

FIR System Identification Using Feedback

T. J. Moir

School of Engineering, AUT University, Auckland, New Zealand.
Email: Tom.Moir@aut.ac.nz

Received September 20th, 2013; revised October 18th, 2013; accepted October 25th, 2013

Copyright © 2013 T. J. Moir. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This paper describes a new approach to finite-impulse (FIR) system identification. The method differs from the traditional stochastic approximation method as used in the traditional least-mean squares (LMS) family of algorithms, in which we use deconvolution as a means of separating the impulse-response from the system input data. The technique can be used as a substitute for ordinary LMS but it has the added advantages that can be used for constant input data (*i.e.* data which are not persistently exciting) and the stability limit is far simpler to calculate. Furthermore, the convergence properties are much faster than LMS under certain types of noise input.

Keywords: LMS; Feedback; System Identification

1. Introduction

Recursive parameter (or weight) estimation has been used since the early days of least-mean squares (LMS) [1]. Along with its variants such as Normalised-LMS (NLMS) [2], it has become the adaptive algorithm of choice due both to its simplicity, good tracking ability and the fact that it applies to finite-impulse-response (FIR) filters which are always stable. There have been many attempts at adaptive infinite-impulse-response (IIR) [3,4] filters using algorithms such as recursive-least-squares (RLS) [2,5], but they are less popular due to stability problems. Besides this, the tracking ability of RLS is rather *ad-hoc* in nature using a forgetting factor and the algorithm complexity for L weights is $O(L^2)$ operations rather than $O(2L+1)$ for LMS [6]. Adaptive filters are proven successful in a wide range of signal processing applications such as noise-cancellation [7,8], adaptive arrays [9], time-delay estimation [10], echo cancellation [11] and channel equalization [12].

Despite the popularity of LMS it has a few drawbacks. To name a few, LMS will only give unbiased estimates when the driving signal is rich in harmonics (a persistently exciting input signal [13]) and when this driving signal is not white, the convergence of the algorithm is affected depending on the eigen-value spread of the correlation matrix [2,14,15]. Usually many *ad-hoc* approaches that employ variable step-size have been used to try and overcome this problem [16]. Gradient based algo-

rithms such as LMS are based on steepest-descent and do not have as fast convergence as RLS. The literature is quite old for many of these approaches indicating that little has changed in many respects though there have been some more recent approaches to the same problem [17]. This paper addresses these problems by introducing a new concept in weight estimation which is not based on minimisation of any form of least-squares criterion, recursive or otherwise. Instead the paper uses classical control-theory to separate the convolution of weights to system input by the use of high-gain and integrators. Although LMS also use high-gain and integrators with feedback, the LMS approach is always geared towards correlation and the solution of the Wiener-Hopf equation. This is in fact the reason for some of its limitations on special cases. The approach used here is entirely deterministic in nature and instead based on purely control theory. The control-loop used results in deconvolution, or the separation of the two convolved signals (the impulse response and the driving signal). The novelty in the solution is the fact that a special lower-triangular convolution matrix is used in the feedback path of this control system.

2. Feedback Deconvolution Loop 1

Consider an unknown system $W(z^{-1})$ driven by a known input signal u_k which can be either random or deterministic

$$s_k = W(q^{-1})u_k \tag{1}$$

Although the system to be identified is single-input single-output, the method here works on a vector of outputs consisting of $n + 1$ consecutive outputs.

We define here z^{-1} as the z -transform operator and q^{-1} is the backwards shift operator defined for a scalar discrete signal at sample instant k as $y_{k-1} = q^{-1}y_k$ or for a vector $\mathbf{x}_{k-1} = q^{-1}\mathbf{x}_k$. A vector output is obtained from the block convolution “*” of the system input and the impulse response vector thus

$$s_k = W * u_k \tag{2}$$

where the weight and input vectors are defined respectively as: $W = [w_0, w_1, \dots, w_n]^T$, $u_k = [u_k, u_{k-1}, \dots, u_{k-n}]^T$ and are of order $n + 1$ each giving

$$W * u_k = \sum_{j=0}^k w_j u_{k-j} \tag{3}$$

From the convolution Equation (3), we consider only the first $n + 1$ terms. s_k becomes a vector of dimension $n + 1$ written in terms of past values of k as

$$s_k = [s_k, s_{k-1}, \dots, s_{k-n}]^T.$$

A matrix format of convolution now follows accordingly:

$$s_k = T(u_k)W_k \tag{4}$$

Here $T(u_k)$ is an $n + 1$ square lower-triangular Toeplitz matrix given by

$$T(u_k) = \begin{bmatrix} u_k & 0 & 0 & 0 & 0 & 0 \\ u_{k-1} & u_k & 0 & 0 & 0 & 0 \\ u_{k-2} & u_{k-1} & u_k & 0 & 0 & 0 \\ u_{k-3} & u_{k-2} & u_{k-1} & u_k & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & u_k & 0 \\ u_{k-n} & u_{k-(n-1)} & u_{k-(n-2)} & \dots & \dots & u_k \end{bmatrix} \tag{5}$$

This will be known as a *convolution matrix*. This is distinct from a *correlation matrix* which is met in least-squares problems.

Now consider a time-variant multivariable control-system as shown in **Figure 1**. The control-loop forward-path consists of a matrix of integrators with gain K . Its error vector is defined as

$$e_k = s_k - y_k = s_k - T(u_k)\hat{W}_k \tag{6}$$

The \hat{W}_k vector represents the estimate of the true weight vector W_k . Assume that the closed-loop multivariable system is stable. Then the error for large gain K becomes (via the usual action of negative feedback) $e_k \rightarrow 0$ so that $T(u_k)\hat{W}_k \rightarrow s_k$ and hence $\hat{W}_k \rightarrow W$ as $k \rightarrow \infty$. If we initially assume the closed-loop system can be made always stable then for a time-varying system, the control-loop output must track any such changes.

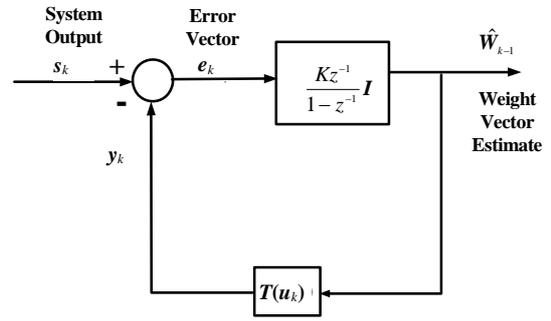


Figure 1. Multivariable System 1.

In algorithm form, the method is quite simple.

Furthermore, if s_k and u_k are reversed as inputs to the control-system, then the inverse of the system impulse response will be estimated instead, provided of course enough weights are allocated. The above algorithm is not optimal in any least-squares sense since no cost function is minimised. In fact this is an approach to deconvolution using feedback. Note that in **Figure 1** integrators are chosen since they have infinite gain at dc (giving zero steady-state error to step inputs of weight-vector), but other types of controller are possible. Although not explored here, it is possible to include loop-shaping to the control-loop by adding extra integrators and phase-lead compensation. This was explored elsewhere for the ordinary LMS algorithm [18].

3. Stability and Convergence of the Loop

For a given batch of $n + 1$ samples it is fairly straightforward to determine the stability of the system of Section 2.

Figure 1 is a time-variant multivariable system, but for a given constant vector u_k we can treat the closed-loop system as time-invariant. (A more complete explanation is shown in Appendix when the input vector is time-varying). Calculating an expression for the error vector in the z -domain

$$e(z^{-1}) = s(z^{-1}) - z^{-1}T(u_k)\hat{W}(z^{-1}) \tag{7}$$

and

$$\hat{W}(z^{-1}) = \frac{Kz^{-1}}{1 - z^{-1}}e(z^{-1}) \tag{8}$$

Substitute (8) into (7) and re-arrange

$$\left[I + T(u_k)\frac{Kz^{-1}}{1 - z^{-1}} \right] e(z^{-1}) = s(z^{-1}) \tag{9}$$

Simplifying further

$$\left[I(1 - z^{-1}) + T(u_k)Kz^{-1} \right] e(z^{-1}) = (1 - z^{-1})s(z^{-1}) \tag{10}$$

The roots for z of the determinant of the return-difference matrix are the closed-loop poles of the system

satisfying[19]:

$$\rho_c(z) = \det[z\mathbf{I} - (\mathbf{I} - \mathbf{T}(\mathbf{u}_k)\mathbf{K})] = 0 \quad (11)$$

That is, the closed-loop characteristic polynomial of the matrix $\mathbf{I} - \mathbf{T}(\mathbf{u}_k)\mathbf{K}$ must have roots (eigen-values) which lie within the unit-circle on the z -plane. Now since $\mathbf{T}(\mathbf{u}_k)$ is a lower-triangular Toeplitz matrix, it follows $\mathbf{I} - \mathbf{T}(\mathbf{u}_k)\mathbf{K}$ is also a lower-triangular Toeplitz matrix. Furthermore, it is well established that the eigen-values of such a matrix are just the its diagonal elements which in this case are the $n + 1$ repeated roots according to $\rho_c(z) = (z - 1 + u_k K)^{n+1} = 0$ giving

$$z_i = 1 - u_k K, i = 1, 2, \dots, n+1 \quad (12)$$

For stability all roots must lie within the unit circle in the z -plane. This gives us

$$|1 - u_k K| < 1 \quad (13)$$

Excluding the special case when $u_k = 0$, the gain K must satisfy:

$$0 < u_k K < 2 \quad (14)$$

Equation (15) clearly poses no problem provided u_k and K are always positive. Then

$$0 < K < \frac{2}{u_k} \quad (15)$$

However, if u_k is negative then K (from 14) must also be negative for stability. This will require a time-varying K whose sign tracks the same sign as u_k . It is interesting to see that the stability is only dependant on the single input u_k and is also independent of the system order. This is different from the LMS case where the step-size depends on the variance of the input signal and the number of weights. Of course u_k is renewed with every new buffer of data and so it is not just one value which the stability is dependent on.

Now we assume that the true weight vector has a constant but unknown set of weights, say \mathbf{W}_0 . This is modelled by a step z -transform $\mathbf{W}_0 \frac{1}{1 - z^{-1}}$.

From **Figure 1**, the weight vector estimate is found from the multivariable system according to,

$$\hat{\mathbf{W}}(z^{-1}) = \frac{K}{1 - z^{-1}} \mathbf{e}(z^{-1}) \quad (16)$$

and the error vector is found from

$$\mathbf{e}(z^{-1}) = \mathbf{s}(z^{-1}) - z^{-1} \mathbf{T}(\mathbf{u}_k) \hat{\mathbf{W}}(z^{-1}) \quad (17)$$

Substitute (17) into (16) and re-arranging, gives the multivariable transfer function matrix relating the estimated vector with signal vector.

$$\hat{\mathbf{W}}(z^{-1}) = [\mathbf{I} - (\mathbf{I} - \mathbf{K}\mathbf{T}(\mathbf{u}_k))z^{-1}]^{-1} \mathbf{K}\mathbf{s}(z^{-1}) \quad (18)$$

which as previously, requires all the eigen-values of the matrix $\mathbf{I} - \mathbf{T}(\mathbf{u}_k)\mathbf{K}$ to lie within the unit-circle on the z -plane. With stability satisfied via (15), we can now examine the convergence by applying the step input vector in the z -domain thus: $\mathbf{W}(z^{-1}) = \mathbf{W}_0 \frac{1}{1 - z^{-1}}$

We can write in (18) that

$$\mathbf{s}(z^{-1}) = \mathbf{T}(\mathbf{u}_k)\mathbf{W}(z^{-1})$$

and by applying a step input vector in the z -domain thus:

$$\mathbf{W}(z^{-1}) = \mathbf{W}_0 \frac{1}{1 - z^{-1}}$$

So that (18) becomes

$$\hat{\mathbf{W}}(z^{-1}) = [\mathbf{I} - (\mathbf{I} - \mathbf{K}\mathbf{T}(\mathbf{u}_k))z^{-1}]^{-1} \mathbf{K}\mathbf{T}(\mathbf{u}_k)\mathbf{W}_0 \frac{1}{1 - z^{-1}} \quad (19)$$

By applying the z -transform final-value theorem to (19) we get

$$\lim_{k \rightarrow \infty} \hat{\mathbf{W}}_k = \lim_{z \rightarrow 1} \hat{\mathbf{W}}(z^{-1})(1 - z^{-1}) \quad (20)$$

$$\hat{\mathbf{W}}_k \rightarrow \mathbf{W}_0 \quad (21)$$

So that the weights converge to the true values provided the closed-loop system is stable.

Algorithm 3.1: Deconvolution Loop 1. Select magnitude of the loop gain K_0 .

Loop: {Fetch recent input and output data vectors \mathbf{u}_k , \mathbf{s}_k using at least $n + 1$ samples. Monitor the first sample \mathbf{u}_k within the vector \mathbf{u}_k and make $K = K_0 \text{sgn}(u_k)$, 1) *Update vector error:* $\mathbf{e}_k = \mathbf{s}_k - \mathbf{T}(\mathbf{u}_k)\mathbf{W}_{k-1}$, 2) *Update Weight Vector:*

$$\hat{\mathbf{W}}_k = \hat{\mathbf{W}}_{k-1} + \mathbf{K}\mathbf{e}_k \}$$

where in the above $\mathbf{T}(\mathbf{u}_k)$ is formed by Equation (5). For L weights, the algorithm has $O(L^2)$ operations.

4. Feedback Deconvolution Loop 2

The problem with the control loop discussed in section 2 is that the stability satisfies (15) which makes the loop gain dependent on the sign of u_k , the first sample of each vector of input data fed to the unknown system. This means that the input sign must be monitored and K switched in sign. A slight modification can overcome this problem. Consider **Figure 2**.

It can be seen that the convolution matrix $\mathbf{T}(\mathbf{u}_k)$ has now been added to the forward path as well as the feedback path. We now have

$$\hat{\mathbf{W}}(z^{-1}) = \frac{K}{1 - z^{-1}} \mathbf{T}(\mathbf{u}_k) \mathbf{e}(z^{-1}) \quad (22)$$

and

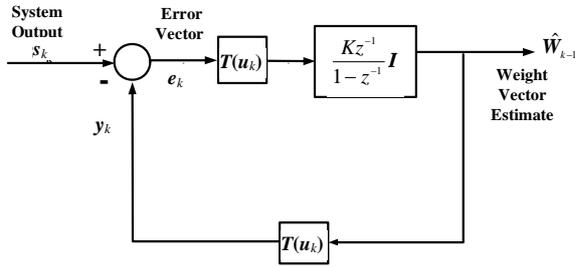


Figure 2. Multivariable System 2.

$$e(z^{-1}) = s(z^{-1}) - z^{-1}T(u_k)\hat{W}(z^{-1}) \quad (23)$$

Substituting the error vector (23) into (22) and following a similar approach to Section 2, we find the relationship

$$\hat{W}(z^{-1}) = [I - (I - KT(u_k))z^{-1}]^{-1} KT(u_k)s(z^{-1}) \quad (24)$$

and using the z -transform of (4)

$$\hat{W}(z^{-1}) = [I - (I - KT^2(u_k))z^{-1}]^{-1} KT^2(u_k)W(z^{-1}) \quad (25)$$

The stability becomes the solution of the roots of the polynomial found from the return-difference matrix $I - (I - KT^2(u_k))z^{-1} = 0$. Here however we have the product of two lower-triangular matrices $T^2(u_k)$. The product of two such identical lower-triangular matrices is a matrix of the form

$$T^2(u_k) = \begin{bmatrix} u_k^2 & 0 & 0 & 0 & 0 & 0 \\ x & u_k^2 & 0 & 0 & 0 & 0 \\ x & x & u_k^2 & 0 & 0 & 0 \\ x & x & x & u_k^2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & u_k^2 & 0 \\ x & x & x & x & \cdots & u_k^2 \end{bmatrix} \quad (26)$$

(where the x represents cross-terms) which is another lower-triangular matrix with diagonal elements u_k^2 . Following the same arguments as section (2) we can easily show that the condition for stability of the multivariable loop is now

$$0 < K < \frac{2}{u_k^2} \quad (27)$$

which makes the gain always positive. Furthermore from (25), for constant weights we can also show that the weights converge to their true values asymptotically $\hat{W}_k \rightarrow W_0$.

LMS algorithm which has a condition for convergence in the mean-square equivalent to

$$0 < \mu < \frac{2}{(n+1)\sigma^2}$$

where σ^2 is the variance of the input driving noise and μ is the step-size. Clearly (27) when taken over a large number of batches of data and the input driving signal is random, becomes

$$0 < \mu < \frac{2}{\sigma^2},$$

which is not dependent on the system order n .

Algorithm 4.1: Deconvolution Loop 2; Select magnitude of loop gain μ .

Loop: {Fetch recent input and output data vectors u_k , s_k using at least $n+1$ samples. Monitor the first sample u_k within the vector u_k and make $K = \mu$ where

$$0 < \mu < \frac{2}{u_k^2}$$

1) Update vector error: $e_k = s_k - T(u_k)W_{k-1}$, 2) Update Weight Vector:

$$\hat{W}_k = \hat{W}_{k-1} + KT(u_k)e_k$$

where in the above $T(u_k)$ is formed by Equation (5). For L weights the algorithm also has $O(L^2)$ operations.

5. Illustrative Examples

Example 1: Consider an FIR system with three unknown weights:

$$W(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} = 1 - 1.5z^{-1} + 0.5z^{-2}$$

Let the system be driven by unity-variance white noise which is dc-free. Select a forward path gain (and LMS step-size) of $K = 0.2$.

Figures 3-5 show the weight convergence of Algorithms 3.1, 4.1 and LMS respectively for zero additive measurement noise.

It can be seen that the new algorithms have overshoot but the convergence time is very similar. Algorithm 4.2 has around 90% overshoot than Algorithm 3.1. If we compare a norm of the weight-error $\|\mathbf{e}_k\| = \|\mathbf{W}_{k-1} - \hat{\mathbf{W}}_{k-1}\|$ for each case we see the comparison in Figure 6.

LMS gives the fastest performance of the three algo-

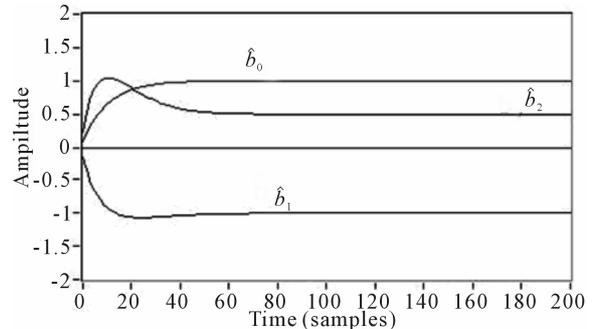


Figure 3. Weight convergence for Algorithm 3.1. No measurement noise.

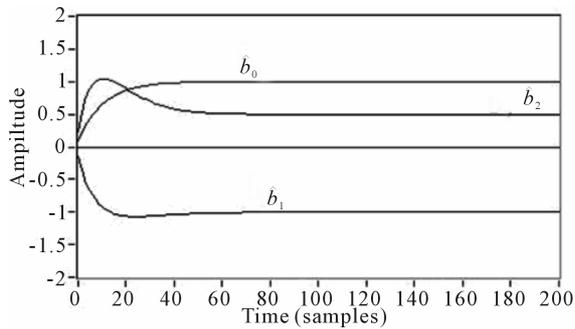


Figure 4. Weight convergence for Algorithm 4.1. No measurement noise.

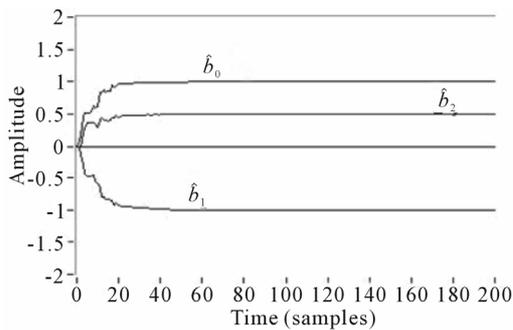


Figure 5. Weight convergence for LMS algorithm. No measurement noise.

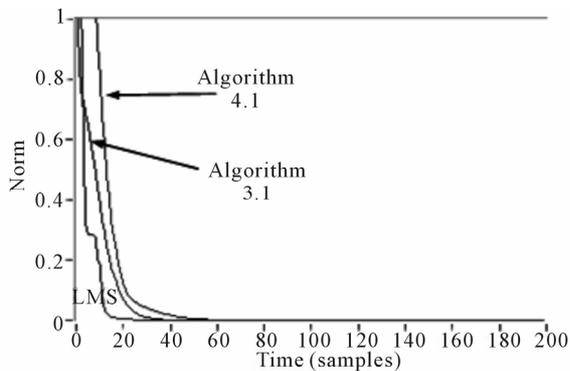


Figure 6. Comparison of weight-error norm for zero measurement noise.

gorithms. The norm is taken rather than mean-square error because the new algorithms have vector-based errors as opposed to LMS which has a scalar error.

If we add zero-mean uncorrelated white measurement noise for a SNR of 12dB and repeat the above simulation we get some interesting results. In Figures 7-9 it is seen that the LMS estimates are not as smooth as the new algorithms.

The smooth convergence is illustrated in Figure 10 by comparing the weight-error norms. There are large error fluctuations as compared with the feedback algorithms which give similar much smoother performance. Of course the LMS case can be much improved by lowering

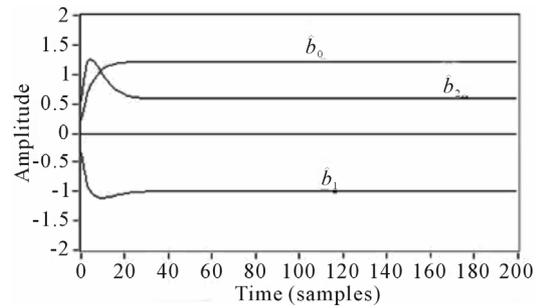


Figure 7. Weight convergence for Algorithm 3.1. 12dB SNR.

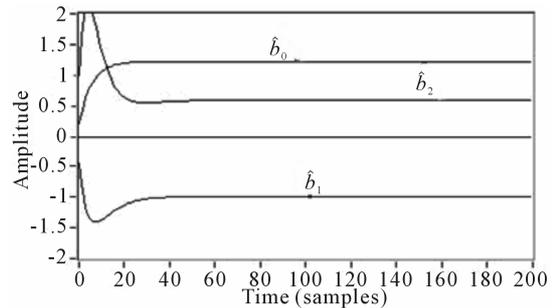


Figure 8. Weight convergence for Algorithm 4.1. 12dB SNR.

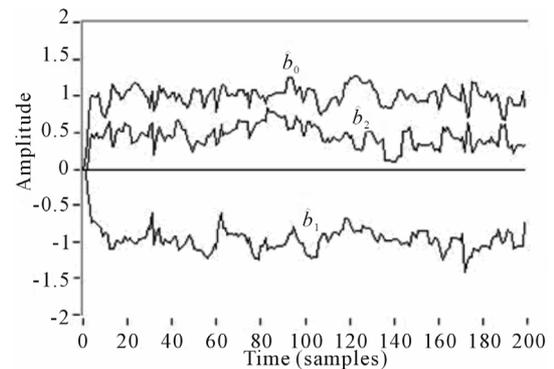


Figure 9. Weight convergence for LMS algorithm. 12dB SNR.

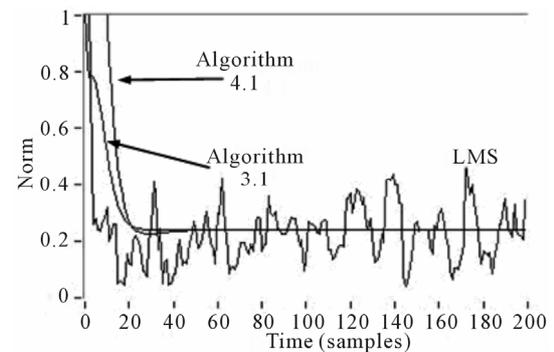


Figure 10. Comparison of error norm for 12dB measurement noise and $K = 0.2$.

the step-size at the expense of a much slower convergence rate. Figure 11 shows a comparison of the weight-

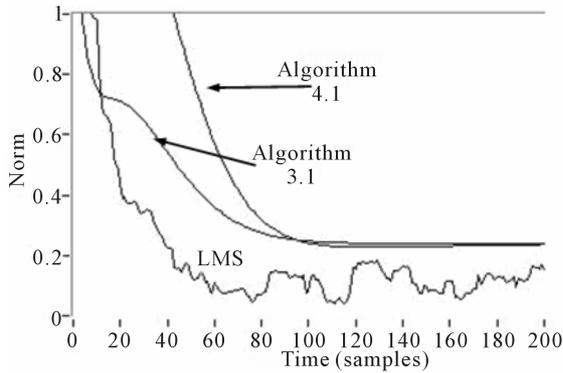


Figure 11. Comparison of weight-error error norm for 12dB measurement noise and gain reduction $K = 0.05$.

error norms for a gain $K = 0.05$.

LMS is the superior of the three if convergence rate is sacrificed but still has noisy fluctuations in the weight-error norm.

We can conclude from this example that the new algorithms give smoother weight estimates than LMS but LMS can outperform the new algorithms when the measurement noise is zero provided the loop gain is not too high.

Example 2: It is well established that ordinary LMS has problems when the eigen-value spread of the correlation matrix is large [2]. This leads to slow convergence and no amount of increasing the step-size will help since instability will result. Consider a system with zero measurement noise and with two weights to be estimated

$$W(z^{-1}) = b_0 + b_1 z^{-1} = 1 - 0.5z^{-1}$$

which is driven by filtered noise u_k by passing a white-noise signal ζ_k of variance 0.003025 through an autoregressive filter of order two.

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} + \zeta_k$$

The parameters of the autoregressive model are chosen as $a_1 = 1.9114$ and $a_2 = -0.95$ to make the eigenvalue spread of the correlation matrix $\chi(\mathbf{R}_{uu}) = 100$ [2]. (For the previous example we can say that $\chi(\mathbf{R}_{uu}) = 1$) The driving white noise ζ_k is suitably scaled so that the filtered variance of u_k is unity. The step-size (gain) of the loop was set initially to $K = 0.2$, which was the maximum step-size that the LMS algorithm could tolerate without becoming unstable. **Figure 12** shows the LMS estimates when the step-size is at its maximum that it can tolerate without instability.

The LMS algorithm takes about 1000 steps to converge. If the step-size increases further then the LMS algorithm becomes unstable. Whereas if we look at **Figure 13** we can see that Algorithm 3.1 is perfectly stable with gain to spare.

The weight-error norms are compared for the same step-size in **Figure 14**. In order to make a fair comparison

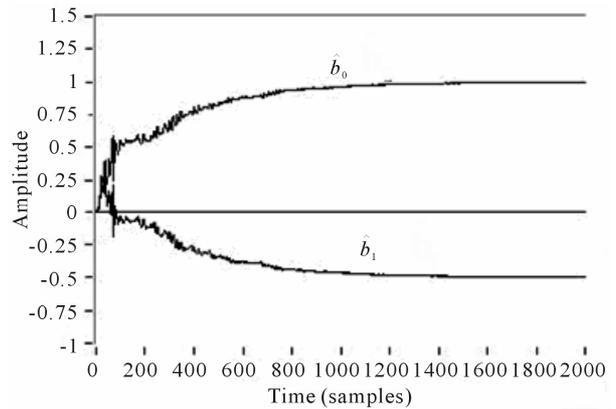


Figure 12. Weight convergence for LMS algorithm and $\chi(\mathbf{R}_{uu}) = 100$.

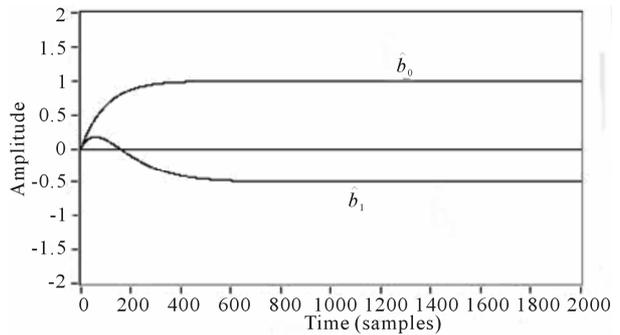


Figure 13. Weight convergence for Algorithm 3.1 and $\chi(\mathbf{R}_{uu}) = 100, K = 0.2$.

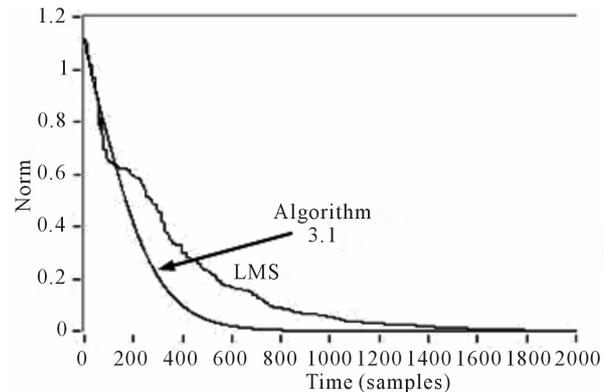


Figure 14. Comparison of weight-error error norm for $\chi(\mathbf{R}_{uu}) = 100, K = 0.2$.

with LMS, it should be pointed out that Algorithm 3.1 in **Figure 14** does not use the maximum step-size (gain).

A comparison of Algorithm 3.1 and 4.1 is shown in **Figure 15** for this example when the gain (step-size) of the loop is increased to $K = 10$.

The LMS case is not shown in **Figure 15** since it becomes unstable. Although Algorithm 4.1 has a much

slower convergence rate than Algorithm 3.1, it is still at least five times as fast as the fastest achievable LMS (by comparison with **Figure 12**). The reason for the significant increase in convergence rate is because the new methods do not use a correlation matrix at all and hence there is no equivalent inversion of such a matrix as in ordinary least-squares problems. Algorithm 3.1 is around 100 times faster than LMS. For the large eigen-value spread case.

It is worth comparing with the RLS algorithm for this case (**Figure 16**). An initial error covariance matrix was set up to be $\text{diag} \{1000, 1000\}$ (for two parameters) in order to get a fast convergence. Unlike LMS, RLS is known not to be sensitive to correlation matrix eigen-value spread [2].

Algorithm 4.1 and LMS (not shown) are much slower than RLS but Algorithm 3.1 is about twice as fast as RLS as shown in **Figure 16**. The gain was adjusted on Algorithm 3.1 in order to achieve the fastest convergence.

Example 3: The problem of an input signal which is not persistently exciting.

Consider the non-minimum phase system

$$W(z^{-1}) = \frac{1 - 2z^{-2}}{1 - 1.5z^{-1} + 0.5z^{-2}}.$$

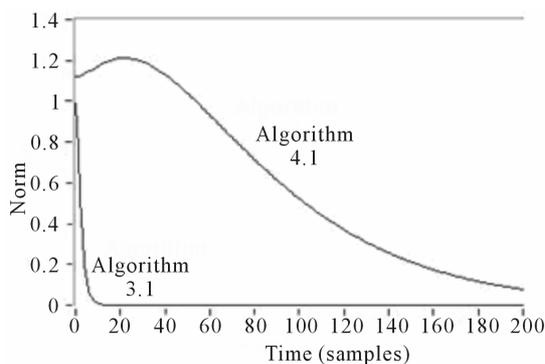


Figure 15. Weight-error norm for Algorithm 4.1 and Algorithm 3.1, $K = 10$ and $\chi(R_{uu}) = 100$.

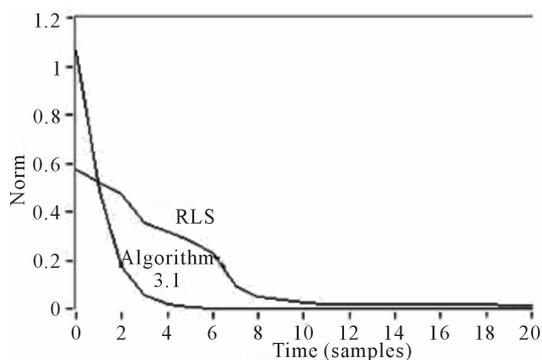


Figure 16. Weight-error norm for Algorithm 3.1 and RLS, $K = 15$ and $\chi(R_{uu}) = 100$.

The system has an FIR equivalent system by using a Taylor series expansion. Hence we find that (to six weights of approximation):

$$W(z^{-1}) = 1 - z^{-1} - 1.5z^{-2} - z^{-3} - 0.25z^{-4} + 0.25z^{-5} + \dots$$

A steady dc (unit step) signal is fed to the input of the system and in steady-state a comparison of various recursive estimation schemes was made. Note that the input here is essentially a step input and not dc per se, but nevertheless some algorithms have difficulty with this type of driving signal.

It was found for 4 weights that the new methods of Algorithm 3.1 and 4.1 both converged to the exact values $\hat{W}_{3.1} \rightarrow [1, -1, -1.5, -1, -0.25 + 0.25]$, $\hat{W}_{4.1} \rightarrow [1, -1, -1.5, -1, -0.25 + 0.25]$ in around 6 iterations. Conversely the LMS algorithm converged to $\hat{W}_{LMS} \rightarrow [-0.378, -0.418, -1.416, -0.355, -0.261 - 0.172]$. Clearly LMS gives the complete wrong answer. However, RLS converges very close and as fast as algorithms 3.1 and 4.1 to the values $\hat{W}_{RLS} \rightarrow [0.999, -0.999, -1.499, -0.999, -0.25, 0.249]$.

6. Conclusion

Two new algorithms have been demonstrated which use feedback instead of correlation methods to estimate the weights of an unknown FIR system. It has been shown that the new algorithms give much smoother estimates of the weights than ordinary LMS. Other than that there is little difference until a driving signal is used whose correlation matrix has widely dispersed eigen-values. Under such conditions the Algorithm 4.1 has at least five times faster convergence and Algorithm 3.1 has about one hundred times faster than ordinary LMS. The disadvantage of Algorithm 3.1 is that the gain needs to be switched in sign as the sign of the first input sample changes. Both of the new algorithms have the property that the gain is not dependent on the number of estimated weights.

REFERENCES

- [1] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," *IRE Wescon Convention Record*, Vol. 4, 1960, pp. 96-104.
- [2] S. Haykin, "Adaptive Filter Theory," *Englewood Cliffs*, Prentice Hall, New Jersey, 1986.
- [3] J. J. Shynk, "Adaptive IIR Filtering," *IEEE of ASSP Magazine*, Vol. 6, No. 2, 1989, pp. 4-21. <http://dx.doi.org/10.1109/53.29644>
- [4] P. Hagander and B. Wittenmark, "A Self-Tuning Filter for Fixed-Lag Smoothing," *IEEE Transactions on Information Theory*, Vol. 23, 1977, pp. 377-384. <http://dx.doi.org/10.1109/TIT.1977.1055719>
- [5] L. Ljung and T. Soderstrom, "Theory and Practice of Re-

- cursive Estimation,” MIT Press, Cambridge, 1987.
- [6] L. R. Vega and H. Rey, “A Rapid Introduction to Adaptive Filtering,” Springer, New York, 2013. <http://dx.doi.org/10.1007/978-3-642-30299-2>
- [7] J. V. Berghe and J. Wouters, “An Adaptive Noise Canceller for Hearing Aids Using Two nearby Microphones,” *Journal of the Acoustical Society of America*, Vol. 103, No. 6, 1998, pp. 3621-3626. <http://dx.doi.org/10.1121/1.423066>
- [8] B. Widrow, “A Microphone Array for Hearing Aids,” *IEEE of Circuits and Systems Magazine*, Vol. 1, 2001, pp. 26-32. <http://dx.doi.org/10.1109/7384.938976>
- [9] L. Horowitz and K. Senne, “Performance Advantage of Complex LMS for Controlling Narrow-Band Adaptive Arrays,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 29, 1981, pp. 722-736. <http://dx.doi.org/10.1109/TASSP.1981.1163602>
- [10] F. Reed, P. L. Feintuch and N. J. Bershad, “Time Delay Estimation Using the LMS Adaptive Filter—Static Behavior,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 29, 1981, pp. 561-571. <http://dx.doi.org/10.1109/TASSP.1981.1163614>
- [11] B. Farhang-Boroujeny, “Fast LMS/Newton Algorithms Based on Autoregressive Modeling and Their Application to Acoustic Echo Cancellation,” *IEEE Transactions on Signal Processing*, Vol. 45, 1997, pp. 1987-2000. <http://dx.doi.org/10.1109/78.611195>
- [12] S. U. H. Qureshi, “Adaptive Equalization,” *Proceedings of the IEEE*, Vol. 73, 1985, pp. 1349-1387. <http://dx.doi.org/10.1109/PROC.1985.13298>
- [13] P. C. Young, “Recursive Estimation and Time-Series Analysis,” 2nd Edition, Springer-Verlag, Berlin, 2011. <http://dx.doi.org/10.1007/978-3-642-21981-8>
- [14] S. Haykin and B. Widrow, “Least-Mean-Square Adaptive Filters (Adaptive and Learning Systems for Signal Processing, Communications and Control),” Wiley Interscience, 2003.
- [15] E. Eweda, “Comparison of RLS, LMS, and Sign Algorithms for Tracking Randomly Time-Varying Channels,” *IEEE Transactions on Signal Processing*, Vol. 42, 1994, pp. 2937-2944. <http://dx.doi.org/10.1109/78.330354>
- [16] T. Aboulnasr and K. Mayyas, “A Robust Variable Step-Size LMS-Type Algorithm: Analysis and Simulations,” *IEEE Transactions on Signal Processing*, Vol. 45, 1997, pp. 631-639. <http://dx.doi.org/10.1109/78.558478>
- [17] R. C. Bilcu, P. Kuosmanen and K. Egiazarian, “A Transform Domain LMS Adaptive Filter with Variable Step-Size,” *IEEE of Signal Processing Letters*, Vol. 9, 2002, pp. 51-53. <http://dx.doi.org/10.1109/97.991136>
- [18] T. J. Moir, “Loop-Shaping Techniques Applied to the Least-Mean-Squares Algorithm,” *Signal, Image and Video Processing*, Vol. 5, 2011, pp. 231-243. <http://dx.doi.org/10.1007/s11760-010-0157-9>
- [19] A. G. J. MacFarlane, “Return-Difference and Return-Ratio Matrices and Their Use in Analysis and Design of Multivariable Feedback Control Systems,” *Proceedings of the Institution of Electrical Engineers*, Vol. 117, 1970, pp. 2037-2049. <http://dx.doi.org/10.1049/piee.1970.0367>

Appendix. State-Space Description

We can look at the more general problem when the input is time-varying but the weights are constant by writing the algorithm in state-space format. For Algorithm 3.1 we have

$$\hat{W}_k = \hat{W}_{k-1} + K[s_k - T(u_k)\hat{W}_{k-1}] \quad (1)$$

which becomes

$$\hat{W}_k = [I - KT(u_k)]\hat{W}_{k-1} + Ks_k \quad (2)$$

The time-varying matrix $I - KT(u_k)$ must have roots (eigen-values) which lie within the unit-circle on the z -plane. This gives us the same as (13) for the stability limit on the gain K .

$$|1 - u_k K| < 1 \quad (3)$$

Equation (3) clearly poses no problem provided u_k and K are always positive. Then

$$0 < K < \frac{2}{u_k} \quad (4)$$

but when u_k becomes negative we must change the sign of K making $K = K_0 \text{sgn}(u_k)$ where K_0 is always positive.

If the true weights are constant $W_k = W_0$ and there is zero measurement noise then we can write (2) as

$$\hat{W}_k = [I - KT(u_k)]\hat{W}_{k-1} + KT(u_k)W_0 \quad (5)$$

Now define the weight-error vector $\epsilon_k = W_0 - \hat{W}_k$ and $\epsilon_{k-1} = W_0 - \hat{W}_{k-1}$ with $s_k = T(u_k)W_0$, and we can write (1) in weight-error format as the homogeneous vector difference equation

$$\epsilon_k - [I - KT(u_k)]\epsilon_{k-1} = 0 \quad (6)$$

Now for some initial condition error ϵ_0 , (6) has solution

$$\epsilon_k = [I - KT(u_k)]^k \epsilon_0 \quad (7)$$

Now write the lower triangular matrix

$$I - KT(u_k) = (1 - Ku_k)I + V(u_k) \quad (8)$$

where the diagonal elements $(1 - Ku_k)I$ of the lower triangular matrix have been separated leaving a square matrix $V(u_k)$ of dimension $(n + 1)$ with zeros in its diagonal

$$V(u_k) = K \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -u_{k-1} & 0 & 0 & 0 & 0 & 0 \\ -u_{k-2} & -u_{k-1} & 0 & 0 & 0 & 0 \\ -u_{k-3} & -u_{k-2} & -u_{k-1} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ -u_{k-n} & -u_{k-(n-1)} & -u_{k-(n-2)} & \cdots & -u_{k-1} & 0 \end{bmatrix}$$

Due to its sparsity, such a matrix when raised to the same power as its dimension will always be zero *i.e.* $V^{n+1}(u_k) = 0$. Therefore from (7), by using the Binomial theorem

$$[I - KT(u_k)]^k = [(1 - Ku_k)I + V(u_k)]^k \quad (9)$$

which, provided (3) holds dies out for large k , taking the weight-error with it. Hence

$$\epsilon_k = W_0 - \hat{W}_k \rightarrow 0 \text{ as } k \rightarrow \infty \quad (10)$$

For example for 3 weights at time $k = 6$

$$\begin{aligned} [(1 - Ku_k)I + V(u_k)]^6 &= (1 - Ku_k)^6 I + 6(1 - Ku_k)^5 V(u_k) + 15(1 - Ku_k)^4 V^2(u_k) \\ \epsilon_6 &= [(1 - Ku_k)^6 I + 6(1 - Ku_k)^5 V(u_k) + 15(1 - Ku_k)^4 V^2(u_k)] \epsilon_0. \end{aligned}$$

Algorithm 4.1 follows in a similar manner with $\epsilon_k = [I - KT^2(u_k)]\epsilon_{k-1}$.