NON-WIENER EFFECTS IN RECURSIVE LEAST SQUARES ADAPTATION

A. A. (Louis) Beex and James R. Zeidler

DSPRL – ECE 0111
Virginia Tech
Blacksburg, VA 24061-0111, USA

Communications & Information Systems, 28505
SPAWAR Systems Center
San Diego, CA 92152, USA

ABSTRACT

In a number of adaptive filtering applications, non-Wiener effects have been observed for the (normalized) least-mean-square algorithm. These effects can lead to performance improvements over the fixed Wiener filter with the same model structure, and are characterized by dynamic behavior of the adaptive filter weights. Here we investigate whether such non-Wiener effects can also occur in the recursive least squares algorithm, and under which circumstances. Examples show that non-Wiener effects can also occur with the recursive least squares algorithm, in particular when the exponential forgetting factor is small. The latter corresponds to a short memory depth, the need for which one generally associates with tracking of time-varying phenomena.

1. INTRODUCTION

In several wide sense stationary scenarios involving narrowband processes, such as noise canceling, prediction, and equalization, adaptive filters have been shown to exhibit better performance than expected from the optimal filters of the same structure [1]. Recently we have shown that such non-Wiener effects in NLMS (normalized least-mean-square) adaptive filtering are due to the capability of adaptive filters to produce dynamic, time-varying behavior of their weights as they attempt to track an underlying more general optimal Wiener filter [2]. It is the extent of success in tracking the latter time-varying situation that leads to advantageous performance in several scenarios of practical interest.

As the RLS (recursive least squares) algorithm is often used in comparisons of adaptive filtering algorithms, the question arises as to whether it can exhibit non-Wiener effects also, and – if so – to what extent in comparison to those in NLMS. An earlier preliminary investigation did not reveal non-Wiener effects in the RLS algorithm [1].

In Section 2 the RLS algorithm with exponential forgetting is summarized, together with its underlying error criterion. The NLMS algorithm is summarized in Section 3. Simulation results that compare RLS with NLMS are presented in Section 4, in particular for a case where non-Wiener effects have been observed to be strong when using the NLMS algorithm.

2. RLS SUMMARY

The RLS algorithm adapts the weight vector $\mathbf{w}_n$ of a linear combiner with input vectors $\mathbf{u}_i$ so that its outputs $\hat{d}_i = \mathbf{w}_n^H \mathbf{u}_i$ are the best approximation – in a weighted least squares sense – to the desired signal d_i over the interval $i \in [1, n]$. The corresponding error criterion is as follows.

$$\begin{aligned} J_{\mathrm{LS},n} &= \sum_{i=1}^{n} \lambda^{n-i} \left|\varepsilon_i\right|^2 \\ &= \sum_{i=1}^{n} \lambda^{n-i} \left|d_i - \mathbf{w}_n^H \mathbf{u}_i\right|^2 \end{aligned} \tag{1}$$

The exact least squares solution is found from the following equation [3],

$$\left(\sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}_i \mathbf{u}_i^H \right) \mathbf{w}_n = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}_i d_i^* \tag{2}$$

which can be recognized as a time-averaged form of the usual Wiener-Hopf equation.

$$\mathbf{R}_n \mathbf{w}_n = \mathbf{p}_n \tag{3}$$

The left- and right-hand sides of (2) can be written recursively, leading to the recursive LS algorithm (RLS).

$$\begin{aligned} \boldsymbol{\psi}_n &= \lambda^{-1} \mathbf{P}_{n-1} \\ \mathbf{k}_{1,n} &= \frac{\boldsymbol{\psi}_n \mathbf{u}_n}{1 + \mathbf{u}_n^H \boldsymbol{\psi}_n \mathbf{u}_n} \\ e_{1,n} &= d_n - \mathbf{w}_{n-1}^H \mathbf{u}_n \\ \mathbf{w}_n &= \mathbf{w}_{n-1} + \mathbf{k}_{1,n} e_{1,n}^* \\ \mathbf{P}_n &= \boldsymbol{\psi}_n - \mathbf{k}_{1,n} \mathbf{u}_n^H \boldsymbol{\psi}_n \end{aligned} \tag{4}$$

Starting the algorithm with

$$\begin{aligned} \mathbf{P}_0 &= \delta^{-1} \mathbf{I} \\ \mathbf{w}_0 &= \mathbf{0} \end{aligned} \tag{5}$$

is known as soft constraint initialization, and corresponds [4] to minimization of the following function.

$$\min_{\mathbf{w}_n} \left(\delta \lambda^n \left\| \mathbf{w}_n \right\|^2 + \sum_{i=1}^{n} \lambda^{n-i} \left|\varepsilon_i\right|^2 \right) \tag{6}$$

Note that when n is large, and λ is less than 1, the criterion becomes essentially the weighted least squares criterion. In fact, the effective length of the window – or memory depth of the algorithm – is:

0-7803-7946-2/03/$17.00 ©2003 IEEE.

$$L_{\mathrm{RLS}} = (1-\lambda)^{-1} \tag{7}$$

so that the interval of minimization is effectively:

$$i \in [\![n - L_{\mathrm{RLS}} + 1, n]\!] \tag{8}$$

where the $[\![$ symbol indicates that the lower interval limit is soft, i.e. the interval is actually larger, but the contributions are relatively smaller. Interpreting the criterion in (1) as the solution to an over-determined set of equations, the soft lower limit means that equations from outside the effective interval – i.e. those corresponding to the far past – do not modify the solution very much that corresponds to the interval suggested by (8).

3. NLMS SUMMARY

The NLMS algorithm is summarized as follows.

$$\begin{aligned} e_n &= d_n - \hat{d}_n \\ \hat{d}_n &= \mathbf{w}_n^H \mathbf{u}_n \end{aligned} \tag{9}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \bar{\mu} \frac{e_n^*}{\mathbf{u}_n^H \mathbf{u}_n} \mathbf{u}_n \tag{10}$$

An interpretation of this algorithm for large step-size $\bar{\mu}$, in particular for $\bar{\mu} = 1$, is that the weight vector $\mathbf{w}_n$ is picked from the manifold $\mathfrak{M}$ of weight vectors that produce an *a posteriori* error of zero [5].

$$\begin{aligned} &\min_{\mathbf{w}_{n+1} \in \mathfrak{M}} \left\| \mathbf{w}_{n+1} - \mathbf{w}_n \right\| \\ &\mathfrak{M} = \left\{ \mathbf{w}_{n+1} : d_n - \mathbf{w}_{n+1}^H \mathbf{u}_n = 0 \right\} \end{aligned} \tag{11}$$

4. SIMULATION RESULTS

Dynamic weight behavior has been observed when using NLMS with large step-sizes in the ANC (adaptive noise canceling) scenario reflected in Fig. 1 [1]. The desired and reference processes are first-order autoregressive [AR(1)] processes in additive white Gaussian measurement noise with signal-to-noise ratio of 20 dB. The AR(1) processes are driven by the same unit-variance, zero-mean, white Gaussian noise, independent of the measurement noise, and characterized by the poles $p_d = 0.999 \exp(j\frac{\pi}{3})$, and $p_u = 0.999 \exp\left(j\left\{\frac{\pi}{3} + \frac{2\pi}{120}\right\}\right)$.

The motivation for this particular choice of parameters is that relatively strong non-Wiener effects have been demonstrated in this scenario when NLMS is used with large step-sizes [2]. For a filter with four taps, using an exponential forgetting factor $\lambda = 0.875$, the magnitudes of the instantaneous errors of RLS(4) and NLMS(4) (with $\bar{\mu} = 1$) are shown in Fig. 2 during steady-state, and compared with those of the optimal four-tap Wiener filter design, WF(4). We note that all filters are operating on the same process realization, to facilitate direct comparison.

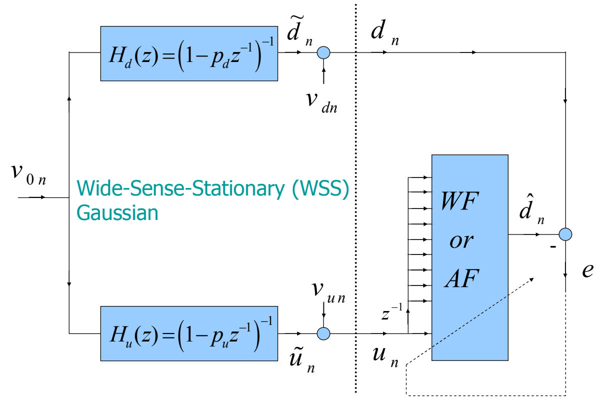


Figure 1: ANC Simulation Setup.

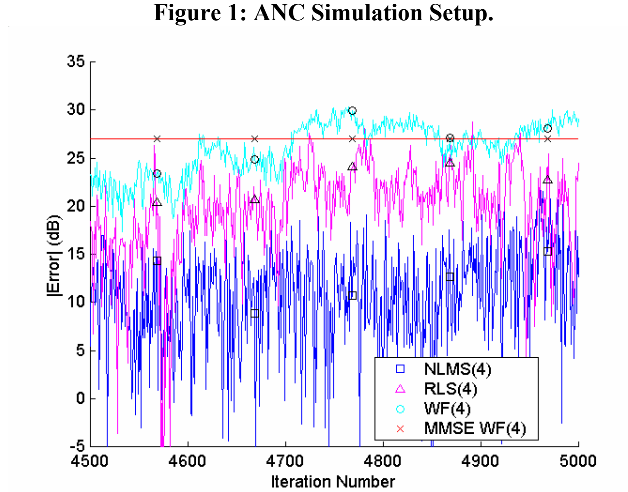


Fig. 2: Instantaneous Filtering Errors.

We observe that the instantaneous error for WF(4) varies around its theoretically expected value, MMSE WF(4), while the instantaneous errors for RLS(4) and NLMS(4) are markedly smaller, almost on a sample-by-sample basis. These observed behaviors are typical. Time-averaging over the entire steady state interval yields MSEs for RLS(4) and NLMS(4) that are markedly lower than for WF(4) and MMSE WF(4). Consequently, we – firstly – observe that RLS exhibits a non-Wiener effect, and – secondly – that it is somewhat weaker than in NLMS.

The typical dynamic weight behavior is shown in Fig. 3, for the real part of the first weight of RLS(4), and compared with its NLMS(4) and WF(4) counterparts.

We observe that the dynamic behavior in the NLMS weights is relatively smooth, while the dynamic behavior of the RLS weights is very noisy. From (7), the equivalent memory depth of the RLS algorithm, for $\lambda = 0.875$, is eight samples. This can be interpreted as solving eight noisy equations for four unknowns. Consequently, as there is no constraint on the weight vector itself in RLS, from iteration to iteration, the weight vector solution can vary drastically as a result of noise (in addition to the gradual variation as a result of the underlying time-varying desired data structure). In NLMS, for $\overline{\mu} = 1$, the change in weight vector is minimal from iteration to iteration.

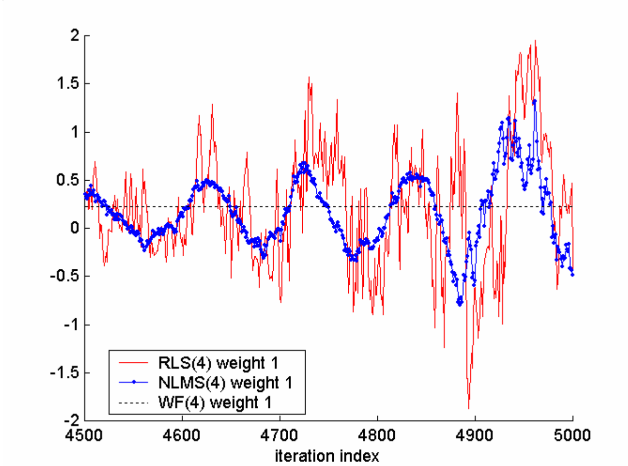
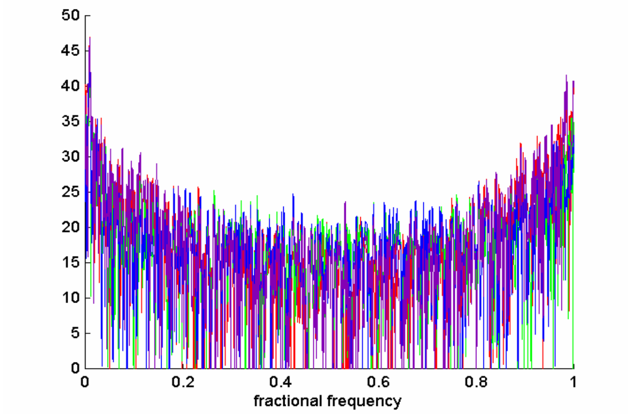
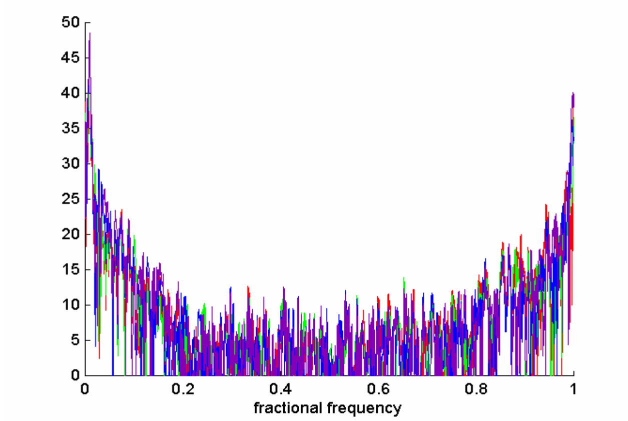


Figure 3: Dynamic Weight Behavior RLS and NLMS.

The spectrograms in Fig. 4, of the RLS(4) and NLMS(4) weights, confirm this difference in noisiness.



(a)

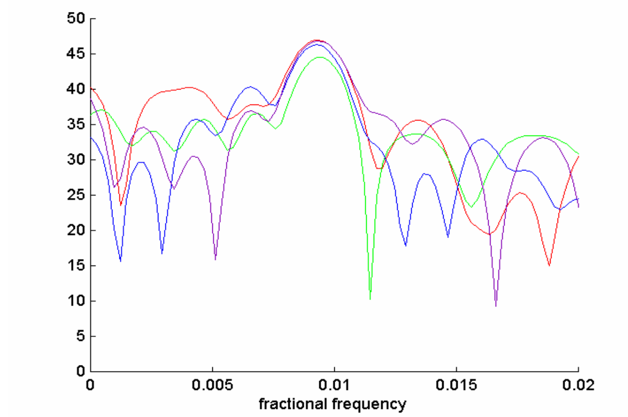


(b)

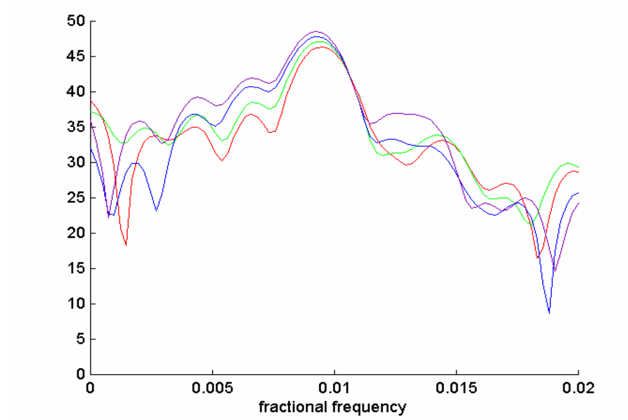
Figure 4: Spectrograms of a) RLS(4) and b) NLMS(4) Weights.

As was to be anticipated from Fig. 3, the NLMS weights are spectrally much purer than the RLS weights.

Figs. 5 show close-ups of Figs. 4 near the most dominant spectral feature, which all four weights are seen to have in common.



(a)



(b)

Figure 5: Close-Up of Spectrograms for a) RLS(4) and b) NLMS(4) Weights.

For the given scenario, the NLMS weights attempt to track the time-varying weights of an equivalent Wiener filter underlying the desired process [2]. The underlying process was shown to be approximately first-order Markov, with the location of its spectral peak determined by the difference frequency of the poles characterizing the desired and reference processes [6]. The latter frequency is the inverse of 120, i.e. a fractional frequency of 1/120=0.008$\underline{3}$ (where the underbar indicates a repeating digit). We observe that the spectrograms of all four weights peak at the same frequency, slightly to the left of 0.01, as anticipated. Note that the 120-sample period can also be observed directly in Fig. 3, in particular for NLMS.

In the above example we've seen that RLS exhibits a non-Wiener effect, in terms of its performance exceeding the performance of the corresponding Wiener filter, and in terms of weight behavior that is dynamic rather than fixed.

We now repeat the above experiment for 100 independent realizations, and evaluate the MSE for RLS and NLMS for various filter orders, and for various forgetting factors (effective memory depths) and step-sizes respectively. Figs. 6 and 7 show the average MSE results for RLS and NLMS respectively.

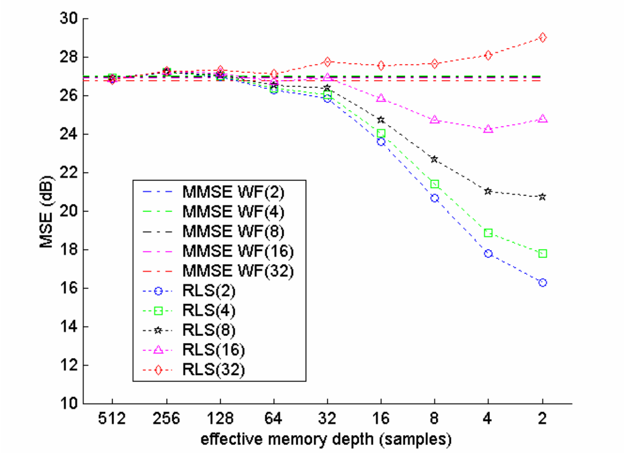


Figure 6: Non-Wiener Performance in RLS.

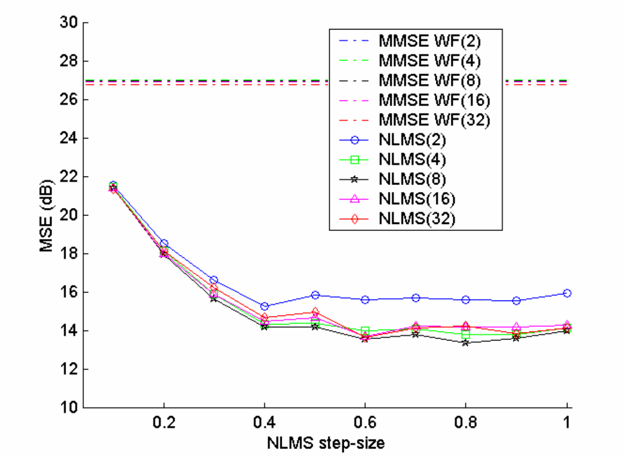


Figure 7: Non-Wiener Performance in NLMS.

We observe from Fig. 6 that the non-Wiener effect in RLS is strongest when the filter order is lowest and the effective memory depth is shortest. Furthermore, as the effective memory depth approaches or exceeds 120, the periodicity associated with the underlying data structure, the non-Wiener effect in RLS vanishes. When filter order and memory depth are both low, effectively a small number of equations is solved for a small number of unknowns. Consequently, while subject to noise, the RLS weight vector can approach the underlying time-varying data structure reasonably well. As the effective memory depth gets larger, RLS finds the one weight vector that fits the larger set of equations best, thereby effectively averaging the underlying time-varying data structure over the extent of the memory depth. Note that the results for RLS(32) support the earlier finding in which no non-Wiener effects were observed for RLS [1].

From Fig. 7 we observe that the non-Wiener effects in NLMS are generally stronger than those in RLS, that they persist over a wide range of step-sizes, and that they don't diminish with increasing filter order. The latter is explained by the instantaneous nature of the NLMS algorithm, in which the weight vector increment is minimized while finding the weight vector in the manifold of weight vectors – defined in (11) – that yields a small *a posteriori* error for the current desired signal value. This instantaneous nature of NLMS, at large step-sizes, facilitates better tracking of the weight dynamics underlying the desired data, without the increased averaging inherent in RLS as model order and memory depth increase.

5. CONCLUSIONS

The time-varying, dynamic weight behavior observed to be possible with NLMS adaptive filters was shown to be possible with RLS adaptive filters in much the same adaptive noise canceling scenario. While in NLMS such non-Wiener behavior can just as easily occur with longer filters as with shorter ones, the averaging nature of the RLS algorithm – i.e. the assumption that the RLS weight vector solution is constant over its effective memory depth – leads to a progressive loss (and vanishing) of the non-Wiener effect as filter length and memory depth increase.

6. REFERENCES

[1] M. Reuter, K. Quirk, J. Zeidler, and L. Milstein, "Nonlinear effects in LMS adaptive filters," *Proc. Symp. 2000 on Adaptive Systems for Signal Processing, Communications and Control*, Lake Louise, Alberta, Canada, pp. 141-146, October 2000.

[2] A. A. (Louis) Beex and James R. Zeidler, "Steady State Dynamic Weight Behavior in (N)LMS Adaptive Filters," in *Advances in Least-Mean-Square Adaptive Filters,* eds. S. Haykin and B. Widrow, New York: Wiley, July 2003, in press.

[3] Simon Haykin, *Adaptive Filter Theory,* 3rd ed, New Jersey: Prentice Hall, 1996.

[4] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, Vol. 11, No. 3, pp. 18-60, July 1994.

[5] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*, Prentice-Hall, 1984.

[6] A. A. (Louis) Beex and James R. Zeidler, "Data Structure and Non-Linear Effects in Adaptive Filters," *14th International Conference on Digital Signal Processing* (DSP2002), Santorini, Greece, pp. 659-662, 1-3 July 2002.