Smart Digital Stethoscope Chip Simulation Design

By

Bayisa Taye

A Final Research Report Submitted to Adama Science and Technology University

Adama, Ethiopia
May 2023

Abstract

Auscultation is an important key for the physical (respiratory and circulatory) examination and is helpful in diagnosing various disorders. Auscultation is performed for the purposes of examining the circulatory and respiratory sounds and gastrointestinal system (bowel sounds). Besides inconsistencies in the propagation of the normal sounds, there are also several types of specific irregularities that can be heard in respiratory sounds. However, detection of abnormal sounds during auscultation needs extensive training and experience. So, the separation of these heart sound signals (HSS) and the lung sound signals (LSS) is of great research interest. Heart sounds (HS) and lung sound (LS) separation is a challenging research task for respiratory specialists and cardiologists.

Independent Component Analysis (ICA) over auscultation separation is a challenging signal processing problem. So in this study, we successfully evaluated and compared various performance parameters for heart and lung sound signal separation based on Independent Component Analysis (ICA) algorithms. The empirical results demonstrate the effectiveness of various ICA algorithms with a performance superiority over these reference techniques for various performance metrics.

Currently, the growth of micro and nano (very large-scale integration-ultra large-scale integration) electronics technology has greatly impacted biomedical signal processing devices. These high-speed micro and nano technology devices are very reliable despite their capacity to operate at tremendous speed, and can be designed to consume less power in minimum response time, which is particularly useful in biomedical products like portable digital stethoscope. Stethoscope signal interpretation is difficult due to the presence of interference generated by the heart sound. This may lead to some noise and errors during the listening of lung sounds.

The digital filters with fixed coefficients demonstrate acceptable performance to cancel the noise when the desired signal has variable stationary characteristics. However, when the desired signal has fluctuating characteristics, digital filters suffer performance degradation in eliminating the noise. To overcome the error rate at an optimum level, in this work, the authors proposed a novel non-linear artificial neural network (ANN) based adaptive line

enhancer (least mean square (LMS) and normalized least mean square (NLMS)) architecture to separate the real-time auscultation sound signals effectively.

Adaptive line enhancer (ALE) design with LMS filter and ALE design with NLMS filter are implemented in Verilog hardware description language (HDL) language to obtain both the network and adaptive algorithm in cadence Taiwan Semiconductor Manufacturing Company (TSMC) 90 nm standard cell library environment for ASIC level implementation. Native compiled simulator (NC) sim and RC lab were used for functional verification and design constraints and the physical design is implemented in Encounter to obtain the Geometric Data Stream (GDS II). In this ALE (LMS/NLMS) architecture, the area occupied is 0.08 m, the total power consumed is 5.05 mW and the computation time is 0.82 μs for ALE LMS design and the area occupied is 0.14 m, the total power consumed is 4.54 mW and the computation time is 0.03 μs for ALE-NLMS design.

Finally, for our proposed ANN ALE LMS architecture, the area occupied is 0.12m, the total power consumed is 6.65 mW and the computation time of the proposed system is 0.3 μseconds. For ANN ALE NLMS architecture, the area occupied is 0.18m, the total power consumed is 6.16 mW and the computation time of the proposed system is 0.8 μseconds that will pave a better way in future electronic stethoscope design.

The rapid technological scaling of the metal-oxide-semiconductor (MOS) devices aids in mapping multiple applications for a specific purpose on a single chip which motivates us to design a sophisticated, small and reliable application specific integrated circuit (ASIC) chip for future real time medical signal separation and processing (digital stethoscopes and digital microelectromechanical systems (MEMS) microphone).

Keywords: Auscultation, Adaptive Line Enhancer (ALE), Least mean square (LMS), Artificial Neural Networks (ANN) and Independent Component Analysis (ICA)

Acknowledgment

We would also like to express my gratitude to Mr. Eshetu Tessema, Head of the Electronics & Communication Eng. Department for providing an excellent computational work environment in the department leading to smooth completion of my research work.

We extend deepest gratitude to Mr.Tadesse Hailu, Associate Dean for Research and Technology Transfer, SOEEC, Adama Science and Technology university, Adama, for his supportive suggestions and fruitful guidelines.

We would like to express our earnest gratitude and regards to Mr. Gemechu Dengia for his perpetual motivation and guidance that has helped us to carry on with this work.

We would like to express my deep sense of gratitude, utmost indebtedness and heartfelt thanks to the all the faculty members of the department for their valuable advice and encouragement.

We would also like to thank the School of Electrical Engineering & Computing for their help and cooperation at times of need.

Our sincere appreciation goes to the administration of Adama Science & Technology University for providing me the financial assistance during this work. We would also like to thank the non-teaching staffs of my school for their cooperation.

We would also like to express my gratitude to Dr. K. Sathesh, Associate Professor, MITS for his intangible support and guidance helped me to complete this work smoothly.

We are also thankful to my family and friends who supported us to convert our dreams into reality.

Last but not the least; we would like to wind up by paying our heartfelt thanks and prayers to the Almighty, for his love and grace and taking care of us in every step of my way.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Adaptive Line Enhancer	ALE
Adaptive Noise Cancellation	ANC
Adaptive Filter	AF
Application-Specific Integrated Circuit	ASIC
Artificial Neural Network	ANN
Adaptive Linear Prediction	APP
Blind Source Separations	BSS
Convolutive Blind Source Separation	CBSS
Electronic Design Automation	EDA
Electromyogram	EMG
Extended- Infomax	Ex-Infomax
Graphic Data Stream	GDS
Generic Clock Controllers	GCLKs
Heart Sound Signals	HSS
Independent Component Analysis	ICA
Input/Output Block	IOB
Information Maximization	Infomax
Least Mean Squares	LMS
Lookup Tables	LUTs
Lung Sound Signals	LSS
Normalized Least Mean Squares	NLMS
Robust Accurate, Direct ICA	RADICAL

Signal to Noise Ratio	SNR
Signal to Interference Ratio	SIR
Very Large-Scale Integration	VLSI
Verilog Hardware Description Language	HDL
Taiwan Semiconductor Manufacturing Company	TSMC

Chapter 1

Introduction

1.1 Background

Stetho-Us will pave a pathway to the growth of portable digital electronic stethoscope device by examine yourself without leaving your home and send the auscultation results to your doctor. Stetho-Us is intended for telemedical applications, enabling integration with, among others, HIS, EDM, or telemedical systems. They rely on medical ASIC chip working together with a wireless stethoscope and dedicated application. Thanks to using unique technologies that ensure control over examination quality, portable digital stethoscopes can be used by patients at their homes. Patients will receive an accurate result of auscultation, highest quality sound, a spectrogram and an alert on appearance of abnormal lung sounds.

Auscultation [1-2] is the most important and effective clinical technique for evaluating a patient's respiratory function. Auscultation of the chest is a diagnostic method used by physicians, owing to its simplicity and non-invasiveness. Lung sound signal (LSS) measurements are taken to aid in the diagnosis of various diseases [3-5]. Their interpretation is difficult however due to the presence of interference generated by the heart

The internal parts of the body such as the heart and lungs repeat mechanical movements to produce vibration, and the vibration propagates the sound signal [6-9]. Physicians listen to the sound signals through a stethoscope to observe the function of the heart and lungs. The stethoscope relays both sound signals but the heart produces high spectrum signals whereas the lungs produce low spectrum signals during inhalation and exhalation [10-11]. We are focusing on the respiratory system to enable early detection and diagnosis of lung disease by isolating and enhancing lung sound signal transmission through a specially-designed

stethoscope.

Lung auscultation is a diagnostic method used for checking the integrity of lung function. In lung auscultation, physicians use a stethoscope to listen for changes in lung sounds to assess whether a patient has any obvious lung abnormalities. Lung sounds [12-13] can be divided into normal and abnormal sounds. Normal breath sounds can be divided into bronchial, vesicular-bronchial, vesicular, and tracheal sounds, while abnormal breath sounds can be divided into crackles, rhonchi, and wheezes. Patients with lung disease have abnormal breath sounds, so abnormal breath sounds are an important component in the diagnosis of lung diseases.

The separation of lung sound signals from the heart sound signal in real time to achieve an expected output signal is very difficult [14-15]. This is because during the signal separation, the signals are converted into the frequency domain to extract the expected signal, which increases computation complexity. In this work we design suitable ALE designs for better auscultation.

1.2 Statement of the Problem

There has been considerable increase in interest and efforts to develop new algorithms and implementing existing methods for successful heart sound signal (HSS) and LSS separation. Various bio medical signal processing techniques [16-20] have been used to filter heart sounds from lung sounds for signal inference reduction. The digital filters with fixed coefficients demonstrate acceptable performance to cancel the noise when the desired signal has variable stationary characteristics.

However, when the desired signal has fluctuating characteristics, digital filters suffer performance degradation in eliminating the noise. As a remedy to this, adaptive noise cancellation method is used. In the adaptive noise cancellation method, the coefficients of the digital filter are updated by using an adaptive algorithm, until an error is minimized. To overcome the error rate such as signal to noise ratio (SNR), mean square error (MSE) the overall computation time should be obtained at an optimum level, by adaptive line enhancer (ALE) architecture using least mean square (LMS) algorithm is used for auscultation. Researchers also used neural networks for detection of abnormal sounds during auscultation, but those networks need extensive training (update the weights updation and mapping of

inputs to outputs) and experience for separation.

The least mean square (LMS) algorithm and normalized LMS (NLMS) (Figure 1) is widely used for adaptive filtering. The input signal x(n) contains a lung, heart and noise sources, and the e(n) obtains the system output compared with adaptive filter output y(n) and desired signal d(n); and finally, the filter output y(n) is obtained from the finite impulse response (FIR) filter in each iteration with adapted weight w(n).

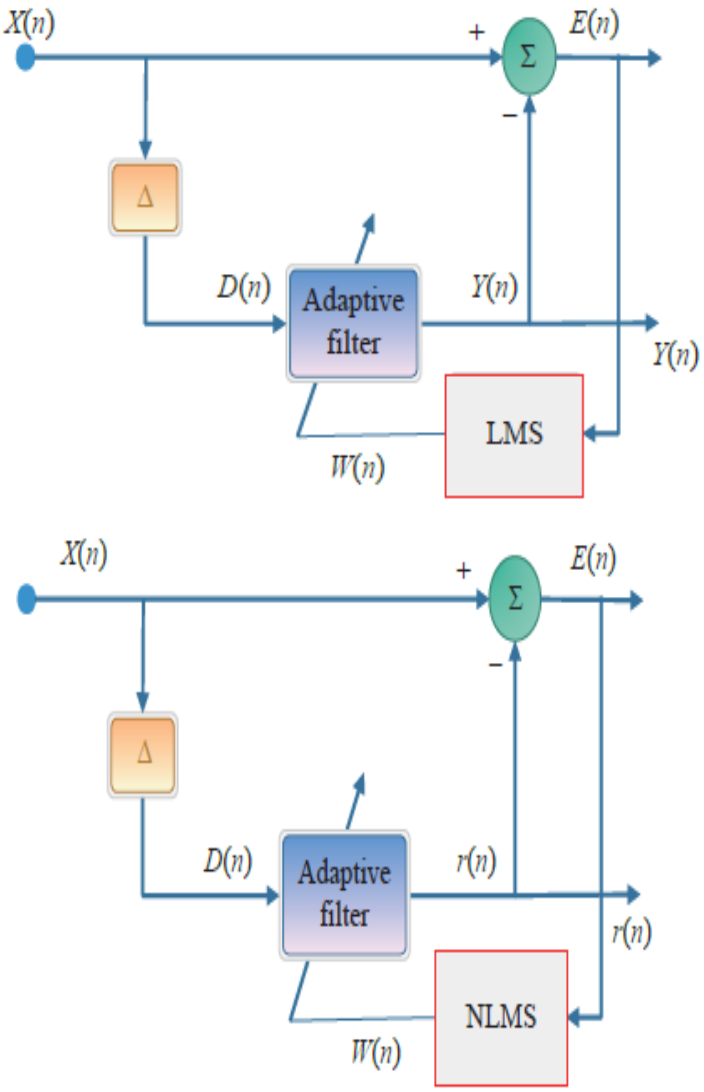


Figure. 1 Adaptive line enhancer with LMS - NLMS Adaptive algorithm [1]

To the best of our knowledge, all the previous works mainly focused on software base and none of their research implemented their design into hardware-based aspects.

But in this work, we implemented hardware realization of ALE-adaptive design successfully into an application specific integrated circuit (ASIC) chip. Hardware realization of chip is

through Verilog Hardware Description Language (HDL) coding. To verify the functionality and timing, the physical design uses an industry standard tool such as "CADENCE" in order to obtain the graphic data stream (GDSII) and to validate the architecture.

1.3 Research Contribution

Detailed version of different biomedical signal processing techniques like time frequency techniques (wavelet transform and Fourier transform-based methods), band-pass filtering, adaptive noise cancellation and adaptive filtering with adaptive algorithms (least mean squares (LMS) and normalized LMS (NLMS)) for auscultation noise removal and for effective separation of HSS from the LSS is presented.

A number of ICA approaches have been used for signal analysis, and even more ICA algorithms exist; however, the impact of using different algorithms on the results in auscultation is largely unexplored. This work compared and identified the various metrics for auscultation signal based on ICA algorithms.

This project also contributed the hardware realization of the chip (especially bio signal modeling, biomechanics and cognitive based) for ALE (LMS/NLMS design) and ANN-ALE (LMS/NLMS design). By applying this design and architecture, we can design ASIC implementation human-centered products and systems for improving safety, health, comfort and human performance.

1.4 Project Aim

Real Time sound signals separation technique is used for detecting and observing disease through sound signals in a medical apparatus.

↓

Stethoscope is an instrument that employs this technique of separation of two sound signals such as heart sound signals (HSS) and lung sound signals (LSS) by placing its flat sensor on the human body to receive and convey the sounds of internal organs to the diagnostician's ears through its earphones.

↓

Conventional Stethoscope contains a condenser microphone with preamplifier to obtain sound signals from the working human body organs.

↓

In this early model, the observed sound signal is a mix of sounds from the heart, lungs, as well as some internal and external interference or background noise as usually happens in electronic devices during the sound signal observation.

↓

Observed input signals are usually classified into heart and lung sound signals and both organs produce various sound signals. The heart sounds have high spectrum but the lung sounds contain less spectrum so the lung sound signal is separated and isolated to detect and diagnose disease in a respiratory system.

Various techniques to reduce the noise and other interference based on computerized de-noising are used to observe sound

Discrete Cosine Transforms, Fourier Transforms and Discrete Wavelet Transforms, but still the de-noising systems have not fulfilled biomedical real time signal processing needs.

Adaptive noise cancellation (ANC) with an adaptive algorithm is used to eliminate noise and other interference of the input signal from the human body through the stethoscope.

An adaptive filter is used with various adaptive algorithms such as LMS, RLS, and KF to extract and compute the sound signals but the technique processes both signals - they are not separated into either the heart or the lung signal.

This drawback can be overcome by **designing new Adaptive Line Enhancer (ALE)** with Adaptive algorithms.

- ALE architecture is similar to the ANC, the difference being that the ANC's architecture contains two input signals namely the input sound signals and reference signal, whereas in the ALE's architecture, the input signal is considered the reference signal
- Separates both sound signals and additionally reduces error with the help of adaptive FIR filters and adaptive filter algorithms (LMS, RLS, and KF)
- One major advantage of the ALE is that during the run time, the adaptive FIR filter co-efficient or weights update each iteration so the co-efficient values change in each tap from a feedback of adaptive filter tap output in each iteration, thus reducing errors and obtaining better Signal-to-Noise ratio (SNR)

1.3 Objectives:

General Objective:

- To design and develop a novel cognitive architecture for real time bio-signal separation.
- To estimate the performance accuracy of portable digital stethoscope chip design

Specific Objectives:

- To analyze biomedical signals for both normal and abnormal sounds through adaptive filters
- To propose new cognitive architectures (Artificial Neural Network- adaptive line enhancer with adaptive algorithms) for fast computation and noise cancellation
- To implement final hardware realization of the chip through Verilog HDL coding and to verify the functionality timing, and physical design using an industry-standard tool such as CADENCE in order to obtain the GDSII (Geometric Data Stream).and validate the architectures

1.4 Organization

This report is organized as follows: Sect. 2 describes various bio-medical signal processing techniques and its basic concepts are employed for auscultation noise removal and for effective separation of heart sound signal (HSS) from the lung sound signal (LSS).

Sect. 3 briefly presents various performance metrics that are evaluated in this report. Heart sounds (HS) and lung sound (LS) separation is a challenging research task for respiratory specialists and cardiologists. In this study, various performance parameters for heart and lung

sound signal separation based on Independent Component Analysis (ICA) is evaluated and compared.

Sect. 4 will focus on adaptive line enhance with adaptive algorithms (ALE LMS/NLMS) hardware realization of the chip and Sect. 5 will present our proposed ANN-ALE design followed by hardware realization of the chip simulation results. Finally, Future scope and concluding remarks are provided in Sect. 6 followed by references.

Chapter 2

Literature Review

2.1 Digital Stethoscopes and Auscultation Aspects

Respiratory illnesses are the third leading cause of death amongst the non-communicable diseases. The current COVID-19 pandemic, however, also highlights the impact of communicable respiratory syndromes. In the clinical routine, prolonged post-anesthetic respiratory instability worsens the patient outcome.

Even though early and continuous, long-term cardio-respiratory monitoring has been proposed or even proven to be beneficial in several situations, implementations thereof are sparse. Researchers [2] recently presented, multimodal patch stethoscope to estimate Einthoven electrocardiogram (ECG) Lead I and II from a single 55 mm ECG lead.

Using the stethoscope and ECG subsystems, the pre-ejection period (PEP) and left ventricular ejection time (LVET) were estimated. They conclude that the estimation of ECG, PEP, LVET, and respiratory parameters is feasible using a wearable, multimodal acquisition device and encourage further research in multimodal signal fusion for respiratory signal estimation.

The potential areas where digital medicine can be disruptive are: accessibility to quality medical care, centralization of specialties in large cities, dehumanization of medical treatment, lack of resources to access evidence-supported treatments, among others.

This review [3] presents some of the advances that are guiding the new revolution in medicine, discusses the potential and potential barriers to implementation, and suggests crucial elements for the path of incorporation of digital medicine.

The stethoscope is a powerful tool that is easy to use and allows for direct impact on patient care. With multiple sophisticated advancements made in medicine, which aid in clinical diagnosis and management, none of the modalities compare to the simplicity and vitality of a stethoscope. Thus, a digital stethoscope encapsulation was designed [4], adding new functionality to this advanced medical device.

Several encapsulation concepts were developed and prototyped using 3D printing. They were easily fitted to the digital stethoscope and tested in a hospital environment for ergonomics, acoustic and electric signals acquisition. The best concept was chosen with the help of a physician's opinion and the final prototype performance was very satisfactory.

A study [5] in which a single observer compared an analog stethoscope with the Thinklabs one electronic stethoscope in a clinical setting to determine if there was a significant difference in the diagnostic utility of the devices. Furthermore, there is also potential to drastically impact patient care by appreciating disease processes earlier and to prevent fatal outcome [6]. Medical care can also be provided in areas that are underserved or which do not have medical facilities by applying digital stethoscope technology to telemedicine to allow remote assessment.

The emergence of a digital stethoscope [7-10] has only made this historic tool even more refined. The physician can now hear heart and lung sounds with more accuracy and precision. Through this advancement, there is now potential to auscultate for obstructive coronary artery disease, and other bruits and obstructive vascular diseases such as carotid artery stenosis, and examine multiple frequencies that may comprise pulmonary auscultation. Lung and Heart sounds are two major bio sound signals.

The phenomena of pathological changes in these systems produce abnormal sounds. Physicians use a stethoscope to hear the input signal from the human body but it cannot be clearly observed for several reasons, such as varied positioning of the stethoscope condenser phone, or a gap between the body and the condenser phone, or the occurrence of some external and internal noise sources along with the input signal.

Thus, the observed sound signals which contains noise or signal interference, which masks the heart and lung sound signals leading physicians to miss abnormal sounds. Various Bio medical signal processing techniques [11] have been used to reduce signal inference. A. Mondal et. al [12] presented a method for smart auscultation by proposing a novel blind recovery of the original cardiac and respiratory sounds from a single observation mixture, in the frame work

of nonnegative matrix factorization (NMF) and through nonlinear kernel function. Researchers [13] studied the second moment behavior of the adaptive line enhancer (ALE) for a nonstationary input consisting of a fixed amplitude random phase sine wave plus a white Gaussian process with periodic power variations. We also noted a method for filtering sinusoidal noise with a variable bandwidth filter that is capable of tracking a sinusoid's drifting frequency [14]. Researchers also used Neural Networks [15] for detection of abnormal sounds during auscultation needs extensive training and experience.

2.2 Need for Adaptive Line Enhancer (ALE) Design:

In biomedical signal processing, various techniques [39-47] are employed for Auscultation noise removal and for effective separation of heart sound signal (HSS) from the lung sound signal (LSS). These signal processing techniques are Time-Frequency Techniques (wavelet transform and Fourier transform-based methods), band-pass filtering and adaptive filtering [39-40] with adaptive algorithms.

The discrete wavelet transform (DWT) computes coefficients for a dyadic scale sequence. This means that the wavelet coefficients are only calculated for scales based on the power of two. The resolution of the DWT is not as good as the resolution of the continuous wavelet transform (CWT), but its computation time is highly reduced since the coefficients are not calculated for every scale and integration is replaced by summation, which is more easily implemented.

A time-domain signal may be transformed into a frequency-domain signal by applying the Fourier transform (FT). The resulting Fourier coefficients are indicative of which frequencies are contained in a given time-domain signal. In practice, the discrete Fourier transform (DFT) is implemented to obtain the frequency-domain signal for discrete time domain signals.

In order to compute the FT of a signal in a faster and more efficient manner, it is vital importance is that the FT can only be properly applied if the signal being analyzed is assumed stationary. A stationary signal is a signal whose statistical characteristics do not change with time. Thus FT is not suitable for the analysis of biomedical signals. To determine how the frequency content of a signal changes over time by dividing the signal into blocks and the spectrum of each block is computed. In an effort to avert the disadvantage of the FT, the short time Fourier transform (STFT) was developed.

The STFT is implemented by cutting the signal of interest into smaller blocks, where each block is assumed stationary and the FT is performed on each one of them. In order to improve the results, blocks overlap each other and each block is multiplied by a window function that is tapered at its endpoints (this is called windowing) to mitigate spectral smearing. The spectrum is thus determined by computing spectra of overlapping signal blocks. The main deficiency of STFT is that the length of the window is fixed and, thus, is not an effective way to describe structures much smaller than the window length.

Although the wavelet transform overcomes this limitation by allowing for a variable window length, there is a fundamental reciprocal relation that exists between the central frequency of a wavelet and its window length. Thus, the wavelet transform (WT) does not provide precise estimates of low frequency components with short duration or narrow-band high frequency components. In view of the above disadvantages time-frequency techniques (WT, FT, STFT) fails to favor.

There are applications where a particular band of frequencies needs to be filtered from a wider range of mixed signals. The band-pass filter is a suitable candidate for achieving this task. Methods based on linear band-pass fixed filtering are not suitable for separation of the heart sound signal (HSS) from the lung sound signal (LSS) because of spectral overlap of these two signals paves the way for adaptive noise cancelling techniques. Given the time-variance or non-stationarity of the signals in question, time-domain adaptive noise cancelling techniques have been implemented.

2.3. Basic Adaptive Filter Model

By using Adaptive Filters [42][44], the optimization problem on which all adaptive signal processing functions are designed can be resolved. The adaptive filter consists of two parts namely adaptive FIR filter and Adaptive algorithm (LMS, NLMS, RLS & KF) as shown in Figure 2. The filter is used to compute and calculate the filter output signal *y(n)*.

The set of FIR filter coefficients are continuously updating or adjusted by the adaptive algorithm. The adaptive algorithm is responsible for adjusting the filter coefficients or weights so that the filter output *y(n)* becomes as very close as possible to a desired signal *d(n)*. In many cases, the adaptive algorithm adjusts the filter coefficients a little bit to

minimize a certain level of the error signal *e(n)* from Equation (1), defined as the variation among the desired signal *d(n)* and the filter output *y(n)*,

$$e(n) = d(n) - y(n) \tag{1}$$

The adaptive filter function is used to minimize the error value as well as improve the overall system performance.

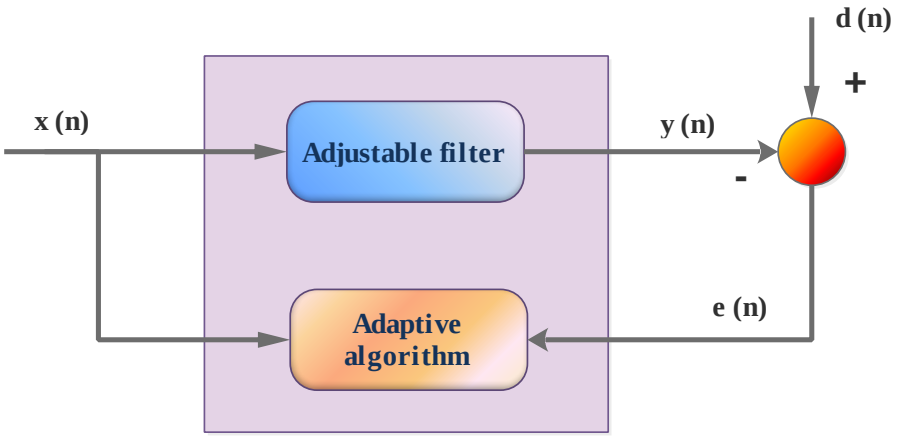


Figure 2. Block diagram of the general Adaptive Filtering

2.4 Adaptive Noise Cancellation

Noise Cancellation [44] 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. Figure 3. shows the architecture of the adaptive noise canceller. Equation (2) states that the output of the adaptive filter *y(n)* in ANC is described as,

$$y(n) = x(n) + (r(n)) - w(n)) \tag{2}$$

The adaptive filter coefficient updates each iteration from the feedback of the output signal *y(n)* with reference signal *r(s)* as shown in the Equation (3).

$$w(n) = y(n) + r(n) \tag{3}$$

The mean square estimation of the input signal in adaptive filter is described as in the Equation (4).

$$e[y(n)^2] = e[x(n)^2] + e[r(n)^2 - w(n)^2] \tag{4}$$

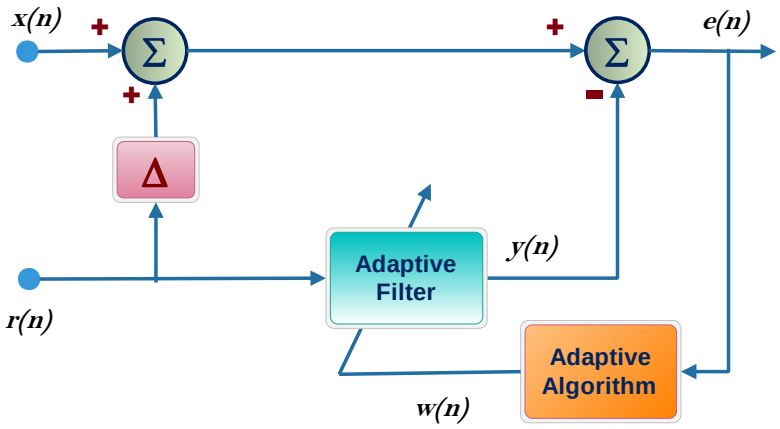


Figure 3. Adaptive Noise Cancellation

2.5. Need for Adaptive Filter

Least Mean Squares (LMS)[41-45] filtering has been used extensively in many different applications, such as control, communications, medical, and geophysical signal processing. Recursive or adaptive filtering refers to a particular design procedure, where we learn more about the unknown filter's model, each time new experimental evidence becomes available.

Adaptive filters [41] are used in applications where the system and/or time are varying. The development of efficient algorithms for adaptive filtering has been the focus of intensive research for the last three decades. The efficiency of an algorithm is measured against a number of performance indices such as convergence rate, tracking rate, computational complexity, round-off error robustness, and numerical stability. In a parallel processor environment, the issue of computational parallelism and pipeline ability become major design factors. The adjustable filter in the above system is called the forward predictor, which might have any basic filter structure and it's shown in Figure 4. Equation (5) states that the output of the adaptive filter $y(n)$ as,

$$y(n) = \sum_{i=1}^{M} a_i x(n-i) \tag{5}$$

The error signal $e_f(n) = x(n) - y(n)$ is referred to as the M^{th} order forward prediction error. Minimizing $|e_f(n)|^2$ results in a conventional Wiener filtering problem with a solution for the

optimal forward predictor coefficients as given by Equation (6), defining the autocorrelation function of the input at delay k as $r(k) = e[x(n)\, x(n-k)]$.

$$\underline{a} = \Re \underline{r} \tag{6}$$

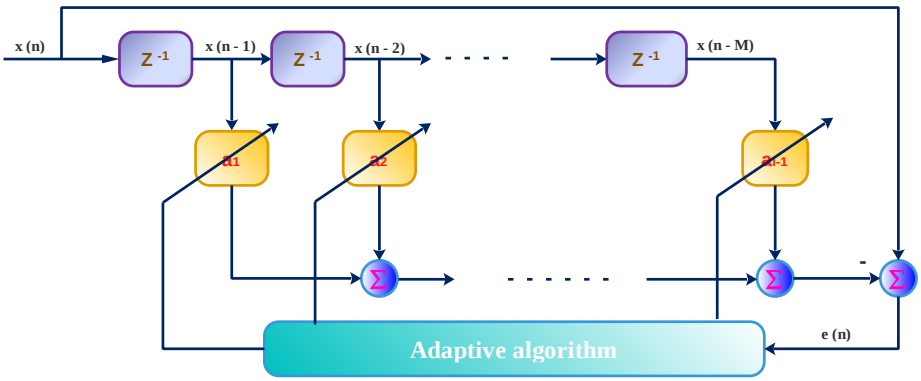


Figure 4. Block diagram of the Transversal Forward Prediction

Similarly, the backward predictor of order M has its desired signal $d(n) = x(n-M)$ and its input x(n) as shown in Figure 5. The backward predictor estimates $x(n-M)$ as a linear combination of $x(n), x(n-1), \ldots. x(n-M+1)$ by minimizing the error signal $e_b(n) = x(n-M) - y(n)$ in some sense. The filter output in this case is referred to as the M^{th} order backward prediction of the input $x(n)$ and is given in Equation (7).

$$y(n) = \sum_{i=1}^{M} b_i x(n-i+1) \tag{7}$$

The error signal $e_b(n)$ is referred to as the M^{th} order backward prediction error. Minimizing $|e_b(n)|^2$ results in a conventional Wiener filtering problem with a solution for the optimal backward predictor coefficients b given in Equation (8).

$$\underline{b} = \Re \underline{r}_{\underline{b}} \tag{8}$$

This indicates that the optimal backward predictor coefficients are the same as the optimal forward predictor coefficients of the same order but arranged in reverse order such that in Equation (9).

$$b_i = a_{M+1-i,} \quad for\ i = 1, 2, \ldots, M \tag{9}$$

The backward prediction-error filter is the system with input *e(n)* and output *e(n)*. For the backward predictor with optimal coefficients it also holds that the input sequence *x(n)* and the backward prediction error $e_b(n)$ are uncorrelated. Moreover, the j^{th} order backward prediction error $e_{bj}(n)$ for j=0,1,2,…..Mare uncorrelated with one another.

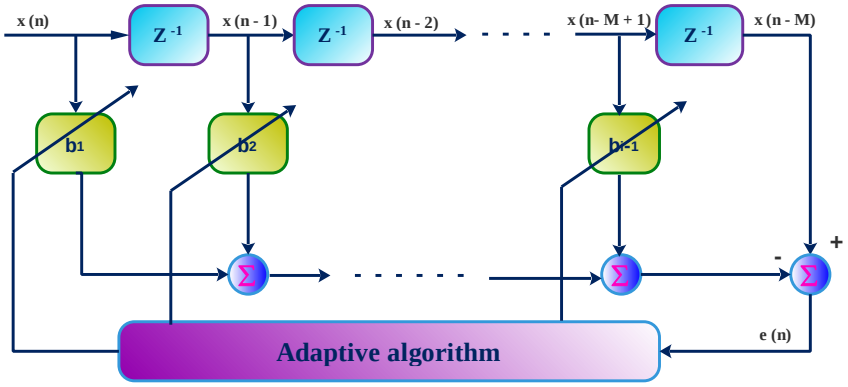


Figure 5. Block diagram of the Transversal Backward Prediction

2.6 Least Mean Square (LMS) Filters

LMS filters [41-45] are a class of adaptive filters that are able to "learn" an unknown transfer function. LMS filters use a gradient descent method in which the filter coefficients are updated based on the instantaneous error signal. Adaptive filters are often used in communication systems, equalizers, and noise removal.

A digital signal processing library includes LMS filter functions that drive on Q15, Q31, and floating-point data types. The library also contains normalized LMS filters in which the filter coefficient adaptation is independent of the level of the input signal.

2.7. Normalized LMS Filters

Similarly, the normalized-LMS algorithm [42-47] is the same as an LMS algorithm but the only factor included is normalization to improve the adaptation rate of the filter, while the other remaining functions are the same as an adaptive filter. A digital signal processing library includes LMS filter functions that drive on Q15, Q31, and floating-point data types.

A normalized least mean square (NLMS) filter consists of two major parts as shown in Figure 6. The first component is a standard transversal or FIR filter. The second component is a

coefficient update. The NLMS filter has two input signals. The "input" feeds the FIR filter while the "reference input" corresponds to the desired output of the FIR filter. That is, the FIR filter coefficients are updated so that the output of the FIR filter matches the reference input. The filter coefficient update mechanism is based on the difference between the FIR filter output and the reference input. This "error signal" tends towards zero as the filter adapts. The NLMS processing functions accept the input and reference input signals and generate the filter output and error signal.

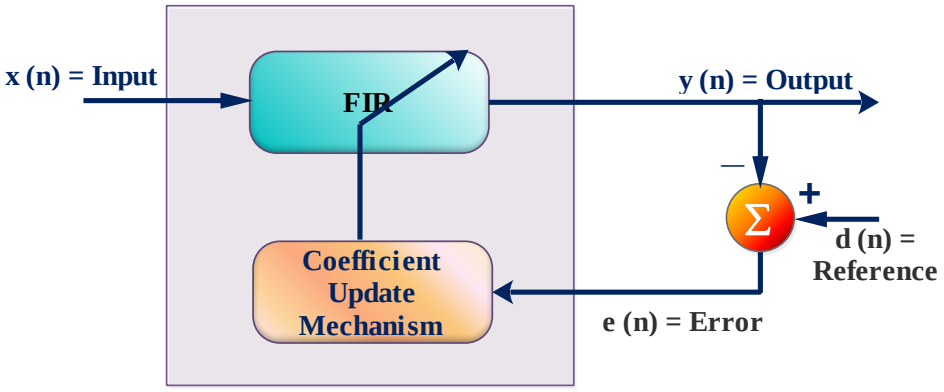


Figure 6. Internal structure of the NLMS Adaptive Filter

The functions operate on blocks of data and each call to the function processes block Size samples through the filter. *x(n)* points to input signal, *d(n)* points to reference signal, *y(n)* points to output signal and *e(n)* points to error signal. All arrays contain block Size (sample) values.

The functions operate on a block-by-block basis. Internally, the filter coefficients *w(n)* are updated on a sample-by-sample basis. The convergence of the LMS filter is slower compared to the normalized LMS algorithm.

The output signal *y(n)* is computed by a standard FIR filter as in Equation (10).

$$y(n) = w(0) * x(n) + w(1) * x(n-1) + w(2) * x(n-2) + \ldots + w(\mathrm{N}-1) * x(n-\mathrm{N}+1) \quad (10)$$

The error signal equals the difference between the reference signal *d[n]* and the filter output as in Equation (11).

$$e[n] = d[n] - y[n] \quad (11)$$

After each sample of the error signal is computed the instantaneous energy of the filter state variables is calculated as in Equation (12).

$$e = x(n)^2 + x(n-1)^2 + ... + x(n-N+1)^2 \quad (12)$$

The filter coefficients *w[k]* are then updated on a sample-by-sample basis as in Equation (13).

$$w(k+1) = w(k) + e(n) * (mu/e) * x(n-k), \quad for \quad k = 0,1,....., N-1 \quad (13)$$

where mu is the step size and controls the rate of coefficient convergence. In the adaptive signal processing each filter tap coefficients are stored in time reversed order as in Equation (14).

$$\{w(N-1), w(N-2), ..., w(1), w(0)\} \quad (14)$$

During the run time the adaptive filter consists of a state array of size numTaps (*N*) + block Size *(x(n))* - *1* as in Equation (15). Samples in the state buffer are stored in the order:

$$\{x(n-N+1), x(n-N), x(n-N-1), x(n-N-2), ...x(0), x(1), ...x(n-1)\} \quad (15)$$

Note that the length of the state exceeds the length of the coefficient array by blockSize-1 samples. The increased state length allows order, which is traditionally used in FIR filters, to be avoided and yields a significant speed improvement. The state variables are updated after each block of data is processed.

2.8 Adaptive Line Enhancer (ALE)

ALE [39-47] is similar to the ANC architecture (Figure 7) but ANC uses two input signals whereas the ALE uses the input signal as the reference signal so the output signal is nearly error-free. In this architecture, the input signal is taken to be the reference signal to obtain the minimum error rate of the output signal. The adaptive algorithms provide feedback to the system to update the FIR filter coefficient.

The adaptive algorithms are classified into Least Mean Square (LMS), Normalised-LMS, Recursive Least Square (RLS) and Kalman Filter (KF). The LMS algorithms are specific to different applications.

For noise cancellation these adaptive algorithms which update the adaptive FIR filter coefficients which processes the signals in an iterative manner to ensure minimum error and the output of the adaptive filter is separated from the reference signals. The output can be analyzed by varying the adaptive filter order, for instance, N = 2, 4, 8, 16 and 32 in order to minimize errors, Signal-to-Noise Ratio (SNR) and Mean Square Error (MSE) in the shortest response time.

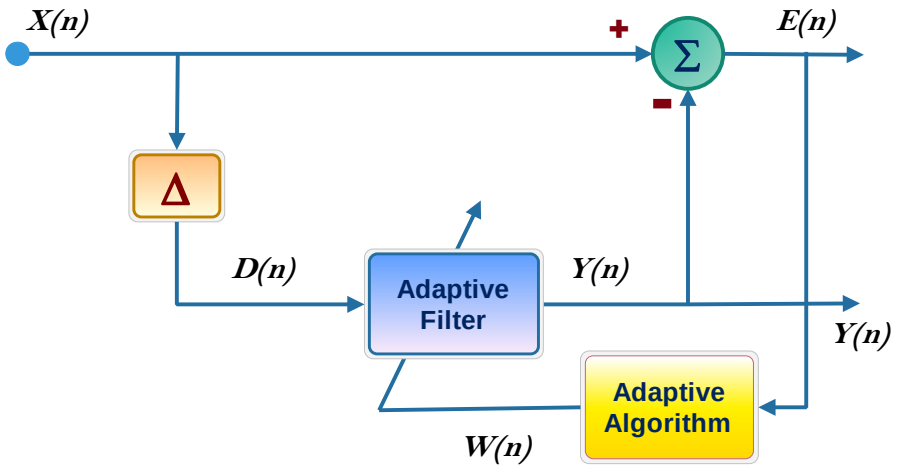


Figure 7. Adaptive Line Enhancer (ALE)

2.9 Chapter Summary

An adaptive implementation of a linear prediction filter which was updated using the LMS algorithm will provide a real time capability for many situations. This configuration was chosen as the Adaptive Line Enhancer (ALE), due to the initial application of separating sinusoids from broadband noise. It is generally applicable, however, whenever there is a significant difference between the correlation times of the signal and the noise.

Consequently, the ALE and related adaptive prediction filter implementations have been investigated for a wide range of applications including instantaneous frequency estimation, spectral analysis, and narrowband interference rejection, predictive de-convolution, intrusion detection and adaptive carrier recovery for digital data communications receivers.

In this Chapter various bio signal processing techniques are discussed. Superiority of adaptive line enhancer (ALE) over adaptive noise cancellation are explored. Adaptive filters (least mean square (LMS) Filters and normalized LMS) designs are discussed.

Chapter 3

Auscultation Performance Metrics Computation

3.1 Introduction

Auscultation of the chest is a diagnostic method used by physicians, owing to its simplicity and noninvasiveness. Various algorithms and techniques have been presented in the literature during last few decades and analyzed different aspects. They include high pass filter, adaptive filtering algorithms, wavelet based denoising algorithms, time frequency filtering, modulation filtering and independent component analysis.

Blind source separations (BSS) [16] is widely used to biomedical signal processing, audio and array signal processing and digital communication. They used a variety of independent component analysis algorithm (ICA) algorithms [17-36] (Fastica, Infomax, ExInfomax, JADE and Kernel) for many biomedical applications. Infomax Algorithm is widely used algorithm for blind source separation in EEG and fMRI data analysis.

ICA have been used [17] to extract common hemodynamic sources for a group of functional magnetic resonance images (fMRI). A new second-order Hessian-free algorithm for Infomax is introduced by [18] which achieve asymptotically quadratic convergence.

Efficient chip design for convolutive blind source separation (CBSS) adopted using the information maximization (Infomax) [19] method consists mainly of Infomax filtering modules and scaling factor computation modules. Infomax Algorithm is also implemented in Visible Light Communication (VLC) signal separation in [20] based on the artificial neural network for the analysis of covariance of the values from signals.

They are also implemented for the extraction of class-discriminant information in remote sensing hyper spectral image classification [21] and magneto encephalography (MEG)-based real-time brain computing interfaces (BCI) [22]. Various researches have been done using Extended- Infomax algorithm. They were applied to character recognition of Brain-Computer Interface (BCI) system [23] [24] based on P300 (a kind of evoked potentials).

This algorithm also used in various research areas, analysis of fMRI data. The Infomax algorithm and its extended version that adapts to sub and super-Gaussian distributions havebeen widely used in various research areas, including analysis of fMRI data.

A constrained version of the extended Infomax algorithm is used as an example to show the benefits obtained from the non- orthogonal constrained framework.

Kernel Algorithm is widely used algorithm for blind source separation in electromyogram (EMG) signals [25] for diagnosing neuromuscular disorders. Blind source separation (BSS) using kernel independent component-based analysis have been studied [27] [30] in-depth.

Fast ICA Algorithm [26] is also implemented to calculate the utility harmonic impedance [28], porosity defect detection [29] and smart antenna systems [31].

Kernel Algorithm is widely used algorithm for blind source separation for nonlinear and non-Gaussian process monitoring [32] [34], fault monitoring [33], fault detection, nonlinear feature extraction and data driven fault diagnosis [35] [36]. Kernel is also implemented in performance monitoring the high order non- Gaussian characteristics in chemical process [37].

RADICAL (Robust, Accurate, Direct freelance part Analysis Algorithm estimates the independent sources exploitation differential entropy estimator supported m-spacing estimator. Joint Approximation Diagonalization of Eigen matrices (JADE) algorithm exploits the fourth order moments so as to separate the source signals from mixed signals.

In this work, we make the following unique contributions:

- While there is an increasing demand for blind source separation techniques, to the best of our knowledge, there is no comparative study published in Auscultation field to calculate their performance metrics.
- **This is the first initiative attempt to calculate their metrics based on various ICA algorithms in auscultation separation signals.**
- Various important performance metrics evaluated in this work are:
    - ✓ Absolute Error Rate (AER),
    - ✓ Correlation Coefficient (r),
    - ✓ Mean Square Error (MSE),
    - ✓ Root Mean Square Error(RMSE),
    - ✓ Normalized Mean Square Error (NMSE),

- ✓ Peak Signal to Noise Ratio (PSNR),
- ✓ Signal to Noise Ratio (SNR),
- ✓ Improved Signal to Noise Ratio (ISNR),
- ✓ Signal to Interference Ratio (SIR),
- ✓ Amari-error,
- ✓ Frobenius error
- ✓ Maximum Signal to Noise Ratio (SNR-MAX)

- This work evaluated ICA algorithms based on various important performance metrics which paves the way for better algorithm comparison.

Here, neither the original sources nor the mixture matrix is known. This can be the Blind Separation of Sources (BSS) where the aim is to get a non-observable set of signals, the sources, from another set of observable signals are considered as mixtures.

3.2 Independent Component Analysis (ICA)

BSS problem is simply tackled by exploiting the upper higher signal statistics and improvement techniques. The original source vector **S** is of size M x N and also the mixing matrix **A** is of size **M x M**, where, **M** is that the variety of statistical independent sources and **N** is that the variety of samples in every source.

The results of the separation method are that the demixing matrix **W** which might be used to obtain and acquire the estimated statistical independent sources from the mixtures.

ICA method is one of the BSS techniques to separate the input signals from mixing signals. Schematic illustration of the ICA mathematical model in shown in Figure 8.

The original supply vector **S** is of size M x N and also the mixing matrix **A** is of size **M x M**, where, **M** is that the variety of statistical independent sources and **N** is that the variety of samples in every source.

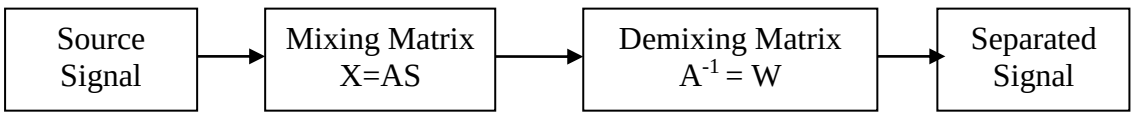


Figure 8. Schematic Illustration of the mathematical model used to perform ICA decomposition [38]

The results of the separation method are that the demixing matrix **W** which might be used to obtain and acquire the estimated statistical independent sources, Ŝ from the mixtures. This method is described by Equation 1 and a schematic illustration of the mathematical model [38] in shown in Figure 8.

$$X = AS \rightarrow \hat{S} = WX \tag{16}$$

3.3 Preprocessing for ICA:

Some preprocessing is beneficial before making an attempt to estimate **W**.

(i) The determined signals should be focused by subtracting their mean E{**x**}

$$\tilde{X} = X = E[X] \tag{17}$$

(ii) Then they are whitened, which implies they are linearly remodeled so the components are uncorrelated and has unit variance.

(iii) Whitening is performed via eigenvalue decomposition of the variance matrix, $\mathbf{V\Lambda V^T}$, **V** is that the matrix of orthogonal eigenvectors and **Λ** may be a square matrix with the corresponding eigenvalues. The whitening is done by multiplication with the transformation matrix **P**

3.4 Metrics Formulas:

- ***Mean Square Error (MSE)***

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(S_{est}(i) - S_{actual}(\mathrm{i}))^2 \tag{18}$$

Sest - Estimated signal
Sactual - Actual signal
N – Length of the signal

- ***Signal to Noise Ratio (SNR)***

$$SNR_{dB} = 10\log_{10}\left[\frac{\sum_{k=0}^{N-1} S(\mathrm{k})^2}{\sum_{k=0}^{N-1}\left[S(k) - \hat{Y}(k)\right]^2}\right] \tag{19}$$

S(k) – Input Signal
$\hat{Y}(k)$ - Estimated output signal

N – Number of samples

- ***Absolute Error Rate (AER)***

$$AER = \frac{|S_i - S_{\text{Ref}}|}{S_{\text{Ref}}} \tag{20}$$

$|.|$ - Absolute value
Si - Output signal
SRef - Reference Signal

- ***Mean Square Error (MSE)***

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(S_{est}(i) - S_{actual}(\text{i}))^2 \tag{21}$$

Sest - Estimated signal
Sactual - Actual signal
N – Length of the signal

- ***Root Mean Square Error (RMSE)***

$$RMSE = \frac{1}{N}\sum_{K=1}^{N}\left[s(\text{k}) - y(k) \right]^2 \tag{22}$$

s(k) – Actual signal

$y(k)$ - output signal
N – Number of samples

- ***Normalized Mean Square Error (NMSE)***

$$NMSE = \frac{\sum_{i=1}^{N}(S_{est}(i) - S_{actual}(\text{i}))^2}{\sum_{i=1}^{N}(S_{noisy}(i) - S_{actual}(\text{i}))^2} \tag{23}$$

S_{est} - Estimated signal
S_{actual} - Actual signal
S_{noisy} - Noisy signal
N – Number of samples

- ***Peak Signal to Noise Ratio (PSNR)***

$$PSNR = 20\log_{10}\left(\frac{64}{RMSE}\right) \tag{24}$$

- ***Signal to Noise Ratio (SNR)***

$$SNR_{dB} = 10\log_{10}\left[\frac{\sum_{k=0}^{N-1} S(\mathrm{k})^2}{\sum_{k=0}^{N-1}\left[S(k) - \hat{Y}(k)\right]^2}\right] \tag{25}$$

S(k) – Input Signal

$\hat{Y}(k)$ - Estimated output signal

N – Number of samples

- ***Improved Signal to Noise Ratio (ISNR)***

$$ISNR_{dB} = 10\log_{10}\left(\frac{\left(\mathrm{s}(k) - x(k)\right)^2}{\left(s(k) - y(k)\right)^2}\right) \tag{26}$$

s(k) – Input Signal

x(k) – Mixed Signal

y(k) – Output Signal

- ***Signal to Interference Ratio (SIR)***

$$SIR_{dB} = 10\log\left(\frac{\sum_{t=1}^{T}\left\|s_t\right\|^2}{\sum_{t=1}^{T}\left\|y_t - s_t\right\|^2}\right) \tag{27}$$

s – Source signals s= {s1, s2...sT}

y – Demixed signals y= (y1, y2…. yT}

- ***Amari-error***

$$d(U,V) = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{\sum_{j=1}^{m}\left|B_{ij}\right|}{\max_j\left|B_{ij}\right|} + \sum_{j=1}^{m}\frac{\sum_{i=1}^{m}\left|B_{ij}\right|}{\max_i\left|B_{ij}\right|}\right) - 2 \tag{28}$$

U, V – Matrices

B = UV-1(It is necessary to normalize each row and of U and V)

Amari error lies on [0 to (m-1)]

- ***Frobenius error***

$$d_F(\hat{W}, W_p) = \left\| \hat{W} W_p^{-1} - I_{mxm} \right\| F \tag{29}$$

3.5 Metrics Computation:

For blind source separation of signals, we have calculated various important performance parameters and shown in Table 1. The horizontal axis represents the samples and the vertical axis represents amplitude in all the graphs. Here sampling frequency 44100 Hz and time be 2.5 sec is used. The reference lung and heart sound signals, mixed (noise) signals and algorithm outputs are shown in the Figure 9-15.

Table 1: Performance Metrics Evaluation

MSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.0386	0.6316	1.667	0.6011	0.6013	0.6011
Heart	1.0079	0.0299	0.0395	0.7982	1.2243	1.2242

RMSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.0191	0.7947	1.2911	0.7753	0.7753	0.7753
Heart	1.0039	0.1728	0.1988	0.8934	1.1065	1.1065

NMSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	24.3943	34.6069	34.6318	0.000101	0.0151	0.0043
Heart	0.8609	1.4856	1.3545	1.39E-05	1.1917	1.2038

SNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	12.8735	13.4886	18.9033	0.0265	12.8736	12.8735
Heart	24.3404	-2.4221	2.6362	-0.0264	24.3414	24.3404

SIR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	0.7677	0.7362	0.7333	0.6997	0.7039	0.6835
Heart	0.7849	-0.3422	-0.0855	1.3561	0.6137	0.4696

ISNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	-24.7953	16.7682	20.9836	2.9277	22.4222	22.4199
Heart	-17.6389	1.7841	3.0011	0.7523	18.4842	18.4841

PSNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	35.9589	38.1194	38.9041	38.3341	38.3329	38.3344
Heart	36.0894	51.3716	50.1547	37.1025	35.2446	35.2442

SNR MAX

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.6205	0.9028	0.9021	0.8881	0.8884	0.8881
Heart	0.8883	1.1389	0.9167	1.6219	1.6181	1.6217

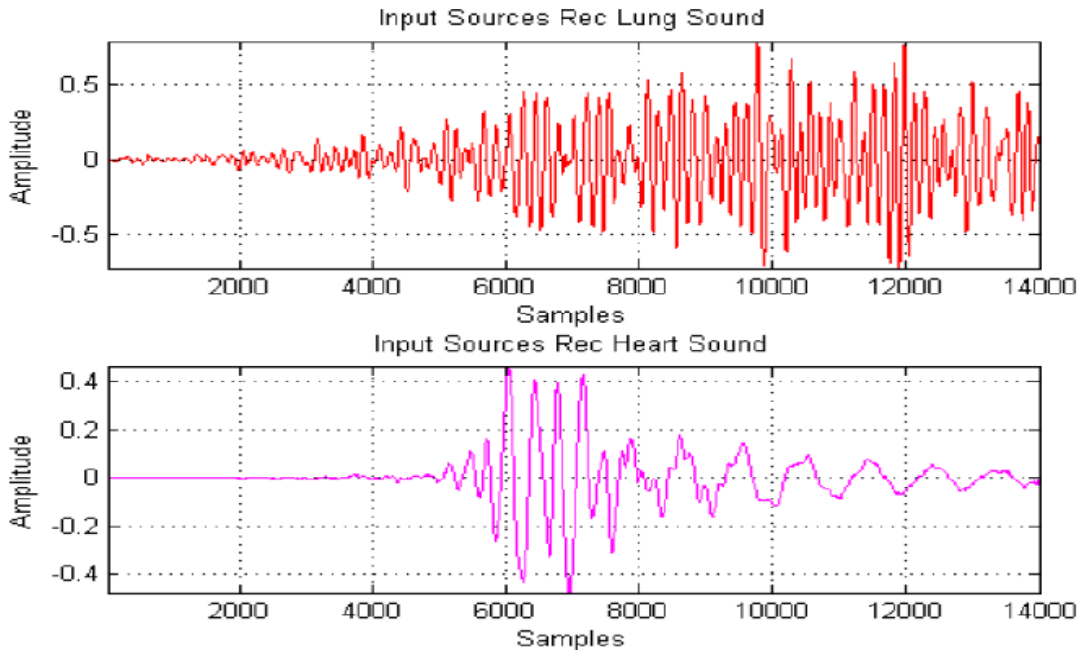
AMARI ERROR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung & Heart	-0.1831	-2.4597	-1.5749	0.012	12.9476	12.211

FORBENIUS ERROR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung & Heart	0.111	1.0064	-1.5749	0.012	12.9476	12.211

3.6 Simulation Result:



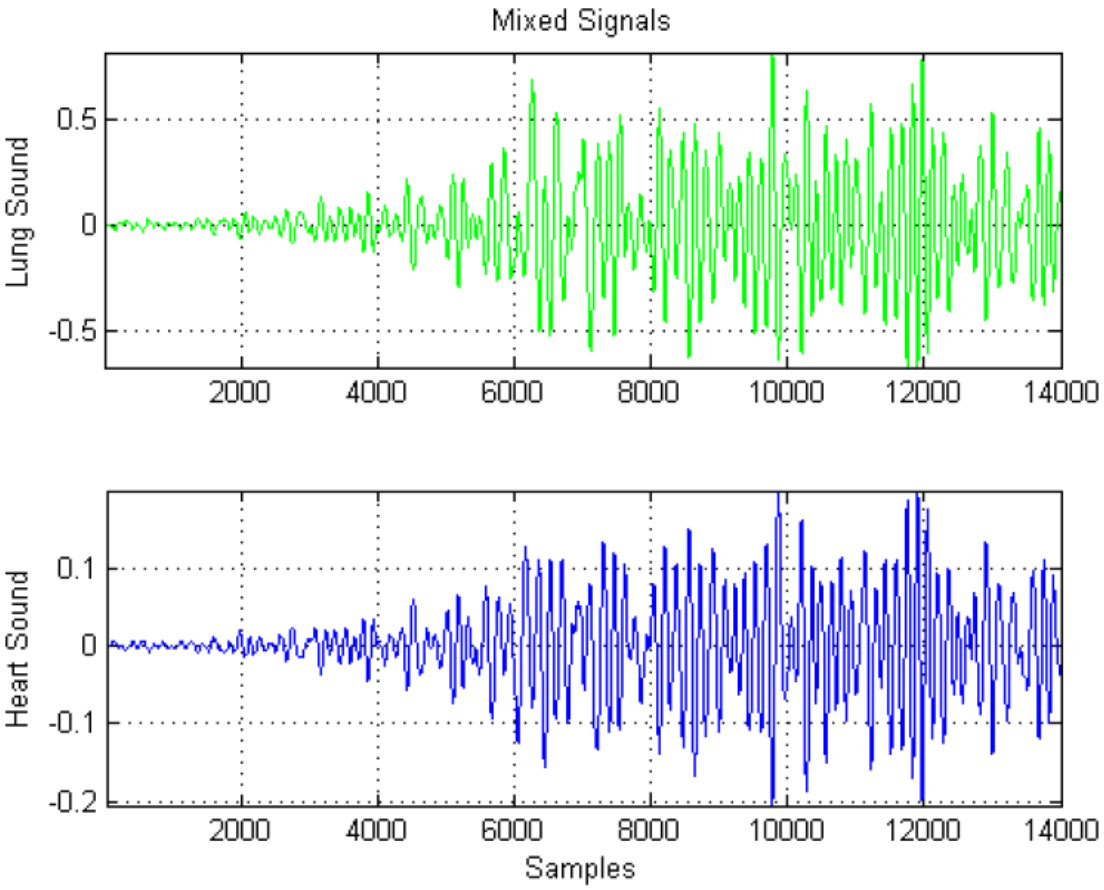


Fig 9: Input Heart and Lung Signals

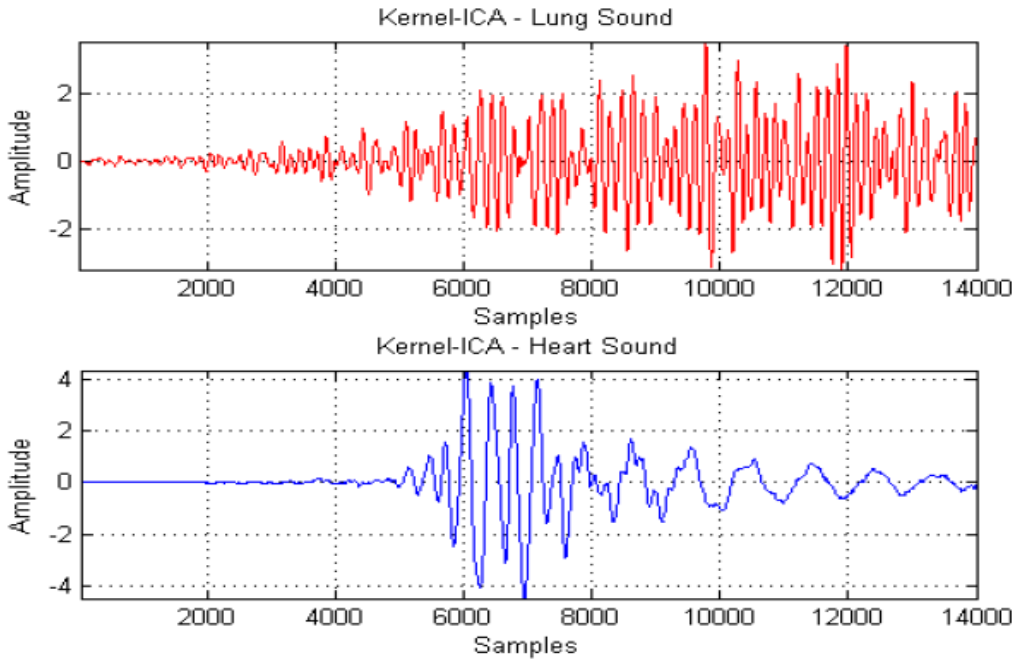


Fig 10: Simulation Output – KERNEL ICA

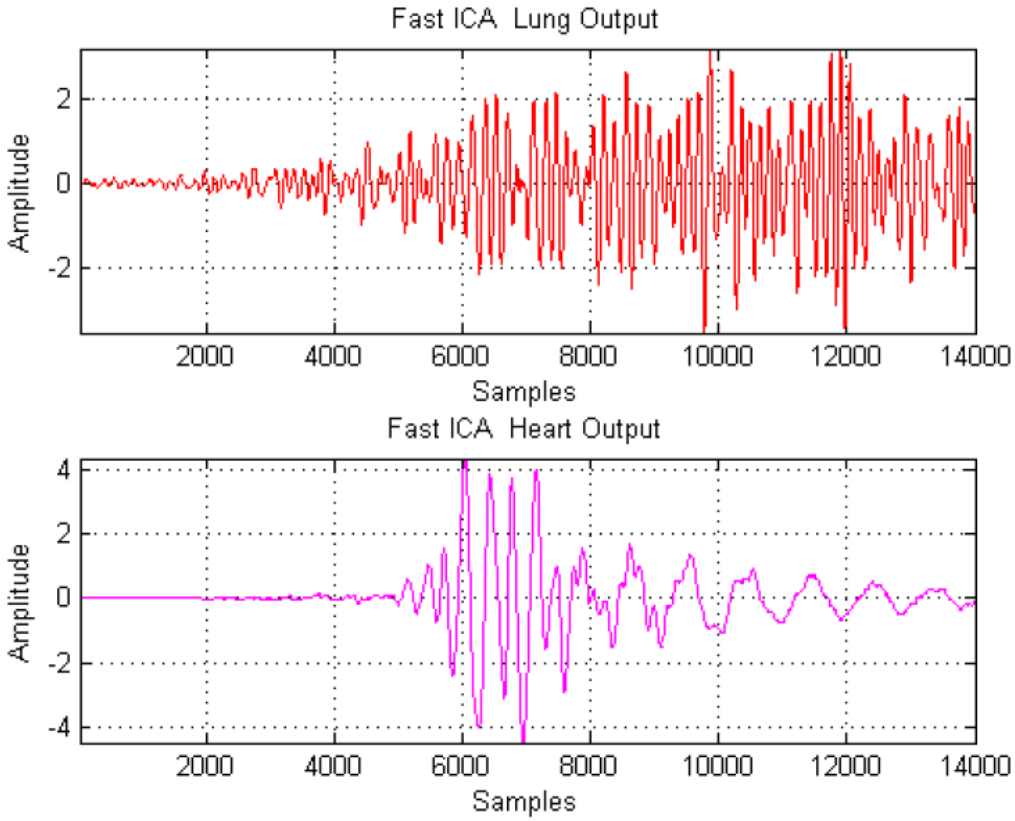


Fig 11: Simulation Output – FastICA

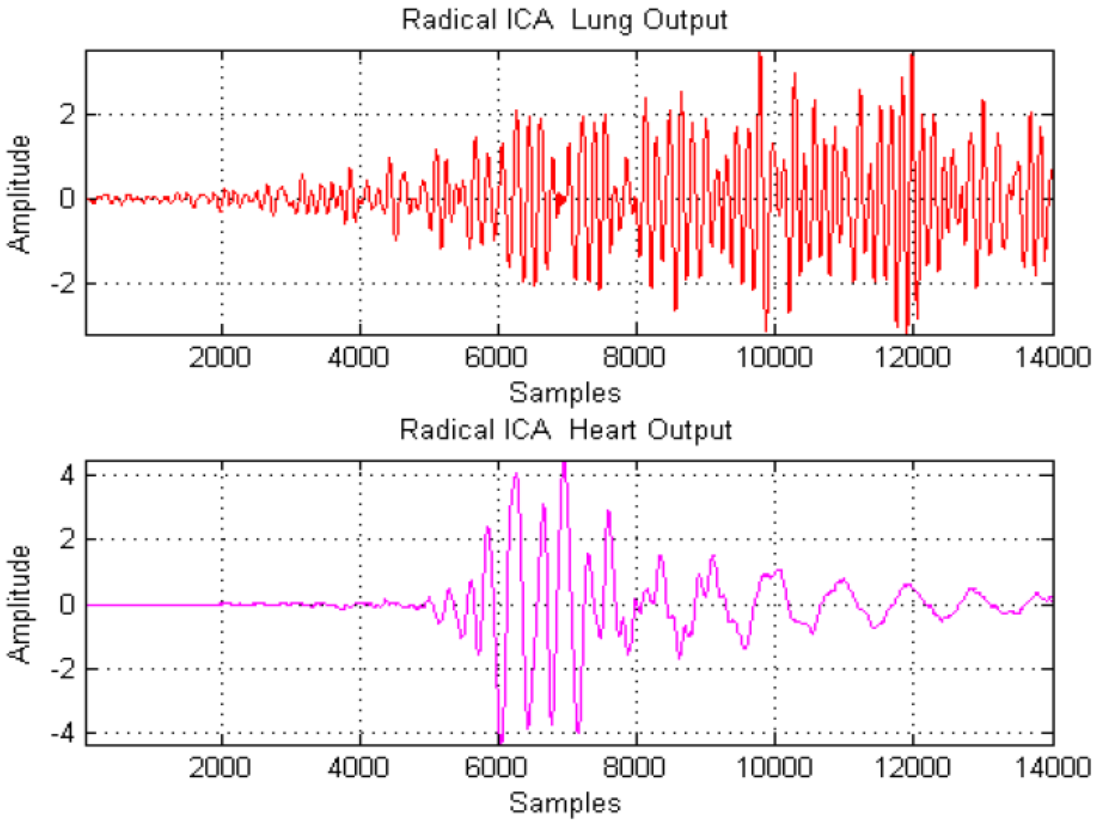


Fig 12: Simulation Output – Radical ICA

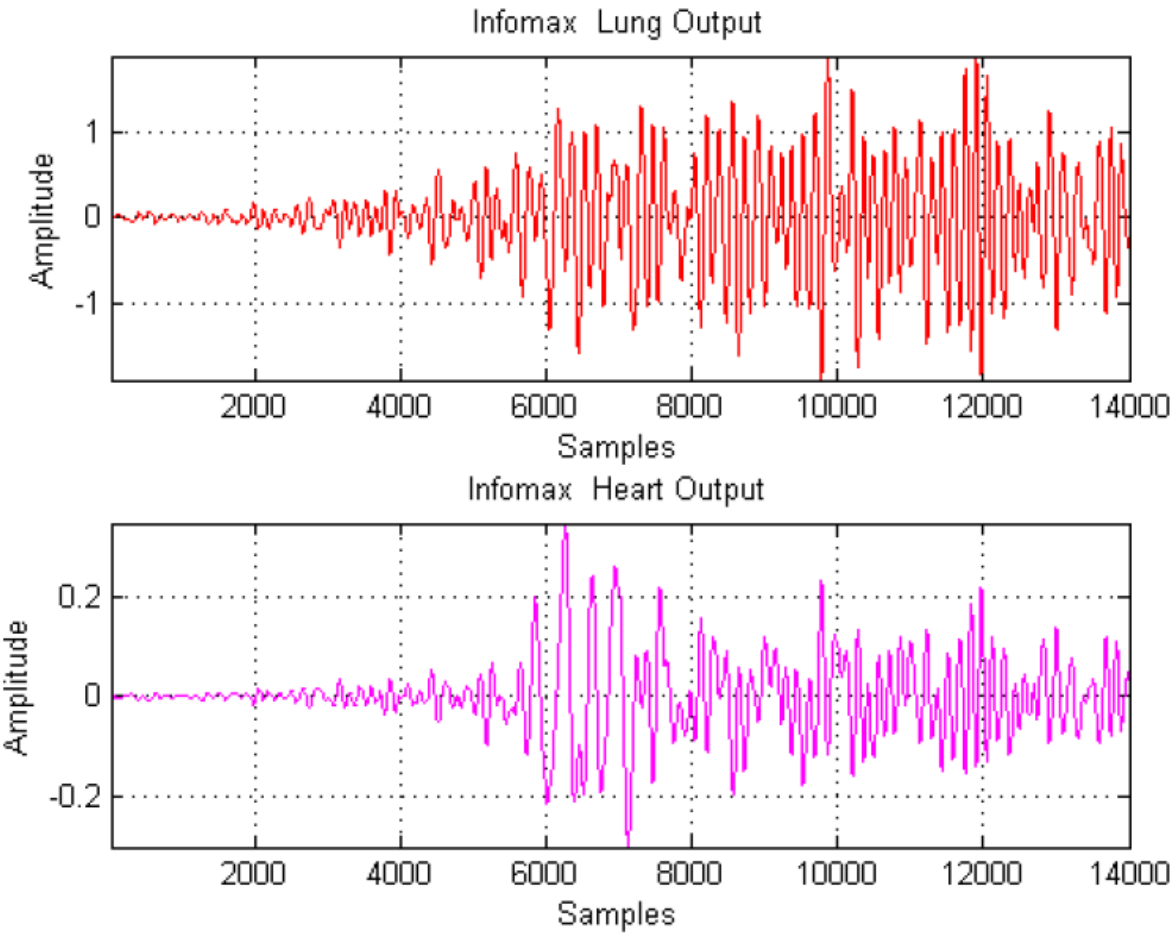


Fig 13: Simulation Output – Infomax ICA

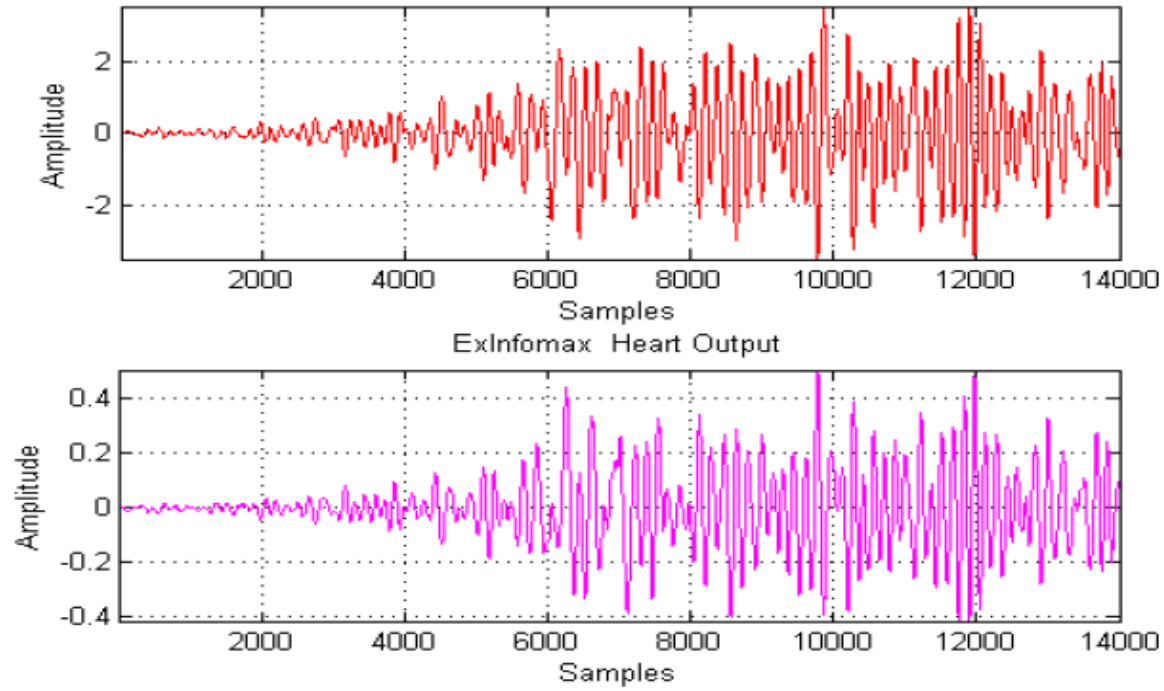


Fig 14: Simulation Output – ExInfomax ICA

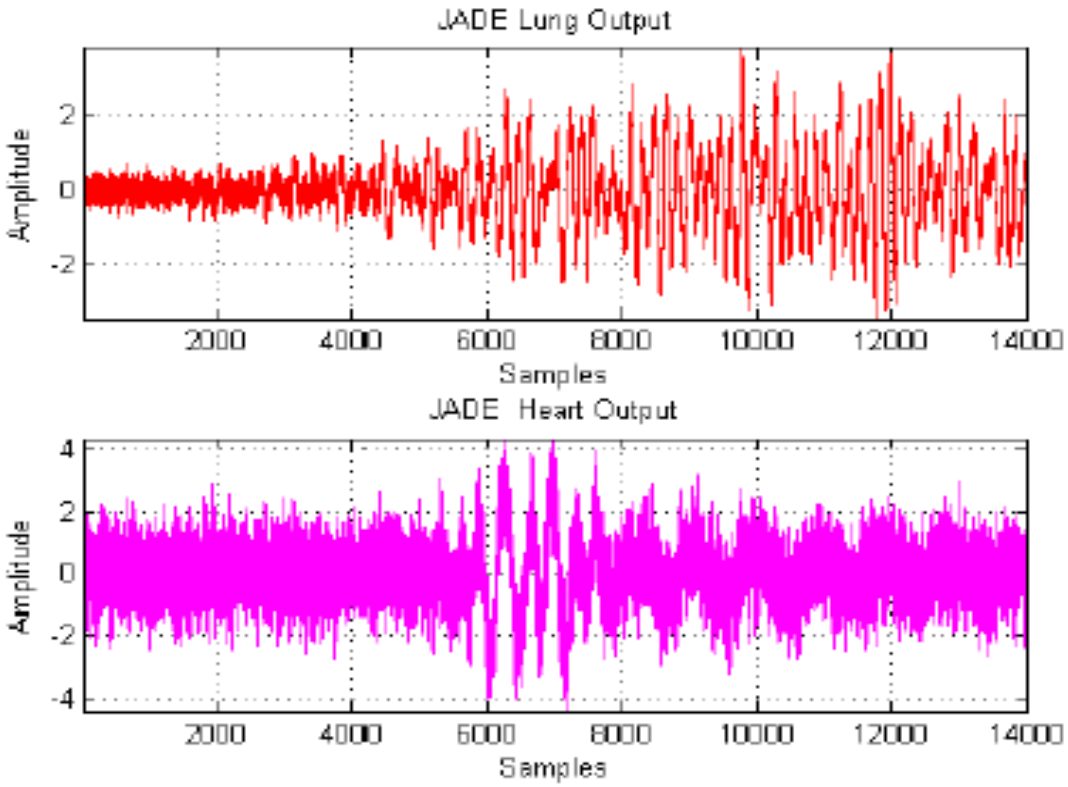


Fig 15: Simulation Output – JADE ICA

3.7 Discussion

Different metrics were involved in this study, as presented in Table 1, which poses a challenge for determining the most successful algorithm for heart-sound separation technique. Depending on our metrics requirement, we can choose any ICA algorithm to meet our research outputs.

It is worth noting that, in order to use the algorithm, a mixed signal is used (Original signal + Mixed Matrix). The output recovered has better signal separation (heart and lung sound signals) besides noisy mixed signals.

From the Table 1 it is clearly pictured that

- Kernal algorithm perform best for both Frobenius Error and Amari error whereas Ex-Infomax algorithm for SNR metrics.
- For SIR and SNR MAX metrics, Fastica for lung sound whereas Kernal shown better values in heart sound separation. For SNR metrics, Ex-Infomax for lung

sounds whereas JADE and Radical algorithms shown higher impact in heart sounds separation For NMSE Kernal algorithm performs better separation for both lung and heart sounds.

Infomax and Ex-Infomax algorithm:

- Infomax algorithm favors higher than Ex-Infomax in case of MSE, RMSE, NMSE, PSNR, SNR MAX, SIR and Frobenius Error. But Ex-Infomax algorithm dominates in ISNR and SNR than Infomax algorithm.

- Ex-Infomax algorithm is an extended version of Infomax algorithm whose different mixing matrix and mathematical scaling tends to show their improvements in only two metrics (ISNR and SNR).

- Whereas the algorithm shown lesser efficient in other metrics evaluation when compared to Infomax algorithm.

- This descript that an extended version is efficient for some metrics only and original Infomax algorithm shown better results for many metrics which proves their efficiency.

Kernal and Fastica algorithm:

- From the Table.1 it is clearly pictured that Kernel algorithm favors higher than Fastica in case of MSE, RMSE, NMSE, PSNR, ISNR, Frobenius Error and Amari Error. But Fastica algorithm dominates only in SNR than Kernel for both lung and heart sound.

- For SNR MAX and SIR, Fastica algorithm for lung sound separation and Kernel-ICA algorithm for heart sound separation performs in an effective manner.

- **Amari error lies in the Range of [0,1].** So Kernal value (0.012) performs best than other algorithmsAdaptability of algorithms will be varied according to our real time applications and metrics we have chosen to evaluate and estimate.

JADE and Radical algorithm:

- JADE algorithm performs better for Frobenius Error, whereas RADICAL algorithm for Amari error.
- From the Table.1 it is clearly pictured that both the ICA algorithm algorithms dominate in RMSE, PSNR, ISNR and SNR for lung sound and heart sound separation.
- For MSE and NMSE metrics, Radical for lung sound and JADE for heart sound shown equal separation values.
- For SIR metrics, both algorithms perform equal separation for both lung and heart sounds.
- For SNR MAX metrics, Radical for heart sound separation is higher, whereas both algorithms shown same value for lung sound separation.

3.8 Chapter Summary

We can apply these algorithms to clinical application in terms of separation efficiencies. In this work, a detailed comparison among various widely used ICA algorithm for blind source separation (BSS) was presented.

A number of BSS algorithm approaches have been used for signal analysis, and even more algorithms exist; however, the impact of using different algorithms on the results in auscultation is largely unexplored. This analysis will be used to compare and identify the best strategy for extracting auscultation signal based on the use of ICA algorithms.

While our results are not indicative of finding an optimal solution to the problem, we do feel that we have made progress. This study opens several lines for future work. Analyzing the existing tradeoffs and evaluation of other metrics for other algorithms are some of the future works of this research. This work can be extended by following the same fashion for other signal analysis and may vary with different engineering applications.

Chapter 4

Hardware realization of ALE LMS/NLMS Design

4.1 Xilinx Implementation

In this chapter, ALE with LMS/NLMS architecture is developed in Verilog HDL. The real time input signal is sampled and quantized in 32-bit IEEE standard data format (Institute of Electrical and Electronics Engineers (IEEE) 754 single precession) and processed through the adaptive filter respectively.

The suitable adaptive filter order (N = 8 & 16) is optimized and analyzed in Matlab environment. The electronic design automation (EDA) tool is employed in various levels of implementation such as simulation, synthesis, physical design, and verification using CADENCE RC Lab and ENCOUNTER to implement Taiwan Semiconductor Manufacturing Company (TSMC) 90 nm technology standard library for verification of various levels of VLSI design steps and ASIC level design steps to obtain a minimum error rate, size, and delay, while ensuring that the entire architecture can consume the least power.

The output results are discussed briefly up to GDS II. The hardware realization of the proposed architecture ALE-LMS is shown in Fig. 16 and the device utilization of the architecture obtained in Xilinx is shown in Tab. 2.

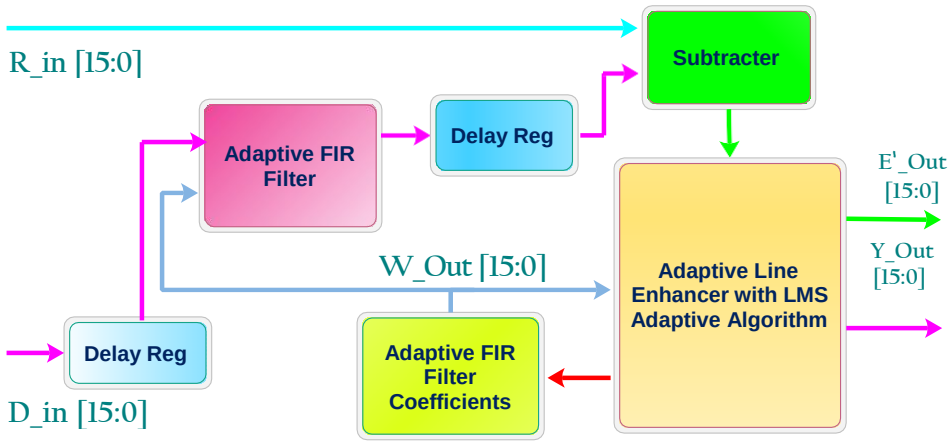


Fig 16: Architecture of ALE with LMS algorithm

To optimize the time and power after the design is implemented, the HDL coding in register transfer level (RTL) and the RTL view of the proposed architecture is achieved from cadence

environment in RC lab. The power and timing constraints used to optimize the minimum and maximum level of the device to be utilized depending on the standard cell library.

The routing is used to route the power and clock buses in the entire design. The lambda rule is used to check the entire design to complete the above steps and finally the design is obtained after the geometry verification. If any violation occurs in the design during the verification, it cannot obtain GDS II.

A cadence encounter tool is used to implement the proposed architecture by following the above steps to obtain the final design. Finally, the design is converted to a graphical database structure (GDS) II format and shown in Fig. 17 and Fig. 18 for LMS 8 and LMS 16. The proposed system is simulated and synthesized to obtain major design parameters considered in VLSI technology such as area, gates, power and timing analysis obtained in cadence RC synthesizer.

Table 2: Device utilization of ALE - LMS architecture in Xilinx environment

No	Parameter	LMS_8_Tap	LMS_16_Tap
1	**No of Slices**	1434 out of 2448 **58%**	3111 out of 2448 **127%**
2	**No of Slice FFs**	291 out of 4896 **5%**	629 out of 4896 **13%**
3	**No of LUTs**	2793 out of 4896 **57%**	5833 out of 4896 **119%**
4	**No of IOBs**	67 out of 158 **42%**	67 out of 158 **42%**
5	**No of GCLKs**	1 out of 24 **4%**	1 out of 24 **4%**
6	**Completion time**	23.34 secs	73.81 secs
7	**Memory usage**	162684 kilobytes (159)	241788 kilobytes (236)

4.2 LMS Cadence Simulation

The parameter values of area and gates are shown in Tab. 3. The utilization of power is represented in Tab. 4 and the total time to complete the process is shown in Tab. 5.

The performance comparison of the ALE architecture is analyzed for LMS algorithm with two different adaptive filter orders 8 & 16. The result was obtained by the utilization of VLSI design major parameters like area, gates, power and computation time for the architecture, which are represented in Tab. 6.

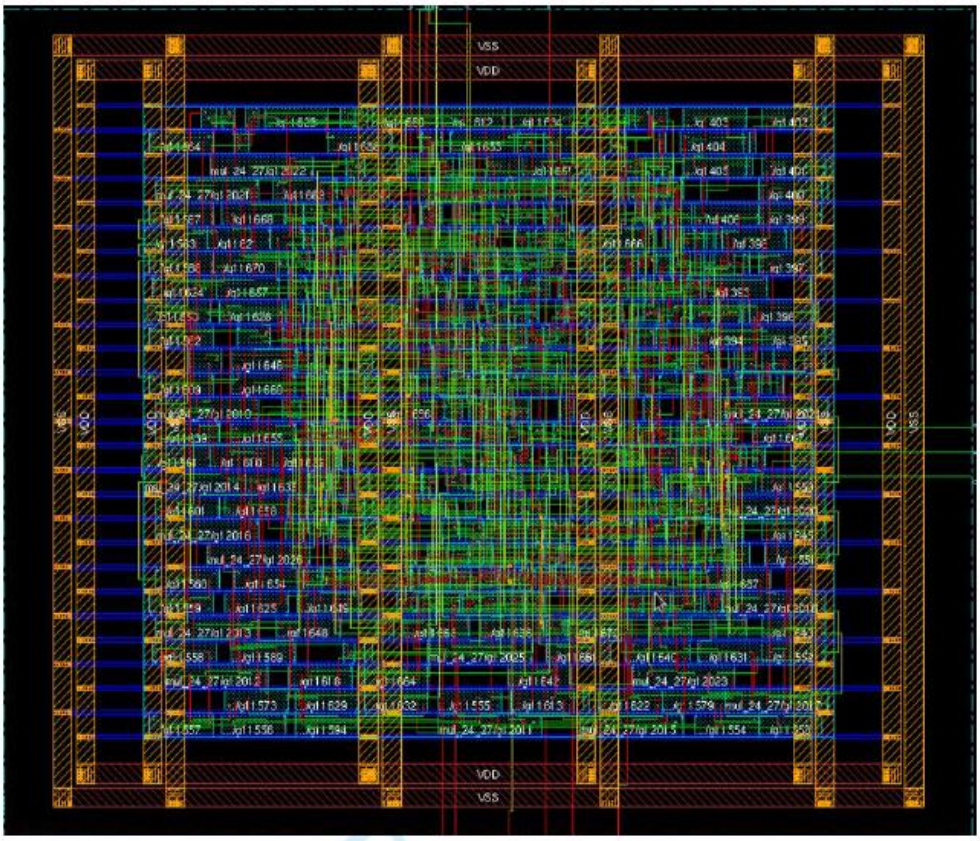


Fig 17: Physical design in cadence encounter (LMS 8)

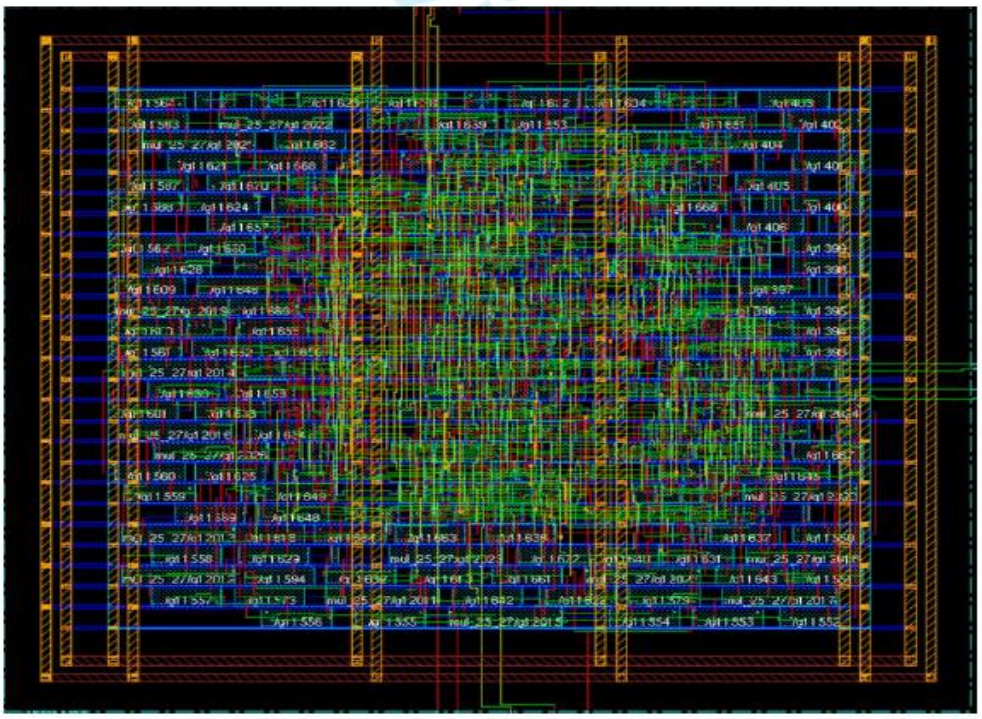


Fig 18: Physical design in cadence encounter (LMS 16)

4.3 Metrics Utilization of ALE-LMS Architecture

Table 3: Area and Gate Utilization of ALE-LMS Architecture

Parameter	LMS 8 (µm)		LMS 16 (µm)	
	Gates	Area	Gates	Area
Sequential	298	7193.58	544	13587.9
Inverter	762	1730.27	1541	3508.23
Logic	8235	74698.5	15949	145799

Table 4: Power Utilization of ALE-LMS Architecture

Filter order	Leakage power(nW)	Dynamic power(nW)	Total power(nW)
LMS 8	351361	4698831	5050192
LMS 16	685299	9549936	10235234.828

Table 5: Timing Utilization of ALE-LMS Architecture

Filter order	Fall time(Ps)	Rise time(Ps)	Total time(Ps)
LMS 8	583512	239840	823352
LMS 16	691790	261851	953641

Table 6: Performance analysis of ALE architecture (Area, power, delay and gates in LMS filter order N = 8 & 16)

Parameter	LMS 8	LMS 16
Area (µm)	83622.3	162895
Gates	9285	18034
Power (nW)	5050192	10235234.828
Time (Ps)	823352	953641

From the table 3-6, we can deduce that gates and area of LMS order 16 consumes more area/gates/power utilization than LMS order 8.

4.4 NLMS Cadence Simulation

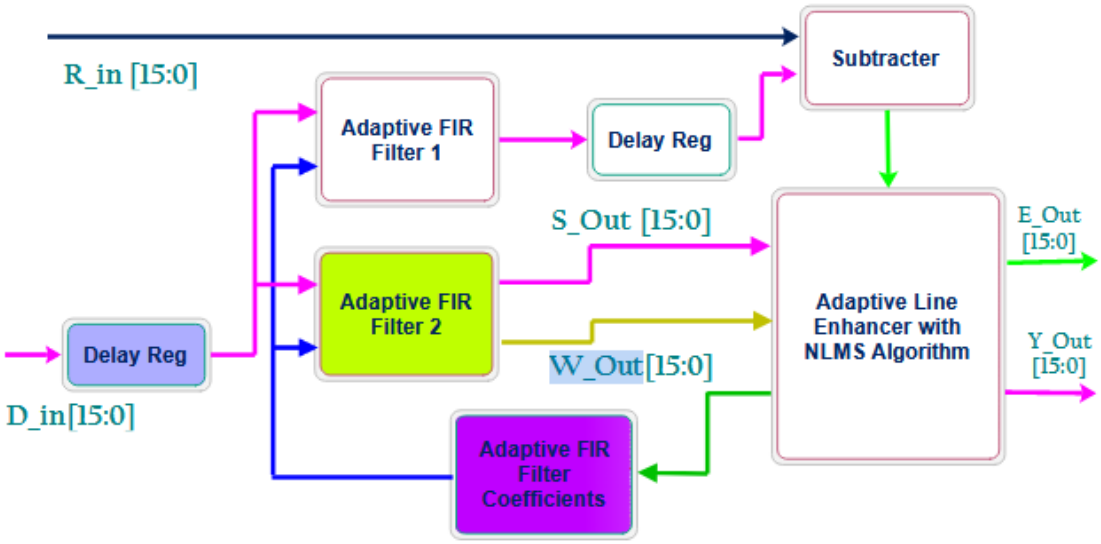


Fig. 19 Architecture of ALE with NLMS algorithm

ALE with NLMS design is shown in Fig. 19. To implement hardware realization of chip through Verilog HDL coding and to verify the functionality and timing, the physical design uses an industry standard tool such as "CADENCE" in order to obtain the GDSII and to validate the architecture. Final design is shown in Fig. 20

Table 7: Device utilization of ALE-NLMS architecture in Xilinx environment

No.	Parameter	LMS_8_Tap	LMS_16_Tap
1	No. of slices	2257 out of 2448 92%	5101 out of 2448 208%
2	No. of slice flip-flops（FFs）	573 out of 4896 11%	1052 out of 4896 21%
3	No. of lookup tables（LUTs）	4119 out of 4896 84%	9347 out of 4896 190%
4	No. of input/output block（IOB）	67 out of 158 42%	67 out of 158 42%
5	No. of generic clock controllers（GCLKs）	2 out of 24 4%	2 out of 24 4%
6	Completion time	44.08 s	121.25 s
7	Memory usage	221372 KB	331900 KB

The hardware realization of the proposed architecture ALE-NLMS is shown in Fig. 19 and the device utilization of the architecture obtained in Xilinx is shown in Tab. 7.

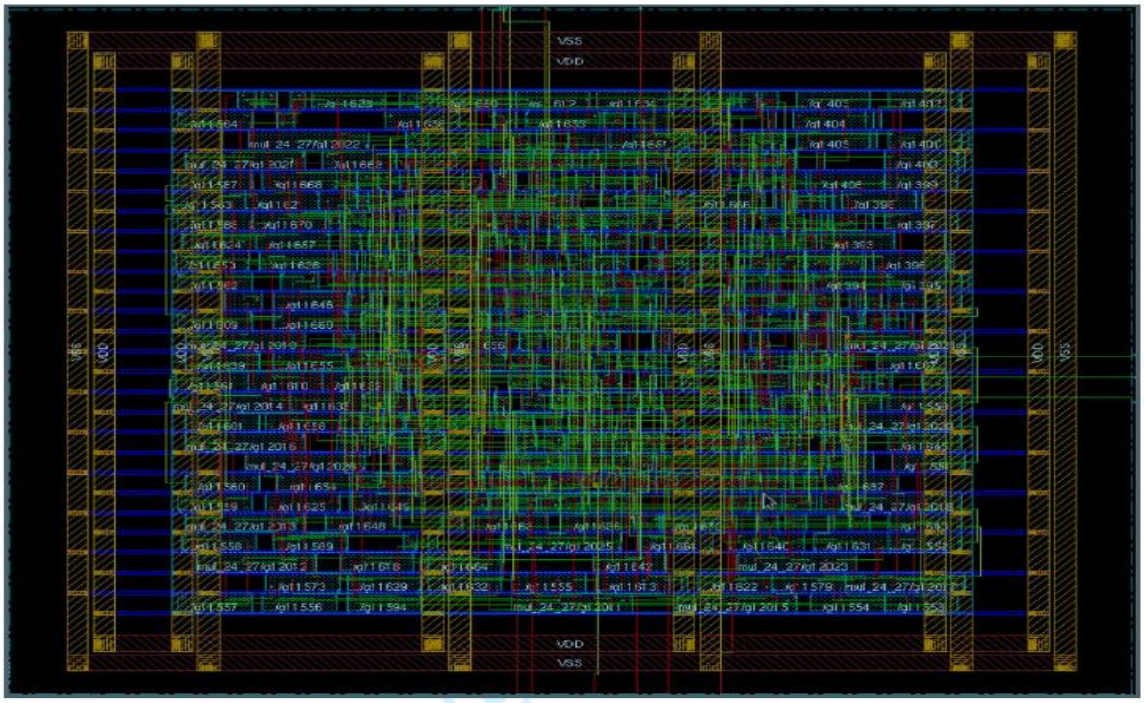


Fig. 20 Final design of ALE-NLMS 16

4.5 Metrics Utilization of ALE-LMS Architecture

The utilization of power is represented in Tab. 9 and the total time to complete the process is shown in Tab. 10. The performance comparison of the ALE architecture is analyzed for NLMS algorithm with two different adaptive filter orders 8 & 16.

Major parameters like area, gates, power utilization and computation time for the architectures are computed in Tab. 11.

Table 8: Area and gate Utilization of ALE-NLMS Architecture

Parameter	NLMS 8 (µm)		NLMS 16 (µm)	
	Gates	Area	Gates	Area
Sequential	830	21919.8	1063	26799.6
Inverter	1193	2713.49	3148	7176.17
Logic	14480	118815	31991	270465

From the table 8, we can deduce that gates and area of NLMS order 16 consumes more than NLMS order 8. The performance comparison of the ALE architecture analyzed both adaptive algorithms LMS and NLMS with two different adaptive filter orders 8 & 16.

Major parameters like area, gates, power utilization and computation time for the architecture shown in Tab.8 - 12.

Table 9: Power Utilization of ALE-NLMS Architecture

Filter order	Leakage power(nW)	Dynamic power(nW)	Total power(nW)
NLMS 8	619941	3927030	4546971
NLMS 16	1286139	11080538.561	12366677.898

From the table 9, we can deduce that leakage and dynamic power of NLMS order 16 consumes more than NLMS order 8.

Table 10: Timing Utilization of ALE-NLMS Architecture

Filter order	Fall time	Rise time	Total time
NLMS 8	261722	80230	341952
NLMS 16	273091	89521	362612

From the table 10, we can deduce that simulation time of NLMS order 16 consumes more than NLMS order 8.

Table 11: Performance analysis of ALE architecture (Area, power, delay and gates in NLMS filter order N = 8 & 16)

Parameter	NLMS 8	NLMS 16
Area (µm)	143448	304441
Gates	16503	36202
Power (nW)	4546971	12366677.5898
Time (Ps)	3419527	3626129

Table 12: Performance analysis of ALE architecture (Area, power, delay and gates in LMS & NLMS filter order N = 8 & 16)

Parameter	LMS 8	LMS 16	NLMS 8	NLMS 16
Area (µm)	83622.3	162895	143448	304441
Gates	9285	18034	16503	36202
Power (nW)	5050192	10235234.828	4546971	12366677.5898
Time (Ps)	823352	953641	3419527	3626129

From the table 11, we can deduce that parameter utilization of NLMS order 16 consumes more than NLMS order 8. NLMS 32 will consume more than NLMS 16 and continues according to the filter order.

4.6 Chapter Summary

In this chapter, ALE (LMS) and ALE (NLMS) architecture and their performance analysis are computed. Area occupied of ALE (LMS) design is 0.08 m, the total power consumed is 5.05 mW and the computation time is 0.82 μs.

Area occupied of ALE NLMS design is 0.14 m, the total power consumed is 4.54 mW and the computation time is 0.03 μs.

Due to the heterogeneity of metrics, it is difficult to make a direct comparison with LMS & NLMS designs, however their choice may vary with different signal processing industrial applications.

LMS has less computational complexity (simple and robust) than computational time for NLMS is better than LMS.

Chapter 5

Hardware Realization of ANN-ALE LMS/NLMS Design

5.1. Introduction

Observed input auscultation signals are usually classified into heart and lung sound signals. The heart sounds have high spectrum but the lung sounds contain less spectrum so the lung sound signal is separated and isolated to detect and diagnose disease in a respiratory system. Various techniques are used to reduce the noise and other signal interferences. Earlier systems for de-noising biomedical signals for noise cancellation use adaptive noise cancellation with adaptive algorithms (ANC).

An adaptive line enhancer with adaptive algorithms (ALE) is an improvement over the earlier ANC-based systems because of its ability to separate the LSS from the HSS resulting in quick and accurate diagnoses of disease. ANNs will be integrated with an ALE system to achieve results with minimum computation.

Training these networks in supervised learning technique and sigmoid activation function to achieve a better solution in real time. They conclusively provide the optimal solution in real time lung sound signal separation.

Artificial Neural Networks (ANNs) can solve many real world problems in the areas of signal processing and bio medical engineering areas. Artificial neural networks (ANNs) have been used successfully in solving pattern classification and recognition problems, function approximation and predictions.

For fast computation and processing of biomedical signals, Artificial Neural Networks (ANNs) are integrated to reduce response time from input to output system. Normally a NN is classified into two types namely feed forward and feedback neural network which use different learning algorithms to train the network.

Feed-forward ANNs (FFNN) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. They are extensively used in pattern recognition.

5.2. Artificial Neural Network -ALE Design (ANNs)

Feedback networks are one of the most powerful techniques in large computations to obtain the exact or expected output using feedback. These networks use a feedback loop to process the signal in both directions.

The feedback networks dynamically change the values until they reach an equilibrium point. Figure 21. shows the proposed architecture for integrated ANN ALE with adaptive algorithm.

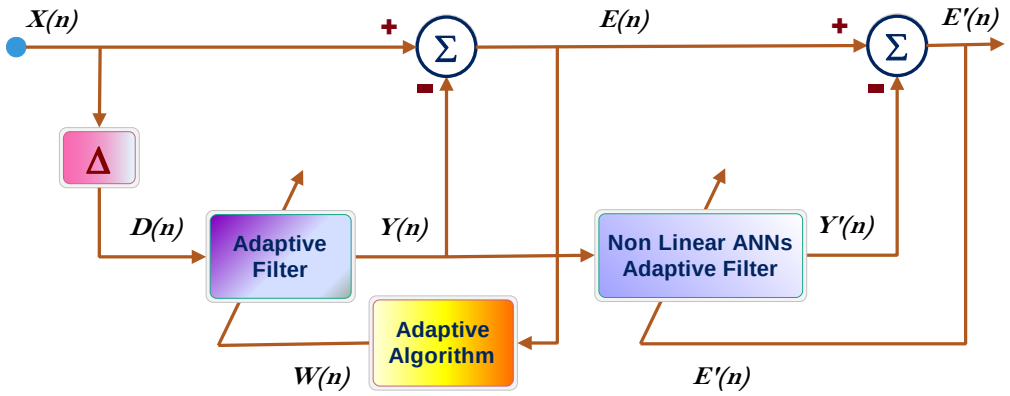


Figure 21. Adaptive line enhancer with Non-Linear ANNs Design

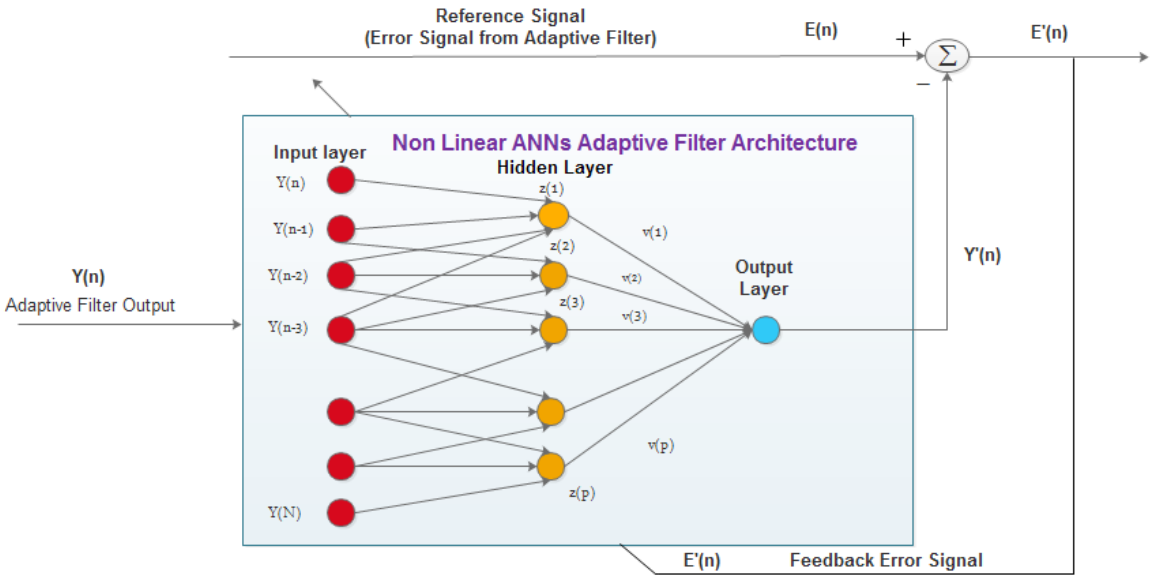


Figure 22. Non-Linear ANNs Design

The proposed neural network (FFNN) contains three layers in a supervised learning network for both feed forward and feedback. The three-layered network comprises the input, hidden,

and output layers. During the training phase, the training data is fed into the input layer. The data is propagated to the hidden layer and then to the output layer.

The error signal E(n) [48] is given by:

$$E'(n) = E(n) - Y'(n) \tag{30}$$

Where E(n) is the desired signal, Y'(n) is the estimated signal from the Non Linear ANN adaptive filter and it is calculated by:

$$Y'(n) = f\left(\sum_{j=1}^{P}(\mathrm{v}_j(n)z_j(n) + a\right) \tag{31}$$

where v(n) is the second weight vector at the hidden layer, P is the number of neurons and a is the bias value at the output layer, z(n) is the neuron value at the hidden layer [48] and it is calculated by:

$$z(n) = f\left(\sum_{i=1}^{N}(\mathrm{w}_j(\mathrm{n})\,\mathrm{Y}_j(\mathrm{n}) + \mathrm{b}_i)\right) \tag{32}$$

where w(n) is the first weight vector at the hidden layer, Y(n) is the reference signal, f(x) is the sigmoid function, b is the bias value at the hidden layer and N is the number of inputs to the neuron. The neuron transfer function is chosen to be the hyperbolic tangent nonlinear sigmoid function.

All the weights are initialized to a small random value between 1 and -1 and are updated at each iteration using the error back propagation algorithm. This learning algorithm is most commonly used to train the NN. The new weight values [48] are given by:

$$w(n+1) = w(n) + (h_1\,\mathrm{E}'(n)v(n)z'(n)\,\mathrm{Y}(n)) + \alpha(w(n-1) - w(n)) \tag{33}$$

$$v(n+1) = v(n) + (h_2\, E'(n)\, y'(n) z'(n)\, z(n)) + \beta(v(n-1) - v(n))$$

(34)

Where h1 and h2 are the learning rate, α and β are the momentum factor, z'(n) is the derivative of the hidden layer neuron, and y'(n) is the derivative of the output layer neuron.

5.3 Implementation of ANN-ALE Design (LMS/NLMS)

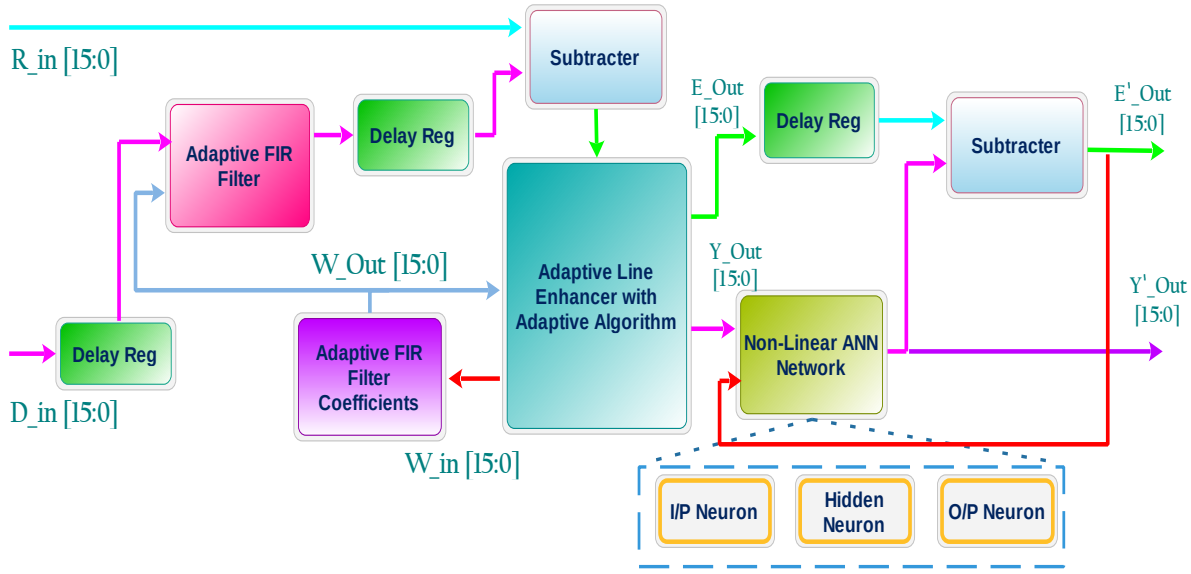


Figure 23: Architecture of ANN ALE with LMS algorithm

Table 13: Device utilization of ANN - ALE - LMS architecture in Xilinx environment

No	Parameter	LMS_8_Tap	
1	**No of Slices**	2848 out of 2448	**116%**
2	**No of Slice FFs**	737 out of 4896	**15%**
3	**No of LUTs**	5268 out of 4896	**107%**
4	**No of IOBs**	184 out of 158	**116%**
5	**No of GCLKs**	1 out of 24	**4%**
6	**Completion time**	75.30secs	
7	**Memory usage**	207612 kilobytes	

The hardware realization of the proposed architecture ANN-ALE-LMS/NLMS is shown in Fig. 23 and Fig.24 and the device utilization of the architecture obtained in Xilinx is shown in Tab. 13. and Tab. 14.

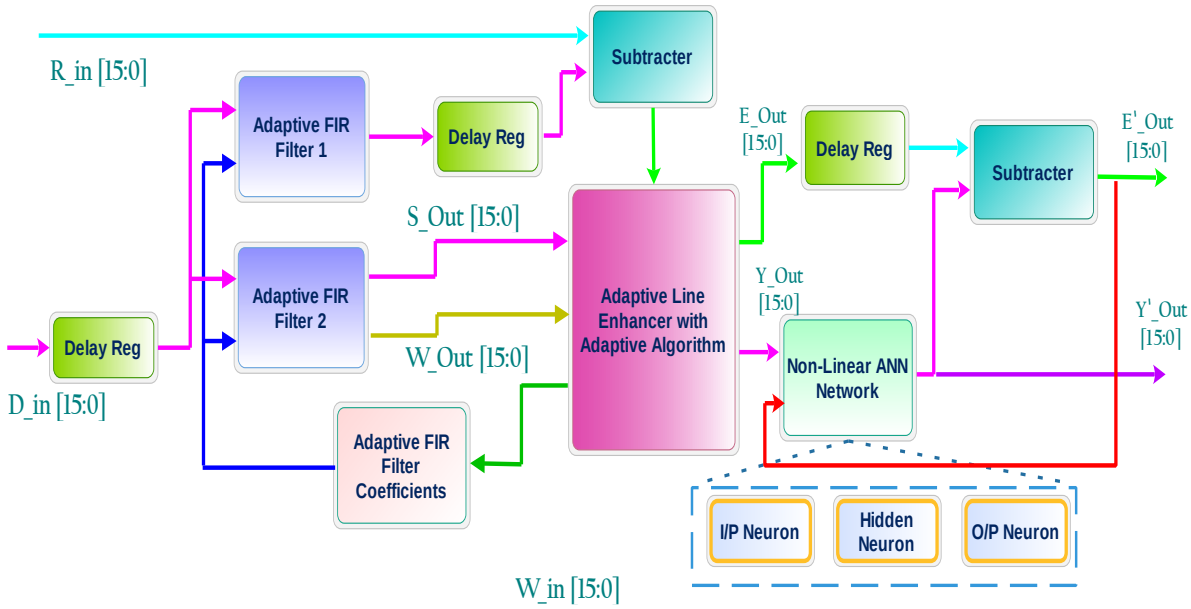


Figure 24: Architecture of ANN ALE with LMS algorithm

Table 14: Device utilization of ANN- ALE - NLMS architecture in Xilinx environment

S.No	Parameter	NLMS_8_Tap	
1	**No of Slices**	2650 out of 2448	**108%**
2	**No of Slice FFs**	870 out of 4896	**17%**
3	**No of LUTs**	4669 out of 4896	**95%**
4	**No of IOBs**	184 out of 158	**116%**
5	**No of GCLKs**	2 out of 24	**8%**
6	**Completion time**	96.95secs	
7	**Memory usage**	270652 kilobytes	

5.4 ASIC Design Steps

The physical design is implemented in CADENCE ENCOUNTER physical design environment with TMSC 90nm standard cell library. To implementing the proposed

architecture in a physical design to followed the steps floor plan, power constraints, timing constraints, nano and global routing, placed the design in transistor level, clock constraints, design rule checking and geometry verification to obtain the final implementation of the design in the form of GDS II.

The goal of the cadence university software program [49-50] is to grant easy access to leading electronic design automation and Tensilica processor configuration and extension tools for educational institutions around the world.

Floor Planning

This is the initial stage of the RTL to GDS II conversion for specifying the area and placing the design in its working environment. Here, the core to die, core to IO boundary, and where it is to be placed in the working environment

Power Planning

In this stage the layers of power (VDD) and ground (VSS) are chosen and the default material for both layers is assigned. There are two ways of power planning implemented namely power ring and power strip. The power planning used to implement the IO pads and the selected inner layers with the required number of sets with acceptable spacing.

Special Routing

Proper layer material is chosen to make the interconnection between the power nets.

Placement

This involves placing the entire design in a specified floor planning area. The designs are made to implement the standard cell library

Pre-Time Analysis

In this analysis, the users created by the library are considered, such as best case and worst case to obtain the setup and hold time for the entire design.

Optimization

Entire optimized design which depends on the standard cell library. Here the 90nm standard cell library is chosen to optimize the design.

Clock Synthesis

This stage involves the creation and optimization of global and local clocks for the whole design. The clocks are designed to reduce the delay between the nets and devices

Nano Routing

In this stage, delay and time to flow the signal or data between the devices are reduced

Post-Time Analysis

In this analysis, the users created by the library are considered, such as best case and worst case to obtain the setup and hold time for the entire design.

Verification

This is the final stage of the physical design implementation, during which the entire design is verified at various levels, for instance, connectivity, geometry etc.,

GDS II

Finally, the design is converted to a graphical database structure (GDS) II format.

5.5 Metrics Utilization of ANN - ALE-LMS/NLMS Architecture

The proposed system is simulated and synthesized to obtain major design parameters considered in VLSI technology such as area, gates, power and timing analysis obtained in cadence RC synthesizer. The parameter values of area and gates shown in Table 15, the utilization of power represented in Table 16 and the total time to complete the process is shown in Table 17. The performance comparison of the ANN-ALE architecture analyzed for LMS/NLMS algorithm adaptive filter order 8. From the table 15, we can deduce that gates and area utilization of NLMS order 8 consumes more than LMS order 8.

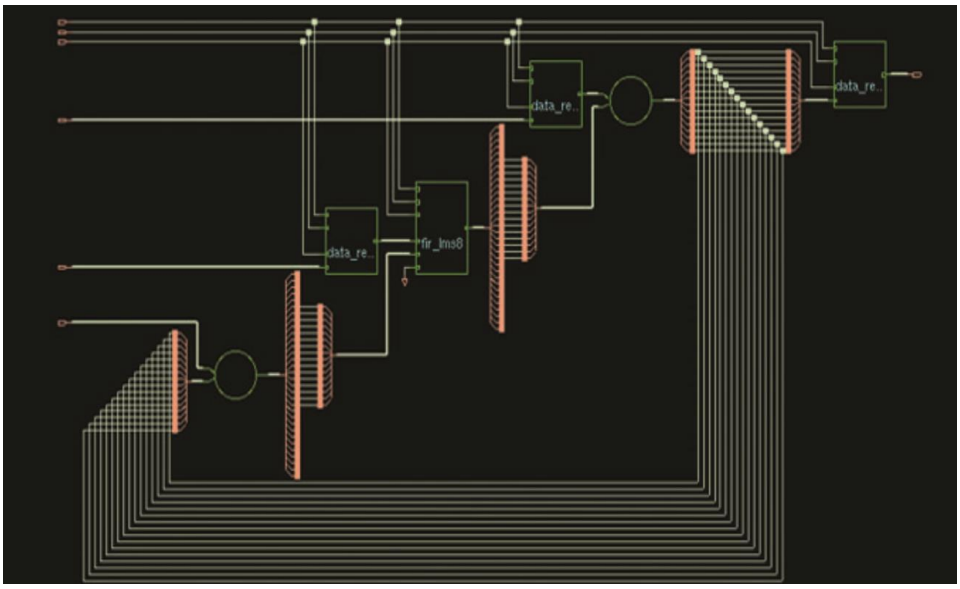
Table 15: Area and gate Utilization of ANN ALE Architecture

Parameter	LMS 8		NLMS 8	
	Gates	Area	Gates	Area
Sequential	654	16223.4	1196	30949.6
Inverter	1250	2838.38	1681	3821.59
Logic	11745	106213	17990	150329

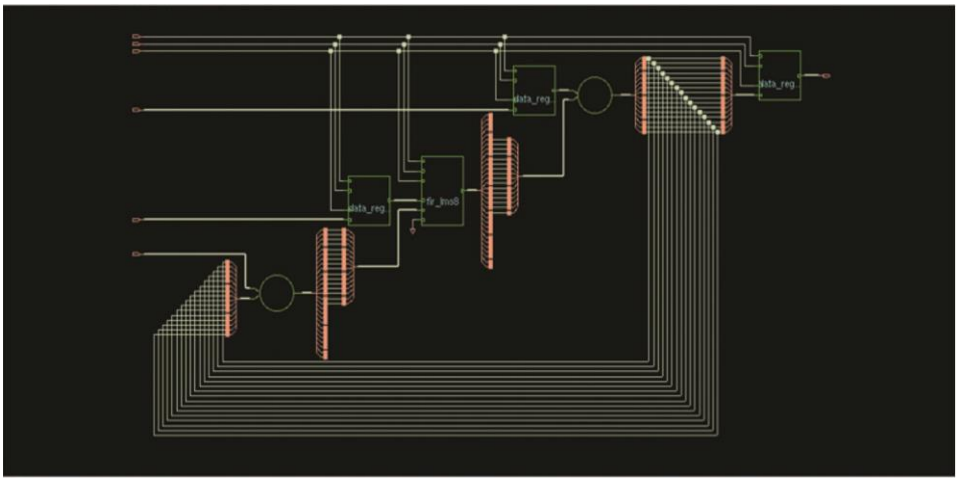
Major parameters like area, gates, power utilization and computation time for the architecture are represented in Table 15 - 18. Final chip design is shown in Fig.25. From the table 16, we can deduce that leakage and dynamic power of NLMS order 8 consumes more than LMS order 8.

Table 16: Power Utilization of ALE-NLMS Architecture

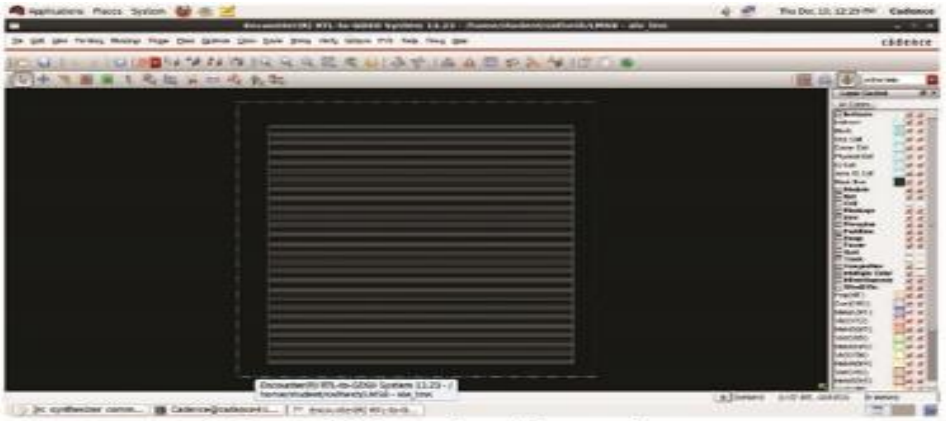
Filter order	Leakage power(nW)	Dynamic power(nW)	Total power(nW)
LMS 8	525991	6119681	6645672
NLMS 8	794572	5362872	6157443



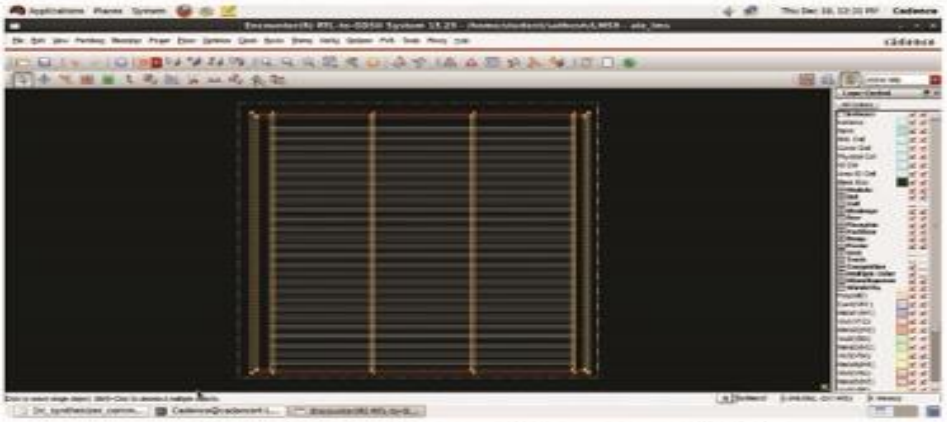
a) RTL VIEW of ANN ALE-LMS



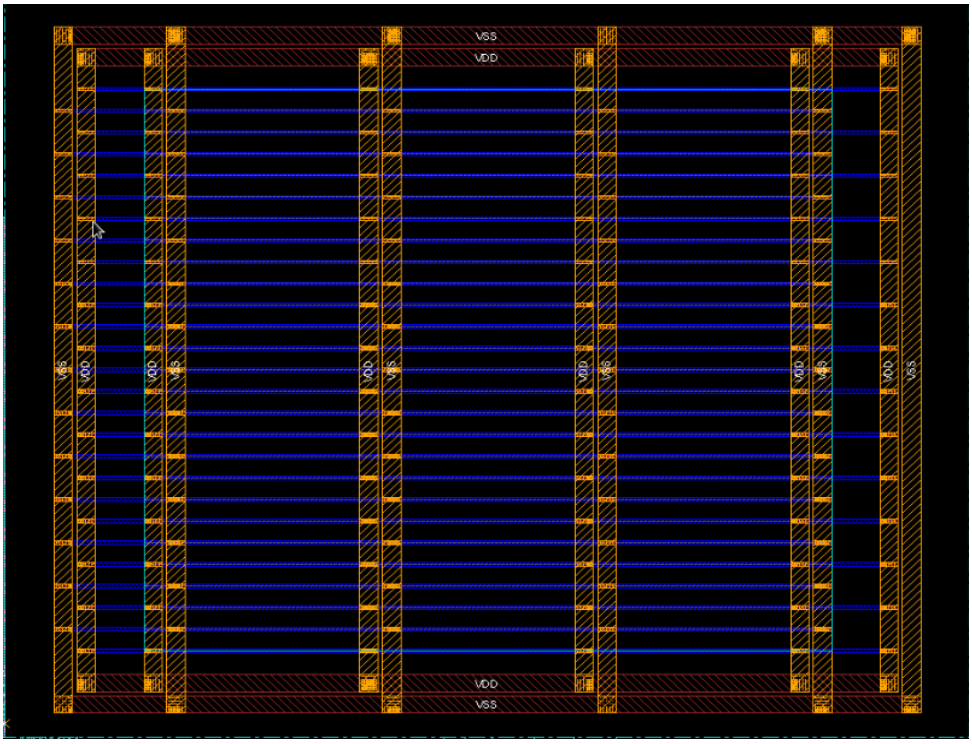
(b) Synthesized RTL view of ANN ALE-LMS



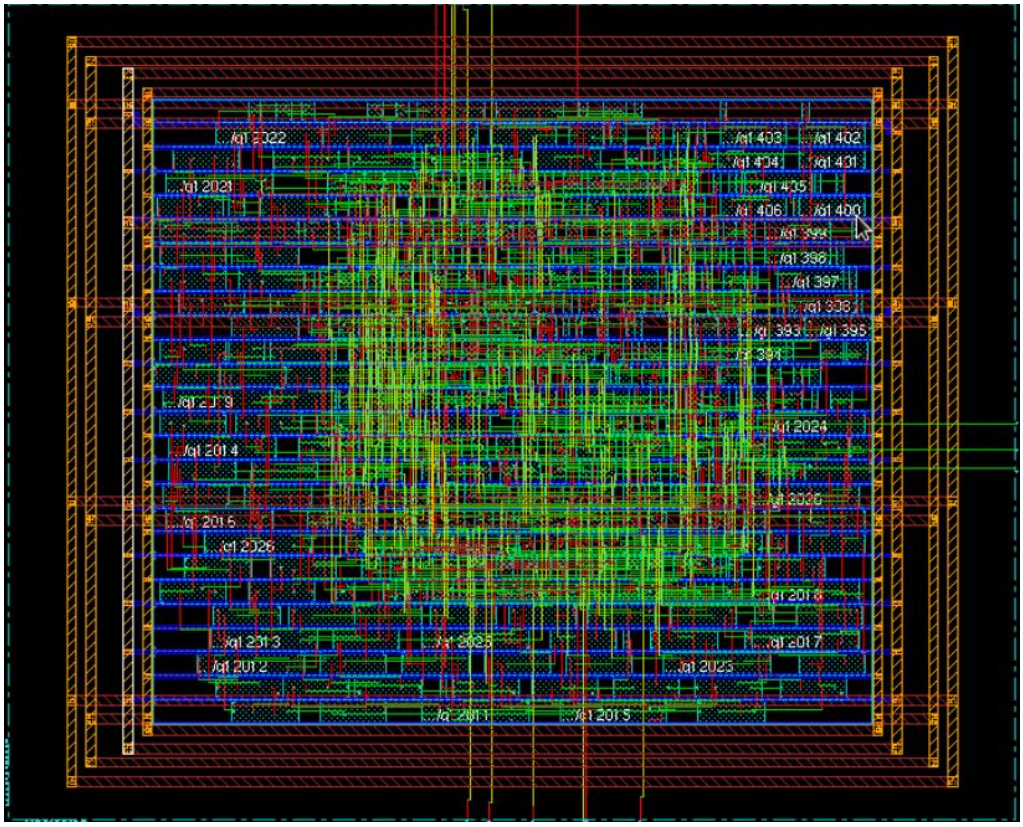
c) Specifying the Floorplan



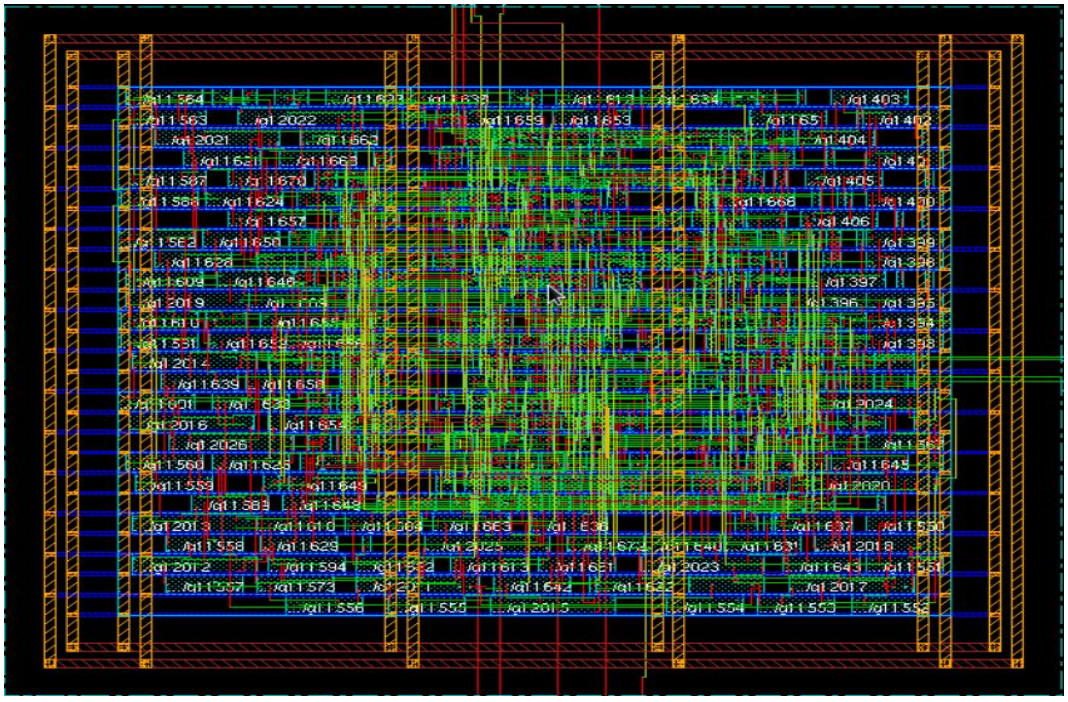
d) Adding Power Ring and Strip



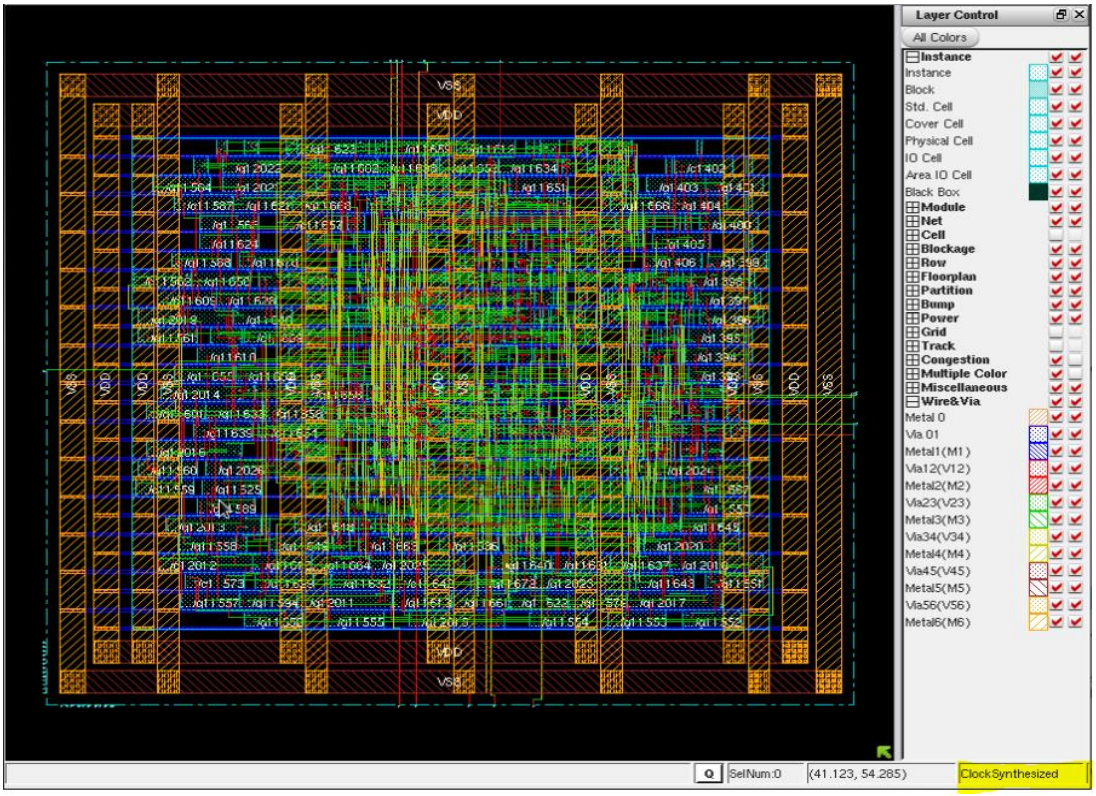
(e) Special routing of ANN ALE-LMS



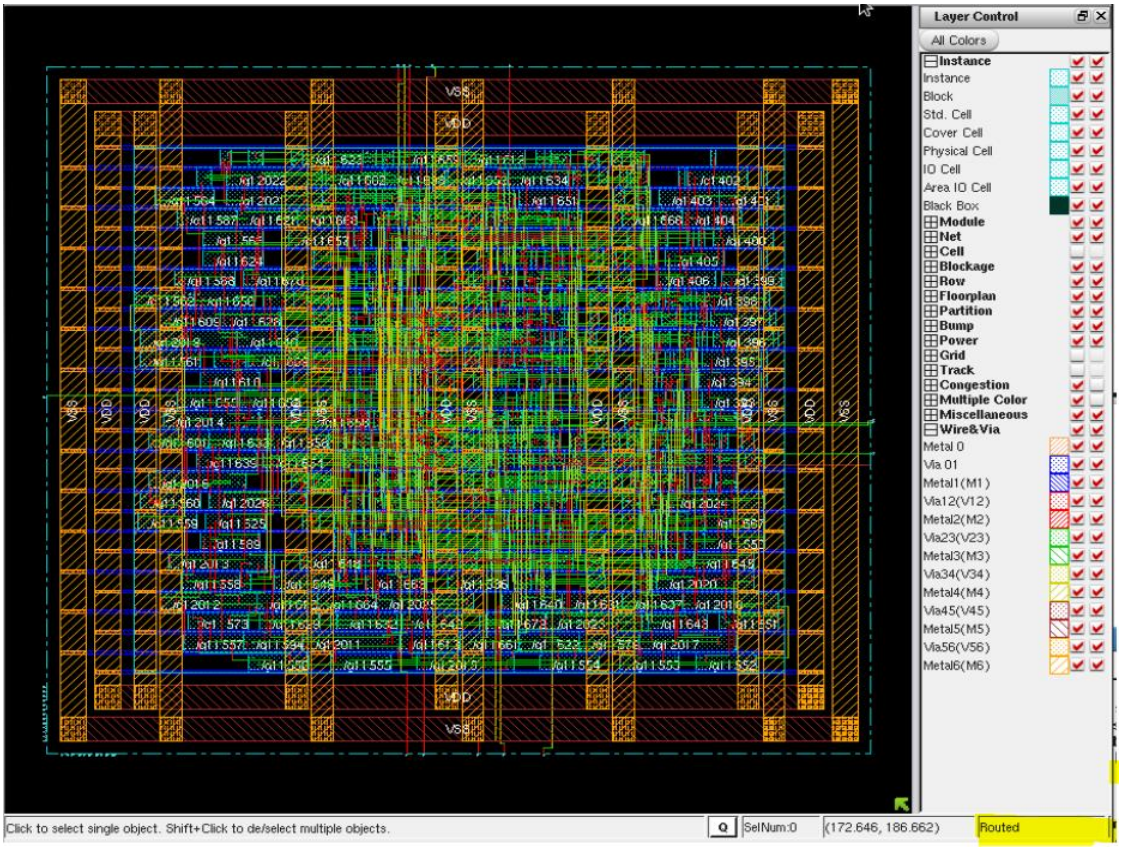
(f) Placing the design of ANN ALE-LMS



(g) Optimized the design of ANN ALE-LMS



(h) Synthesized clock tree of ANN ALE-LMS



(i) Optimized the Nano routing of ANN ALE-LMS

Fig. 25 ASIC Design Steps for ANN ALE-LMS 8

[(a) RTL View of ANN ALE-LMS (b) Synthesized RTL view of ANN ALE-LMS (c) Specifying the Floor Plan (d) Adding Power Ring and Strip (e) Special routing of ANN ALE-LMS (f) Placing the design of ANN ALE-LMS (g) Optimized the design of ANN ALE-LMS (h)Synthesized clock tree of ANN ALE-LMS (i) Optimized the Nano routing of ANN ALE-LMS]

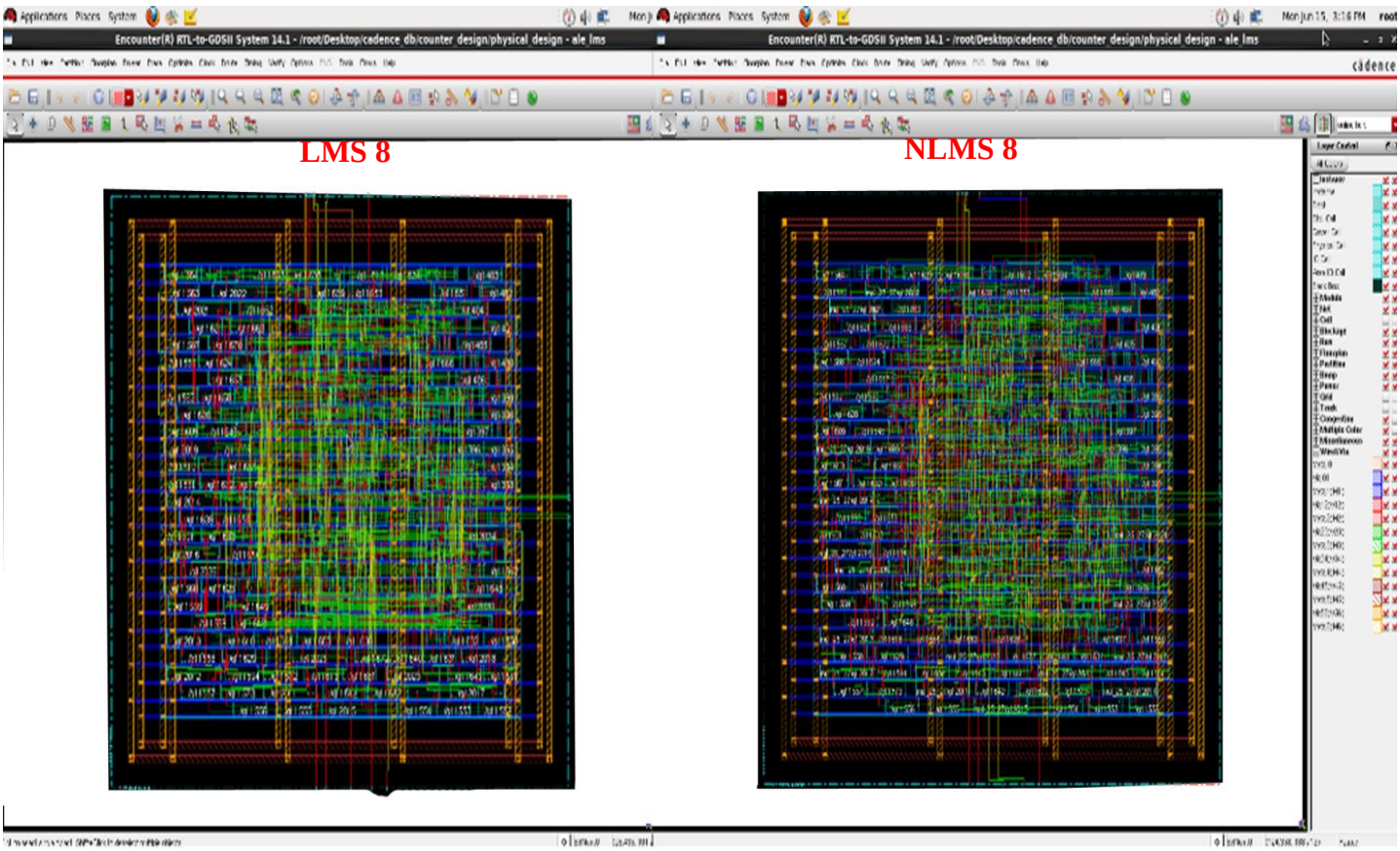


Figure 26 Final design of ANN ALE-LMS/NLMS 8

Table 17: Timing Utilization of ALE-NLMS Architecture

Filter order	Fall time	Rise time	Total time
LMS 8	80230	261722	341952
NLMS 8	239840	605110	844950

From the table 17, we can deduce that simulation time of NLMS order 8 consumes more than LMS order 8.

Table 18: Performance analysis of ANN ALE architecture

Parameter	LMS 8	NLMS 8
Area (µm)	125275	185101
Gates	13649	20867
Power (nW)	6645672	6157443
Time (Ps)	341952	844950

From the table 18, we can deduce that NLMS order 8 consumes more gates/area/power utilization than LMS order 8. NLMS simulation time also increased double than LMS 8.

5.6 Chapter Summary

The neural network-based implementation of adaptive line enhancer with adaptive algorithms provides the better performance compared to a traditional adaptive line enhancer.

During the implementation of the proposed system, major parameters including delay time, area, and power were optimized in Cadence Encounter.

In this ANN - ALE LMS architecture, the area occupied is 0.12m, the total power consumed is 6.65 mW and the computation time of the proposed system is 0.3 µseconds.

In ANN - ALE NLMS architecture, the area occupied is 0.18m, the total power consumed is 6.16 mW and the computation time of the proposed system is 0.8 µseconds.

ANN - ALE LMS design is better compared to ANN - ALE NLMS design according to the area and time computation metrics.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

Stetho-Us is a portable stethoscope which contains a single chip which enables many applications in medical and health related technologies (Orthopedic, Coronary artery disease, congestive heart failure, heart attack sensing). ALE and our proposed ANN-ALE will focus on sensing auscultation and provides high SNR signal output signals whereas ASIC focusses on chip implementation. Any pain, tightness, pressure and discomfort (angina) can be sensed and enhanced output through ASIC which leads to correct diagnosis and prompt treatment.

Initially in this work, a detailed comparison among widely used ICA for auscultation signals was presented. A number of ICA approaches have been used for signal analysis, and even more ICA algorithms exist; however, the impact of using different algorithms on the results in auscultation is largely unexplored. This analysis will be used to compare and identify the best strategy for extracting auscultation signal based on the use of ICA algorithms.

This work implemented an adaptive line enhancer with adaptive algorithm design (ALE LMS/NLMS) for extracting auscultation signal. The hardware implementation of adaptive filters is a challenging issue in real time practical auscultation. In this work, hardware realization of ALE with LMS/NLMS algorithm design is shown in cadence TSMC 90 nm standard cell library environment. NC sim and RC lab were used for functional verification and design constraints, and the physical design is implemented in encounter to obtain the GDS II. In this architecture, the area occupied is 0.08 mm, the total power consumed is 5.05 mW and the computation time of the proposed system is 0.82 μs for LMS design and the area occupied is 0.14 mm, the total power consumed is 4.54 mW and the computation time of the proposed system is 0.03 μs for NLMS design that will pave a better way in future electronic stethoscope design.

Finally, we proposed a new cognitive architecture (Artificial neural network- Adaptive line enhancer with LMS/NLMS algorithms) for fast computation and noise cancellation. The results obtained (ANN - ALE (LMS/NLMS)) proved that the computation of real time sound signal separation can be hardware realized and implemented in a chip which will be the most useful in future medical auscultation stethoscopes design.

6.2 Recommendation

Due to the heterogeneity of metrics, it is difficult to make a direct comparison with LMS & NLMS designs, however their choice may vary with different signal processing industrial applications. LMS has less computational complexity (simple and robust) than NLMS but their drawbacks are low convergence rate and signal to noise ratio. All our results are simulation and hardware implementation will be our future goal.

The next step is to fabricate the chip and perform real-world tests. If the performance is consistent with the models, the technology could enter the market as currently designed. Thus, our strategy is to license the technology to one of the large manufacturers. The intellectual property will be transferred to the company which will serve as the licensor. The outcome, a readily available advancement in a fundamental diagnostic tool will improve medical care, notably in rural and underserved areas.

Small and medium sized business organization can use their chip fabrication facility when our product reaches the full commercialization stage in the future. Additionally, they can support systems integration to generate device that can be practiced in the healthcare domain. They can also support our marketing and outreach efforts to medical instrumentation manufacturers and the various stakeholders in the healthcare domain.

6.3 Future Scope

We are requesting the Office of Research Affairs of Adama Science and Technology University (ASTU) to extend the phase-II of this project for ASIC Chip Implementation of our proposed Auscultation design.

- IC design can lead to many future applications. One of the major applications is that we can design carbon nano tube (CNT) based micro electro mechanical system (MEMS) microphone. This CNT based MEMS microphone (CBMM) design to replace the microphone currently used in electronic stethoscopes.

- This study opens several lines for future work. Analyzing the design with other adaptive algorithms and evaluation of various other metrics are some of the future works of this research.

- This work can be extended by following the same fashion for other signal analysis and may vary with different engineering applications.

- Proposing various novel cognitive architectures for real time bio-signal separation will be our future goal.

- This work will lead to develop novel cognitive models and architectures and its ASIC implementation (especially bio signal modeling, biomechanics and cognitive based) to understand signal capabilities through signal enhancement, feature extraction, system identification and therapy.

- Domestic researchers are voluminous and bountiful. They are expert in disjunct fields. But the researchers in cognitive/Reliability/VLSI/Biomedical fields are discrete. Researchers in Ethiopia will be motivated to implement in ASIC which proves the project superiority. The proposed design also be implemented in nano scale technology and MEMS technology in future.

Publications

Published:

S.Rajkumar et.al.; "Auscultation Performance Metrics Computation using ICA Algorithms", Proceedings of 2nd Deep Learning Indaba-X Ethiopia Conference 2021, Adama Science and Technology University, Ethiopia, January 27-29, 2022.

Published:

Rajkumar et.al.; "Chip Layout for Adaptive Line Enhancer Design using Adaptive Filtering Algorithms and Metrics Computation for Auscultation Signal Separation", Journal of Beijing Institute of Technology, 31(3): 317-326, 2022 (Scopus Indexed)

Under Process:

Bayisa Taye Mulatu et.al.; "Metrics Computation for Auscultation Signal Separation by Radial Bias Function and ANN based ALE filter", 2023.

References

1. K. Sathesh, S. Rajkumar, and N. K Goyal, “Least mean square (LMS) based neural design and metric evaluation for auscultation signal separation” *Biomed Signal Process Control*, vol. 59, pp. 101784, 2020.

2. Klum M., Urban M., Tigges T, et al. “Wearable Cardiorespiratory Monitoring Employing a Multimodal Digital Patch Stethoscope: Estimation of ECG, PEP, LVET and Respiration Using a 55 mm Single-Lead ECG and Phonocardiogram”, *Sensors (Basel),* Vol. 20, No. 7, April 2020.

3. Araiza, G.D., Jordán-Ríos, A., Sierra-Fernández, C., Juárez-Orozco L.E., “On stethoscopes, patient records, artificial intelligence and zettabytes: a glimpse into the future of digital medicine in Mexico”. *Arch Cardiol Mex*, Vol. 90, No. 2, pp. 193–199, January 2020.

4. Baptista, R., Silva, H., and Rocha, M. “Design and development of a digital stethoscope encapsulation for simultaneous acquisition of phonocardiography and electrocardiography signals: the Smart Heart case study”, *Journal of Medical Engineering & Technology*, Vol. 44, No. 4, pp. 153–161, May 2020.

5. Silverman, B., Balk, M. “Digital Stethoscope—Improved Auscultation at the Bedside”. *Am. J. Cardiol*, Vol. 123, No. 6, pp. 984–985, March 2019.

6. Swarup, S., Makaryus, A.N. “Digital stethoscope: Technology update”. *Med. Devices*, Vol. 11, pp. 29–36, January 2019.

7. D. Emmanouilidou, E. D. McCollum, D. E. Park and M. Elhilali, "Computerized Lung Sound Screening for Pediatric Auscultation in Noisy Field Environments," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 7, pp. 1564-1574, July 2018

8. A. Tessema, D. Nemomssa, L. Simegn. “Acquisition and Classification of Lung Sounds for Improving the Efficacy of Auscultation Diagnosis of Pulmonary Diseases”. *Med Devices*., vol. 15, pp. 89-102, 2022

9. S. H. Lee et al., “Fully portable continuous real-time auscultation with a soft wearable stethoscope designed for automated disease diagnosis”. *Sci. Adv,* Vol..8, 2022.

10. F. Cardoso et.al., “Lung Auscultation Using the Smartphone—Feasibility Study in Real-World Clinical Practice”. *Sensors,* Vol.21, 4931, 2021.

11. G. Shah, P. Koch and C. B. Papadias, “On the Blind Recovery of Cardiac and Respiratory Sounds”, *IEEE Journal of Biomedical and Health Informatics*, Vol. 19, No.1, pp.151-157, January 2015.

12. A. Mondal; I. Saxena; H. Tang; P. Banerjee, "A noise reduction technique based on nonlinear kernel function for heart sound analysis”, *IEEE Journal of Biomedical and Health Informatics*, February 2017.

13. N. J. Bershad, E. Eweda and J. C. M. Bermudez, "Stochastic Analysis of an Adaptive Line Enhancer/Canceler with a Cyclostationary Input", *IEEE Transactions on Signal Processing*, Vol. 64, No. 1, pp. 104-119, January 2016.

14. J. W. Kelly, D. P. Siewiorek, A. Smailagic and W. Wang, "An Adaptive Filter for the Removal of Drifting Sinusoidal Noise Without a Reference", *IEEE Journal of Biomedical and Health Informatics*, Vol. 20, No. 1, pp. 213-221, January 2016.

15. F. Pan, P. He, C. Liu, T. Li, A. Murray and D. Zheng, "Variation of the Korotkoff Stethoscope Sounds During Blood Pressure Measurement: Analysis Using a Convolutional Neural Network," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1593-1598, Nov. 2017.

16. S. Afzal et.al, "Novel Approaches to Identify Clusters Using Independent Components Analysis with Application", *Mathematical Problems in Engineering*, vol. 2023, 13 pages, 2023.

17. B. Sen and K. K. Parhi, "Extraction of common task signals and spatial maps from group fMRI using a PARAFAC- *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 1113-1117.

18. P. Tillet, H. T. Kung and D. Cox, "Infomax-ICA using Hessian-free optimization," *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans*, LA, pp. 2537-2541, 2017

19. J. C. Wang et al., "VLSI Design for Convolutive Blind Source Separation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp. 196-200, 2016.

20. E. S. Juan, I. Soto, G. Salinas and P. Adasme, "Separation of VLC signals using FastIca and InfoMax", *Proceedings of the First South American Colloquium on Visible Light Communications (SACVLC), Santiago*, pp. 1-6. 2017.

21. N. Falco, J. A. Benediktsson and L. Bruzzone, "A Study on the Effectiveness of Different Independent Component Analysis Algorithms for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2183-2199, 2014.

22. L. Breuer, J. Dammers, T. P. L. Roberts and N. J. Shah, "A Constrained ICA Approach for Real-Time Cardiac artifact rejection in magnetoencephalography", *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 405-414, 2014.

23. C. Yuan and J. Zhang, "Extraction of single-trial evoked potentials with Extended Infomax ICA algorithm and its applications to BCI systems," *Proceedings of the 33rd Chinese Control Conference*, Nanjing, pp. 7139- 7144, 2014.

24. W.L.Lee, et.al., "An Improved P300 Extraction Using ICA-R For P300-BCI Speller," *Proceedings of the 35th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, 7064- 7067, 2013.

25. G. R. Naik, S. E. Selvan and H. T. Nguyen, "Single-Channel EMG Classification with Ensemble-Empirical- Mode-Decomposition-Based ICA for Diagnosing Neuromuscular Disorders," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 7, pp. 734-743, July 2016.

26. Rong Yuan, Shengli Xie, Zhenni Li, Zhaoshui He, "Adaptive Fast Independent Component Analysis Methods for Mitigating Multipath Effects in GNSS Deformation Monitoring", *Journal of Sensors*, vol. 2022, Article ID 4604950, 2022.

27. S. Basiri, E. Ollila and V. Koivunen, "Alternative Derivation of FastICA with Novel Power Iteration Algorithm," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1378-1382, Sept. 2017.

28. X. Zhao and H. Yang, "A New Method to Calculate the Utility Harmonic Impedance Based on FastICA," *IEEE Transactions on Power Delivery*, vol. 31, no. 1, pp. 381-388, Feb. 2016
.
29. R. Luan, G. Wen, R. Zhang, Z. Chen and Z. Zhang, "Porosity defect detection based on FastICA-RBF during pulsed TIG welding process," *Proceedings of* the 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, pp. 548-553, 2017

30. S. He, Z. Tong, M. Tong, S. Tang, M. Li and L. Liang, "Research on sound separation and identification of trapped miners based on fastica algorithm," Proceedings of the 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), Macau, pp. 228-231, 2017.

31. G. Fontgalland and P. I. L. Ferreira, "Combining Antenna Array Elements by Using ICA Method for Remote Sensing of Sources," in *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 234-237, 2017.

32. L. Cai, X. Tian and S. Chen, "Monitoring Nonlinear and Non-Gaussian Processes Using Gaussian Mixture Model-Based Weighted Kernel Independent Component Analysis," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 1, pp. 122-135, Jan. 2017.

33. Y. Zhang, W. Du and X. G. Li, "Observation and Detection for a Class of Industrial Systems," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6724-6731, Aug. 2017.

34. Y. Zhang and Q. Jia, "Complex Process Monitoring Using KUCA with Application to Treatment of Waste Liquor," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 2, pp. 427-438, March 2018.

35. L. Feng and R. Sun, "Dynamic kernel independent component analysis approach for fault detection and diagnosis," *2017 Chinese Automation Congress (CAC)*, Jinan, 2017, pp. 2193-2197.

36. Q. X. Zhu, Q. Q. Meng, Y. Xu and Y. L. He, "Research and application of KICA-AROMF based fault diagnosis,"*2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, Taipei, 2017, pp. 215-220

37. X. Peng, Y. Tian, Y. Tang, W. Du, W. Zhong and F. Qian, "An online performance monitoring using statistics pattern based kernel independent component analysis for non-Gaussian process," *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, 2017, pp. 7210-7216.

38. Z. Huang et.al, "Key Feature Extraction Method of Electroencephalogram Signal by Independent Component Analysis for Athlete Selection and Training", *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 6752067, 2022.

39. N. J. Bershad, E. Eweda and J. C. M. Bermudez, "Stochastic Analysis of an Adaptive Line Enhancer/Canceler with a Cyclostationary Input", *IEEE Transactions on Signal Processing*, Vol. 64, No. 1, pp. 104-119, January 2016.

40. J. Taghia and R. Martin, "A Frequency-Domain Adaptive Line Enhancer with Step-Size Control Based on Mutual Information for Harmonic Noise Reduction", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, No. 6, pp. 1140-1154, June 2016.
41. J. W. Kelly, D. P. Siewiorek, A. Smailagic and W. Wang, "An Adaptive Filter for the Removal of Drifting Sinusoidal Noise Without a Reference", *IEEE Journal of Biomedical and Health Informatics*, Vol. 20, No. 1, pp. 213-221, January 2016.

42. G. Shah, P. Koch and C. B. Papadias, "On the Blind Recovery of Cardiac and Respiratory Sounds", *IEEE Journal of Biomedical and Health Informatics*, Vol. 19, No.1, pp.151-157, January 2015.

43. Ramli, RM, Noor, AOA & Samad, SA, 'Development of an adaptive line enhancer using nonlinear neural networks', *Proceedings of the 11th WSEAS international conference on Electronics, Hardware, Wireless and Optical Communications*, Cambridge, UK, pp. 133-137, February 22 – 24, 2012.

44. Roshahliza M Ramli, Ali O Abid Noor & Salina Abdul Samad, "A Review of Adaptive Line Enhancers for Noise Cancellation", *Australian Journal of Basic and Applied Sciences*, Vol. 6, No. 6, pp. 337-352. June 2012.

45. Gnitecki, J and Moussavi, Z.M, "Separating heart sounds from lung sounds-accurate diagnosis of respiratory disease depends on understanding noises", *IEEE Engineering in Medicine and Biology Magazine*, Vol. 26, No. 1, pp. 20-29, January 2007.

46. A. Mondal; I. Saxena; H. Tang; P. Banerjee, "A noise reduction technique based on nonlinear kernel function for heart sound analysis", *IEEE Journal of Biomedical and Health Informatics*, Vol. 22, No. 3, pp. 775-784. May 2018.

47. Roshahliza M Ramli, Ali O Abid Noor & Salina Abdul Samad, "A Review of Adaptive Line Enhancers for Noise Cancellation", *Australian Journal of Basic and Applied Sciences*, Vol. 6, No. 6, pp. 337-352. June 2012.

48. S. Rajkumar, K. Sathesh, and N. K Goyal, "Neural network-based design and evaluation of performance metrics using adaptive line enhancer with adaptive algorithms for auscultation analysis", *Neural Comput & Applic*, vol. 32, pp. 15131-15153, 2020.

49. University Program. (n.d.). https://www.cadence.com/en_US/home/company/cadence-academic-network/university-software-program.html/, Accessed May 20 2023.

50. *Home*. (n.d.). https://trial.cadence.com/ Accessed May 20 2023.

Budget Requirements:

S.N	ITEMS	UNIT COST	AMOUNT IN BIRR
1	Field subsistence	724*10 days	7 240.00
2	Travel Cost *	4days*2500birr/day	10,000.00
3	HDMI-VGI cable converter*	500birr*3 piece	1500.00
4	External hard disk, 2TB*	3500birr*1 piece	3500.00
5	Laptop*	62,000birr*1Laptop	62,000.00
6	Stationeries*	2000.00	2000.00
	Total Amount		**86240.00**

[11] The budget and items must be relevant to the proposed project and should be justified. It can be modified depending on the project.

Not Provided/Utilized - *

Utilized Budget Requirements:

No	ITEMS	UNIT COST	AMOUNT IN BIRR
1	Field subsistence	724*10 days	7240.00
	Total Amount		**7240.00**

Simulation Code

```
clear all
close all
clc

figure;p=plot(s);

title('EEG Signal')

fs = 500;

N=length(s);

waveletFunction = 'db8';
                [C,L] = wavedec(s,8,waveletFunction);

                cD1 = detcoef(C,L,1);
                cD2 = detcoef(C,L,2);
                cD3 = detcoef(C,L,3);
                cD4 = detcoef(C,L,4);
                cD5 = detcoef(C,L,5); %GAMA
                cD6 = detcoef(C,L,6); %BETA
                cD7 = detcoef(C,L,7); %ALPHA
                cD8 = detcoef(C,L,8); %THETA
                cA8 = appcoef(C,L,waveletFunction,8); %DELTA
                D1 = wrcoef('d',C,L,waveletFunction,1);
                D2 = wrcoef('d',C,L,waveletFunction,2);
                D3 = wrcoef('d',C,L,waveletFunction,3);
                D4 = wrcoef('d',C,L,waveletFunction,4);
                D5 = wrcoef('d',C,L,waveletFunction,5); %GAMMA
                D6 = wrcoef('d',C,L,waveletFunction,6); %BETA
                D7 = wrcoef('d',C,L,waveletFunction,7); %ALPHA
                D8 = wrcoef('d',C,L,waveletFunction,8); %THETA
                A8 = wrcoef('a',C,L,waveletFunction,8); %DELTA

                 Gamma = D5;
figure; subplot(5,1,1); plot(1:1:length(Gamma),Gamma);title('GAMMA');

                 Beta = D6;
                subplot(5,1,2); plot(1:1:length(Beta), Beta); title('BETA');

                 Alpha = D7;
              subplot(5,1,3); plot(1:1:length(Alpha),Alpha); title('ALPHA');

                 Theta = D8;
               subplot(5,1,4); plot(1:1:length(Theta),Theta);title('THETA');
                 D8 = detrend(D8,0);

                 Delta = A8;
                  %figure,
                plot(0:1/fs:1,Delta);subplot(5,1,5);plot(1:1:length(Delta),
                Delta);title('DELTA');

D5 = detrend(D5,0);

xdft = fft(D5);
freq = 0:N/length(D5):N/2;
```

```
xdft = xdft(1:length(D5)/2+1);
figure;subplot(511);plot(freq,abs(xdft));title('GAMMA-FREQUENCY');
[~,I] = max(abs(xdft));
fprintf('Gamma:Maximum occurs at %3.2f Hz.\n',freq(I));

D6 = detrend(D6,0);
xdft2 = fft(D6);
freq2 = 0:N/length(D6):N/2;
xdft2 = xdft2(1:length(D6)/2+1);
% figure;
subplot(512);plot(freq2,abs(xdft2));title('BETA');
[~,I] = max(abs(xdft2));
fprintf('Beta:Maximum occurs at %3.2f Hz.\n',freq2(I));
```

ICA ALGORITHM:

```
clear all;
close all;
clc
K=6;
N=2500;
k=1:N;


figure(1)
subplot(7,1,1);
plot(s1,'r');
ylabel('c3');
title('Input Sources Baseline');
subplot(7,1,2);
plot(s2,'b');
ylabel('c4');
subplot(7,1,3);
plot(s3,'g');
ylabel('p3');
subplot(7,1,4);
plot(s4,'m');
ylabel('p4');
subplot(7,1,5);
plot(s5,'r');
ylabel('o1');
subplot(7,1,6);
plot(s6,'b');
ylabel('o2');
% subplot(7,1,7);
% plot(s7);
% ylabel('EOG');
xlabel('Samples');


INPUT=[s1;s2;s3;s4;s5;s6];


X=zeros(K,N);
randn('seed',1);
A=0.01*randn(K);
disp('Mixing matrix');
disp(A)
```

```
%H=A*[s1;s2;s3;s4;s5;s6;s7];
H=A*[s1;s2;s3;s4;s5;s6];

for i=1:K
    H(i,:)=H(i,:)-1/N*sum(H(i,:));
end
% H


figure(2)
subplot(7,1,1);
plot(k,H(1,:),'r');
ylabel('c3');
title('Mixed Signals Baseline');
% axis([ 1 1000 -20 20 ]);
subplot(7,1,2);
plot(k,H(2,:),'b');
ylabel('c4');
%axis([ 1 1000 -20 20]);
subplot(7,1,3);
plot(k,H(3,:),'g');
% axis([ 1 1000 -20 20 ]);
ylabel('p3');
subplot(7,1,4);
plot(k,H(4,:),'m');
ylabel('p4');
% axis([ 1 1000 -20 20 ]);
subplot(7,1,5);
plot(k,H(5,:),'r');
ylabel('o1');
% axis([ 1 1000 -20 20 ]);
subplot(7,1,6);
plot(k,H(6,:),'b');
ylabel('o2');
% axis([ 1 1000 -20 20 ]);
% subplot(7,1,7);
% plot(k,H(7,:));
% % axis([ 1 1000 -20 20 ]);
xlabel('Samples');

A=double(H);
m=mean(A,2);
A=A- m(:,ones(1,size(A,2)));
covarianceMatrix=cov(A');
[y, x] = eig (covarianceMatrix);
whiteningMatrix = x^(-.5)* y';
dewhiteningMatrix = y * sqrt (x);
new_A=whiteningMatrix*A;

for i=1:6
    for j=(i+1):6
        temp=x(i,i);
        temp_2=y(:,i);
        if temp<x(j,j)
            x(i,i)=x(j,j);
            x(j,j)=temp;
            y(:,i)=y(:,j);
            y(:,j)=temp_2;
        end
    end
end
```

```
d=x
v=y
d.^(1/2);
Y=inv(d.^(1/2))*v'*H;

epsilon=0.0001;
rand('seed',1);% insert this before using rand
W=rand(K);
mu=0.01;
for p=1:K
    W(:,p)=W(:,p)/norm(W(:,p));
    exit=0;
    count=0;
    iter=1;
    while exit==0;
          count=count+1;
          temp=W(:,p);
        W(:,p)=1/N*Y*((temp'*Y).^3)'-3*temp;

              u(:,p)=temp'*Y(:,p);
              temp1 = (exp(-u(:,p))-1)./(exp(-u(:,p))+1);
              temp2 = eye(size(W))+(temp1)*u(:,p)';
              dw=temp2*W(:,p);
              W(:,p)=W(:,p)+mu*dw;

          ssum=zeros(K,1);
          for counter=1:p-1
              ssum=ssum+(W(:,p)'*W(:,counter))*W(:,counter);
          end
          W(:,p)=W(:,p)-ssum;
          W(:,p)=W(:,p)/norm(W(:,p));
          if(abs((dot(W(:,p),temp)))<1+epsilon)&(abs((dot(W(:,p),temp)))>1-
epsilon)
                        exit=1;
          end
          iter=iter+1;
   end
end
%rand('seed', 1);
out=W'*Y;
        Kurt=[W]

%MSE
MSE1=1/length(s1)*sum ((s1-out(1,:)).^2)
 MSE2=1/length(s2)*sum ((s2-out(2,:)).^2)
  MSE3=1/length(s3)*sum ((s3-out(3,:)).^2)
   MSE4=1/length(s4)*sum ((s4-out(4,:)).^2)
    MSE5=1/length(s5)*sum ((s5-out(5,:)).^2)
     MSE6=1/length(s6)*sum ((s6-out(6,:)).^2)


%RMSE
    Q1=1/length(s1)*sum((s1-out(1,:)).^2);
RMSE1=sqrt(Q1)
 Q2=1/length(s2)*sum((s2-out(2,:)).^2);
RMSE2=sqrt(Q2)
 Q3=1/length(s3)*sum((s3-out(3,:)).^2);
RMSE3=sqrt(Q3)
 Q4=1/length(s4)*sum((s4-out(4,:)).^2);
RMSE4=sqrt(Q4)
 Q5=1/length(s5)*sum((s5-out(5,:)).^2);
```

```
RMSE5=sqrt(Q5)
 Q6=1/length(s6)*sum((s6-out(6,:)).^2);
RMSE6=sqrt(Q6)

%NMSE
Io=s1;In=H(1,:);Imf=out(1,:);
NMSE1 = nmse(Io,In,Imf)
  Io=s2;In=H(2,:);Imf=out(2,:);
  NMSE2 = nmse(Io,In,Imf)
   Io=s3;In=H(3,:);Imf=out(3,:);
   NMSE3 = nmse(Io,In,Imf)
    Io=s4;In=H(4,:);Imf=out(4,:);
    NMSE4 = nmse(Io,In,Imf)
     Io=s5;In=H(5,:);Imf=out(5,:);
     NMSE5 = nmse(Io,In,Imf)
      Io=s6;In=H(6,:);Imf=out(6,:);
      NMSE6 = nmse(Io,In,Imf)

%(PSNR)Peak Signal to Noise Ratio
PSNR1=20*log10(64/RMSE1)
 PSNR2=20*log10(64/RMSE2)
  PSNR3=20*log10(64/RMSE3)
   PSNR4=20*log10(64/RMSE4)
    PSNR5=20*log10(64/RMSE5)
     PSNR6=20*log10(64/RMSE6)

%SNR MAX
[ys,WW]=SNR_Max(out(1,:));
SNR_MAX1=WW
[ys,WW]=SNR_Max(out(2,:));
SNR_MAX2=WW
[ys,WW]=SNR_Max(out(3,:));
SNR_MAX3=WW
[ys,WW]=SNR_Max(out(4,:));
SNR_MAX4=WW
[ys,WW]=SNR_Max(out(5,:));
SNR_MAX5=WW
[ys,WW]=SNR_Max(out(6,:));
SNR_MAX6=WW

% %(SNR)signal to noise ratio
% SNR1 = 10*log10(sum(s1).^2/sum((s1-out(1,:)).^2))
%  SNR2 = 10*log10(sum(s2).^2/sum((s2-out(2,:)).^2))
%   SNR3 = 10*log10(sum(s3).^2/sum((s3-out(3,:)).^2))
%    SNR4 = 10*log10(sum(s4).^2/sum((s4-out(4,:)).^2))
%     SNR5 = 10*log10(sum(s5).^2/sum((s5-out(5,:)).^2))
%      SNR6 = 10*log10(sum(s6).^2/sum((s6-out(6,:)).^2))


%(ISNR) Improved Signal to Noise Ratio
%ISNR= 10*log10(s1-H(1,:)).^2/(s1-out(1,:)).^2
mixed_norm1 = norm(s1-H(1,:)).^2;
restored_norm1 = norm(s1-out(1,:)).^2;
ISNR1= 10*log10(mixed_norm1/restored_norm1)


mixed_norm2 = norm(s2-H(2,:)).^2;
restored_norm2 = norm(s2-out(2,:)).^2;
ISNR2= 10*log10(mixed_norm2/restored_norm2)

%SIR(SIGNAL TO INTERFERENCE RATIO)
```

```
 sf_norm1= (mod(s1,2)).^2;
 sf_norm_sum1= sum(sf_norm1, 2);
 diff_sf_out1= out(1,:)-s1;
 diff_sf_out_norm1= (mod(diff_sf_out1,2)).^2;
 diff_sf_out_norm_sum1= sum(diff_sf_out_norm1,2);
 SIR1= 10*log10(sf_norm_sum1/diff_sf_out_norm_sum1)

 sf_norm2= (mod(s2,2)).^2;
 sf_norm_sum2= sum(sf_norm2, 2);
 diff_sf_out2= out(2,:)-s2;
 diff_sf_out_norm2= (mod(diff_sf_out2,2)).^2;
 diff_sf_out_norm_sum2= sum(diff_sf_out_norm2,2);
 SIR2= 10*log10(sf_norm_sum2/diff_sf_out_norm_sum2)

 sf_norm3= (mod(s3,2)).^2;
 sf_norm_sum3= sum(sf_norm3, 2);
 diff_sf_out3= out(3,:)-s3;
 diff_sf_out_norm3= (mod(diff_sf_out3,2)).^2;
 diff_sf_out_norm_sum3= sum(diff_sf_out_norm3,2);
 SIR3= 10*log10(sf_norm_sum3/diff_sf_out_norm_sum3)

 sf_norm4= (mod(s4,2)).^2;
 sf_norm_sum4= sum(sf_norm4, 2);
 diff_sf_out4= out(4,:)-s4;
 diff_sf_out_norm4= (mod(diff_sf_out4,2)).^2;
 diff_sf_out_norm_sum4= sum(diff_sf_out_norm4,2);
 SIR4= 10*log10(sf_norm_sum4/diff_sf_out_norm_sum4)

 sf_norm5= (mod(s5,2)).^2;
 sf_norm_sum5= sum(sf_norm5, 2);
 diff_sf_out5= out(5,:)-s5;
 diff_sf_out_norm5= (mod(diff_sf_out5,2)).^2;
 diff_sf_out_norm_sum5= sum(diff_sf_out_norm5,2);
 SIR1= 10*log10(sf_norm_sum5/diff_sf_out_norm_sum5)

%Frobenius Error
%norm(D-Dhat,'fro').^2/(M*N)
FrobeniusError1= norm(W-v,'fro').^2/(6*6)

%Amori Error Code

%Normalization of v
normA = v - min(v(:));
normA = normA ./ max(normA(:));
v_norm=normA ;
%Normalization of W
normA = W - min(W(:));
normA = normA ./ max(normA(:));
W_norm=normA;
```