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 1 - INTRODUCTION TO BIOMEDICAL SIGNALS – SBMA5202

## UNIT 1 - INTRODUCTION TO BIOMEDICAL SIGNALS

### 1.1 Bioelectric signals

Living organisms are made up of many component systems: the human body includes several systems. Each system is made up of several subsystems that carry on many physiological processes.

Cardiac system: rhythmic pumping of blood throughout the body to facilitate the delivery of nutrients, and pumping blood through the pulmonary system for oxygenation of the blood itself.

Physiological processes are complex phenomena, including nervous or hormonal stimulation and control; inputs and outputs that could be in the form of physical action that could be biochemical material, neurotransmitters, or information;

Most physiological processes are accompanied by signals of several types that reflect their mechanical, electrical, or nature and activities:

biochemical, in the form of hormones and neurotransmitters, electrical, in the form of potential or current, and physical, in the form of pressure or temperature Some examples of bioelectric signals are ElectroCardioGram

ElectroNeuroGram

ElectroOculoGram

ElectroEncephaloGram

ElectroGastroGram

#### 1.1.1 Action potential

Many cells in the body, and in particular those associated with nerve and muscle fibres, can be excited either electrically or chemically. An electrochemical stimulus can induce changes in the permeability of the cell membrane to different ions and cause the cell to become active. This means that the flow of ions across the cell membrane changes abruptly and hence also the volume of charge on each side of the membrane. This is accompanied by a corresponding abrupt change in the trans-membrane potential so that the cell becomes depolarised, sometimes having a slight change in the potential in the opposite direction to its equilibrium state. The cell will eventually repolarise but usually at a slower rate than that at which it depolarises.

Once a cell becomes depolarised, the changes in the conditions surrounding the cell can act as a stimulus to adjacent cells and thereby a corresponding activation of these cells takes place

also. In nerve and muscle cells the impulse generated by depolarisation of the cells can be passed from one cell to the next via axons and synapses, so that the stimulus passes along a nerve or muscle fibre as a wave with a repolarisation wave following behind.

A cell in its resting state is said to be polarized. Most cells maintain a resting potential of the order of −60 to −100 mV until some disturbance or stimulus upsets the equilibrium. When a cell is excited by ionic currents or an external stimulus, the membrane changes its characteristics. It begins to allow Na+ ions to enter the cell.This movement of Na+ ions constitutes an ionic current, which further reduces the membrane barrier to Na+ ions.

K+ ions try to leave the cell as they were in higher concentration inside the cell in the preceding resting state, but cannot move as fast as the Na+ ions. Net result is the inside of the cell becomes positive

with respect to the outside due to an imbalance of K+. New state of equilibrium is reached after the rush of Na+ ions stops. It represents the beginning of the action potential, with a peak value of about +20 mV for most cells. An excited cell displaying an action potential is said to be depolarized, this process is called depolarization.

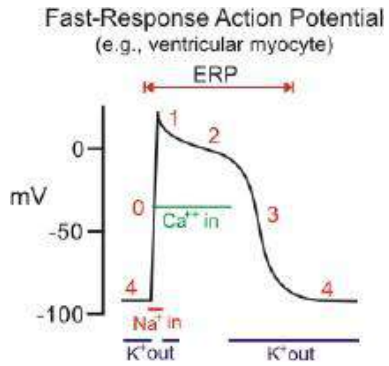



**Fig. 1.1 Action Potential**

After a certain period of being in the depolarized state the cell becomes polarized again and returns to its resting potential via a process known as repolarization. Principal ions involved in repolarization are K+. Voltage dependent K+ channels changes membrane permeability for K+. K+ concentration is much higher inside the cell: net efflux of K+ from the cell,the inside becomes more negative, effecting repolarization back to the resting potential. Nerve and muscle cells repolarize rapidly: action potential duration of about 1 ms. Heart muscle cells repolarize slowly: action potential duration of 150 − 300 ms

## 1.2 Electro-Neurogram

The ENG is an electrical signal observed as a stimulus and the associated nerve action potential propagate over the length of nerve. ENGs may be recorded using contcentric needle electrodes or Ag-AgCl electrodes at the surface of the body. In order to minimize muscle contraction strong but short stimulus is applied (100 V amplitude, 100-300 μs). ENGs have amplitudes of the order of 10 μV. Conduction velocity in a peripheral nerve measured by stimulating a motor nerve and measuring the related activity at two points at known distances along its course.

Stimulus: 100 V , 100 − 300μs. ENG amplitude: 10 μV ;

Amplifier gain: 2, 000; Bandwidth 10 − 10, 000 Hz.

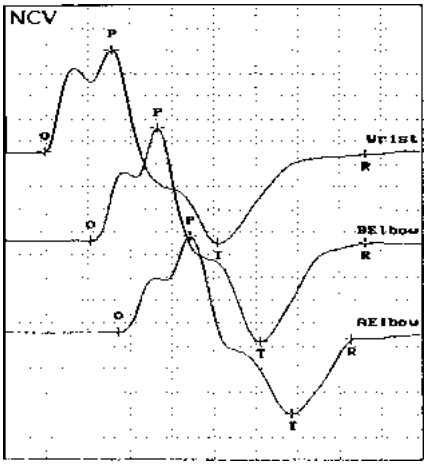



**Fig. 1.2 Nerve conduction velocity**

Typical nerve conduction velocity:

45 − 70 m/s in nerve fibers;

0.2 − 0.4 m/s in heart muscle;

0.03 − 0.05 m/s in timedelay fibers between the atria and ventricles. Neural diseases may cause a decrease in conduction velocity.

## 1.3 Electro-oculogram

An electrooculogram is a signal that can be used for measuring the resting potential of the retina in the eye. The human eye is polarized, with the front of the eye being positive and the back of the eye being negative. This is caused by a concentration of negatively charged nerves in the retina on the back of the eye. As the eye moves the negative pole moves relative to the

face and this change in the dipole potential can be measured on the skin in micro volts. To translate this voltage into a position, two sets of electrodes are used to measure the differential voltage in the vertical and horizontal direction.

There are four different types of conjugate eye movements. These eye movements fall into two specific categories:

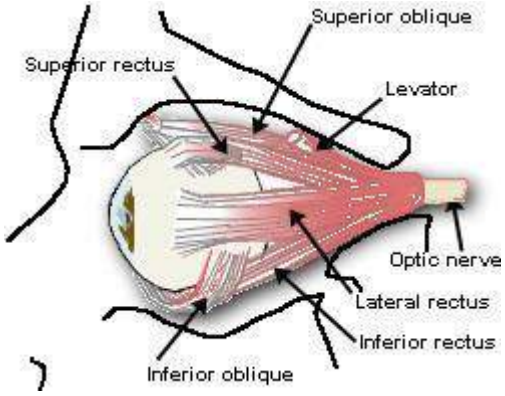



**Fig. 1.3 Structure of eye**

1.4 Eye movements that function to stabilize the position of the eye in space during head movements (Reflex eye movements).

1.5 Eye movements that function to redirect the line of sight to follow a moving target or to attend to a new target of interest (Voluntary eye movements).

The electrical potentials are generated by the permanenet potential difference which exists between the the cornea and the ocular fundus (cornea-retinal potential, 10-30mV: the cornea being positive).

This potential difference sets up an electrical field in the tissues surrounding the eye. As the eye rotates, the field vector rotates correspondingly. Therefore, eye movements can be detected by placing electrodes on the skin in the area of the head around the eyes. Vertical movements of the eyes are best measured by placing the electrodes on the lids, while horizontal eye movements can be best measured by placing the electrodes on the external canthi (the bone on the side of the eye).

The underlying assumption of this method of recording eye movements is that the movement of the electric field in the conducting tissues surrounding the eye is related, in a simple (usually assumed to be linear) way to the movements of the eye itself. Due to the non-uniformity of these tissues and the shapes of the tissues surrounding them, this can only be an approximation to the biological reality. However, for horizontal eye movements within the range of 30 degrees, the potential measured is assumed to be linear to the actual movement of the eye in the orbit. The resolution of EOG is considered to be about 1 degree. Because it is a relatively simple technique, EOG is still commonly used clinically for testing eye movements in patients.

For a fixed eye position, the EOG is far from being constant in magnitude, but can be influenced by a number of external factors. These factors include

1.5.1 The noise generated between the electrodes' contacts and the skin

1.5.2 The metabolic state of the tissues (pO2, pCO2, and temperature)

1.5.3 Visual stimulation

1.5.4 Contraction of facial muscles

In addition, recorded EOG, particularly for vertical eye movements, is quite sensitive to movements of the eye lids. In summary there are a number of external factors which can complicate the interpretation of the EOG, and for that reason EOG is considered highly sensitive to artifacts. The considerable artifacts which can be introduced through the contact between the electrode contacts and the skin can be minimized by reducing the resistance between the electrodes and the skin.

**1.4 Electro-encephalogram**

EEG or brainwaves represent the electricalactivityofthebrain.

Main parts ofthe brain are : Zerebrum,

cerebellum,

brainstem(midbrain,ponsmedulla,

reticularformation),

thalamus

(betweenthe midbrainandthehemispheres).

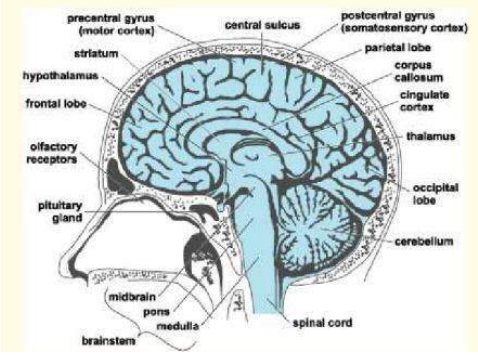



**Fig. 1.4 Structure of Human Brain**

Cerebrum is divided into two hemispheres which is separated by a longitudinal fissure with a large connective band of fibers:corpus callosum. Outer surface of the cerebral hemispheres(cerebralcortex) compose of neurons(greymatter) in convoluted patterns, separated into regions by fissures(sulci). Beneath the cortex lie nerve fibers that lead to other parts of the brain and the body (whitematter). Cortical potentials generated due to excitatory and inhibitory post-synaptic potentials are developed by cell bodies and dendrites of pyramidal neurons. Physiological control processes, thought processes, and external stimuli generate signals in the corresponding parts of the brain.

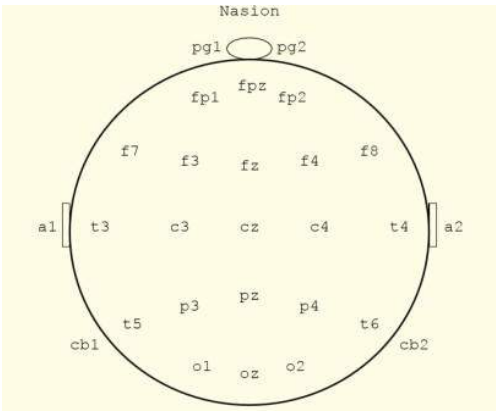



**Fig.1.5 10-20 Electrode System**

Special EEG techniques are:

Needleelectrodes,

Naso-pharyngealelectrodes,

Electrocorticogram(ECoG)fromexposedcortex,

Intracerebralelectrodes

Evocative techniques for recording the EEG are given below:

Initial recording at rest (eyesopen,eyesclosed), hyperventilation (after breathing at 20 respirations per minute for 2–4 minutes), photic stimulation (with 1–50 flashes of light per second), auditory stimulation with loud clicks, sleep (different stages),and pharmaceuticals or drugs

EEG rhythms or frequency bands are classified as:

Delta(δ): $0.5 \leq f < 4$ Hz; Theta(θ): $4 \leq f < 8$ Hz; Alpha(α): $8 \leq f \leq 13$ Hz;and Beta(β): $f > 13$ Hz.

EEG rhythms are associated with physiological and mental processes. Alpha wave represents principal resting rhythm of the brain. It is common in wakeful, resting adults, especially in the occipital area with bilateral synchrony. Auditory and mental arithmetic tasks with the eyes closed lead to strong alpha waves. These are suppressed when the eyes are opened. Alpha wave is replaced by slower rhythms at various stages of sleep. Theta waves represents the beginning stages of sleep. Delta waves represents deep-sleep stages. High-frequency beta waves shows the background activity in tense and anxious subjects.

Spikes and sharp waves represents epileptogenic regions.

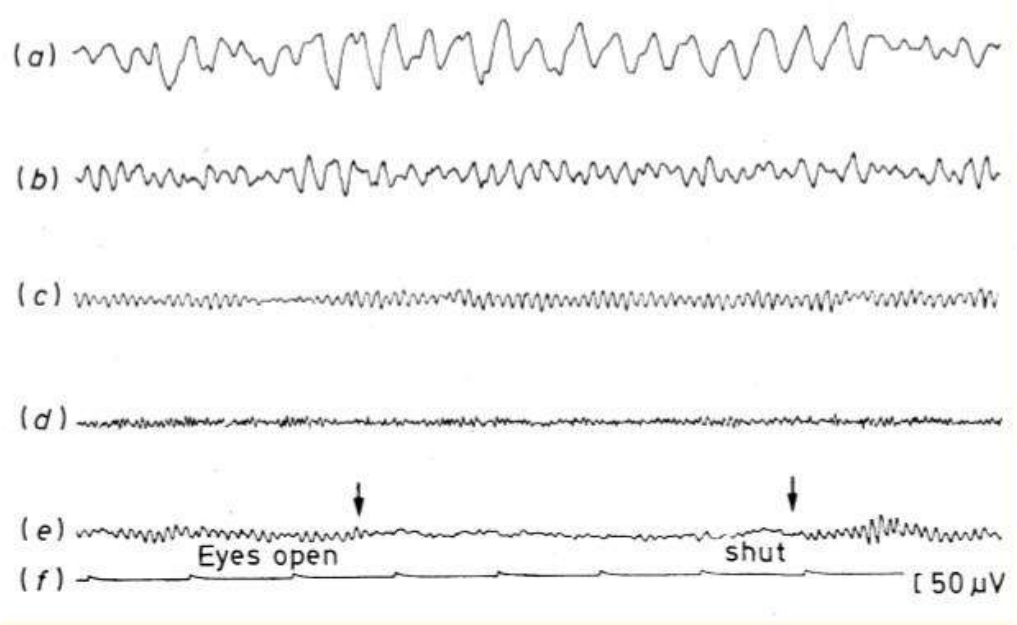



**Fig 6 a) delta rhythm b) theta rhythm c) alpha rhythm d) beta rhythm e) blocking of alpha rhythm by eye opening f) 1s time marker and 50µV**

### 1.4.1 Evoked Potential

Event-related potential is known as evoked potential. It includes the ENG or the EEG in response to light, sound, electrical, or other external stimuli.Short-latency ERPs dependent

upon the physical characteristics of the stimulus. Longer-latency ERPs are influenced by the conditions of presentation of the stimuli. Somato sensory evoked potentials are useful forn on invasive evaluation of the nervoussystem from a peripheral receptor to the cerebral cortex. Median nerveshort-latency SEPs are obtained by placing stimulating electrodes 2−3 cm apart over the median nerve at the wrist with electrical stimulation at 5−10 pps, each stimulus pulse less than 0.5 ms,about 100 V (producingavisiblethumbtwitch). SEPs recorded from the surface of the scalp. Latency, duration, and amplitude of the response measured.

**1.5 Electro-cardiogram**

The cardiovascular system of the human body is essentially one of the heart acting as a pump to force blood around the body. The blood acts as a transport system to carry oxygen, nutrients and chemical agents to all organs, limbs and tissue in the body as well as to transport waste products and toxins to organs for disposal. In fact, the heart actually operates as a double pump and the circulatory system consists of two separate circuits as shown in Fig.7. The heart has four chambers, the left and right atria on top and the left and right ventricles on the bottom.

Blood is gathered from all parts of the body into the right atrium, from whence it is then transferred to the right ventricle. The right ventricle contracts to force blood out to the lungs where carbon dioxide is removed from it and fresh oxygen is absorbed. From the lungs the re-oxygenated blood travels back to the heart and into the left atrium. This loop is called the pulmonary circulation. Blood is then transferred to the right ventricle, which contracts with strength to force the blood out under pressure to all limbs and organs in the body. Once oxygen and nutrients have been distributed via the blood to nourish all of the cells around the body and waste products have been collected and delivered for excretion, the blood returns to the right atrium again. The continuous rhythmic pumping of the heart is caused by contractions of the muscles within the walls of each chamber which pumps blood from chamber to chamber and throughout the circulatory system. These cardiac rhythms are controlled by specific mechanisms operating within in the heart that transmit action potentials or electrical impulses along nerve fibres to the cells within the muscles in order to activate them at the appropriate points in the cardiac cycle.

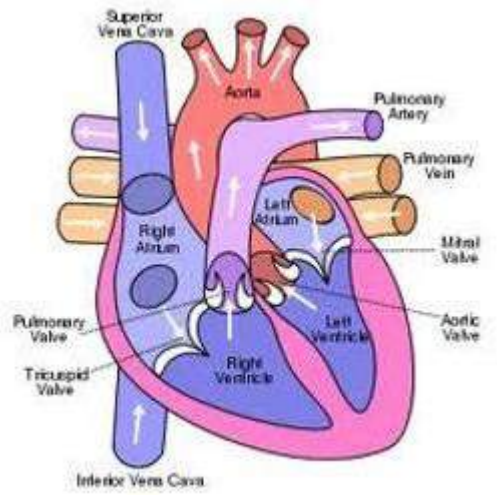



**Fig. 1.7 Human Heart**

**Electro-stimulation of the Heart**

Figure shows the main elements of the heart's electro-conduction system. The sino-atrial (SA) node is a group of cells located in the upper right atrium. This node contains special

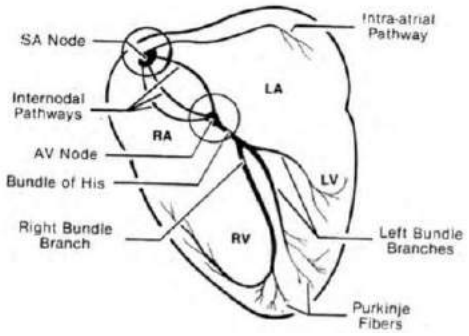



**Fig. 1.8 Electro-conduction system of Heart**

Electro chemically stimulated cells which depolarise and repolarise rhythmically without the need for external influence.

Once the trans-membrane potential in the cell reaches a certain threshold, the cell self-depolarises giving rise to an associated action potential, and then repolarises more gradually. It does this in a continuous and rhythmical manner, thus effectively providing the electrical oscillator which repeatedly generates the trigger stimulus to operate the nerve fibres of the heart and the muscles of the chambers to maintain a regular heartbeat. When the SA node 'fires', the resulting electrochemical stimulus spreads across the muscles in the walls of the right and left atria causing them to contract. Blood is consequently forced out of the atria and into the lower ventricles on both sides of the heart. The stimulus moves quickly from the sino-atrial node towards the atrio-ventricular (AV) node in approximately 30 - 50 ms. In order to

allow the atria transfer their contents to the ventricles before the latter contract due to the approaching action potential, the AV node operates as a delay unit slowing down the transmission of the action potential by a further 110 ms before the stimulus is passed on by the AV node. The impulse is then transferred from the AV node towards the ventricles via a branch of fibres known as the Bundle of His which splits into left and right bundle branches.

Once the impulse reaches the left and right bundle branches it travels very quickly via the Purkinje Fibres which excite the muscles in the walls of the ventricles from the bottom upwards. The impulse can reach the furthest fibres just 60 ms after leaving the AV node. The action potential now causes ventricular contraction which forces the blood from the ventricles out into the pulmonary and systemic circulations. The excitation of such a large number of cells at the same time creates a significant electrical signal and a resulting electric field which is emitted outward from the heart to the surface of the body. These emanating electric signals can be detected using electrodes placed on the surface of the body i.e. on the subject's chest or limbs. The recorded electrical signal detected in this manner is what is known today as the Electrocardiogram or ECG signal.

An idealised human ECG is shown in Fig. 9. It can be seen that there are several distinct components which make up the entire signal profile that is measured over a single complete cardiac cycle. The main components are identified as the P-wave, The QRS complex and the T-wave. Other segments and intervals which have a clinical importance from a diagnostic are defined. The amplitude of the QRS complex of a signal measured on a subject’s chest is typically between 1 – 5 mV.

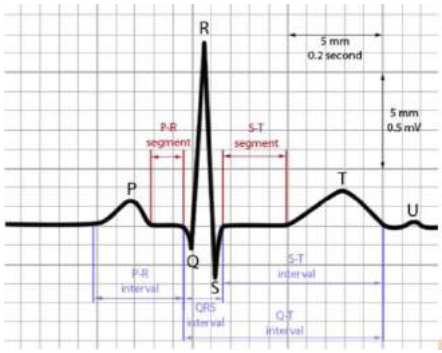



**Fig.1.9 Ideal ECG Signal**

The different components of the ECG correspond to different events occurring in the heart over a cardiac cycle. The P wave is associated with the depolarisation of the cells in the muscles of the atria which cause the atria to contract and transfer blood to the ventricles. The QRS complex corresponds to the sharp depolarisation of the cells in the numerous and strong

muscles of the ventricles. This period is known as ventricular systole. The repolarisation of the cells in the atrial muscles is masked by the QRS complex and cannot be observed independently. The T-wave corresponds to the repolarisation of the cells in the ventricular muscles during their resting phase known as ventricular diastole. The duration, shape and rhythm of these components and of the segments between them can provide invaluable insight into the state of the heart and the cardiovascular system. Einthoven developed the Leads known as the Einthoven triangle comprising Lead I, II and III as indicated which represent different pairings of the electrodes, each providing a different aspect of the electrical activity in the heart.

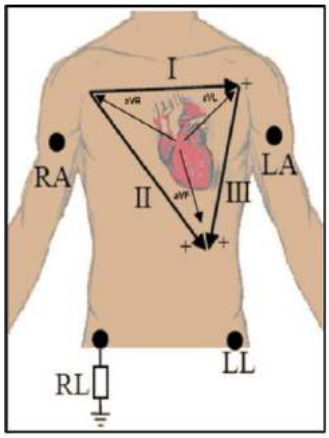



**Fig. 1.10 Einthovan Triangle**

These lead configurations are measured as follows:

Lead I = LA – RA Lead II = LL – RA Lead III = LL - LA

By averaging the potential measured at the three main locations and using the resultant as a new reference, three additional lead configurations known as Augmented Leads can be obtained.

### 1.6 Electro-gastrogram

EGG, similar to an electorcardiogram (EKG) of the heart, records the electrical signals that travel through the muscles of the stomach controlling the muscles' contractions. Additionally, EGG measures stomach wall nerve activity before and after food ingestion. EGG has existed in the past, but was cumbersome, recording only one electrical channel at a time. Therefore there was little capability for muscle activity comparison and in order to record electrical signals in various parts of the stomach needed to be repeated numerous times. The new EGG

system records on four channels simultaneously allowing for complete activity recording in a short time frame.

EGG is a non-invasive test, relatively inexpensive, and easy to perform. Electrodes are placed cutaniously on the abdominal skin over the stomach. While the patient is lying down relaxing, the electrodes record the electrical activity of the stomach. Initially, the gastric electrical activity is recorded after fasting, then again after a small meal is ingested. Sometimes EGG is done in conjunction with or after gastric emptying studies to diagnose and manage functional dyspepsia and idiopathic gastroparesis. The EGG test lasts approximately one hour.

EGG is an appropriate diagnostic tool when there is a suspicion that the nerves controlling stomach muscles or the stomach muscles themselves are not working normally. EGG can be used for a variety of gastrointestinal motility disorders or for patients with no known GI disorder who are suffering from unexplained nausea. EGG frequently identifies dysrhythmias, especially after meals, in patients with gastroparesis, chronic dyspepsia, anorexia nervosa and bulimia, cyclic vomiting syndrome, and other conditions characterized by a delayed gastric emptying.

Gastric myoelectric activity is composed of mainly two complementary rhythms; slow wave activity, responsible for muscle contraction timing, and electrical response activity, responsible for triggering peristaltic contractions. EGG measures too much activity, tachygastria, too little activity, bradygastria, or mixed dysfunction of both too much and too little activity.

Utilizing computer analysis, the power of the stomach muscle electrical current is measured. In a normal stomach muscle, the regular electrical rhythm generates an increased current after a meal. In persons with stomach muscle or nerve irregularities, the post-meal electrical rhythm is irregular or voltage does not increase.

**1.7 Bio-impedance signals**

Bioimpedance is a physiological property related to a tissue's resistance to electrical current flow and its ability to storeelectricalcharge. Ininvivohumanapplications,itis typically measured through metallic electrodes (transducers) placed on the skin and around an anatomic location of interest (e.g., the wrist). These electrical properties are predominantly a function of the underlying tissue being gauged, including the specific tissue types present (blood, adipose, muscle, bone, etc.), the anatomic configuration (i.e., bone or muscle orientation and quantity), and the state of the tissue (normal or osteoporotic bone, edematous vs. normally hydrated tissue, etc). Significant impedance differences exist between the varying tissue types, anatomic configurations, and tissue state, each of which may provide a unique mechanism for

distinguishing between people.

Bioimpedance can be measured by applying a small sinusoidal current between a pair of electrodes attached to the skin. The injected current establishes an electrical field within the tissue and results in a measurable voltage difference between the two electrodes. Thus, potential voltage difference is a function of the underlying tissue impedance. Specifically, the alternating current version of Ohm's law, V = IZ, can be used to relate the voltage V and current I to the bioimpedance Z of the tissue sample. Many tissues exhibit dispersive characteristics, meaning that their electrical properties are dependent on the frequency at which they are measured. Typically, the frequency of the alternating current is swept over a specific band and enables so- called electrical impedance spectroscopy.

As a result, complex bioimpedance, $Z(\omega)$, combines resistive and reactive components, $Z(\omega) = R(\omega) + jX(\omega)$, where R is the frequency dependent tissue resistance, X is the frequency dependent tissue reactance, $\omega$ is the signal frequency, and j represents the imaginary quantity $\sqrt{-1}$.

Resistance and reactance are dependent on the tissue being measured and the configuration and geometry of the impedance-measuring probe (i.e., electrode size and electrode spacing). In terms of its dependence on the tissue, resistance is primarily associated with the ability of a tissue to carry charge (i.e., current flow through ionic solutions, both intra- and extra-cellular), and reactance is associated with the ability of a tissue to store charge (i.e., the capacitive nature of a cell's double membrane). The anatomy of the forearm proximal to the wrist include skeletal bones (radius and ulna), arteries, veins, nerves, muscles, adipose, skin, and interstitial fluids. Over the frequency range of 10 kHz to 10MHz reported values of bone conductivity and adipose conductivity are relatively stable. In muscle, skin, and blood, however, the conductivity monotonically increases with frequency. Person-to-person differences at the wrist include: size, skin thickness, skin water content, bony anatomy (bone sizes), vascular branch size and locations, sub-dermal water content, and adipose/muscle/bone/vasculature content within the sensing region. All of these parameters will have an impact on the actual impedance measured at the wrist. For example, difference in wrist size would represent a change in electrode location and difference in the content, size, and distribution of the underlying tissue types would represent a person-specific conductivity.

## 1.8 Objectives of biomedical signal analysis

Information Gathering —measurement of phenomena to interpret a system**.**

Diagnosis —detection of malfunction, pathology, or abnormality.

Monitoring —obtaining continuous or periodic information about a system.

Therapyandcontrol —modification of th behavior of a system based upon the outcome of the activities listed above to ensure a specific result.

Evaluation —objective analysis to determine the ability to meet functional requirements, obtain proof of performance, perform quality control, or quantify the effect of treatment.

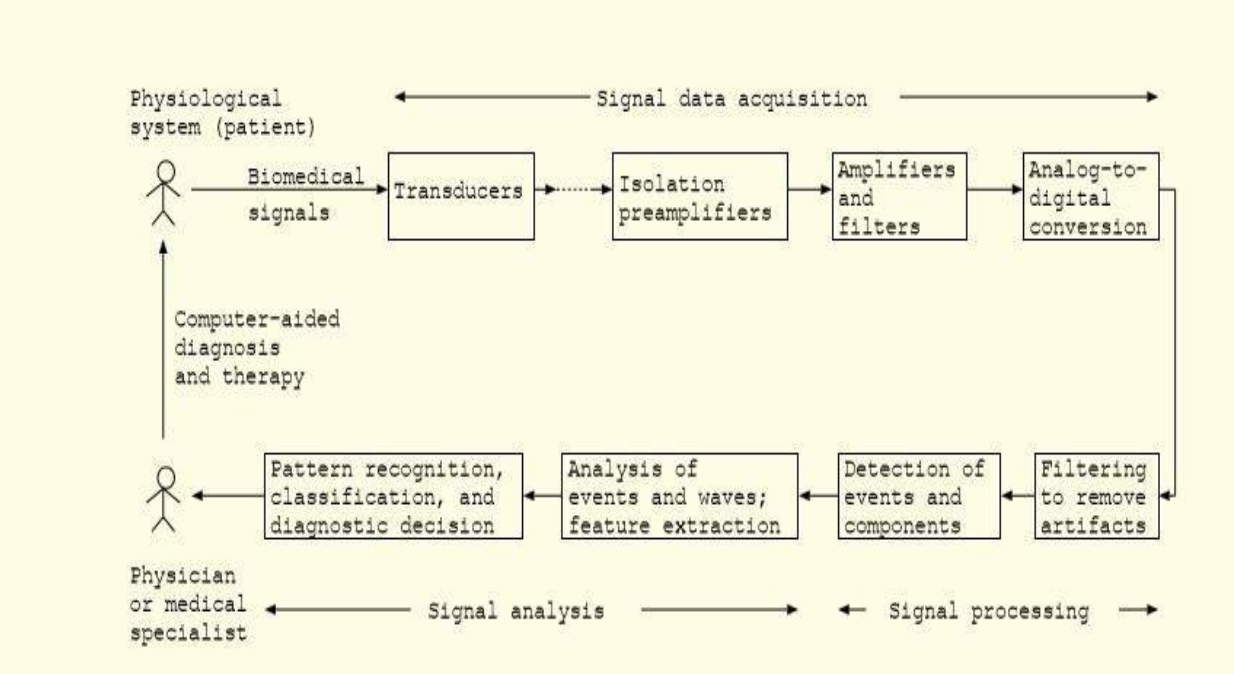



**Fig.1.11 Computer-aided diagnosis and therapy based upon biomedical signal analysis**

## 1.9 **Difficulties in biomedical signal analysis**

Accessibility of the variables to measurement. Variability of the signal source.

Inter-relationships and interactions among physiological systems.

Effects of the instrumentation or procedure on the system.

Physiological artifacts and interference. Energy limitations.

Patient safety.

**Questions:**

1.Explain in detailed about the bioelectric signal which give the functions of Heart (ECG)

2. Discuss about the bioelectric signal which is responsible for the electrical activity of Neurons (ENG)

3. Discuss about the bioelectric signal which is responsible for the electrical activity of Brain (EEG)

4. Explain in detailed about the bioelectric signal which give details of the functions of Eye (EOG)

5. Discuss about the bioelectric signal which is responsible for the electrical activity of Nerves and muscles of the stomach (EGG)

6. There are some difficulties in analysis of the bio electric signal, Discuss the problem in detailed also mention about the term action potential and Evoked potential.

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 2 – REAL TIME TRANSFORMS – SBMA5202

## UNIT 2 - REAL TIME TRANSFORMS

### 2.1 Convolution

Convolution is a mathematical way of combining two signals to form a third signal. It is the single most important technique in Digital Signal Processing. Using the strategy of impulse decomposition, systems are described by a signal called the *impulse response*. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.

First, the input signal can be decomposed into a set of impulses, each of which can be viewed as a scaled and shifted delta function. Second, the output resulting from each impulse is a scaled and shifted version of the impulse response. Third, the overall output

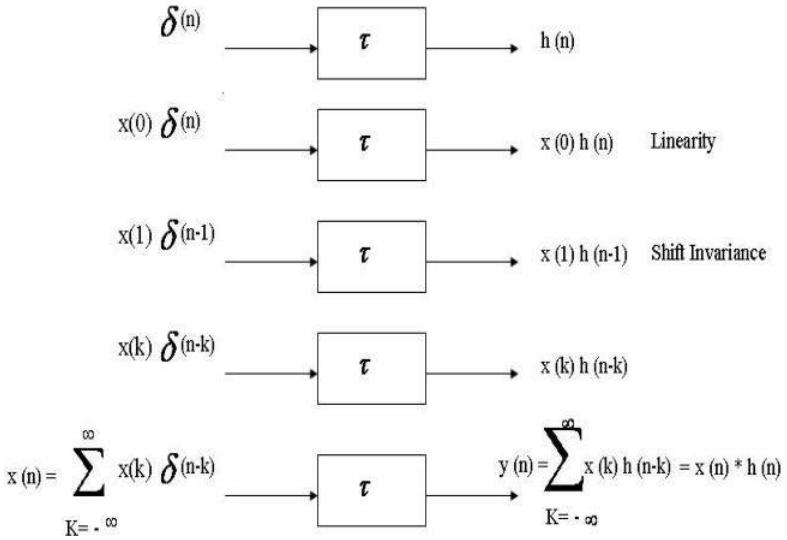
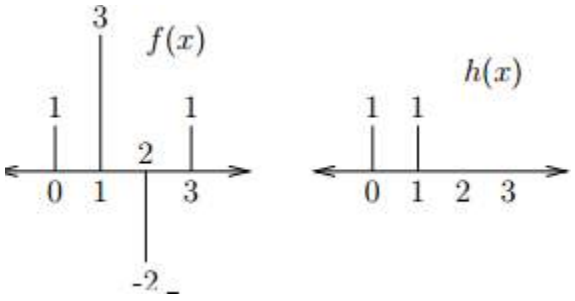



**Fig.2.1. Convolution of two signals**

signal can be found by adding these scaled and shifted impulse responses. In other words, if we know a system's impulse response, then we can calculate what the output will be for any possible input signal.

One dimensional linear discrete convolution is defined as: $g(x) = X\infty\ s=-\infty\ f(s)\ h(x - s) = f(x) * h(x)$

For example, consider the convolution of the following two functions:




This convolution can be performed graphically by reflecting and shifting h(x),.The samples of f(s) and h(s − x) that line up vertically are multiplied and summed:

$g(0) = f(-1)h(1) + f(0)h(0) = 0 + 1 = 1$

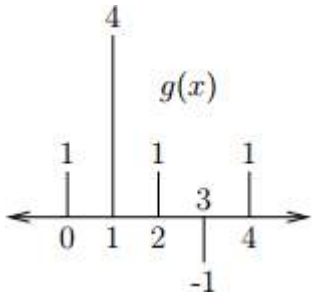
$g(1) = f(0)h(1) + f(1)h(0) = 1 + 3 = 4$

$g(2) = f(1)h(1) + f(2)h(0) = 3 + -2 = 1$

$g(3) = f(2)h(1) + f(3)h(0) = -2 + 1 = -1$

$g(4) = f(3)h(1) + f(4)h(0) = 1 + 0 = 1$

The result of the convolution is as shown below:




Notice that when f(x) is of length 4, and h(x) is of length 2,

the linear convolution is of length 4 + 2 − 1 = 5.

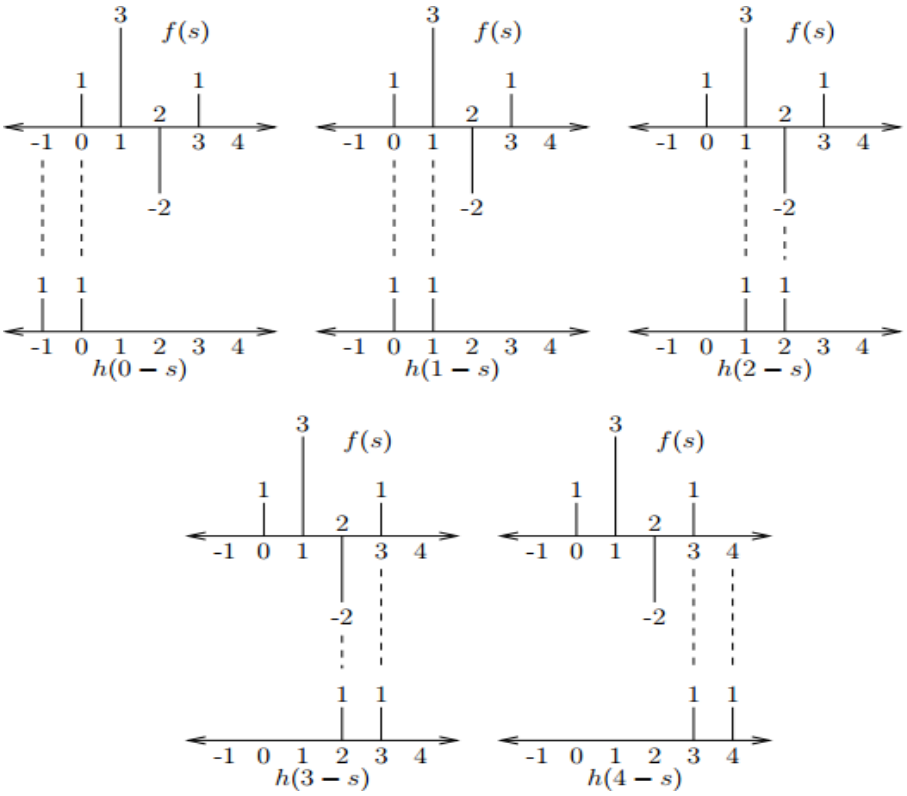



**Fig. 2.2 Linear Convolution using Graphical method**

### 2.1.1 Circular Convolution

One dimensional circular discrete convolution is defined as:

$g(x) = f(s)\, h((x - s)$

For M = 4, the convolution can be performed using circular reflection and shifts of h(x).T he samples of f(s) and h((s − x) mod M) that line up vertically are multiplied and summed:

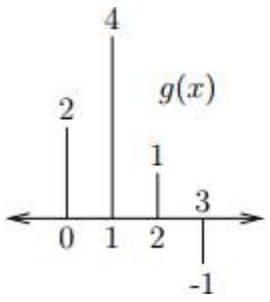
$g(0) = f(3)h(1) + f(0)h(0) = 1 + 1 = 2$

$g(1) = f(0)h(1) + f(1)h(0) = 1 + 3 = 4$

$g(2) = f(1)h(1) + f(2)h(0) = 3 + -2 = 1$

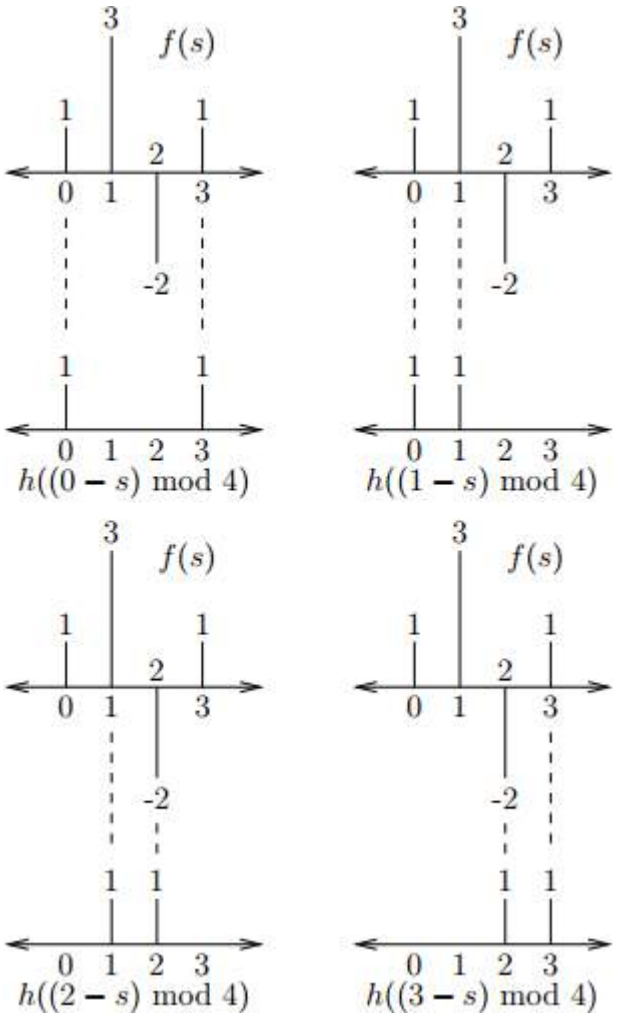
$g(3) = f(2)h(1) + f(3)h(0) = -2 + 1 = -1$

The result of the convolution is as shown below




Notice that f(x) and h(x) are both treated as if they are of length 4, and the circular convolution is also of length 4.

**Fig.2.2 Circular convolution using graphical method**




## 2.2.Correlation

Correlation is used to compare the similarity of two sets of data. Correlation computes a measure of similarity of two input signals as they are shifted by one another. The correlation result reaches a maximum at the time when the two signals match best. If the two signals are identical, this maximum is reached at t = 0 (no delay). If the two signals have similar shapes but one is delayed in time and possibly has noise added to it then correlation is a good method to measure that delay.

### Cross Correlation

The crosscorrelation between two real signals x and y is given by

$$C_{xy}(\tau) \equiv \int_{-\infty}^{\infty} x(t)y(t-\tau)dt$$

where the time shift or m is called the lag

***Autocorrelation***

Autocorrelation refers to the correlation of a time series with its own past and future values. Autocorrelation is also sometimes called "lagged correlation" or "serial correlation", which refers to the correlation between members of a series of numbers arranged in time. Positive autocorrelation might be considered a specific form of "persistence", a tendency for a system to remain in the same state from one observation to the next

The autocorrelation of a real signal s is given by

$$C_s(\tau) \equiv \int_{-\infty}^{\infty} s(t)s(t-\tau)dt$$

**2.3. Discrete Fourier Transform**

The Discrete Fourier Transform (DFT) is the equivalent of the continuous Fourier Transform for signals known only at N instants separated by sample Times (i.e. a finite sequence of data)

Let f(t) be the continuous signal which is the source of the data. Let N samples be denoted as f[0], f[1], ……f[N-1]

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

The Fourier Transform of the original signal, f(t) is given as:

Each sample f(k) can be regarded as an impulse having area f[k]. Then, since the integrand exists only

$$F(j\omega) = \int_{o}^{(N-1)T} f(t)e^{-j\omega t}dt$$
$$= f[0]e^{-j0} + f[1]e^{-j\omega T} + \ldots + f[k]e^{-j\omega kT} + \ldots f(N-1)e^{-j\omega(N-1)T}$$

$$\text{ie.} \quad F(j\omega) = \sum_{k=0}^{N-1} f[k]e^{-j\omega kT}$$

at the sample points:

The continuous Fourier transform could be evaluated over a finite interval (usually the fundamental period $T_0$) rather than from -∞ to +∞, the waveform

was periodic ,similarly, since there are only a finite number of input data points, the DFT treats the data as if it were periodic (i.e.f(N) to f(2N-1)).

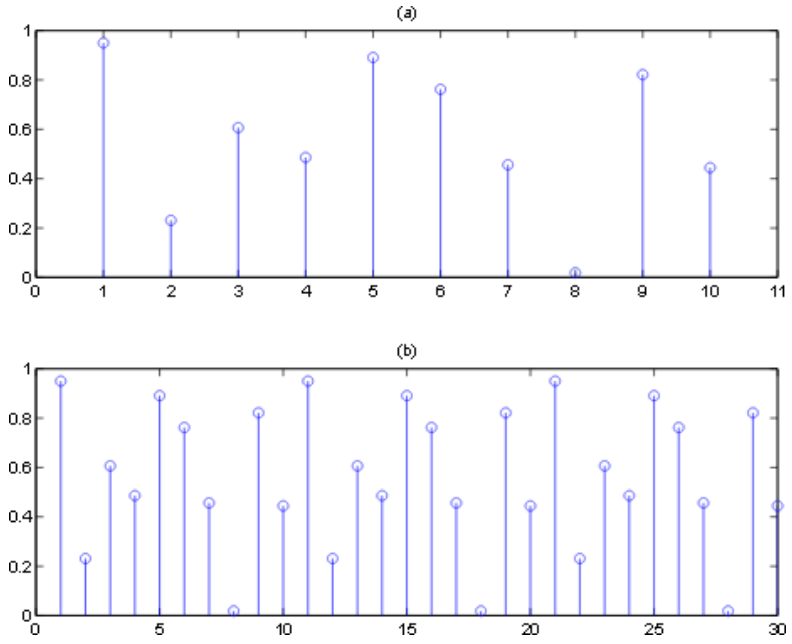



**Fig. 2.3.a) Sequence of n samples b) Periodicity in DFT**

Since the operation treats the data as if it were periodic, we evaluate the DFT equation for the fundamental frequency and its harmonics.

set $\omega = 0, \frac{2\pi}{NT}, \frac{2\pi}{NT} \times 2, \ldots \frac{2\pi}{NT} \times n, \ldots \frac{2\pi}{NT} \times (N-1)$

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \quad (n = 0 : N-1)$$

F[n] is the discrete Fourier Transform of f(k)

**Inverse Discrete Fourier Transform**

The Inverse transform is given as

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \quad (n = 0 : N-1)$$

## 2.4. FFT algorithm.

While the DFT transform above can be applied to any complex valued series, in practice for large series it can take considerable time to compute, the time taken being proportional to the square of the number on points in the series. A much faster algorithm has been developed by Cooley and Tukey around 1965 called the FFT (Fast Fourier Transform). The only requirement of the the most popular implementation of this algorithm (Radix-2 Cooley-Tukey) is that the number of points in the series be a power of 2. The computing time for the radix-2 FFT is proportional to

$$N \log_2(N)$$

So for example a transform on 1024 points using the DFT takes about 100 times longer than using the FFT, a significant speed increase. Note that in reality comparing speeds of various FFT routines is problematic, many of the reported timings have more to do with specific coding methods and their relationship to the hardware and operating system.

**Sample transform pairs and relationships**

- The Fourier transform is linear, that is a f(t) + b g(t) ---> a F(f) + b G(f)

$a\ x_k + b\ y_k$ ---> $a\ X_k + b\ Y_k$

- Scaling relationship f(t / a) ---> a F(a f) f(a t) ---> F(f / a) / a
- Shifting

$f(t + a)$ ---> $F(f)\ e^{-j\ 2\ pi\ a\ f}$

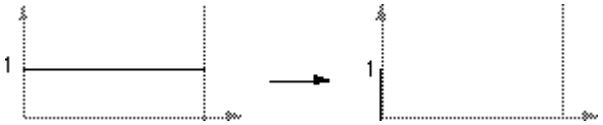
- Modulation

$f(t)\ e^{j\ 2\ pi\ a\ t}$ ---> F(t - a)

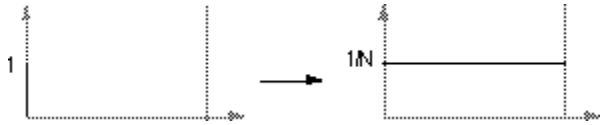
- Duality

$X_k$ ---> $(1/N)\ x_{N-k}$

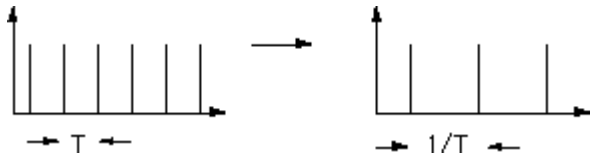
Applying the DFT twice results in a scaled, time reversed version of the original series.
The transform of a constant function is a DC value only.



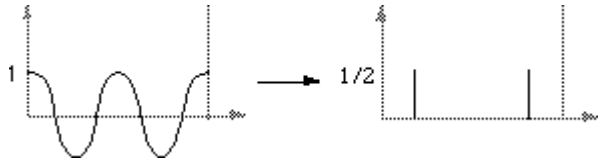

· The transform of a delta function is a constant



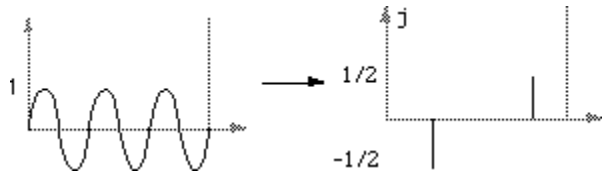

· The transform of an infinite train of delta functions spaced by T is an infinite train of delta functions spaced by 1/T.



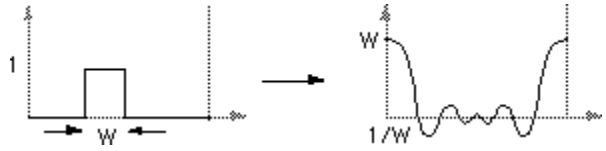

· The transform of a cos function is a positive delta at the appropriate positive and negative frequency.




· The transform of a sin function is a negative complex delta function at the appropriate positive frequency and a negative complex delta at the appropriate negative frequency.




· The transform of a square pulse is a sinc function




More precisely, if f(t) = 1 for |t| < 0.5, and f(t) = 0 otherwise then F(f) = sin(pi f) / (pi f)

· Convolution

f(t) **x** g(t) ---> F(f) G(f)

F(f) **x** G(f) ---> f(t) g(t) $x_k$ **x** $y_k$ ---> N $X_k$ $Y_k$

$x_k$ $y_k$ ---> (1/N) $X_k$ **x** $Y_k$

Multiplication in one domain is equivalent to convolution in the other domain and visa versa. For example the transform of a truncated sin function are two delta functions convolved with a sinc function, a truncated sin function is a sin function multiplied by a square pulse.

The transform of a triangular pulse is a $\text{sinc}^2$ function. This can be derived from first principles but is more easily derived by describing the triangular pulse as the convolution of two square pulses and using the convolution-multiplication relationship of the Fourier Transform.

**Decimation in Time**

The radix-2 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: a sum over the even-numbered discrete-time indices $n$=[0,2,4,…,$N$−2] and a sum over the odd-numbered indices $n$=[1,3,5,…,$N$−1] as in Equation:

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) e^{-\left(i\frac{2\pi nk}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi\times(2n)k}{N}\right)} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} + e^{-\left(i\frac{2\pi k}{N}\right)} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} \\
&= \mathrm{DFT}_{\frac{N}{2}}\left[[x(0), x(2), \ldots, x(N-2)]\right] + W_N^k \, \mathrm{DFT}_{\frac{N}{2}}\left[[x(1), x(3), \ldots, x(N-1)]\right]
\end{aligned}
$$

The mathematical simplifications in Equation reveal that all DFT frequency outputs $X(k)$ can be computed as the sum of the outputs of two length-$N2$ DFTs, of the even-indexed and odd-indexed discrete-time samples, respectively, where the odd-indexed short DFT is multiplied by a so-called twiddle factor term $WkN=e-(i2\pi kN)$. This is called a decimation in time because the time samples are rearranged in alternating groups, and a radix-2 algorithm because there are two groups. Figure graphically illustrates this form of the DFT computation, where for convenience the frequency outputs of the length-$N2$ DFT of the even-indexed time samples are denoted $G(k)$ and those of the odd-indexed samples as $H(k)$. Because of the periodicity with $N2$ frequency samples of these length-$N2$ DFTs, $G(k)$ and $H(k)$ can be used to compute **two** of the length-$N$ DFT frequencies, namely $X(k)$ and $X(k+N2)$, but with a different twiddle factor. This reuse of these short-length DFT outputs gives the FFT its computational savings.

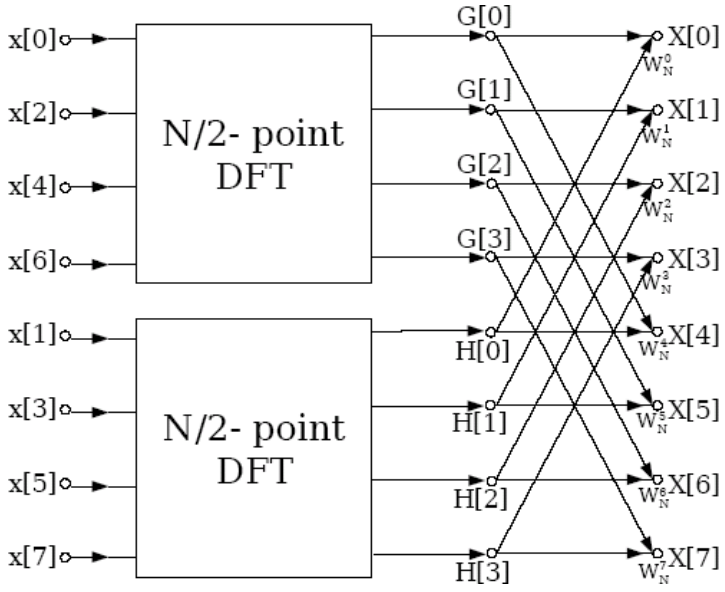



**Fig.2.4 Decimation in time**

**Decimation in Frequency**

$$
\begin{aligned}
X(2r) &= \sum_{n=0}^{N-1} x(n) W_N^{2rn} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2r\left(n+\frac{N}{2}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2rn} 1 \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + x\left(n+\tfrac{N}{2}\right)\right) W_{\frac{N}{2}}^{rn} \\
&= \mathrm{DFT}_{\frac{N}{2}} \left[x(n) + x\left(n+\tfrac{N}{2}\right)\right] \\
X(2r+1) &= \sum_{n=0}^{N-1} x(n) W_N^{(2r+1)n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + W_N^{\frac{N}{2}} x\left(n+\tfrac{N}{2}\right)\right) W_N^{(2r+1)n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right) W_{\frac{N}{2}}^{rn} \\
&= \mathrm{DFT}_{\frac{N}{2}} \left[\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right]
\end{aligned}
$$

The radix-2 decimation-in-frequency algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: computation of the even-numbered discrete-frequency indices $X(k)$ for $k=[0,2,4,\ldots,N-2]$ and computation of the odd-numbered indices $k=[1,3,5,\ldots,N-1]$ (or $X(2r+1)$

The mathematical simplifications in the above equations reveal that both the even-indexed and odd-indexed frequency outputs $X(k)$ can each be computed by a length-$N2$ DFT. The inputs to these DFTs are sums or differences of the first and second halves of the input signal, respectively, where the input to the short DFT producing the odd-indexed frequencies is multiplied by a so-called twiddle factor term $WkN=e-(i2\pi kN)$. This is called a decimation in frequency because the frequency samples are computed separately in alternating groups, and a radix-2 algorithm because there are two groups. Figure graphically illustrates this form of the DFT computation. This conversion of the full DFT into a series of shorter DFTs with a simple preprocessing step gives the decimation-in-frequency FFT its computational savings.

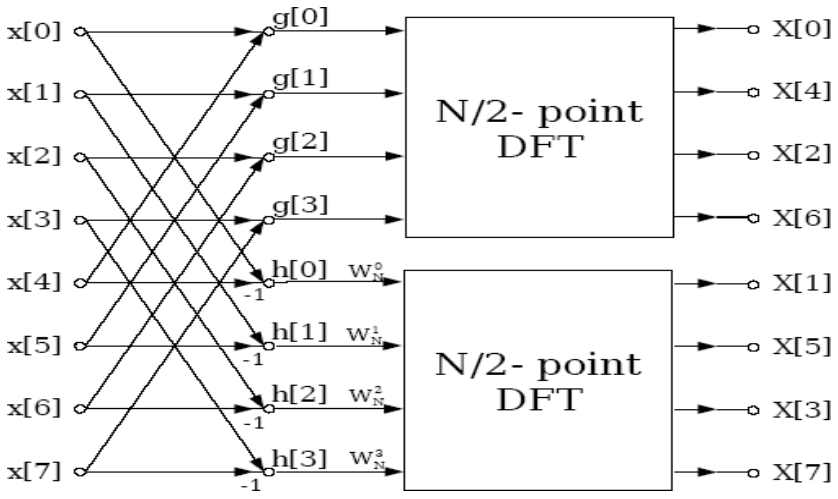



**Fig. 2.5 Decimation in Frequency**

## 2.5. Real Time Transforms

### 2.5.1 Discrete Cosine Transform

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. DCT is defined as

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right],$$

for u= 0,1, 2, , N-1. Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right],$$

for x= 0, 1, 2,…. ,N-1

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0. \end{cases}$$

There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT".[1][2] Its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of *overlapping* data.

**2.5.2 Walsh Transform:**

The matrix product of a square set of data d and a matrix of basis vectors consisting of Walsh functions. By taking advantage of the nested structure of the natural ordering of the Walsh functions, it is possible to speed the transform up from $O(n^2)$ to O(n ln n) steps, resulting in the so-called fast Walsh transform. Walsh transforms are widely used for signal and image processing, and can also be used for image compression

The Walsh functions consist of trains of square pulses (with the allowed states being -1 and 1) such that transitions may only occur at fixed intervals of a unit time step, the initial state is always+1, and the functions satisfy certain other orthogonality relations. In particular, the Walsh functions of order n are given by the rows of the Hadamard matrix $H_{2^n}$ when arranged in so-called "sequency" order.

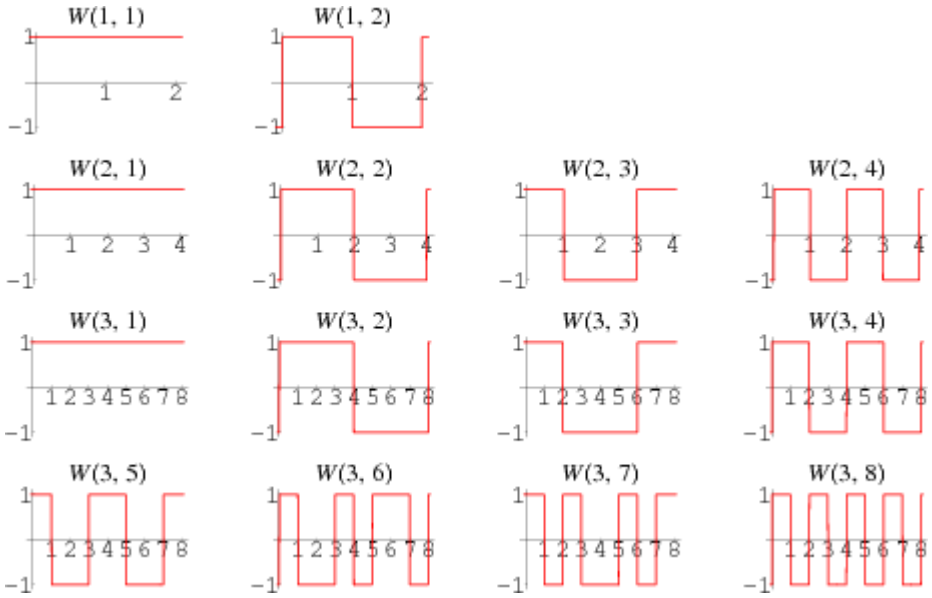



**Fig.2.6 Walsh Function**

### 2.5.3 Hadamard Transform

The Hadamard transform (also known as the Walsh–Hadamard transform, Hadamard–Rademacher–Walsh transform, Walsh transform, or Walsh–Fourier transform) is an example of a generalized class of Fourier transforms. It performs an orthogonal, symmetric, involutive, linear operation on real numbers (or complex numbers, although the Hadamard matrices themselves are purely real).

The Hadamard transform can be regarded as being built out of size-2 discrete Fourier transforms (DFTs), and is in fact equivalent to a multidimensional DFT of size $2 \times 2 \times \cdots \times 2 \times 2$. It decomposes an arbitrary input vector into a superposition of Walsh functions.

The Hadamard transform $H_m$ is a $2^m \times 2^m$ matrix, the Hadamard matrix (scaled by a normalization factor), that transforms $2^m$ real numbers $x_n$ into $2^m$ real numbers $X_k$. The Hadamard transform can be defined in two ways: recursively, or by using the binary (base-2) representation of the indices $n$ and $k$.

Recursively, we define the $1 \times 1$ Hadamard transform $H_0$ by the identity $H_0 = 1$, and then define $H_m$ for $m > 0$ by:

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

where the $1/\sqrt{2}$ is a normalization that is sometimes omitted.

For $m > 1$, we can also define $H_m$ by:

$$H_m = H_1 \otimes H_{m-1}$$

Thus, other than this normalization factor, the Hadamard matrices are made up entirely of 1 and −1.

Equivalently, we can define the Hadamard matrix by its (*k*, *n*)-th entry by writing

$$k = \sum_{i=0}^{m-1} k_i 2^i = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \cdots + k_1 2 + k_0$$

and

$$n = \sum_{i=0}^{m-1} n_i 2^i = n_{m-1}2^{m-1} + n_{m-2}2^{m-2} + \cdots + n_1 2 + n_0$$

where the $k_j$ and $n_j$ are the binary digits (0 or 1) of *k* and *n*, respectively. Note that for the element in the top left corner, we define: $k = n = 0$. In this case, we have:

$$(H_m)_{k,n} = \frac{1}{2^{\frac{m}{2}}} (-1)^{\sum_j k_j n_j}$$

This is exactly the multidimensional $2 \times 2 \times \cdots \times 2 \times 2$ DFT, normalized to be unitary, if the inputs and outputs are regarded as multidimensional arrays indexed by the $n_j$ and $k_j$, respectively.

Some examples of the Hadamard matrices follow.

$$H_0 = +1$$
$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

(This $H_1$ is precisely the size-2 DFT. It can also be regarded as the Fourier transform on the two-element *additive* group of **Z**/(2).)

$$H_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$H_3 = \frac{1}{2^{\frac{3}{2}}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

$$(H_n)_{i,j} = \frac{1}{2^{\frac{n}{2}}}(-1)^{i \cdot j}$$

where $i \cdot j$ is the bitwise dot product of the binary representations of the numbers i and j. For example, if $n \geq 2$, then $(H_n)_{3,2} = (-1)^{3 \cdot 2} = (-1)^{(1,1)\cdot(1,0)} = (-1)^{1+0} = (-1)^1 = -1$, agreeing with the above (ignoring the overall constant). Note that the first row, first column of the matrix is denoted by $(H_n)_{0,0}$.

The rows of the Hadamard matrices are the Walsh functions.

**2.5.4 Wavelet Transform**

A wavelet is a mathematical function used to divide a given function or continuous-time signal into different scale components. All wavelet transforms may be considered forms of time- frequency representation for continuous-time (analog) signals and so are related to harmonic analysis. Almost all practically useful discrete wavelet transforms use discrete-time filterbanks. These filter banks are called the wavelet and scaling coefficients in wavelets nomenclature. These filterbanks may contain either finite impulse response (FIR) or infinite impulse response (IIR) filters. The wavelets forming a continuous wavelet transform (CWT) are subject to the uncertainty principle of Fourier analysis respective sampling theory: Given a signal with some event in it, one cannot assign simultaneously an exact time and frequency response scale to that event. The product of the uncertainties of time and frequency response scale has a lower bound. Thus, in the scaleogram of a continuous wavelet transform of this signal, such an event marks an entire region in the time-scale plane, instead of just one point. Also, discrete wavelet bases may be considered in the context of other forms of the

uncertainty principle.

Wavelet transforms are broadly divided into three classes: continuous, discrete and multiresolution-based.

**Continuous wavelet transforms (continuous shift and scale parameters)**

In continuous wavelet transforms, a given signal of finite energy is projected on a continuous family of frequency bands (or similar subspaces of the $L^p$ function space $L^2(\mathbf{R})$ ). For instance the signal may be represented on every frequency band of the form [*f*, 2*f*] for all positive frequencies $f > 0$. Then, the original signal can be reconstructed by a suitable integration over all the resulting frequency components.

The frequency bands or subspaces (sub-bands) are scaled versions of a subspace at scale 1. This subspace in turn is in most situations generated by the shifts of one generating function $\psi$ in $L^2(\mathbf{R})$, the *mother wavelet*. For the example of the scale one frequency band [1, 2] this function is

$$\psi(t) = 2\,\mathrm{sinc}(2t) - \mathrm{sinc}(t) = \frac{\sin(2\pi t) - \sin(\pi t)}{\pi t}$$

with the (normalized) sinc function. That, Meyer's, and two other examples of mother wavelets are:

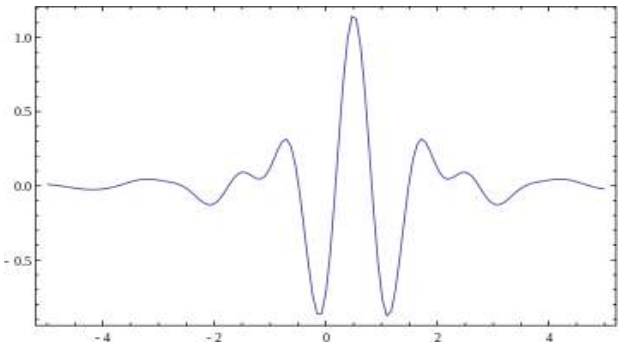



**Fig.2.7 Meyer**

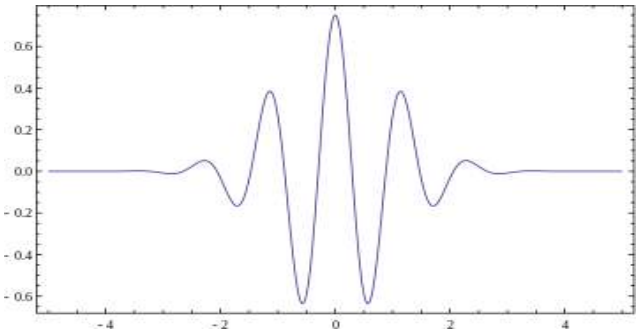



**Fig.2.8 Morlet**

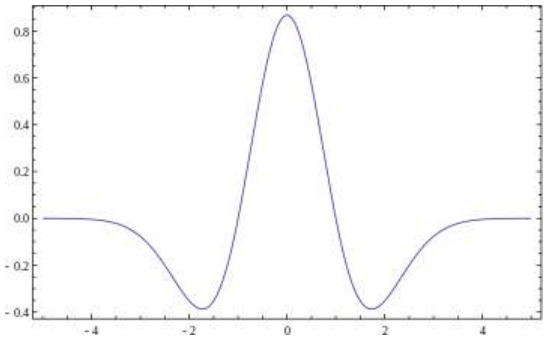



**Fig.2.9 Mexican hat**

The subspace of scale *a* or frequency band [1/*a*, 2/*a*] is generated by the functions (sometimes called *child wavelets*)

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right),$$

where *a* is positive and defines the scale and *b* is any real number and defines the shift. The pair (*a*, *b*) defines a point in the right halfplane $\mathbf{R}+ \times \mathbf{R}$.

The projection of a function *x* onto the subspace of scale *a* then has the form

$$x_a(t) = \int_{\mathbb{R}} WT_\psi\{x\}(a,b) \cdot \psi_{a,b}(t)\, dt$$

with *wavelet coefficients*

$$WT_\psi\{x\}(a,b) = \langle x, \psi_{a,b}\rangle = \int_{\mathbb{R}} x(t)\psi_{a,b}(t)\, dt.$$

See a list of some Continuous wavelets.

For the analysis of the signal *x*, one can assemble the wavelet coefficients into a scaleogram of the signal.

**Discrete wavelet transforms (discrete shift and scale parameters)**

It is computationally impossible to analyze a signal using all wavelet coefficients, so one may wonder if it is sufficient to pick a discrete subset of the upper halfplane to be able to reconstruct a signal from the corresponding wavelet coefficients. One such system is the affine system for

some real parameters $a > 1$, $b > 0$. The corresponding discrete subset of the halfplane consists of all the points ($a^m$, $na^m b$) with $m$, $n$ in **Z**. The corresponding *baby wavelets* are now given as

$$\psi_{m,n}(t) = a^{-m/2}\psi(a^{-m}t - nb).$$

A sufficient condition for the reconstruction of any signal $x$ of finite energy by the formula

$$x(t) = \sum_{m\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} \langle x,\, \psi_{m,n}\rangle \cdot \psi_{m,n}(t)$$

is that the functions $\{\psi_{m,n} : m, n \in \mathbb{Z}\}$ form a orthonormal basis of $L^2$(**R**).

**Multiresolution based discrete wavelet transform**

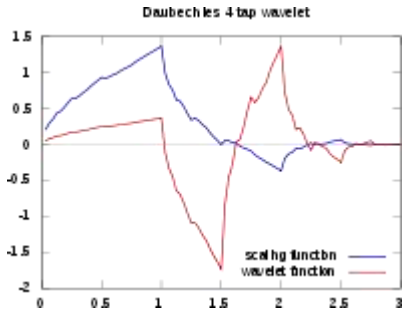



**Fig.2.10 D4 wavelet**

In any discretised wavelet transform, there are only a finite number of wavelet coefficients for each bounded rectangular region in the upper halfplane. Still, each coefficient requires the evaluation of an integral. In special situations this numerical complexity can be avoided if the scaled and shifted wavelets form a multiresolution analysis. This means that there has to exist an auxiliary function, the *father wavelet* φ in $L^2$(**R**), and that $a$ is an integer. A typical choice is $a = 2$ and $b = 1$. The most famous pair of father and mother wavelets is the Daubechies 4-tap wavelet. Note that not every orthonormal discrete wavelet basis can be associated to a multiresolution analysis; for example, the Journe wavelet admits no multiresolution analysis.

From the mother and father wavelets one constructs the subspaces

$$V_m = \operatorname{span}(\phi_{m,n} : n \in \mathbb{Z}), \text{ where } \phi_{m,n}(t) = 2^{-m/2}\phi(2^{-m}t - n)$$
$$W_m = \operatorname{span}(\psi_{m,n} : n \in \mathbb{Z}), \text{ where } \psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n).$$

The father wavelet $V_i$keeps the time domain properties, while the mother wavelets $W_i$keeps the frequency domain properties.

From these it is required that the sequence

$$\{0\} \subset \cdots \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots \subset L^2(\mathbb{R})$$

forms a multiresolution analysis of $L^2$ and that the subspaces $\ldots, W_1, W_0, W_{-1}, \ldots$ are the orthogonal "differences" of the above sequence, that is, $W_m$ is the orthogonal complement of $V_m$ inside the subspace $V_{m-1}$,

$$V_m \oplus W_m = V_{m-1}.$$

In analogy to the sampling theorem one may conclude that the space $V_m$ with sampling distance $2^m$ more or less covers the frequency baseband from 0 to $2^{-m-1}$. As orthogonal complement, $W_m$ roughly covers the band $[2^{-m-1}, 2^{-m}]$.

From those inclusions and orthogonality relations, especially $V_0 \oplus W_0 = V_{-1}$, follows the existence of sequences $h = \{h_n\}_{n\in\mathbb{Z}}$ and $g = \{g_n\}_{n\in\mathbb{Z}}$ that satisfy the identities

$$\phi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} g_n \phi(2t - n),$$

$g_n = \langle \phi_{0,0},\ \phi_{-1,n} \rangle$ so that

$$\psi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} h_n \phi(2t - n).$$ and

so that $h_n = \langle \psi_{0,0},\ \phi_{-1,n} \rangle$

The second identity of the first pair is a refinement equation for the father wavelet φ. Both pairs of identities form the basis for the algorithm of the fast wavelet transform.

From the multiresolution analysis derives the orthogonal decomposition of the space $L^2$ as

$$L^2 = V_{j0} \oplus W_{j0} \oplus W_{j0-1} \oplus W_{j0-2} \oplus W_{j0-3} \oplus \ldots$$

For any signal or function $S \in L^2$ this gives a representation in basis functions of the corresponding subspaces as

$$S = \sum_k c_{j0,k}\phi_{j0,k} + \sum_{j \le j0} \sum_k d_{j,k}\psi_{j,k}$$

where the coefficients are

and $c_{j0,k} = \langle S, \phi_{j0,k} \rangle$
$d_{j,k} = \langle S, \psi_{j,k} \rangle$.

**Questions:**

1.Obtain the Linear Convolution of the following Sequences

$\mathbf{x(n) = \{1, 2, 0.5, 1\}}$ $\mathbf{h(n) = \{1, \underset{\uparrow}{2}, 1, -1\}}$

2.Find the Circular convolution of the given sequences

x1(n) = {2, 1, 2, -1} and x2(n) = {1, 2, 3, 4 }.

3.Determine the Autocorrelation of the sequence x(n) = {1, 2, 3, 4 }

4. Obtain the cross correlation of the sequences

x(n) = {1, 1, 2, 2 } and y(n) = {1, 0.5, 1 }.

5.Find the DFT of the sequence for N=4, also find the magnitude and phase response

$$x(n) = \begin{cases} 1, & \text{for } 0 \le n \le 2 \\ 0, & \text{otherwise} \end{cases}$$

6.Obtain the DFT of the 8-point sequence by FFT –DIT algorithm

x(n) = { 2, 1, 2, 1, 1, 2, 1, 2}

7.Obtain the DFT of the 8-point sequence by FFT –DIF algorithm

x(n) = { 2, 1, 2, 1, 1, 2, 1, 2}

8.Obtain the DCT of the given image f(x) = {1, 2, 4, 7}

9. Obtain the DCT of the given image

$$\mathbf{f(x)} = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix}$$

10.Find the Hadamard transform of the image given by f(x) = {1, 2, 0, 3 }

11.Find the Hadamard transform of the image given by

$$\mathbf{f(x)} = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix}$$

12.Find the Walsh transform of the image given by f(x) = { 1,2,0, 3 }

13.Find the Walsh transform of the image given by

$$\mathbf{f(x)} = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix}$$

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 3 – EVENT DETECTION – SBMA5202

## UNIT 3 - EVENT DETECTION

### 3.1Detection of events and waves

### 3.1.1 Derivative based operators in QRS detection

QRS complex has the largest slope (rate of change of voltage) in a cardiac cycle ventricles. As the rate of change is given by the derivative operator, the operation would be the most logical starting point in an attempt to develop an algorithm to detect the QRS complex. The derivative operator enhances the QRS, although the resulting wave does not bear any resemblance to a typical QRS complex. The slow P and T waves have been suppressed by the derivative operators, while the output is the highest at the QRS. However, given the noisy nature of the results of the derivative-based operators, it is also evident that significant smoothing will be required before further processing can take place. Derivative-based algorithm for QRS detection progresses as

$$y_0(n) = |x(n) - x(n-2)|.$$

follows: the smoothed three-point first derivative yo (n) of the given signal x(n) is approximated as

The second derivative is approximated as

$$y_1(n) = |x(n) - 2x(n-2) + x(n-4)|.$$

The two results are weighted and combined to obtain

$$y_2(n) = 1.3y_0(n) + 1.1y_1(n).$$

The result y2(n) is scanned with a threshold of 1.0. Whenever the threshold is crossed, the subsequent eight samples are also tested against the same threshold. If at least six of the eight points pass the threshold test, the segment of eight samples is taken to be a part of a QRS complex. The procedure results in a pulse with its width proportional to that of the QRS complex; however, the method is sensitive to noise.

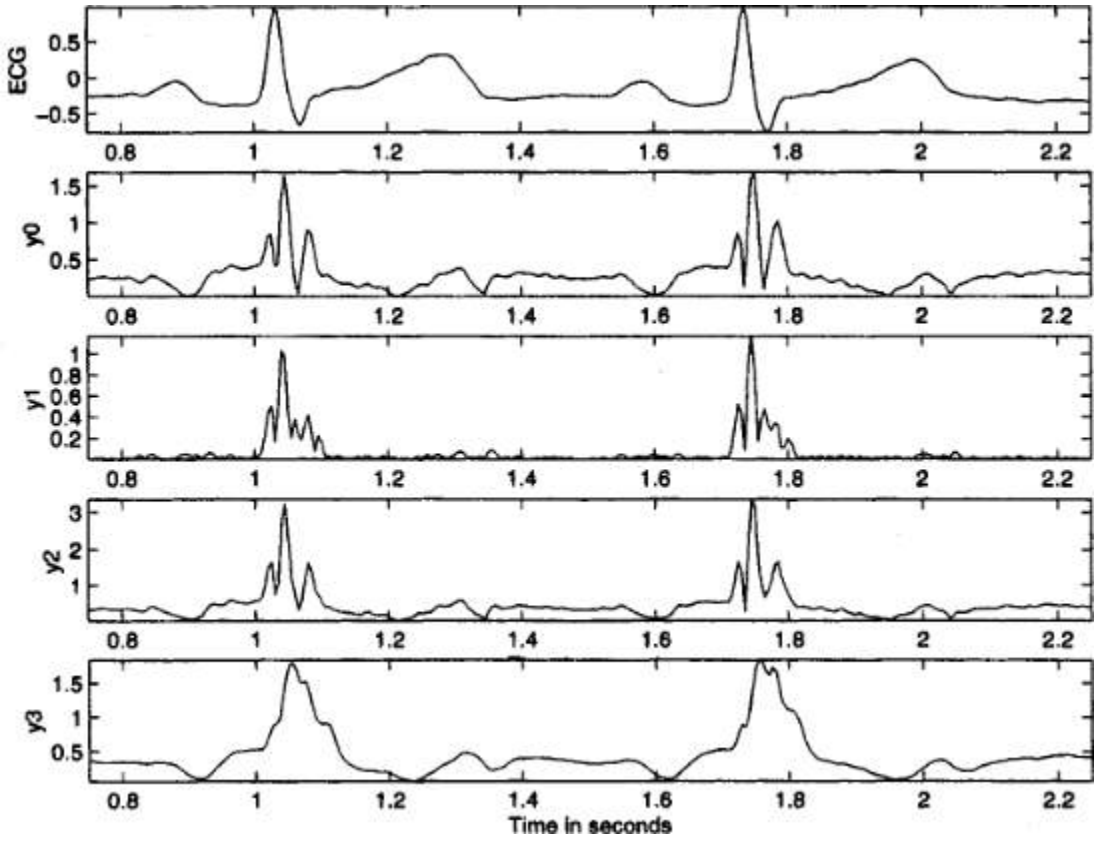



**Fig.3.1. From top to bottom: two cycles of a filtered version of the ECG signal, output yo(n)of the first-derivative-based operator, output yl(n) of the second-derivative-based operator and the result y3(n) of passing y2(n) through the 8-point MA filter**

### 3.1.2 Pan Tompkins algorithm

Pan and Tompkins proposed a real-time QRS detection al- gorithm based on analysis of the slope, amplitude, and width of QRS complexes. The algorithm includes a series of filters and methods that perform lowpass, high- pass, derivative, squaring, integration, adaptive thresholding, and search procedures.

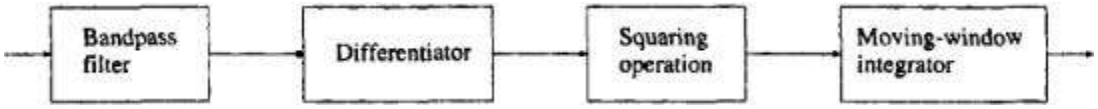


**Fig. 3.2 Block diagram of the Pan-Tompkins algorithm for QRS detection**

**Low pass filter:** The recursive lowpass filter used in the Pan-Tompkins algorithm has integer coefficients to reduce computational complexity, with the transfer function

defined as

The output y(n) is related to the input x(n) as

$$H(z) = \frac{1}{32}\,\frac{(1-z^{-6})^2}{(1-z^{-1})^2}.$$

The output y(n) is related to the input x(n) as

$$y(n) = 2y(n-1) - y(n-2) + \frac{1}{32}\,[x(n) - 2x(n-6) + x(n-12)].$$

With the sampling rate being 200 Hz, the filter has a rather low cutoff frequency of fc =

11 Hz, and introduces a delay of 5 samples or 25 ms.The filter provides an attenuation greater than

35 dB at 60 Hz,and effectively suppresses power-line interference, if present.

**Highpass filter**: The highpass filter used in the algorithm is implemented as an allpass

filter minus a lowpass filter. The lowpass component has the transfer function the input - output relationship is

$$H_{lp}(z) = \frac{(1-z^{-32})}{(1-z^{-1})};$$

the input - output relationship is

$$y(n) = y(n-1) + x(n) - x(n-32).$$

The transfer function Hhp(z) of the highpass filter is specified as

$$H_{hp}(z) = z^{-16} - \frac{1}{32}\,H_{lp}(z)$$

Equivalently, the output p(n) of the highpass filter is given by the difference equation

$$p(n) = x(n-16) - \frac{1}{32}\,[y(n-1) + x(n) - x(n-32)]$$

The high pass filter has a cutoff frequency of 5 Hz and introduces a delay of 80 ms.

**Derivative operator**: The derivative operation used by Pan and Tompkins is specified as

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

and approximates the ideal operator up to 30 Hz. The derivative procedure suppresses the low-frequency components of the P and T waves, and provides a large gain to the high-frequency components arising from the high slopes of the QRS complex.

**Squaring:** The squaring operation makes the result positive and emphasizes large differences resulting from QRS complexes; the small differences arising from P and T waves are suppresses. The high- frequency components in the signal related to the QRS complex are further enhanced.

**Integration**: As observed in the previous subsection, the output of a derivative- based operation will exhibit multiple peaks within the duration of a single QRS complex. The Pan-Tompkins algorithm performs smoothing of the output of the preceding operations through a moving-window integration filter as

$$y(n) = \frac{1}{N}\,[x(n-(N-1)) + x(n-(N-2)) + \cdots + x(n)]$$

The choice of the window width N is to be made with the following considerations: too large a value will result in the outputs due to the QRS and T waves being merged, whereas too small a value could yield several peaks for a single QRS. A window width of N = 30 was found to be suitable for fb = 200 Hz.

**Adaptive thresholding**: The thresholding procedure in the Pan-Tompkins algo- rithm adapts to changes in the ECG signal by computing running estimates of signal and noise peaks. A peak is said to be detected whenever the final output changes direction within a specified interval. In the following discussion, SPKI represents the peak level that the algorithm has learned to be that corresponding to QRS peaks, and NPKI represents the peak level related to non- QRS events (noise, EMG,etc.).

THRESHOLD I1 and THRESHOLD I2 are two thresholds used to categorize peaks detected as signal (QRS) or noise. Every new peak detected is categorized as a signal peak or a noise peak. If

a peak exceeds THRESHOLD I1 during the first step of analysis, it is classified as a QRS (signal) peak. If the searchback technique the peak.should be above THRESHOLDI2 to be called a QRS.The peak levels and thresholds are updated after each peak is detected and classified as if PEAKI is a signal peak; if PEAKI
is a noise peak;

SPKI = 0.125 PEAKI + 0.875 SPKI NPKI = 0.125 PEAKI + 0.875 NPKI
THRESHOLD I1 = NPKI + 0.25(SPKI - NPKI); THRESHOLD I2 = 0.5 THRESHOLD II.
The updating formula for SPKI is changed to SPKI = 0.25 PEAKI + 0.75 SPKI
If a QRS is detected in the searchback procedure using THRESHOLD 12.

**3.1.3 Search back procedure**: The Pan-Tompkins algorithm maintains two RR-interval averages: RR AVERAGE1 is the average of the eight most-recent beats, and RR AVERAGE2 is the average of the eight most-recent beats having RR intervals within the range specified by

RR LOW LIMIT = 0.92 x RR AVERAGE2 and RR HIGH LIMIT = 1.16 x RR AVERAGE2.
Whenever a QRS is not detected for a certain interval specified as

RR MISSED LIMIT = 1.06 x RR AVERAGE2, the QRS is taken to be the peak between the established thresh- olds applied in the searchback procedure.

**Correlation analysis**

**ACF and CCF in rhythm analysis**

Cross Correlation is defined as

$$x \cdot y = \langle x, y \rangle = \sum_{n=0}^{N-1} x(n)\, y(n),$$

where the signals $x(n)$ and $y(n)$ have $N$ samples each. The dot product represents the projection of one signal on to the other, with each signal being viewed as an $N$-dimensional vector. The dot product may be normalized by the geometric mean of the energies of the two signals to obtain a correlation coefficient as

$$\gamma_{xy} = \frac{\sum_{n=0}^{N-1} x(n)\, y(n)}{\left[\sum_{n=0}^{N-1} x^2(n)\ \sum_{n=0}^{N-1} y^2(n)\right]^{1/2}}.$$

The means of the signals may be subtracted out, if desired, as in Equation

In the case of two continuous-time signals $x(t)$ and $y(t)$, the projection of one signal on to the other is defined as

$$\theta_{xy} = \int_{-\infty}^{\infty} x(t)\, y(t)\, dt.$$

When a shift or time delay may be present in the occurrence of the epoch of interest in the two signals being compared, it becomes necessary to introduce a time-shift parameter to compute the projection for every possible position of overlap. The shift parameter facilitates searching one signal for the occurrence of an event matching that in the other signal at any time instant within the available duration of the signals. The CCF between two signals for a shift or delay of $\tau$ seconds or $k$ samples may be obtained as

$$\theta_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)\, y(t+\tau)\, dt,$$

or

$$\theta_{xy}(k) = \sum_{n} x(n)\, y(n+k).$$

The range of summation in the latter case needs to be limited to the range of the available overlapped data. A scale factor, depending upon the number of data samples used, needs to be introduced to obtain the true CCF, but is neglected here

An extended version of the correlation coefficient $\gamma_{xy}$ in Equation to include time shift

When the ACF or the CCF is computed for various shifts, a question arises about the data samples in one of the signal segments beyond the duration of the other. We may add zeros to one of the signals and increase its length by the maximum shift of interest, or we may use the true data samples from the original signal record, if available. The latter method was used wherever possible in the following illustrations.

In the case of random signals, we need to take the expectation or sample average of the outer product of the vectors formed by the available samples of the signals. Let $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$ and $\mathbf{y}(n) = [y(n), y(n-1), \ldots, y(n-N+1)]^T$ represent the $N$-dimensional vectorial form of the two signals $x(n)$ and

$y(n)$ with the most-recent $N$ samples being available in each signal at the time instant $n$. If $\mathbf{x}(n)$ and $\mathbf{y}(n)$ are sample observations of random processes, their CCF is defined as

$$\boldsymbol{\Theta}_{xy} = E[\mathbf{x}(n)\,\mathbf{y}^T(n)],$$

The outer product, which is an $N \times N$ matrix, provides the cross-terms that include all possible delays (shifts) within the duration of the given signals.

All of the equations given above may be modified to obtain the ACF by replacing the second signal $y$ with the first signal $x$. The signal $x$ is then compared with itself.

The ACF displays peaks at intervals corresponding to the period (and integral multiples thereof) of any periodic or repetitive pattern present in the signal. This property facilitates the detection of rhythms in signals such as the EEG: The presence of the $\alpha$ rhythm would be indicated by a peak in the neighborhood of $0.1$ $s$. The ACF of most signals decays and reaches negligible values after delays of a few milliseconds, except for periodic signals of infinite or indefinite duration for which the ACF will also exhibit periodic peaks. The ACF will also exhibit multiple peaks when the same event repeats itself at regular or irregular intervals. One may need to compute the ACF only up to certain delay limits depending upon the expected characteristics of the signal being analyzed.

The CCF displays peaks at the period of any periodic pattern present in *both* of the signals being analyzed. The CCF may, therefore, be used to detect rhythms present in common between between two signals, for example, between two channels of the EEG.

**Murmur detection**

Cardiac mechanical activity is appraised by auscultation and processing of heart sound recordings (known as *phonocardiographic signals*—PCG), which is an inexpensive and noninvasive procedure. The importance of classic auscultation has decreased due to its inherent restrictions: the performance of human ear with its physical limitations, the subjectivity of the examiner, difficult skills that take years to acquire and refine, etc. Anyway, the PCG has preserved its importance in pediatric cardiology, cardiology, and internal diseases, evaluating congenital cardiac defects,and primary home health care, where an intelligent stethoscope with decision support abilities would be valuable. Mostly, heart sounds consist of two regularly repeated thuds, known as S1 and S2, each appearing one after the other, for every heart beat. The time interval between S1 and S2 is the systole, while the S2 and next S1 gap corresponds to the diastole. S1 implies the closing of the tricuspid and mitral valves immediately preceding the systole, while S2 corresponds to the closing of the aortic and pulmonary valves at the end of systole. The normal

blood flow inside the heart is mainly laminar and therefore silent; but when the blood flow becomes turbulent it causes vibration of surrounding tissue and hence the blood flow is noisy and perceivable, originating the murmur, which according to the instant they appear are sorted into systolic or diastolic. Murmurs are some of the basic signs of pathological changes to be identified, but they overlap with the cardiac beat and can not be easily separated by the human ear. The automatic detection of murmurs strongly depends on the appropriate features (data representation), which mostly are related to timing, morphology, and spectral properties of heart sounds. Although cardiac murmurs are nonstationary signals and exhibit sudden frequency changes and transients, t is common to assume linearity of the feature sets extracted from heart sounds (time and spectral features, frequency representation with time resolution, and parametric modeling. o capture nonstationary transients and fast changes of PCG, the time–frequency features are widely used in heart sound analysis.

Consider the case expressed by the relation

$$y(t) = x(t) * h(t).$$

### 3.2 Homomorphic filtering

As in the case of the multiplicative homomorphic system, our goal is to convert the convolution operation to addition. From the convolution property of the Fourier transform, we know that

$$Y(\omega) = X(\omega)\, H(\omega).$$

Thus, application of the Fourier transform converts convolution to multiplication. Now, it is readily seen that the multiplicative homomorphic system may be applied to convert the multiplication to addition. Taking the complex logarithm of $Y(\omega)$, we have

$$\log[Y(\omega)] = \log[X(\omega)] + \log[H(\omega)]; \qquad X(\omega) \neq 0,\ H(\omega) \neq 0\ \forall\omega.$$

[Note: $\log_e[X(\omega)] = \hat{X}(\omega) = \log_e[|X(\omega)|\angle X(\omega)] = \log_e[|X(\omega)|] + j\angle X(\omega)$, where $|X(\omega)|$ and $\angle X(\omega)$ are the magnitude and phase spectra of $x(t)$.]

A linear filter may now be used to separate the transformed components of $x$ and $h$, with the assumption as before that they are separable in the transformed space. A series of the inverses of the transformations applied initially will take us back to the original domain.

While the discussion here has been in terms of application of the Fourier transform, the general formulation of the homomorphic filter by Oppenheim and Schafer [174] is in terms of the $z$-transform. However, the Fourier transform is equivalent to the $z$-transform evaluated on the unit circle in the $z$-plane, and the Fourier transform is more commonly used in signal processing than the $z$-transform.

Figure · gives a block diagram of the steps involved in a homomorphic filter for convolved signals. The path formed by the first three blocks (in the top row) transforms the convolution operation at the input to addition. The third block with the inverse Fourier transform is used to move back to a pseudo time domain. The last three blocks (in the bottom row) perform the reverse transformation, converting addition to convolution. The filter in between deals with (transformed) signals that are combined by simple addition.

When a sample observation or template of a typical version of a signal event is available, it becomes possible to design a filter that is *matched* to the characteristics of the event and maximizes the $SNR$ of the output. If a signal that contains repetitions of the event with almost the same characteristics is passed through the *matched filter*, the output should provide peaks at the time instants of occurrence of the event.

**1. Matched filters**

## Derivation of the transfer function of the matched filter

In order to derive the transfer function, $H(\omega)$, of the matched filter let the signal $x(t)$ be the input to the matched filter. The Fourier transform of $x(t)$ is

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\ \exp(-j\omega t)\ dt.$$

The output of the matched filter, $y(t)$, is given by the inverse Fourier transform of $Y(\omega) = X(\omega)H(\omega)$, as follows:

$$\begin{aligned} y(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)\ H(\omega)\ \exp(+j\omega t)\ d\omega \\ &= \int^{\infty} X(f)\ H(f)\ \exp(+j\,2\pi\, ft)\ df. \end{aligned}$$

In the second expression of the equation given above, the frequency variable has been changed from $\omega$ in radians per second to $f$ in $Hz$.

Consider the presence of white noise at the input, with the PSD

$$S_{\eta i}(f) = \frac{P_{\eta i}}{2},$$

where $P_{\eta i}$ is the average noise power at the input. Then, the noise PSD at the output is

$$S_{\eta o}(f) = \frac{P_{\eta i}}{2}\ |H(f)|^2.$$

The average output noise power is

$$P_{\eta o} = \frac{P_{\eta i}}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df.$$

The $RMS$ value of the noise in the absence of any signal is $\sqrt{P_{\eta o}}$.

Letting $t = t_0$ the magnitude of the instantaneous output signal at $t = t_0$ is

$$M_y = |y(t_0)| = \left| \int_{-\infty}^{\infty} X(f)\, H(f)\, \exp(+j\, 2\pi\, f t_0)\, df \right|.$$

Thus, the $SNR$ at the output is $\frac{M_y}{\sqrt{P_{\eta o}}}$.

To derive the optimal transfer function of the matched filter, we could maximize the $SNR$, which is equivalent to maximizing the expression

$$\frac{M_y^2}{P_{\eta o}} = \frac{\text{instantaneous peak power of signal}}{\text{noise mean power}},$$

which represents peak-power $SNR$ [212].

For a given signal $x(t)$, the total energy is a constant, given by

$$E_x = \int_{-\infty}^{\infty} x^2(t)\, dt = \int_{-\infty}^{\infty} |X(f)|^2\, df.$$

Let us consider the following ratio:

$$\frac{M_y^2}{E_x\, P_{\eta o}} = \frac{\left| \int_{-\infty}^{\infty} H(f)\, X(f)\, \exp(+j\, 2\pi\, f t_0)\, df \right|^2}{\frac{P_{\eta i}}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df \int_{-\infty}^{\infty} |X(f)|^2\, df}.$$

The quantity $E_x$ is a constant for a given input signal; hence, maximizing the expression in Equation is equivalent to maximizing the expression in Equation .

In order to determine the condition for maximization of the expression in Equation recall Schwarz's inequality for two arbitrary complex functions $A(f)$ and $B(f)$:

$$\left| \int_{-\infty}^{\infty} A(f)\, B(f)\, df \right|^2 \leq \left[ \int_{-\infty}^{\infty} |A(f)|^2\, df \right] \left[ \int_{-\infty}^{\infty} |B(f)|^2\, df \right]$$

For any two real functions $a(t)$ and $b(t)$, the corresponding inequality is

$$\left[ \int_{-\infty}^{\infty} a(t)\, b(t)\, dt \right]^2 \leq \left[ \int_{-\infty}^{\infty} a^2(t)\, dt \right] \left[ \int_{-\infty}^{\infty} b^2(t)\, dt \right].$$

For any two vectors **a** and **b**, Schwarz's inequality states that

$$|\mathbf{a}.\mathbf{b}| \leq |\mathbf{a}|\, |\mathbf{b}|,$$

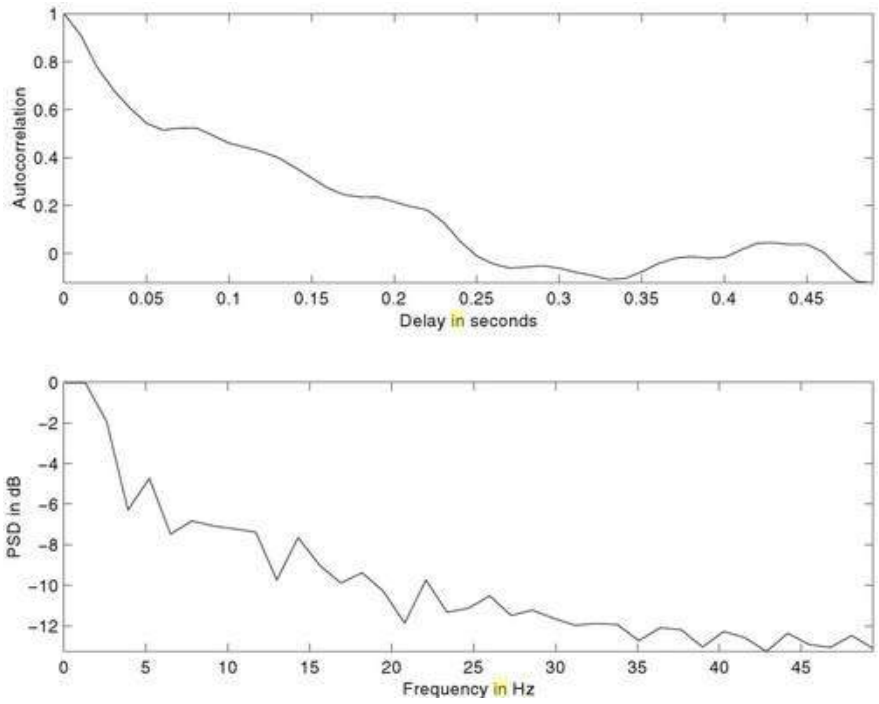
and

$$|\mathbf{a} + \mathbf{b}| \leq |\mathbf{a}| + |\mathbf{b}|.$$

## 2. Wavelet detection – Spike and wave detection

**Problem:** *Propose a method to detect spike-and-wave complexes in an EEG signal. You may assume that a sample segment of a spike-and-wave complex is available.*

**Solution:** A spike-and-wave complex is a well-defined event in an EEG signal. The complex is composed of a sharp spike followed by a wave with a frequency of about 3 $Hz$; the wave may contain a half period or a full period of an almost-




sinusoidal pattern. One may, therefore, extract an epoch of a spike-and-wave complex from an EEG channel and use it for template matching with the same formula

### 3.3 Extraction of vocal tract response and other applications

**Problem:** *Design a homomorphic filter to extract the basic wavelet corresponding to the vocal-tract response from a voiced speech signal.*

**Solution:** , voiced speech is generated by excitation of the vocal tract, while it is held in a particular form, with a glottal waveform that may be approximated as a series of pulses. The voiced speech signal may, therefore, be expressed in discrete-time terms as $y(n) = x(n) * h(n)$, where $y(n)$ is the speech signal, $x(n)$ is the glottal waveform (excitation sequence), and $h(n)$ is the impulse response of the vocal tract (basic wavelet). The $*$ symbol represents convolution, with the assumption that the vocal-tract filter may be approximated by an LSI filter. We may, therefore, use the homomorphic filter for convolved signals as introduced in the preceding section to separate $h(n)$ and $x(n)$.

The glottal excitation sequence may be further expressed as $x(n) = p(n) * g(n)$, where $p(n)$ is a train of ideal impulses (Dirac delta functions) and $g(n)$ is a smoothing function, to indicate that the physical vocal-cord system cannot produce ideal impulses but rather pulses of finite duration and slope

**Questions:**

1. Explain about the Event detection problem with suitable case study.
2. Explain about derivation based method for QRS detection in ECG signal.
3. With suitable block diagram explain the Pan–Tompkins algorithm for QRS detection.
4. Discuss about the Correlation Analysis of EEG Rhythms with suitable example.
5. Derive the transfer function of the matched filter.
6. Explain about Crossspectral Techniques which are suitable for the event detection of EEG signal .
7. Design a matched filter to detect spike-and-wave complexes in an EEG signal. A reference spike-and-wave complex is available.
8. Propose an algorithm to detect the P wave in the ECG signal.
9. Propose a homomorphic filter to separate two signals that have been combined through the convolution operation.
10. Explain the Operations involved in a homomorphic filter for convolved signal with suitable block diagram

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 4 – ECG DATA REDUCTION – SBMA5202

## UNIT 4 - ECG DATA REDUCTION

### 4.1 Direct data compression Techniques:

A data reduction algorithm seeks to minimize the number of code bits stored by reducing the redundancy present in the original signal. We obtain the reduction ratio by dividing the number of bits of the original signal by the number saved in the compressed signal. A data reduction algorithm must also represent the data with acceptable fidelity.

In biomedical data reduction, we usually determine the clinical acceptability of the reconstructed signal through visual inspection. We may also measure the residual, that is, the difference between the reconstructed signal and the original signal. Such a numerical measure is

$$\text{PRD} = \left\{ \frac{\sum_{i=1}^{n} [x_{org}(i) - x_{rec}(i)]^2}{\sum_{i=1}^{n} [x_{org}(i)]^2} \right\}^{\frac{1}{2}} \times 100\ \%$$

the percent root-mean-square difference, PRD, given by

where n is the number of samples and xorg and xrec are samples of the original and reconstructed data sequences.

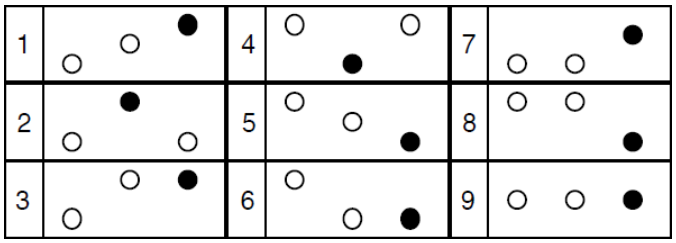
***Turning Point***

The original motivation for the turning point (TP) algorithm was to reduce the sampling frequency of an ECG signal from 200 to 100 samples/s .The algorithm developed from the observation that, except for QRS complexes with large amplitudes and slopes, a sampling rate of 100 samples/s is adequate.TP is based on the concept that ECG signals are normally oversampled at four or five times faster than the highest frequency present. For example, an ECG used in monitoring may have a bandwidth of 50 Hz and be sampled at 200 sps in order to easily visualize the higher- frequency attributes of the QRS complex. Sampling theory tells us that we can sample such a signal at 100 sps. TP provides away to reduce the effective sampling rate by half to 100 sps by selectively saving important signal points (i.e., the peaks and valleys or turning points).

The algorithm processes three data points at a time. It stores the first sample point and assigns it as the reference point X0. The next two consecutive points become X1 and X2. The algorithm retains either X1 or X2, depending on which point preserves the turning point (i.e., slope change) of the original signal. Fig. shows all the possible configurations of three consecutive sample points. In each frame, the solid point preserves the slope of the original three points. The algorithm saves this point and makes it the reference point X0 for the next iteration. It then samples the next two points, assigns them to X1 and X2, and repeats the process.

We use a simple mathematical criterion to determine the saved point. First consider a sign(x) operation

$$sign(x) = \begin{cases} 0 & x = 0 \\ +1 & x > 0 \\ -1 & x < 0 \end{cases}$$




| Pattern | $s_1 = sign(X_1 - X_0)$ | $s_2 = sign(X_2 - X_1)$ | NOT($s_1$) OR ($s_1 + s_2$) | Saved sample |
|---|---|---|---|---|
| 1 | +1 | +1 | 1 | $X_2$ |
| 2 | +1 | –1 | 0 | $X_1$ |
| 3 | +1 | 0 | 1 | $X_2$ |
| 4 | –1 | +1 | 0 | $X_1$ |
| 5 | –1 | –1 | 1 | $X_2$ |
| 6 | –1 | 0 | 1 | $X_2$ |
| 7 | 0 | +1 | 1 | $X_2$ |
| 8 | 0 | –1 | 1 | $X_2$ |
| 9 | 0 | 0 | 1 | $X_2$ |

**Fig. 4.1. Turning point Algorithm**

We then obtain s1 = sign(X1 – X0) and s2 = sign(X2 – X1), where (X1 – X0) and (X2 – X1)

are the slopes of the two pairs of consecutive points. If a slope is zero,this operator produces a zero result. For positive or negative slopes, it yields +1 or–1 respectively. A turning point occurs only when a slope changes from positive to negative or vice versa. We use the logical Boolean operators, NOT and OR, as implemented in the C language to make the final judgment of when a turning point occurs. In the C language, NOT(c) = 1 if c = 0; otherwise NOT(c) = 0. Also logical OR means that (a OR b ) = 0 only if a and b are both 0. Thus, we retain X1 only if {NOT(s1) OR (s1 + s2)} is zero, and save X2 otherwise. In this expression, (s1 + s2) is the arithmetic sum of the signs produced by the sign function. The final effect of this processingis a Boolean decision whether to save X1 or X2. Point X1 is saved only when the slope changes from positive to negative or vice versa. This computation could be easily done arithmetically, but the Boolean operation is computationally much faster.

The TP algorithm is simple and fast, producing a fixed reduction ratio of 2:1. After selectively discarding exactly half the sampled data, we can restore the original resolution by interpolating between pairs of saved data points. A second application of the algorithm to the already reduced data increases the reduction ratio to 4:1. Using data acquired at a 200-sps rate, this produces compressed data with a 50-sps effective sampling rate. If the bandwidth of the acquired ECG is 50 Hz, this approach violates sampling theory since the effective sampling rate is less than twice the highest frequency present in the signal. The resulting reconstructed signal typically has a widened QRS complex and sharp edges that reduce its clinical acceptability. Another disadvantage of this algorithm is that the saved points do not represent equally spaced time intervals. This introduces short term time distortion. However, this localized distortion is not visible when the reconstructed signal is viewed on the standard clinical monitors and paper recorders.

**AZTEC**

Originally developed to preprocess ECGs for rhythm analysis, the AZTEC (Amplitude Zone Time Epoch Coding) data reduction algorithm decomposes raw ECG sample points into plateaus and slopes.It provides a sequence of line segments that form a piecewise-linear approximation to the ECG. The algorithm consists of two parts—line detection and line processing. Figure(a) shows the line detection operation which makes use of zero-order interpolation (ZOI) to produce horizontal lines. Two variables Vmx and Vmn always reflect the highest and lowest elevations of the current line. Variable LineLen keeps track of the number of

samples examined. We store a plateau if either the difference between Vmxi and Vmni is greater than a predetermined threshold Vth or if LineLen is greater than 50. The stored values are the length (LineLen – 1) and the average amplitude of the plateau (Vmx + Vmn)/2.

Figure (b) shows the line processing algorithm which either produces a

plateau or a slope depending on the value of the variable LineMode. We initialize LineMode to

_PLATEAU in order to begin by producing a plateau. The production

of an AZTEC slope begins when the number of samples needed to form a plateau is less than three. Setting LineMode to _SLOPE indicates that we have entered slope production mode. We then determine the direction or sign of the current slope by subtracting the previous line amplitude V1 from the current

amplitude Vsi. We also reset the length of the slope Tsi. The variable Vsi records the current line amplitude so that any change in the direction of the slope can be tracked. Note that Vmxi and Vmni are always updated to the latest sample before line detection begins. This forces ZOI to begin from the value of the latest sample.

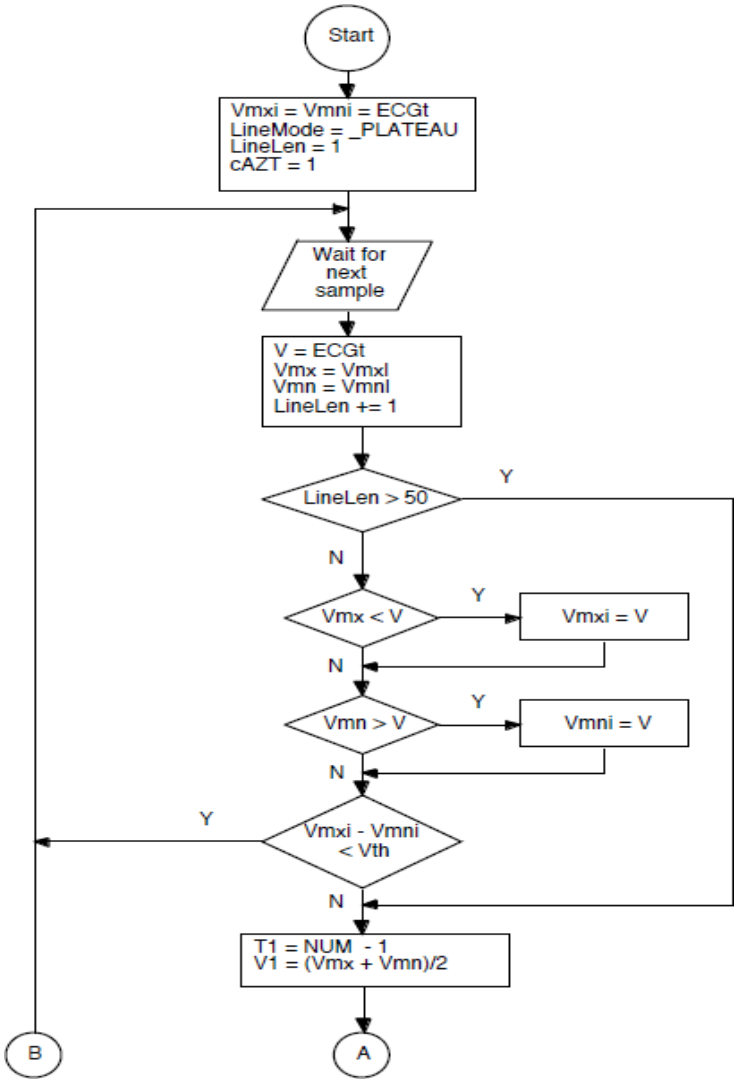



***Fig. 2(a) Flowchart for the line detection operation of the AZTEC algorithm***

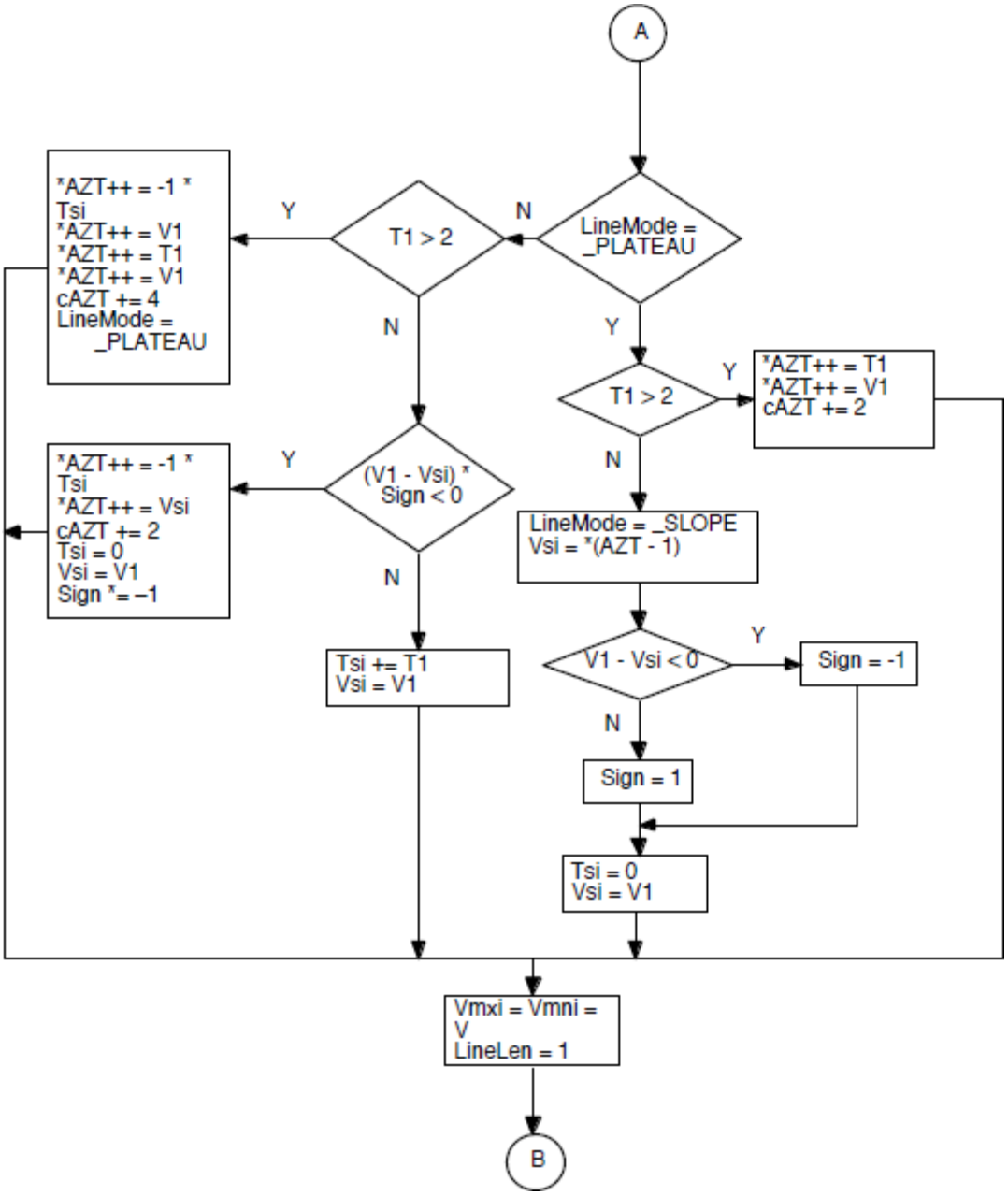



**Fig. 4.2(b) Flowchart of the line processing operation of the AZTEC algorithm**

When we reenter line processing with LineMode equal to _SLOPE, we either save or update the slope. The slope is saved either when a plateau of more than three samples can be formed or when a change in direction is detected. If we detect a new plateau of more than three samples, we store the current slope and the new plateau. For the slope, the stored values are its length

Tsi and its final elevation V1. Note that Tsi is multiplied by –1 to differentiate a slope from a plateau (i.e., the minus sign serves as a flag to indicate a slope). We also store the length and the amplitude of the new plateau, then reset all parameters and return to plateau production.

If a change in direction is detected in the slope, we first save the parameters for the current slope and then reset sign, Vsi, Tsi, Vmxi, and Vmni to produce a new AZTEC slope. Now the algorithm returns to line detection but remains in slope production mode. When there is no new plateau or change of direction, we simply update the slope's parameters, Tsi and Vsi, and return to line detection with LineMode remaining set to _SLOPE. AZTEC does not produce a constant data reduction ratio. The ratio is frequently as great as 10 or more, depending on the nature of the signal and the value of the empirically determined threshold.

**Cortes Algorithm**

The CORTES (Coordinate Reduction Time Encoding System) algorithm is a hybrid of the TP and AZTEC algorithms. It attempts to exploit the strengths of each while sidestepping the weaknesses.

CORTES uses AZTEC to discard clinically insignificant data in the isoelectric region with a high reduction ratio and applies the TP algorithm to the clinically significant high-frequency regions (QRS complexes). It executes the AZTEC and TP algorithms in parallel on the incoming ECG data. Whenever an AZTEC line is produced, the CORTES algorithm decides, based on the length of the line, whether the AZTEC data or the TP data are to be saved. If the line is longer than an empirically determined threshold, it saves the AZTEC line. Otherwise it saves the TP data points. Since TP is used to encode the QRS complexes, only AZTEC plateaus, not slopes, are implemented. The CORTES algorithm reconstructs the signal by expanding the AZTEC plateaus and interpolating between each pair of the TP data points. It then applies parabolic smoothing to the AZTEC portions to reduce discontinuities.

**FAN Algorithm**

Originally used for ECG telemetry, the Fan algorithm draws lines between pairs of starting and ending points so that all intermediate samples are within some specified error tolerance. We start by accepting the first sample X0 as the nonredundant permanent point. It functions as the origin and is also called the originating point.We then take the second sample X1 and draw two slopes {U1, L1}. U1 passes

through the point (X0, X1 + ε), and L1 passes through the point (X0, X1 – ε). If the third

sample X2 falls within the area bounded by the two slopes, we generate two new slopes {U2, L2} that pass through points (X0, X2 + ε) and (X0, X2 – ε). We compare the two pai rs of slopes and retain the most converging (restrictive) slopes (i.e., {U1, L2} in our example). Next we assign the value of X2 to X1 and read the next sample into X2. As a result, X2 always holds the most recent sample and X1 holds the sample immediately preceding X2. We repeat the process by comparing X2 to the values of the most convergent slopes. If it falls outside this area, we save

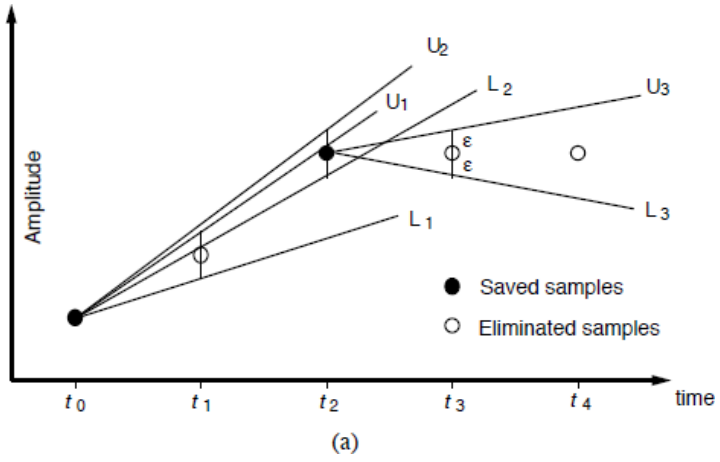
the length of the line T and its final amplitude X1 which then becomes the new originating point X0, and the process begins anew. The sketch of the slopes drawn from the originating sample to future samples forms a set of radial lines similar to a fan, giving this algorithm its name. When adapting the Fan algorithm to C-language implementation, we create the variables,

$$X_{U2} = \frac{X_{U1} - X_0}{T} + X_{U1}$$

XU1, XL1, XU2, and XL2, to determine the bounds of X2.

$$X_{L2} = \frac{X_{L1} - X_0}{T} + X_{L1}$$

where $T = t_T - t_0$.




(a)

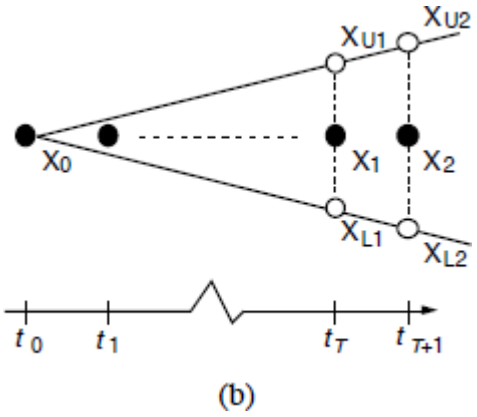



***Fig. 4.3 a) Upper and lower slopes (U and L) are drawn within error threshold ε around sample points taken at t1, t2, … (b) Extrapolation of XU2 and XL2 from XU1, XL1, and X0.***

We reconstruct the compressed data by expanding the lines into discrete points. The Fan algorithm guarantees that the error between the line joining any two permanent sample points and any actual

(redundant) sample along the line is less than or equal to the magnitude of the preset error tolerance. The algorithm's reduction ratio depends on the error tolerance. When compared to the TP and AZTEC algorithms, the Fan algorithm produces better signal fidelity for the same reduction ratio

**1. Transformation Compression Techniques:**

**Karhunen-Loeve Transform**

The Karhunen-Loeve Transform (KLT) (also known as Hotelling Transform and Eigenvector Transform) is closely related to the Principal Component Analysis (PCA) and widely used in data analysis in many fields.

$$\mathbf{\Sigma}_x \phi_k = \lambda_k \phi_k \qquad (k = 1, \cdots, N)$$

$$\begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \sigma_{ij} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} \\ \phi_k \\ \\ \end{bmatrix} = \lambda_k \begin{bmatrix} \\ \phi_k \\ \\ \end{bmatrix} \qquad (k = 1, \cdots, N)$$

or in matrix form:

$$\mathbf{\Sigma}_x = \mathbf{\Sigma}_x^{*T} \phi_i$$

As the covariance matrix is Hermitian (symmetric ifis real), its eigenvector 's are orthogonal:

$$\langle \phi_i, \phi_j \rangle = \phi_i^T \phi_j^* = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$N \times N$

$\mathbf{\Phi} \triangleq [\phi_1, \cdots, \phi_N]$

The $N$eigenequations above can be combined to be expressed as:

$\mathbf{\Sigma}_x \mathbf{\Phi} = \mathbf{\Phi} \mathbf{\Lambda}$

$$\begin{bmatrix} \ddots & \cdots & \cdots \\ \vdots & \sigma_{ij} & \vdots \\ \cdots & \cdots & \ddots \end{bmatrix} [\phi_1, \cdots, \phi_N] = [\phi_1, \cdots, \phi_N] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_N \end{bmatrix}$$

or in matrix form:

Here $\mathbf{\Lambda}$ is a diagonal matrix $\mathbf{\Lambda} = diag(\lambda_1, \cdots, \lambda_N)$. Left multiplying $\mathbf{\Phi}^T = \mathbf{\Phi}^{-1}$ on $\mathbf{\Sigma}_x$

both sides, the covariance matrix can be diagonalized:

$\mathbf{\Phi}^{*T} \mathbf{\Sigma}_x \mathbf{\Phi} = \mathbf{\Phi}^{-1} \mathbf{\Sigma}_x \mathbf{\Phi} = \mathbf{\Phi}^{-1} \mathbf{\Phi} \mathbf{\Lambda} = \mathbf{\Lambda}$

Now, given a signal vector $\mathbf{x}$ , we can define a unitary (orthogonal if $\mathbf{x}$ is real) Karhunen-Loeve Transform of as:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \mathbf{\Phi}^{*T}\mathbf{x} = \begin{bmatrix} \phi_1^{*T} \\ \vdots \\ \phi_N^{*T} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots, \\ x_N \end{bmatrix}$$

where the ith component of the transform vector is the projection of onto :

$$y_i = \langle \phi_i, \mathbf{x} \rangle = \phi_i^T \mathbf{x}^*$$

$$\mathbf{\Phi} = (\mathbf{\Phi}^{*T})^{-1} \qquad \mathbf{y} = \mathbf{\Phi}^{*T}\mathbf{x}$$

Left multiplying on both sides of the transform, we get the inverse transform:

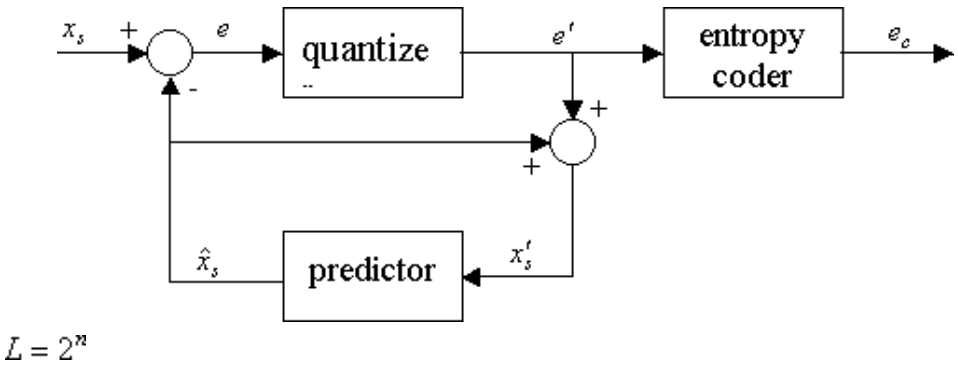
**Other data compression Techniques:**

**DPCM**

**D**ifferential **p**ulse **c**ode **m**odulation (DPCM) is a procedure of converting an analog into a digital signal in which an analog signal is sampled and then the difference between the actual sample value and its

predicted value (predicted value is based on previous sample or samples) is quantized and then encoded forming a digital value. DPCM code words represent differences between samples unlike PCM where code words represented a sample value. Basic concept of DPCM - coding a difference, is based on the fact that most source signals show significant correlation between successive samples so encoding uses redundancy in sample values which implies lower bit rate. Realization of basic concept (described above) is based on a technique in which we have to predict current sample value based upon previous samples (or sample) and we have to encode the difference between actual value of sample and predicted value (the difference between samples can be interpreted as prediction error). Because it's necessary to predict sample value DPCM is form of predictive coding.

DPCM compression depends on the prediction technique, well-conducted prediction techniques lead to good compression rates, in other cases DPCM could mean expansion comparing to regular PCM encoding.




$L = 2^n$

***Fig.4.4 DPCM encoder (transmitter)***

**Huffman coding**

Huffman coding exploits the fact that discrete amplitudes of quantized signal do not occur with equal probability (Huffman, 1952). It assigns variable-length code words to a given quantized data sequence according to their frequency of occurrence. Data that occur frequently are assigned shorter code words.

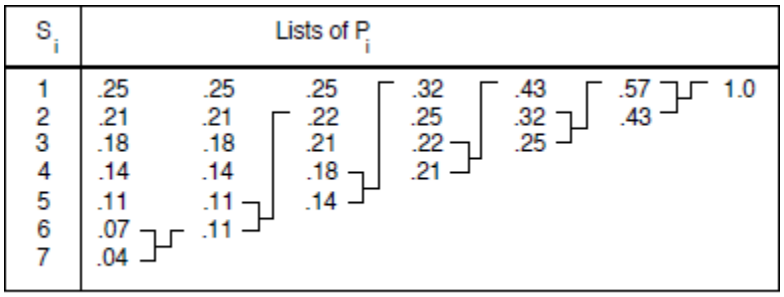
***Static Huffman coding***

As an example, assume that we wish to transmit the set of 28 data points

{1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7}

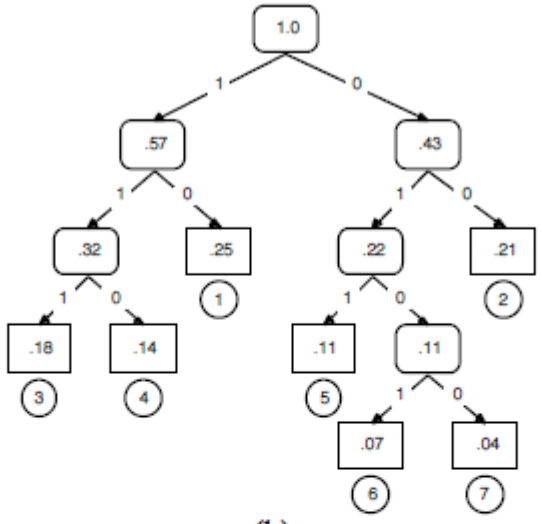
The set consists of seven distinct quantized levels, or symbols. For each symbol, Si, we calculate its probability of occurrence Pi by dividing its frequency of occurrence by 28, the total number of data points. Consequently, the construction of a Huffman code for this set begins with seven nodes, one associated with each Pi. At each step we sort the Pi list in descending order, breaking the ties arbitrarily.

The two nodes with smallest probability, Pi and Pj, are merged into a new node with probability Pi + Pj. This process continues until the probability list contains




(a)

a single value.




(b)

| Symbols, $S_i$ | 3-bit binary code | Probability of occurrence, $P_i$ | Huffman code |
|---|---|---|---|
| 1 | 001 | 0.25 | 10 |
| 2 | 010 | 0.21 | 00 |
| 3 | 011 | 0.18 | 111 |
| 4 | 100 | 0.14 | 110 |
| 5 | 101 | 0.11 | 011 |
| 6 | 110 | 0.07 | 0101 |
| 7 | 111 | 0.04 | 0100 |

(c)

***Fig.4. 5 Illustration of Huffman coding. (a) At each step, Pi are sorted in descending order* and the two lowest Pi are merged. (b) Merging operation depicted in a binary tree. (c) Summary of Huffman coding for the data set.**

The process of merging nodes produces a binary tree as in Figure (b). When we merge two nodes with probability Pi + Pj, we create a parent node with two children represented by Pi and Pj. The root of the tree has probability 1.0. We obtain the Huffman code of the symbols by traversing down the tree, assigning 1 to the left child and 0 to the right child. The resulting code words have the prefix property (i.e., no code word is a proper prefix of any other code word). This property ensures that a coded message is uniquely decodable without the need for lookahead.

Figure (c) summarizes the results and shows the Huffman codes for

the seven symbols. We enter these code word mappings into a translation table and use the table to pad the appropriate code word into the output bit stream in the reduction process. The reduction ratio of Huffman coding depends on the distribution of the source symbols. In our example, the original data requires three bits to represent the seven quantized levels. After Huffman coding, we can calculate the expected code word

length

$$E[l] = \sum_{i=1}^{7} l_i P_i$$

in our example, resulting in an expected reduction ratio of 3:2.65.

The reconstruction process begins at the root of the tree. If bit 1 is received, we traverse down the left branch, otherwise the right branch. We continue traversing until we reach a node with no child. We then output the symbol corresponding to this node and begin traversal from the root again.

The reconstruction process of Huffman coding perfectly recovers the original data. Therefore it is a lossless algorithm. However, a transmission error of a single bit may result in more than one decoding error. This propagation of transmission error is a consequence of all algorithms that produce variable- length code words.

**Questions:**

1. With suitable example discuss about the direct data compression techniques.
2. Give a detailed view of AZTEC algorithm.
3. Explain in detailed about the Cortes Algorithm for data reduction.
4. Explain the different steps of Karhunen-Loeve Transform.
5. Explain data compression techniques using Differential pulse code modulation (DPCM)
6. Discuss in details the procedures for data compression using Huffman coding.

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 5 – FILTERING – SBMA5202

**UNIT 5 - FILTERING**

**5.1.Time domain filtering**

Certain types of noise may be filtered directly in the time domain using digital filters. Advantage is spectral characterization of the signal and noise not required—atleast in a direct manner.Time-domain processing may also be faster than frequency-domain filtering.

**Synchronous averaging**

Linear filters fail when the signal and noise spectra overlap. Synchronized signal averaging can separate are repetitive signal from noise without distorting the signal. ERP or SEP epochs may be obtained a number of times by repeated application of the stimulus; averaged using the stimulus as trigger to align the epochs. If noise is random with zero mean, uncorrelated with signal, averaging will improve the SNR. yk(n): one realization of a signal, with k =1,2,...,M representing the ensemble index,

n =1,2,...,N representing the time-sample index. M:number of copies, events, epochs, or realizations.

N:number of samples in each signal. yk(n)= xk(n)+ ηk(n), xk(n):original uncorrupted signal, ηk(n):noise in kth copy of signal.

$$\sum_{k=1}^{M} y_k(n) = \sum_{k=1}^{M} x_k(n) + \sum_{k=1}^{M} \eta_k(n);\ n = 1, 2, \ldots, N$$

For each instant of time n,add M copies of signal:

If the repetitions of the signal are identical and aligned,

$$\Sigma_{k=1}^{M} x_k(n) = Mx(n)$$

If noise is random, has zero mean and variance $\sigma^2$,

$$\Sigma_{k=1}^{M} \eta_k(n) \rightarrow 0$$

as M increases, with a variance of $M\sigma^2$.

RMS value of noise in the averaged signal = $\sqrt{M}\sigma\eta$. Thus SNR of signal increases by M $\sqrt{M}$ or $\sqrt{M}$.

Larger the number of epochs or realizations averaged, better the SNR of the result. Synchronized

averaging is a type of ensemble averaging. Algorithmic description of synchronized averaging: Obtain number of realizations of signal or event.

Determine reference point for each realization. Trigger, stimulus, QRS in ECG, etc. 3.Extract parts of the signal corresponding to the events and add them to a buffer. Various parts may have different durations. Alignment of copies at trigger point is important; the tail ends of all parts may not be aligned. 4.Divide the result by the number of events added.

**Moving average filters**

When ensemble of several realizations are not available, synchronized averaging is not possible. Temporal averaging for nois eremoval is used. Temporal statistics used instead of ensemble statistics Temporal window of samples moved to obtain

Output at various points of time: moving-window averaging or moving-average (MA) filter.

Average weighted combination of samples General form of MA filter:

$$y(n) = \sum_{k=0}^{N} b_k \, x(n-k)$$

x and y: input and output of filter.

bk: filter coefficients or tap weights. N:order of filter.

Effect of division by the number of samples used (N +1) Included in the values of the filter coefficients

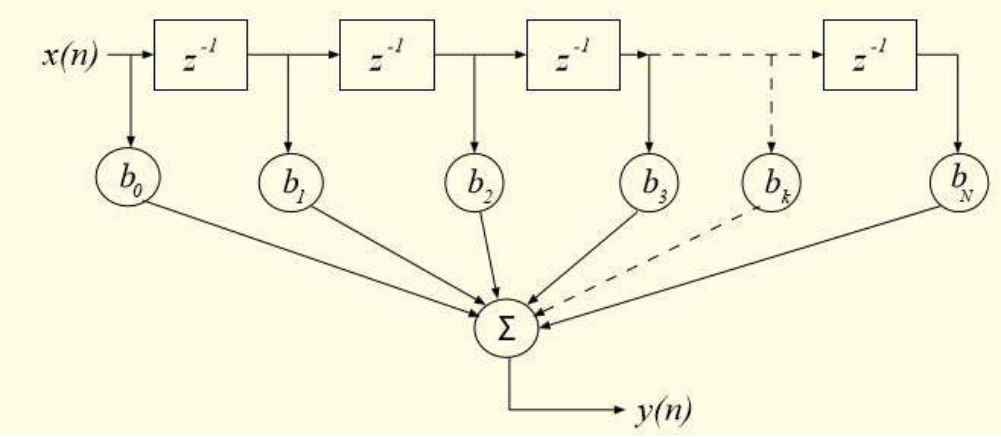



**Fig. 5.1 Signal-flowdiagram of a moving-average filter of order N. Applying the z-transform, we get the transfer function**

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N} b_k \, z^{-k} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}$$

X(z) and Y (z): z-transforms of x(n) and y(n)

MA filter to remove noise—on Hann or Hanning filter

$$y(n) = \frac{1}{4}[x(n) + 2x(n-1) + x(n-2)].$$

Impulse response: let x(n)= δ(n).

$$h(n) = \tfrac{1}{4}[\delta(n) + 2\delta(n-1) + \delta(n-2)].$$

Transfer function of the Hanning filter:

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

Double-zero at z = −1.

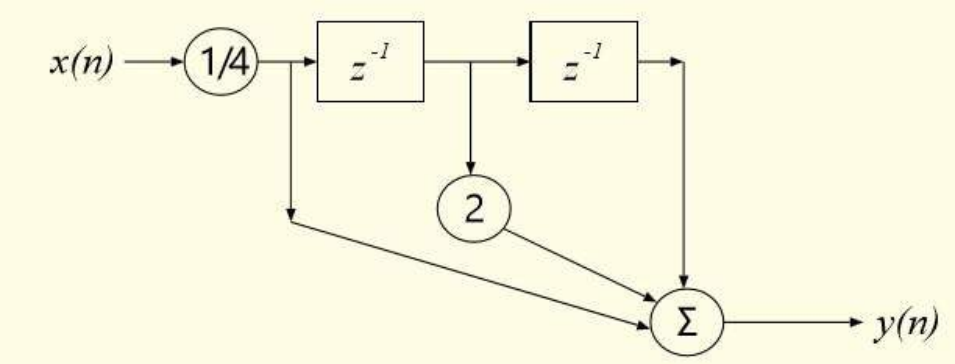



**Fig.5.2 Signal-flow diagram of the Hanning filter**

An MA filter is a finite impulse response (FIR) filter:

Impulse response h(k) has a finite number of terms: h(k)= bk, k =0,1,2,...,N. An FIR filter may be realized non-recursively with no feedback.

Output depends only on the present input sample and a few past input samples.

Filter is a set of tap weights of the delay stages. Transfer function has no poles except at z =0: the filter is inherently stable.

Filter has linear phase if the series of tap weights is symmetric or antisymmetric. Frequency response:

substitute z = exp(jωT) in H(z),

T: sampling interval in seconds, T =1/fs, f:frequencyin Hz, fs: sampling frequency, ω:radian

frequency, ω =2πf

Set T =1 anddeal with normalized frequency in the range $0 \leq \omega \leq 2\pi$ or $0 \leq f \leq 1$;

then f =1 or ω =2π represents the sampling frequency, lower frequency values r epresented as normalized fraction of fs.

$$H(\omega) = \frac{1}{4}[1 + 2e^{-j\omega} + e^{-j2\omega}]$$

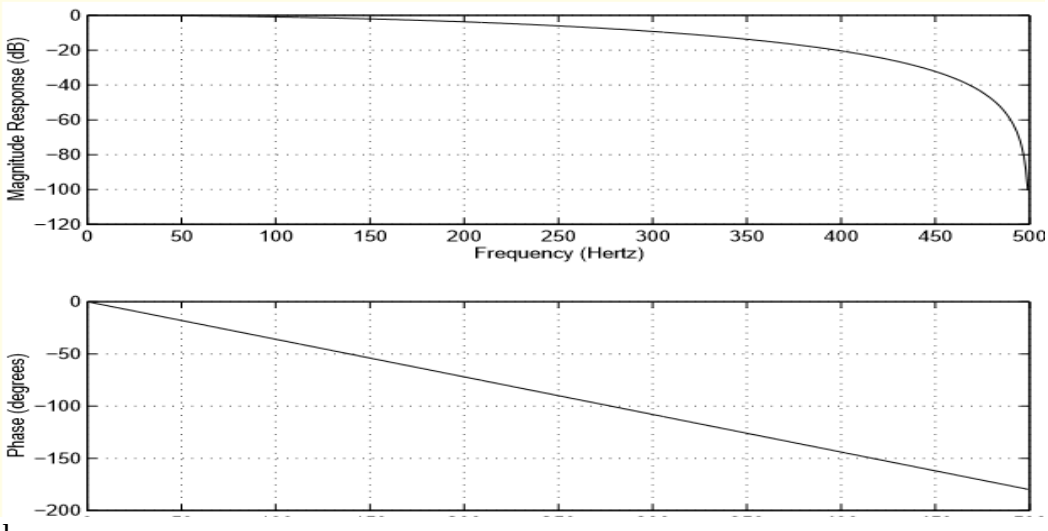
Frequency response of the Hanning filter:

Letting exp(−jω) =cos(ω)−j sin(ω),

$$H(\omega) = \frac{1}{4}[\{2 + 2\cos(\omega)\}e^{-j\omega}]$$

$$|H(\omega)| = \left|\frac{1}{2}\{1 + \cos(\omega)\}\right|$$

$$\angle H(\omega) = -\omega.$$




Magnitude and phase responses:

**Fig. 5.3 Magnitude and phase responses of the Hanning (smoothing) filter**

## 5.2 Frequency domain filters

Filters may be designed in the frequency domain to provide specific low pass, highpass, bandpass, or band-reject (notch) characteristics. Implemented in software after obtaining FT of input signal, or converted into equivalent time-domain filters. Most commonly used designs are Butterworth, Chebyshev, elliptic, and Besselfilters.Well-established in the analog-filter domain: commence with ananalog design H(s) and apply the bilinear transformation to obtain a digital filter H(z) Frequency-domain filters may also be specified directly in terms of the values of the desired frequency response at certain frequency samples only, then transformed into the equivalent time-domain filter coefficients via the inverse Fourier transform.

**Design of Butterworth filters**

**Removal of high frequency noise: Butterworth low pass filter**

Butterworth filters has maximally flat magnitude response in the pass-band. For butterworth lowpass filter of order N, frst 2N −1 derivatives of squared magnituderesponse are zero at Ω=0,where Ω =analog radian frequency. Butterworth filter response is monotonic in the pass-band as well as in the stop-band.

Basic Butterworth low pass filter function is given as

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{j\Omega}{j\Omega_c}\right)^{2N}}$$

where Ha is the frequency response of the analog filter and

Ωc is the cut off frequency in radians/s. Butterworth filter completely specified by Cut off frequency Ωc andorder N. As the order N increases, the filter response becomes more flat in the pass-band, and transition to the stop-band becomesfaster or sharper.

$$|H_a(j\Omega_c)|^2 = \tfrac{1}{2} \text{ for all } N.$$

ChangingtotheLaplacevariable s,

$$H_a(s)H_a(-s) = \frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}}$$

Polesofsquaredtransferfunctionlocatedwithequalspacingaroundacircleofradius Ωc in the s-plane, distributed symmetrically on either side of imaginary axis s = jΩ. Nopole on imaginary axis; poles on real axis for odd N. Angular spacing between poles is πN.If Ha(s)Ha(−s) has a pole at s = sp, it will have a pole at s = −sp as well.

For the filter coefficients to be real,complex poles must appear in conjugate pairs.

To obtain a stable and causal filter, form Ha(s) with only the N poles on the left- handside of the s-plane. Pole positions in the s-plane given by

$$s_k = \Omega_c \exp\left[j\pi\left(\frac{1}{2}+\frac{(2k-1)}{2N}\right)\right]$$

k =1,2,...,2N

Once pole positions obtained in the s-plane, derive the transfer function in the analog Laplace domain

$$H_a(s) = \frac{G}{(s-p_1)(s-p_2)(s-p_3)\cdots(s-p_N)}$$

where pk, k =1,2,...,N,are the N poles in the left-half of the s-plane, and G is again factor specified as needed or to normalize the gain at DC(s =0) to be unity.

Bilineartransformation(BLT):

$$s = \frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right]$$

Butterworth circle in the s-plane maps to a circle in the z-plane with real axis intercepts at z = 2−ΩcT /2+ΩcT and z = 2+ΩcT/ 2−ΩcT

Polesat s = sp and s = −sp inthe s-plane map to z = zp and z =1/zp.

Poles in the z-plane not uniformly spaced around the transformed Butterworth circle. For stability, all poles of H(z) must lie within the unit circle in the z-plane.

Unit circle in the z-plane: z = exp(jω). For points on the unit circle, we have

$$s = \sigma + j\Omega = \frac{2}{T}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right) = \frac{2j}{T}\tan\left(\frac{\omega}{2}\right)$$

For the unit circle, σ =0; therefore, continuous-time(s-domain) frequency variable Ω Related to discrete-time (z-domain) frequency variable ω as

Ω=2 Ttan(ω/2) and ω =2tan−1(ΩT/ 2)

Non linear relationship warps frequency values: mapped from the imaginary (vertical) axis in the s-plane to the unit circle in the z-plane should be taken in to account in specifying cut off frequencies.

**Removal of low-frequency noise: Butterworth high pass filters**

Butterworth high pass filter is specified directly in the discrete-frequency domain as

$|H(k)|^2 = 1/(1+ kc/ k)^{2N}$

**Optimal filtering: WienerFilter**

Wiener filter theory provides for optimal filtering by taking into account the statistical characteristics of the signal and noise processes. The filter parameters are optimized with reference to a performance criterion. The output is guaranteed to be the best achievable result under the conditions imposed and the information provided.

Single-input,single-output,FIR filter with real input signal values and real coefficients. Figure shows the signal-flow diagram of a transversal filter with coefficients or tap weights wi , i =0,1,2,...,M −1, input x(n),andoutput ˜ d(n).

Output d(n) =an estimate of some “desired” signal d(n) that represents the ideal, uncorrupted signal. If we assume that the desired signal is available, estimation error between the output and the desired signal:

e(n)= d(n)− ˜ d(n).

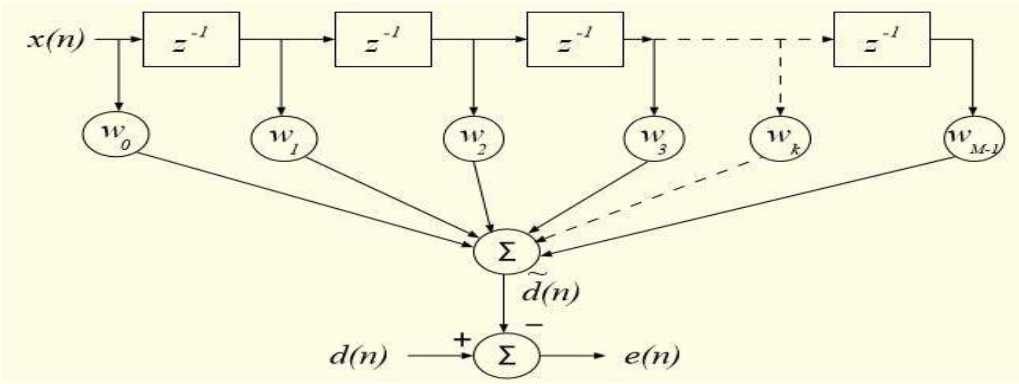



Fig.5.4 Block diagram of the Wiener filter

d(n) =output of a linear FIR filter = convolution of the input x(n) with the tap-weight sequence wi:

$$\tilde{d}(n) = \sum_{k=0}^{M-1} w_k \, x(n-k)$$

wi is also the impulse response of the filter.

For easier handling of the optimization procedures, the tap-weight sequence may be written as an M ×1tap-weightvector:

w =[w0,w1,w2,...,wM−1]T, where the bold-faced character w represents a vector; superscript T: vector transposition.

Tap weights convolved with M values of input. Write the M input values as an M ×1 vector: x(n)=[x(n),x(n−1),...,x(n−M +1)]T. Vector x(n) varies with time: at a given instant n the vector contains the current input sample x(n) and the

preceding (M−1) input samples x(n−1) to x(n−M +1).

Wiener filter theory estimates the tap-weight sequence that minimizes the MS value of the estimation error;

output= minimum mean-squared error (MMSE) estimate of the desired response: optimal filter.

Mean-squarederror (MSE) defined as J(w)= E[e2(n)]

= E[ {d(n)−$w^T$x(n)} {d(n)−$x^T$ (n)w}]

= E[$d^2$ (n)]−w $^T$E[x(n)d(n)]−E[d(n)$x^T$(n)]w + $w^T$ E[x(n)$x^T$(n)]w.

Assumption: input vector x(n) and desired response d(n) are jointly stationary. Then: $E[d^2(n)]$ =variance of d(n) = $\sigma^2$ d, assuming that the meanof d(n) is zero.

E[x(n)d(n)] = M×1 vector=cross-correlation between input vector x(n) and desired response d(n):

Θ = E[x(n)d(n)]

Θ =[θ(0),θ(−1),...,θ(1−M)]T, where

θ(−k)= E[x(n−k)d(n)],k =0,1,2,...,M −1

E[d(n)xT(n)] is the transpose of E[x(n)d(n)]; therefore $\Theta^T$ = $E[d(n)x^T(n)]$.

$E[x(n)x^T(n)]$= autocorrelation of input vector x(n) computed as the outer product of the vector with itself:

Φ = $E[x(n)x^T(n)]$

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^T\boldsymbol{\Theta} - \boldsymbol{\Theta}^T\mathbf{w} + \mathbf{w}^T\boldsymbol{\Phi}\mathbf{w}$$

MSE expression is simplified to

MSE is a second-order function of the tap-weight vector w. To determine the optimal tap- weight vector, denoted by wo, differentiate J(w) with respect to w, set it to zero, and solve the resulting equation

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = -2\boldsymbol{\Theta} + 2\boldsymbol{\Phi}\mathbf{w} \rightarrow 0.$$

Condition for the optimal filter:

Φwo = Θ.

Wiener-Hopf equation or the normal equation. Optimal Wiener filter:

wo = Φ−1 Θ.

**Adaptive noise cancellation**

Primary input orobservedsignal x(n) is a mixture of the signal of interest v(n) and the“primary noise” m(n):

x(n)= v(n)+ m(n)

It is desired that the interference or noise m(n) be estimated and removed from x(n) In order to obtain the signal of interest v(n)

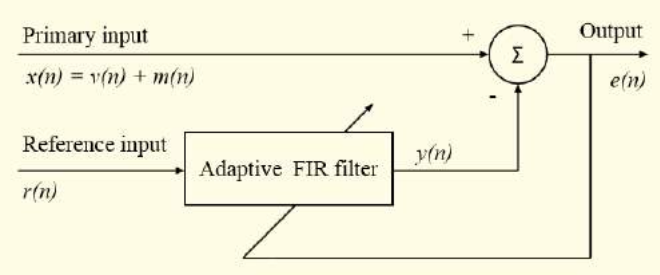



**Fig.5.5 :Block diagram of a generic adaptive noise canceler (ANC) or adaptive filter**

It is assumed that v(n) and m(n) are uncorrelated. ANC requires a second input:"reference input" r(n), uncorrelated with the signal of interest v(n) but closely related to or correlated with the interference or noise m(n) in some manner that need not be known. The ANC filters or modifies the reference input r(n) to obtain a signal y(n) as close to noise m(n) as possible y(n) is subtracted from primary input to estimate desired signal:

$\tilde{v}(n) = e(n) = x(n) - y(n)$.

Assume that the signal of interest v(n), the primary noise m(n), the reference input r(n), and the primary noise estimate y(n) are statistically stationary and have zero means.

Because v(n) is uncorrelated with m(n) and y(n) and all of them have zero means, we have

$E[v(n)\{m(n) - y(n)\}] = E[v(n)]\, E[m(n) - y(n)] = 0$.

$E[e^2(n)] = E[v^2(n)] + E[\{m(n) - y(n)\}^2]$.

Output e(n) used to control the adaptive filter. In ANC, the objective is to obtain an output e(n) that is a least squares fit to the desired signal v(n). Achieved by feeding the output back to the adaptive filter and adjusting the filter to minimize the total output power.

System output: error signal for the adaptive process. Signal power $E[v^2(n)]$ unaffected as the filter is adjusted to minimize $E[e^2(n)]$; the minimum output power is $\min E[e^2(n)] = E[v^2(n)] + \min E[\{m(n) - y(n)\}^2]$.As the filter is adjusted so that $E[e^2(n)]$ is minimized, $E[\{m(n) - y(n)\}^2]$ is minimized.

Thus the filter output y(n) is the MMSE estimate of the primary noise m(n). Moreover, when $E[\{m(n) - y(n)\}^2]$ is minimized, $E[\{e(n) - v(n)\}^2]$ is minimized, because e(n) − v(n) = m(n) − y(n). Adapting the filter to minimize the total output power is equivalent to causing the output e(n) to be the MMSE estimate of the signal of interest v(n) for the given structure and adjustability of the adaptive filter and for the given reference input. Output e(n) contains signal of interest v(n) and noise. The output noise is given by e(n) − v(n) = ˜v(n) − v(n) = m(n) − y(n).

Minimizing $E[e^2(n)]$ minimizes $E[\{m(n) - y(n)\}^2]$;

Therefore minimizing the total output power minimizes the output noise power. Because the signal component v(n) in the output remains unaffected, minimizing the total output power maximizes the output SNR. output power is minimum when $E[e^2(n)] = E[v^2(n)]$.

When this condition is achieved, E[{m(n) − y(n)}2] = 0.

We then have y(n) = m(n) and e(n) = v(n):

then, the output is a perfect and noise free estimate of the desired signal. Optimization of the filter may be performed by expressing the error in terms of the tap weight vector and applying the procedure of choice.

$$y(n) = \sum_{k=0}^{M-1} w_k \, r(n-k)$$

wk, k = 0, 1, 2, . . . ,M − 1, are the tap weights, and M is the order of the filter.
Estimation error e(n) or output of ANC:

e(n) = x(n) − y(n).Define the tap weight

vector at time n as $w(n) = [w0(n), w1(n), \ldots, wM-1(n)]^T$ . Tap input vector at each time instant n: $r(n) = [r(n), r(n-1), \ldots, r(n-M+1)]^T$. Then, estimation error e(n):

e(n) = x(n) − wT (n) r(n).

### 5.3 LMS and RLS algorithms in adaptive filtering LMS Algorithm

Adjust the tapweight vector to minimize the MSE. Squaring the estimation error e(n)

$$e^2(n) = x^2(n) - 2\,x(n)\,\mathbf{r}^T(n)\,\mathbf{w}(n) + \mathbf{w}^T(n)\,\mathbf{r}(n)\,\mathbf{r}^T(n)\,\mathbf{w}(n).$$

Squared error is a second order or quadratic function of the tap weight vector and the inputs, and may be depicted as a concave hype rparaboloidal, bowl like surface. Aim of optimization: reach the bottom of the bowl like function. Gradient based methods may be used for this purpose. Taking the expected values of the entities and the derivative with respect to the tap weight vector, we may derive the WienerHopf equation for the ANC. The LMS algorithm takes a simpler approach: assume the square of the instantaneous error to stand for an estimate of the MSE.

LMS algorithm based on the method of steepest descent:

new tapweight vector w(n + 1) given by the present tapweight vector w(n) plus a correction proportional to the negative of the gradient Δ(n) of the squared error:

w(n + 1) = w(n) − μ Δ (n). Parameter μ controls stability and rate of convergence: larger the value of μ, larger is the gradient of the error

that is introduced, and the faster is the convergence. LMS algorithm approximates Δ (n) by the derivative of the squared error with respect to the tapweight vector as

$$\tilde{\nabla}(n) = -2\,x(n)\,\mathbf{r}(n) + 2\,\{\mathbf{w}^T(n)\,\mathbf{r}(n)\}\,\mathbf{r}(n)$$

$$= -2\,e(n)\,\mathbf{r}(n)$$

Using this estimate of the gradient we get w(n + 1) = w(n) + 2 μ e(n) r(n).This is known as the WidrowHoff LMS algorithm.

Advantages of LMS algorithm:

simplicity and ease of implementation. Although the method is based on the MSE and gradient based optimization, the filter expression itself is free of differentiation, squaring, or averaging.

**RLS Algorithm**

RLS was discovered by Gauss but laid unused or ignored until 1950 when Plackett rediscovered the original work of Gauss from 1821. In general, the RLS can be used to solve any problem that can be solved by adaptive filters. For example, suppose that a signal d(n) is transmitted over an echoey, noisy channel that causes it to be received as

$$x(n) = \sum_{k=0}^{q} b_n(k) d(n-k) + v(n)$$

where $v(n)$ represents additive noise. We will attempt to recover the desired signal $d(n)$ by use of a $p+1$ -tap FIR filter

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k) x(n-k) = \mathbf{w}_n^T \mathbf{x}_n$$

where $\mathbf{x}_n = [x(n) \quad x(n-1) \quad \dots \quad x(n-p)]^T$ is the vector containing the $p+1$ most recent samples of $x(n)$. Our goal is to estimate the parameters of the filter $\mathbf{w}$, and at each time *n* we refer to the new least squares estimate by $\mathbf{w}_n$ As time evolves, we would like to avoid completely redoing the least squares algorithm to find the new estimate for $\mathbf{w}_{n+1}$, in terms of $\mathbf{w}_n$

The benefit of the RLS algorithm is that there is no need to invert matrices, thereby saving computational power. Another advantage is that it provides intuition behind such results as the Kalman filter.

he idea behind RLS filters is to minimize a cost function $C$ by appropriately selecting the filter coefficients $\mathbf{w}_n$, updating the filter as new data arrives. The error signal $e(n)$ and desired signal $d(n)$

are defined in the negative feedback diagram below:

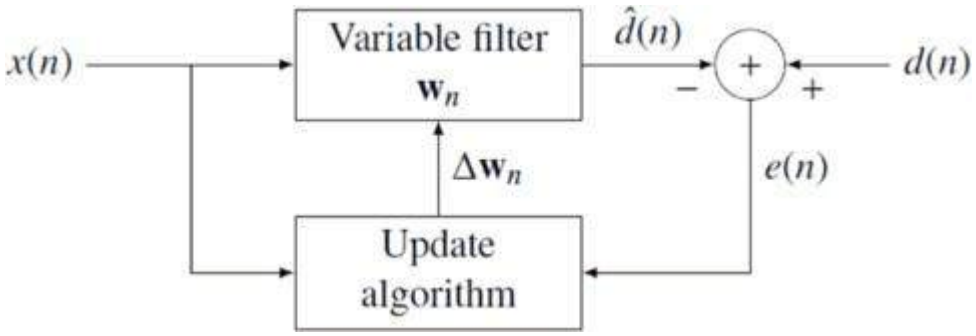



**Fig.5.5. Block diagram for RLS algorithm**

The error implicitly depends on the filter coefficients through the estimate $\hat{d}(n)$:

$$e(n) = d(n) - \hat{d}(n)$$

The weighted least squares error function $C$—the cost function we desire to minimize—being a function of e(n) is therefore also dependent on the filter coefficients:

$$C(\mathbf{w}_n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$$

where $0 < \lambda \leq 1$is the "forgetting factor" which gives exponentially less weight to older error samples.

The cost function is minimized by taking the partial derivatives for all entries $k$of the coefficient vector $\mathbf{w}_n$ and setting the results to zero

$$\frac{\partial C(\mathbf{w}_n)}{\partial w_n(k)} = \sum_{i=0}^{n} 2\lambda^{n-i} e(i) \frac{\partial e(i)}{\partial w_n(k)} = -\sum_{i=0}^{n} 2\lambda^{n-i} e(i)\, x(i-k) = 0 \qquad k = 0, 1, \cdots, p$$

Next, replace $e(n)$ with the definition of the error signal

$$\sum_{i=0}^{n} \lambda^{n-i} \left[ d(i) - \sum_{l=0}^{p} w_n(l) x(i-l) \right] x(i-k) = 0 \qquad k = 0, 1, \cdots, p$$

Rearranging the equation yields

$$\sum_{l=0}^{p} w_n(l) \left[ \sum_{i=0}^{n} \lambda^{n-i}\, x(i-l) x(i-k) \right] = \sum_{i=0}^{n} \lambda^{n-i} d(i) x(i-k) \qquad k = 0, 1, \cdots, p$$

This form can be expressed in terms of matrices

$$\mathbf{R}_x(n)\, \mathbf{w}_n = \mathbf{r}_{dx}(n)$$

where $\mathbf{R}_x(n)$ is the weighted sample covariance matrix for $x(n)$, and $\mathbf{r}_{dx}(n)$ is the equivalent estimate for the cross-covariance between $d(n)$ and $x(n)$. Based on this expression we find the coefficients which minimize the cost function as

$$\mathbf{w}_n = \mathbf{R}_x^{-1}(n)\, \mathbf{r}_{dx}(n)$$

## 6 Application of these techniques in removal of artifacts in bio-signals.

### Adaptive Cancellation of the Maternal ECG to Obtain the Fetal ECG

Widrow et al. proposed a multiple reference ANC for removal of the maternal ECG in order to obtain the fetal ECG. Combined ECG obtained from a single abdominal lead. Maternal ECG was obtained via four chest leads

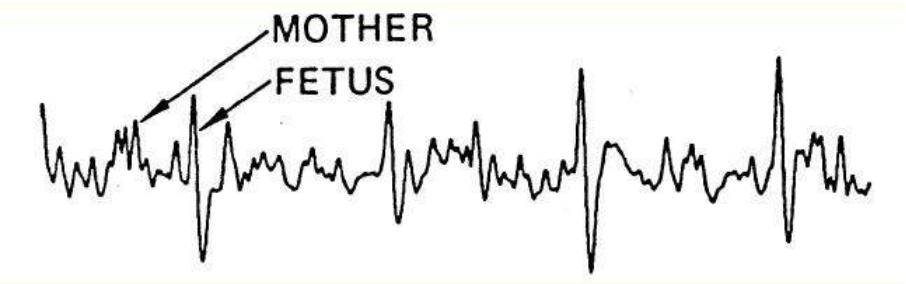



**Fig.5.6 Result of adaptive cancellation of the maternal chest ECG from**

**Questions:**

1. Explain about the Linear shift invariant filters with block diagram.
2. Give the function of individual block of Moving average filters with block diagram.
3. Explain in detail about the Signal flow diagram and transfer function of Hanning filter
4. Justify how the LMS algorithms is used in the removal of artifacts
5. Justify how the RLS algorithms is used in the removal of artifacts
6. Give the block diagram of Wiener filter
7. Explain in different steps that how the Adaptive filter is work for noise cancellation
8. Design a butterworth filter for the removal of high frequency noise and low frequency noise.