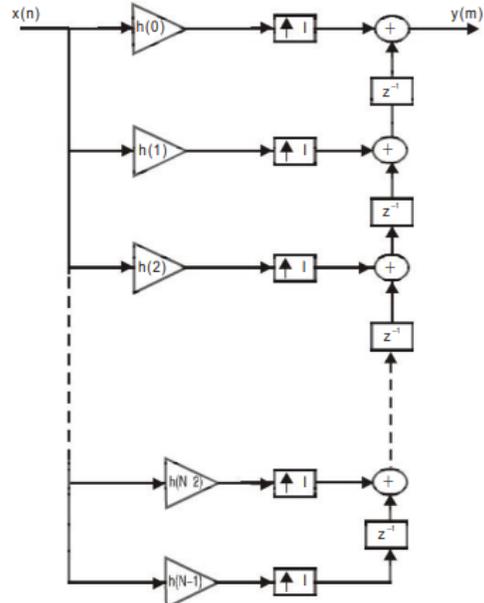


Fig 10 : Efficient structure for direct form FIR filter with interpolator.

5.8 POLYPHASE DECIMATION

The **polyphase decomposition** is dividing an N^{th} order filter into L -sections of sub-filters that can be realized in parallel. In this decomposition the sub-filters will differ only in phase characteristics and so they are called **polyphase filters**. Therefore, the process of dividing the filter into sub-filters is called polyphase decomposition.

The polyphase decomposition of filters leads to a realization structure for filters with reduced computations. Hence, the polyphase decomposition can be applied to (anti-aliasing) filters used for decimators and (anti-imaging) filters used for interpolators in order to realize computationally efficient multirate digital signal processing systems.

In decimator, a lowpass filter called anti-aliasing filter is employed at the input in order to bandlimit the input signal, so that aliasing is avoided in the output spectrum of decimator.

Consider a decimator with sampling rate reduction factor, L . Let, $H(z)$ be the transfer function of lowpass anti-aliasing FIR filter at the input of decimator as shown in fig 11. In order to reduce the computations in FIR filter, polyphase decomposition can be applied to FIR filter to decompose into L sub-filters as shown in fig 12.



Fig 11 : Decimator with anti-aliasing filter.

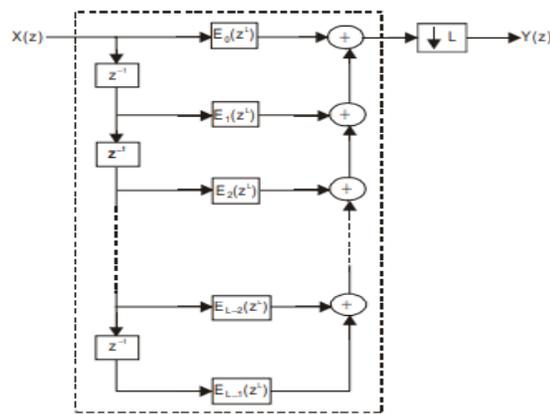


Fig12 : Decimator with type-I decomposition of anti-aliasing filter.

The decimator in the structure of fig 12 will select only one sample in every L samples of the output. Therefore, further reduction in computations can be obtained if the decimator is shifted to input of sub-filters (using identity 15) as shown in fig 13. The structure shown in fig 9.26 is computationally efficient decimator structure.

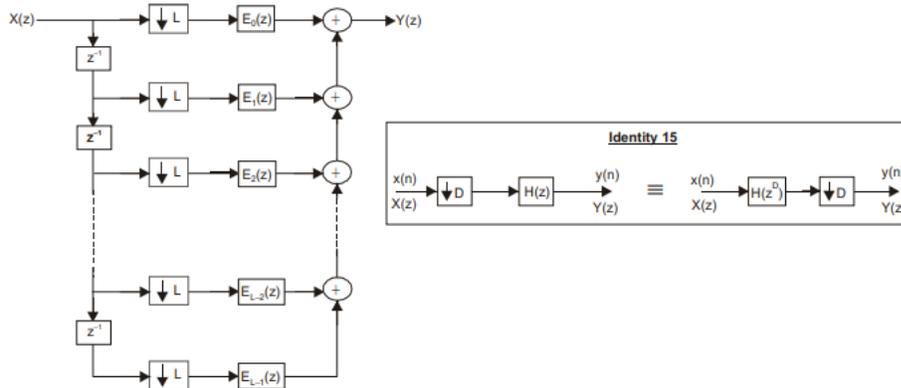


Fig 13 : Computationally efficient decimator structure.

5.9 POLYPHASE INTERPOLATOR

In interpolator, a lowpass filter called anti-imaging filter is employed at the output in order to eliminate the multiple images in the output spectrum of interpolator.

Consider an interpolator with sampling rate multiplication factor, L .

Let, $H(z)$ be the transfer function of lowpass anti-imaging FIR filter at the output of interpolator as shown in fig 14. In order to reduce the computations in FIR filter, polyphase decomposition can be applied to FIR filter to decompose into L sub-filters as shown in fig 15.

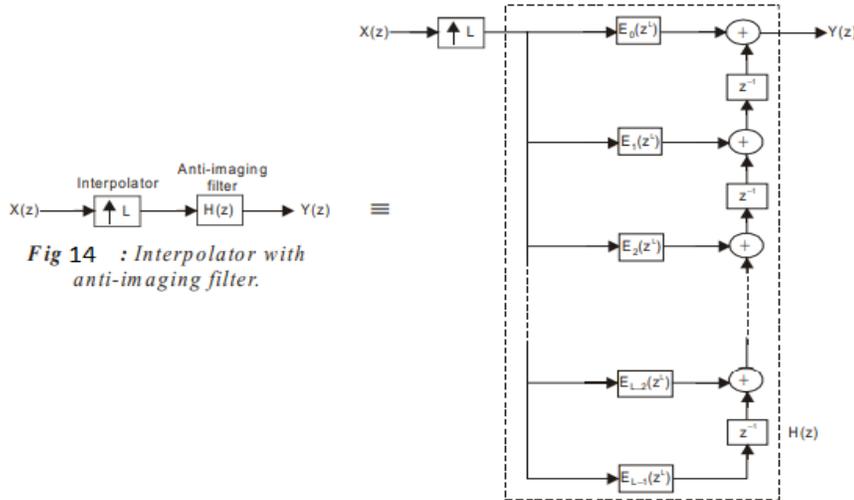


Fig 14 : Interpolator with anti-imaging filter.

Fig 15 : Interpolator with type-I decomposition transposed structure of anti-imaging filter.

The interpolator in the structure of fig 15 will introduce large number of zeros in the input signal so that only one in every L samples will be non-zero. Therefore, further reduction in computations can be obtained if the interpolator is shifted to output of sub-filters (using identity 18) as shown in fig 16.

The structure shown in fig 16 is computationally efficient interpolator structure.

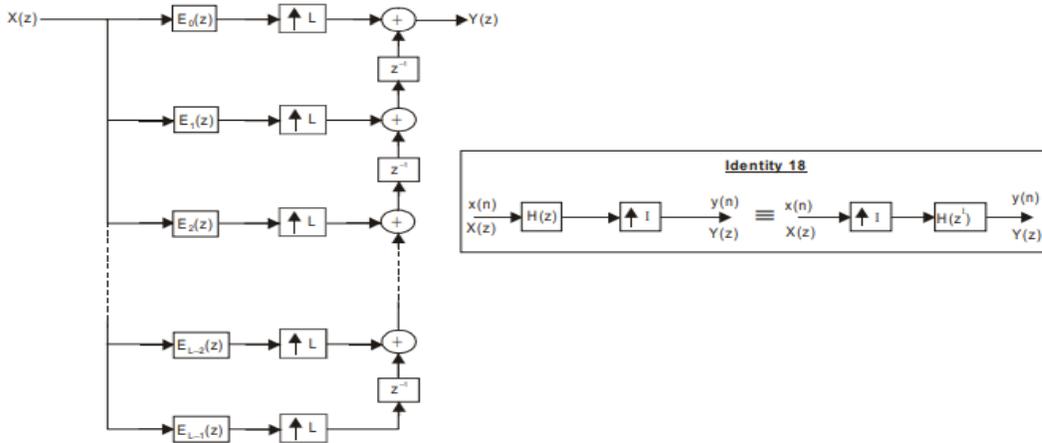


Fig 16 : Computationally efficient interpolator structure.

5.10 EXAMPLES OF BIOMEDICAL SIGNAL PROCESSING

Electrical biosignals, or bioelectrical time signals, usually refers to the change in electric current produced by the sum of an electrical potential difference across a specialized tissue, organ or cell system like the nervous system. Thus, among the best-known bioelectrical signals are:

- Electroencephalogram (EEG)

- Electrocardiogram (ECG)
- Electromyogram (EMG)
- Mechanomyogram (MMG)
- Electrooculography (EOG)
- Galvanic skin response (GSR)
- Magnetoencephalogram (MEG)

EEG, ECG, EOG and EMG are measured with a differential amplifier which registers the difference between two electrodes attached to the skin. However, the galvanic skin response measures electrical resistance and the MEG measures the magnetic field induced by electrical currents (electroencephalogram) of the brain.

With the development of methods for remote measurement of electric fields using new sensor technology, electric biosignals such as EEG and ECG can be measured without electric contact with the skin. This can be applied for example for remote monitoring of brain waves and heart beat of patients who must not be touched, in particular patients with serious burns.

Electrical currents and changes in electrical resistances across tissues can also be measured from plants.

Biosignals may also refer to any non-electrical signal that is capable of being monitored from biological beings, such as mechanical signals (e.g. the mechanomyogram or MMG), acoustic signals (e.g. phonetic and non-phonetic utterances, breathing), chemical signals (e.g. pH, oxygenation) and optical signals

5.10.1 ELECTROENCEPHALOGRAPHY

(**EEG**) is an electrophysiological monitoring method to record electrical activity of the brain. It is typically noninvasive, with the electrodes placed along the scalp, although invasive electrodes are sometimes used in specific applications. EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp. Diagnostic applications generally focus on the spectral content of EEG, that is, the type of neural oscillations (popularly called "brain waves") that can be observed in EEG signals.

EEG is most often used to diagnose epilepsy, which causes abnormalities in EEG readings. It is also used to diagnose sleep disorders, coma, encephalopathies, and brain death. EEG used to be a first-line method of diagnosis for tumors, stroke and other focal brain disorders but this use has decreased with the advent of high-resolution anatomical imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT). Despite limited spatial resolution, EEG continues to be a valuable tool for research and diagnosis, especially when millisecond-range temporal resolution (not possible with CT or MRI) is required.

Derivatives of the EEG technique include evoked potentials (EP), which involves averaging the EEG activity time-locked to the presentation of a stimulus of some sort (visual, somatosensory, or auditory). Event-related potentials (ERPs) refer to averaged EEG responses that are time-

locked to more complex processing of stimuli; this technique is used in cognitive science, cognitive psychology, and psychophysiological research.

5.10.2 ELECTROCARDIOGRAPHY (ECG or EKG*) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. These electrodes detect the tiny electrical changes on the skin that arise from the heart muscle depolarizing during each heartbeat.

In a conventional *12 lead ECG*, ten electrodes are placed on the patient's limbs and on the surface of the chest. The overall magnitude of the heart's electrical potential is then measured from twelve different angles ("leads") and is recorded over a period of time (usually 10 seconds). In this way, the overall magnitude and direction of the heart's electrical depolarization is captured at each moment throughout the cardiac cycle. The graph of *voltage* versus *time* produced by this noninvasive medical procedure is referred to as an **electrocardiogram** (abbreviated *ECG* or *EKG*).

During each heartbeat, a healthy heart will have an orderly progression of depolarization that starts with pacemaker cells in the sinoatrial node, spreads out through the atrium, passes through the atrioventricular node down into the bundle of His and into the Purkinje fibers spreading down and to the left throughout the ventricles. This orderly pattern of depolarization gives rise to the characteristic ECG tracing. To the trained clinician, an ECG conveys a large amount of information about the structure of the heart and the function of its electrical conduction system. Among other things, an ECG can be used to measure the rate and rhythm of heartbeats, the size and position of the heart chambers, the presence of any damage to the heart's muscle cells or conduction system, the effects of cardiac drugs, and the function of implanted pacemakers.

5.10.3 ELECTROMYOGRAPHY (EMG) is an electrodiagnostic medicine technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is performed using an instrument called an **electromyograph**, to produce a record called an **electromyogram**. An electromyograph detects the electrical potential generated by muscle cells^[2] when these cells are electrically or neurologically activated. The signals can be analyzed to detect medical abnormalities, activation level, or recruitment order, or to analyze the biomechanics of human or animal movement.

EMG testing has a variety of clinical and biomedical applications. EMG is used as a diagnostics tool for identifying neuromuscular diseases, or as a research tool for studying kinesiology, and disorders of motor control. EMG signals are sometimes used to guide botulinum toxin or phenol injections into muscles. EMG signals are also used as a control signal for prosthetic devices such as prosthetic hands, arms, and lower limbs.

EMG then acceleromyograph may be used for neuromuscular monitoring in general anesthesia with neuromuscular-blocking drugs, in order to avoid postoperative residual curarization (PORC).

Except in the case of some purely primary myopathic conditions EMG is usually performed with another electrodiagnostic medicine test that measures the conducting function of nerves. This is called a nerve conduction studies (NCS). Needle EMG and NCSs are typically indicated when there is pain in the limbs, weakness from spinal nerve compression, or concern about some other

neurologic injury or disorder. Spinal nerve injury does not cause neck, mid back pain or low back pain, and for this reason, evidence has not shown EMG or NCS to be helpful in diagnosing causes of axial lumbar pain, thoracic pain, or cervical spine pain. Needle EMG may aid with the diagnosis of nerve compression or injury (such as carpal tunnel syndrome), nerve root injury (such as sciatica), and with other problems of the muscles or nerves. Less common medical conditions include amyotrophic lateral sclerosis, myasthenia gravis, and muscular dystrophy.

The **mechanomyogram** (MMG) is the mechanical signal observable from the surface of a muscle when the muscle is contracted. At the onset of muscle contraction, gross changes in the muscle shape cause a large peak in the MMG. Subsequent vibrations are due to oscillations of the muscle fibres at the resonance frequency of the muscle. The mechanomyogram is also known as the phonomyogram, acoustic myogram, sound myogram or vibromyogram

The MMG may provide a useful alternative to the electromyogram (EMG) for monitoring muscle activity. It has a higher signal-to-noise ratio than the surface EMG and thus can be used to monitor muscle activity from deeper muscles without using invasive measurement techniques. It is currently the subject of research activity into prosthetic control and assistive technologies for the disabled.

5.10.4 ELECTROOCULOGRAPHY (EOG) is a technique for measuring the corneo-retinal standing potential that exists between the front and the back of the human eye. The resulting signal is called the electrooculogram. Primary applications are in ophthalmological diagnosis and in recording eye movements. Unlike the electroretinogram, the EOG does not measure response to individual visual stimuli.

To measure eye movement, pairs of electrodes are typically placed either above and below the eye or to the left and right of the eye. If the eye moves from center position toward one of the two electrodes, this electrode "sees" the positive side of the retina and the opposite electrode "sees" the negative side of the retina. Consequently, a potential difference occurs between the electrodes. Assuming that the resting potential is constant, the recorded potential is a measure of the eye's position.

The EOG is used to assess the function of the pigment epithelium. During dark adaptation, resting potential decreases slightly and reaches a minimum ("dark trough") after several minutes. When light is switched on, a substantial increase of the resting potential occurs ("light peak"), which drops off after a few minutes when the retina adapts to the light. The ratio of the voltages (i.e. *light peak* divided by *dark trough*) is known as the *Arden ratio*. In practice, the measurement is similar to eye movement recordings (see above). The patient is asked to switch eye position repeatedly between two points (alternating looking from center to the left and from center to the right). Since these positions are constant, a change in the recorded potential originates from a change in the resting potential.

5.10.5 MAGNETOENCEPHALOGRAPHY (MEG) is a functional neuroimaging technique for mapping brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers. Arrays of SQUIDs (superconducting quantum interference devices) are currently the most common magnetometer, while the SERF (spin exchange relaxation-free) magnetometer is being investigated for future machines.

Applications

of MEG include basic research into perceptual and cognitive brain processes, localizing regions affected by pathology before surgical removal, determining the function of various parts of the brain, and neurofeedback. This can be applied in a clinical setting to find locations of abnormalities as well as in an experimental setting to simply measure brain activity.^[1]

Synchronized neuronal currents induce weak magnetic fields. The brain's magnetic field, measuring at 10 femtotesla (fT) for cortical activity and 10^3 fT for the human alpha rhythm, is considerably smaller than the ambient magnetic noise in an urban environment, which is on the order of 10^8 fT or 0.1 μ T. The essential problem of biomagnetism is, thus, the weakness of the signal relative to the sensitivity of the detectors, and to the competing environmental noise.

Origin of the brain's magnetic field. The electric current also produces the EEG signal.

The MEG (and EEG) signals derive from the net effect of ionic currents flowing in the dendrites of neurons during synaptic transmission. In accordance with Maxwell's equations, any electrical current will produce a magnetic field, and it is this field that is measured. The net currents can be thought of as current dipoles, i.e. currents with a position, orientation, and magnitude, but no spatial extent. According to the right-hand rule, a current dipole gives rise to a magnetic field that points around the axis of its vector component.

To generate a signal that is detectable, approximately 50,000 active neurons are needed.^[6] Since current dipoles must have similar orientations to generate magnetic fields that reinforce each other, it is often the layer of pyramidal cells, which are situated perpendicular to the cortical surface, that gives rise to measurable magnetic fields. Bundles of these neurons that are orientated tangentially to the scalp surface project measurable portions of their magnetic fields outside of the head, and these bundles are typically located in the sulci. Researchers are experimenting with various signal processing methods in the search for methods that detect deep brain (i.e., non-cortical) signal, but no clinically useful method is currently available.

It is worth noting that action potentials do not usually produce an observable field, mainly because the currents associated with action potentials flow in opposite directions and the magnetic fields cancel out. However, action fields have been measured from peripheral nerves.

5.10.6 ESTIMATION OF HEART RATE IN ECG

Heart rate is the speed of the heartbeat measured by the number of contractions of the heart per minute (bpm). The heart rate can vary according to the body's physical needs, including the need to absorb oxygen and excrete carbon dioxide. It is usually equal or close to the pulse measured at any peripheral point. Activities that can provoke change include physical exercise, sleep, anxiety, stress, illness, and ingestion of drugs.

The normal resting adult human heart rate ranges from 60–100 bpm. Tachycardia is a fast heart rate, defined as above 100 bpm at rest.^[2] Bradycardia is a slow heart rate, defined as below 60 bpm at rest. During sleep a slow heartbeat with rates around 40–50 bpm is common and is

considered normal. When the heart is not beating in a regular pattern, this is referred to as an arrhythmia. Abnormalities of heart rate sometimes indicate disease.

The *maximum heart rate* (HR_{max}) is the highest heart rate an individual can achieve without severe problems through exercise stress,^{[11][12]} and generally decreases with age. Since HR_{max} varies by individual, the most accurate way of measuring any single person's HR_{max} is via a cardiac stress test. In this test, a person is subjected to controlled physiologic stress (generally by treadmill) while being monitored by an ECG. The intensity of exercise is periodically increased until certain changes in heart function are detected on the ECG monitor, at which point the subject is directed to stop. Typical duration of the test ranges ten to twenty minutes.

Adults who are beginning a new exercise regimen are often advised to perform this test only in the presence of medical staff due to risks associated with high heart rates. For general purposes, a formula is often employed to estimate a person's maximum heart rate. However, these predictive formulas have been criticized as inaccurate because they generalized population-averages and usually focus on a person's age. It is well-established that there is a "poor relationship between maximal heart rate and age" and large standard deviations relative to predicted heart rates

The human heart beats more than 3.5 billion times in an average lifetime.

The heartbeat of a human embryo begins at approximately 21 days after conception, or five weeks after the last normal menstrual period (LMP), which is the date normally used to date pregnancy in the medical community. The electrical depolarizations that trigger cardiac myocytes to contract arise spontaneously within the myocyte itself. The heartbeat is initiated in the pacemaker regions and spreads to the rest of the heart through a conduction pathway. Pacemaker cells develop in the primitive atrium and the sinus venosus to form the sinoatrial node and the atrioventricular node respectively. Conductive cells develop the bundle of His and carry the depolarization into the lower heart.

The human heart begins beating at a rate near the mother's, about 75-80 beats per minute (BPM). The embryonic heart rate then accelerates linearly for the first month of beating, peaking at 165-185 BPM during the early 7th week, (early 9th week after the LMP). This acceleration is approximately 3.3 BPM per day, or about 10 BPM every three days, an increase of 100 BPM in the first month.

After peaking at about 9.2 weeks after the LMP, it decelerates to about 150 BPM (+/-25 BPM) during the 15th week after the LMP. After the 15th week the deceleration slows reaching an average rate of about 145 (+/-25 BPM) BPM at term. The regression formula which describes this acceleration before the embryo reaches 25 mm in crown-rump length or 9.2 LMP weeks is:

$$\text{Age in days} = \text{EHR}(0.3) + 6$$

There is no difference in male and female heart rates before birth

The **QRS complex** is a name for the combination of three of the graphical deflections seen on a typical electrocardiogram (EKG or ECG). It is usually the central and most visually obvious part

of the tracing. It corresponds to the depolarization of the right and left ventricles of the human heart. In adults, it normally lasts 0.06–0.10 s; in children and during physical activity, it may be shorter. The Q, R, and S waves occur in rapid succession, do not all appear in all leads, and reflect a single event, and thus are usually considered together. A **Q wave** is any downward deflection after the **P wave**. An **R wave** follows as an upward deflection, and the **S wave** is any downward deflection after the R wave. The **T wave** follows the S wave, and in some cases an additional **U wave** follows the T wave.

Q wave

Normal Q waves, when present, represent depolarization of the interventricular septum. For this reason, they are referred to as septal Q waves and can be appreciated in the lateral leads I, aVL, V5 and V6.

Pathologic Q waves occur when the electrical signal passes through stunned or scarred heart muscle; as such, they are usually markers of previous myocardial infarctions, with subsequent fibrosis. A pathologic Q wave is defined as having a deflection amplitude of 25% or more of the subsequent R wave, or being > 0.04 s (40 ms) in width and > 2 mm in amplitude. However, diagnosis requires the presence of this pattern in more than one corresponding lead.

Myocardial infarctions with pathological Q waves are referred to as ST elevation MIs.

R wave progression

Looking at the precordial leads, the r wave usually progresses from showing a rS-type complex in V₁ with an increasing R and a decreasing S wave when moving towards the left side. There is usually an qR-type of complex in V₅ and V₆ with the R-wave amplitude usually taller in V₅ than in V₆. It is normal to have a narrow QS and rSr' patterns in V₁, and so is also the case for qRs and R patterns in V₅ and V₆. The *transition zone* is where the QRS complex changes from predominately negative to predominately positive (R/S ratio becoming >1), and this usually occurs at V₃ or V₄. It is normal to have the transition zone at V₂ (called "early transition"), and at V₅ (called "delayed transition"). In biomedical engineering, the maximum amplitude in the R wave is usually called "R peak amplitude", or just "R peak".^{[7][8]} Accurate R peak detection is essential in signal processing equipment for heart rate measurement and it is the main feature used for arrhythmia detection.

The definition of *poor R wave progression* (PRWP) varies in the literature, but a common one is when the R wave is less than 2–4 mm in leads V₃ or V₄ and/or there is presence of a reversed R wave progression, which is defined as R in V₄ $<$ R in V₃ or R in V₃ $<$ R in V₂ or R in V₂ $<$ R in V₁, or any combination of these.^[6] *Poor R wave progression* is commonly attributed to anterior myocardial infarction, but it may also be caused by left bundle branch block, Wolff–Parkinson–White syndrome, right and left ventricular hypertrophy as well as by faulty ECG recording technique.

J-point

The point where the QRS complex meets the ST segment is the J-point. The J-point is easy to identify when the ST segment is horizontal and forms a sharp angle with the last part of the QRS complex. However, when the ST segment is sloped or the QRS complex is wide, the two features do not form a sharp angle and the location of the J-point is less clear. There is no consensus on the precise location of the J-point in these circumstances. Two possible definitions are:

5. 11ANALYSIS OF EMG SIGNAL

Electromyography (EMG) is an electrodiagnostic medicine technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is performed using an instrument called an **electromyograph**, to produce a record called an **electromyogram**. An electromyograph detects the electrical potential generated by muscle cells when these cells are electrically or neurologically activated. The signals can be analyzed to detect medical abnormalities, activation level, or recruitment order, or to analyze the biomechanics of human or animal movement.

One basic function of EMG is to see how well a muscle can be activated. The most common way that can be determined is by performing a maximal voluntary contraction (MVC) of the muscle that is being tested.

Muscle force, which is measured mechanically, typically correlates highly with measures of EMG activation of muscle. Most commonly this is assessed with surface electrodes, but it should be recognized that these typically only record from muscle fibers in close approximation to the surface.

Several analytical methods for determining muscle activation are commonly used depending on the application. The use of mean EMG activation or the peak contraction value is a debated topic. Most studies commonly use the maximal voluntary contraction as a means of analyzing peak force and force generated by target muscles. According to the article, Peak and average rectified EMG measures: Which method of data reduction should be used for assessing core training exercises?, concluded that the “average rectified EMG data (ARV) is significantly less variable when measuring the muscle activity of the core musculature compared to the peak EMG variable.” Therefore, these researchers would suggest that “ARV EMG data should be recorded alongside the peak EMG measure when assessing core exercises.” Providing the reader with both sets of data would result in enhanced validity of the study and potentially eradicate the contradictions within the research.

EMG can also be used for indicating the amount of fatigue in a muscle. The following changes in the EMG signal can signify muscle fatigue: an increase in the mean absolute value of the signal, increase in the amplitude and duration of the muscle action potential and an overall shift to lower frequencies. Monitoring the changes of different frequency changes the most common way of using EMG to determine levels of fatigue. The lower conduction velocities enable the slower motor neurons to remain active.

A motor unit is defined as one motor neuron and all of the muscle fibers it

innervates. When a motor unit fires, the impulse (called an action potential) is carried down the motor neuron to the muscle. The area where the nerve contacts the muscle is called the neuromuscular junction, or the motor end plate. After the action potential is transmitted across the neuromuscular junction, an action potential is elicited in all of the innervated muscle fibers of that particular motor unit. The sum of all this electrical activity is known as a motor unit action potential (MUAP). This electrophysiologic activity from multiple motor units is the signal typically evaluated during an EMG. The composition of the motor unit, the number of muscle fibres per motor unit, the metabolic type of muscle fibres and many other factors affect the shape of the motor unit potentials in the myogram.

Nerve conduction testing is also often done at the same time as an EMG to diagnose neurological diseases.

Some patients can find the procedure somewhat painful, whereas others experience only a small amount of discomfort when the needle is inserted. The muscle or muscles being tested may be slightly sore for a day or two after the procedure.