

A New Technique to Implement Filtered X-LMS Algorithm for Active Noise Control Applications Using Reconfigurable Logic

N.J.R. Muniraj and R.S.D. Wahidhabanu

Shri Sakthi Institute of Engineering and Technology, Shri Sakthi Nagar, L and T Bypass,
Civil Aerodrome (via), Venkittapuram Post, Coimbatore-641014, India

Abstract: This study proposes the implementation of Filtered X-LMS algorithm into VLSI using Verilog HDL for active noise control applications. This ASIC chip was designed, simulated and synthesized using Xilinx FPGA Virtex 2P (2vp40ff1148-6) and the workability of the algorithm was tested for noise cancellation and verified using MATLAB. The operating frequency was about 24 MHZ.

Key words: X-LMS, algorithm, filtered, FPGA, ASIC

INTRODUCTION

The Filtered-XLMS algorithm used in active noise control applications suffers from convergence due to the output delay caused by the secondary path of the system. To overcome the same, a computationally efficient method is identified for removing the output delay within the error signal used to update the controller coefficients, so that the standard LMS algorithm is efficiently employed using VLSI technology. Several Filtered X-LMS algorithm have been proposed in the literature (Rupp, 1997; Rupp and Sayed, 1995). However, the real time implementation of Filtered X-LMS algorithms has largely remain unexplored. Recently, a computationally-efficient pipelined LMS algorithm using Virtex FPGA was proposed in Ting and Roger (2005).

In this study, we present a VLSI architecture for the Filtered X-LMS task for active noise control applications. The architecture employs two-input, one-output Filtered X-LMS chip with self contained control logic. The system can be operated at a frequency of 24 MHz has been designed for fabrication in 0.6 μ m technology using Xilinx FPGA Virtex 2P (2vp40ff1148-6) tools. Adaptive filtering has been a key enabling ingredient in many current as well as newly emerging communication technologies. Today, adaptive filtering techniques are used to overcome channel limitations and mitigate echoes in high speed digital subscriber loops and gigabyte networks (Bernard and Samuel, 2002).

In order to compensate the effect of secondary path transfer function, the conventional LMS algorithm is modified by placing an identical filter in the reference signal path to the weight update of the LMS algorithm (Johnson and Larimore, 1997) thus realizing the Filtered X-LMS algorithm. The Filtered X-LMS is currently the most

popular method of adapting a filter where, a transfer function exists in the error path. Such instances arise in the case of active control of sound and vibration. The derivations rely on the linearity of adaptive filter and the error channel. In order to ensure the convergence of the algorithm, the input to the error correlator is filtered by a copy (estimate) of the secondary-path transfer function. Further, the Filtered X-LMS algorithm would precisely compensate for the phase shift introduced by the secondary path. The Filtered X-LMS algorithm requires knowledge of the secondary path transfer function. The Filtered X-LMS algorithm exhibits stable behaviour even when the coefficients change within the time scale associated with the dynamic response of the forward path. Further, the maximum step length depends not only on the length of the adaptive filter and the filtered reference signal but also on the delays in the forward path. In this study, a simple and novel technique is proposed to implement the Filtered X-LMS Algorithm into hardware which would trade off between quality and cost of implementation. Having inspired by the research work done at Georgia Institute Technology, the present research has been carried out. The study includes, a brief review of the Filtered X-LMS algorithm, the proposed architecture, VERILOG HDL and the simulation and synthesis using Xilinx FPGA Virtex 2P (2vp40ff1148-6) with implementation. A brief conclusion is also given highlighting the salient features of the method implemented.

ALGORITHM

The conventional LMS algorithm is likely to be unstable in applications where system characteristics vary dynamically because of phase shift. The conventional

LMS algorithm is also unstable when speed of the dynamically varying system is greater than the speed of adaptation of the LMS. The step size determines the speed of convergence (Emmanuel and Barrie, 2002). Smaller the step size smaller is the adaptation error after convergence. The Filtered X-LMS algorithm is well suited for both broadband and narrowband control tasks with a structure that could be adjusted based on the problem at hand. The algorithms structure and operation are ideally suited for architectures of standard DSP chips, due to the algorithms extensive use of Multiply/Accumulate (MAC) operation. It behaves robustly in the presence of physical and numerical errors caused by finite precision calculations. These algorithms are technically found to be easier to setup and tune to real world environment. A conventional adaptive algorithm such as the LMS algorithm is likely to be unstable to the phase shift (delay) introduced by the forward path. So a well-known Filtered-XLMS algorithm, however, an adaptive Filter algorithm, was identified to be suitable for active control applications. It was developed from the LMS algorithm, where a model of the dynamic system between the Filter output and the estimate, i.e., the forward path was introduced between the input signal and the algorithm for the adaptation of the coefficient vector. The Filtered-XLMS algorithm was suitable for applications where a dynamic system exist between the filter output and the estimate. That could be derived from the standard LMS algorithm by commuting the order of the filter and the channel. The algorithm employed filtered version of the input signal values that was created by filtering every input signal.

FIR Filter output is as usual given by the vector inner product:

$$y(n) = w^T(n) x(n) \quad (1)$$

$$\text{Where } x(n) = [x(n), x(n-1), \dots, x(n-M+1)]^T \quad (2)$$

is the input signal vector to the adaptive Filter and

$$w(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad (3)$$

is the adjustable Filter coefficient vector.

In control applications, the estimation error $e(n)$ is denoted by the difference between the desired signal (desired response) $d(n)$ and the output signal from the forward path or system under control $y_C(n)$:

$$e(n) = d(n) - y_C(n) \quad (4)$$

The conventional LMS algorithm is likely to be unstable in control applications. The conventional LMS

algorithm will in some cases also find a poor solution when it converges. This can be explained by the fact that the LMS algorithm uses a gradient estimate $x(n) e(n)$ which is not correct in the mean. A compensated algorithm has been obtained by filtering the reference signal to the coefficient adjustment algorithm using a model of the forward path (Fig. 1).

The algorithm obtained is the well-known Filtered-X LMS algorithm defined by:

$$y(n) = w^T(n) x(n) \quad (5)$$

$$e(n) = d(n) - y_C(n) \quad (6)$$

$$x_C^*(n) = c^* T x(n) \quad (7)$$

$$w(n+1) = w(n) + \mu x_C^*(n) e(n) \quad (8)$$

Where

$$x_C^*(n) = \begin{bmatrix} \sum_{i=0}^{I-1} c_i^* x(n-i) \\ \sum_{i=0}^{I-1} c_i^* x(n-i-1) \\ \vdots \\ \sum_{i=0}^{I-1} c_i^* x(n-i-M+1) \end{bmatrix} \quad (9)$$

and c_i^* is the coefficients of an estimated FIR filter model of the forward path:

$$h_C^*(n) = \begin{cases} c_i^* & \text{if } n \in \{0, \dots, I-1\} \\ 0 & \text{else} \end{cases} \quad (10)$$

It is in practice to use an estimate of the impulse response for the forward path. As a result, the reference

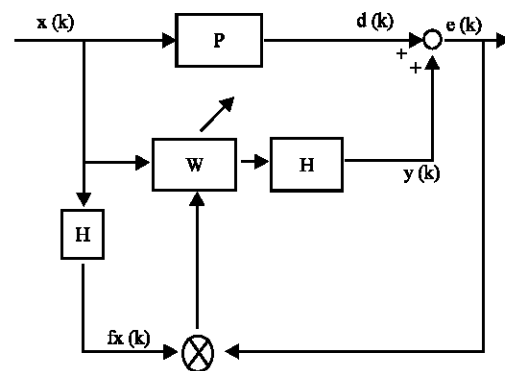


Fig. 1: Block diagram of filtered X-LMS algorithm

signal $x_C^*(n)$ would be an approximation and differences between the estimate of the forward path and the true forward path influence both the stability properties and the convergence rate of the algorithm. However, the algorithm is robust to errors in the estimate of the forward path. The model used should introduce a time delay corresponding to the forward paths at the dominating frequencies (Haykins, 1996).

In the case of narrow-band reference signals to the algorithm, e.g. $\sin(\omega_0 t)$, the algorithm would converge with phase errors in the estimate of the forward path with up to 90° , provided that the step length μ is sufficiently small. Furthermore, phase errors in the estimate of the forward path smaller than 45° would have only a minor influence on the algorithm convergence rate. The filtered-x LMS algorithm relies principally on the assumption that the adaptive FIR filter and the forward path commute. This is approximately true if the adaptive filter varies in a time scale, which is slow in comparison with the time constant for the impulse response of the forward path. We can thus write as follows:

$$\sum_{i=0}^{I-1} c_i \sum_{m=0}^{M-1} w_m(n-i) \mu(n-i-m) \approx \sum_{m=0}^{M-1} w_m(n) \sum_{i=0}^{I-1} c_i \mu(n-m-i)$$

$$\mu < \frac{2}{ME[x^2(n)]} \quad (11)$$

When

$$w(n) = w(n-i), i. \{1, 2, \dots, I-1\} \quad (12)$$

Where I is the length of the impulse response of the forward path. In practice, the filtered x-LMS algorithm exhibits stable behaviour even when the coefficients change within the time scale associated with the dynamic response of the forward path. In order to ensure that the action of the LMS algorithm is stable the maximum value for the step length μ should be given approximately by:

$$\mu_{\max} \approx \frac{2}{E[x_C^2(n)(M+\delta)]}$$

However, in the case of the Filtered-x LMS algorithm, it has been found that the maximum step length μ not only depends on the length of the adaptive filter and filtered reference signal but also on the delays in the forward path C . If the reference signal $x_C^*(n)$ is a white noise process it has thus been found that an upper limit for the step length μ is given by:

$$\mu_{\max} \approx \frac{2}{E[x_C^2(n)(M+\delta)]}$$

where δ is the overall delay in the forward path (in samples).

The basic principle behind the Filtered X-LMS algorithm as shown in the Fig. 1 is that the input vector $x(k)$ is filtered through the adaptive filter coefficients vector $w(n-1)$ to produce the filter output vector $y(k)$. This output vector is passed through the secondary path filter to produce the secondary actuator response at the sensor $y(k)$.

The adaptive transversal filter output is evaluated along with the error signal. The adaptive transversal filter coefficients using the relationship:

$$w(k+1) = w(k) + \mu e(k) x(k) \quad (13)$$

The current error sample $e(n)$ is evaluated using the relationship

$$e(k) = d(k) + y(k) \quad (14)$$

$S(n)$ is the transfer function of secondary path. It is to be noted that error here is formed by adding the signal rather than subtracting them to be compatible with real world sensors such as microphones and accelerometers. The input signal $x(k)$ is filtered through the estimate of the secondary path to produce the filtered-x signal $fx(k)$. Now $fx(k)$ and $e(k)$ are used to calculate the normalized gradient vector and this is used to update the adaptive filter coefficients. Here, the original input becomes filtered by the channel before entering the filter and the error appears directly at the output of the adaptive filter (Haykins, 1996).

RESULTS AND DISCUSSION

Using matlab and FPGA: The application of Adaptive filters for active noise control using Filtered-XLMS algorithm is implemented using matlab for a random input and it is plotted in Fig. 2 for various adaptation step sizes and it clearly shows that the algorithm converges for the step size of 0.01 to 0.10, here the convergence of error is very high when compared to LMS adaptive algorithm and stability is maintained.

For different step sizes the variance, mean, mean square error and the signal to noise ratio for the Filtered-XLMS algorithm is tabulated and plotted (Table 1). It picturizes the mean, variance, MSE and SNR values for various step sizes and differentiates clearly the convergence of the algorithm for various step sizes, giving the different values (Fig. 3).

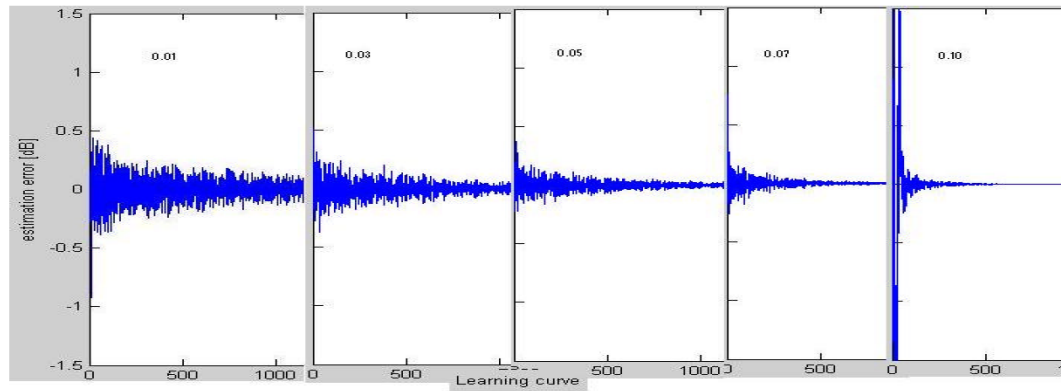


Fig. 2: Performance of active noise control using Filtered-XLMS at various step size

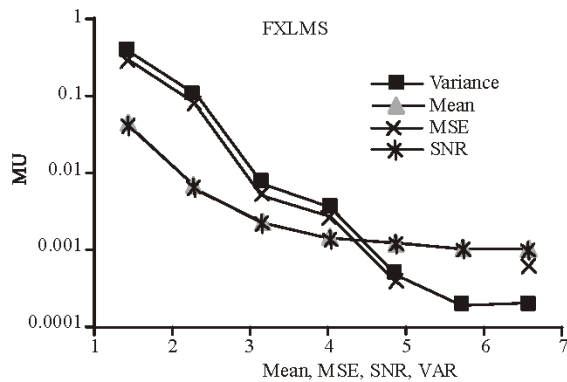


Fig. 3: Mean, Variance, MSE and SNR plot for various step size

Table 1: Mean, Variance, MSE and SNR values for various step size

MU	Variance	Mean	MSE	SNR
1E-06	0.3618	0.0405	0.3104	0.0405
6E-06	0.1058	0.0065	0.0833	0.0065
0.00001	0.0069	0.0021	0.0051	0.0021
0.00006	0.0034	0.0014	0.0025	0.0014
0.0001	.0005	0.002	3.94E-04	0.0012
0.0006	0.0002	0.001	4.94E-04	0.001
0.001	0.00021	0.001	5.94E-04	0.001

User-programmable gate arrays, called Field-Programmable Gate Arrays (FPGAs), have emerged and have changed the way electronic systems are designed and implemented. FPGA chips are prefabricated as arrays of identical programmable logic blocks with routing resources and are configured by the user into the desired circuit functionality. The most popular FPGA architectures use either a Look-Up-Table (LUT) or a multiplexer-configuration as the basic building block.

With the growing complexity of the logic circuits that could be packed on an FPGA chip, having automatic synthesis tools that implement logic functions has become indispensable on these architectures.

Conventional synthesis approaches fail to produce satisfactory solutions for FPGAs, since the constraints imposed by the FPGA architectures are quite different. The design is coded in Verilog HDL (hardware description language), a more generalized method of describing the behavior of logic systems than logic equations (Jhonson and Larimore, 1997). The system was visualized as a set of black boxes called modules. The top level module was broken into successively less complex functions until the bottom level was reached (RTL level description of the function). The Filtered XLMS algorithm was simulated and examined to assess whether the simulation had achieved the desired result using Modelsim 5.8. Its equivalent timing diagram is shown in Fig. 4. for 16-bit input data.

Logic synthesis takes the circuit description at the register-transfer level and generates an optimal implementation in terms of an interconnections of logic gates. Schematic capture is, probably, still most popular method of defining logic for FPGAs and many ASICs. It is a CAD systems dedicated to logic design. Logic functions of complexity ranging from an inverter to multi-bit counters are stored in a library which describes both their functionality and a graphic symbol. The designer calls up the symbols from the library, places them on the screen of a PC or workstation and connects them with wires and busses.

The modification of the design takes place at two levels. At the visual level, the designer is creating a visual representation of the logic which he requires, in terms of familiar symbols for the components. At a level below this is a netlist which defines the location of the each component on the screen and the way each is connected to the other components in the design (Meyer, 2006).

The RTL description of Filtered XLMS algorithm is first optimized for an objective function such as minimum chip area, meeting the performance constraints, low

power, etc. This step is called logic optimization. The optimized representation is then mapped to some primitive cells present in a library. The final implementation is in terms of interconnections of gates, functional units, registers, is synthesized in Xilinx 6.3i and implemented on a Virtex XC2VP40-6 chip is shown in Fig. 5.

Floorplanning allows us to predict the interconnect delay by estimating the interconnect length (Michael *et al.*, 2001). The objectives of floorplanning are to minimize the chip area and minimize delay. The Filtered XLMS algorithm is floor planned and its result is shown in Fig. 6

The locations of various modules on the chip are determined (placement) and the interconnections of the circuit are routed between or through the placed modules.

Also, the pad locations for inputs and outputs are determined in this step (Geoff, 1996). The final layout is sent for fabrication and the layout for the Filtered-XLMS algorithm is shown in Fig. 7.

The timing simulation data generated from Place and Route are generated to carry out the timing simulation for the final verification of the design. Best performance is achieved by mapping the basic components to use the minimum Array-of-Slices (AoSs). The arrangement of the component's position in the Virtex device is important in minimizing the interconnect delay. The components used to implement the Filtered X-LMS have been optimized for the Virtex FPGA circuit. Characteristics comparison (Table 2, 3 and 4) shows LMS and FILTERED-XLMS algorithm and the device utilization.

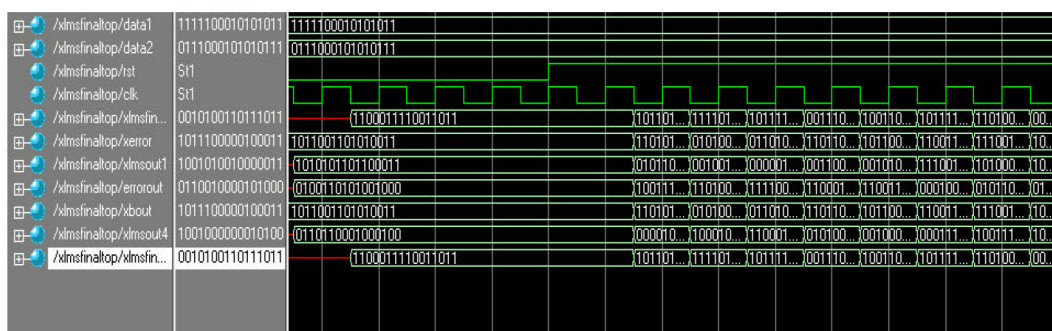


Fig. 4: Timing diagram of filtered X-LMS algorithm

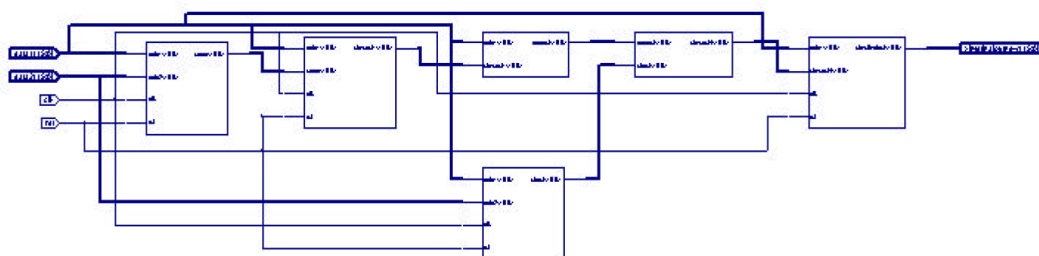


Fig. 5: Synthesis resu

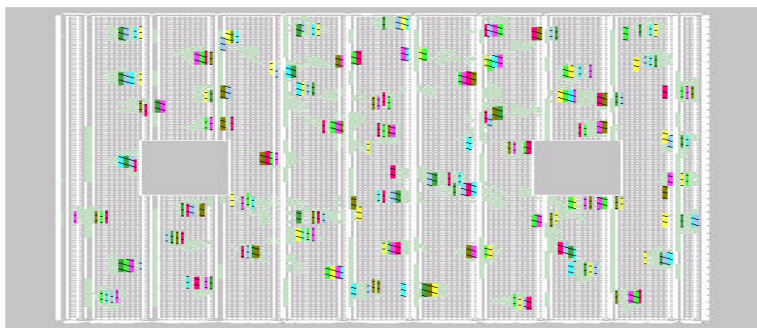


Fig. 6: Floorplanning report in FPGA

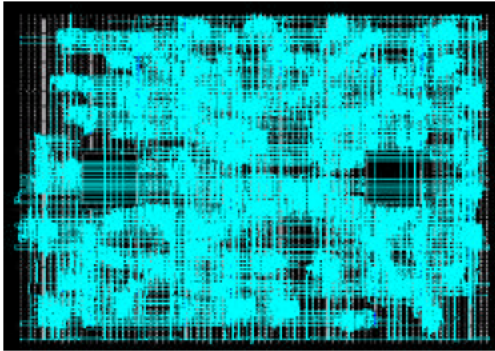


Fig. 7: Implementation report in FPGA

Table 2: Characteristics of LMS and filtered XLMS

Architecture	Registers	Adders/subtractors	Multipliers	Comparators
LMS	137	46	24	15
FXLMS	520	185	97	60

Table 3: Characteristics of LMS and filtered XLMS

Architecture	Gatecount	Power	Frequency	IOB's	Memory usage
LMS	113423	798 mW	18 MHz	2400	146 MB
FXLMS	453,520	873 mW	24MHz	2400	209 MB

Table 4: Device utilization summary of filtered XLMS

Selected device:	2vp40ff1148-6
Number of slices:	3756 out of 19392 19%
Number of slice flip flops:	672 out of 38784 1%
Number of 4 input LUTs:	6886 out of 38784 17%
Number of bonded IOBs:	49 out of 804 6%
Number of MULT18X18s:	97 out of 192 50%
Number of GCLKs:	1 out of 16 6%

CONCLUSION

In this study, a high-speed FPGA implementation of FILTERED-XLMS adaptive filter is presented. The algorithm have been successfully implemented on the Virtex: 2vp 40 ff148-6. chip. The powerful design of FILTERED-XLMS algorithm using reconfigurable logic shortens the design cycle and provides good utilization of the device and also provides synthesizer options, i.e., chooses between optimization speed versus size of the design. As inferred from speed analysis, the Filtered-XLMS algorithm, exhibits considerably more speed consumption at the RTL coding stage.

The gate count, area analysis, registers, adders/subtractors, comparators, speed of multipliers and frequency of Filtered-XLMS algorithm proves to be exceptional. Also, error floor and the consistency of the error is, in general, better for the Filtered-XLMS algorithm compared to LMS algorithm. The MATLAB results also proves that it is effective.

ACKNOWLEDGEMENT

The authors thank Dr. S. Thangavelu, Chairman, Shri Shakthi Institute of Engineering and Technology, Coimbatore for the guidance, support and providing research facilities.

REFERENCES

- Bernard Widrow, Samuel D. Stearns, 2002. Adaptive Signal Processing. Pearson Education Asia, (2nd Edn.).
- Emmanuel, C. Ifeachor and Barrie W. Jervis, 2002. Digital Signal Processing. A. practical approach. Pearson Education Asia, (2nd Edn.).
- Geoff Bostock, 1996. FPGA's and Programmable LSI: A Designer's Handbook.
- Haykins, S., 1996. Adaptive Filter Theory. (3rd Edn.), Prentice Hall Inc., N.J.
- Johnson, C.R., Jr and M.G. Larimore, 1997. Comments on and addition to An Adaptive Recursive LMS Filter. Proc. IEEE., pp: 1399-1401.
- Lok-Kee Ting, Roger Woods, Senior Member 2005. Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers. IEEE. Transactions on Very Large Scale Integration, Vol. 13.
- Meyer-Baese, U., 2006. Digital Signal Processing with Field Programmable Gate Arrays. (2nd Edn.), Springer International Edition.
- Michael John Sebastian Smith, 2001. Application-Specific Integrated Circuits. (5th Edn.), Pearson Education Inc, Asia.
- Rupp, M. and A.H. Sayed, 2005. Modified FXLMS algorithms with improved convergence performance, Conf. Rec. 29th Asilomar Conf. Signals, Sys., Comput., 2: (Los Alamitos, CA., IEEE Comput. Soc. Press, 1995), 1225-1259.
- Rupp, M., 1997. Saving Complexity of modified Filtered-XLMS and delayed update LMS algorithm, IEEE Trans., Circuits Syst. II: Analog and Digital. Signal Processing, pp: 45-48.
- Scott, C. Douglas. An Efficient Implementation of the Modified Filtered-XLMS Algorithm. IEEE Signal Processing Letters, EDICS category No., SPL. SA. 2.3.
- Treichler, Johnson, Larimore. Theory and design of Adaptive Filters. Prentice Hall of India Pvt. Ltd., pp: 269.