A Sliding Window RLS-like Adaptive Algorithm for Filtering alpha-stable Noise

Murat Belge and Eric L. Miller

Abstract— **We introduce a sliding window adaptive RLS-like algorithm for filtering alpha-stable noise. Unlike previously introduced stochastic gradient type algorithms, the new adaptation algorithm minimizes the L_p norm of the error exactly in a sliding window of fixed size. Therefore, it behaves much like the RLS algorithm in terms of convergence speed and computational complexity compared to previously introduced stochastic gradient based algorithms which behave like the LMS algorithm. It is shown that the new algorithm achieves superior convergence rate at the expense of increased computational complexity.**

I. Introduction

In the vast majority of signal processing applications it has been assumed that the signal or noise under investigation can be modeled by a Gaussian distribution law. This assumption has been justified by the central limit theorem and strong analytical properties of Gaussian pdf which leads to linear algorithms. However, in many real-world problems the noise encountered is more impulsive in nature than that predicted by a Gaussian distribution. Examples are underwater acoustic noise, low frequency atmospheric noise and many types of man-made noise [1]. Systems optimized under the Gaussian assumption often yield unacceptable performance when subjected to impulsive, non-Gaussian noise [12]. There exists a class of distributions, called alpha-stable distributions, than can be used to model these types of noise [1].

With the introduction of alpha-stable distributions to the signal processing community, a number of different adaptive filtering approaches have been proposed for filtering these processes [3], [5], [4], [7]. All of the algorithms that have been introduced so far can be classified as an LMS variant [8] which basically updates the filter coefficients by using an instantaneous approximation to the gradient of the cost function. The only difference of these algorithms from the conventional LMS algorithm is the minimization of the L_p norm of the error at the output of the adaptive filter instead of the usual Euclidean norm. The Least Mean p-Norm (LMP) algorithm [6] was derived exactly as described above. Later, motivated by the normalized versions of the LMS algorithm, Arikan *et. al.* developed the Normalized LMP (NLMP) algorithm [3]. Both LMP and NLMP suffer from the same problem that has plagued the LMS algorithm. Namely, when the input to the adaptive filter is highly correlated the convergence is very slow. The RLO algorithm [7] is an alternative to LMP and NLMP, however its implementation requires some a-priori information on the error statistics and the filter inputs [7].

In this work, we develop a sliding window adaptation algorithm which is similar to the RLS algorithm [2] both in terms of derivation and convergence characteristics. The new algorithm provides much increased convergence rate at the expense of increased computational complexity. A block implementation of the new algorithm decreases the computational cost substantially.

Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA 02148. This work was supported by an ODDR&E MURI under Air Force Office of Scientific Research contract F49620-96-1-0028, a CAREER Award from the National Science Foundation MIP-9623721, and the Army Research Office Demining MURI under Grant DAAG55-97-1-0013.

II. Sliding Window Least Mean p-norm Adaptation Algorithm

The objective of an adaptation algorithm is to minimize the averaged error at the output of the filter by adjusting the coefficients of the filter. Adaptive estimation of a time-varying finite impulse response system is usually obtained by limiting the filtering memory. Here, we adopt a true finite memory or a sliding window approach for the adaptation of filter coefficients. That is we minimize the averaged L_p norm of the error in a window of size L:

$$J_{\underline{w}}(n) = \sum_{k=n-L+1}^{n} |d(k) - \underline{w}^t(n)\underline{x}(k)|^p = \sum_{k=n-L+1}^{n} |e(k)|^p \tag{1}$$

where $d(k)$ is the desired signal at time k, $\underline{w}(n)$ is the vector of optimal filter coefficients at time n, $\underline{x}(k) = [x(k)\ x(k-1) \ldots x(k-N+1)]^t$ contains the N most recent samples of the input signal and $1 \leq p < 2$. Taking the gradient of $J_{\underline{w}}(n)$ with respect to $\underline{w}(n)$ and equating the result to zero we obtain:

$$\sum_{k=n-L+1}^{n} |e(k)|^{p-1} \mathrm{sign}[e(k)] \underline{x}(k) = \underline{0} \tag{2}$$

which can be written as

$$\sum_{k=n-L+1}^{n} u(k)\underline{x}(k)\underline{x}^t(k)\underline{w}(n) = \sum_{k=n-L+1}^{n} u(k)d(k)\underline{x}(k) \tag{3}$$

where $u(k) = |e(k)|^{p-2}$, and (3) is obtained by substituting $\frac{e(k)}{|e(k)|}$ for $\mathrm{sign}[e(k)]$ in (2) and rearranging. Defining $\underline{r}(n) = \sum_{k=n-L+1}^{n} u(k)d(k)\underline{x}(k)$ and $R(n) = \sum_{k=n-L+1}^{n} u(k)\underline{x}(k)\underline{x}^t(k)$ we obtain the following expression for $\underline{w}^*(n)$ which minimizes the p-norm of the error in a window of size L:

$$\underline{w}^*(n) = R^{-1}(n)\underline{r}(n) \tag{4}$$

Note, however, that $\underline{w}^*(n)$ cannot be readily obtained from (4) since $R(n)$ and $\underline{r}(n)$ are functions of $\underline{w}^*(n)$. However, we can devise an iterative scheme to solve for $\underline{w}^*(n)$ at each point in time. Such an approach leads to the following algorithm:

Algorithm 1.
1. $\underline{w}^0(n) = \underline{w}^*(n-1)$
2. Compute $u^j(k) = |d(k) - \underline{x}^t(k)\underline{w}^j(n)|^{p-2}$, $k = n-L+1,..,n$
3. Compute $R^j(n) = \sum_{k=n-L+1}^{n} u^j(k)\underline{x}(k)\underline{x}^t(k)$ and $\underline{r}^j(n) = \sum_{k=n-L+1}^{n} u^j(k)d(k)\underline{x}(k)$
4. $\underline{w}^{j+1}(n) = \left[R^j(n)\right]^{-1} \underline{r}^j(n)$
5. If $\frac{\|\underline{w}^{j+1}(n)-\underline{w}^j(n)\|}{\|\underline{w}^j(n)\|} < \epsilon$ then, $\underline{w}^*(n) = \underline{w}^{j+1}(n)$, stop; else $j = j+1$, goto step 2

For each new input sample, we apply algorithm 1 to obtain the optimal filter coefficients, $\underline{w}^*(n)$, for the current time. Steps 2-5 of Algorithm 1 constitute the so-called *iterative re-weighted least squares* (IRLS) method which has been suggested and applied in several contexts [11], [9], [10]. The convergence of the IRLS algorithm can be guaranteed by making the following modification [9]:

$$u^j(k) = \begin{cases} u^j(k) & \text{if } u^j(k) \le \frac{1}{\mu} \\ \frac{1}{\mu} & \text{if } u^j(k) > \frac{1}{\mu} \end{cases} \tag{5}$$

where μ is a small positive constant.

Note that, in step 4 of Algorithm 1, we need the inverse of $R^j(n)$. Rather than first computing $R^j(n)$ and then inverting this matrix to obtain $\underline{w}^{j+1}(n)$ we may consider computing $\left[R^j(n)\right]^{-1}$ directly in step 3 of Algorithm 1. To this end, consider the following expression for $i = L,..,1$:

$$R^j(n-i+1) = R^j(n-i) + u^j(n-i+1)\underline{x}(n-i+1)\underline{x}^t(n-i+1) \tag{6}$$

Defining, $P^j(n) = \left[R^j(n)\right]^{-1}$, and applying the matrix inversion lemma [2] to (6) we obtain:

$$P^j(n-i+1) = P^j(n-i) - \frac{1}{\alpha(n-i)}\underline{g}(n-i)\underline{g}^t(n-i) \tag{7}$$

where $\underline{g}(n-i) = P^j(n-i)\underline{x}(n-i+1)$ and $\alpha(n-i) = 1/u^j(n-i+1) + \underline{x}^t(n-i+1)\underline{g}(n-i)$. Equation (7) implies that the matrix $P^j(n)$ can be obtained by a series of recursive updates starting from the matrix $P^j(n-L)$ at the beginning of the window. We assume that $P^j(n-L) = \frac{1}{\sigma}I$, which corresponds to a soft initialization [2]. Then, in step 4 of Algorithm 1, we compute $P^j(n)$ by using (7) instead of $R^j(n)$ and then obtain $\underline{w}^{j+1}(n)$ in step 5 by $\underline{w}^{j+1}(n) = P^j(n)\underline{r}(n)$.

The complexity of the algorithm given above is $O(MLN^2)$ where M is the number of IRLS iterations (steps 2-5 of Algorithm 1) needed. Because of the similarity of the algorithm to RLS, we call the new approach the recursive least mean p-norm algorithm (RLMP). The direct implementation of the RLMP algorithm is infeasible for most applications because of its high computational complexity which is dominated by construction of $P^j(n)$. However, a *subsampled* version of the RLMP algorithm where the filter coefficients are updated once at every k iterations, $k > 1$ being the subsampling rate, can be considered. In particular, if $k = L$, the complexity of the RLMP algorithm is $O(MN^2)$ per iteration. In this case, the filter coefficients are updated once for every data block of length L. As we will see in Section 3, the average number of IRLS iterations can be quite low making the subsampled version of the RLMP algorithm a viable alternative to its stochastic gradient type counterparts.

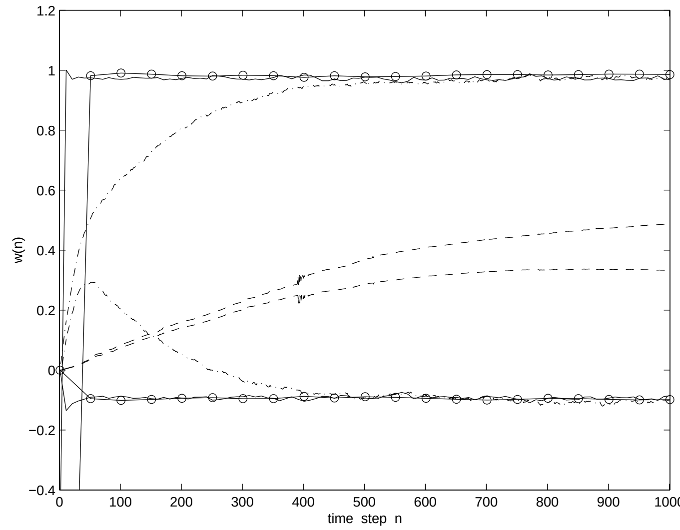

Fig. 1. Transient behavior of tap weight adaptations for RLMP with $L = 50$, $k = 10$ (solid line), RLMP with $L = 100$, $k = 50$ (circles), NLMP (dashed line) and LMP (dash-dotted line) algorithms.

III. Simulation Study

In this section, we compare the performance of the RLMP algorithm to that of NLMP and LMP algorithms. In the NLMP and LMP algorithms, the coefficients of the adaptive filter is updated as follows [3]:

$$\underline{w}(n+1) = \underline{w}(n) + \alpha\frac{|e(n)|^{p-1}\text{sign}[e(n)]}{h(n)}\underline{x}(n) \tag{8}$$

where $h(n) = 1$ for LMP and $h(n) = \|x(n)\|_p^p + \gamma$, with $\gamma > 0$ being a small constant, for the NLMP algorithm. Following [3], we consider the following AR process:

$$x(n) = 0.99x(n-1) - 0.1x(n-2) + u(n) \tag{9}$$

where $u(n)$ is an alpha-stable sequence of i.i.d. random variables with $\alpha = 1.2$, $\beta = 0$ and $\gamma = 1$. A simulation is performed to identify the coefficients of the AR process with the $p = 1.1$ norm. Figure 1 shows the transient behavior of the tap weights of the adaptive filter and Fig. 2 shows the norm of the error between the true and the estimated parameters, defined as $E(n) = 20\log_{10}\|\underline{w}_{true} - \underline{w}(n)\|_2$. Both figures were obtained by averaging the results of 100 independent trials. The parameters of the RLMP algorithm are: $\mu = 10^{-6}$, $\sigma = 10^{-4}$, $\epsilon = 10^{-2}$. The RLMP algorithm was implemented for two different window sizes corresponding to $L = 50$ and $L = 100$ samples. For $L = 50$, the filter coefficients were updated once for every 10 iterations and for $L = 100$, the filter coefficients were updated once per 50 data samples. For a window of $L = 50$ samples, the RLMP algorithm produces a steady-state tap

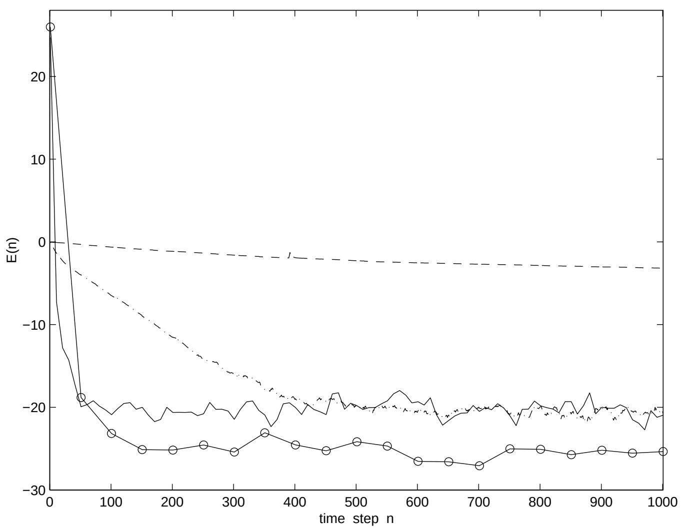

Fig. 2. Transient behavior of tap weight error powers for RLMP with $L = 50$, $k = 10$ (solid line), RLMP with $L = 100$, $k = 50$(circles), NLMP (dashed line) and LMP (dash-dotted line) algorithms.

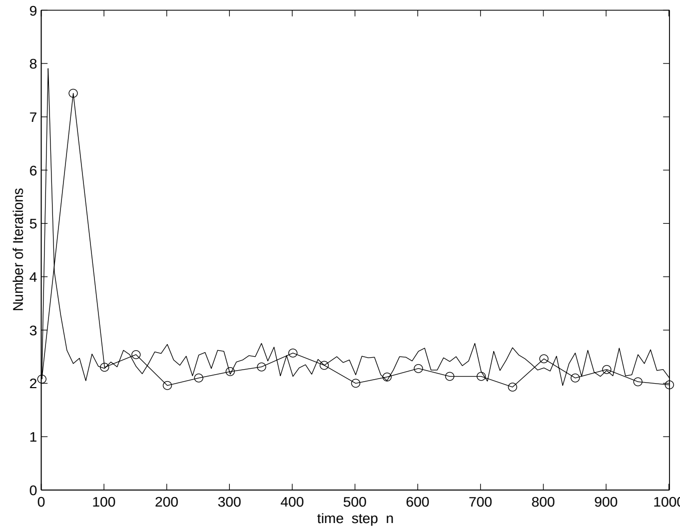

Fig. 3. Fig. 3 Ensemble average of the number of IRLS iterations needed at each time instant for RLMP with $L = 50$, $k = 10$ (solid line) and RLMP with $L = 100$, $k = 50$(circles).

weight error of approximately -20dB. This figure can be made smaller/larger by adjusting the window length. The step sizes of the NLMP and the LMP algorithms were set to produce the same steady state tap weight error as RLMP. The step size for NLMP was found to be 3.8×10^{-2} and 5×10^{-5} for LMP. Figure 1 and 2 show that RLMP provides a large improvement in convergence rate over the NLMP and LMP. To give an idea about the number of iterations needed for the outer IRLS iterations to converge, we computed the number of IRLS iterations at each discrete time, n, by averaging 100 trials. The results are displayed in Fig. 3. It is seen that at most 3 iterations are sufficient to obtain a relative error $\frac{\|\underline{w}^{j+1}(n)-\underline{w}^{j}(n)\|}{\|\underline{w}^{j}(n)\|}$ of about 10^{-2}.

Examining Algorithm 1 in detail, we see that the most efficient implementation of the RLMP algorithm requires $ML(2N^2+3N+2)$ multiplications, $2ML$ divisions and ML nonlinear operations per update of the filter coefficients. In general, for $k = L$ (i.e. filter coefficients are updated once for each data block of L samples), the computational effort required by the RLMP algorithm is approximately M times that of a single RLS update plus M nonlinear operations. From Fig. 3, we see that, for this example, the ensemble average of M is actually quite low and approximately 2.5 at the steady state.

IV. Conclusion

In this letter, we described a novel adaptation algorithm for filtering alpha-stable noise. The new algorithm is derived by minimizing the averaged L_p error at the output of the filter in a window of fixed size. Simulations show that the new algorithm provides much improved convergence rate compared to other stochastic gradient based adaptation algorithms for alpha-stable noise environments. The major disadvantage of the algorithm is its computational complexity. We proposed a subsampled implementation of the RLMP algorithm which reduces the computational complexity to $O(MN^2)$ per data sample. Current research is focused on frequency domain implementations of the RLMP algorithm which will further reduce the computational complexity.

References

[1] C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications*, New York: Wiley, 1995.

[2] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[3] O. Arikan, M. Belge, E. Cetin, E. Erzin, “Adaptive filtering approaches for non-Gaussian stable processes,” *Proc. ICASSP’95*, v. 2, pp. 1400-1403, 1995.

[4] G. Aydin, O. Tanrikulu, E. Cetin, “Robust least mean mixed norm adaptive filtering for α-stable random processes,” *Proc. IEEE ISCAS’97*, v. 4, pp. 2296-2299, 1997.

[5] S. Kalluri, G. R. Arce, “Adaptive weighted myriad filter algorithms for robust signal processing in α-stable noise environments,” *IEEE Trans. Signal Processing*, v. 46, n. 2, pp. 322-334, 1998.

[6] M. Shao and C. L. Nikias, “Signal processing with fractional lower order moments: Stable processes and their applications,” *Proc. IEEE*, v. 81, pp. 986-1009, 1993

[7] J. S. Bodenschatz and C. L. Nikias, “Recursive local orthogonality filtering,”

[8] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, *IEEE Trans. Signal Processing*, v. 45, n. 9, pp. 2293-2300, 1997. Englewood Cliffs, NJ: Prentice-Hall, 1985

[9] J. Schroeder, R. Yarlagadda, J. Hershey, “L_p normed minimization with applications to linear predictive modeling for sinusoidal frequency estimation,” *Signal Processing*, v. 24, n. 2, pp 193-216, 1991.

[10] E. E. Kuruoglu, P. J. W. Rayner, W. J. Fitzgerald, “Least l_p-norm impulsive noise cancellation with polynomial filters,” *Signal Processing*, v. 69, pp. 1-14, 1998.

[11] M. Belge, M. E. Kilmer, E. L. Miller, “Wavelet domain image restoration with adaptive edge preserving regularization,” http://www.cdsp.neu.edu/info/students/belge/index.html, to appear in IEEE Trans. Image Processing.

[12] G. A. Tsihrintzis and C. L. Nikias, “Performance of optimum and suboptimum receivers in the presence of impulsive noise modeled as an alpha-stable process,” *IEEE Trans. Commun.*, v. 43, pp. 904-914, March 1995.