EXTRACTION OF FOETUS HEARTBEAT

by

ROHIT IYER	21BEC1019
RICHA JHA	21BEC1023
TUSHAR CHOPRA	21BEC1069
BHAVYA OSTWAL	21BEC1669

A project report submitted to

Dr. DHEEREN KU MAHAPATRA

SCHOOL OF ELECTRONICS ENGINEERING

for the course of

BECE301L – DIGITAL SIGNAL PROCESSING

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

JULY 2023

ABSTRACT

This report presents a comprehensive investigation and application of Fetal ElectroCardiogram (ECG) Extraction from the mother's womb. The Fetal ECG is derived from multiple thoracic and abdominal ECG signals of pregnant women. The methodology involves the utilization of the Adaptive Noise Cancellation Technique to estimate corrupted signals caused by additive noise or interference.

The adaptive filter's weights within the Adaptive Noise Cancellation system are updated using three algorithms: the Least Mean Square (LMS) algorithm, Normalized Least Mean Square (NLMS) algorithm, and Leaky Least Mean Square (LLMS) algorithm. Both Single Input and Single Output (SISO) systems and Multiple Input and Single Output (MISO) systems are implemented. Notably, the NLMS algorithm exhibits superior performance compared to the LMS and LLMS algorithms in both system configurations.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Dheeren Ku Mahapatra,** Associate Professor, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. K. Mohanaprasad** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

TABLE OF CONTENTS

SERIAL NO.		TITLE	PAGE NO.
		ABSTRACT	3
		ACKNOWLEDGEMENT	4
1		INTRODUCTION	6
	1.1	EXPLANATION	6
	1.2	PROBLEM FORMULATION	7
2		PROPOSED SOLN.	8
	2.1	ADAPTIVE NOISE CANCELLATION SYSTEM	8
	2.2	ADAPTIVE FILTERS	10
	2.3	ADAPTIVE ALGORITHMS	10
3		PROJECT IMPLEMENTATION	15
	3.1	SISO IMPLEMENT.	15
	3.2	MISO IMPLEMENT.	17
	3.3	COMPARISON SISO/MISO	19
4		SOURCE CODE IMPLEMENTATION	20
	4.1	MATLAB CODE	20
5		RESULTS AND INFERENCE	44
	5.1	INPUT SIGNAL	44
	5.2	SISO & MISO I/O	45
6		CONCLUSION	53
7		REFERENCES	54

CHAPTER 1

INTRODUCTION

1.1 EXPLANATION

In the context of fetal monitoring during delivery, obtaining accurate recordings is essential. One method involves placing an electrode on the fetal scalp, but this becomes relevant only after the membranes protecting the fetus have been broken (antepartum). Before that, non-invasive techniques are preferred. Among these, examining the fetal electrocardiogram (FECG) from ECG-recordings on the mother's skin (cutaneous recordings) is of significant importance. The FECG provides crucial insights into the fetus's health and condition, making the analysis of the fetal heart rate (FHR) a routine procedure for evaluating the well-being of the fetus. Additionally, the fetal cardiac waveform holds valuable diagnostic information, especially for identifying arrhythmias.

However, the main challenge lies in separating the fetal heartbeat from the mother's heartbeat signal and other types of noise, particularly the MECG signal, which has a higher amplitude than the fetal ECG. Effective removal of the MECG is crucial for processing the fetal signal accurately and diagnosing the fetus.

Several methods have been proposed to extract the desired fetal signal, such as singular value decomposition (SVD), wavelet transform, and adaptive noise cancellation (ANC) using adaptive algorithms. Adaptive filters, in particular, show promise in this task, as they can dynamically adjust their impulse response to filter out correlated signals and adapt to non-stationary conditions.

Among the adaptive algorithms used for updating filter coefficients, the LMS (least mean square) algorithm, NLMS (normalized least mean square) algorithm, and LLMS (leaky least mean square) algorithm are commonly employed.

In summary, extracting the fetal heartbeat from the mother's womb is a challenging task due to various noise sources, but utilizing adaptive noise cancellation approaches with updated filter coefficients through adaptive algorithms proves to be a more suitable and effective method for fetal monitoring and diagnosis.

1.2 PROBLEM FORMULATION

The primary challenges encountered during the project implementation revolve around effectively isolating the Fetal Electrocardiogram (FECG) from the Maternal Electrocardiogram (MECG) while minimizing noise interference. Additionally, the project entails successful implementation in both Single Input and Single Output (SISO) systems, as well as Multiple Input and Single Output (MISO) systems.

CHAPTER 2

PROPOSED SOLUTION

2.1 ADAPTIVE NOISE CANCELLATION SYSTEM

Adaptive noise cancellation is an alternative technique of estimating signals corrupted by additive noise or interference. Its advantage lies in that, with no priori estimates of signal or noise, levels of noise rejection are attained that would be difficult or impossible to achieve by other signal processing methods of removing noise. Its cost, inevitably, is that it needs two inputs - a primary input containing the corrupted signal and a reference input containing noise correlated in some unknown way with the primary noise. The reference input is adaptively filtered and subtracted from the primary input to obtain the signal estimate. Adaptive filtering before subtraction allows the treatment of inputs that are deterministic or stochastic, stationary or time-variable. In this project, computer simulations for all cases are carried out using Matlab software and experimental results are presented that illustrate the usefulness of adaptive noise.

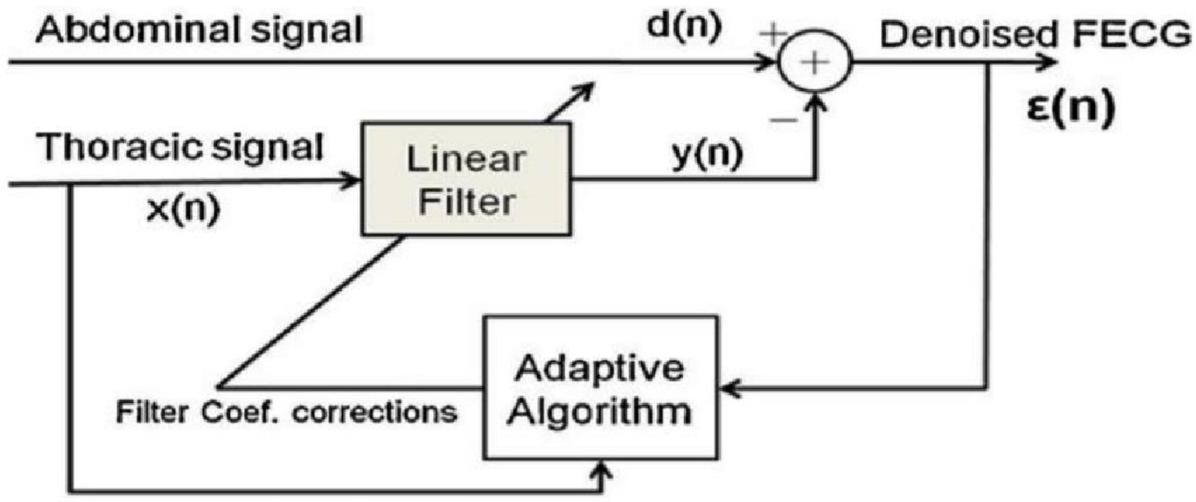


Fig 2.1- ANC

Noise cancellation is a variation of optimal filtering that involves producing an estimate of the noise by filtering the reference input and then subtracting this noise estimate from the primary input containing both signal and noise. It makes use of an auxiliary or reference input which contains a correlated estimate of the noise to be cancelled.

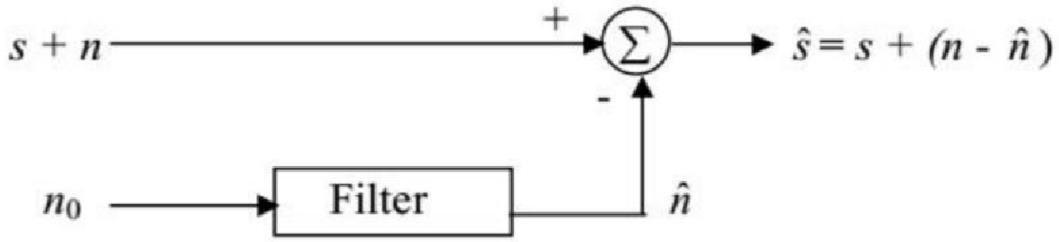


Fig 2.2: Noise Filtering

Subtracting noise from a received signal involves the risk of distorting the signal and if done improperly, it may lead to an increase in the noise level. This requires that the noise estimate should be an exact replica of n. If it were possible to know the relationship between n and , or the characteristics of the channels transmitting noise from the noise source to the primary and reference inputs are known, it would be possible to make a close estimate of n by designing a fixed filter. However, since the characteristics of the transmission paths are not known and are unpredictable, filtering and subtraction are controlled by an adaptive process. Hence an adaptive filter is used that is capable of adjusting its impulse response to minimize an error signal, which is dependent on the filter output. The adjustment of the filter weights, and hence the impulse response, is governed by an adaptive algorithm. With adaptive control, noise reduction can be accomplished with little risk of distorting the signal. In fact, Adaptive Noise Canceling makes possible attainment of noise rejection levels that are difficult or impossible to achieve by direct filtering.

2.2 ADAPTIVE FILTERS

Adaptive filters are digital filters with an impulse response, or transfer-function, that can be adjusted or changed over time to match desired system characteristics. Unlike fixed filters, which have a fixed impulse response, adaptive filters do not require complete a priori knowledge of the statistics of the signals to be filtered. Adaptive filters require little or no a prior knowledge and moreover, have the capability of adaptively tracking the signal under non-stationary circumstances. For an adaptive filter operating in a stationary environment, the error-performance surface has a constant shape as well as orientation. When, however, the adaptive filter operates in a non-stationary environment, the bottom of the error-performance surface continually moves, while the orientation and curvature of the surface may be changing too. Therefore, when the inputs are non-stationary, the adaptive filter has the task of not only seeking the bottom of the error performance surface, but also continually tracking it.

2.3 ADAPTIVE ALGORITHMS

An adaptive algorithm is a set of instructions to perform a function that can adapt in the event of changes in environment. Adaptive algorithms are able to intelligently adjust their activities in light of changing circumstances to achieve the best possible output.

2.3.1 LEAST MEAN SQUARE ALGORITHM (LMS):

If it were possible to make exact measurements of the gradient vector at each iteration, and if the step-size parameter μ is suitably chosen, then the tap-weight vector computed by using the method of steepest-descent would indeed converge to the optimum Wiener 6 solution. In reality, however, exact measurements of the gradient vector are not possible, and it must be estimated from the available data. In other words, the tap weight vector is updated in accordance with an algorithm that adapts to the incoming data. One such algorithm is the least mean square (LMS) algorithm. A significant feature of LMS is its simplicity; it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. We have earlier found that gradient vector,

$$\nabla(n) = -2p + 2Rw(n)$$

To estimate this, we estimate the correlation matrix R and cross correlation matrix p by instantaneous estimates i.e.

$R'(n)=u(n)u^H(n)$…..1
$p'(n)=u(n)d^*(n)$...2

Correspondingly, the instantaneous estimate of the gradient-vector is

$\nabla'(n) = -2\, u(n)\, d^*(n) + 2\, u(n)u^H(n)w(n)$...3

The estimate is unbiased in that its expected value equals the true value of the gradient vector. Substituting this estimate in the steepest descent algorithm, we get a new recursive relation for updating the tap-weight vector:

$w'(n+1) = w'(n)+\mu u(n)[d^*(n) - u^H(n)w'(n)]$..4

Equivalently the LMS update equation can be written in the form of a pair of relations:

$e(n)=d(n)-u^H(n)w'(n)$...5
$w'(n+1)=w'(n)+\mu u(n)e^*(n)$…...6

The first equation defines the estimation error e(n), the computation of which is based on the current estimate of the tap weight vector w′(n). The term $\mu u(n)e^*(n)$ in the second equation represents the correction that is applied to the current estimate of the tap-weight vector. The iterative procedure is started with the initial guess w′(0), a convenient choice being the null vector; w′(0) = 0. The algorithm described by the equation (4) or equivalently by the equations (5) and (6), is the complex form of the adaptive least mean square (LMS) algorithm. It is also known as the stochastic-gradient algorithm. The instantaneous estimates of R and p have relatively large variances. It may therefore seem that the LMS algorithm is incapable of good performance. Ideally, the minimum mean-squared error J_{min} is realized when the coefficient vector w(n) of the transversal filter approaches the optimum value w_0. The steepest descent algorithm does realize this idealized condition as the number of iterations, n approaches infinity, because it uses exact measurements of the gradient vector at each iteration. On the other hand, LMS relies on a noisy estimate of the gradient vector, with the result that the tap weight vector only approaches the optimum value after a large number of iterations and then executes small fluctuations about w0. Consequently, use of LMS results in a mean-squared error J(∞) after a large no. of iterations.

2.3.2 NORMALIZED LEAST MEAN SQUARE ALGORITHM (NLMS):

The main drawback of the LMS algorithm is that it is sensitive to the scaling of its input. This makes it very hard to choose a learning rate μ that guarantees stability of the algorithm. The Normalized least mean 8 squares (NLMS) algorithm is a variant of the LMS algorithm that solves this problem by normalizing with the power of the input.

NLMS ALGORITHM SUMMARY:

The iterative procedure is started with the initial guess w′(0), a convenient choice being the null vector; w′(0) = 0.

From equation (6), we get

$w'(n+1) = w'(n)+\mu u(n)e^*(n)$

In real time scenario, input signal power will not remain constant. It affects the convergence rate of the filter and also it gives gradient noise amplification problems. So the step size is normalized in NLMS to overcome this problem

• Modified formula for convergence factor

•$\mu(n)=\beta/((c+\|Xn\|)^2)$ 7

– $\mu(n)$ = step size

– β = normalized step-size ($0 < \beta < 2$)

– c = safety factor

• Weight vector:

From equation(6), we get

$w(n+1) = w(n) + \mu(n)u(n)e^*(n)$ or

w(n+1)=w(n)+(β/($\|x(n)\|$^2))u(n)e(n) 8

2.3.3 LEAKY LEAST MEAN SQUARE ALGORITHM(LLMS):

A problem can occur when the autocorrelation matrix associated with the input process has one or more zero eigenvalues. In this case, the adaptive filter will not converge to a unique solution. In addition, some uncoupled coefficients (weights) may grow without bound until hardware overflow or underflow occurs. This problem can be remedied by using coefficient leakage.

This LMS algorithm can be written as the following equation

w(n+1) = (1-μr)w(n) + μe(n)u(n)..9

where the adaptation constant μ and the leakage coefficient r are small positive values.

2.3.4 COMPARISON OF LEAST MEAN SQUARE ALGORITHMS:

LMS Algorithm	Leaky LMS Algorithm	Normalized LMS algorithm
It converges with a fixed step size μ.	A leakage factor 'γ' is introduced here along with step size	It has a time-varying step size β.
It converges when $0<\mu<2/\lambda_{max}$	It converges When $0<\gamma<<1$.	It converges when $0<\beta<2$.
Converges slowly.	It introduces leaky value so that it give more stability	Converges comparatively high.
Updates the filter coefficients by using the following equation: $w_{n+1} = w_n + \mu e(n) x^*(n)$	LLMS updates the filter coefficients by using the following equation: $w_{n+1} = (1 - \mu\gamma) w_n + \mu e(n) x^*(n)$	NLMS updates the filter coefficients by using the following equation: $w_{n+1} = w_n + \beta \frac{x^*(n)}{\lVert x(n) \rVert^2} e(n)$

CHAPTER 3
PROJECT IMPLEMENTATION

In foetus heart beat extraction, we implement ANC in both single input and single output systems and multiple input and single output systems.

3.1 SINGLE INPUT AND SINGLE OUTPUT SYSTEM:

In this SISO implementation, all the thoracic signals are averaged first and then given to the adaptive filter whose tap coefficients are updated using least mean square (LMS), normalized least mean square (NLMS) and leaky least mean square algorithms (LLMS). The average of all the thoracic signals is given as reference input to the adaptive noise canceller (ANC) and the average of all the abdomen signals is given as primary input. As a result, fetus heartbeat is obtained as output. Single input and single output system is the classic system with single transmitting antenna at the source and single receiving antenna at the destination. Single input single output (SISO) is easier for wireless communication system to transmit and receive signal. Single input and Single output systems are also known as single variable control systems. The throughput of the system depends upon the channel bandwidth and signal to noise ratio.

Advantages:

- SISO systems have less complexity
- Designing is simple along with easy implementation.
- Less expensive.
- Only one filter is used for the entire signal.

Disadvantages:

- Channel capacity in other techniques is much better than SISO systems.
- Interference and fading occurs.
- Less error correction.

Applications:

- SISO systems are mainly used in satellite, radio CDMA and GSM systems.

- Multiple systems like Bluetooth, Wi-Fi, radio broadcasting, TV etc. use SISO systems.
- In this project, the implementation of adaptive noise cancellation (ANC) system in single input single output system (SISO) is done.

It is as shown in Fig 3.1

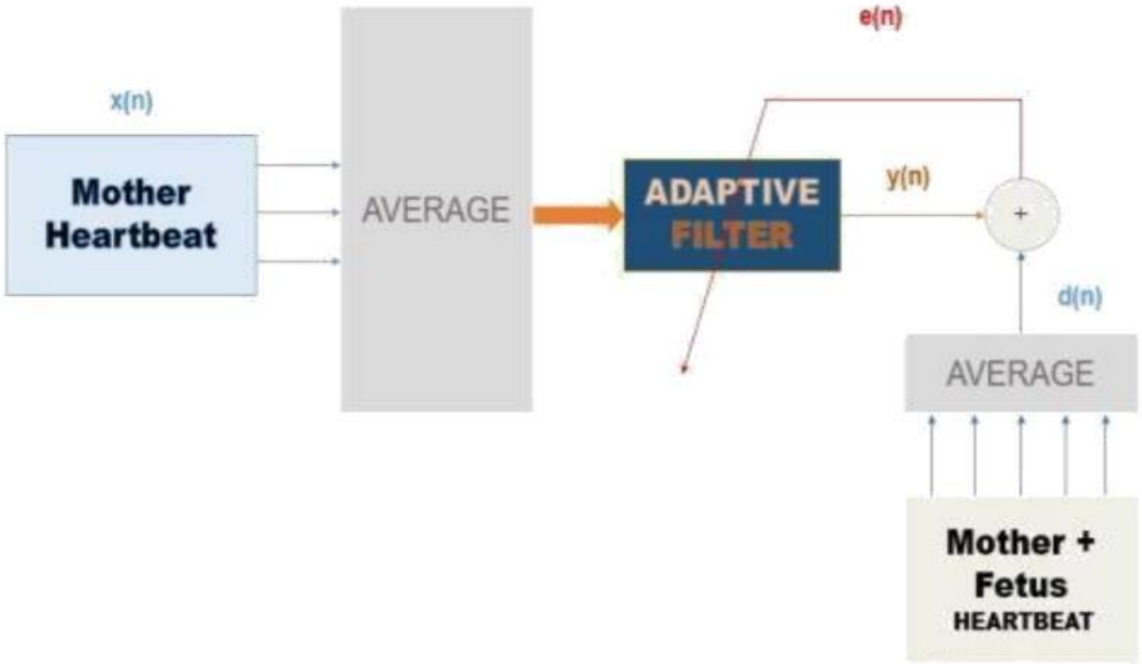


Fig 3.1 SISO Implementation

Here output

e(n) = d(n) - y(n)

y(n) is the filter output

d(n) = (a1+a2+a3+a4+a5)/ 5

Where a1, a2, a3, a4, a5 are the abdomen signals.

3.2 MULTIPLE INPUT AND SINGLE OUTPUT SYSTEMS:

MISO or the multiple input and single output system is a scheme of RF wireless communication systems in which there are multiple transmitting antennas at the source and single receiving antennas.

Primary signal: The average of abdominal input signals is considered as primary signal.

Reference Signal:

Thoracic signals are applied to different filters and its average is calculated. In this ANC-MISO the reference signal (or) individual mother signals are given as multiple input to the adaptive filters to reduce the error and compared with the input signal to obtain the desired foetus heartbeat.

Advantages:

- To reduce the effects of multipath wave propagation, delay, packet loss etc.
- More antennas are used at the receiving end in MISO systems.
- Error correction increases with increase in number of adaptive filters and hence output will be accurate.

Disadvantages:

- Complexity of the system increases as the number of filter are increased.
- Increase in the number of adaptive filters leads to costly affairs.
- Difficulty in system implementation due to more number of filters.

Applications:

- MISO scheme has various applications in Digital television, W-LAN’s, metropolitan area networks (MANs), and mobile communications.
- In this project, the implementation of adaptive noise cancellation (ANC) system in multiple input single output system (MISO) is done.

It is as shown in Fig 3.2

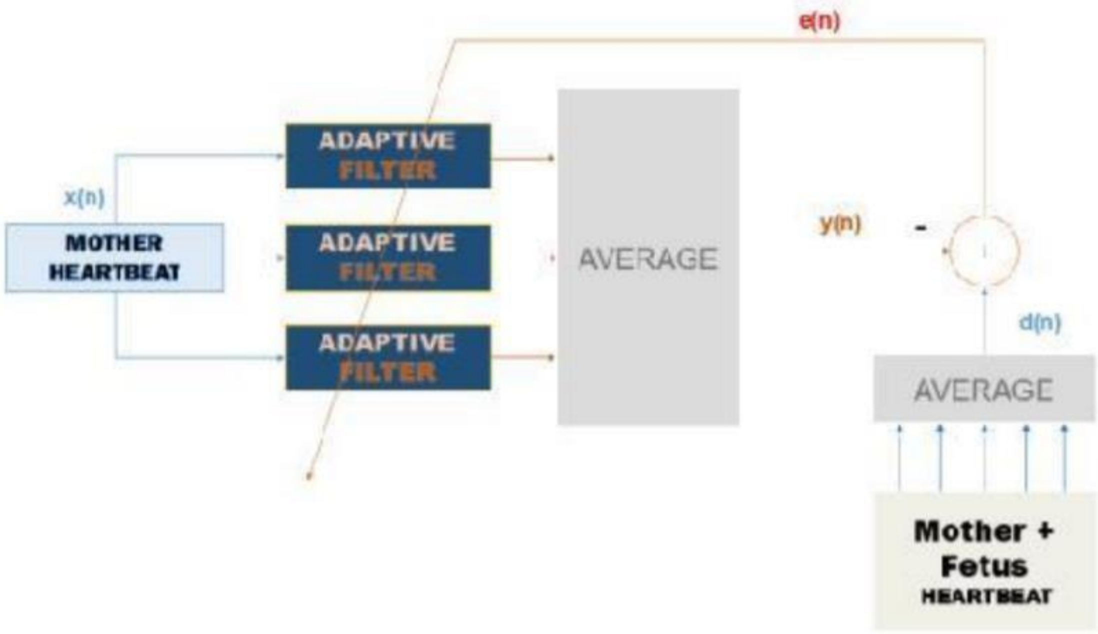


Fig 3.2 MISO implementation

As shown in the Fig3.2,

every thoracic signal is given as a reference signal to the individual adaptive filter first and then the average is calculated. The average of all the abdomen signals is given as primary input. As a result, fetus heartbeat is obtained as output.

Here output

e(n)=d(n)-y(n)

y(n) = $(t1 + t2 + t3)*1/3$

where t1, t2, t3 are the filtered outputs.

d(n) = (a1+a2+a3+a4+a5)/ 5

Where a1, a2, a3, a4, a5 are the abdomen signals.

3.3 COMPARISON BETWEEN SISO AND MISO SYSTEMS:

SISO	MISO
It is a single input single output system.	It is a multiple input single output system.
Only a single filter is required.	Multiple filters are required.
In this system, average signal of all thoracic signals is given to the adaptive filter.	In this system, every thoracic signal is given as input to individual adaptive filter and then the average is calculated.
Hardware complexity is less compared to MISO	Hardware complexity is more compared to MISO.

CHAPTER 4
SOURCE CODE IMPLEMENTATION

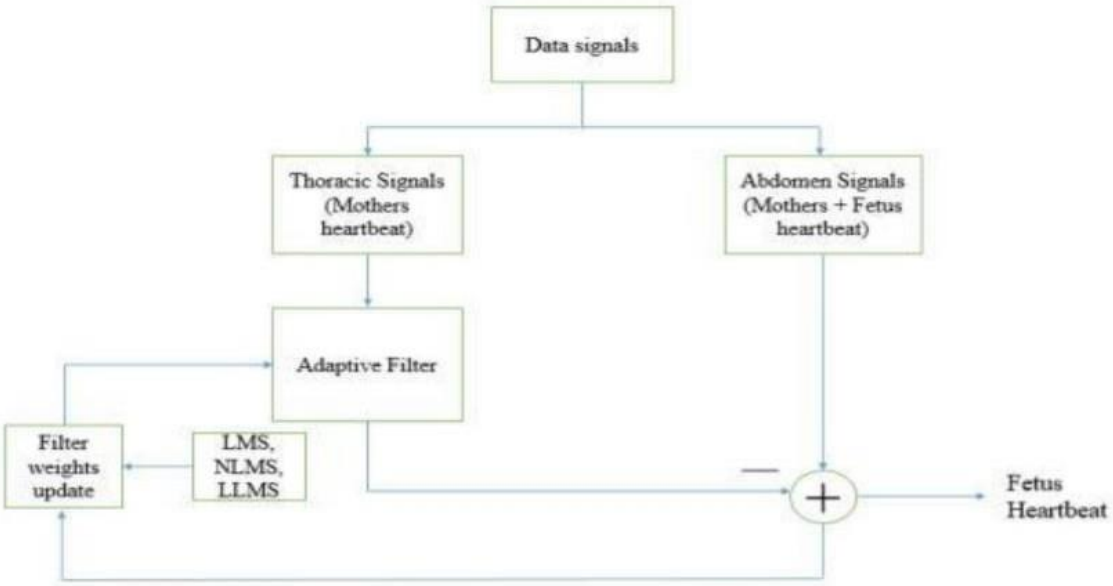


Fig 4: Schematic representation of source code

As shown in the schematic representation, first both the thoracic signals and abdomen signals are separated from the data signal.. As a result, foetus heartbeat is obtained. Now, the extraction of the foetus heartbeat is done in the MISO system and the filtering operation will be completed after we get the minimum error. As a result, fetus heartbeat is obtained as output.

4.1 MATLAB CODE:

```
%clc

% clear all;

% close all;

load foetal_ecg.dat %%%loading the input signal%%%

S= foetal_ecg; % source signal

Fs=500; % sampling Frequency
```

```
t=S(:,1); % Time samples

%%%PLOTTING INPUT SIGNALS%%%%%%

figure

d1=S(:,2); %%%Abdominal signal 1

subplot(3,1,1);

plot(t,d1,'r');

xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Abdominal signal 1');

d2=S(:,3); %%%Abdominal signal 2

subplot(3,1,2);

plot(t,d2,'r');

xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Abdominal signal 2');

d3=S(:,4); %%%Abdominal signal 3

subplot(3,1,3);

plot(t,d3,'r');

xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Abdominal signal 3');

d4=S(:,5); %%%Abdominal signal 4

figure

subplot(2,1,1);

plot(t,d4,'r');
```

```
xlabel('time period(sec)');
ylabel('amplitude(mV)');
title('Abdominal signal 4');
d5=S(:,6); %%%Abdominal signal 5
subplot(2,1,2);
plot(t,d5,'r');
xlabel('time period(sec)');
ylabel('amplitude(mV)');
title('Abdominal signal 5');
figure
x1=S(:,7); %Thoracic signal 1
subplot(3,1,1);
plot(t,x1,'b');
xlabel('time period(sec)');
ylabel('amplitude(mV)');
title('Thoracic signal 1');
x2=S(:,8); %Thoracic signal 2
subplot(3,1,2);
plot(t,x2,'b');
xlabel('time period(sec)');
ylabel('amplitude(mV)');
title('Thoracic signal 2');
x3=S(:,9); %Thoracic signal 3
subplot(3,1,3);
plot(t,x3,'b');
```

```
xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Thoracic signal 3');

d=(d1+d2+d3+d4+d5)/5; %%% AVERAGE OF ABDOMINAL SIGNALS

x=(x1+x2+x3)/3; %%% AVERAGE OF THORACIC SIGNALS

figure

subplot(2,1,1);

plot(t,d,'r');

xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Average of abdominal signals[Mother+Fetus]');

subplot(2,1,2);

plot(t,x,'b');

xlabel('time period(sec)');

ylabel('amplitude(mV)');

title('Average of Thoracic signals(Mother)');


%SISO Implementation

%%% Generating ANC using LMS Algorithm

p=12;%order of filter

mu=0.0000005; % Step size

[A1,L,yl]=lms(x,d,mu,p);%calling LMS function



%%% Generating ANC using NLMS Algorithm
```

```
beta=0.001;%normalized step size

p=12;%order of filter

[A,LN,yn]=nlms(x,d,beta,p);%calling NLMS function



% %%% Generating ANC using LLMS Algorithm

gama= 0.001;%leakage coefficient

p=12;%order of filter

mu=0.0000009; % Step size

[AL,LL,yll]= llms(x,d,mu,gama,p);%calling LLMS function


%%%%plotting of lms signal%%%%

figure

subplot(2,1,1)

plot(t,L,'r');

legend('SISO-LMS');

title('Plot of the error signal-SISO');

ylabel('amplitude(mV)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);

%%%%%plotting of filtered signal of lms in siso%%%

subplot(2,1,2)

plot(t,yl,'r');

legend('SISO-LMS');

title('Plot of the filtered output-SISO');
```

```
ylabel('amplitucde(mV)');

xlabel('time(SEC)-->');

axis([0 2 -10 10]);


%plotting nlms signals

figure

subplot(2,1,1)

plot(t,LN,'r');

legend('SISO-NLMS');

title('Plot of the error signal-SISO');

ylabel('amplitude(mV)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);

%%%%%plotting of filtered signal of nlms in siso%%%

subplot(2,1,2)

plot(t,yn,'r');

legend('SISO-NLMS');

title('Plot of the filter output-SISO');

ylabel('amplitucde(mV)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);


%%%%plotting of llms signal%%%%

figure

subplot(2,1,1)
```

```
plot(t,LL,'r');
legend('SISO-LLMS');
title('Plot of the error signal-SISO');
ylabel('amplitude(mV)');
xlabel('time(SEC)-->');
axis([0 5 -30 30]);
%%%%%plotting of filtered signal of lms in siso%%%
subplot(2,1,2)
plot(t,yll,'r');
legend('SISO-LLMS');
title('Plot of the filter output-SISO');
ylabel('amplitucde(mV)');
xlabel('time(SEC)-->');
axis([0 2 -10 10]);

%COMPARISION
%for error
figure
plot(t,L,'r-',t,LN,'b-',t,LL,'g--');
legend('LMS-error','NLMS-error','LLMS-error');
title('Plot of the LMS,NLMS,LLMS output[Foetus signal]-error-SISO ');
ylabel('amplitude(mV)');
xlabel('time(SEC)-->');
axis([0 5 -30 30]);
```

```
%for output

figure

plot(t,yl,'r-',t,yn,'b-',t,yll,'g--');

legend('LMS-output','NLMS-output','LLMS-output');

title('Plot of the LMS,NLMS,LLMS output -SISO');

ylabel('amplitude(mV)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);



%% ERROR BETWEEN SISO and MISO%%

figure

% % LMS ERROR:

yl=reshape(yl,1,2500);

lms_error = abs(ym1-yl);

subplot(311)

plot(lms_error);

% NLMS ERROR:

nlms_error = abs(ym2-yn);

subplot(312)

plot(nlms_error);

% % LLMS ERROR:

yll=reshape(yll,1,2500);

llms_error = abs(ym3-yll);

subplot(313)
```

```
plot(llms_error);

%% SNR MISO%%
% % LMS ERROR:
lms_snr_miso = 20 * log10(rms(ym1)/rms(d) );

% NLMS ERROR:
nlms_snr_miso = 20 * log10(rms(ym2)/rms(d) );

% % LLMS ERROR:
llms_snr_miso = 20 * log10(rms(ym3)/rms(d) );

%% SNR SISO%%
% % LMS ERROR:
lms_snr_siso = 20 * log10(rms(yl)/rms(d) );

% NLMS ERROR:
nlms_snr_siso = 20 * log10(rms(yn)/rms(d) );

% % LLMS ERROR:
llms_snr_siso = 20 * log10(rms(yll)/rms(d) );
%%
%LMS Function
% function [w,y,e,W] = lms(x,d,mu_step,M)
% N = length(x); % number of data samples
```

```
% y = zeros(N,1); % initialize filter output vector
% w = zeros(M,1); % initialize filter coefficient vector
% e = zeros(N,1); % initialize error vector
% W = zeros(M,N); % filter coefficient matrix for coeff. history
% for n = 1:N
%  if n <= M % assume zero-samples for delayed data that isn't available
%       k = n:-1:1;
%       x1 = [x(k); zeros(M-numel(k),1)];
%  else
%       x1 = x(n:-1:n-M+1); % M samples of x in reverse order
%  end
%  y(n) = w'*x1; % filter output
%  e(n) = d(n) - y(n); % error
%   w = w + mu_step*e(n)'*x1; % update filter coefficients
%   W(:,n) = w; % store current filter coefficients in matrix
% end
% end
function [A,E,Y] = lms(x,d,mu,nord)
X=convm(x,nord);
[M,N]=size(X);
if nargin < 5, a0 = zeros(1,N); end
a0=a0(:).';
Y(1)=a0*X(1,:).';
E(1)=d(1) - Y(1);
A(1,:) = a0 + mu*E(1)*conj(X(1,:));
```

```
if M>1
for k=2:M-nord+1;
Y(k,:)=A(k-1,:)*X(k,:).';%ouput equation
E(k,:) = d(k) - Y(k,:);%error signal
A(k,:)=A(k-1,:)+mu*E(k)*conj(X(k,:));%update equation
end
end
end


% NLMS Function:
%%NLMS CALGORITHM FOR THE SOURSE CODE%%
function [A,E,Y] = nlms(x,d,beta,nord)
X=convm(x,nord);
[M,N]=size(X);
if nargin < 5, a0 = zeros(1,N); end%initialization
a0=a0(:).';
Y(1)=a0*X(1,:).';
E(1)=d(1) - a0*X(1,:).';
DEN=X(1,:)*X(1,:)'+0.0001;
A(1,:) = a0 + beta/DEN*E(1)*conj(X(1,:));
    if M>1
    for k=2:M-nord+1;
    Y(k)=A(k-1,:)*X(k,:).';%output equation
    E(k) = d(k) - A(k-1,:)*X(k,:).';%error signal
    DEN=X(k,:)*X(k,:)'+0.0001;%normalizing the input signal
```

```
        A(k,:)=A(k-1,:)+ beta/DEN*E(k)*conj(X(k,:));%update equation
        end
        end
end


%LLMS Function:
%%LLMS FUNCTION OF THE SOURCE CODE%%
function [A,E,Y]= llms(x,d,mu,gama,nord,a0)
X=convm(x,nord);


[M,N]=size(X);
if nargin < 6, a0 = zeros(1,N); end
a0=a0(:).';
Y(1)=a0*X(1,:).';
E(1)=d(1) - Y(1);
A(1,:)=(1-mu*gama)*a0+mu*E(1)*conj(X(1,:));
if M>1
for k=2:M-nord+1
Y(k,:)=A(k-1,:)*X(k,:).';%output signal
E(k,:) = d(k) - Y(k,:);%error signal
A(k,:)=(1-mu*gama)*A(k-1,:)+mu*E(k)*conj(X(k,:));%update eqtn
end
end
end
```

```
% MATLAB code for MISO system
clc;
clear all;
close all;
load('foetal_ecg.dat');
x=foetal_ecg;
% time signal;
timesig=x(:,1);
t=x(:,1);
% abdnomial signals
abdomin1=x(:,2);
abdomin2=x(:,3);
abdomin3=x(:,4);
abdomin4=x(:,5);
abdomin5=x(:,6);
%thoriad signals
thoirad1=x(:,7);
thoirad2=x(:,8);
thoirad3=x(:,9);
% figure
% subplot(5,1,1);
% plot(timesig,abdomin1);
% title('abdomin1');
% xlabel('time[s]');
```

```
% ylabel('amplitude mV');
% subplot(5,1,2);
% plot(timesig,abdomin2);
% title('abdomin2');
% ylabel('amplitude mV');
% xlabel('time');
% subplot(5,1,3);
% plot(timesig,abdomin3);
% title('abdomin3');
% xlabel('time');
% ylabel('amplitude mV');
% subplot(5,1,4);
% plot(timesig,abdomin4);
% title('abdomin4');
% xlabel('time');
% ylabel('amplitude mV');
% subplot(5,1,5);
% plot(timesig,abdomin5);
% title('abdomin5');
% xlabel('time');
% ylabel('amplitude mV');
% figure
% subplot(3,1,1);
% plot(timesig,thoirad1,'r');
% title('thoirad1');
```

```
% xlabel('time');
% ylabel('amplitude mV');
% subplot(3,1,2);
% plot(timesig,thoirad2,'r');
% title('thoirad2');
% xlabel('time');
% ylabel('amplitude mV');
% subplot(3,1,3);
% plot(timesig,thoirad3,'r');
% title('thoirad3');
% xlabel('time');
% ylabel('amplitude mV');
d=(abdomin1+abdomin2+abdomin3+abdomin4+abdomin5)/5;
a=thoirad1;
a1=thoirad2;
a2=thoirad3;
%% Applying for LMS Algorithm
mue= 0.00000002;
nord=12;
X=convm(a,nord);
X1=convm(a1,nord);
X2=convm(a2,nord);
%Applying LMS algorithm using lms basic function.
[A,A1,A2,E1,ym1] = lms1(X,X1,X2,d,mue,nord);
%% Applying for NLMS Algorithm
```

```
beta=0.005;
nord=12;
X=convm(a,nord);
X1=convm(a1,nord);
X2=convm(a2,nord);
%Applying NLMS algorithm using lms basic function.
[A,A1,A2,E2,ym2] = nlms1(X,X1,X2,d,beta,nord);
%% Applying for LLMS Algorithm
mu=0.0000002;
gammax=0.001;
nord=12;
X=convm(a,nord);
X1=convm(a1,nord);
X2=convm(a2,nord);
%Applying LMS algorithm using llms basic function.
[W,W1,W2,E3,ym3] = llms1(X,X1,X2,d,mu,gammax,nord);
%% Plotting signals
%%%%plotting of lms signal%%%%
figure
subplot(2,1,1)
plot(t,E1(1:2500),'r');
legend('MISO-LMS');
title('Plot of the error signal-MISO');
ylabel('amplitude(uVOLTS)');
xlabel('time(SEC)-->');
```

```
axis([0 5 -30 30]);
%%%%%plotting of filtered signal of lms in miso%%%
subplot(2,1,2)
plot(t,ym1(1:2500),'r');
legend('MISO-LMS');
title('Plot of the filtered output-MISO');
ylabel('amplitucde(uVOLTS)');
xlabel('time(SEC)-->');
axis([0 2 -10 10]);

%plotting nlms signals
figure
subplot(2,1,1)
plot(t,E2(1:2500),'r');
legend('MISO-NLMS');
title('Plot of the error signal-MISO');
ylabel('amplitude(uVOLTS)');
xlabel('time(SEC)-->');
axis([0 5 -30 30]);
%%%%%plotting of filtered signal of nlms in miso%%%
subplot(2,1,2)
plot(t,ym2(1:2500),'r');
legend('MISO-NLMS');
title('Plot of the filter output-MISO');
ylabel('amplitucde(uVOLTS)');
```

```
xlabel('time(SEC)-->');

axis([0 5 -30 30]);


%%%%plotting of llms signal%%%%

figure

subplot(2,1,1)

plot(t,E3(1:2500),'r');

legend('MISO-LLMS');

title('Plot of the error signal-MISO');

ylabel('amplitude(uVOLTS)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);

%%%%%plotting of filtered signal of lms in siso%%%

subplot(2,1,2)

plot(t,ym3(1:2500),'r');

legend('MISO-LLMS');

title('Plot of the filter output-MISO');

ylabel('amplitucde(uVOLTS)');

xlabel('time(SEC)-->');

axis([0 2 -10 10]);


%COMPARISION

%for error

figure

plot(t,E1(1:2500),'r-',t,E2(1:2500),'b-',t,E3(1:2500),'g--');
```

```
legend('LMS-error','NLMS-error','LLMS-error');

title('Plot of the MISO - LMS,NLMS,LLMS output[Foetus signal]-error ');

ylabel('amplitude(uVOLTS)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);


%for output

figure

plot(t,ym1(1:2500),'r-',t,ym2(1:2500),'b-',t,ym3(1:2500),'g--');

legend('LMS-output','NLMS-output','LLMS-output');

title('Plot of the LMS,NLMS,LLMS output -MISO');

ylabel('amplitude(uVOLTS)');

xlabel('time(SEC)-->');

axis([0 5 -30 30]);




% MISO Calling Functions:

% CONVM Function:

function[X] = convm(x,p)

N = length(x)+2*p-2; x = x(:);

xpad = [zeros(p-1,1);x;zeros(p-1,1)];

for i=1:p

X(:,i)=xpad(p-i+1 :N-i+1);

end;
```

```
end
% LMS Function:
function [A,A1,A2,E,y] = lms1(X,X1,X2,d,mu,nord,a0)
[M,N] = size(X);
[M1,N1] = size(X1);
[M2,N2] = size(X2);
if nargin < 7, a0 = zeros(1,N); end
a0 = a0(:).';
y1=  zeros(1,M);
y2= zeros(1,M1);
y3= zeros(1,M2);
E=zeros(1,M);
E1=zeros(1,M1);
E2=zeros(1,M2);
A=zeros(size(X));
A1=zeros(size(X1));
A2=zeros(size(X2));
y1(1)= a0*X(1,:).';
y2(1)= a0*X1(1,:).';
y3(1)= a0*X2(1,:).';
E(1) = d(1) - a0*X(1,:).';
A(1,:) = a0 + mu*E(1)*conj(X(1,:));
A1(1,:) = a0 + mu*E(1)*conj(X1(1,:));
A2(1,:) = a0 + mu*E(1)*conj(X2(1,:));
if M>1
```

```
for k=2:M-nord+1;
y1(k) = A(k-1,:)*X(k,:).';
y2(k) = A1(k-1,:)*X1(k,:).';
y3(k) = A2(k-1,:)*X2(k,:).';
y(k)=(y1(k)+y2(k)+y3(k))/3;
E(k) = d(k) - y(k);
A(k,:) = A(k-1,:) + mu*E(k)*conj(X(k,:));
A1(k,:) = A1(k-1,:) + mu*E(k)*conj(X1(k,:));
A2(k,:) = A2(k-1,:) + mu*E(k)*conj(X2(k,:));
end;
end;
end
% NLMS Function:
function [A,A1,A2,E,y] = nlms1(X,X1,X2,d,beta,nord,a0)
[M,N] = size(X);
[M1,N1] = size(X1);
[M2,N2] = size(X2);
if nargin < 7, a0 = zeros(1,N); end
a0 = a0(:).';
y1= zeros(1,M);
y2= zeros(1,M1);
y3= zeros(1,M2);
E=zeros(1,M);
E1=zeros(1,M1);
E2=zeros(1,M2);
```

```
A=zeros(size(X));
A1=zeros(size(X1));
A2=zeros(size(X2));
y1(1)= a0*X(1,:).';
y2(1)= a0*X1(1,:).';
y3(1)= a0*X2(1,:).';
E(1) = d(1) - a0*X(1,:).';
DEN=X(1,:)*X(1,:)'+X1(1,:)*X1(1,:)' +X2(1,:)*X2(1,:)'+ 0.0001;
A(1,:) = a0 + beta/DEN*E(1)*conj(X(1,:));
A1(1,:) = a0 + beta/DEN*E(1)*conj(X1(1,:));
A2(1,:) = a0 + beta/DEN*E(1)*conj(X2(1,:));
if M>1
for k=2:M-nord+1;
y1(k) = A(k-1,:)*X(k,:).';
y2(k) = A1(k-1,:)*X1(k,:).';
y3(k) = A2(k-1,:)*X2(k,:).';
y(k)=(y1(k)+y2(k)+y3(k))/3;
E(k) = d(k) - y(k);
DEN=X(k,:)*X(k,:)'+X1(k,:)*X1(k,:)' +X2(k,:)*X2(k,:)'+ 0.0001;
A(k,:) = A(k-1,:) + beta/DEN*E(k)*conj(X(k,:));
A1(k,:) = A1(k-1,:) +beta/DEN*E(k)*conj(X1(k,:));
A2(k,:) = A2(k-1,:) + beta/DEN*E(k)*conj(X2(k,:));
end;
end;
end
```

```
% LLMS Function:

function [W,W1,W2,E,y] = llms1(X,X1,X2,d,mu,gammax,nord,a0)

[M,N] = size(X);

[M1,N1] = size(X1);

[M2,N2] = size(X2);

if nargin < 8, a0 = zeros(1,N);

end

a0 = a0(:).';

y1= zeros(1,M);

y2= zeros(1,M1);

y3= zeros(1,M2);

E=zeros(1,M);

E1=zeros(1,M1);

E2=zeros(1,M2);

W=zeros(size(X));

W1=zeros(size(X1));

W2=zeros(size(X2));

y1(1)= a0*X(1,:).';

y2(1)= a0*X1(1,:).';

y3(1)= a0*X2(1,:).';

E(1) = d(1) - a0*X(1,:).';

W(1,:) = (1 -mu*gammax)*a0 + mu*E(1)*conj(X(1,:));

W1(1,:) = (1 -mu*gammax)*a0 + mu*E(1)*conj(X1(1,:));

W2(1,:) = (1 -mu*gammax)*a0 + mu*E(1)*conj(X2(1,:));

if M>1
```

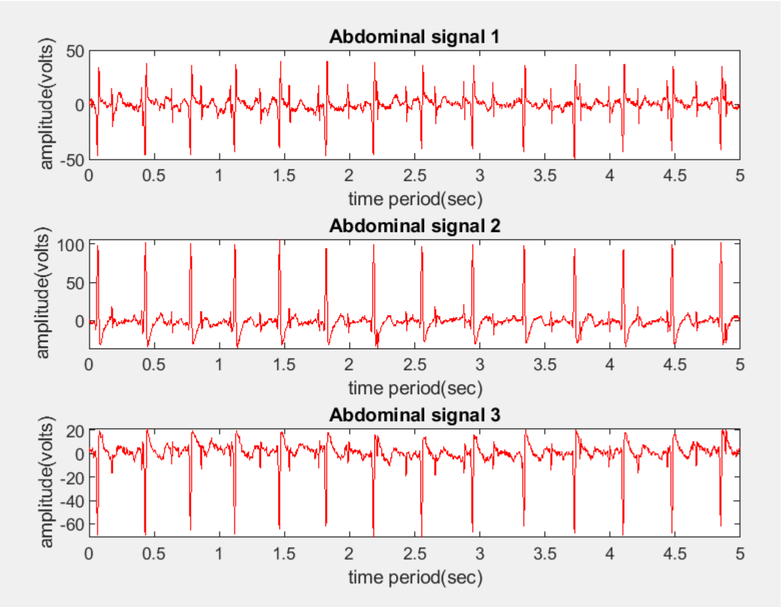
```
for k=2:M-nord+1
y1(k) = W(k-1,:)*X(k,:).';
y2(k) = W1(k-1,:)*X1(k,:).';
y3(k) = W2(k-1,:)*X2(k,:).';
y(k)=(y1(k)+y2(k)+y3(k))/3;
E(k) = d(k) - y(k);
W(k,:) = (1 -mu*gammax)*W(k-1,:) + mu*E(k)*conj(X(k,:));
W1(k,:) = (1 -mu*gammax)*W1(k-1,:) + mu*E(k)*conj(X1(k,:));
W2(k,:) = (1 -mu*gammax)*W2(k-1 ,:) + mu*E(k)*conj(X2(k,:));
end
end;
end
```

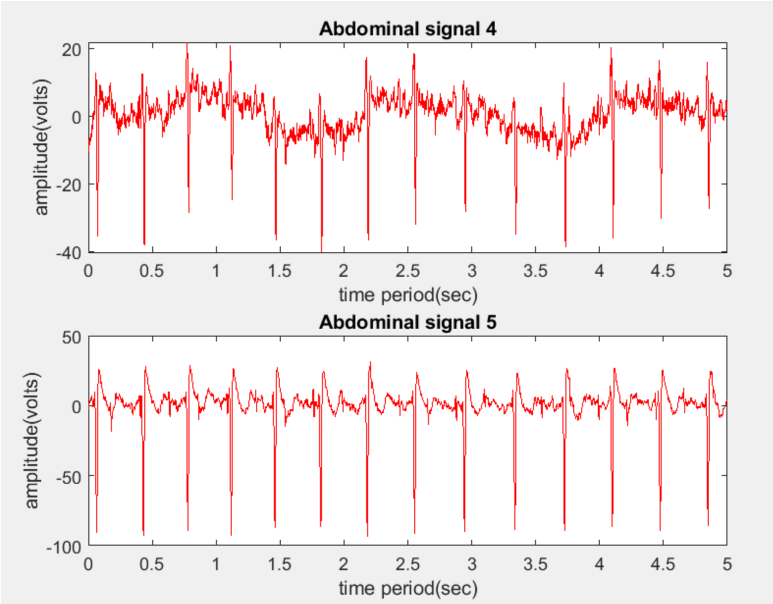
CHAPTER 5
RESULTS AND DISCUSSIONS

After the execution of the MATLAB code for the extraction of fetus heartbeat from mother's heartbeat in both SISO and MISO systems, we got some results. The obtained results will be explained in this chapter.

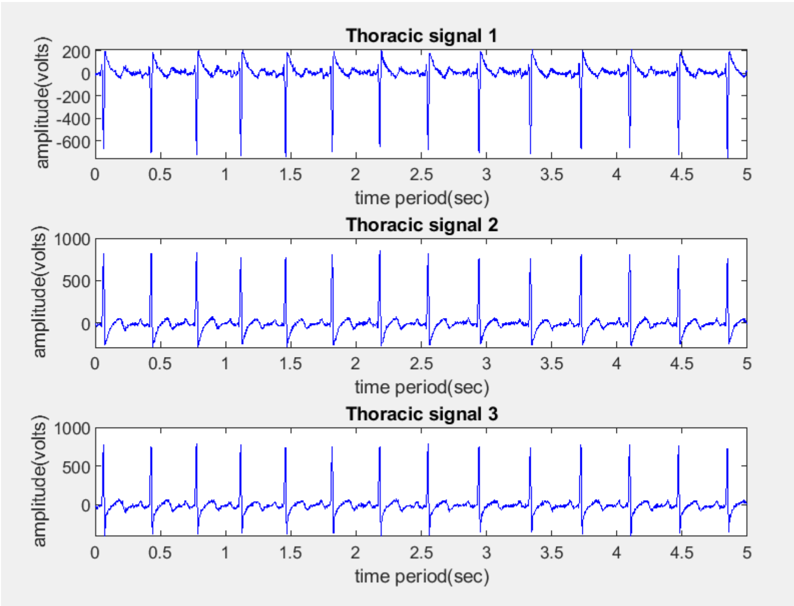
5.1 INPUT SIGNALS

Given data signal consists of five abdomen signals and three thoracic signals. The abdomen signals consist of both mother's heartbeat and fetus heartbeat and the thoracic signals consists of only mother's heartbeat. First both the abdomen signals and thoracic signals were separated from the given data signal. The below shown fig represents the abdomen signals.



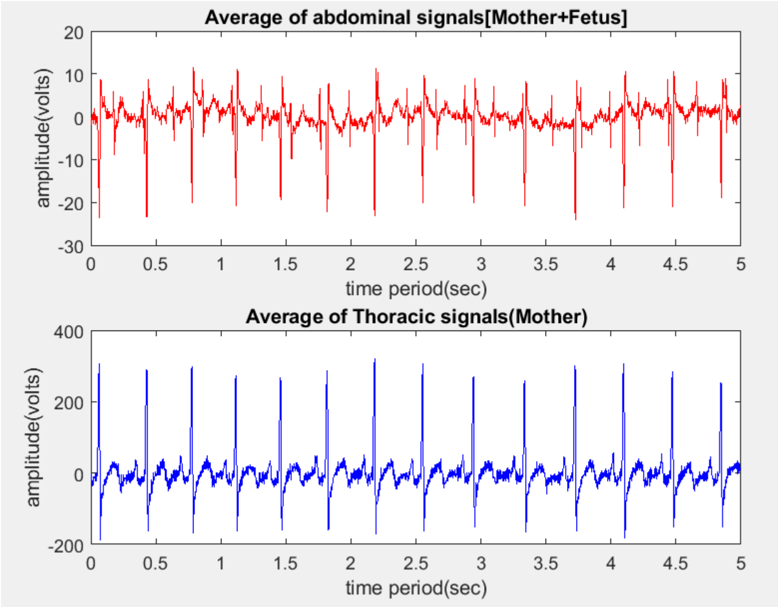


The below shown fig represents all three thoracic signals consisting of only the mother's heartbeat.



5.2 SISO INPUT AND OUTPUT SIGNALS

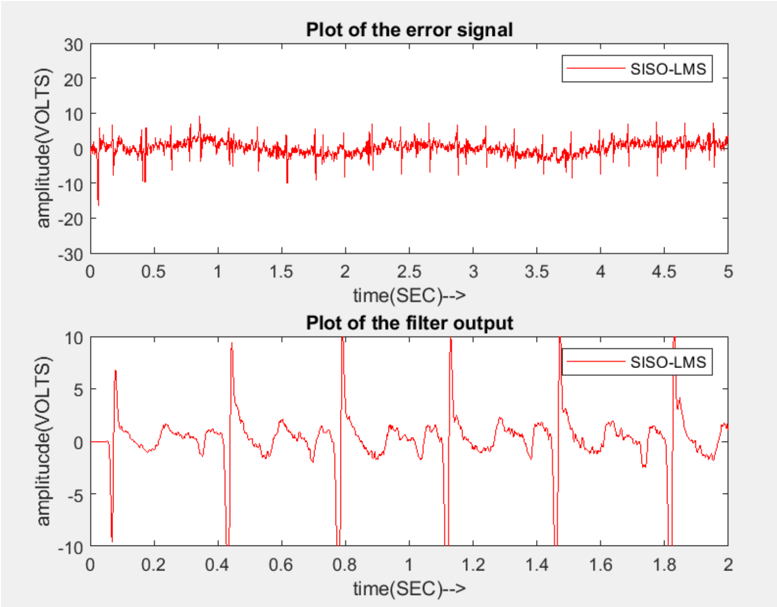
The average of all the thoracic signals was given as reference input. The average of all the abdomen signals was given as primary input. The following plot represents the primary and reference inputs to the SISO system for all algorithms.



The primary and reference inputs are same for the three algorithms. The filter weights were updated using LMS, NLMS and LLMS algorithms.

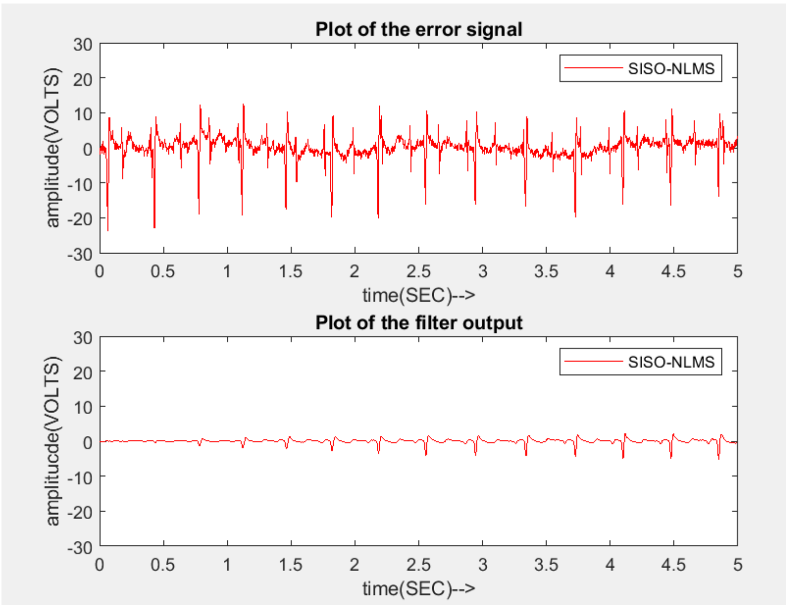
LMS ALGORITHM:

Length of the LMS filter in the SISO implementation is taken as p+1=12.The filter order is taken by trial and error analysis by carefully observing the plots for minimal noise with the help of different filter orders. 19 The length of the filter order (p+1) is 12 the step size varies between 0< μ< 2.1861 ∗ 10-8 The step size taken was μ = 2*10-8 The output of the SISO system when the filter weights were updated using LMS as shown.



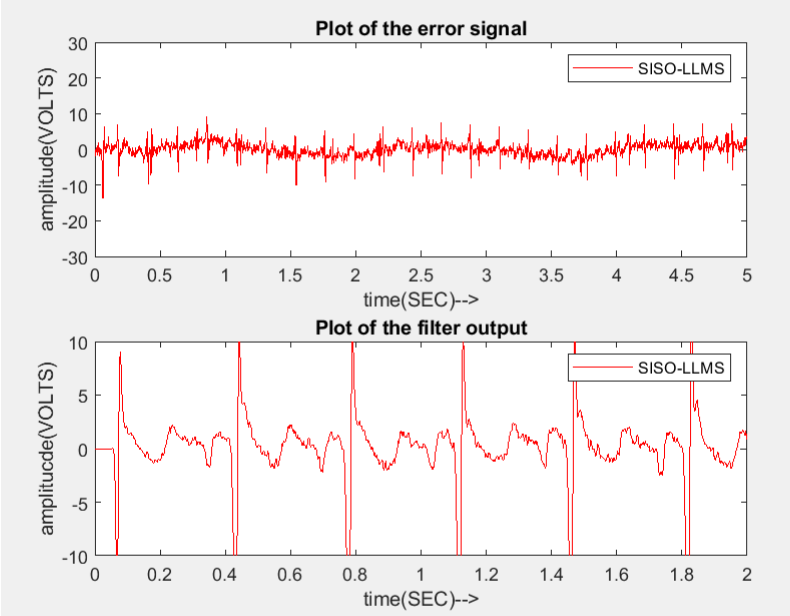
NLMS ALGORITHM:

The step size taken is adaptive w.r.t power of the input signal. We took the step size ‘μ’ values and reference input values into consideration and took the normalized step size with in the range. The normalized step size taken was = 0.009 The output of the SISO system when the filter weights were updated using NLMS as shown.

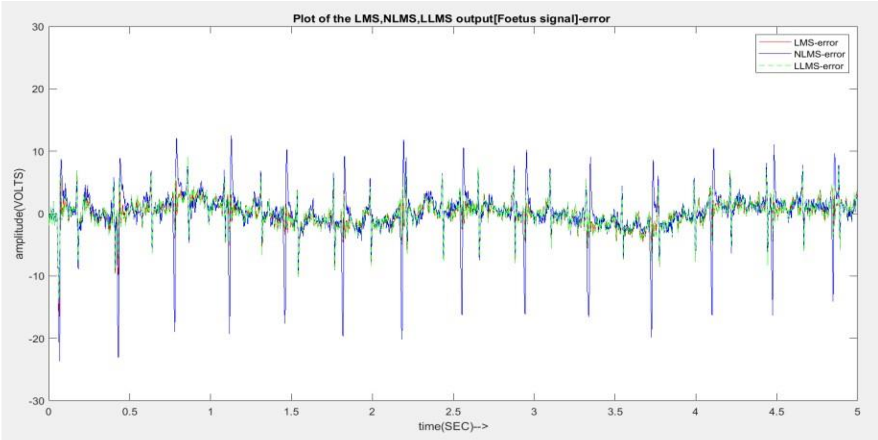


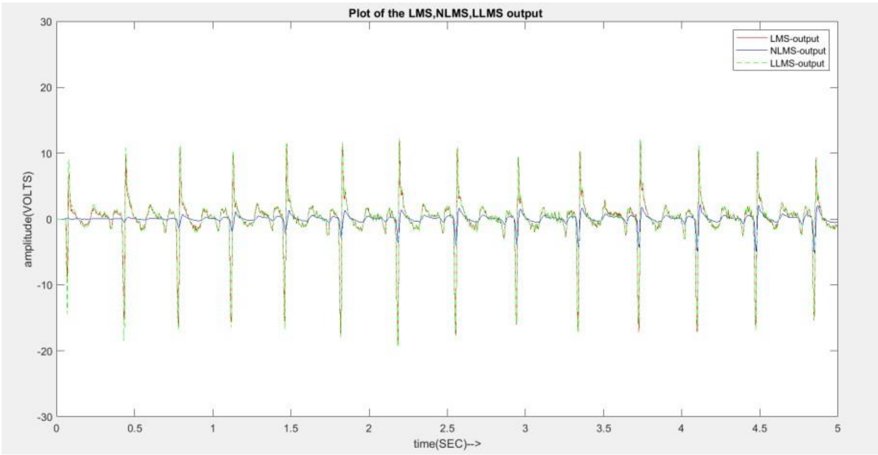
LLMS ALGORITHM:

A leaky coefficient to give stability to LMS adaptive filter which forms a LLMS algorithm. We know that LLMS converges when 0< γ << 1 When using the LLMS algorithm the leaky coefficient value taken was = 0.99998 The following figures represents the filtered output of the SISO system for all the algorithms.



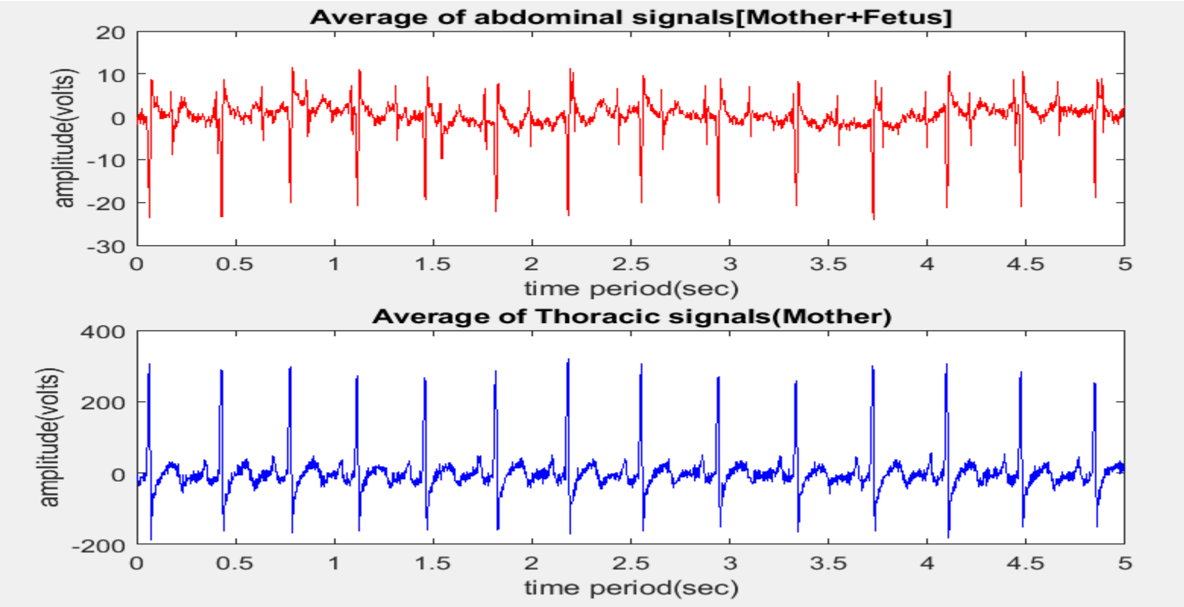
We used three different algorithms such as LMS, NLMS and LLMS. We know that, the rate of convergence can be known by calculating the time taken for an algorithm to filter the signal. The convergence rate order is LLMS>LMS> NLMS From the figure below, we can observe that peaks in the fetus ECG output plot for NLMS algorithm are smaller compared to LMS and LLMS. Here peaks represent the mother's heartbeat. 21 From the Figure below, we can also observe that NLMS algorithm extracted the fetus heartbeat with minimal noise (peaks are minimum) . From this we can say that NLMS algorithm had better performance in extracting the fetus heartbeat from the mother's heartbeat when compared to other algorithms LMS and LLMS.





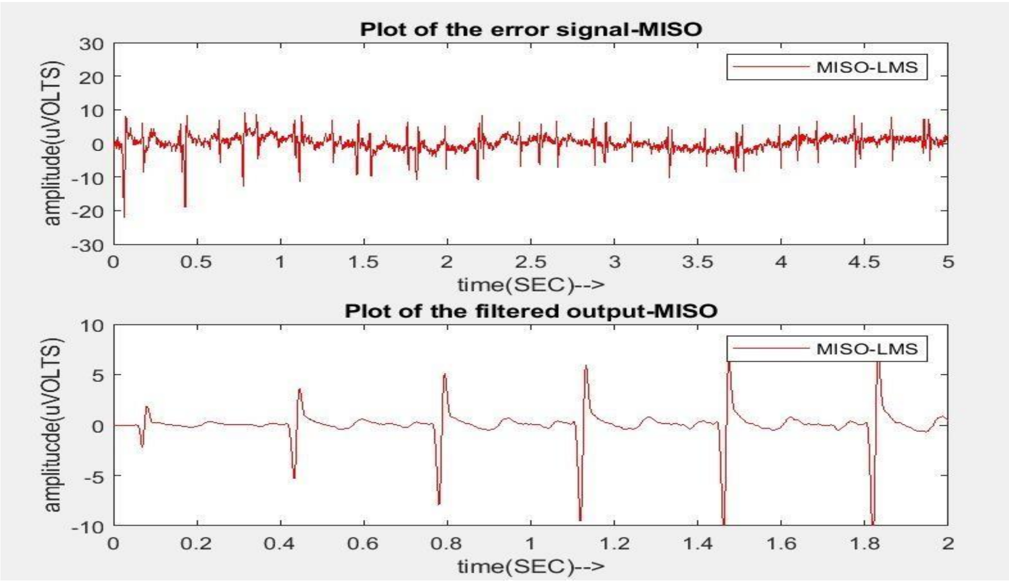
5.3 MISO INPUT AND OUTPUT SIGNALS

Adaptive noise cancellation system was implemented in MISO system. The filter weights were updated by using different adaptive algorithms such as least mean square algorithm (LMS), normalized least mean square algorithm (NLMS) and leaky least mean square algorithm (LLMS). We know that MISO is a multiple input system. So, every thoracic signal is given as reference signal to the individual adaptive filter first and then the average is calculated. The average of all the abdomen signals was given as primary input. The fig below represents the primary and reference inputs to the MISO system for all algorithms.



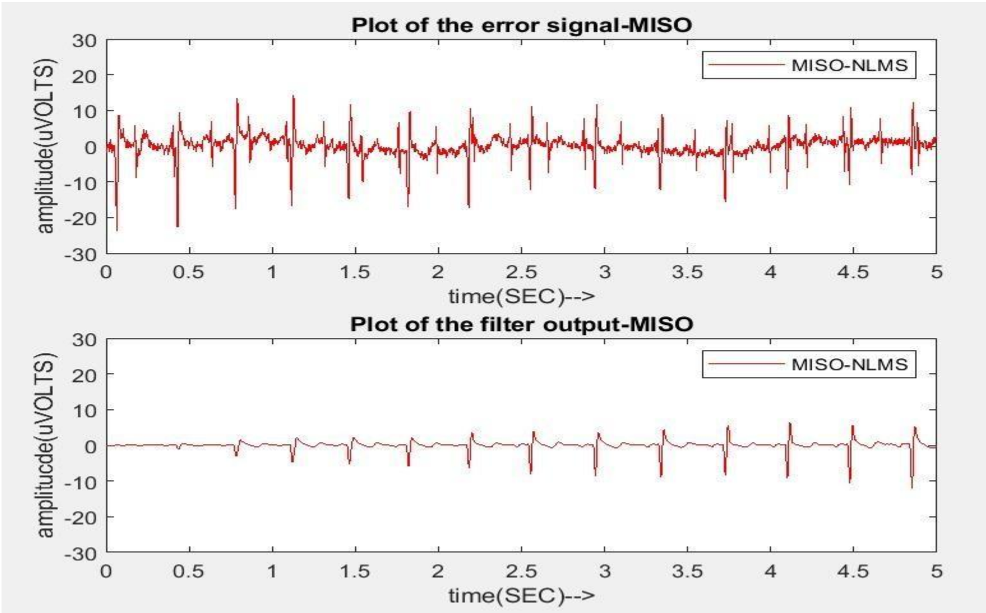
LMS ALGORITHM:

Length of the LMS filter in the MISO implementation is taken as p+1=12.The filter order is taken by trial and error analysis by carefully observing the plots for minimal noise with the help of different filter orders. The length of the filter order (p+1) is 12 the step size varies between0< μ< 2.1861 ∗ 10-8 The step size taken was μ = 2*10-8



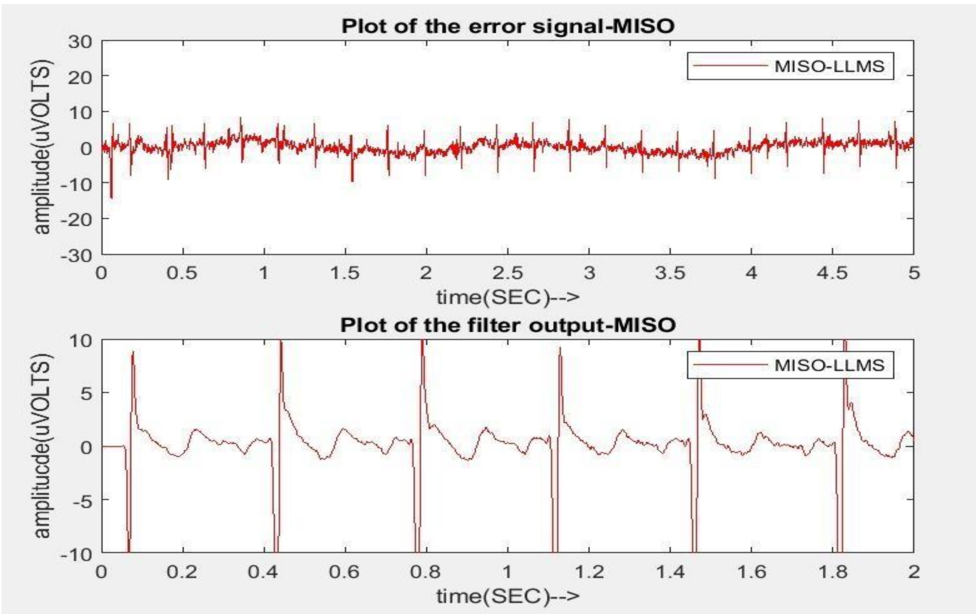
NLMS ALGORITHM:

The normalized step size taken was = 0.001

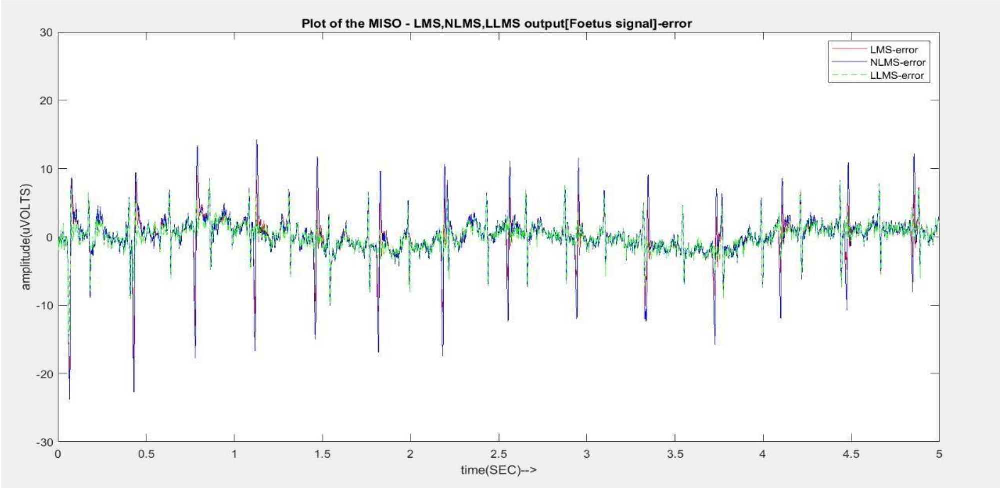


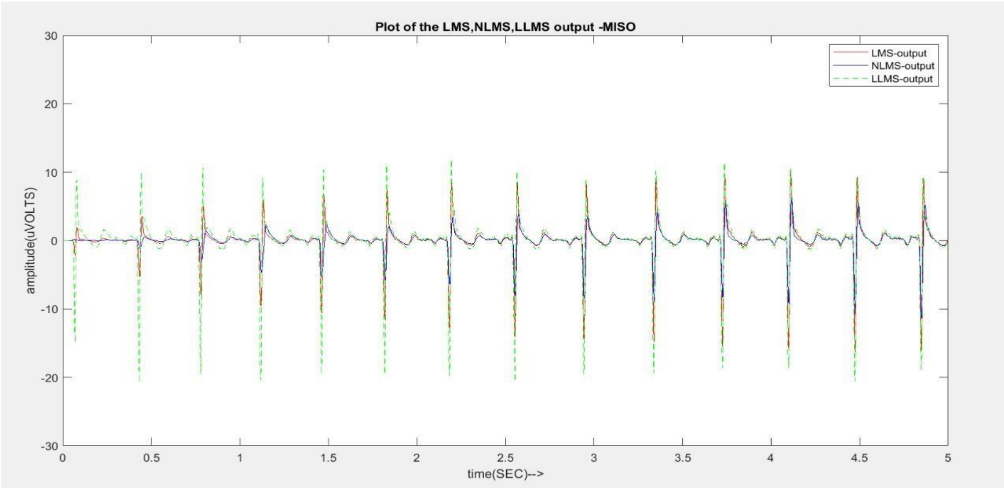
LLMS ALGORITHM:

The Fig below represents the filtered output of the MISO system for all the algorithms.



We used three different algorithms such as LMS, NLMS and LLMS for MISO. The convergence speed for all the algorithms can be known by using timer functions like (tic-toc) while implementing the code. We know that, the rate of convergence can be known by calculating the time taken for an algorithm to filter the signal.





The convergence rate order is LLMS>LMS>NLMS.

From the Fig 5.8, we can observe that peaks in the fetus ECG output plot for NLMS algorithm are smaller compared to LMS and LLMS. Here peaks represent the mother's heartbeat.

From the Fig 5.8, we can also observe that NLMS algorithm extracted the fetus heartbeat with minimal noise .

From this we can say that NLMS algorithm had better performance in extracting the fetus heartbeat from the mother's heartbeat when compared to other algorithms LMS and LLMS.

CHAPTER 6

CONCLUSIONS

The adaptive noise cancellation technique (ANC) successfully separated the fetal heartbeat from the combined maternal and fetal heartbeat, even in scenarios where the signal strengths of both the mother and fetus varied. This success was observed in both the single input single output (SISO) and multiple input single output (MISO) systems.

Within the SISO system, the algorithms used for updating the filter weights included the least mean square (LMS), normalized least mean square (NLMS), and leaky least mean square (LLMS) algorithms. Among these, the LLMS algorithm demonstrated faster convergence speed in extracting the fetal ECG compared to LMS and NLMS. However, the NLMS algorithm outperformed the others in fetal ECG extraction with minimal noise.

In the multiple input single output (MISO) system, the number of filters used increased compared to the SISO system, necessitating more filtering. The MISO system also exhibited faster convergence speed with the LLMS algorithm and superior performance with minimal noise using the NLMS algorithm, when compared to the LMS and LLMS algorithms.

CHAPTER 7
REFERENCES

[1] H. H. S. s. Mansoureh Ghodsi, "Extracting fetal heart signal from noisy maternal ECG by multivariate singular spectrum analysis," Statistics and its interface, vol. 3, pp. 399-411, 2010.

[2] B. P. A. A. B. Prasanth K, "Fetal ECG extraction using adaptive filters," International journal of advanced research in electrical, electronics and instrumentation engineering, vol. 2, no. 4, pp.1483-1487, April 2013.

[3] D.-l. L. J.-q. Z. Y.-j. Z. Shi-jin Liu, "Extraction of fetal electrocardiogram using recursive least squares and normalized least mean square algorithms," 3rd international conference on advanced computer control, pp. 333- 336, 2011.

[4] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "Fetal electrocardiogram extraction by blind source subspace separation," IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 47, NO. 5, MAY 2000

[5] R. B. &. J. Gawande, "Fetal ECG extraction using wavelet transform," vol. 1, no. 4, pp. 19-22, 2013.

[6] S. L. G. L. Y. Z. D. M. XiaoPing Zeng, "Fetal ECG extraction by combining single-channel SVD and cyclostationarity-based blind source separation," vol. 6, pp. 367-376, August 2013.

[7] S. L. J. Z. Yanjun ZENG, "Extraction of fetal ECG signal via Adaptive noise cancellation approach," pp. 2270-2273.

[8] B. D. M. J. V. Lieven De Lathauwer, "Fetal electrocardiogram extraction by blind source subspace separation," vol. 47, pp. 567-572, May 2000.

[9] M. H. Hayes, Statistical Digital Signal Processing and Modeling, New york: John Wiley & sons. Inc, 1996.