Universal Journal of Computer Science and Engineering Technology
1 (2), 93-98, Nov. 2010.
© 2010 UniCSE, ISSN: 2219-2158

Adaptive System Simulation and Noise Analysis Toolbox (ASSNAT) : The Open-Source Toolbox Developed with Newer Features for Adaptive System Simulation

Ajoy Kumar Dey, Susmita Saha
Department of Signal Processing
Blekinge Tekniska Hogskola (BTH)
Karlskrona, Sweden
e-mail: dey.ajoykumar@yahoo.com

Avijit Saha, Shibani Ghosh
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
e-mail: avijit003@gmail.com

***Abstract*—This paper introduces Adaptive System Simulation and Noise Analysis Toolbox (ASSNAT) version 1.1 (v1.1), which is an open- source MATLAB based software package for simulation and analysis of Adaptive signal processing systems and noises, with its new feature Learning curve method, where we can make an advanced level comparative study based analysis. ASSNAT v1.1 contains a variety of adaptive systems, filter algorithm and a wide range of input signals for simulation in a user friendly graphical interface. Central and advanced features, underlying models and algorithms, and case studies are presented in this paper to demonstrate the capabilities of this toolbox and its suitability for educational and research purposes.**

Keywords- adaptive systems; simulation; open-source; matlab; toolbox; algorithm; learning curve; ASSNAT v1.1

I. INTRODUCTION

The primary task of signal processing is to extract information from a signal or modify a signal according to the needs of the application. There are sorts of devices that perform these kinds of tasks, they can be physical hardware devices or software codes or combination of both. Adaptive systems are best-suited in these signal processing applications [2]. Adaptive systems are devices that adjust themselves to an ever-changing environment; the structure of an adaptive system changes in such a way that its performance improves through a continuing interaction with its surroundings. Its superior performance in non stationary environments results from its ability to track slow variations in the statistics of the signals and to continually seek optimal designs [3]. Some notable applications, in areas ranging from biomedical engineering [4] to wireless communications [5], include the suppression of interference arising from noisy measurement sensors, the elimination of distortions introduced when signals travel through transmission channels, and the recovery of signals embedded in a multitude of echoes created by multipath effects in mobile communications [6]. Keeping in mind this vast importance of adaptive systems in real-life applications, a software package that can simulate a multitude of adaptive systems using various algorithms, should come as a very handy tool for students, researchers and professionals equally. Furthermore, if such a software package is open-source, it should further facilitate research and development in this sector.

This paper describes a new open-source MATLAB-based software package named Adaptive System Simulation and Noise Analysis Toolbox (ASSNAT) [1] and its fundamental features as well as the newer features included in its latest version (Version 1.1). Among a variety of high level scientific languages, MATLAB was chosen as the platform for this toolbox because of its robust matrix-oriented calculations, excellent plotting capabilities and simplified graphical environment. ASSNAT v1.1 provides the opportunity to simulate a variety of real-life adaptive system applications, using different adaptive algorithms and wide range of input signal possibilities [1].

The paper organization is as follows: Section II describes the core features of ASSNAT v1.1 and gives a brief outlook of its Graphical User Interface (GUI). Section III discusses the advanced features in the new version of ASSNAT. Section IV discusses the algorithms and models of the various systems that can be simulated by this software. Section V shows case studies of simulations done by ASSNAT v1.1 and their implications. Finally, Section VI presents a conclusion and an indication towards the future scope of this work.

II. FEATURES OF ASSNAT

A. Outlines

ASSNAT v1.1 is a free, open-source and portable software toolbox run in the platform of MATLAB. Though, MATLAB is not really an open-source software, but, in our toolbox, we've tried to keep the content as much open-source as possible. Except a few GUI related functions used from MATLAB library, all other attributes of the toolbox have been directly programmed by us, the codes of which is freely available with the toolbox package. To the best of authors' knowledge, ASSNAT is the first free software project in the field of adaptive signal processing.

Corresponding Author: Avijit Saha, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

In this software package, we've created a platform for simulating different adaptive systems using different adaptive algorithms for a wide range of input signals, all in a single user interface. For clarity, let us consider an adaptive system application shown in Fig. 1. It is a generic schematic of any adaptive system. The actual block diagram of an adaptive system may vary from application to application, but almost all of them include the blocks shown in the diagram of Fig. 1. The input signal can be any type of stationary or non-stationary signal. In ASSNAT v1.1, both types of signal can be generated for simulation. The parameters that characterize the shape, size, style, and duration of the signal, are all user definable. Besides, there is also option for importing a signal data from an external source, e.g. file.

The comparison and error signal generation method depends on the actual application selected. In our software, we've provided options for three types of system applications, all of which are very realistic problems frequently faced in adaptive systems [7]. These applications are:

a) System identification
b) Adaptive Line Enhancement
c) Adaptive Noise Cancellation

For calculation of required filter coefficients necessary for the application, different adaptive algorithms can be used. In ASSNAT, the following five algorithms can be implemented:

a) Least Mean Square (LMS)
b) Leaky Least Mean Square (LLMS)
c) Normalized Least Mean Square (NLMS)
d) Recursive Least Square (RLS)
e) Fast Transversal Filter (FTF)

The required parameters for the implementation of any of these algorithms can be specified in the software. After the specification of the system, the adaptive algorithm to be used, and the input signal, the system can be simulated to produce the output signal. Both the output signal and the intermediate error signal can be viewed in the user interface. Determination of the output of a system for a given input is one of the primary applications of this software. For particular applications, the error signal represents the noise present in the input signal; hence, this software can also be used as a noise analyzer. The simulation time and convergence depends on the adaptive algorithm selected. So, this software can also be used as a tool for comparison of different algorithms to find their relative efficacy in a particular application. To summarize, ASSNAT v1.1 can serve as a multi-purpose software toolbox for simulation of adaptive systems.

Next, in this section, we'll discuss about the user interface and features of this software. A brief discussion of the algorithms and models of system applications will be done in a later section.

B. Getting Started and Main Graphical User Interface

ASSNAT can be launched from the MATLAB command prompt by typing:

```
>> ASSNAT
```

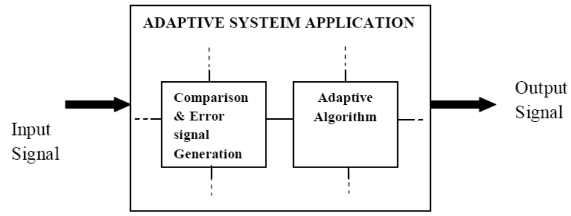


Figure 1. A generic schematic of an Adaptive System Application.

This will initialize the software and open the main Graphical User Interface window which looks like Fig. 2. From this window, all the operations of the software can be managed. Different sections of this window are explained below:

1- In this box, the original output is shown which is generated for the given input using the desired algorithm.
2- In this box, the estimated output is shown.
3- Error signal can be seen in this box.
4- This panel is for the selection of the desired adaptive algorithm and its necessary parameters. If user does not provide a value for a parameter, a default value for that parameter is used.
5- This is the main 'Activation panel' which contains 3 push buttons. RUN button starts the simulation and shows the output, CLEAR button clears the previous output and makes the window ready for the next simulation, and ZOOM button is for zooming the output according to user's convenience.
6- In this box, the length of the signal can be specified.
7- This panel is for the selection of the desired application and the input signal as well.
8- From this panel a user can view the actual system model of a particular application.
9- This panel, named as "Algorithm Analysis" enables the user to view the codes of the algorithms used.
10- From this box, information about the software toolbox and its developers can be obtained.
11- This box "Learning Curves" opens the new graphical user interface, where we find the learning curves for individual adaptive algorithms.

The GUI has been intentionally kept as simple and easy-to-navigate as possible. It has been developed using the GUI library of MATLAB.

III. ADVANCED FEATURES OF ASSNAT V1.1

A new advanced feature "Learning Curves" has been added in ASSNAT v1.1 to do the comparative study based analysis between the adaptive algorithms through their learning curves. In this method, the signal analysis is easier and more user friendly compare to the version 1.0. In this new feature, we use the existing adaptive algorithms, which we have used in the main GUI of ASSNAT, and some new added adaptive algorithms. The user can easily enter the new features from the main GUI of ASSNAT v1.1 by clicking the button of "Learning Curves". It initiates a new GUI window that looks like Fig 3.

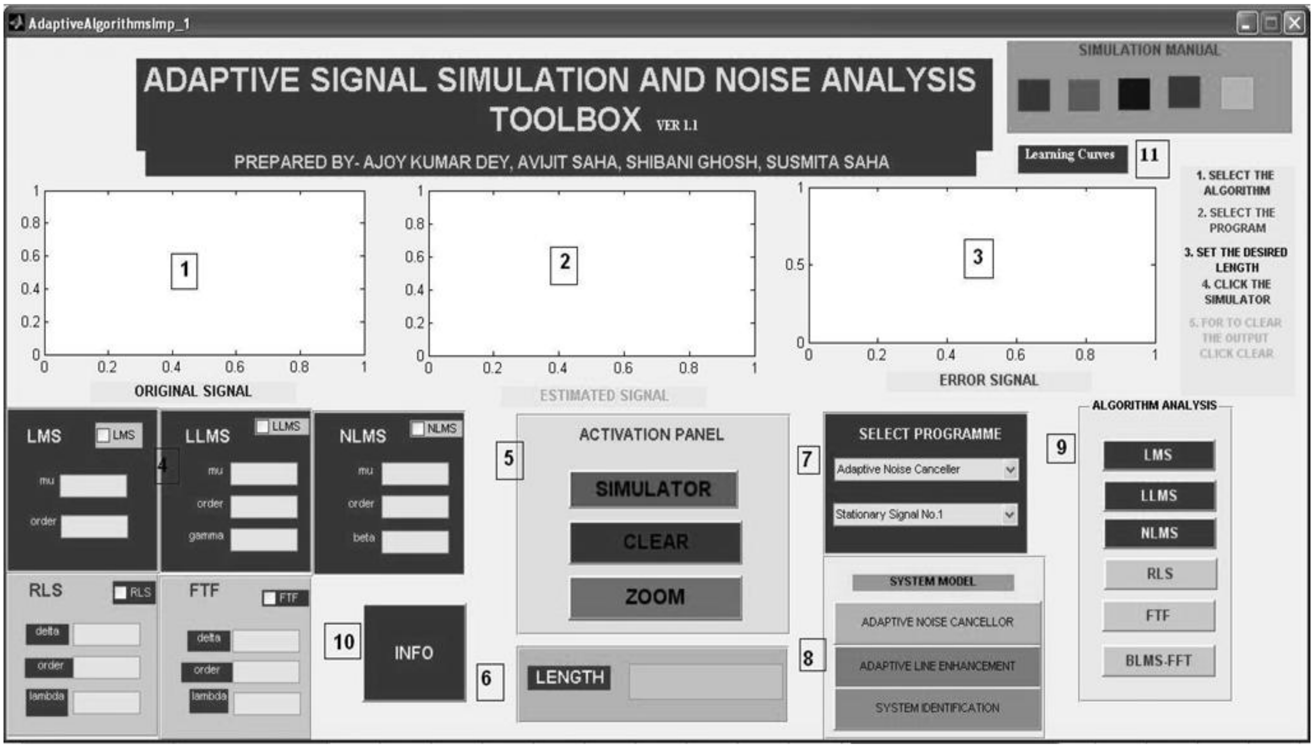


Figure 2. Graphical User Interface of ASSNAT.

This window actuates the advanced level implementations where the user can specify the input signal and no. of points as well. And user can find three kinds of observation regarding every particular adaptive algorithm, like Learning Curve, Filter Coefficients vs. No. of Iterations and Convergence function. User can chose the multi algorithm mode to find out the comparison between algorithms. In every observation user can change the filter order according to desire. And the user can frequently change the coefficients like Lambda, mu, gamma, beta as well to find out the exact output relevant to his/her work.

IV. Algorithms and Models

A. Used Adaptive Algorithms

As mentioned in the previous section, ASSNAT provides the facility of using five algorithms for adaptive signal analysis. These algorithms are briefly discussed here. For a more detailed review of the algorithms discussed, one may refer to [2] and [3].

1) Least Mean Square (LMS): LMS algorithm is a stochastic gradient algorithm, which means that the gradient of the error performance surface with respect to the free parameter vector changes randomly from one iteration to the next. This stochastic nature combined with the non linear feedback, is responsible for making a detailed convergence analysis of the LMS algorithm. LMS incorporates an iterative procedure that makes successive corrections to the weight vector in the direction of the negative of the gradient vector which eventually leads to the minimum mean square error. The weight vector equation assumes the simple form of (1).

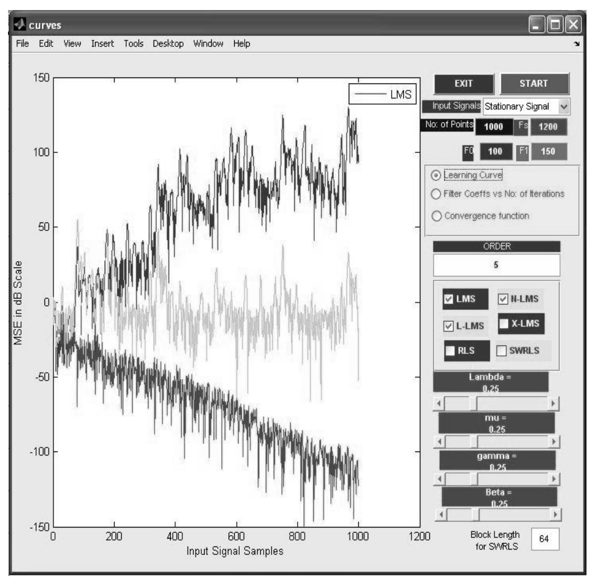


Figure 3. Learing Curves of ASSNAT VER 1.1

$$w_{n+1} = w_n + \mu e(n) x^*(n) \tag{1}$$

Where, w = weight vector (subscript denoting the coefficient number), x = input signal (* denoting conjugate of the signal), e = error signal (defined by difference between desired signal and output signal) and μ = LMS coefficient.

Update for k-th coefficient requires only one multiplication and one addition, making the algorithm simple and robust. The output signal is simply the multiplication of weight vector with input signal, given by (2).

$$y_n = w^h x(n) \tag{2}$$

The LMS algorithm initiated with some arbitrary value for the weight vector is seen to converge and stay stable for

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{3}$$

Where, λ_{max} is the largest eigenvalue of the correlation matrix R. One of the literatures [8] also provides an upper bound for μ based on several approximations as μ being less than or equal to 1/(3trace(R)).

2) Leaky Least Mean Square (LLMS): The LLMS algorithm avoids the case of instability of modes in LMS adaptive filter by introducing a leakage coefficient γ into the LMS algorithm as shown in (4).

$$w_{n+1} = (1 - \mu\gamma)w_n + \mu e(n)x^*(n);\ \ 0 < \gamma \ll 1 \tag{4}$$

Where, all other symbols bear the same meaning as in (1).

The effect of this leakage coefficient is to force the filter coefficients to zero if either the error *e(n)* or the input *x(n)* becomes zero, and to force any un-damped modes of the system to zero. The step size μ for convergence of the leaky LMS algorithm in the mean becomes

$$0 < \mu < \frac{2}{\lambda_{max} + \gamma} \tag{5}$$

3) Normalized Least Mean Square (NLMS): The NLMS filter is a variant of the LMS algorithm that normalizes the LMS coefficient with the power of the input. The algorithm is given by (6).

$$w_{n+1} = w_n + \beta \frac{x^*(n)}{|x(n)|^2} e(n) \tag{6}$$

Where β is normalized step size and $|x(n)|^2$ is the power of the input signal. β is related to μ by the relation:

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

The condition for the mean square convergence is $0<\beta<2$.

4) Recursive Least Square (RLS): The recursive least squares algorithm utilizes continuously updated estimates of the autocorrelation matrix of the input signal vector and the cross correlation vector between the input and the desired response, which go back to the beginning of the adaptive process. The weight updated equation for the Recursive Least Squares algorithm is:

$$w_n = w_{n-1} + \alpha(n)x(n) \tag{8}$$

where, $$\alpha(n) = d(n) - w_{n-1}^T x(n) \tag{9}$$

Here *d(n)* is the desired signal and $\alpha(n)$ is called the priori error which is the error that would occur if the filter coefficients were not updated.

5) Fast Transversal Filter (FTF): The basic idea behind the FTF algorithm is to avoid using the backward variables. By using only forward prediction variables and adding a small regularization constant and a leakage factor, a robust numerically stable FTF algorithm is obtained. The weight equation of the FTF algorithm is given by (10).

$$w_n = w_{n-1} + e(n)x(n) \tag{10}$$

Where, the symbols have the same significance as in (1). The error signal *e(n)* is found by (11).

$$e(n) = \xi(n)\gamma(n) \tag{11}$$

Here, $\xi(n)$ is the mean-square error given by (12).

$$\xi(n) = d(n) - x^T(n)w_{n-1} \tag{12}$$

The leakage factor $\gamma(n)$ is given by (13).

$$\gamma(n) = \frac{1}{1 + |x(n)|^2} \tag{13}$$

B. Models of the Applicaions Analyzed

As mentioned in Section II, three applications can be analyzed using the algorithms described. The models of these three problems are discussed here.

1) System Identification (SI): In order to identify an unknown system or plant, first, the system is modeled by an FIR filter with M adjustable coefficients [9]. Then, both the plant and model are excited by an input sequence *x(n)* and the error signal between the plant output and the output of the model is obtained as in Fig. 4(a). The output of the model *ŷ(n)* is given by (14).

$$\hat{y}(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{14}$$

Where, h(k) is the model coefficient. If *y(n)* is the output of the actual plant, then, the error sequence *e(n)* can be found by (15).

$$e(n) = y(n) - \hat{y}(n) \tag{15}$$

h(k) is selected in order to minimize the square summation of the error sequence *e(n)*. This least-square criterion leads to the set of linear equations (16) for determining the filter coefficients.

$$\sum_{k=0}^{M-1} h(k)r_{xx}(l-k) = r_{yx}(l);\ l = 0,1,2,\dots,M-1 \tag{16}$$

Where, $r_{xx}(l)$ is the autocorrelation of the sequence *x(n)* and $r_{yx}(l)$ is the cross correlation of the system output with the input sequences.

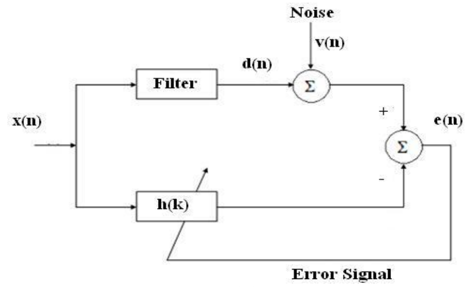
2) Adaptive Line Enhancement (ALE): In adaptive line enhancer a delayed replica of the primary signal is made. The primary and the delayed primary signals will be correlated but the noise components will not correlate, hence autocorrelation coefficients of noise decay much faster, revealing original signal [10]. The model is shown in Fig. 4(b). In this model, the adaptive linear predictor was used to estimate the narrowband interference for the purpose of suppressing the interference from the input sequence *v(n)*. The desired signal *x(n)* is either a spectral line or a relatively narrowband signal and *w(n)* represents a wideband noise component that masks x(n). The ALE being a self-tuning filter having peak frequency at the band of x(n), the wideband signal w(n) is suppressed and spectral line is enhanced with respect to noise power.

3) Adaptive Noise Cancellation (ANC): In ANC, the “primary” input contains the corrupted signal and a “reference” input contains the noise signal which is correlated with the primary noise. The reference input is sent to the adaptive filter and is subtracted from the primary input in order to obtain the desire signal estimate [11]. This model is shown in Fig. 4(c). The figure is self-explaining, the filter coefficients are optimized by adaptive algorithm used on the error signal.

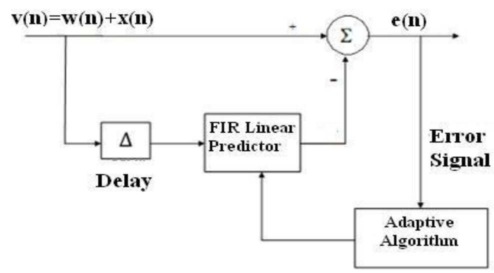
V. Case Study

It is definitely impossible to show detailed simulation results for all the possible combinations of applications, algorithms and input signals in the scope of this paper. So, two case studies of two particular situations simulated using ASSNAT v1.1 are presented here.

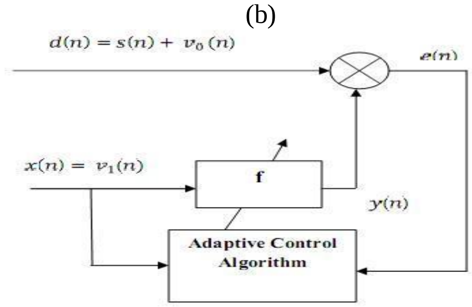
In these simulation, we’ve used two of the default input signals provided in the toolbox, namely “Stationary Signal No.1 and 2” for ANC and ALE using LMS algorithm. Output graphs for the two sets of parameter values (summarized in Table 1) are shown in Fig. 5. From the table and the figure, we can easily compare the two sets of data and obtain deductions relevant to our interest.



(a)

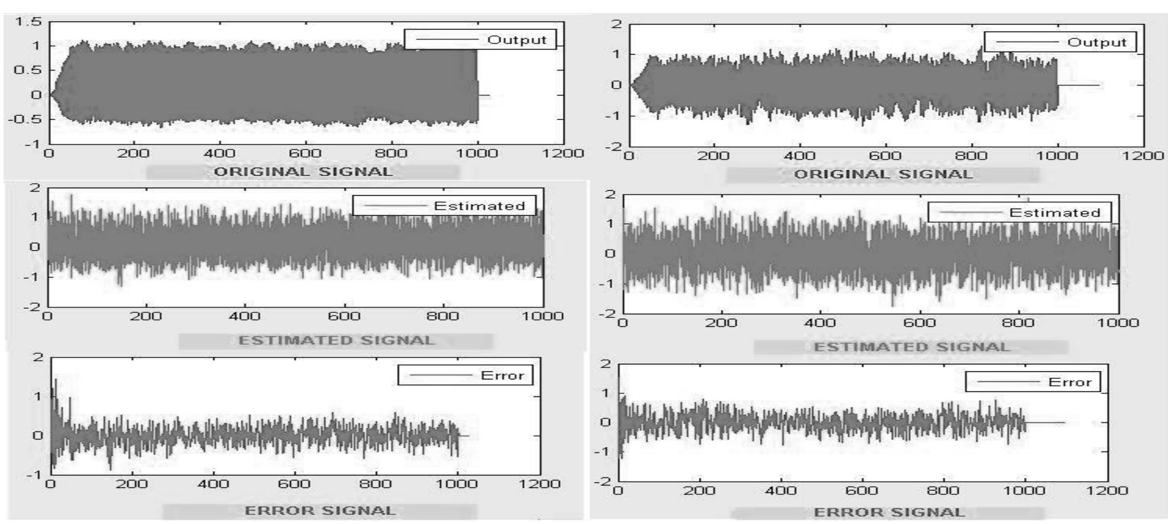


(b)



(c)

Figure 4. Models of different Adaptive System Applications: (a) SI, (b) ALE, (c) ANC.



(a) (b)

Figure 5. Graph of different signals obtained from ASSNAT v1.1 for the case studies: (a) Case study-1, (b) Case Study-2.

VI. Conclusion

This paper has presented the new and advanced open source toolbox for Adaptive System Simulation and Noise Analysis namely ASSNAT and its updated features for getting the clear and exact vision of the adaptive system and noises under study. The main platform of this toolbox is MATLAB. ASSNAT v1.1, which is the updated version of ASSNAT v1.0 introduced in [1], comes with a variety of algorithms, system applications and input signals that can be simulated. Realistic models, open-source codes, easy-to-use GUI, the new "Learning Curves" feature and a range of possible applications make ASSNAT well-suited for educational and research purposes. In future, the scope of this toolbox can be extended to include more real-life applications and algorithms.

References

[1] Ajoy Kumar Dey, Avijit Saha, Shibani Ghosh and Susmita Saha, "An open-source toolbox for adaptive system simulation and noise analysis", Proc. Of 2010 Intl. Conf. on Modeling, Simulation and Control (ICMSC 2010), pp. 322-326, Nov 2010, Cairo, Egypt.

[2] B. Widrow and S. D. Stearns, "Adaptive signal processing", Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.

[3] M. H. Hayes, "Statistical Digital Signal Processing and Modeling", 2nd ed., John Wiley & Sons, Inc., Wiley India, 2002.

[4] R. Rangayyan, "Biomedical Signal Analysis : A Case-Study Approach", Wiley-IEEE Press, 2001, 552 p.

[5] S. Catreux, V. Erceg, D. Gesbert, and R. W. Heath Jr., "Adaptive modulation and MIMO coding for broadband wireless data networks", Communications Magazine, IEEE, Vol. 40, no. 6, pp. 108 – 115, Jun 2002.

[6] J. Salz and J. H. Winters, "Effect of Fading Correlation on Adaptive Arrays in Digital Mobile Radio", IEEE transactions on vehicular technology, vol. 43, no.4, pp.1049-1057, Nov 1994.

[7] L. Håkansson, "Computer Exercises in SAS310: Adaptive Signal Extraction", Available: http://www.asb.tek.bth.se/courses/etc004/ovning/datorovn.pdf (2009, May 15).[Online]

[8] M. Chakraborty, R. Shaik, and M. H. Lee, "A Block Floating Point Based Realization of the Block LMS Algorithm", Circuits and Systems II: Express Briefs, IEEE Transactions on, vol. 53, no. 9, pp. 812 – 816, Sept 2006.

[9] L. Ljung, "System Identification: Theory for the users", Englewood Cliffs, NJ, Prentice-Hall, Inc., 1987.

[10] D. W. Tufts, L. J. Griffiths, B. Widrow, J. Glover, J. McCool, and J. Treichler, "Adaptive line enhancement and spectrum analysis", Proceedings of the IEEE, vol. 65, no. 1, pp. 169 – 173, Jan 1977.

[11] B. Widrow, J. R. Glover Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications:, Proceedings of the IEEE, vol. 63, no. 12, pp. 1692 – 1716, Dec 1975.

AUTHORS PROFILE

Ajoy Kumar Dey is doing his M.Sc in Department of Electrical Engineering with emphasis on Signal Processing at the Blekinge Institute of Technology (BTH), Karlskrona, Sweden. He graduated from the Daffodil International University, Dhaka, Bangladesh having majored in Telecommunication Engineering in 2008. His research interests are applied mathematical model, such as topological network theory, different geometry of continuum mechanism, pattern recognition, and mathematical foundations of neural network. He is currently working on intelligent acoustic surveillance model and binaural speech Enhancement. He has worked on the Performance Analysis of DSK Kits using CCS Integrated Development Environment, Design and Development of Approximation and prediction with feed forward Neural Network, Character Reorganization with Associative Network, Signal compression and reconstruction with principle component analysis (PCA), Radial Basis Function Networks and Genetic Algorithm for system Identification and filter design. He has worked as a peer reviewer under the Research Methodology Course.

TABLE I. Specifications for Case Studies

Name of the Attribute	CASE STUDY- 1	CASE STUDY- 2
Algorithm	LMS	LMS
Specified Values	$\mu = 0.008$ order = 32 Length = 1000	$\mu = 0.008$ order = 200 Length = 1000
Selected Application	ANC	ALE
Input Signal	Stationary Signal No.1	Stationary Signal No.2

Avijit Saha received his B.Sc. degree in Electrical and Electronic Engineering (EEE) in 2009 from Bangladesh University of Engineering and Technology (BUET), where he is currently working his way towards his M.Sc. in EEE. He has worked on fault identification and localization in analog integrated circuits during his undergraduate research work, finding a novel method for fault testing in analog ICs. He has also worked on CMOS circuit based THz oscillators. Being interested in physics, characterization and modeling of solid-state devices, he is currently studying on different FET devices in his M.Sc thesis. Besides his studies and research work, he is currently working as a Lecturer in Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. He has been a member of IEEE since his undergraduate years, and is also a member of IEEE Communication Society. He has received many scholarships throughout his High School and Undergraduate studies.

Shibani Ghosh obtained her B.Sc in EEE from Bangladesh University of Engineering and Technology (BUET) in 2009. She is pursuing her M.Sc degree in EEE from the same university. She is also a full-time Lecturer in University of Information Technology and Sciences (UITS), Dhaka, Bangladesh, where she teaches freshmen and sophomores elementary courses like Electrical Circuits, Machines, Digital Electronics etc. She also supervises undergraduate student projects and serves as the Lab In-charge of the Electrical Circuit Lab. She has studied and worked on various research topics like simulation of load flows in power systems, power management algorithms in dc-to-dc converters, adaptive system simulation, efficacy of OFDM systems, voltage controlled oscillators etc. She is currently carrying her research on Thin Film Transistors and is interested to continue her research in nano-scale devices and technology. She is a member of IEEE.

Susmita Saha is doing her M.Sc in Department of Electrical Engineering with emphasis on Signal Processing at the Blekinge Institute of Technology (BTH), Sweden. She has completed her B.Sc Engineering in Electronics and Telecommunication Engineering from Daffodil International University, Bangladesh. Her research interests are signal processing, multidimensional signal processing, adaptive signal processing. She is currently working on the development of intelligent hearing system. She has worked on Demodulation by Bayesian Techniques in Mobile-satellite based networks, Adaptive system simulation,. Identification of load on a PoL voltage converter. She is the author of many research publications and has worked as a peer reviewer under the course of Research Methodology.