ISSN (e): 2250 – 3005 || Volume, 08 || Issue, 12|| December – 2018 ||
International Journal of Computational Engineering Research (IJCER)

Novel Variable Step Size Complementary Pair Least Mean Square (VSS-CPLMS) Algorithm for System Identification

Saurabh R Prasad[1*], Bhalchandra B Godbole[2]
[1]Department of Electronics and Telecommunication Engineering, DKTE Society's Textile and Engineering Institute, Ichalkaranji, India
[2] Department of Electronics Engineering, KBP College of Engineering, Satara, India
* Corresponding Author: S R Prasad

ABSTRACT: *This paper presents the implementation of novel Complementary Pair Variable Step Size (CP-VSS-LMS) algorithm that uses two filters working in parallel, one with large step size called as speed mode filter and the other with small step size called as accuracy mode filter. This dual filter structure is originally proposed by MinSoo Park and Woo-Jin Song and later improved by Bilcu Radu C. by incorporating Variable step size for faster convergence and smaller misadjustment. But the algorithm proposed by them suffers from the drawback of increased computational complexity. Because of higher computational complexity, this algorithm has higher implementation cost which render it unsuitable especially for applications with long impulse response such as echo cancellation. To overcome this drawback, the proposed algorithm uses the combination of VSSLMS and signed LMS feature. In Signed LMS, the multiplication of coefficient update equation is replaced by addition, thus computational complexity is reduced significantly without significant increase in mean square error (MSE).*
KEYWORDS: *adaptive; least mean square; complementary pair; system identification.*

Date of Submission: 16-01-2019 Date of acceptance: 28-01-2019

I. INTRODUCTION

In fixed filter, the filter coefficients are calculated at design time, and they do not change thereafter whereas in adaptive filter the filter coefficients keep on changing continuously under the control of optimizing algorithm. The adaptive filtering involves two basic processes namely; filtering process and adaptation process which work interactively with each other. Thus an adaptive filter may be defined as, a filter that self adjusts its transfer function under control of an optimizing algorithm. Although the adaptive filters can be designed as linear and non linear, but due to simplicity in design and analysis linear adaptive filters are much more popular. As the power of digital signal processors has increased accompanied with decrease in its price adaptive filters have become commercially viable. Adaptive filters find applications in telecommunication, geophysical signal processing, biomedical signal processing, radar and sonar signal processing, mobile phones, MODEM, camcorders, digital cameras, and medical monitoring equipments etc.

An adaptive digital filter can be implemented either as FIR or IIR. FIR filters are characterized by advantages like linear phase response, guaranteed stability, no feedback requirement but also disadvantages like higher order and thus more computational time compared to their counterpart. However, the advantages of FIR filter outweigh their disadvantages, therefore adaptive filters are usually implemented as FIR filter. Adaptive algorithm allows the filter to learn the initial statistics of the input signal and to track them afterwards for any further changes. The error signal, which is the difference between primary signal and estimated output of adaptive filter, is used to tune filter coefficients in order to minimize the cost function. The most commonly employed cost function is the mean square of the error signal (MSE) [1]. The general plan of adaptive filter is represented in Fig. 1.

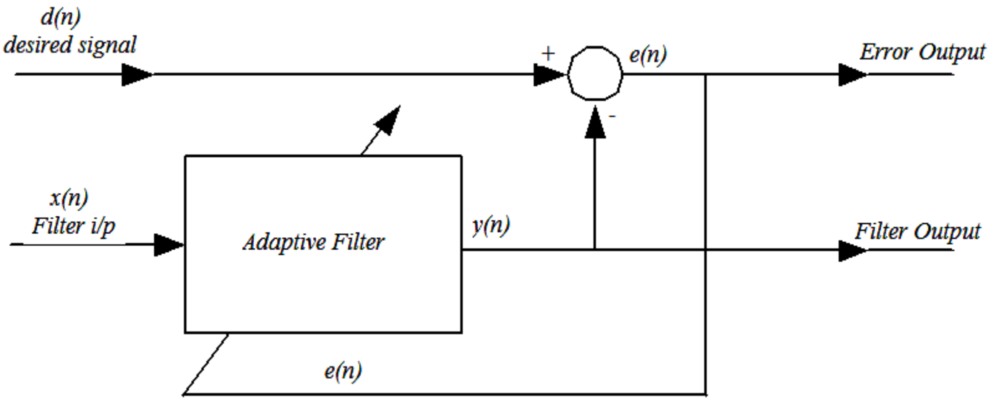

Fig. 1.Adaptive Filter Structure

The underlying FIR filter used in adaptive filter is most commonly a transversal filter which is also known as tapped delay line structure as shown in Fig. 2. This structure requires only delay line, adders, and multipliers as digital hardware resources. Here w_0, $w_1 \ldots w_{M-1}$ are filter coefficients and Z^{-1} represents delay line. The adaptive filtering algorithm iteratively adjusts filter coefficients to minimize the cost function.

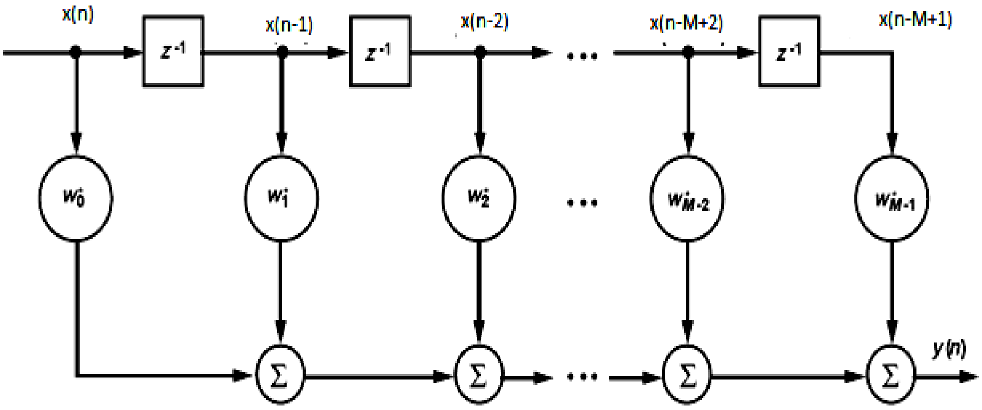

Fig. 2.Transversal FIR Filter

1.1 Signal Enhancement application- Biomedical Signal Processing:

There are various applications where fixed filters cannot work. When signal and noise have distinct spectrum and the characteristics of the signal, noise, and channel are known, fixed filters are the choice. On the other hand when there is a spectral overlap between signal and noise or when the characteristics of the input signal, noise, and dynamics of channel are unknown or change with time, fixed filters are of no use but adaptive filters only come for rescue. The practical example is fetal ECG as shown in Fig. 3 where fetal ECG is signal and mother's ECG and myographic signals are noise. Both have the same frequency spectrum and noise is stronger than the signal. Such examples need adaptive filters.

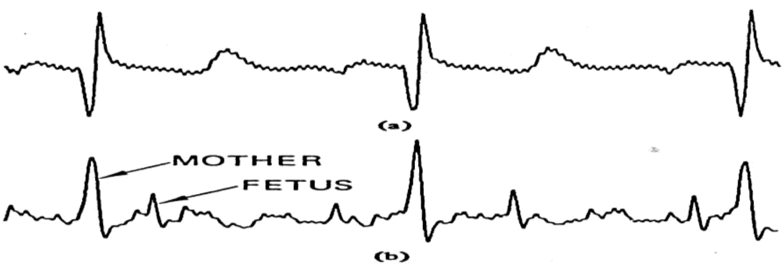

Fig.3. a) Clean ECG, b) ECG of fetus

1.2 Performance Parameters

Following are the performance parameters of the adaptive filters that may be used to evaluate the effectiveness of the improved algorithm.

1.2.1 **Convergence rate**:
This is defined as the number of iterations required for the algorithm to converge to its steady state mean square error in case of stationary input. Thus a faster rate of convergence means that it requires less no of iterations to converge. Usually real time applications require faster rate of convergence.

1.2.2 Computational complexity:
The computational complexity of the algorithm influences the price of the processor needed to implement the adaptive filter. It can be evaluated with the number of operations the algorithm requires to accomplish single iteration. In addition, the size of memory for storing the data and program is also another important consideration. The need for reducing the computational complexity comes from two aspects. Firstly, for the inherently demanding algorithm like RLS, the $O(L^2)$ complexity will occupy a large amount of computation and storage resources. Secondly, some applications like echo cancellation themselves are very much hardware demanding since in acoustic echo cancellation (AEC) problem, the echo paths spread over a relatively long length in time. For a sampling rate of 8 kHz, this would mean 800-2000 taps. In such situations, even the O (L) algorithm like NLMS is also much hardware demanding.

1.2.3 Numerical Robustness:
The implementation of adaptive filtering algorithms on a digital computer that has finite word-lengths, results in quantization errors. These errors sometimes can cause numerical instability of the adaptation algorithm. Adaptive filtering algorithm is numerically robust when its digital implementation using finite-word-length operations is stable.

1.2.4 Mathematical tractability:
An algorithm should be mathematically tractable, means that the convergence of the algorithm can be mathematically proved. Mathematical tractability is a desired feature of any adaptive algorithm.

1.2.5 Misadjustment:
As per the theory of optimal filter, no other filter than the optimal filter gives minimum MSE. The misadjustment parameter indicates a filter under consideration is how much closer to the optimal filter. The misadjustment is defined as excess MSE to minimum MSE. Misadjustment is given by following expression,

$$M = \frac{J_{ex}}{J_{\min}}$$

$$M = \frac{Jss - J\min}{J\min}$$

1.2.6 Stability
An algorithm is said to be stable if the mean-squared error converges to a finite value.

1.3 Applications of Adaptive Filters
There are following four categories of applications of adaptive filter.

1.3.1 **System Identification (Channel modeling)**
As shown in Fig. 4, system identification application requires both the unknown system and the adaptive filter driven by same input. After performing several iterations the filter estimates a linear model that closely matches with the unknown system.

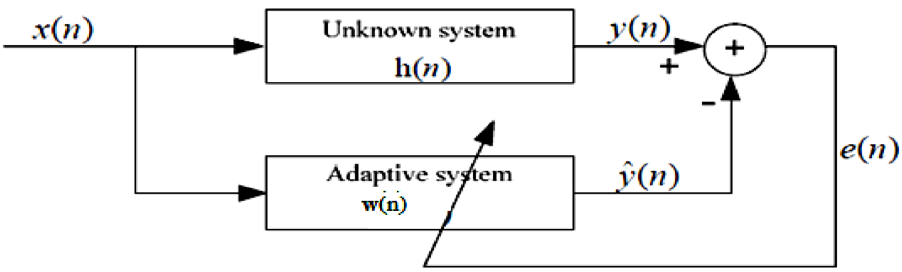

Fig.4. System Identification

1.3.2 **Equalization (Inverse Channel Modeling)**
As shown in Fig. 5, in equalization application unknown system and adaptive filter are connected in cascade and the input signal drives this cascade connection and delay line. To minimize the error signal, the adaptive

algorithm brings the transfer function of the adaptive filter closer to the inverse of the unknown system's transfer function. The adaptive equalizers are widely used in routers and repeaters.

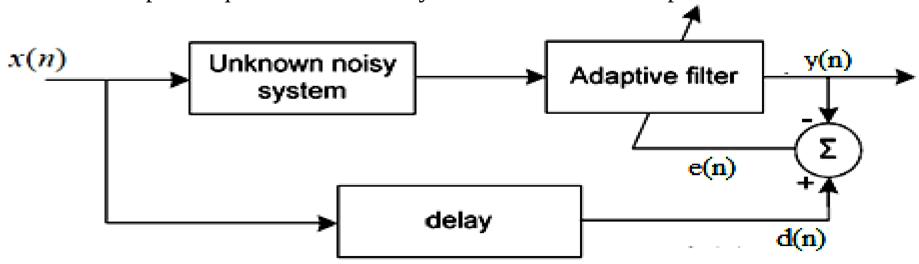

Fig.5. Equalization (Inverse Channel Modeling)

1.3.3 **Signal Enhancement (Noise cancellation)**
As shown in Fig. 6, in signal enhancement application, correlated noise is the input signal which drives the adaptive filter. The primary signal consists of a signal x(n) corrupted by an additive noise $w_1(n)$ and the filter input signal is $w_2(n)$ which is correlated with $w_1(n)$ but uncorrelated with x(n). The adaptive filter output is estimate of noise $w_1(n)$ present in primary signal which gets subtracted to produce clean sound. This application can be found in hearing aids and noise cancellation system in aircraft, and outdoor news recording.

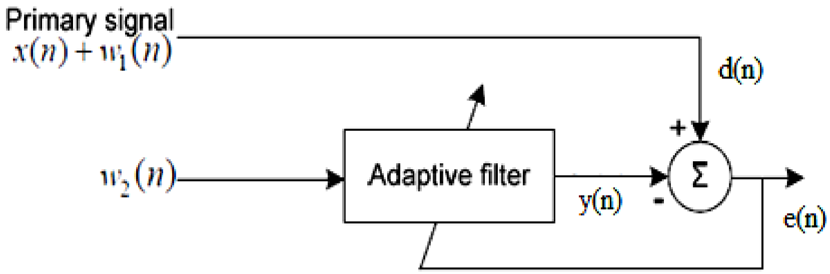

Fig.6. Signal Enhancement (Noise Cancellation)

1.3.4 **Prediction (Beamforming)**
If adaptive filter is configured as shown in Fig.7, it forms prediction system which is able to predict the present sample of the input signal using past values of the signal. This configuration is widely used in beamforming in smart antenna, where antenna can sense Direction of Arrival (DOA) of signal and for proper beam steering. Another application of this system could be in speech coding for prediction based compression.

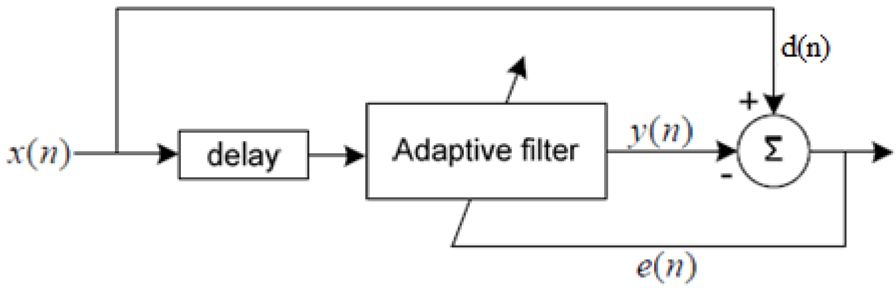

Fig.7. Prediction (Beamforming)

II. WEINER FILTER

Wiener mathematically solved the optimization problem of adaptive filter, and the resulting filter is known as Weiner filter or optimal filter. The output of FIR filter with transversal structure (convolution sum) is given in (1) where **w** is the filter coefficient and **x** is the input signal vector.

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

Error signal e(n) which is the difference between primary signal (desired signal) d(n) and filter output y(n) given in (2).

$$e(n) = d(n) - y(n) \quad (2)$$

$$J(\mathbf{w}) = E\{e^2(n)\} \quad (3)$$

Weiner uses Mean Square Error (MSE) as cost function of optimization given in (3), and derived that the optimization problem of minimization of cost function can be obtained by (4) where **R** is the correlation matrix of input signal and **p** is the cross correlation vector between input signal and error signal.

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \quad (4)$$

The optimal solution mentioned in (4) is the quadratic function of filter coefficients and thus gives unique local minima represented in Fig. 8 by bottom of the bowl shaped curve of MSE surface w.r.t first order filter coefficients w_0 and w_1.

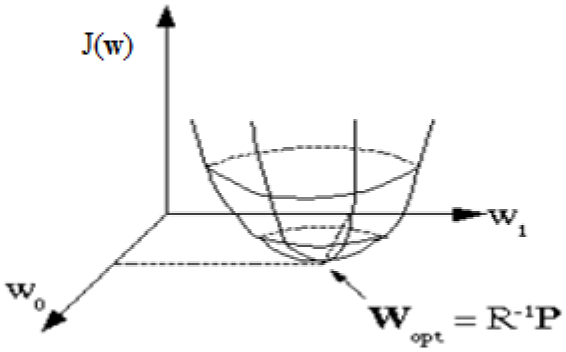

Fig. 8.Wiener solution for first order filter

Although Wiener gives one step optimal solution still it is of little practical significance because
1) It is computationally intensive as matrix inversion is required
2) It requires priori statistical information of the input signal in the form of autocorrelation matrix and cross correlation vector, which is generally not available
So instead of Weiner solution, all practical adaptive filters find approximate optimal solution by the method of iteration.

2.1 Gradient Search Algorithm (GSA)

This coefficient update equation of GSA is given in (5), where **g**(n) is the direction vector, which is $\mathbf{g} = -\mathbf{H}^{-1}\nabla\mathbf{J}$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{g}(n) \quad (5)$$

Thus,

$$\mathbf{w}(\mathrm{n}+1) = \mathbf{w}(\mathrm{n}) - \mathbf{H}^{-1}\nabla\mathbf{J}$$

(6)

Equation (6) is called as Newton's algorithm. From this, it is evident that Newton's algorithm converges in just one step if the Hessian matrix **H** and gradient $\nabla\mathbf{J}$ are known. The computation of inverse of **H** is computationally intensive. So, (6) cannot be practically used for real time applications. So, instead of direct computation of $\mathbf{H}^{-1}$, various iterative methods are used for approximation that results into different algorithms.

2.2 Steepest Descent Algorithm

Steepest Descent Algorithm (SDA) algorithm starts by assuming small initial filter coefficients (zero in most cases), and then by finding the gradient of the MSE cost function, the coefficients are updated iteratively at each step. That is, if the MSE gradient is positive, it implies the error is increasing positively, which indicates to reduce the weights. In the same way if the gradient is negative, it indicates to increase the weights. SDA approximates $\mathbf{H}^{-1}$ required in Newton's method by substituting $\mathbf{H} = 2\mathbf{I}$, and an additional parameter μ, called as convergence rate (step size) parameter is introduced to control the rate of convergence. The factor ½ is added for computational simplicity purpose only. Here Newton's method gets converted into (7),

$$\mathbf{w}(\mathrm{n}+1) = \mathbf{w}(\mathrm{n}) - \frac{\mu}{2}\nabla\mathbf{J} \quad (7)$$

But the gradient of cost function is given in (8)

$$\nabla\mathbf{J} = -2\mathbf{P} + 2\mathbf{R}\mathbf{w} \quad (8)$$

Coefficient update equation for SDA is obtained in (9) by substituting the gradient in (8)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \quad (9)$$

Equation (9) is called as SDA, and the flowchart of this is given in Fig.9. According to the quadratic relation between the MSE and the filter coefficient, it is guaranteed that the error performance surface always has minima and therefore, eventually reaches the minimum point after some iteration.

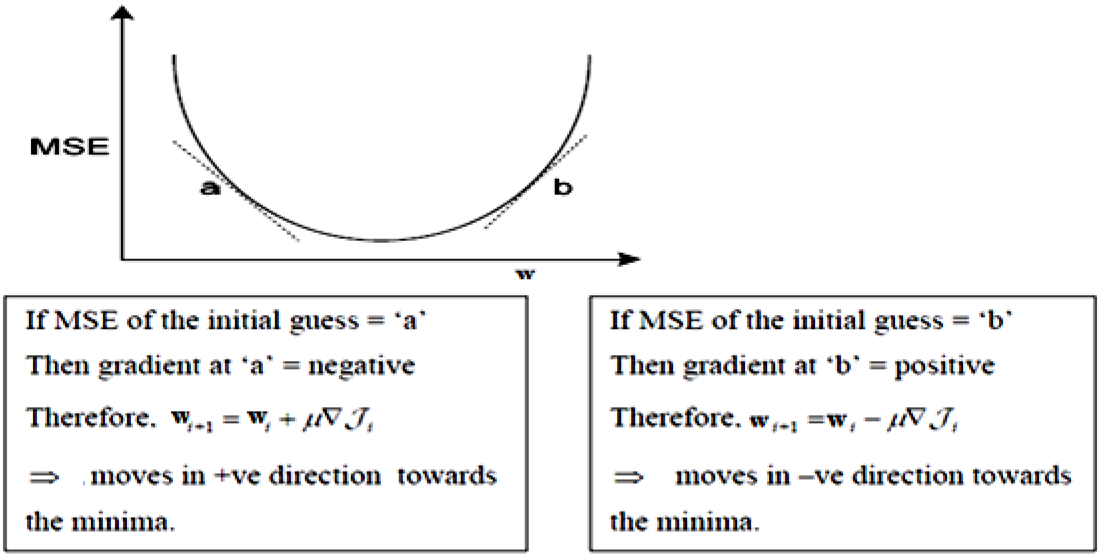

Fig.9. Steepest Descent Algorithm

The steps encountered in SDA are represented in the Flowchart of Fig.10.

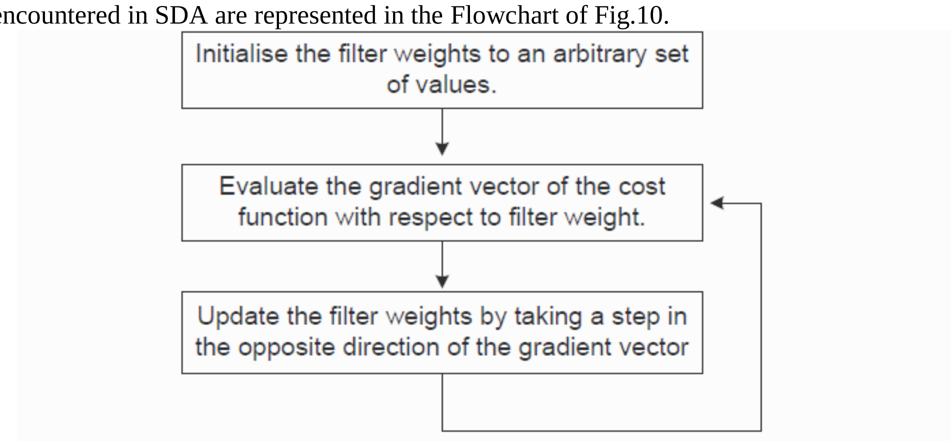

Fig.10. Flowchart of Steepest Descent Algorithm

2.3 LMS Algorithm:

The LMS algorithm has become immensely popular because performance, simplicity and stability of LMS algorithms outweigh other algorithms. The legendry LMS Algorithm was invented in 1959 by Bernard Widrow, and his first doctoral research scholar, Ted Hoff of Stanford University through their studies of pattern recognition. LMS algorithm stands as the benchmark against which all other adaptive filtering algorithms are judged. There are various variants of the LMS algorithm, and the original one mentioned in this section is called as standard LMS algorithm. In stationary environment, the LMS filter is expected to converge to the Wiener filter, and in non stationary environments, the filter is expected to track time variations of the signal and vary its filter coefficients accordingly. The LMS algorithm simply approximates autocorrelation matrix **R** and cross correlation vector **p** by replacing expected value with instantaneous value of the quantity as,

$\mathbf{R} = E[x(n)\, x^T(n)] \approx x(n)\, x^T(n)$

$\mathbf{p} = E[d(n)\mathbf{x}(n)] \approx d(n)\mathbf{x}(n)$

The filter coefficient update equation of LMS algorithm is as in (10)

$$\mathbf{w}(n+1)=\mathbf{w}(n)+ \mu\mathbf{x}(n)e(n) \qquad (10)$$

where,

w(n+1) : filter coefficient for next iteration
w(n) : filter coefficient at current iteration
μ : step size parameter
x(n) : filter input at current iteration
e(n) : error signal at current iteration

Now the computational complexity of LMS algorithm is discussed in following section.
A filter of N taps requires N+1 multiplication and N additions to implement weight update per iteration. The filtering process requires N multiplications and N-1 additions. The error equation generation requires one addition. Therefore total number of operations per iteration is 2N+1 multiplication and 2N addition and consequently this algorithm is called as O (N) algorithm.
There are various variants of the LMS algorithm, and the original one mentioned in this section is called as standard LMS algorithm. In stationary environment, the LMS filter is expected to converge to the Wiener filter, and in non stationary environments, the filter is expected to track time variations of the signal and vary its filter coefficients accordingly.
The basic adaptive algorithms those are widely used include LMS (Least Mean Square), and the RLS (Recursive Least Square). Though RLS algorithm gives speedy convergence, but it's computational complexity prevent it from wide scale popularity. The tradeoff between convergence speed and computational complexity of adaptive filtering turns LMS algorithm superior compared to RLS. It utilizes fewer computational resources and memory than RLS algorithms. The implementation of the LMS algorithms is less complicated than the RLS. One drawback of LMS algorithm is that it can search only local minima but not the global minima. However, by simultaneously starting the search at multiple points, this drawback can be overcome. The algorithm of Standard LMS algorithm is given as below [2].

Initialization: **w**(0)=**0** Algorithm: For n=0: iterations y(n)= $\mathbf{w}^T(n)\mathbf{x}(n)$ e(n)= d(n)-y(n) $\mathbf{w}(n+1)=\mathbf{w}(n)+ \mu\, \mathbf{x}(n)\, e(n)$ end

2.4 Stability Condition for LMS Algorithm

Stability and causality are the two basic requirements that must be fulfilled by any digital filter. This is because a non causal filter would be non-realizable and an unstable filter would be of no use. The LMS algorithm converges in mean square if and only if the step size parameter μ satisfy (11) where $\lambda_{\max}$ is largest eigenvalue of the autocorrelation matrix of input signal.

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{11}$$

But in practical situations, Eq. (11) is not much useful because the eigenvalues of the autocorrelation matrix are not known, and the more useful equation for step size parameter is given in (12).

$$0 < \mu < \frac{2}{\|x(n)\|^2} \tag{12}$$

The term $\|x(n)\|$ is called as Euclidean norm of the input signal vector whose squared term represents the power of the signal which is usually known or can be estimated a priori [3].

III. LMS ALGORITHM USING SPEED MODE AND ACCURACY MODE FILTER (CP-LMS)

The goal of filter design is to construct a computationally simple and numerically robust filter that can be implemented easily on a computer with faster convergence and smaller misadjustment. The proposed algorithm here targets mainly computationally simplicity issue. From theory of LMS algorithm, we know that convergence rate and misadjustment are inversely proportional to each other. Thus with standard LMS, we cannot achieve faster convergence with smaller misadjustment, and we are compelled to sacrifice one for other. But in parallel filter structure named as Complementary Pair LMS algorithm originally proposed by MinSoo Park and Woo-Jin Song, an attempt has been made to achieve both the desirable parameters simultaneously [11]. They have achieved this by using speed mode filter for faster convergence and accuracy mode filter for lower misadjustment. The speed mode filters is characterized by higher value whereas the accuracy mode filter is characterized by lower value of convergence rate parameter (μ) but both are within the limits given by (7). After every fixed number of iterations the MSE of both the filters are compared, and if the local average of the squared error of speed mode is lower than that of accuracy mode filter, then the later is updated with the coefficients of the speed mode filter.

The update equation of the speed filter is given in (13) where $\mathbf{w_1}$ (n), $\mathbf{x}$(n) are the vector of the filter coefficients and the vector of the input, respectively, e_1(n) is the error of the adaptive filter and $e_1(n) = d(n) - y_1(n)$.

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \mu_1 e_1(n)\mathbf{x}(n)\,; \tag{13}$$

The coefficients of the accuracy mode filter are updated by (14):

$$\mathbf{w}_2(n+1) = \left\{ \begin{matrix} \mathbf{w}_1(n+1) & \textit{if } n \bmod T = 0 \textit{ and } C(m) = 1 \\ \mathbf{w}_2(n) + \mu_2 e_2(n)\mathbf{x}(n) & \textit{otherwise} \end{matrix} \right\} \tag{14}$$

Here T is the length of the comparison interval, L is the filter length and C(m) is a criteria defined as in (15). The parameters T should not be too large or too small. Fig.12 represents schematics of system identification using speed mode and accuracy mode adaptive filter (CP-LMS/ VSS-CP-LMS/ Signed-VSS-CP-LMS)

$$C(m) = \left\{ \begin{matrix} 1 & \textit{if } \sum_{m}^{m+T} e_1^2(i) < \sum_{m}^{m+T} e_2^2 \\ 0 & \textit{otherwise} \end{matrix} \right\} \tag{15}$$

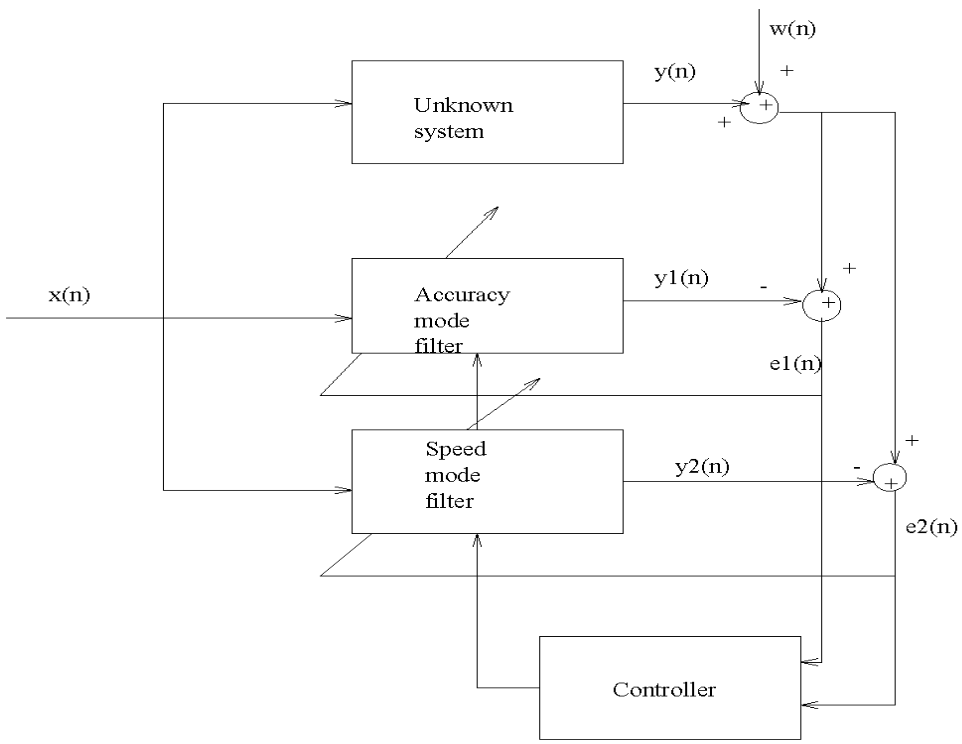

Fig.11 System Identification Using CP-LMS Algorithm

3.1 VSS LMS Algorithm Using Speed Mode and Accuracy Mode (CP-LMS)

Bilcu R.C. further improved this dual mode parallel filter structure by incorporating the concept of VSSLMS algorithm [4, 6]. In this they used fixed step size for Speed mode filter and variable step size for accuracy mode filter. This further improves rate of convergence as well as maintain small misadjustment. The equations encountered in this algorithm are given in steps 1-3.

1. The speed mode filter coefficients are updated by $\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \mu_1 e_1(n)\mathbf{x}(n)$
2. The accuracy mode variable step size filter is updated as,

$$\mathbf{w}_2(n+1) = \left\{ \begin{matrix} \mathbf{w}_1(n+1) & \textit{if } n \bmod T = 0 \textit{ and } C(m) = 1 \\ \mathbf{w}_2(n) + \mu_2(n) e_2(n)\mathbf{x}(n) & \textit{otherwise} \end{matrix} \right\}$$

3. The new step size of accuracy mode filter is,

$$\mu_2(n+1)=\begin{Bmatrix} \dfrac{\mu_1+\mu_2(n)}{2} \ , & if\ n \bmod T=0\ and\ C(m)=1 \\ \max\{\alpha\mu_2(n),\mu_3\} & otherwise \end{Bmatrix}$$

and

$$C(m)=\begin{Bmatrix} 1 & if \displaystyle\sum_{m}^{m+T} e_1^2(i) < \sum_{m}^{m+T} e_2^2 \\ 0 & otherwise \end{Bmatrix}$$

Here μ_3 is the minimum step size, and α is a parameter whose value is in the range of 0 to 1, which is selected as 0.5 in our experimentation. The basic idea of this algorithm is to increase the step-size when the algorithm converges slowly than speed mode, and decrease the step-size when algorithm is near the steady-state. The notation C used here denotes the expectations. The minimum step size μ_3 is chosen in such a way to achieve convergence within the given training signal length. The number of computation required per test interval for this algorithm is: 2T + 2N + 7 multiplications and 2T + 4N + 1 addition.

3.2 Proposed Modified VSS-CP-LMS Algorithm (Signed-VSS-CP-LMS Algorithm)
The above mentioned algorithm suffers from inherent drawback of higher computational complexity; so this algorithm is unsuitable to be used in applications with long impulse response such as echo cancellation. Similarly for other applications too like noise cancellation, system identification etc the implementation cost is higher. To overcome this drawback, the proposed algorithm uses the combination of VSS and Signed LMS feature. In Signed LMS, the multiplication of coefficient update equation is replaced by addition, thus computational complexity is reduced significantly without significant increase in MSE. Simulation results comparing this algorithm with the other parallel filter structure algorithms indicate there is no significant reduction in performance parameters inspite of reduced computations.

3.2 Sign-LMS
The sign-LMS algorithms replace the multiplication operation of standard LMS algorithms with addition operations. Thus it has the advantage of reduced complexity and fast computation compared to the standard LMS algorithm, which is obtained at the cost of small or insignificant reduction in convergence speed and higher value of misadjustment. The 'sign' function used in signed-LMS algorithm is as defined by the following equation.

$$sign(\mathrm{var})=\begin{Bmatrix} 1; & \mathrm{var}>0 \\ 0; & \mathrm{var}=0 \\ -1; & \mathrm{var}<0 \end{Bmatrix}$$

When sign function is applied to standard LMS algorithm, following three types of sign LMS algorithms are obtained.
3.2.1 Sign-error LMS: This algorithm in (16) updates the coefficients of an adaptive filter by applying sign function to the error signal e(n).

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\,\mathrm{sgn}(e(n))\,\mathbf{x}(n) \tag{16}$$

From this weight-update equation, it is observed that when e(n) is zero, this algorithm does not involve multiplication operations. When e(n) is not zero, this algorithm involves only one multiplication operation per iteration.
3.2.2 Sign-data LMS: This algorithm in (17) updates the coefficients of an adaptive filter by applying sign function to the input signal **x**(n).

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu e(n)\,\mathrm{sgn}(\mathbf{x}(n)) \tag{17}$$

From this weight-update equation, it is observed that when input signal **x**(n) is zero, this algorithm does not involve multiplication operations. When **x**(n) is non zero, this algorithm involves only one multiplication operation.
3.2.3 Sign-sign LMS:
This algorithm in (18) updates the coefficients of an adaptive filter by applying sign function to both e(n) and **x**(n). In simulation study presented here, we have used this form of signed algorithm.

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\,\mathrm{sgn}(e(n))\,\mathrm{sgn}(\mathbf{x}(n)) \tag{18}$$

From this weight-update equation, it is observed that when either e(n) or **x**(n) is zero, this algorithm does not involve multiplication operations. When neither e(n) nor **x**(n) is zero, this algorithm involves only one multiplication operation.

IV. SIMULATIONS RESULTS

The simulation study presents the results of system identification application of the proposed algorithm versus its previous counterparts. The unknown system weights are selected as {1, 2}, and the length of adaptive filter is equal to two; i.e. same as the length of unknown system to be identified. The simulation is carried out for 500 iterations and initial 200 iterations are considered as transient period and later iterations are considered for steady state analysis. The reference signal x(n) is generated by random function in Matlab and primary signal d(n) is generated by filtering the reference signal with the channel of unknown transfer function (h). In CPLMS algorithm, for speed mode filter, fixed and large step size equal to 0.0275 is used and for accuracy mode filter a small fixed step size equal to 0.0027 is used. In CPVSS algorithm, the speed mode filter has same step size as in previous case, however, the step size of accuracy mode filter changes continuously from iteration to iteration under the constraint that the minimum step size of it is 0.001. In simulation, used α is a parameter which is used to control the step size of accuracy ode filter which is equal to 0.5 and the iteration interval in which the speed mode filter and accuracy mode filter are compared to check which one performs better is, T=50. If in this interval speed mode filter performs better, then the coefficients of speed mode filter are copied in accuracy mode filter, which is a default filter. Otherwise, the accuracy mode filter updates as per normal LMS update equation. The simulation results are shown in Fig. 12 to Fig.15.

The graph of filter coefficients versus iteration in Fig.12 shows that CPLMS and VSSCPLMS takes approximately 100 iterations while modified algorithm Signed-VSSCPLMS takes approximately 100 iterations to converge to the value of coefficients of unknown system. Considering practical scenario 100 iterations is acceptable value for convergence. At the same time it is observed that the steady state value of convergence of proposed algorithm is comparable to its predecessor high computation demanding algorithms. Thus it can be concluded that inspite of reduced complexity; there is no appreciable difference in proposed algorithm and predecessors in the estimation of unknown system coefficients.

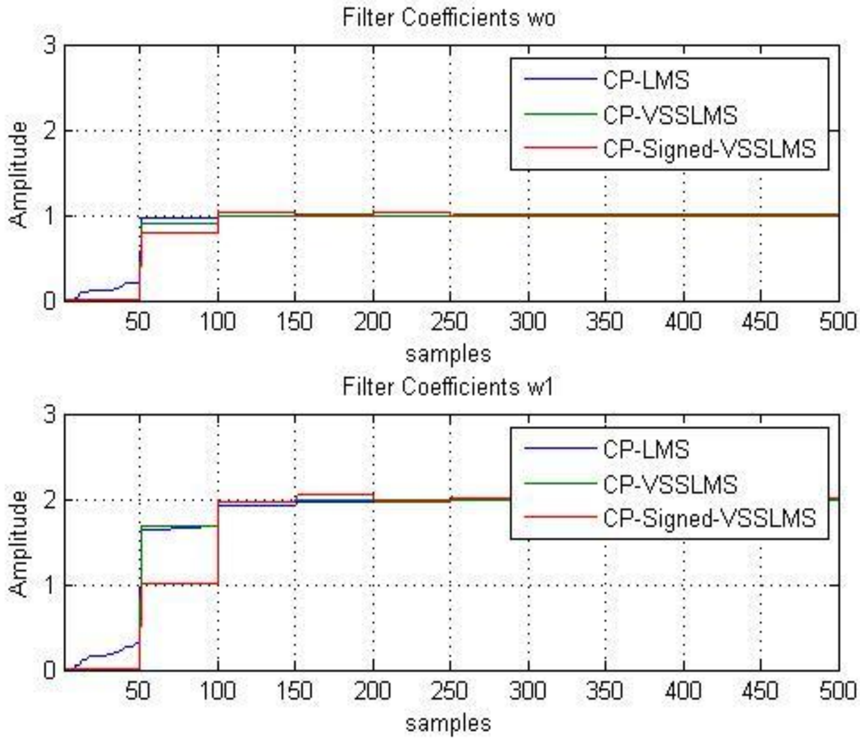

Fig.12. Filter Coefficient vs. Iteration

The mean square error versus iteration graph is represented in Fig. 13 which also indicates that approximately after 100 iterations the MSE value becomes negligible. Thus it can be concluded that the proposed algorithm converges to steady state MSE almost identically well compared to its predecessors.

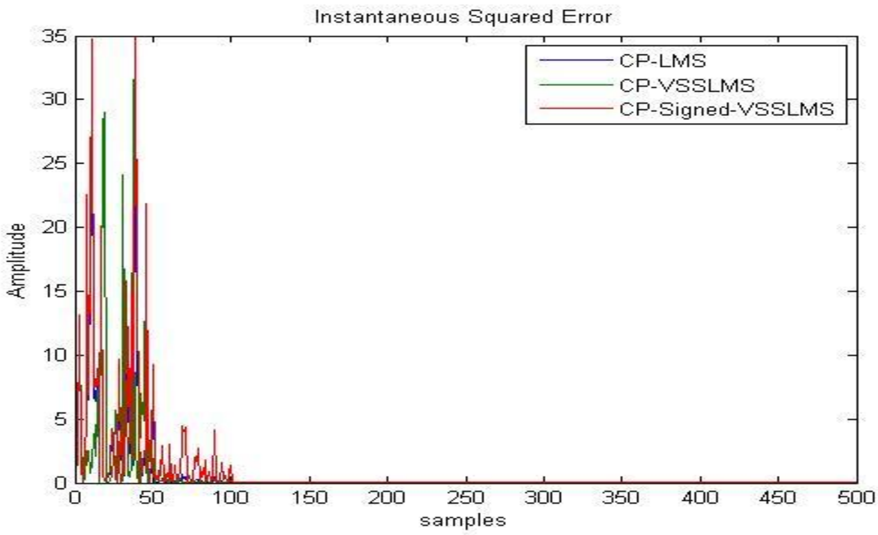

Fig.13. MSE vs. Iteration

The estimation error versus iteration graph in Fig. 14 indicates that approximately after 100 iterations estimation error becomes negligible as the adaptive filter coefficient converges to unknown system coefficient. Thus the error signal is also almost equal for all three algorithms under study, and it tends to zero.

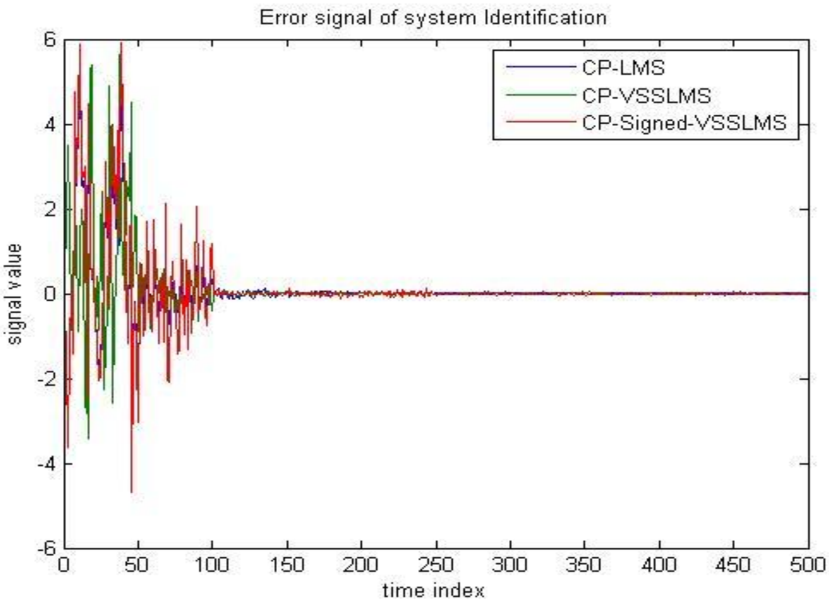

Fig.14. Estimation Error for Three Different Implementations

The final value of estimated coefficient is shown in Fig. 15 which indicates that the actual and estimated weight obtained at the end of iteration cycle also overlaps with each other and that is 99.99% accurate.

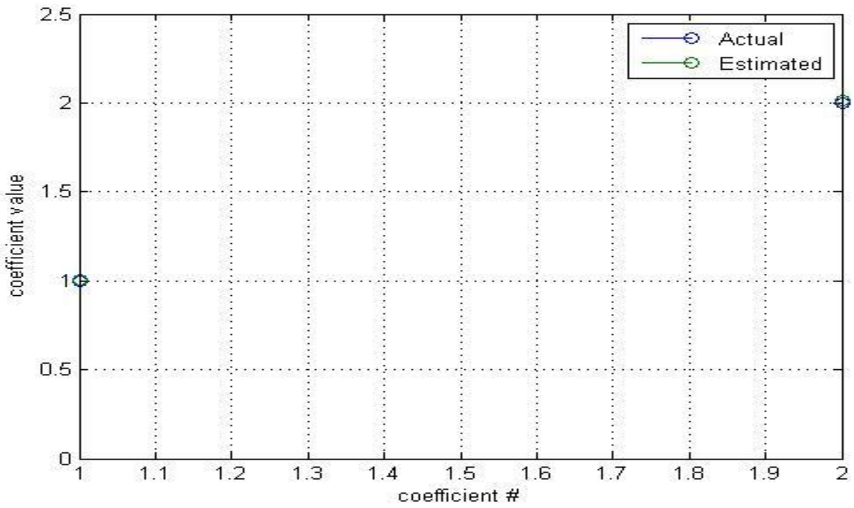

Fig.15. Actual vs. Final Estimated Weight

Table 1.Significant Parameters of Adaptive Filter for Proposed Algorithm and its processors

Parameter	**CPLMS**	**CPVSSLMS**	**Proposed Algorithm Signed CPVSSLMS**
Steady state MSE	0.000009	0.000001	0.000767
Actual coefficients of first order unknown system	{1,2}	{1,2}	{1,2}
Estimated coefficient wo after 500 iterations	0.999998	1.000001	1.005155
Estimated coefficient w1 after 500 iterations	1.999998	1.999999	2.015765
No of iterations required for to achieve misadjustment under 1%	201	180	195

The data represented in Table. 1 shows that inspite of reduced complexity, the performance parameters of proposed algorithms are not degraded to any significant scale.

V. CONCLUSIONS

The major disadvantage of standard LMS algorithm is with this algorithm, faster convergence cannot be obtained along with smaller misadjustment. To overcome this limitation, a parallel structure of VSS-LMS algorithm, named as VSSCPLMS algorithm is put forth by earlier researchers. Though it can obtain faster convergence with low misadjustment but suffer from disadvantage of greedy computational requirement because of inherent computational complexity due to parallel structure. To address this issue, proposed algorithm uses signed-LMS concept which replaces multiplication in coefficient update equation by addition and reduce the complexity. Simulation results show that inspite of reduced complexity, the proposed algorithm keeps performance at par with the earlier algorithm.

REFERENCES

[1]. Diniz, Paulo SR. Adaptive filtering. Springer, 1997.
[2]. Widrow, Bernard. "Thinking about thinking: the discovery of the LMS algorithm." IEEE Signal Processing Magazine 22.1 (2005): 100-106.
[3]. Guo, Lei, Lennart Ljung, and Guan-Jun Wang. "Necessary and sufficient conditions for stability of LMS." IEEE transactions on automatic control 42.6 (1997): 761-770
[4]. Bilcu, Radu Ciprian, Pauli Kuosmanen, and Karen Egiazarian. "A transform domain LMS adaptive filter with variable step-size." IEEE Signal Processing Letters 9.2 (2002): 51-53.
[5]. Min-Soo Park and Woo-Jin Song, A Complementary Pair LMS Algorithm for Adaptive Filtering, IEICE Trans. Fundamentals, vol. E81-A, no. 7, pp. 14931496,July, 1998.
[6]. Bilcu, Radu Ciprian, Pauli Kuosmanen, and Corneliu Rusu. "Improving performances of complementary pair LMS algorithm." Signal Processing Conference, 2000 10th European. IEEE, 2000.