IJCTA, 8(3), 2015, pp. 1243-1249

Control of Vibrations using Adaptive Filters

Bipin Krishna and Manal Ubaidullah

Abstract: Vibrations are a primary factor for wear and tear of a lot of machines, machine parts, structures, vehicles, motors, etc. A small but consistent amount of vibration can also cause major disturbance and damage. It has a long list of causes as well. But human involvement and environmental involvement are the two most major factors that cannot be decremented or controlled. Hence, it is of utmost necessity that these rogue vibrations are suppressed or better yet, controlled. There are various techniques in recent times that have been developed to combat this problem. One of the most effective methods are adaptive filtering algorithms. We will be taking a close look at these algorithms in this paper. Among these adaptive filtering algorithms, the highest convergence rate is possessed by the Least Mean Square Algorithms. These algorithms play a vital role in detecting, analyzing, understanding and controlling of Vibrations. After successful control and suppression of these vibrations, the quality of a particular adaptive filter can be determined using the Mean Square Error of the resultant signals.

***Keywords**:* Vibrations, Least Mean Square (LMS), Adaptive filter.

1. INTRODUCTION

Vibration control of flexible structures is of high importance as these structures are dynamic systems subjected to external excitation. When the excitation equals the natural frequency of the structure, resonance occurs and thereby damaging the structures. Cantilever beams are such structures used for investigation of vibration control. Beams supported with piezoelectric actuators and sensors are called as piezoactuated cantilever beams or smart beams. Though active vibration control with the use of sensors and actuators is not a new concept, it is still a developing area of research in the fields of civil and aerospace applications.

Adaptive filtering is playing an important role not only within the domain of system identification as in [8, 9] but also in vibration control applications [1, 2]. Filters based on LMS and RLS are the two primary algorithms used for adaptation but RLS is attractive because of the faster convergence [3]. Recently, there is a great interest in active vibration control of beam, plate and shell structures [8]. Vibration is an undesirable phenomenon in aerospace, mechanical and civil systems. Certain specific aerospace structures be it curved, flat, thin walled, etc. are operated in harsh and adverse aerodynamic environments, who in turn fall prey to random and uncontrolled vibrational disturbances. Enhancement of the performance of these particular structures is possible with the concept of active vibration [5]-[7]. PZT materials that are electro-mechanically coupled [4] are behave as actuators and sensors in many active vibration control applications.

A number of studies are reported on the modeling of electromechanical coupling with different structural applications in active vibration control. A few experimental studies are also available on the active vibration control of composite beams and plates.

Manipal Institute of Technology (Manipal University), Manipal-576104, Udupi District, Karnataka, India
email: bipin.monayi@gmail.com, manal_jtr@yahoo.in

2. METHODOLOGY

2.1. The Least Mean Squares (LMS) algorithm

The LMS algorithm is quite advantageous and simple to determine. This algorithm shall possesses excellent performance if the adaptive filter system is formulated as an adaptive linear combiner, provided, that this n-dimensional vector input *X (k)* as well as the desired output *d (k)* are present in every iteration, here *X (k*) is represented as,

$$X(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

Here, the n-dimensional arbitrary set of tunable weights *W (k*) is,

$$W(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_n(k) \end{bmatrix}$$

Now that we have the input vector *X (k)*, the approximated output y *(k)* can be determined as the linear combination of the input vector *X (k)* along with the weight vector *W (k)* as given,

$$y(k) = X^T(k)W(k)$$

Hence, the approximated error *e (k)*, the difference between the approximated output *y (k)*, and the required signal *d (k)*, can be calculated as,

$$e(k) = d(k) - y(k) = d(k) - X^T(k)W(k)$$

Now applying expectation on either sides of the following equation,

$$E\{e^2(k)\} = E\{d^2(k)\} - 2 \cdot E\{d(k)X^T(k)\}W(k) + W^T(k)E\{X(k)X^T(k)\}W(k)$$

Hence, $E\{e^2(k)\}$ can be formulated as,

$$E\{e^2(k)\} = E\{d^2(k)\} - 2 \cdot P^T W(k) + W^T(k)RW(k)$$

Since the LMS algorithm descends on the performance surface algorithm, we utilize $e^2(k)$ to approximate the gradient vector, which is,

$$\nabla(k) = \frac{\partial e^2(k)}{\partial W(k)} = -2 \cdot e(k)X(k)$$

From the steepest descent type of adaptive algorithm.

$$W(k+1) = W(k) - \mu\nabla(k)$$

After replacing for $\Delta(k)$, we obtain,

$$W(k+1) = W(k) + 2\mu e(k)X(k)$$

After substituting for *e (k)*, we obtain,

$$W(k+1) = W(k) - \mu\big(y(k) - d(k)\big)X(k) \qquad 2.1$$

It is evident that the linear combination of said input signal, *X (k)* and the weight vector, *W (k)* gives us the approximated output vector *Y (k)*. Now the difference in the approximated output *Y (k)* and the required output *d (k)* gives us the value of error, *e (k)*. This inaccuracy value can be utilized as a feedback signal for adjustment of the weight vector *W (k)*. In an ideal system, the value of this error would be zero or very close to it.

2.2. Understanding the LMS Algorithm

The fact that the LMS algorithm is very simple to follow and is not a hassle while implementation talks about it being a remarkable option for various real-time applications.

The phases for the implementation of the LMS algorithm:

1. The desired response should be defined. Each coefficient weight should be set to zero.

$$w(n)_{-0}, n = 1,2,3, \dots, N \qquad 2.2$$

For every sampling instant (n), carried out phases (2) up to (4):

2. Every sample within the input array should be moved one spot towards the right and the present data sample n should be loaded into the primary spot in the array.

$$y(n) = \sum_{n\ 0}^{N\ 1} w(n)x(n) \qquad 2.3$$

3. The error should be computed prior to the updating of the filter coefficients, viz. find the difference of the result of the adaptive filter and the desired response.

$$e(n) = y(n) - d(n) \qquad 2.4$$

4. The error must be multiplied by μ, the learning rate parameter, so that the updating of the filter coefficients can take place. Now, the result of this multiplication must be further multiplied with the filter input and summed with the previous filter coefficient values.

$$\overrightarrow{w}(n+1) = \vec{w}(n) + \mu\ .e(n).\vec{x}(n) \qquad 2.5$$

2.2. The Normalized LMS

The normalized LMS (NLMS) algorithm can be called as a modification of the typical LMS algorithm. The equation used by the NLMS algorithm for the purpose of updating the coefficients of the adaptive filter is given below:

$$\vec{w}(n+1) = \vec{w}(n) + \mu\ .e(n).\frac{\vec{u}(n)}{\|\vec{u}(n)\|^2} \qquad 2.7$$

Which is also interpreted as,

$$\vec{w}(n+1) = \vec{w}(n) + \mu(n).e(n).\vec{u}(n) \qquad 2.8$$

Where, $\mu(n) = \mu/\|\vec{u}(n)\|^2$

Now, (2.8) which is the NLMS algorithm closely resembles the standard LMS algorithm excluding the fact that in this case the NLMS algorithm makes use of a time-varying learning rate μ (n). This varying learning rate can further enhance the convergence rate of this adaptive filter. The LMS algorithm is a somewhat slower in converging when associated to the NLMS algorithm. But, the tradeoff in this case is that of residual error. The sensitivity to the surmounting of its input vector *x (n)* is the primary drawback of the standard LMS algorithm. This makes choosing a suitable learning rate μ, which assures this algorithm's stability, very difficult.

2.3. Recursive least squares (RLS) algorithms

The operations performed by the standard RLS algorithm for the purpose of updating the adaptive filter's coefficients are given below.

1. Firstly, the output y (n) of the adaptive filter is calculated.
2. Then, the error e (n) is computed utilizing the equation below:
   $e(n) = d(n) – y(n)$.
3. Lastly, the coefficients of the filter are updated by applying the equation below:

$$\vec{w}(n+1) = \vec{w}(n) + e(n) \cdot \vec{K}(n) \qquad 2.9$$

Where $\vec{w}(n)$ is called the filter coefficients vector and $\vec{K}(n)$ is the gain vector. $\vec{K}(n)$ is:

$$\vec{K}(n) = \frac{P(n) \cdot \vec{u}(n)}{\lambda + \vec{u}^T(n) \cdot P(n) \cdot \vec{u}(n)} \qquad 2.10$$

Here, the regularization factor is δ. The typical RLS algorithm utilizes the equation below to update the inverse correlation matrix.

$$P(n+1) = \lambda^{-1}P(n) - \lambda^{-1}\vec{K}(n) \cdot \vec{u}^T(n) \cdot P(n) \qquad 2.11$$

Where, N is termed as filter length and λ is termed as forgetting factor. This algorithm computes the value e^2 (n) and the previous values which are e^2 (n–1), e^2 (n–2)... e^2 (n–N+1). The forgetting factor value ranges from (0, 1). If the forgetting factor were to fall lower than 1, it would signify that this algorithm places a smaller weight on the past value and a larger weight on the current values. The resulting $E[e^2(n)]$ of the RLS algorithms is more accurate than that of the LMS algorithms.

3. EMPIRICAL DATA

3.1. Filter Length Effect:

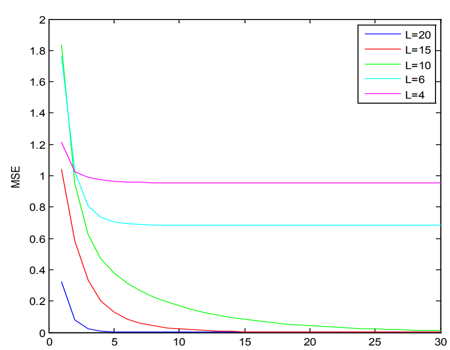

Figure 3.1: Behavior of NLMS algorithm for different filter length.

The NLMS algorithm is a fast convergence algorithm, at the expense of having larger steady state error for small filter length, but for larger filter length we have faster convergence and minimum steady state error. We also need less number of iteration to reach the steady state of our system.

3.2. STEP SIZE EFFECT:

For a very large mu the system the system will ripple until reach the steady state, this will affects the stability of the system, it maybe unstable, but if we decrease the value of mu the ripple will decrease, system becomes stable and converge faster. For a suitable step size our system will converge fast, will be stable and we can get minimum steady state error.

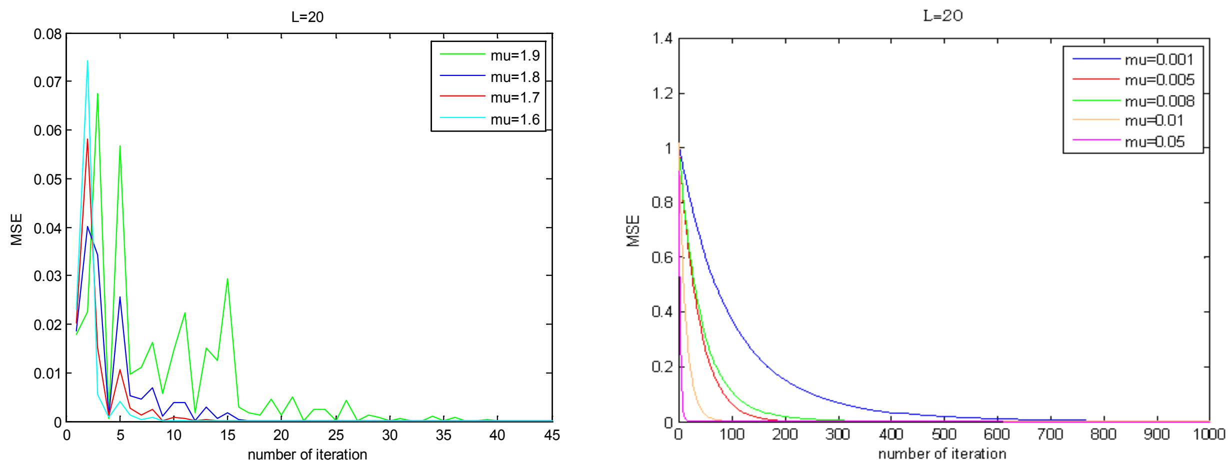

Figure 3.2: Behavior of NLMS algorithm for varying step size (mu).

3.3. CONVERGENCE OF WEIGHTS:

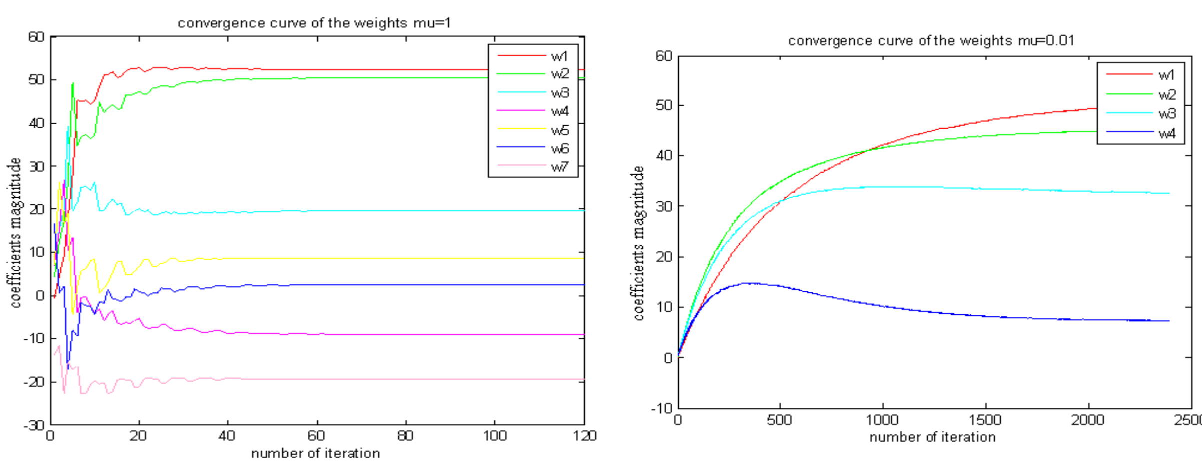

Figure 3.3: Varying Convergence curves of the NLMS algorithm

The NLMS is a fast convergence algorithm but its coefficients will ripple up and down before reaching their steady state.

As we see from the above figures, when the number of coefficients are large our system will converge faster, and the steady state error will be minimum. But by decreasing the number of coefficients we require higher amount of iterations to converge, and the steady state error gets increased. For a system having four coefficients, we note that if we decrease the value of step size the steady state error decrements, but we need a larger number of iterations to converge and reach the steady state.

4. RESULTS

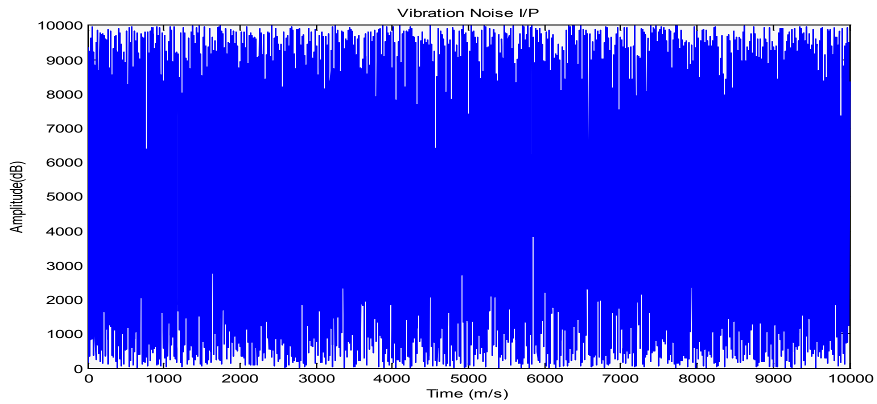

Figure 4.1: Raw input data

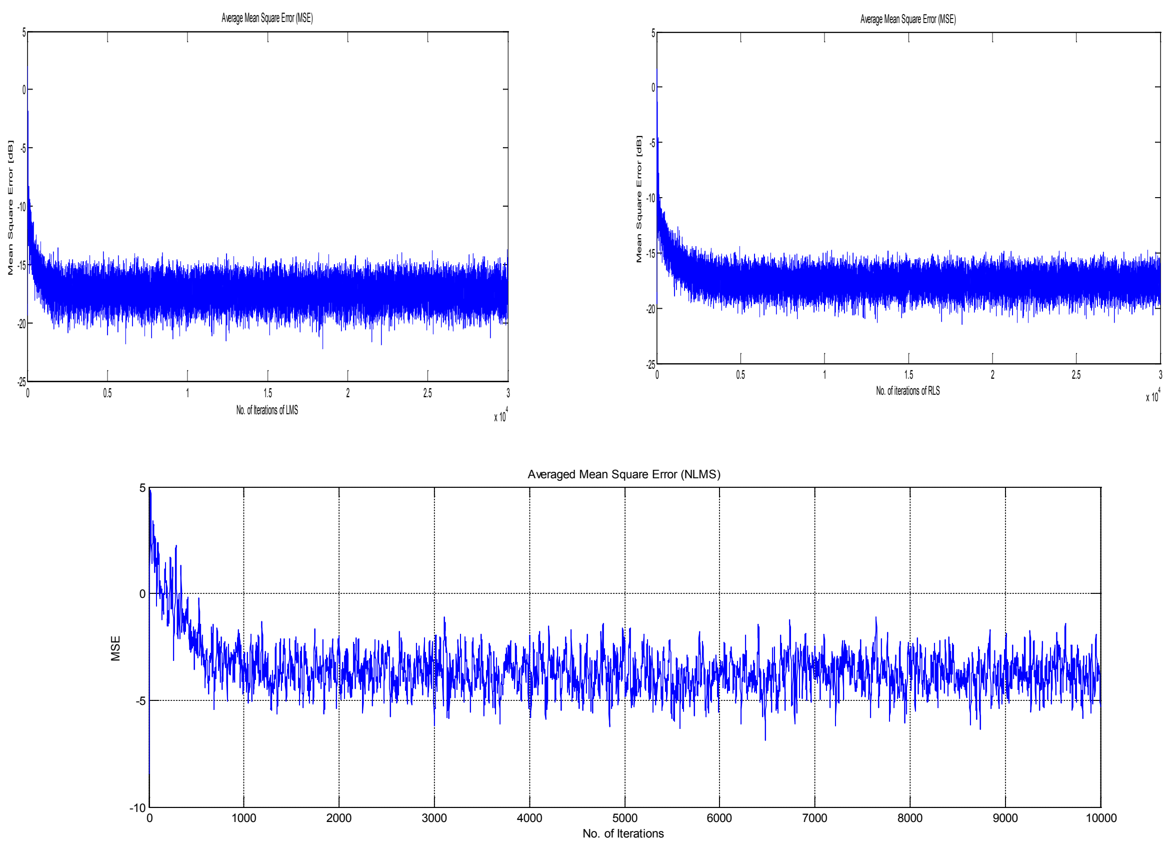

Figure 4.2: The Mean Square Error for the LMS, RLS and NLMS algorithms

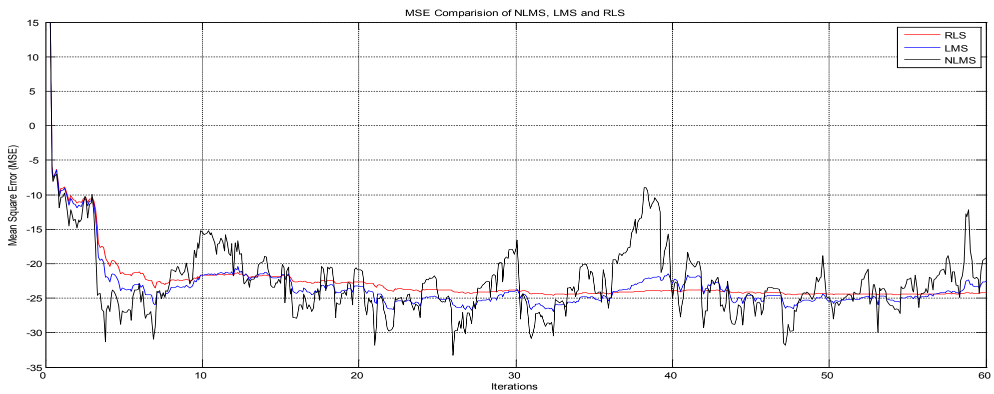

Figure 4.3: Comparison of performance of NLMS, LMS and RLS algorithms

5. CONCLUSION

The proposed work concentrates on the active vibration control of piezoactuated cantilever beam using LMS, NLMS, and RLS based adaptive filters. This algorithm proved to suppress vibrations of cantilever beams at its natural frequency and the percentage of suppression depending on the natural frequencies which is shown in the figures above. In LMS and NLMS algorithms, the mean values of the filter coefficients will keep converging to a more optimum solution. Hence, the filter coefficients will vary about their optimum values. The step size controls the amplitude of the fluctuations. The lower the value of step size, the smaller the fluctuations (less final maladjustments) and the slower the adaptive coefficients will converge to their optimal values.

References

[1] Tetsuo Semba, Matthew T. White, 'Seek Control to suppress vibrations of Hard Disk Drives using adaptive filtering' *IEEE/ASME Transactions on Mechatronics*, Vol. 13, No. 5, October 2008.

[2] S. Neduncheliyan, M. Umapathy, D. Ezhilarasi, 'Simultaneous periodic output feedback control for piezoelectric actuated structures using interval methods', *International Journal on Smart Sensing and Intelligent Systems*, Vol.2, No.3, 2009.

[3] Huang Quanzhen, Gao Zhiyuan, Gao Shouwei, Shao Yong, Zhu Xiaojin, 'Comparison of LMS and RLS Algorithm for Active Vibration Control of Smart Structures', International Conference on Measuring Technology and Mechatronics Automation - ICMTMA, 2011.

[4] L.R. Karl Marx, M. Umapathy, D. Ezhilarasi, 'Embedded Output Feedback Controllers for Piezoelectric Actuated Structures', *World Journal of Modelling and Simulation*, Vol. 5 No. 2, pp. 113-119, 2009.

[5] Komal R. Borisagar and G. R. Kulkarni, 'Simulation and Comparative Analysis of LMS and RLS Algorithms Using Real Time Speech Input Signal', *Global Journal of Researches in Engineering*, Vol.10 Issue 5, 2010.

[6] L.R. Karl Marx, M. Umapathy, A. Girija., D. Ezhilarasi, 'Discrete Time Sliding Mode Control Using Fast Output Sampling Feedback for Piezoelectric Actuated Structures', *Sensors & Transducers Journal*, Vol. 102, Issue 3 March 2009, pp.71-83.

[7] A. Madkour, M. A. Hossain, K. P. Dahal, H. Yu, 'Intelligent Learning algorithms for active vibration control', *IEEE transactions on Systems*, Vol. 37, No.5, 2007.

[8] S.H. Chen, Z.D. Wang and X.H. Liu, 'Active Vibration Control and Suppression for Intelligent Structures', *Journal of Sound and Vibration*, 1997, 200(2), pp.167-177.

[9] Rajiv Kumar, S P Singh and H N Chandrawat, 'Experimental adaptive vibration control of smart structures using LVQ neural network', *Journal of Scientific and Industrial Research,* Vol. 65, pp. 798-807, 2006.