CHAPTER 3

EXISTING ADAPTIVE FILTERING TECHNIQUE
FOR FETAL ECG SIGNAL EXTRACTION

This chapter gives a detailed study of adaptive filtering techniques and the wavelet transform techniques to cancel maternal ECG signal from the abdominal ECG signal.

3.1 INTRODUCTION OF AN ADAPTIVE FILTER

The term filter is used in a system that is designed to extract information about a region of interest from noisy data. The noise can be internal or external to the system due to interfering signals originating from other sources.

The design of the Weiner filter requires prior information about the statistics of the data to be processed. The filter is optimum only when the statistical characteristics of the input data match the prior information on which the design of the filter is based. When this information is not known completely, it may not be possible to design the Weiner filter or else the design may no longer be optimum. This is a two-stage process whereby the filter first “estimates” the statistical parameters of the relevant signals and then “plugs” the obtained results into a non-recursive formula for computing the filter parameters. For real-time operation, this procedure has a disadvantage of requiring excessively elaborate and costly hardware. To overcome this limitation, an adaptive filter may be used.
Digital signal processing (DSP) has been a major player in the current technical advancements such as noise filtering, system identification, and voice prediction. Standard DSP techniques, however, are not enough to solve these problems quickly and obtain acceptable results. Adaptive filtering techniques must be implemented to promote accurate solutions and a timely convergence to that solution.

Figure 3.1 shows general block diagram of adaptive filter. An adaptive filter is defined as a filter whose characteristics can be altered to achieve some objective automatically, without the need for substantial intervention by the user. It is also assumed that the time scale of the modification is very slow compared to the bandwidth of the signal being filtered. Implicit in this assumption is that the system designer could in fact use a time-invariant adaptive filter if only the designer knew enough about the input signals to design the filter before its use. This lack of knowledge may spring from true uncertainty about the characteristics of the signal when the filter is turned on, or because of the slow change in the characteristics of the input signal during the filter’s operation.

In this configuration, the input \( x(n) \), a noise source \( N_1(n) \), is compared with a desired signal \( d(n) \), which consists of a signal \( s(n) \) corrupted by another noise \( N_0(n) \). The adaptive filter coefficients adapt to cause the error signal to be a noiseless version of the signal \( s(n) \). Both of the noise signals for this configuration need to be uncorrelated to the signal \( s(n) \). In addition, the noise sources must be correlated to each other in some way, preferably equal, to get the best results.

Due to the nature of the error signal, the error signal will never become zero. The error signal should converge to the signal \( s(n) \), but not converge to the exact signal. In other words, the difference between the signal \( s(n) \) and the error signal \( e(n) \) will always be greater than zero. The only option is to minimize the difference between those two signals.
By such a system one that is self-designing in that the adaptive filter relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. The algorithm starts from some predetermined set of initial conditions, representing whatever the environment. Yet, in a stationary environment, after successive iterations of the algorithm, it converges to the optimum wiener solution in some statistical sense. In a non-stationary environment, the algorithm offers a tracking capability, in that it can track time variations in the statistics of the input data, provided the variations are sufficiently slow. An adaptive filter is said to be linear if its input-output map obeys the principle of superposition whenever its parameters are held fixed. Otherwise, the adaptive filter is said to be nonlinear. A wide variety of recursive algorithms have been developed in the literature for the operation of the linear adaptive filters.

### 3.2 PERFORMANCE MEASURES IN ADAPTIVE SYSTEMS

The following performance measures are used to determine the best technique over other techniques:

- Convergence rate
- Minimum mean square error
- Computational complexity

![Figure 3.1 General block diagram of adaptive filter](image)
• Stability
• Robustness
• Filter length
• Numerical properties
• Tracking

3.3 ADAPTIVE FILTER SELECTION

Three important factors such as performance, robustness and computational efficiency should be considered by a system designer for choosing an appropriate adaptive filter for the application of interest. Computer simulation provides a detailed analysis and investigation of these issues. The adaptive LMS algorithm is relatively simple to implement, yet powerful enough to evaluate the practical benefits that may result from the application of adaptive to the problem at hand.

Practical applications of adaptive filtering are highly diverse, with each application having peculiarities of its own. Thus, the solution for one application may not be suitable for another.

3.4 CONFIGURATIONS OF ADAPTIVE FILTER

The four major types of adaptive filtering configurations are as follows:

• Adaptive system identification
• Adaptive noise cancellation
• Adaptive linear prediction
• Adaptive inverse system.
All of the above systems are similar in the implementation of the algorithm, but different in system configuration. All 4 systems have the same general parts; an input $x(n)$, a desired result $d(n)$, an output $y(n)$, an adaptive transfer function $w(n)$, and an error signal $e(n)$ which is the difference between the desired output $u(n)$ and the actual output $y(n)$. In addition to these parts, the system identification and the inverse system configurations have an unknown linear system $u(n)$ that can receive an input and give a linear output to the given input.

### 3.5 Adaptive Filtering Technique Using Various Adaptive Algorithms

A number of filter algorithms are discussed in this section; the finite impulse response (FIR) least mean squares (LMS) gradient approximation method are discussed in detail and characteristics of infinite impulse response (IIR) adaptive filters are briefly discussed.

#### 3.5.1 FIR Algorithms using LMS Gradient Approximation Method

Given an adaptive filter with an input $x(n)$, an impulse response $w(n)$ and an output $y(n)$, a mathematical relation for the transfer function of the system is given as the time domain coefficients for an $N$th order FIR filter. Note in the above equation and throughout a boldface letter represents a vector and the super script $T$ represents the transpose of a real valued vector or matrix.

\[
y(n) = w^T(n)x(n) \quad (3.1)
\]

\[
x(n) = [x(n), x(n-1), x(n-2), ..., x(n-(n-1))]w^T(n)x(n) \quad (3.2)
\]

where \[ w^T(n) = [w_0(n), w_1(n), w_2(n) ... w_{N-1}(n)] \] (3.3)
Using an estimate of the ideal cost function, the following equation can be derived.

\[ w(n+1) = w(n) - \mu \Delta_{E[e]^2}(n) \quad (3.4) \]

In the above equation \( w(n+1) \) represents the new coefficient values for the next time interval, \( \mu \) is a scaling factor, and \( \Delta_{E[e]^2}(n) \) is the ideal cost function with respect to the vector \( w(n) \). From the above formula, one can derive the estimate for the ideal cost function

\[ w(n+1) = w(n) - \mu e(n)x(n) \quad (3.5) \]

where \( e(n) = d(n) - y(n) \quad (3.6) \)

\[ y(n) = x^T(n)w(n) \quad (3.7) \]

In the above equation, \( \mu \) is sometimes multiplied by 2, but here it is assumed that it is absorbed by the \( \mu \) factor.

In summary, in the Least Mean Squares Gradient Approximation Method, often referred to as the Method of Steepest Descent, a guess based on the current filter coefficients is made, and the gradient vector, the derivative of the MSE with respect to the filter coefficients, is calculated from the guess. Then a second guess is made at the tap-weight vector by making a change in the present guess in a direction opposite to the gradient vector. This process is repeated until the derivative of the MSE is zero.

3.5.1.1 Convergence of the LMS adaptive filter

The convergence characteristics of the LMS adaptive filter is related to the autocorrelation of the input process as defined by
\[ R_x = E[x(n)x^T(n)] \] (3.8)

These are the two conditions that must be satisfied in order for the system to converge. These conditions include:

- The autocorrelation matrix, \( R_x \), must be positive definite.
- \( 0 < \mu < 1/\lambda_{\text{max}} \), where \( \lambda_{\text{max}} \) is the largest eigenvalue of \( R_x \).

In addition, the rate of convergence is related to the eigenvalue spread. This is defined using the condition number of \( R_x \), defined as \( \kappa = \lambda_{\text{max}} / \lambda_{\text{min}} \), where \( \lambda_{\text{min}} \) is the minimum eigenvalue of \( R_x \). The fastest convergence of this system occurs when \( K = 1 \), corresponding to white noise. This states that the fastest way to train a LMS adaptive system is to use white noise as the training input. As the noise becomes more and more colored, the speed of the training will decrease.

### 3.5.2 Quasi-Newton Adaptive Algorithm

The quasi-Newton adaptive algorithm uses second order statistics to reduce the convergence rate of an adaptive filter, via the Gauss-Newton method. Probably the best known quasi-Newton algorithm is the recursive least squares (RLS) algorithm. It is important to note that even with the increase in convergence rate, the RLS algorithm requires great amount of processing power, which can make it difficult to implement on real-time systems. There are a number of other quasi-Newton algorithms that have fast convergence rates, and that are also feasible alternatives for real time processing.
3.5.3 IIR Adaptive Filters

The primary advantage of IIR filters is that to produce an equivalent frequency response to an FIR filter, they can have a fewer number of coefficients. This in theory should reduce the number of adds, multiplies and shifts to perform a filtering operation. This theory of using IIR filters to reduce the computational burden is the primary motivation for the use of IIR adaptive filters. There are, however, a number of problems that are introduced with the use of IIR adaptive filters.

- The fundamental concern with IIR adaptive filters is the potential for instability due to poles moving outside the unit circle during the training process. Even if the system is initially stable and the final system is stable, there is still the possibility of the system going unstable during the convergence process. Some suggestion has been made to limit the poles to within the unit circle, however, this method requires that the step sizes be small, which considerably reduces the convergence rate.

- Due to the interplay between the movement of the poles and zeros, the convergence of IIR systems tends to be slow. The result is that even though IIR filters have fewer coefficients, therefore few calculations per iteration, the number of iterations may increase cause a net loss in processing time to convergence. This, however, is not a problem with all pole filters.

- In an IIR system, the MSE surface may contain local minimum that can cause a convergence of that system to the local minimum instead of the absolute minimum. More care
need to be taken in the initial conditions in IIR adaptive filters than in FIR adaptive filters.

- IIR filters are more susceptible to coefficient quantization error than FIR, due to the feedback.

There have been a number of studies done on the use of IIR adaptive filters, but due to the problems stated above, they are still not widely used in industry today.

### 3.5.4 Simulation Study of Adaptive Filters

Usual practice to select fetal ECG recovery model is through visual inspection of the algorithm performance. Since the maternal ECG signal contribution to the abdominal signal is unknown, a method to generate simulated signals can be adopted. Maternal ECG signals are simulated based on MIT BH database, Daisy database and Physionet. A 50 Hz power interference signal and random noise is simulated and it is added to the reference signal.

These synthetic ECG signals constitute the methodology for the identification of the model and give many advantages as follows:

- To obtain accurate measures using various models for getting different situations with which more robust solutions can be obtained.
- To obtain wide range of optimal and independent parameters to provide substantial training and information.

The maternal ECG signal obtained from thoracic area is simulated as shown in Figure 3.2.
Figure 3.2 Simulated Maternal ECG Signal

The fetal ECG signal is simulated as shown in Figure 3.3.

Figure 3.3 Simulated Fetal ECG Signal

The measurable signal obtained from the abdominal area of the pregnant woman which has MECG signal, FECG signal and noises is simulated as shown in Figure 3.4.

Figure 3.4 Simulated Abdominal Signal (MECG + FECG +noise)
All the above simulated signals are used to represent the real life conditions and help in diagnosing various congenital heart problems at the earliest.

3.6 ADAPTIVE NEURO FUZZY INFERENCE SYSTEM

The adaptive neuro fuzzy inference technique is used to cancel the MECG. It combines the advantages of neural network and fuzzy logic technique. Due to the adaptation capability of neural network, even if we have a single reference signal without considering the sensitivity of the electrode position, it is possible to estimate the MECG present in the abdominal signal. Since neural network takes longer time for convergence, the fuzzy logic technique is combined for decision making and verification purposes. The efficiency of the technique is proved by the experiments carried out with the simulated signals using conventional methods of LMS filter, adaptive neural networks and ANFIS.

ANC is a process by which the interference signal can be cancelled by identifying a non linear model between a measurable noise source and the corresponding immeasurable interference. This is an extremely useful method when a signal is submerged in a very noisy environment. The basic idea of this method is to pass the corrupted signal (abdominal) through a filter that tends to suppress the interference while leaving the signal unchanged. This is an adaptive process, which means it does not require prior knowledge of signal or interference characteristics.

ANC using linear filters have been used successfully in real world applications such as interference canceling in Electrocardiograms, echo elimination on long distance telephone transmission lines, and antenna side lobe interference canceling. This concept of linear adaptive noise cancellation can be extended to non-linear realms by using nonlinear adaptive systems.
Thus, ANFIS, which is one such nonlinear adaptive system is used to estimate an unknown interference present in the FECG signal.

Over the last few decades, neural networks and fuzzy systems have established their reputation as alternative approaches to signal processing. Both have certain advantages over conventional methods, especially when vague data or prior knowledge is involved. However, their applicability suffered from several weaknesses of the individual models. Neural networks recognize patterns and adapt themselves to cope with changing environments. Fuzzy inference systems (FIS) incorporate human knowledge and perform inferencing and decision-making.

ANFIS takes the advantages of the combination of neural network and fuzzy logic. The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a fuzzy system to represent knowledge in an interpretable manner, in addition to possessing the learning ability of a neural network to optimize its parameters. The drawbacks of both of the individual approaches - the black box behavior of neural networks, and the problems of finding suitable membership values for fuzzy systems could thus be avoided.

3.6.1 Membership Function Used in ANFIS

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. In this study, generalized bell type MF is used for tuning the FIS parameters. This is shown in Figure 3.5. It is specified by three parameters namely the width, centre and slope of the curve.

As the values of these parameters change, the bell shaped functions vary accordingly. It has the advantage of smoothness and concise notation.
3.6.2 FIS Structure and MF Parameter Adjustment

The basic structure of a fuzzy inference system maps input characteristics to input membership functions, input membership function to rules, rules to a set of output characteristics, output characteristics to output membership functions, and the output membership function to a single-valued output or a decision associated with the output. In a conventional fuzzy inference system, an expert who is similar with the target system to be modeled determines the number of rules. In cases where there are no experts available, the number of membership functions assigned to each input is chosen empirically. Also, the fuzzy inference system is applied to modeling systems whose rule structure is essentially predetermined by the user's interpretation of the characteristics of the variables in the model. Here, the shape of the membership functions depends on parameters, and changing these parameters will change the shape of the MF. Instead of just looking at the data to choose the MF parameters, MF parameters can be chosen automatically using ANFIS. Hybrid learning algorithm is used in ANFIS to train the network parameters. It combines the gradient method and the least square estimate to identify the MF parameters.

Though the gradient method is applied to identify the parameters in an adaptive network, the method is generally slow and is likely to become trapped in local minima. Hence, the hybrid rule is used which decreases the
dimension of the search space in the gradient method and also cuts down substantially the convergence time. Since the ANFIS architecture is a multilayer network, gradient method learning rule is used to tune the parameters in the hidden layer and the parameters in the output layer can be identified by the least squares method.

### 3.6.3 ANFIS Architecture

Adaptive Neuro Fuzzy Inference System (ANFIS) is a fuzzy inference system implemented in the framework of adaptive networks. ANFIS can construct input-output mapping based on both human knowledge in the form if-then rules and stipulated input-output pairs by using a hybrid learning procedure. Let us assume that the fuzzy inference system contains 2 inputs \( x \) and \( y \) and one output \( z \). The rule base contains two fuzzy if-then rules of Takagi and Sugeno’s type is given by

**Rule 1**: If \( x \) is A1 and \( y \) is B1, then \( f_1 = p_1x + q_1y + r_1 \).

**Rule 2**: If \( x \) is A2 and \( y \) is B2, then \( f_2 = p_2x + q_2y + r_2 \).

Then the ANFIS architecture is shown in Figure 3.6. The node functions in the same layer are of the same function family as described below.

**Layer 1**: Every node \( i \) in this layer is a square node with a node function

\[
O_i^1 = \mu A_i(x) \quad (3.9)
\]

where \( x \) is the input node \( i \), and \( A_i \) is the linguistic label (small, large, etc…) associated with this node function. In other words, \( O_i^1 \) is the membership function of \( A_i \) and it specifies the
degree to which the given $x$ satisfies the quantifier $A_i$. Usually $\mu A_i(x)$ is chosen to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as

$$\mu A_i(x) = \frac{1}{1 + \left(\frac{x-c_i}{a_i}\right)^{2b_i}} \tag{3.10}$$

Or

$$\mu A_i(x) = \exp\left\{-\left(\frac{x-c_i}{a_i}\right)^2\right\} \tag{3.11}$$

---

**Figure 3.6 ANFIS Architecture**

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic label $A_i$. In fact, any continuous and piecewise differentiable functions, such as commonly used trapezoidal or triangular- shaped membership functions, are also qualified candidates for node functions in this layer; Parameters in this layer are referred to as premise parameters.
**Layer 2**: Every node in this layer is a circle node labeled II which multiplies the incoming signals and sends the product out. For instance,

\[ w_i = \mu A_i(x) \ast \mu B_i(y), \ i \ 1,2 \ldots \]  

(3.12)

Each node output represents the firing strength of a rule. (In fact, other T-norm operators that performs generalized AND can be used as the node function in this layer)

**Layer 3**: Every node in this layer is a circle node labeled N. The \(i\)th node calculates the ratio of the \(i\)th rule’s firing strength to the sum of all rule’s firing strengths:

\[ \bar{w}_i = \frac{w_i}{w_1 + w_2}, \ i = 1,2 \ldots \]  

(3.13)

For convenience, output of this layer will be called normalized firing strengths.

**Layer 4**: Every node \(i\) in this is a square node with a node function

\[ O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \]  

(3.14)

where \(\bar{w}_i\) is the output of layer 3, and \(\{p_i, q_i, r_i\}\) is the parameter set. Parameters in the layer will be referred to as consequent parameters.

**Layer 5**: The signal node in this layer is a circle node labeled \(\Sigma\) that computes the overall output as the summation of all incoming signals, i.e.,

\[ O_i^5 = \text{overall output} = \sum_i \bar{w}_i f_i = \frac{\sum_i \bar{w}_i f_i}{\sum_i \bar{w}_i} \]  

(3.15)
The ANFIS can be trained by a hybrid learning algorithm which combines the gradient descent method and least square method. Each epoch of hybrid learning consists of forward pass and backward pass. In the forward pass, the algorithm uses least-squares method to identify the consequent parameters on the layer 4. In the backward pass, the errors are propagated backward and the premise parameters are updated by gradient descent.

### 3.6.4 FECG Extraction using ANFIS

In this study, the number of membership function for each input variable is determined by a trial and error process and the proposed 6 membership functions generating 36 fuzzy rules. The Flow chart for the extraction of FECG signal is shown in Figure 3.7. The inputs to the ANFIS are (1) abdominal signal (MECG +FECG) acting as the desired signal (2) thoracic signal (TECG) acting as the reference signal as shown in Figure 3.7.

The ANFIS uses hybrid learning technique to calculate the linear, non linear parameters. The output of the ANFIS is the estimated thoracic signal present in the abdominal signal. The error between the estimated thoracic signal and the abdominal signal gives the FECG. Real data was used to illustrate the effectiveness of the method in extracting FECG signals. The training and learning procedure is needed only one time and can be done offline. Thus the computational complexity can be reduced. The ANFIS converts the fuzzy inference engine in to an adaptive network that learns the relationship between the inputs and outputs. Extraction of FECG using ANFIS yielded satisfactory results. In this method, generalized bell shape (gbellmf) MF is used for ANFIS training. Selecting an appropriate number of membership functions is essential for improving the convergence speed of the ANFIS algorithm.
There are two inputs in the input layer. Fuzzification is done by layer 1 (inputmf) which has 6 membership functions to each input. Totally 36 fuzzy rules are used in layer 2 (rule). Layer 3 is the normalizing layer which is not included in this architecture. Layer 4 is the defuzzification layer (outmf). Layer 5 is the summation layer. Two inputs, 6 membership functions generating 36 fuzzy rules yielded 101 nodes, 108 consequent parameters, 36 premise parameters for training data pair of 601 samples.

3.6.5 Simulated Result of Fetal ECG Extraction using ANFIS

Figure 3.8 shows the abdominal ECG, estimated thoracic signal and the extracted fetal ECG using ANFIS method. The estimated thoracic ECG is seen to be closely following the maternal ECG which is present in the abdominal ECG signal. The FECG is extracted by cancelling the thoracic ECG signal from the abdominal ECG signal. In this method, there is an
oscillatory phenomena present in the position of maternal ECG in the extracted signal.

![Image of Fetal ECG Extraction Using ANFIS](image)

**Figure 3.8 Fetal ECG Extraction Using ANFIS**

### 3.7 WAVELET TRANSFORM

The wavelet transform is a time-scale representation technique, which describes a signal by using correlation with translation and dilation of a function called as mother wavelet. The Continuous Wavelet Transform (CWT) is defined as the sum over all times of the continuous signal $f(t)$ multiplied by scaled, shifted versions of the mother wavelet $\Psi((t-\tau)/s)$ is given as
The parameter $s$ is the scale factor that compresses the mother wavelet and $\tau$ is the translation of the mother wavelet along the time axis. The discrete wavelet transform (DWT) is defined by splitting $f(t)$ in to smaller non overlapping parts $f_i(t)$, taking a finite number of scales $N$ and down sampling the discrete wavelet coefficients samples to $M$, the number of samples of $f_i(t)$, as given by

$$DWT_i(s, \tau, N) = \sum_{j=1}^{N} \sum_{k=1}^{N} CWT_i(S_j, \tau_k) \uparrow N$$ (3.17)

The DWT is a batch method, which analyses a finite-length, time-domain signal at different frequency bands with different resolutions by successive decomposition in to coarse approximation and detail information. The approximation is the high scale, low frequency components of the signal and the details are the low scale, high frequency components. The wavelets are used as a decomposition and denoising tool. The wavelet denoising method consists of applying DWT to the original signal, thresholding the detail and approximation coefficients and inversing the threshold coefficients to obtain the time domain denoised data. The performance of the wavelet denoising depends upon the type of wavelet transform, type of the wavelet, thresholding rule and the number of decomposition levels. The steps for denoising are

- Decompose the signal-Choose a wavelet, choose a level ‘n’.
  Compute the wavelet decomposition of the signal ‘s’ at level ‘n’
- After the wavelet decomposition, the wavelet coefficients are modified and then the reconstruction takes place
Reconstruction of the signal-compute wavelet reconstruction using the original approximation coefficients of level ‘n’ and the modified coefficients of level from ‘1 to n’

3.7.1 Bi-orthogonal Wavelet

A bi-orthogonal wavelet is a wavelet where the associated wavelet transform is invertible but not necessarily orthogonal. Designing bi-orthogonal wavelets allows more degrees of freedom than orthogonal wavelets. One additional degree of freedom is the possibility to construct symmetric wavelet functions.

In the bi-orthogonal case, there are two scaling functions \( \phi, \tilde{\phi} \), which may generate different multi-resolution analyses, and accordingly two different wavelet functions \( \psi, \tilde{\psi} \). So the numbers \( M \) and \( N \) of coefficients in the scaling sequences \( a, \tilde{a} \) may differ. The scaling sequences must satisfy the following bi-orthogonality condition.

\[
\sum_{n \in \mathbb{Z}} a_n \tilde{a}_n + 2m = 2 \delta_{m,0} \tag{3.18}
\]

Then the wavelet sequences can be determined as

\[
b_n = (-1)^n a_{M-l-n}, n = 0, ..., M - 1 \quad \text{and} \quad \tilde{b}_n = (-1)^n a_{M-l-n}, n = 0, ..., N - 1.
\]

3.7.2 Coiflets Wavelet

Coiflets are discrete wavelets designed by Ingrid Daubechies, at the request of Ronald Coifman, to have scaling functions with vanishing moments is shown in Figure 3.9. The wavelet is near symmetric, their wavelet functions have \( N / 3 \) vanishing moments and scaling functions \( N / 3 - 1 \), and has been used in many applications using Calderón-Zygmund Operators.
Both the scaling function (low-pass filter) and the wavelet function (High-Pass filter) must be normalised by a factor $\frac{1}{\sqrt{2}}$. Below are the coefficients for the scaling functions for C6-30. The wavelet coefficients are derived by reversing the order of the scaling function coefficients and then reversing the sign of every second one (ie), $C_6$ wavelet = {−0.022140543057, 0.102859456942, 0.544281086116, −1.205718913884, 0.477859456942, 0.102859456942}).

Mathematically, this looks like $B_k = (-1)^k C_{N-1-k}$ where $k$ is the coefficient index, $B$ is a wavelet coefficient and $C$ a scaling function coefficient. $N$ is the wavelet index, (ie) 6 for C6.

![Figure 3.9 Coiflet with two vanishing moments](image)

### 3.7.3 Daubechies Wavelets

The Daubechies wavelets are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this
class, there is a scaling function (also called father wavelet) which generates an orthogonal multiresolution analysis.

In general, the Daubechies wavelets are chosen to have the highest number $A$ of vanishing moments, (this does not imply the best smoothness) for given support width $N=2^A$, and among the $2^{A-1}$ possible solutions the one is chosen whose scaling filter has extremal phase. The wavelet transform is also easy to put into practice using the fast wavelet transform. Daubechies wavelets are widely used in solving a broad range of problems, e.g. self-similarity properties of a signal or fractal problems, signal discontinuities, etc.

The Daubechies wavelets are not defined in terms of the resulting scaling and wavelet functions; in fact, they are not possible to write down in closed form. The graphs below are generated using the cascade algorithm, a numeric technique consisting of simply inverse-transforming $[1 \ 0 \ 0 \ 0 \ 0 \ldots]$ an appropriate number of times.

### 3.7.4 Discrete Meyer's Wavelet

Meyer's wavelet construction is fundamentally a solvent method for solving the two-scale equation. Given a basis $j$ for the approximation space $V_0$, Meyer employed Fourier techniques to derive the DTFT of the two-scale equation coefficients, $g_0[n]$, from $F(w)$; Knowledge of $g_0[n]$ leads naturally to a filter bank interpretation. An implementation becomes attractive when $g_0[n]$ is associated with a rational system function of finite order. But because Meyer's construction has compact support in the frequency domain, there is no rational solution.

Approximating the rational transform of $g_0[n]$ for a Meyer type construction specified to a total mean-square fidelity of greater than 100dB is used. Here both FIR and IIR realizations are considered. The FIR realizations
will be shown to be more efficient than the IIR. The transforms are designed using a construction polynomial \( b(x) \). The superiority of the FIR implementation is because of the consequent energy compaction in the impulse response which is controlled by the choice of \( b(x) \). The IIR transforms are then determined from the FIR impulse response \( g[n] \) via the technique known as Prony's method.

### 3.7.5 Simulated Results of FECG Extraction using Wavelets

Figure 3.10 shows the estimated fetal ECG extraction using Daubechies wavelet and the PSNR value calculated for this wavelet is about 60.6634.

![Figure 3.10 FECG Extraction using Daubechies Wavelet](image)

**Figure 3.10** FECG Extraction using Daubechies Wavelet
Figure 3.11 shows the estimated fetal ECG extraction using Biorthogonal wavelet and the PSNR value calculated for this wavelet is about 59.6937.

![Fetal ECG Extraction using Bi-orthogonal Wavelet](image)

**Figure 3.11** FECG Extraction using Bi-orthogonal Wavelet

Figure 3.12 shows the estimated FECG extraction using Discrete Meyer wavelet and the PSNR value calculated for this wavelet is about 60.7353.

![Fetal ECG Extraction using Discrete Meyer Wavelet](image)
Figure 3.12 FECG Extraction using Discrete Meyer Wavelet

Figure 3.13 shows the estimated FECG extraction using Coiflet wavelet and the PSNR value calculated for this wavelet is about 60.8073.
In this method, the abdominal ECG is first preprocessed by wavelet as shown in Figure 3.14. The wavelet preprocessing includes wavelet decomposition and reconstruction. The wavelet decomposition and reconstruction were performed by various wavelets and only the approximation coefficients are retained as a signal carrying the useful information.
3.7.6.1 Simulated Results of FECG Extraction using Wavelet and ANFIS

Figure 3.15 shows the estimated FECG when the Coiflet wavelet is used for pre-processing and the PSNR value obtained for this wavelet is 65.6916.
Figure 3.16 shows the estimated FECG when the Bi-orthogonal wavelet is used for pre-processing and the PSNR value obtained for this wavelet is 61.7442.

Figure 3.16  FECG Signal Extraction using Bi-orthogonal Wavelet Pre-Processing

Figure 3.17 shows the estimated FECG when the Daubechies wavelet is used for pre-processing and the PSNR value obtained for this wavelet is 63.4534.
Figure 3.17 FECD Signal Extraction using Daubechies Wavelet Pre-Processing

Figure 3.18 shows the estimated FECD when the Discrete Meyer wavelet is used for pre-processing and the PSNR value obtained for this wavelet is about 64.6353.
3.7.7 FECG Extraction using ANFIS and Wavelet Post Processing

In this method, the inputs to the ANFIS are the abdominal signal and the thoracic signal. The error signal is the FECG signal which is decomposed to 5 levels using various wavelets as shown in Figure 3.19.
The approximation coefficient is taken as a noise free FECG signal which is the output from the wavelet post processing block.

3.7.7.1 Simulated Results of FECG extraction using ANFIS and Wavelet

Figure 3.20 shows the estimated fetal ECG extraction when coiflet wavelet is used for post-processing and the PSNR value obtained for this wavelet is 68.795.
Figure 3.21 shows the estimated fetal ECG extraction when Daubechies wavelet is used for post-processing and the PSNR value obtained for this wavelet is 66.9735.

![Figure 3.21 FECG Extraction using Daubechies Post-processing](image)

Figure 3.21 FECG Extraction using Daubechies Post-processing

Figure 3.22 shows the estimated FECG extraction using Bi-orthogonal wavelets and the PSNR value obtained for this wavelet is 63.7945.

Figure 3.22 shows the estimated FECG extraction using Bi-orthogonal wavelets and the PSNR value obtained for this wavelet is 63.7945.
Figure 3.22 FECG Extraction using Biorthogonal Post-processing

Figure 3.23 shows the estimated FECG extraction using Discrete Meyer wavelets and the PSNR value obtained for this wavelet is 67.8788.
3.8 COMPARISON STUDIES OF WAVELET, WAVELET PRE-PROCESSING AND WAVELET POST-PROCESSING METHODS

Figure 3.24 shows the PSNR comparison of wavelet, wavelet pre-processing and wavelet post-processing methods. From all these methodologies used, minimum error occurs only in wavelet post-processing techniques and so the PSNR value is more.
This chapter has dealt with adaptive filter, ANFIS technique and various wavelets for estimating the fetal ECG signal. The performance evaluation of wavelets, wavelets pre and post-processing using Coiflet, Bi-orthogonal, Daubechies and Discrete Meyer wavelets are carried out. ANFIS technique produces high PSNR, less epoch number and less convergence time. From the results, it is evident that the above mentioned ANFIS technique has accurately predicted the fetal ECG in spite of any parameter variation in the composite signal. The performance of the wavelet denoising method depends upon the type of wavelet transform, type of the wavelet, thresholding rule and the number of decomposition levels. Hybridization of ANFIS and wavelet improves the PSNR still further.

Upon comparison, it is evident that the ANFIS followed by wavelet post-processing using Coiflet and Discrete Meyer wavelets produced the best results among all the four wavelets. But it should be noted that there is lack of standard reference data base available in the literature. This means that different methods in the literature cannot be directly compared since they were evaluated using different data sets.