Middle-East Journal of Scientific Research 23 (8): 1909-1920, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.08.22417

Design and Implementation of Real Time Electronic Stethoscope with a Method for Separating Heart Sound from Lung Sound Using TMS320C6713 DSK

K. Sathesh, N.J.R. Muniraj, A.V. Akshitha, Blessy K. Roy, M. Induja, M. Devi Aruna Maharasi

Department of ECE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, India, 641659

Abstract: The basic theme of this paper is to separate the heart sound form lung sound in real time application using the concept of Adaptive Line Enhancer. This model which is presented as one of the solutions is based on the cancellation of heart sound from lung sound using ALE (Adaptive Line Enhancer) with LMS (Least Mean Square) and NLMS (Normalized Least Mean Square) algorithm. Adaptive filter is the filter that includes adjustable coefficients by means of an adaptive algorithm to make filter response optimal according to the given performance criterion. The adaptive algorithm determines the variation of filter coefficients depending on the performance criterion. The adaptation algorithms used in this paper are LMS and NLMS. The ALE technique uses LMS and NLMS algorithm to enhance the quality of the lung sound by cancelling out the heart sound. The initial filter coefficients are loaded to both FIR filter and the Least Mean Square network. This output is compared with the Heart signal and the error signal is generated. The error signal is the required Lung sound. This feedback is repeated until the error becomes equal to an acceptable value. The real-time implementation of this paper is carried out with the help of digital stethoscope and TMS320C6713 DSK. The simulation is done by using Simulink in MATLAB and CCS (Code Composer Studio).

Key words: ALE · CCS · HSS and LSS · LMS and NLMS and TMS320C6713 DSK

INTRODUCTION

Digital signal processors such as the TMS320C6x (C6x) family of processors are like fast special-purpose microprocessors with a specialized type of architecture and an instruction set appropriate for signal processing. The C6x notation is used to designate a member of Texas Instruments' (TI) TMS320C6000 family of digital signal processors. The architecture of the C6x digital signal processor is very well suited for numerically intensive calculations. Based on a very-long-instruction-word (VLIW) architecture, the C6x is considered to be TI's most powerful processor. Digital signal processors are used for a wide range of applications, from Communications and controls to speech and image processing. The general-purpose digital signal processor is dominated by applications in communications (cellular). Applications embedded digital signal processors are dominated by consumer products [1]. They are found in cellular phones, fax/modems, disk drives, radio, printers, hearing aids, MP3 players, high-definition television (HDTV), digital cameras and so on. These processors have become the products of choice for a number of consumer applications, since they have become very cost-effective. They can handle different tasks, since they can be reprogrammed readily for a different application [2, 3].

Nowadays, effective communication is necessary to keep up with the fast-developing world. Effective voice communication is the most important part of it. In the prevailing environment, the noise corrupts the speech signal to such an extent, sometimes, that it is almost impossible to recover the original voice message communicated. That noise is usually given the name of background noise, which affects the intelligibility of the speech signal. To cater the issue of noise effectively we need some new schemes for optimal noise cancellation. In the past, the noise cancellation schemes using two sensors introducing the concept that a noise source is acquired by the second sensor, which cancels out the unwanted noise from the signal acquired by the first sensor, by destructively interfering with it. Using the same concept, several schemes have been proposed till

Corresponding Author: K. Sathesh, Department of ECE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, India, 641659.

now, combining the basic adaptive filtering algorithms and implementing them. Work in this regard is still in progress. The two-sensor concept has been used in this paper as well [4, 5]. A technique based on NLMS, also known as traditional Acoustic Noise Cancellation (ANC) scheme, yielded the cancellation of wideband noise from the corrupted speech signal but didn't address the issue of sinusoidal noise [6].

In real-time environment, sinusoidal noise is added in the speech signal through different sources like ceiling fan, PC fan and engine noise. Sinusoidal noise is thus a significant part of the corrupted signal and its cancellation is vital for acquiring a clean speech signal.

ALE, an existing technique, gained popularity as it gave a better understanding of the spectrum analysis and the behaviour of noise in the spectral sense. It is used to cancel out the sinusoidal part of noise in an effective manner. Thus, using the ALE and NLMS filters collectively, the purpose of cancellation of both the wideband and sinusoidal noise is accomplished. Therefore, a technique, combining the ALE and NLMS filters, proposed in [6] can be effectively implemented for various applications, like reduction of propeller-induced cabin noise during a flight removal of sinusoidal noise in ECG analysis, acoustic noise cancellation in i-phone4 and removal of car- motor sound during journey by car.

Adaptive Filters: Filtering is a signal processing operation. Its objective is to process a signal in order to manipulate the information contained in the signal. In other words, a filter is a device that maps its input signal to another output signal facilitating the extraction of the desired information contained in the input signal. A digital filter is the one that processes discrete-time signals represented in digital format. For time-invariant filters, the internal parameters and the structure of the filter are fixed and if the filter is linear the output signal is a linear function of the input signal. Once the prescribed specifications are given, the design of time-invariant linear filters entails three basic steps namely the approximation of the specification by a rational transfer function the choice of an appropriate structure defining the algorithm and the choice of the form of implementation for the algorithm.

Adaptive linear filters work on the principle that the desired signal or parameters can be extracted from the input through a filtering or estimation operation. The adaptation of the filter parameters is based on minimizing the mean square error between the filter output and a target (or desired) signal.

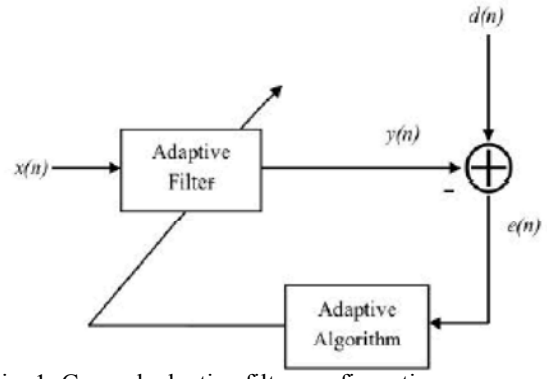

Fig. 1: General adaptive filter configuration

The use of the Least Square Estimation (LSE) criterion is equivalent to the principle of orthogonally in which at any discrete time *m* the estimator is expected to use all the available information available up to time *m*.

Adaptive Filter Configuration: The general set up of an adaptive-filtering environment is illustrated in general adaptive filter configuration Figure 1, when *n* is the iteration number, *x (n)* denotes the input signal, *y (n)* is the adaptive-filter output signal and *d (n)* defines the desired signal. The error signal *e (n)* is calculated as *d (n)-y (n)*. The error signal is then used to form a performance function that is required by the adaptation algorithm in order to determine the appropriate updating of the filter coefficients. The minimization of the objective function implies that the adaptive-filter output signal is matching the desired signal in some sense. At each sampling time, an adaptation algorithm adjusts the filter coefficients $w(n) = [w_0(n)\ w1(n) \ldots w_{N-1}(n)]$ to minimize the difference between the filter output and a desired or target signal.

The complete specification of an adaptive system, as shown in the Figure, consists of three things:

Input: The input of application is defined by the choice of the signals acquired from the environment to be the input and desired-outputs. The number of different application in which adaptive techniques are being successfully used has increased enormously during the last two decades some examples are echo cancellation, equalization of dispersive channels, system identification, signal enhancement, adaptive beam forming noise cancellation and control.

Adaptive Filter Structure: The adaptive filters can be implemented in a number of different structures or realizations. The choice of the structure can influence the

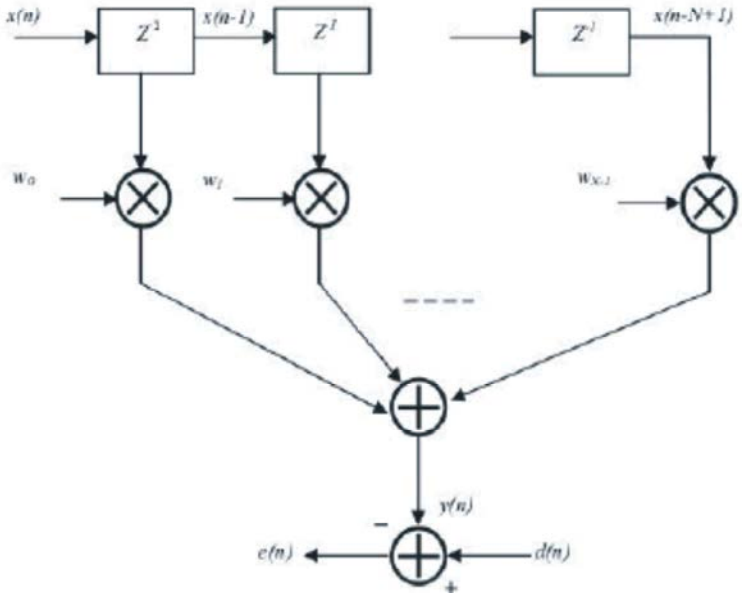

Fig. 2: Transversal FIR filter architecture

computational complexity (amount of arithmetic operations per iteration) of the process and also the necessary number of iterations to achieve a desired performance level. Here are two major classes of adaptive digital filter realization distinguished by the form of the impulse response (FIR) filter and the infinite-duration impulse response (IIR) filters. FIR filters are usually implemented with non-recursive structures, whereas filters utilize recursive realizations.

Adaptive FIR Filter Realizations: The most widely used adaptive FIR filter structure is the transversal filter, also called tapped delay line, that implements an all-zero transfer function with a canonical direct form realization without feedback. For this realization, the output signal y (n) is a linear combination of the coefficients, that yields a quadratic mean-square error (MSE=E [e (n) ^{2}]) function with a unique optimal solution. Other alternative adaptive FIR realizations are also used in order to obtain improvements as compared to the transversal filter structure, in terms of computational complexity, speed of convergence and finite word length properties.

Adaptive IIR Filter Realizations: The most widely used realizations of adaptive filter IIR filters is the canonical direct form realization, due to its simple implementation and analysis. However, there are some inherent problems related to recursive adaptive filters which are structure dependent such as pole-stability monitoring requirements and slow speed of convergence. To address these problems, different realizations were proposed attempting to overcome the limitations of the direct form structure.

Algorithm: The algorithm is the procedure used to adjust the adaptive filter coefficients in order to minimize a prescribed criterion. The algorithm is determined by defining the search method or minimization algorithm determines several crucial aspects of overall adaptive process, such as existence of sub-optimal solutions, biases optimal solution and computational complexity.

Adaptive Line Enhancer: The ALE is a special form of adaptive noise canceller that is designed to suppress the wide-band noise component of the input, while passing the narrow-band signal component with little attenuation. ALE consists of the interconnection of a delay element and a linear predictor, as is illustrated in the block diagram in Figure 3. Adaptive line enhancer (ALE) was first introduced by Windrow in 1975. T The structure of ALE in that paper is shown in Figure.3. ALE is formulated in an *L*-weight linear predictive FIR filter structure, with its output *y* (*k*) being defined as:

$$y(k) = \sum_{l=0}^{L-1} w(k)x(k-l-\Delta) \quad (1)$$

where Δ is the prediction distance of the filter in terms of the sampling interval, *L* is the filter length and *w (k)* is the ALE coefficients (adjustable FIR filter weights).

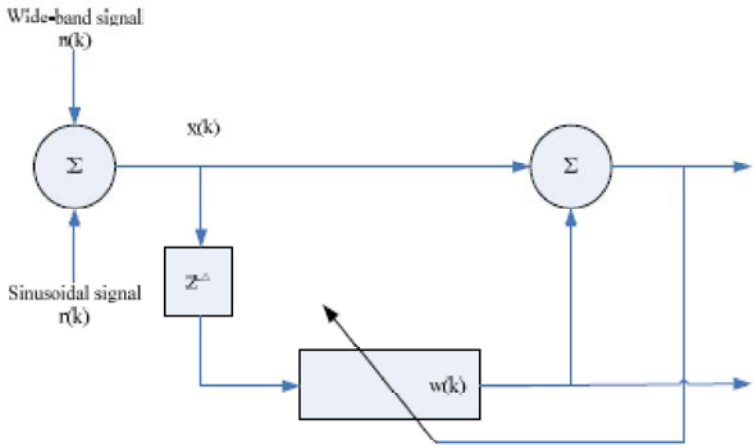

Fig. 3: Adaptive line enhancer structure

Least Mean Square (LMS) adaptive algorithm is often used to adapt the coefficients. It has been shown that ALE method has the potential to separate heart sound from lung sound with a low signal to noise ratio when applied to a one-input, artificially mixed heart-lung sound signal [7, 8].

where x(n) is the input vector of time delayed input values,and is given by

$$X(n)=[x(n)x(n-1)\ldots\ldots.x(n-N+1)]^T \quad (2)$$

$W(n)=[w_0(n)w_1(n)w_2(n)\ldots\ldots w_{N-1}(n)]^T$ represents the coefficients of the adaptive FIR filter tap weight vector at time n and μ is known as the step size parameter and is a small postive constant.ALE, an existing technique, gained popularity as it gave a better understanding of the spectrum analysis and the behaviour of noise in the spectral sense. It is used to cancel out the sinusoidal part of noise in an effective manner. Thus, using the ALE and NLMS filters collectively, the purpose of cancellation of both the wideband and sinusoidal noise is accomplished [10].

Least Mean Square (LMS) Algorithm: The least Mean Square Algorithm was first developed by windrow and Hoff in 1959 through their studies of pattern recognition. There on it has become one of the most widely used algorithms in adaptive filtering. The LMS algorithm is a type of adaptive filter known as stochastic gradient-based algorithm as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution. It is well known and widely used due to its computational simplicity. With each iteration of the LMS algorithm,the filter tap weights of the adaptive filter are updated according to the following formula

$$W(n+1)=w(n)+2\mu e(n)x(n) \quad (3)$$

The step size parameter controls the influence of the updating factor. Selection of suitable value for μ is imperative to the performance of the LMS algorithm. If the value of μ is too small, the time an adaptive filter takes to converge on the optimal solution will be too long; if the value of μ is too large the adaptive filter becomes unstable and its output diverges.

Implementation of the LMS Algorithm: For the implementation of each iteration of the LMS algorithm requires three distinct steps in the following order:

- The output of the filter(n) is calculated using Eq.(4)

$$y(n)=\sum_{i=0}^{N-1} w(n)x(n-i) = wT(n)x(n) \quad (4)$$

- The value of the error estimation is calculated using Eq.(5)

$$e(n)=d(n)-y(n) \quad (5)$$

- The tap weights of the FIR vector are updated in preparation for the next iteration by Eq.(6)

$$W(n+1)=w(n)+2\mu e(n)x(n) \quad (6)$$

The main reason for the popularity of LMS algorithm in adaptive filtering is its computational simplicity that makes its implementation easier than all other commonly

used adaptive algorithms. For each iteration, the LMs algorithm requires 2N additions and 2N+1 multiplications (N for calculating the output, y(n), one for 2μe(n) and an additional N for the scalar by vector multiplication).

Normalized Least Mean Square (NLMS) Algorithm: In the standard LMS algorithm when the convergence factor μ is large, the algorithm experiences a gradient noise amplification problem. In order to solve this difficulty we can use the NLMS algorithm. The correction applied to the weight vector w(n) at iteration n+1 is 'normalized' with respect to the squared Euclidian norm of the input vector x(n) at iteration n. We may view the NLMS algorithm as a time-varying step-size an algorithm, calculating the convergence factor μ as in the Eqn (7)

$$\mu(n) = \frac{\alpha}{c+|x(n)|^2} \tag{7}$$

where α is the NLMS adaption constant, which optimize the convergence rate of the algorithm and should satisfy the condition 0<α<2 and *c* is the constant term for normalization and is always less than 1. Hence, the NLMS is a manifestation of the principle of minimum disturbance.

Implementation of the NLMS Algorithm: The NLMS algorithm is implemented in MATLAB. It is essentially an improvement over LMS algorithm with the added calculation of step size parameter for each iteration,

- The output of the adaptive filter is calculated as:

$$y(n)=\sum_{i=0}^{N-1} w(n)x(n-i) = wT(n)x(n) \tag{8}$$

- The error signal is calculated as the difference between the desired output and the filter output given by:

$$e(n)=d(n)-y(n) \tag{9}$$

- The step size and filter tap weight vectors are updated using the following equations in preparation for the next iteration:
  For I=0,1,2……N-1

$$\mu_i(n) = \frac{\alpha}{c+|x_i(n)|^2} \tag{10}$$

$$w(n+1)=w(n)+\mu_i(n)e(n)x_i(n) \tag{11}$$

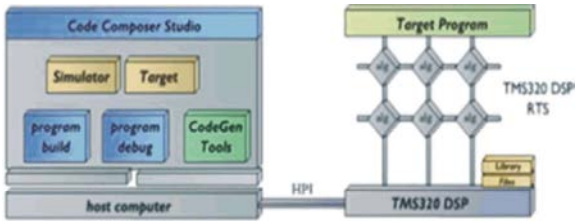

Fig. 4: Code compose studio platform

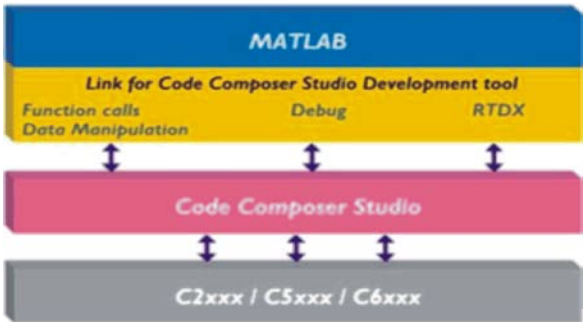

Fig. 5: MATLAB interfacing with CCS and DSP processor

where α is the NLMS adaption constant and *c* is the constant term for normalization. with α=0.02 and c=0.001,each iteration of the NLMS algorithm requires 3N+1 multiplication operations.

Code Composer Studio as Integrated Development Environment: Code Composer Studio is the DSP industry's first fully integrated development environment (IDE) with DSP-specific functionality. With a familiar environment like MS-based C++TM; Code Composer lets you edit, build, debug, profile and manage projects from a single unified environment. Other unique features include graphical signal analysis, injection/ extraction of data signal via file I/O, multi processor debugging, automated testing and customization via a C-interpretive scripting language and much more.

Real- time analysis can be performed using real time data exchange(RTDX). RTDX allows for data exchange between the host PC and target DSK, as well as analysis in real time without stopping the target. Key statistics and performance can be monitored in real time [11]. Through the joint team action group (JTAG), communication with on-chip emulation support occurs to control and monitor program execution. The C6713DSK board includes a JTAG interface through the USB port.

MATLAB Interfacing with CCS and DSP Processor: Figure 5 depicts how the MATLAB is used as an interface for calling code composer studio (CCS) and then the

Fig. 6: Real-time experimental setup using DSP processor

program is loaded into the TI instrument target. First of all MATLAB code for the desired algorithm is written and then simulated for obtaining the results in MATLAB graph window. However, if we want that the code written for MATLAB or the designed simulink model can be loaded into the TI target then we can also perform some real time results depending upon the used algorithm[12].

Real –Time Experimental Setup Using DSP Processor: The basic experimental setup for the hardware implementation of adaptive line enhancer. The real image of the setup is shown in Figureure1. The input signal can be provided to the DSP processor either through the double side probe on MIC-IN or LINE-IN port respectively. The development software, code composer studio and the stimulation software MATLAB version 2007a are installed on the PC and are used for coding the algorithm and linking the coded algorithm to the target processor. The input signal reaches to processor in digital form after the conversion by AIC23 on board codec. The C code is generated using real time workshop available in MATLAB and simulink and loaded on the DSP processor. The input signal is processed according to the code loaded into the processor memory. For interactive feedback 4 DIP switches can be used. When the signal processing is completed the output can be taken at HP OUT or LINE OUT port with the help of headphone [13].

The real time implementation can be done in the following manner: first of all, the simulink model is developed for NLMS algorithm and then connected to the CCS by RTDX link to create a project in the CCS as shown in Figure 7.

When the link get established the CCS opens, the project is created automatically and the process of code generation started. After code generation the code is compiled and during the compilation process compiler and debugger check the generated code for the errors. If there is any error code compile operation gets fail and gives the information about the error. Further these errors can be rectified and compilation process continues. When the compilation gets over the project is rebuild to generate. out executable file which is to be loaded on the target processor. Once the executable file is loaded in the processor memory, the processor is ready to use. Now the input can be applied to the processor through the line-in port using a double sided audio probe and the output can be taken from line-out port using headphones.

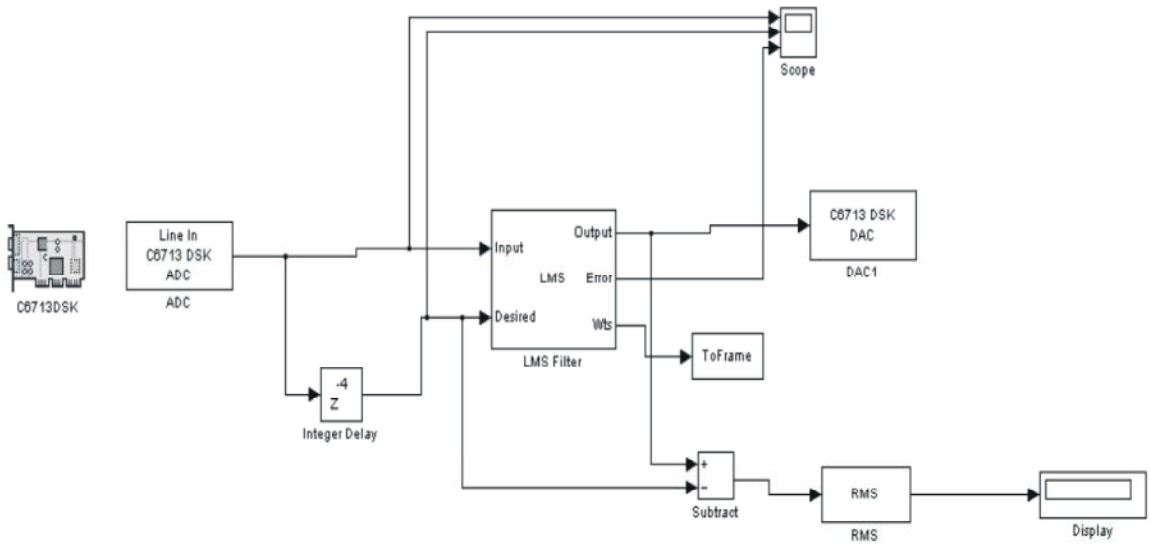

Fig. 7: Model building using RTW

RESULT AND DISCUSSIONS

The results are arranged in two sections; the first section deals with the MATLAB simulation of LMS, NLMS adaptive filter algorithms when the combination of lung and heart sound is applied as an input signal. A fair performance comparison of simulation results for two algorithm is also presented in terms of mean square errors. The next section shows the hardware implementations results for NLMS and LMS algorithms when implemented on TMS320C6713 DSP processor. Primarily the filtering is done for an input signal. The effect of frequency and amplitude variation of a input signal on the filter performance is also investigated. Further the filtering is done when an input signal is taken as an input to make the designed system more practical and implementable.

MATLAB Simulation Results for Adaptive Algorithms

LMS Algorithm Simulation Results:

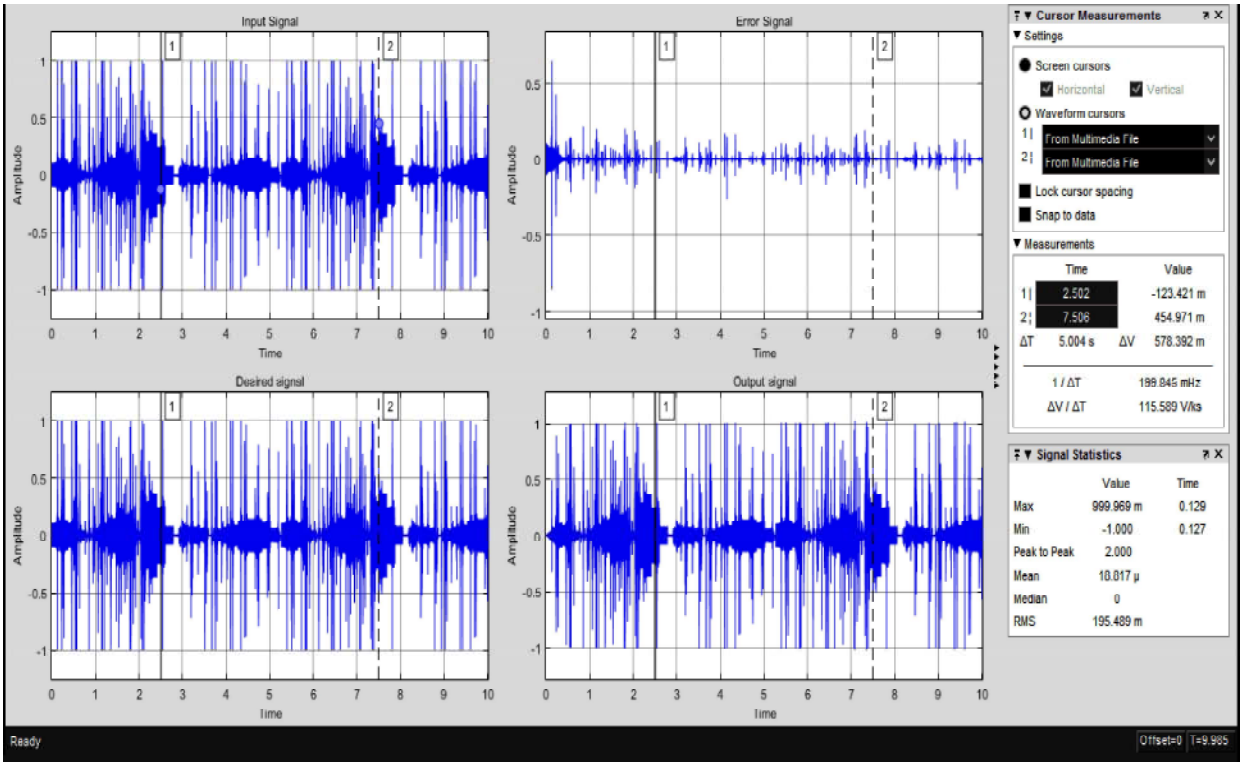

Fig. 8: Input, Desired, Error, Output signal for N=8

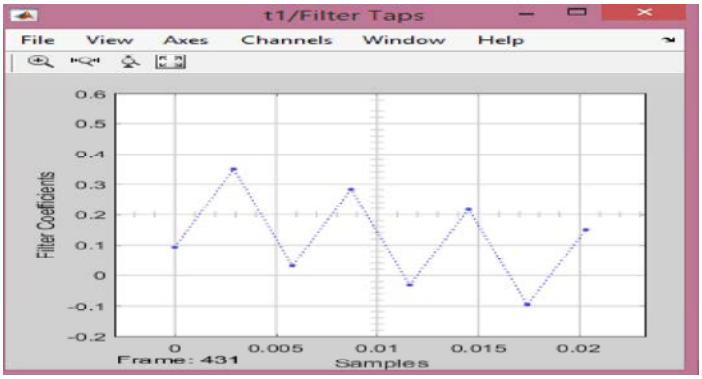

Fig. 9: LMS Filter tap for N=8

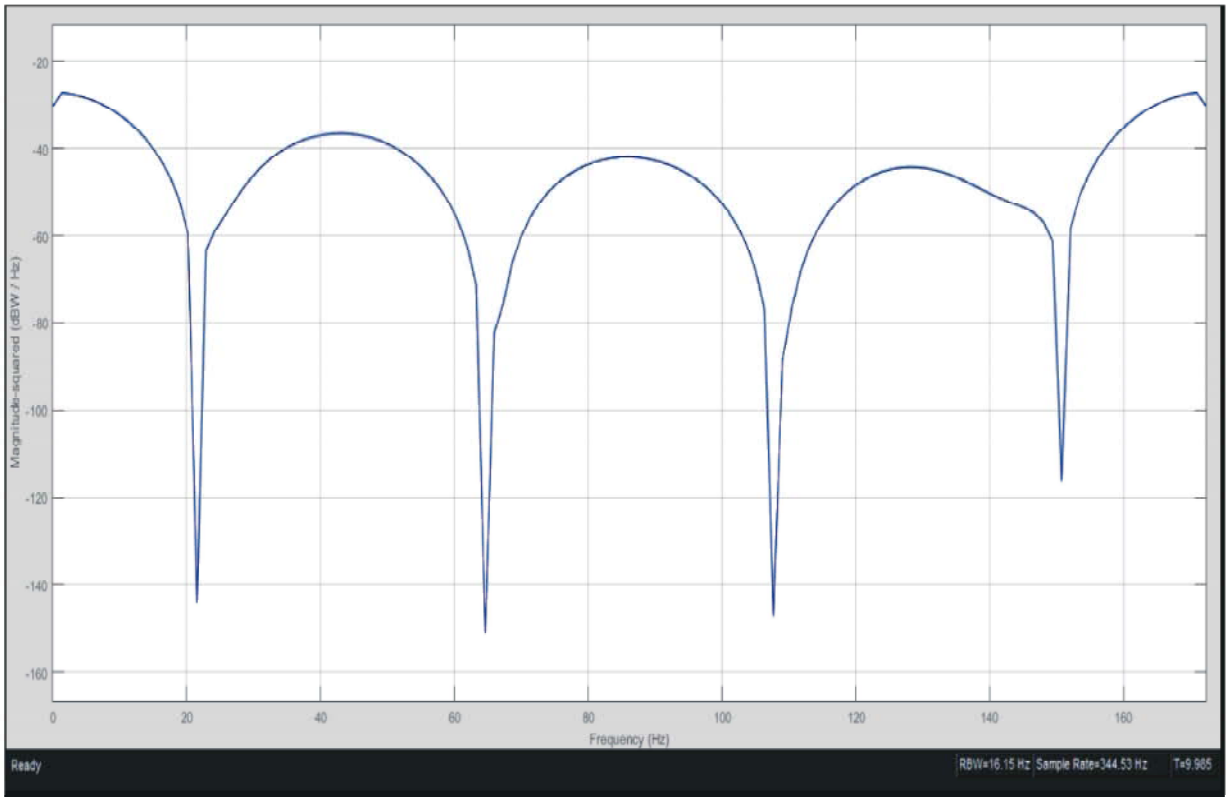

Fig. 10: LMS Frequency response for N=8

NLMS Algorithm Simulation Results:

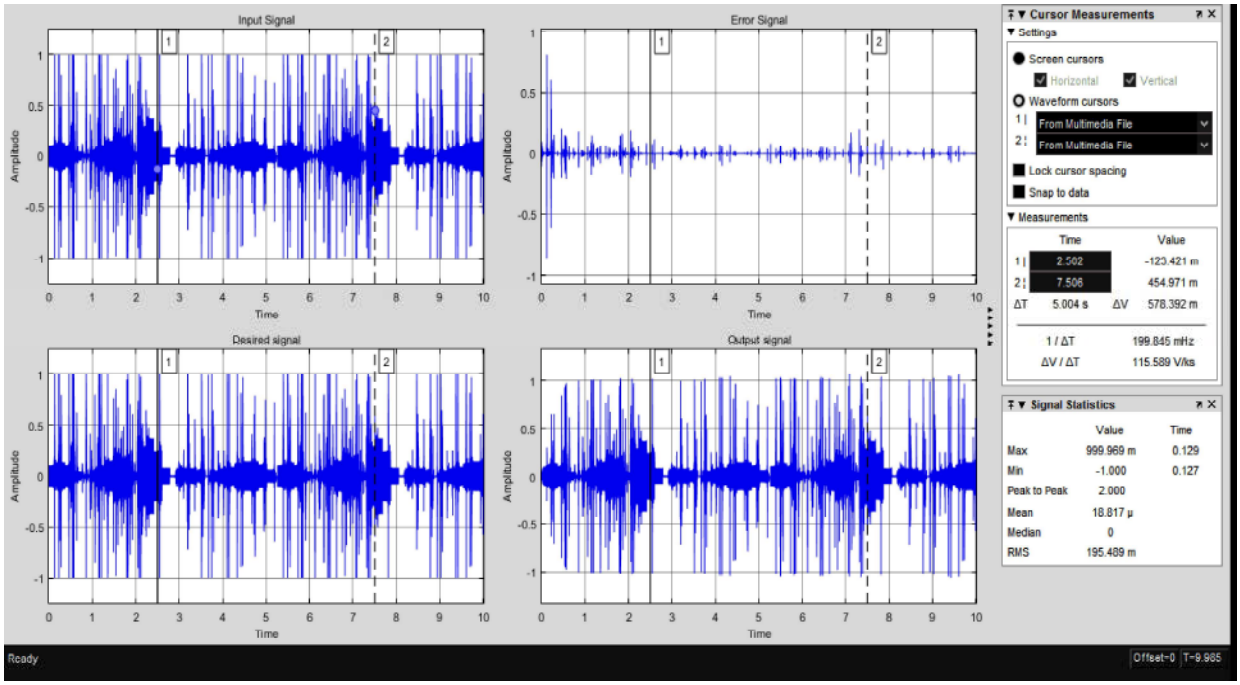

Fig. 11: Input, Desired, Error, Output signal for N=8

Fig. 12: NLMS Filter tap for N=8

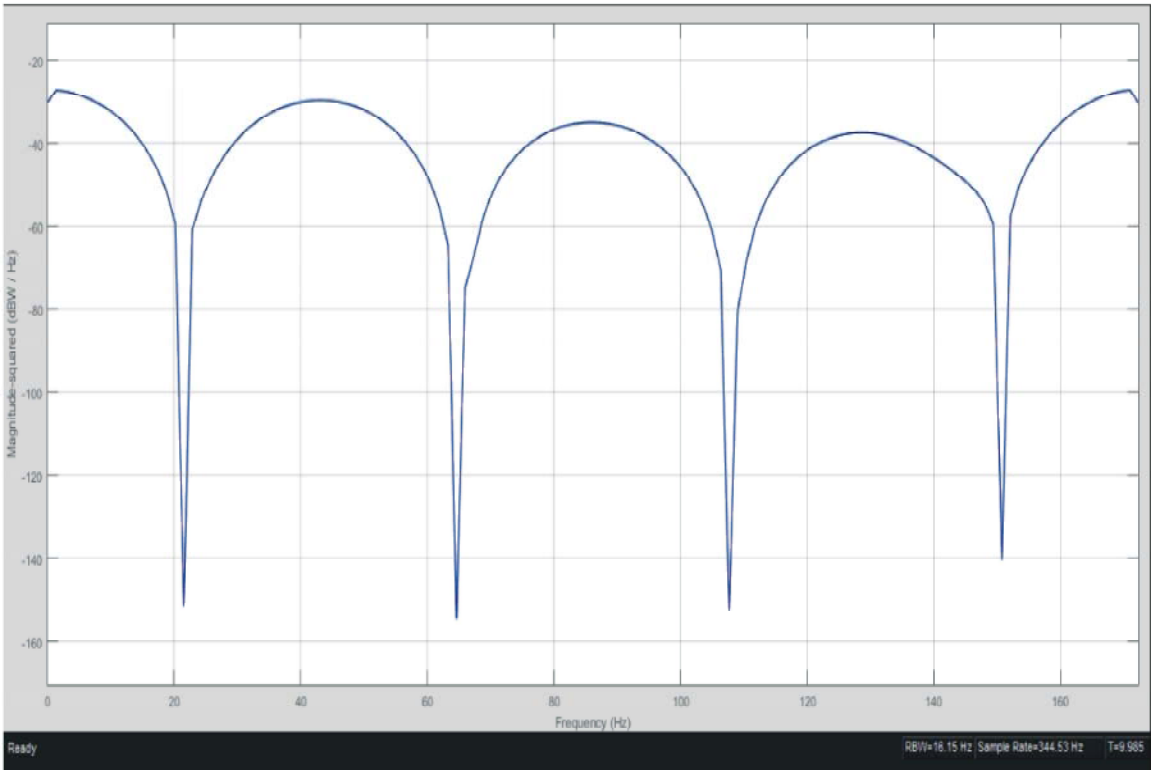

Fig. 13: NLMS Frequency response for N=8

Figure 8-13 shows the results of both LMS and NLMS algorithms, In case of NLMS algorithm step size is not fixed, it varies at each iteration on the basis of input signal energy. Therefore the filter performance improves and less time is required as compared to LMS algorithm.

Performance Comparison of Adaptive Algorithm: The following are the MSE, SNR values with different step size and filter length which were obtained during analysis as shown in the Table 5.1 -5.10.

Table 5.1: MSE, SNR and step size for filter length=2

Step size	MSE	SNR
0	0.212	1
1E-04	0.03953	28.75
2E-04	0.007402	820
0.001	1.244E-08	2.903E+14
0.005	3.179E+15	4.445E+27
0.01	1.821E-15	1.355E+28
0.05	4.367E-16	2.356E+29
0.1	1.694E-21	1.565E+40

Table 5.2: MSE, SNR and step size for filter length=4

Step size	MSE	SNR
1E-04	0.008959	559.7
2E-04	0.005006	1792
0.001	0.004777	1968
0.005	0.003797	3116
0.01	0.002867	5467
0.05	0.0003876	299000
0.1	6.673E-15	10090000

Table 5.3: MSE, SNR and step size for filter length=8

Step size	MSE	SNR
0	0.0212	1
1E-04	0.01433	218.7
2E-04	0.01355	244.6
0.001	0.008643	601.4
0.005	0.0009464	50160
0.01	0.0001083	3827000
0.05	0.0001247	288800000
0.1	0.0001585	1788000

Table 5.4: MSE, SNR and step size for filter length=16

Step size	MSE	SNR
0	0.212	1
1E-04	0.02239	89.64
2E-04	0.01432	219.1
0.001	0.001419	22320
0.005	0.001232	29600
0.01	0.00119	31700
0.05	0.0009451	50300
0.1	0.0007101	89100

Table 5.5: MSE, SNR and Step size for Filter length=32

Step size	MSE	SNR
0	0.212	1
1E-04	0.009591	488.4
2E-04	0.007759	746.2
0.001	0.004486	2232
0.005	0.002022	10990
0.01	0.001545	18830
0.05	0.0009982	45090

MSE, SNR and Step Size for LMS filter

Table 5.6: MSE, SNR and step size for filter length=2

Step size	MSE	SNR
0	0.212	1
0.0001	0.00007895	7270000
0.0002	4.03E-08	2.766E+13
0.001	1.314E-14	2.602E+26
0.005	2.284E-15	8.611E+27
0.01	9.987E-16	4.504E+28
0.05	2.61E-16	6.593E+29
0.1	1.044E-16	4.12E+30

Table 5.7: MSE, SNR and step size for filter length=4

Step size	MSE	SNR
0	0.212	1
0.0001	0.004619	2106
0.0002	0.004223	2519
0.001	0.002063	10550
0.005	0.00008596	6079000
0.01	0.0000574	13630000
0.05	0.00002695	61870000
0.1	9.704E-06	477000000

Table 5.8: MSE, SNR and step size for filter length=8

Step size	MSE	SNR
0	0.212	1
0.0001	0.01106	367.2
0.0002	0.008085	687.3
0.001	0.000671	99780
0.005	0.0000948	4999000
0.01	0.0001127	3539000
0.05	0.0001409	2264000
0.1	0.0001004	4455000

Table 5.9: MSE, SNR and step size for filter length=16

Step size	MSE	SNR
0	0.212	1
0.0001	0.01509	197.2
0.0002	0.006639	1019
0.001	0.001332	25330
0.005	0.0009646	48280
0.01	0.0007136	88230
0.05	0.0001262	2821000
0.1	0.00007793	7396000

Table 5.10: MSE, SNR and step size for filter length=32

Step size	MSE	SNR
0	0.212	1
0.0001	0.01502	199.1
0.0002	0.008282	654.9
0.001	0.006318	3432
0.005	0.0009209	52980
0.01	0.0007364	82850
0.05	0.0002823	563800
0.1	0.0001849	1314000

MSE, SNR and Step Size for NLMS Filter: By comparison LMS with an average MSE of 0.008643, NLMS with an average MSE of 0.000671 is obtained. Thus by the following comparison it is proven that NLMS has better performance than LMS.

The filter order also effects the performance of a noise cancellation system. The MSE changes as we change the filter order. When the filter order is less (<15), the LMS has good MSE as compared to NLMS.

Table 5.11: MSE, SNR with Step size=0.001

Filter length	MSE(LMS)	SNR(LMS)	MSE(NLMS)	SNR(NLMS)
2	1.244E-08	2.903E+14	1.314E-14	2.602E+26
4	0.004777	1968	0.002063	1.055E+04
8	0.008643	601.4	0.000671	9.978E+4
16	0.001419	2.232E+04	0.001332	2.533E+04
32	0.004486	2232	0.003618	3432

For proper filtering filter order should be higher but as we increase the filter order the convergence speed of the filter gets slower, therefore a proper selection of filter order and suitable adaptive algorithm is imperative to the performance of the system.

Table 5.11 shows the performance analysis of adaptive algorithm in terms of MSE with optimum step size of 0.001. The result shows that the NLMS algorithm has only N number of additional multiplications as compared to LMS algorithm for better filtering and is stable. Therefore NLMS is the favourable choice for most of the industries, medical and practical applications.

Hardware Implementation Results Using TMSD320C6713 DSP Processor: The experimental setup for real-time noise cancellation is depicted in chapter 4 (Figure4.9). The model is created in Simulink and is connected to the TMS320C6713 processor using real-time workshop. The model is tested with input signal. The output results are verified with the help of headphones/speaker or CRO.

Real Time Implementation Results with LMS Algorithm:

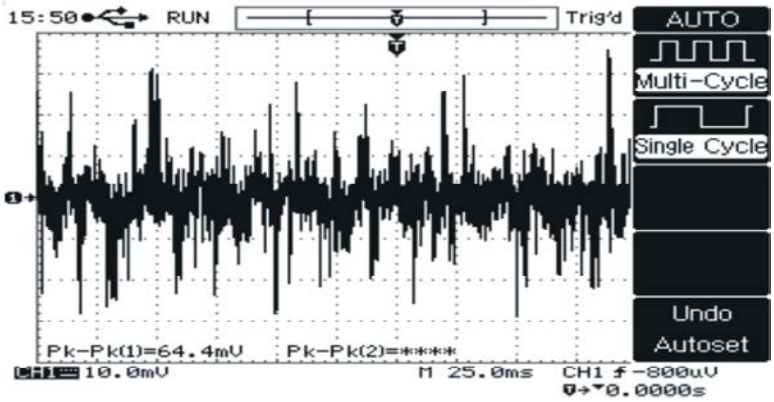

Fig. 14: Output for N=8

Real Time Implementation Results with NLMS Algorithm:

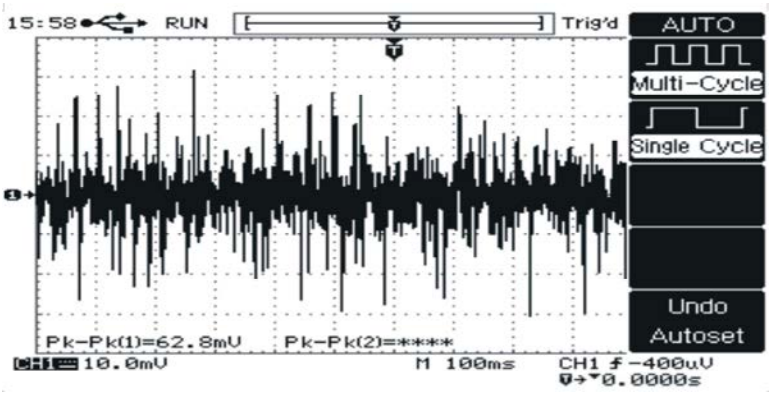

Fig. 15: Output for N=8

CONCLUSION

This paper presents the real-time implementation of heart and lung sound separation using MATLAB R2007a Simulink and CCS(Code Composer Studio v3.1). The real time signal is processed with the help of TMS320C6713 DSK. The ALE uses both LMS and NLMS algorithm to enhance the convergence rate. The adaptation algorithm

adaptively updating μ value according to the given signal has been developed. The performance of the NLMS algorithm is found to be better when compared to LMS algorithm. The MSE and SNR for various step size and filter length have been calculated and analyzed for both LMS and NLMS algorithm.The MSE, SNR for LMS and NLMS are compared and the step-size (0.001) value, filter length=8 are chosen to be the optimum performance criterion. With the help of DSO(Digital Storage Oscilloscope), the real time signal can be viewed. NLMS proves to be better and optimum method than LMS algorithm.

REFERENCES

1. TandonAbhishek and M. Omair Ahmad, 2004. An efficient, low complexity,normalized LMS algorithm for echo cancellation, The 2nd Annual IEEE Northeast Workshop on Circuits and Systems, NEWCAS, 161-164.
2. Riyanto Bambang,2007. Real-time DSP Implementation of Active Noise Control for Broadband Noise Using Adaptive LMS Filter Algorithm, in Proceedings of the International Conference on Electrical Engineering and Informatics Institute Teknologi Bandung, Indonesia, pp: 718-722.
3. Chun-Tang Chao, NopadonManeetien and Chi-Jo Wang, 2005.Construction of an Electronic Stethoscope with Real-Time Heart Sound De-Noising Feature.
4. Cristina Gabriela Saracin, Marin Saracin, Mihai Dascalu and Ana-Maria Lepar, 2009. Echo Cancellation Using The LMS Algorithm, in U.P.B. Sci. Bull., SeriesC, 71(4): 167-174.
5. Saxena, Gaurav, SubramaniamGanesan and Manohar Das, 2008. Real-time Implementation of Adaptive Noise Cancellation, in IEEE International Conference on Electro/Information Technology, pp: 431-436.
6. Jafar Ramadhan Mohammed, 2007. A New Simple Adaptive Noise CancellationScheme Based on ALE and NLMS Filter, Fifth Annual Conference on Communication Networks and Services Research (CNSR'07), IEEE.
7. Hadjileontiadis Leontios J. and Stavros M. Panas, 1997. A wavelet- based reduction of heart sound noise from lung sounds, IEEE Tran. On Biomedical Engineering, 44(7).
8. Pourazad M.T, Z.Moussavi, G.Thomas, Med Biol Eng Comput, 2006. Heart sound cancellation from lung sound recordings using time-frequency filtering, 44: 216-225.
9. Sathesh, K. and N.J.R. Muniraj, 2014. Real Time Heart And Lung Sound Separation Using Adaptive Line Enhancer With NLMS, Journal of Theoretical and Applied Information Technology, 65(2).
10. Sasaoka, N., K. Sumi, Y. Itoh and K. Fujji, 2005. A New Noise Reduction System Based On ALE and Noise Reconstruction Filter, Proc. 2005 ISCAS, pp: 272-275.
11. Tsalaile, T. and S. Sanei, 2007. Separation of heart sound signal from lung sound signal by adaptive line enhancement, in proc. Int.Conf. EUSIPCO.
12. Wolfgang Fohl, Jörn Matthies and Bernd Schwarz, 2009. A FPGA-based adaptive noise cancelling system, Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09), Como, Italy.
13. Yaser Ahmed, Farooq Jawed, Sohaib Zia and Muhammad Sheraz Aga, 2000. Real-time implementation of adaptive channel Equalization algorithms on tms320c6x dsp processors.