FEATUS HEARTBEAT EXTRACTION

Course: Applied Adaptive Signal Processing(ET2583)

Project Supervisor: Gertsovich, Irina

Department of Applied Signal Processing

Group Number:12

Project Members:

Veluru, Chaithanya Kumar Reddy	Sana, Raj Sekhar
950826-6450	950312-7210
Email: chva16@student.bth.se	Email:rasa16@student.bth.se

ABSTRACT

This report presents the detail review and implementation of Fetal Electro Cardiogram Extraction from Mother's womb. The Fetal ECG is constructed by a pregnant woman's multiple thoracic and abdominal ECG signals. Adaptive Noise Cancellation Technique is used for estimating signals corrupted by the additive noise or interference. Least Mean Square algorithm, Normalized Least Mean Square algorithm and Leaky Least Mean Square algorithm are used to update the weights of adaptive filter in the Adaptive Noise Cancellation system. This method is implemented in both single Input and Single Output system and Multiple Input and Single Output system. In both systems, we get better performance when Leaky LMS algorithm is used compared to LMS and NLMS algorithms.

List of Figures

List of Tables

TABLE OF CONTENTS

CHAPTER 1

Introduction

Like for adults, it should be possible to visualize the electrical activity of a fetal heart: the *fetal electrocardiogram* (FECG) contains important indications about the health and condition of the fetus. In this respect, analysis of the (instantaneous) *fetal heart rate* (FHR) has become a routine procedure for the evaluation of the well-being of the fetus. The cardiac waveform reveals important diagnostic information as well, e.g., for the diagnosis of arrhythmia [4].

During delivery accurate recordings can be made by placing an electrode on the fetal scalp. However as long as the membranes protecting the child have not been broken (*antepartum*), one should look for noninvasive techniques. Among the different approaches examination of the FECG from ECG-recordings measured on the mother's skin (*cutaneous* recordings) plays an important role [4].

The dominant noise source is the MECG signal which is larger in amplitude when compared with the fetal ECG. So the removal of MECG is more important for the processing of the signal for fetal monitoring and diagnosing the fetus. The extraction of fetal heart beat from mother's womb is difficult because it is corrupted with mother's heart beat signal and in addition to this there are other types of noise and overlapping frequencies which makes it a difficult task [13].

Various methods are proposed to extract the desired fetus signal. Blind source subspace separation (BSS) [8], Singular value decomposition(SVD) [6], Wavelet transform [5], Adaptive noise cancellation approach [7], using adaptive algorithms [1,8], multivariate singular spectrum analysis [1] are some of the methods used to extract the FECG.

The adaptive filters have the ability to adjust their impulse response to filter out the correlated signal in the input. They require little or no prior knowledge of the signal and noise characteristics. Moreover, adaptive filters have the capability of adaptively tracking the signal under non-stationary conditions and taking into account the factors like ease of implementation and real time executive, extraction of FECG signals using adaptive noise cancellation approach whose filter coefficients are updated using adaptive algorithms is a more suitable approach [14].

The adaptive algorithms which are used for updating filter coefficients are LMS (least mean square) algorithm, NLMS (normalized least mean square) algorithm, and LLMS (leaky least mean square) algorithm.

1.1 Problem Formulation:

The main problems involved in the implementation of project are

- Separating the Fetal Electrocardiogram from Maternal Electrocardiogram with minimal noise.
- Implementation of the project in SISO and MISO systems.
- Analyzing the performance of project with various adaptive filtering methods.

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 attainable 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.

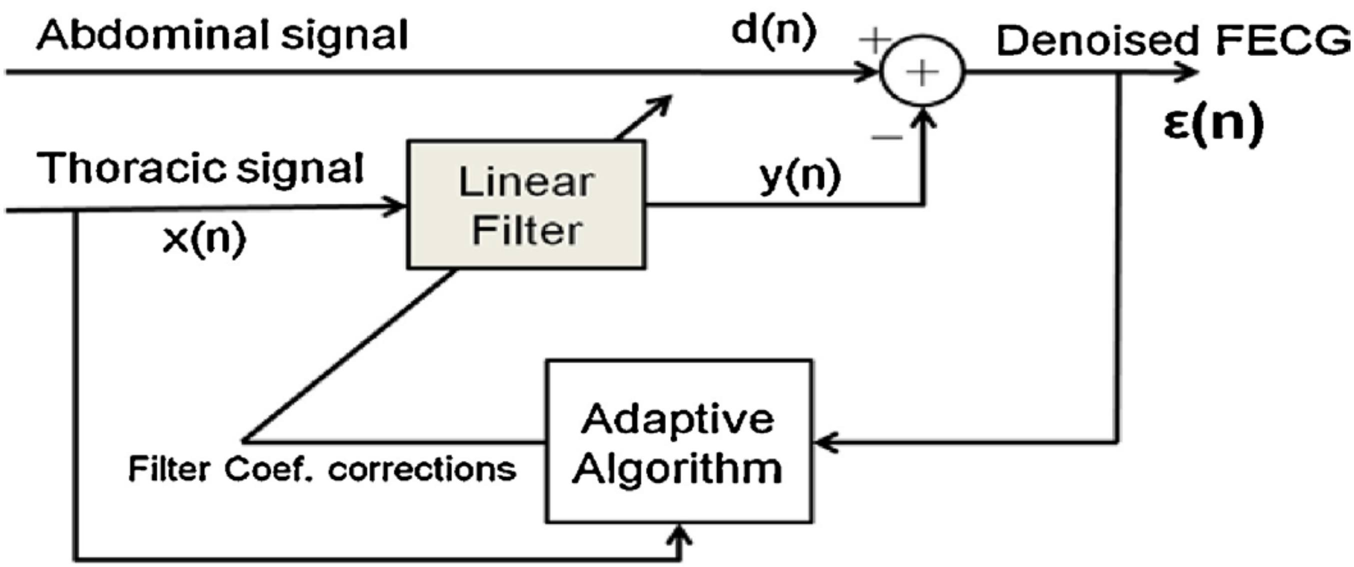

Fig 2.1: ANC system for FECG extraction

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.

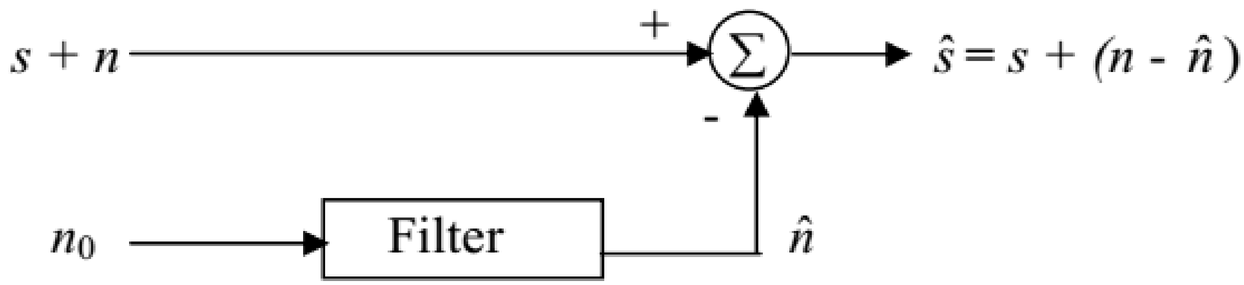

Fig 2.2: Filtering Noise from the signal

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 $\hat{n}$ should be an exact replica of n. If it were possible to know the relationship between n and $\hat{n}$, 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 $\hat{n}$ 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 priori 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 or circumstances. Adaptive algorithms are able to intelligently adjust their activities in light of changing circumstances to achieve the best possible outcome.

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 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 tapweight 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.

$$R'(n) = u(n)u^H(n) \qquad (1)$$

$$p'(n) = u(n)\, d^*(n) \qquad (2)$$

Correspondingly, the instantaneous estimate of the gradient-vector is

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

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:

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

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

$$e(n) = d(n) - u^H(n)w'(n) \tag{5}$$

$$w'(n+1) = w'(n)+\mu u(n)e^*(n) \tag{6}$$

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 μu(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. However, the LMS algorithm, being recursive in nature, effectively averages these estimates, in some sense, during the course of adaptation. 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 $w_{0.}$ Consequently, use of LMS results in a mean-squared error J(∞) after a large no. of iterations.

Excess Mean-Squared Error:

Excess mean-squared error is defined as the amount by which the actual value of $J(\infty)$ is greater than J_{min}.

2.3.2 Normalized Least Mean Square Algorithm (NLMS):

The main drawback of the "pure" 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 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 filter and also it gives gradient noise amplification problem. So the step size is normalized in NLMS to overcome this problem

- Modified formula for convergence factor
- $\mu(n) = \frac{\beta}{c+||X_n||^2}$ (7)
    - μ(n) = step size
    - β = normalized step-size (0 < β < 2)

– c = safety factor (small positive constant)

- Weight vector:

From equation(6), we get

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)u(n)\mathbf{e}^*(n) \text{ or}$$

$$w(n+1) = w(n) + \frac{\beta}{||x(n)||^2}u(n)e(n) \qquad (8)$$

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 “leaky” LMS algorithm can be written as

$$w(n+1) = (1-\mu r)w(n) + \mu e(n)u(n) \qquad (9)$$

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

2.3.4 Comparison of Least Mean Square Algorithms

LMS Algorithm	**Normalized LMS Algorithm**	**Leaky LMS Algorithm**
It has fixed step size μ	It has data dependent adaptive step size β.	It has leaky factor γ along with step size μ
For convergence 0< μ<(2/total input power)	For convergence 0< β.<2	To mitigate the coefficients overflow. The range of Leaky factor should be 0<γ<<1
Convergence speed is slow	Convergence speed is comparably high	The stability of the system is high
The filter coefficients are updated by the equation $w_{n+1} = w_n + \mu e(n)x^*(n)$	The filter coefficients are updated by the equation $w_{n+1} = w_n + \beta e(n)\frac{x^*(n)}{\lVert x(n)\rVert^2}$	The filter coefficients are updated by the equation $w_{n+1} = (1-\mu\gamma)w_n + \mu e(n)x^*(n)$

CHAPTER 3

PROJECT IMPLEMENTATION

In fetus 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 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 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

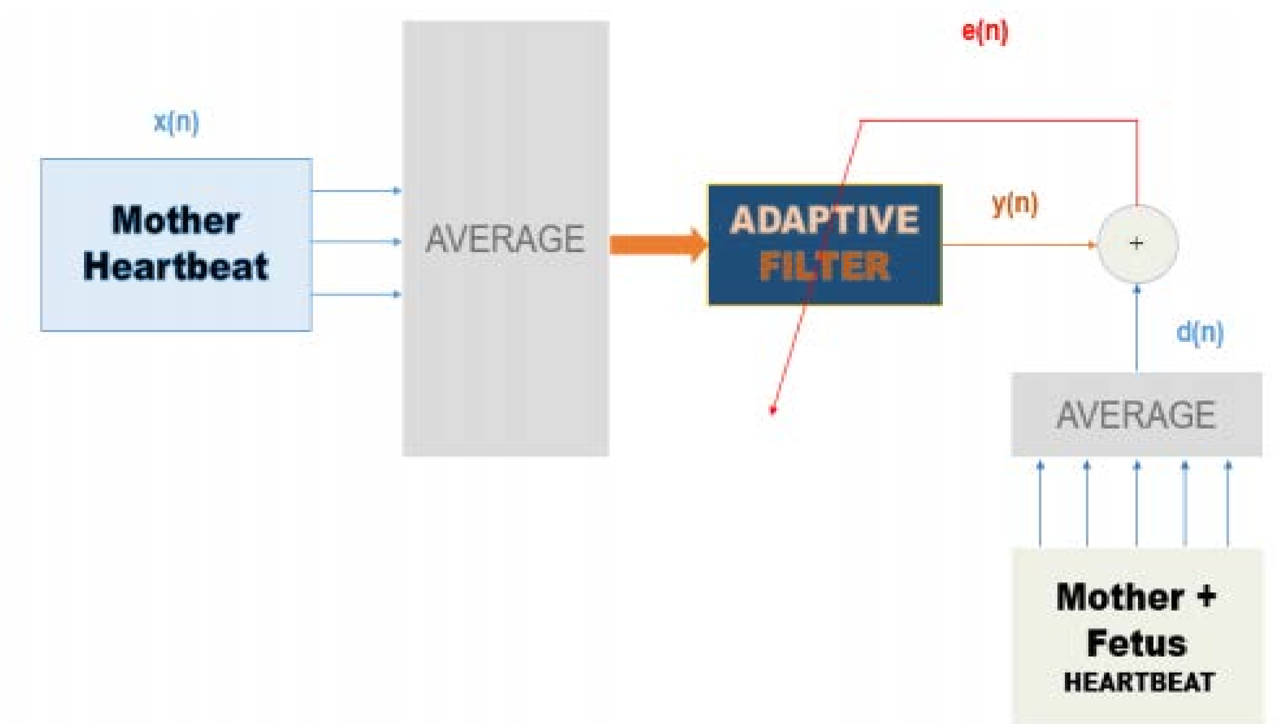

Fig 3.1 SISO Implementation

Here output e(n) = d(n) – y(n)

y(n) is the filter output

$$d(n) = \frac{(a1+a2+a3+a4+a5)}{5}$$

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

3.2 Multiple Input and Single Output System:

- MISO or the multiple input and single output system is a scheme of RF wireless communication system in which there are multiple transmitting antennas at the source and single receiving antenna. MISO is also termed transmit diversity [11].
- Primary signal:

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

- Reference Signal:

Thoracic signals 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 [11,12].

Disadvantages:

- Complexity of the system increases as the number of filter are increased
- Increase in number of adaptive filters leads to costly affairs.
- Difficulty in system implementation due to more number of filters [11,12].

Applications:

- MISO scheme has various applications in Digital television, W-lans, metropolitan area networks (MANs), and mobile communications[11,12].

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

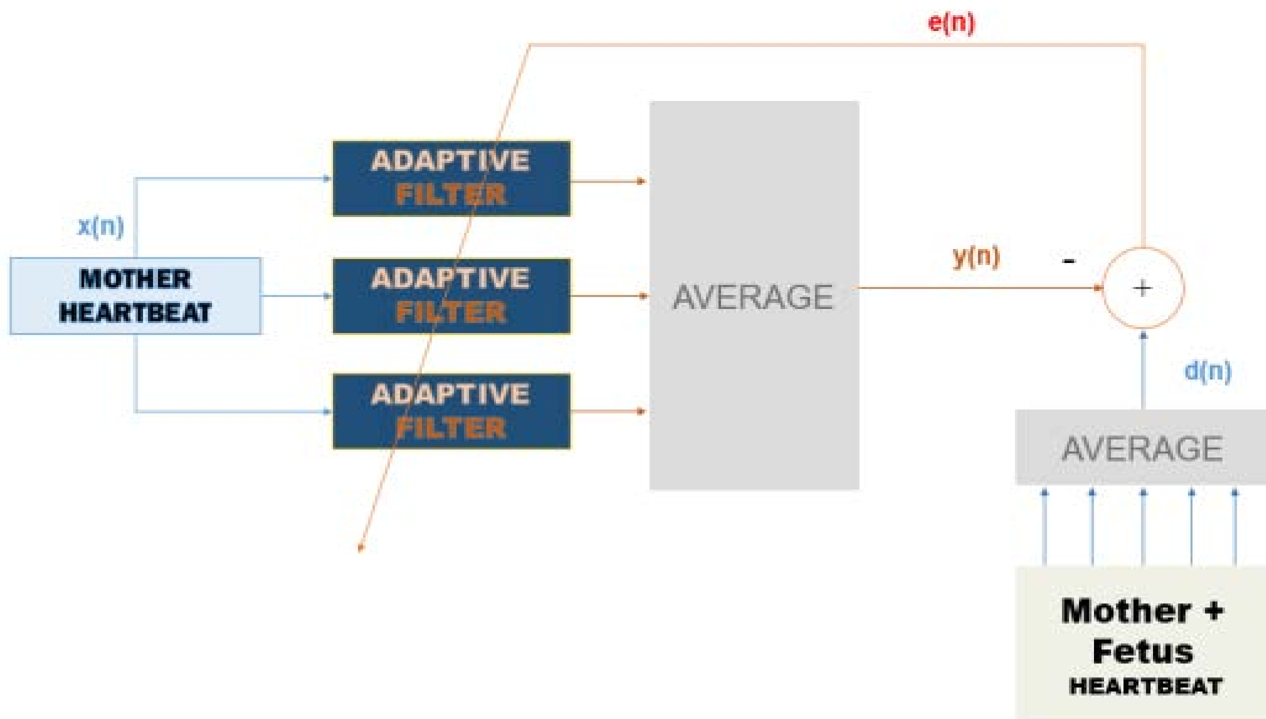

Fig 3.2: MISO implementation

As shown in the Fig, 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 is given as primary input. As a result, fetus heartbeat is obtained as output.

Here output e(n) = d(n) – y(n)

y(n) is the filter output

$$y(n) = \frac{t1+t2+t3}{3}$$

where t1, t2, t3 are the filtered outputs

$$d(n) = \frac{(a1+a2+a3+a4+a5)}{5}$$

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

Table 2: Comparison between SISO and MISO

CHAPTER 4

SOURCE CODE IMPLEMENTATION

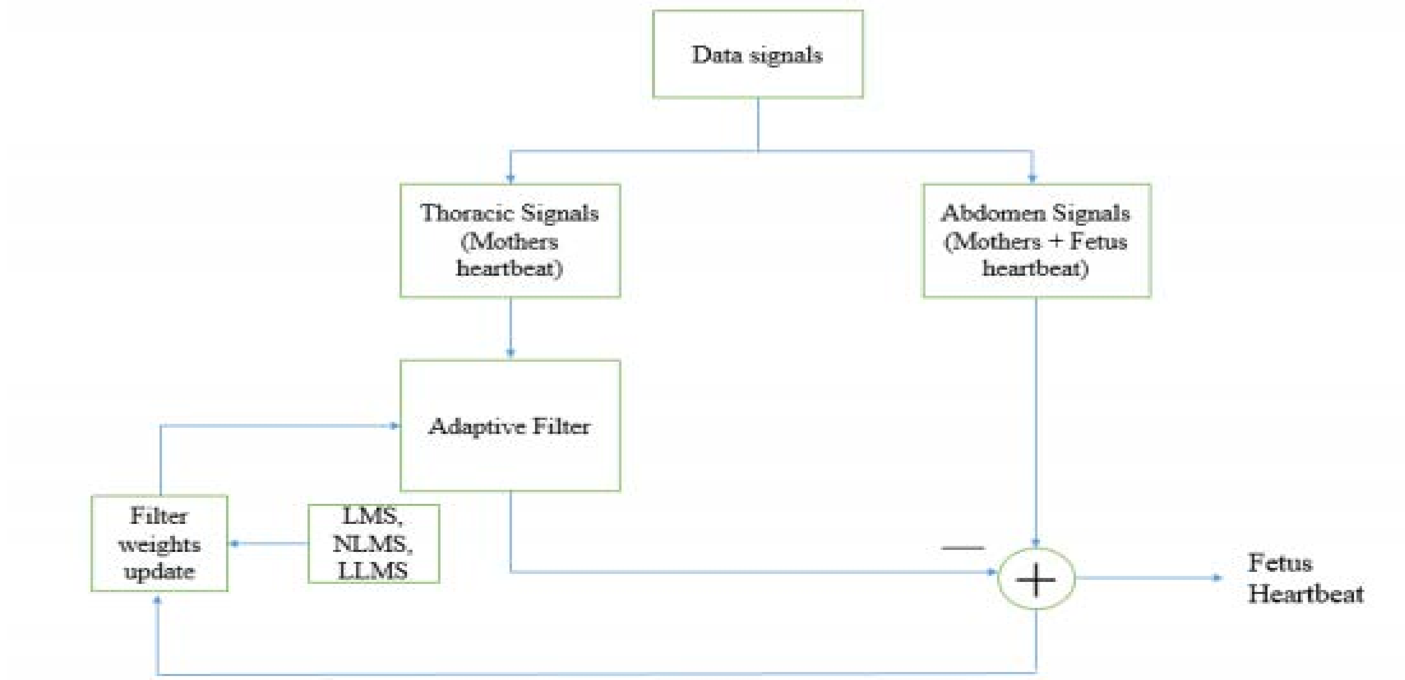

Fig 4.1: Schematic representation of source code

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

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. 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 fig 5.1 represents the abdomen signals

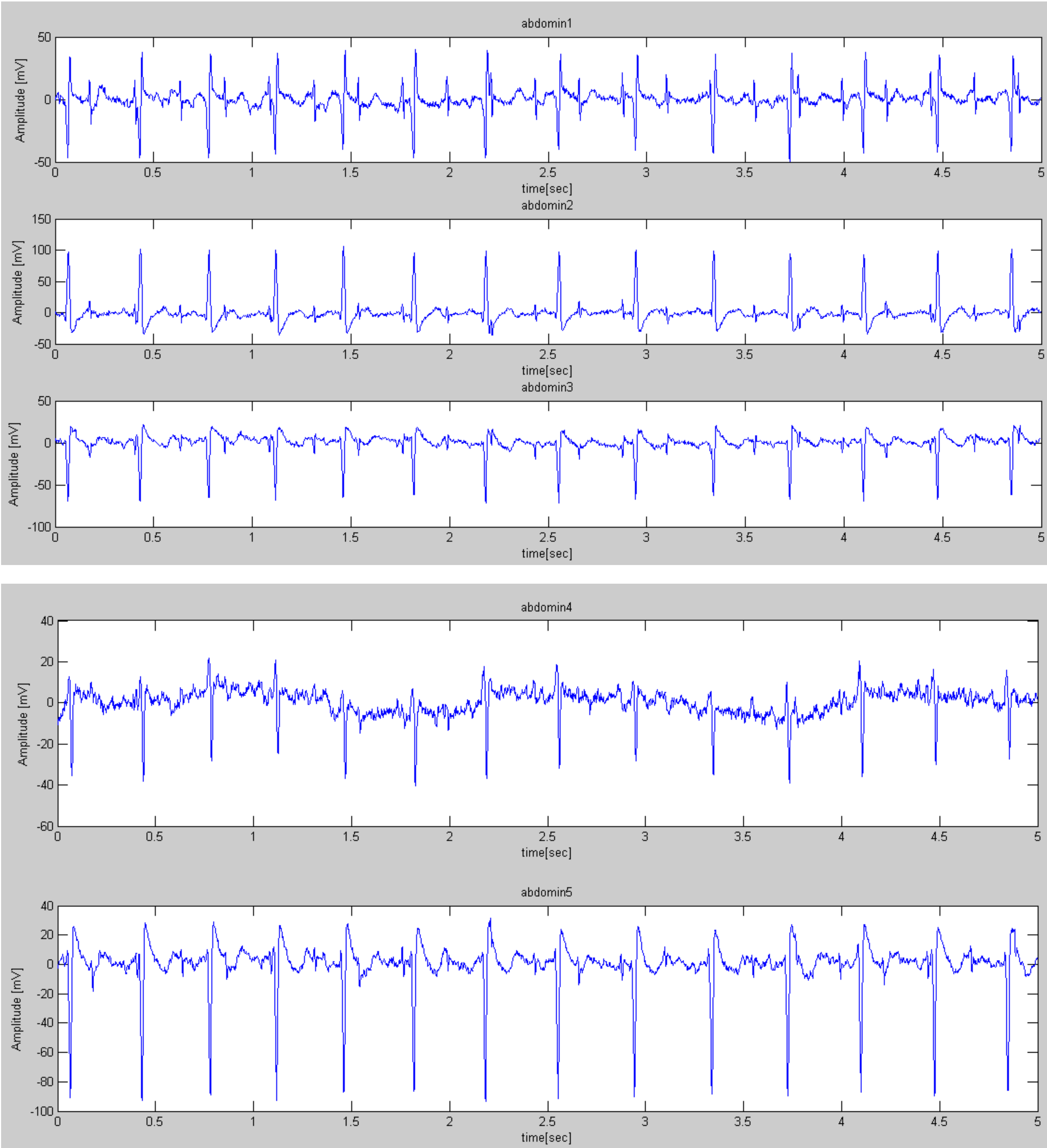

Fig 5.1: Input Abdomen Signals:

- The fig 5.2 represents the three thoracic signals consisting of only mother's heartbeat.

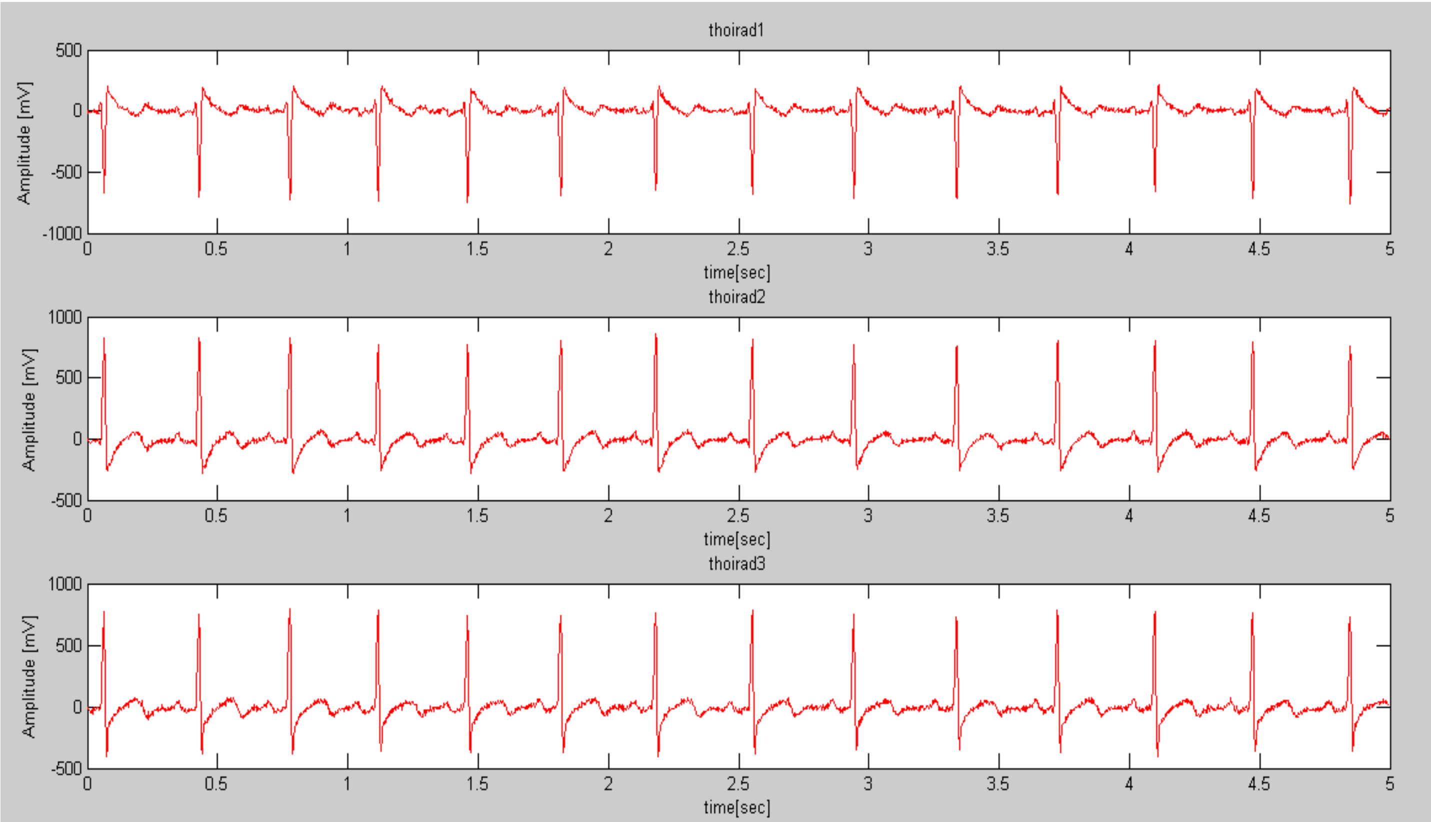

Fig 5.2: Input Thoracic Signals

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.

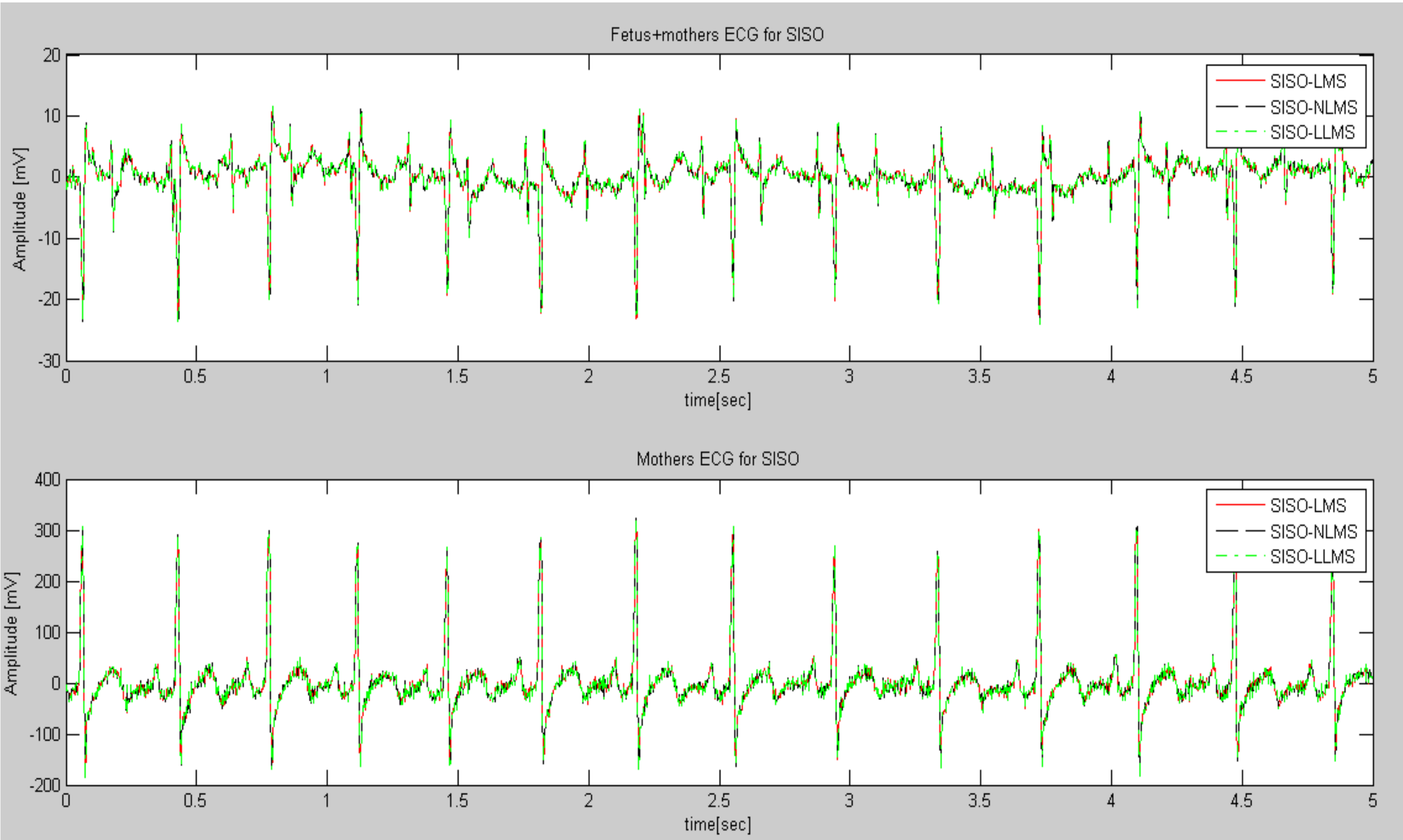

Fig 5.3: SISO Input Signals

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

LMS Algorithm:

$$0 < \mu < \frac{2}{(p+1)E\{|x(n)|^2\}}$$

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.

The length of the filter order (p+1) is 12 the step size varies between

$0 < \mu < 2.1861 * 10^{-8}$
The step size taken was $\mu = 2*10^{-8}$

The output of the SISO system when the filter weights were updated using LMS is shown in fig 5.4 and fig 5.5

NLMS Algorithm:

$$\mu(n) = \frac{\beta}{\|x(n)\|^2}$$

- 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 $\beta = 0.009$
  The output of the SISO system when the filter weights were updated using NLMS is shown in fig5.4 and fig5.5

LLMS Algorithm:

- A leaky coefficient γ to give stability to LMS adaptive filter which forms a LLMS algorithm.
  We know that LLMS converges when $0 < \gamma << 1$
- When using the LLMS algorithm the leaky coefficient value taken was $\gamma = 0.99998$
- The following Fig 5.4 and Fig 5.5 represents the filtered output of the SISO system for all the algorithms.

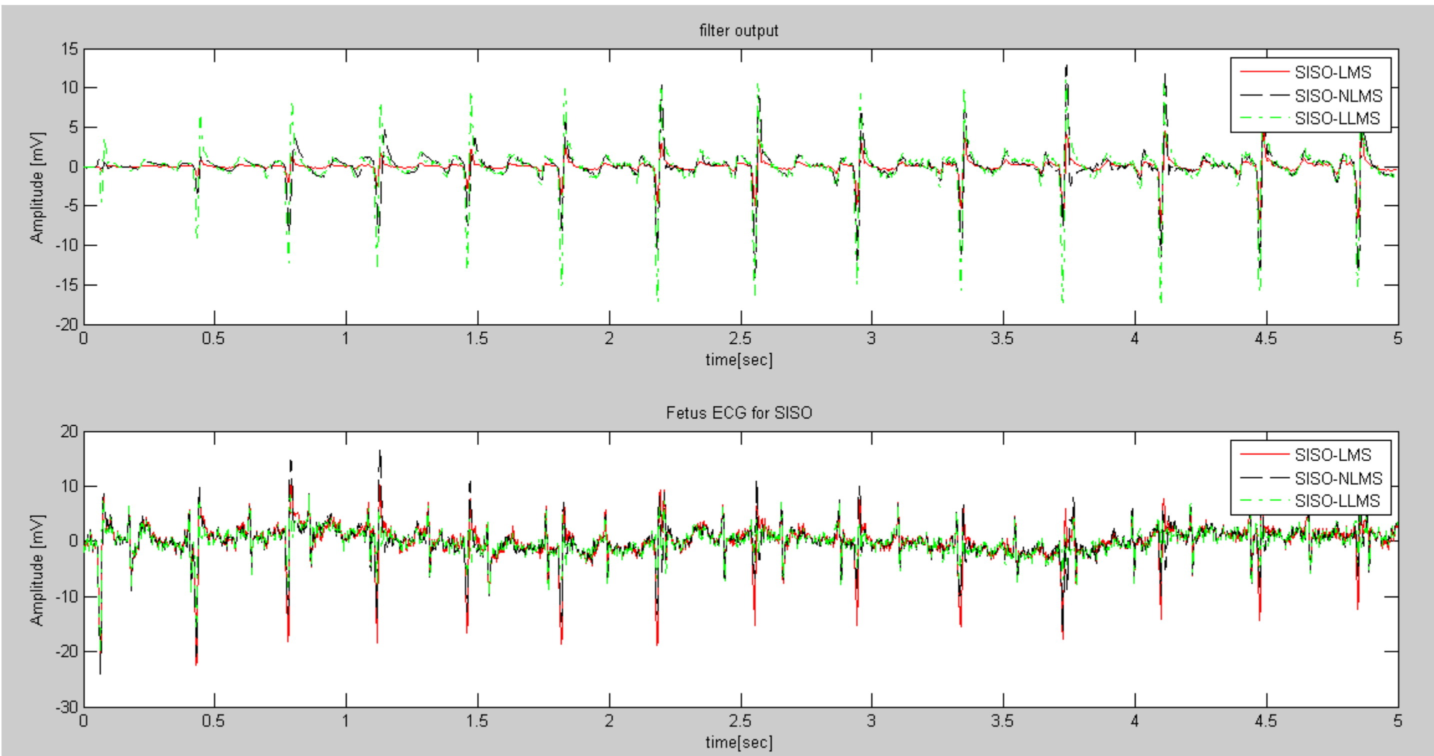

Fig 5.4: SISO output Signals

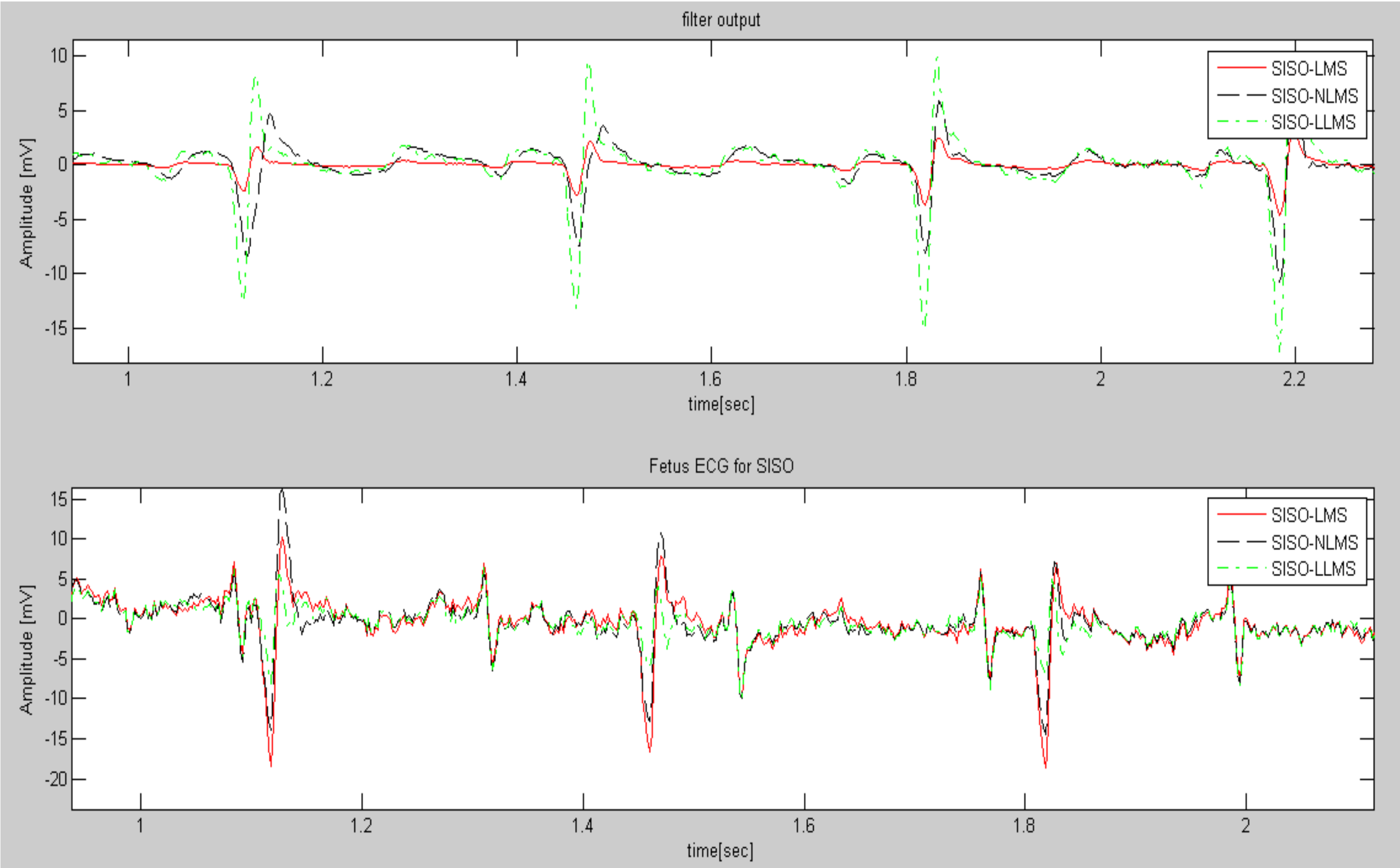

Fig 5.5: Zoomed SISO output

- We used three different algorithms such as LMS, NLMS and LLMS. The convergence speed for all the algorithms can be known.

Function Name	Calls	Total Time	**Self Time***	Total Time Plot (dark band = self time)
siso_finalcode	1	1.817 s	0.636 s	
nlms	1	0.059 s	0.059 s	
scribe.legend.legend	2	0.337 s	0.058 s	
scribe.legend.methods>strsize	168	0.059 s	0.057 s	
lms	1	0.054 s	0.052 s	
llms	1	0.051 s	0.050 s	

- We know that, the rate of convergence can be known by calculating the time taken for an algorithm to filter the signal. Here in SISO system, leaky least mean square algorithm (LLMS) has more convergence speed as it takes least time to filter the signal compared to LMS and NLMS.

- The convergence rate order is LLMS>LMS> NLMS
- From the Fig 5.5, we can observe that peaks in the fetus ECG output plot for LLMS algorithm are smaller compared to LMS and NLMS. Here peaks represent the mother's heartbeat.
- From the Fig 5.5, we can also observe that LLMS algorithm extracted the fetus heartbeat with minimal noise (peaks are minimum)
- In the fetus output at x-coordinate 2.182 we see that LLMS output has small peak. In LMS and NLMS output peaks are large due to presence of some amount of mother's heartbeat.

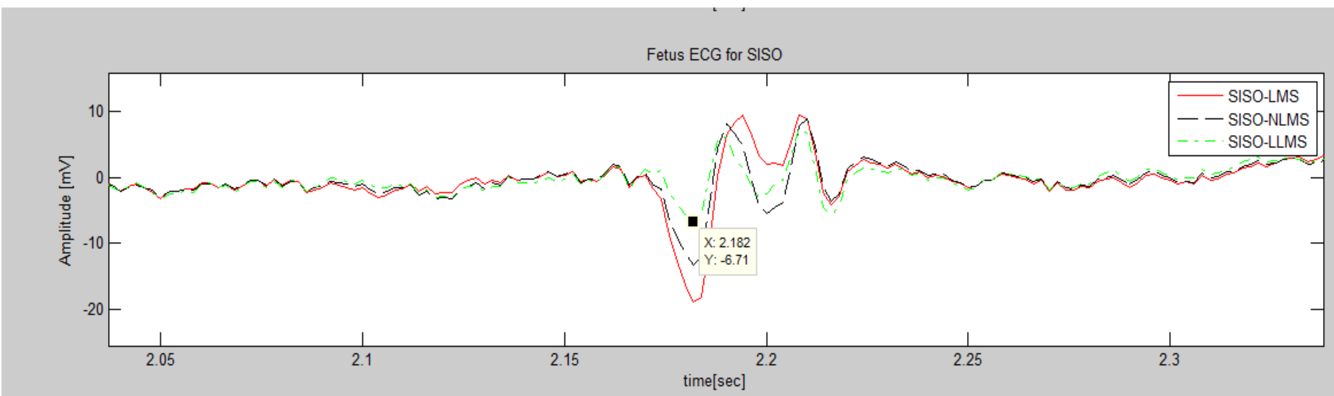

Fig 5.6: Fetus Heartbeat signal

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

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 5.7 represents the primary and reference inputs to the MISO system for all algorithms.

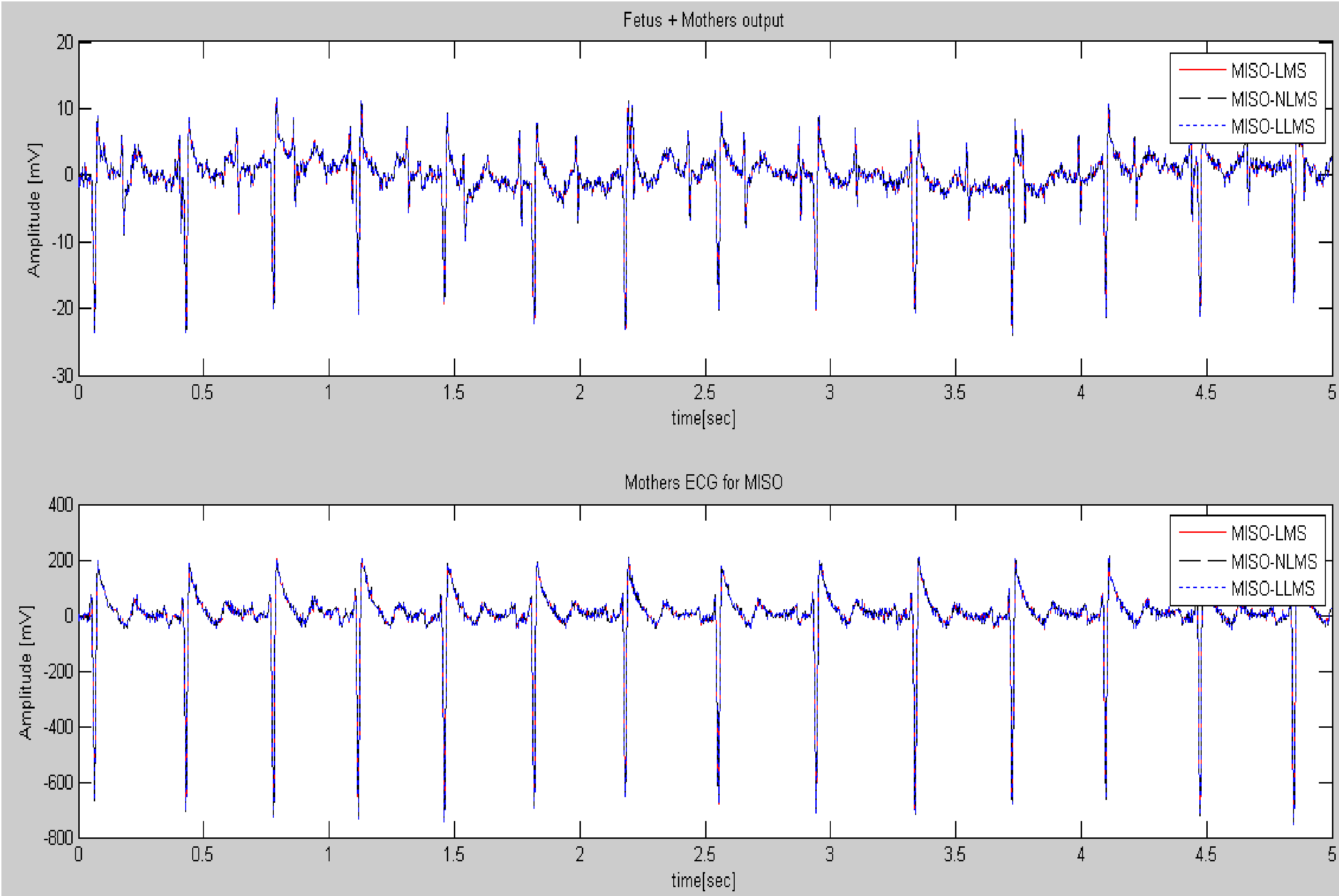

Fig 5.7: MISO Input Signals

LMS Algorithm:

$$0 < \mu < \frac{2}{(p+1)E\{|x(n)|^2\}}$$

- 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 $\mu = 2*10^{-8}$

NLMS Algorithm:

$$\mu(n) = \frac{\beta}{\|x(n)\|^2}$$

- The normalized step size taken was β = 0.005

LLMS Algorithm:

- The leaky factor taken was γ = 0.001
- The Fig 5.8 and Fig 5.9 represents the filtered output of the MISO system for all the algorithms.

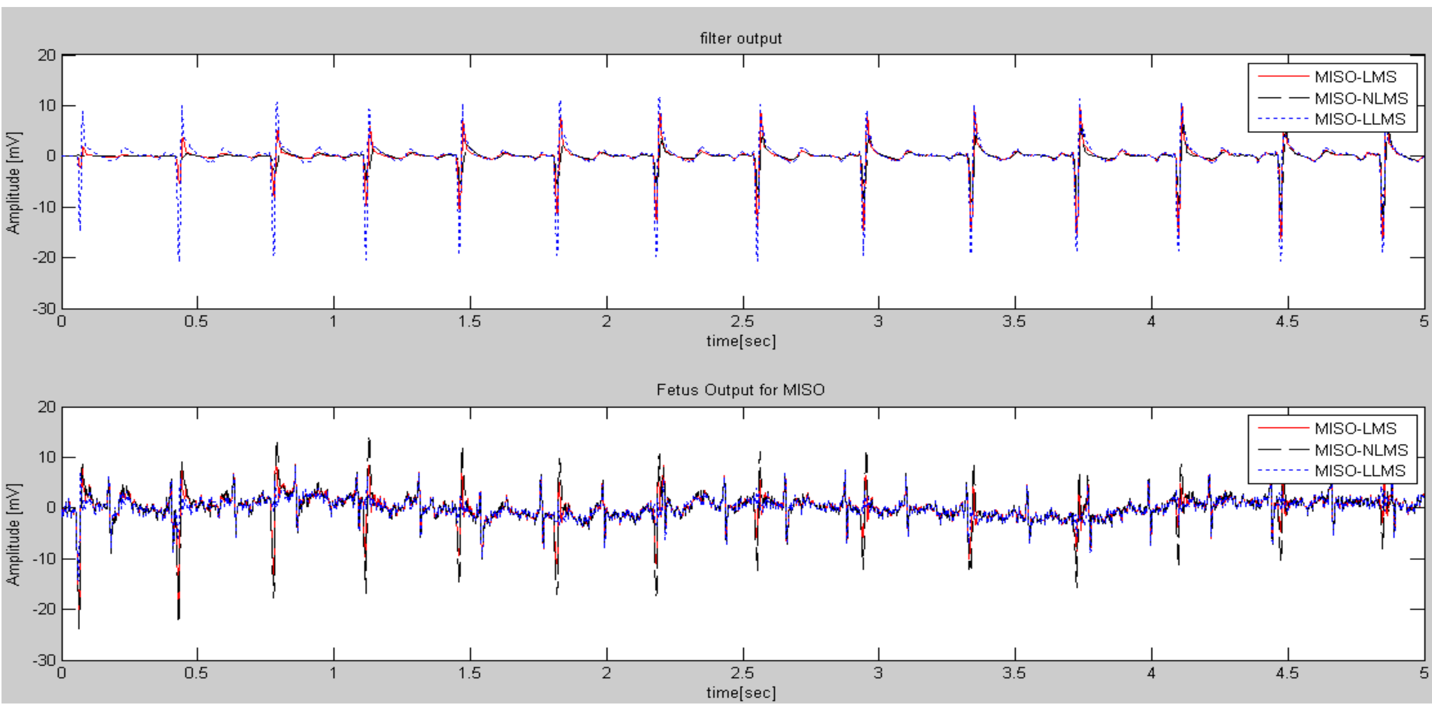

Fig 5.8: MISO Output Signals

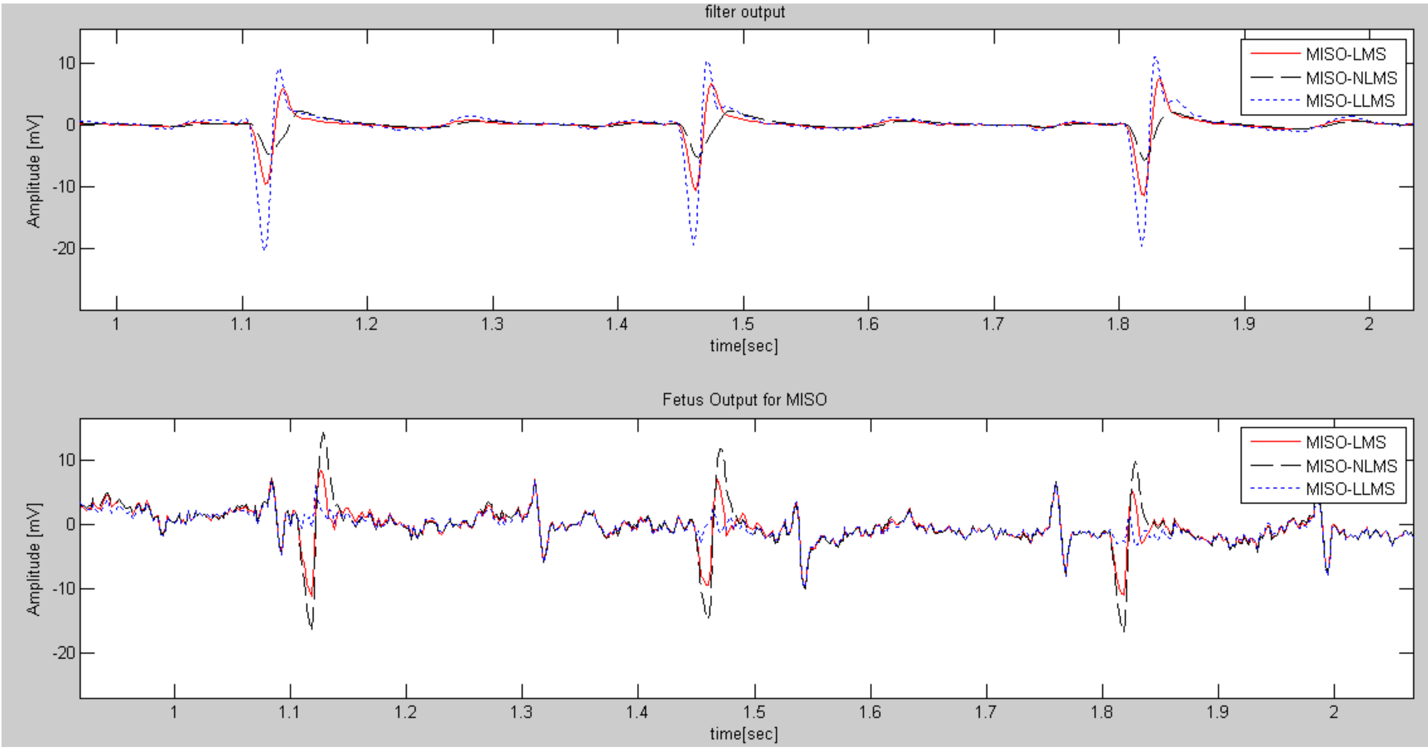

Fig 5.9: MISO Zoomed Output

- 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

Algorithms	Convergence speed
LMS	Elapsed time is 0.053666 seconds.
NLMS	Elapsed time is 0.058317 seconds
LLMS	Elapsed time is 0.049357 seconds.

- We know that, the rate of convergence can be known by calculating the time taken for an algorithm to filter the signal.
- Here in MISO system, leaky least mean square algorithm (LLMS) has more convergence speed as it takes least time to filter the signal compared to LMS and NLMS.
- The convergence rate order is LLMS>LMS>NLMS.
- From the Fig 5.9, we can observe that peaks in the fetus ECG output plot for LLMS algorithm are smaller compared to LMS and NLMS.

- Here peaks represent the mother's heartbeat. From the Fig 5.9, we can also observe that LLMS algorithm extracted the fetus heartbeat with minimal noise (peaks are minimum).
- In the fig 5.10, the fetus output at x-coordinate 0.428. We see that LLMS output has small peak. In LMS and NLMS output peaks are large due to presence of some amount of mother's heartbeat.

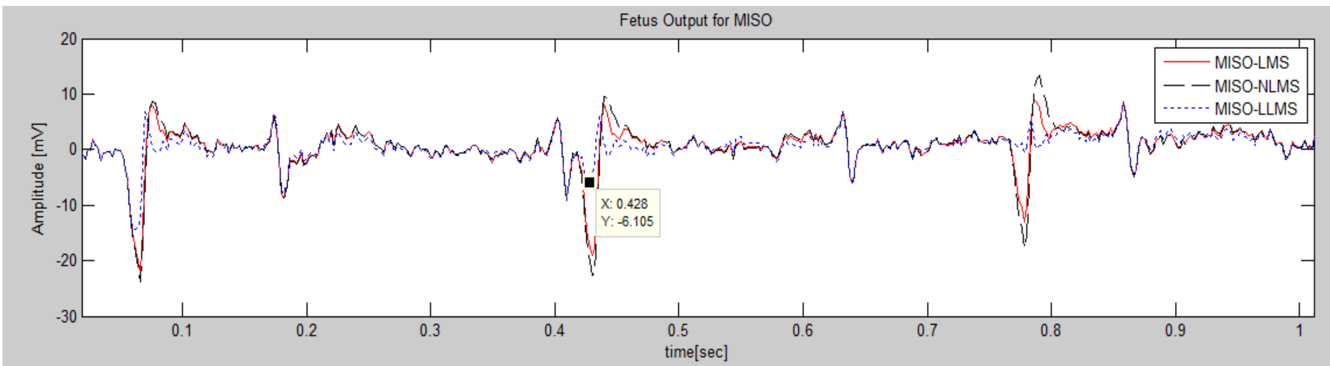

Fig 5.10: Fetus Heartbeat signal

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

CHAPTER 6

CONCLUSIONS

- Adaptive noise cancellation technique (ANC) extracted the fetus heartbeat from the combined fetus and mother’s heartbeat successfully even though signal strengths of both mother and fetus vary from each other .

- In both single input single output (SISO) and multiple input single output (MISO) systems, fetus ECG was successfully extracted using the adaptive noise cancellation (ANC) technique. The algorithms used to update the filter are least mean square (LMS), normalized least mean square (NLMS) and leaky least mean square (LLMS) algorithms.

- In single input single output (SISO) system, leaky least mean square algorithm (LLMS) had more convergence speed in extracting the fetus ECG compared to other algorithms LMS and NLMS and also leaky least mean square algorithm had better performance in extraction with minimal noise.

- In multiple input single output (MISO) system, compared to SISO more filtering was done due to the number of filters in MISO. In MISO also, LLMS had more convergence speed and better performance with minimal noise compared to LMS and NLMS algorithms.

BIBILOGRAPHY

[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.

[10] S. A. K. G. Jyoti Dhiman, "Comparison between Adaptive filter algorithms (LMS, NLMS and RLS)," vol. 2, no. 5, pp. 1100-1103, May 2013.
[11] A. S. J. P. A. P. Nimay Ch. Giri, "Capacity & performance comparison of SISO and MISO system for next generation network," vol. 3, no. 9, pp. 3031-3035, September 2014.
[12] N. Kritika sengar, "Study and capacity evaluation of SISO, MISO and MIMO RF wireless communication systems," vol. 9, pp. 436-440, Mar 2014.

[13] S. Ravindrakumar and K. B. Raja, "Fetal ECG extraction and enhancement in prenatal monitoring — Review and implementation issues," *Trendz in Information Sciences & Computing(TISC2010)*, Chennai, 2010, pp. 16-20.

[14] B. Widrow, et al., “Adaptive Noise Cancelling: Principles and Applications”, Proc. IEEE, vol. 63, pp.1692-1716, Dec. 1975.

APPENDICES

A.1 MATLAB Code:

Matlab code is written for both multiple input and single output system and single input and single output system to extract the featus heart beat from the two systems.

A.1.1 MATLAB code for SISO system

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;
```

LMS Function:

```
%LMS CODE ALGORITHM FOR THE SOURCE CODE FUNCTION
[A,E,Y] = lms(x,d,mu,nord);%lms function
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;
```

NLMS Function:

```
%NLMS ALGORITHM FOR THE SOURCE CODE FUNCTION
[A,E,Y] = nlms(x,d,beta,nord,a0)
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;
```

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 equation

end;
```

```
 end;
```

MATLAB Code for SISO Scheme:

```
% MATLAB code for SISO-ANC system
clc;
clear all;
close all;
% loading the Input data
load('foetal_ecg.dat');
x=foetal_ecg;
% time signal;
% given sampling frequency
Fs=500;
timesig=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(3,1,1);
plot(timesig,abdomin1);
title('abdomin1');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
subplot(3,1,2);
plot(timesig,abdomin2);
title('abdomin2');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
```

```
subplot(3,1,3);
plot(timesig,abdomin3);
title('abdomin3');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
figure
subplot(2,1,1);
plot(timesig,abdomin4);
title('abdomin4');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
subplot(2,1,2);
plot(timesig,abdomin5);
title('abdomin5');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
figure
subplot(3,1,1);
plot(timesig,thoirad1,'r');
title('thoirad1');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
subplot(3,1,2);
plot(timesig,thoirad2,'r');
title('thoirad2');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
subplot(3,1,3);
plot(timesig,thoirad3,'r');
title('thoirad3');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
%% LMS application for fetus Extraction
% Taking the average of Fetus+ Mothers signal
d=(abdomin1+abdomin2+abdomin3+abdomin4+abdomin5)/5;
% Taking the Average of mothers signal mostly
a=(thoirad3+thoirad1+thoirad2)/3;
% Intialising the Step size
```

```
mue= 0.00000002;
% nth order of the filter
nord=12;
X=convm(a,nord);
%Applying LMS algorithm using lms basic function.
[A1,E1,y1] = lms(X,d,mue,nord);
%% Applying NLMS algorithm using nlms basic function
% Defining the beta value
beta=0.009;
nord=12;
X=convm(a,nord);
%Applying nLMS algorithm using lms basic function.
[A2,E2,y2] = nlms(X,d,beta,nord);

%% Applying for LLMS using llms basic function
mu=0.0000002;
gammax=0.001;
nord=12;
X=convm(a,nord);
%Applying LMS algorithm using lms basic function.
[A4,E4,y4] = llms(X,d,mu,gammax,nord);
%% Fetus + mothers;
figure
subplot(2,1,1)
plot(timesig,d(1:2500),'r-');
hold on;
plot(timesig,d(1:2500),'k--');
hold on;
plot(timesig,d(1:2500),'g-.');
hold on;
title('Fetus+mothers ECG for SISO');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
legend('SISO-LMS','SISO-NLMS','SISO-LLMS');
%% Mothers signal
subplot(2,1,2)
plot(timesig,a(1:2500),'r-');
hold on;
```

```
plot(timesig,a(1:2500),'k--');
hold on;
plot(timesig,a(1:2500),'g-.');
hold on;
title('Mothers ECG for SISO');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
legend('SISO-LMS','SISO-NLMS','SISO-LLMS');
%% Filter output
figure
subplot(2,1,1)
plot(timesig,y1(1:2500),'r-');
hold on;
plot(timesig,y2(1:2500),'k--');
hold on;
plot(timesig,y4(1:2500),'g-.');
hold on;
title('filter output');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
%axis([0 1 -15 15])
legend('SISO-LMS','SISO-NLMS','SISO-LLMS')
%% plotting final fetus signalsignal.
subplot(2,1,2);
plot(timesig,E1(1:2500),'r-');
hold on;
plot(timesig,E2(1:2500),'k--');
hold on;
plot(timesig,E4(1:2500),'g-.');
hold on;
title('Fetus ECG for SISO');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
%axis([0 1 -15 15])
legend('SISO-LMS','SISO-NLMS','SISO-LLMS');
```

A.1.2 MATLAB code for MISO system

MATLAB Code for MISO Scheme

```
% MATLAB code for MISO system
clc;
clear all;
close all;
load('foetal_ecg.dat');
x=foetal_ecg;
% time signal;
timesig=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,y1] = 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,y2] = 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,y3] = llms1(X,X1,X2,d,mu,gammax,nord);
%% Plotting signals
% Fetus+mothers signal
figure
subplot(2,1,1)
plot(timesig,d(1:2500),'r-');
hold on;
plot(timesig,d(1:2500),'k--');
hold on;
plot(timesig,d(1:2500),'b:');
hold on;
title('Fetus + Mothers output');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
legend('MISO-LMS','MISO-NLMS','MISO-LLMS')
%% Mothers signal
subplot(2,1,2)
plot(timesig,a(1:2500),'r-');
hold on;
plot(timesig,a(1:2500),'k--');
hold on;
plot(timesig,a(1:2500),'b:');
```

```
hold on;
title('Mothers ECG for MISO');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
legend('MISO-LMS','MISO-NLMS','MISO-LLMS');
%% Filter output
figure
subplot(2,1,1)
plot(timesig,y1(1:2500),'r-');
hold on;
plot(timesig,y2(1:2500),'k--');
hold on;
plot(timesig,y3(1:2500),'b:');
hold on;
title('filter output');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
%axis([0 1 -10 10])
legend('MISO-LMS','MISO-NLMS','MISO-LLMS')
%% Fetus signal
subplot(2,1,2)
plot(timesig,E1(1:2500),'r-');
hold on;
plot(timesig,E2(1:2500),'k--');
hold on;
plot(timesig,E3(1:2500),'b:');
hold on;
title('Fetus Output for MISO');
xlabel('time[sec]');
ylabel('Amplitude [mV]');
%axis([0 1 -10 10])
legend('MISO-LMS','MISO-NLMS','MISO-LLMS')
```

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;
```

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;
```

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;
```

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;
```