Fig. 6. Left to right: outputs of the proposed scheme (N = 64) and classical full-search vector quantization (N = 256) for image Man.

Significant power savings are achieved during both coding and decoding. The fact that the power required for the decoding is decreased in comparison to classical vector quantization is of great significance as the decoder is usually the mobile part in portable applications (wireless applications) where power consumption is the overriding issue. Image quality comparable to or better than the corresponding of full-search vector quantization is achieved. For specific codebook sizes, coding speed is also improved in comparison to classical full-search vector quantization (on a basis of acceptable image quality).

The dominant tradeoff in the proposed scheme is between image quality and power consumption. This tradeoff becomes more critical in applications with low-power consumption requirements, such as portable applications. From the experimental results, it is straightforward that the codebook size is the factor that mainly determines this tradeoff.

References

[1] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies.* Norwell, MA: Kluwer Academic, 1995.

[2] A. P. Chandrakasan, A. Burstein, and R. W. Brodersen, "A low-power chipset for a portable multimedia I/O terminal," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1415–1428, Dec. 1994.

[3] T. Meng, B. Gordon, E. Tsern, and A. Hung, "Portable video-on-demand in wireless communication," *Proc. IEEE*, vol. 83, pp. 659–680, Apr. 1995.

[4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Norwell, MA: Kluwer Academic, 1992.

[5] S. Wuytack, F. Catthoor, F. Franssen, L. Nachtergaele, and H. DeMan, "Global communication and memory optimizing transformations for low-power systems," in *Proc. 1994 Int. Workshop Low-Power Design*, pp. 203–208.

[6] M. F. Barnsley, *Fractal Image Compression.* London: A. K. Peters, 1994.

[7] A. E. Jacquin, "Image coding based on fractal theory of iterative contractive image transformations," *IEEE Trans. Image Processing*, vol. 1, pp. 18–30, Jan. 1992.

[8] ——, "Fractal image coding: A review," *Proc. IEEE*, vol. 81, pp. 1451–1464, Oct. 1993.

[9] J. Rabaey, L. Guerra, and R. Mehra, "Design guidance in the power dimension," in *Proc. IEEE ICASSP*, 1995.

[10] J. M. Rabaey, "Exploring the power dimension," in *Proc. IEEE 1996 Custom Integrated Circuits Conf.*, pp. 215–220.

[11] D. Lidsky and J. M. Rabaey, "Early power exploration: A world wide web application," in *Proc. 33rd Design Automation Conf.*, Las Vegas, 1996.

[12] P. Landman and J. Rabaey, "Architectural power analysis: The dual bit type method," *IEEE Trans. VLSI Systems*, vol. 3, pp. 173–187, June 1995.

[13] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design.* Norwell, MA: Kluwer Academic, 1995.

[14] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990, pp. 423–425.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.

Robust Adaptive Filtering Algorithms for α-Stable Random Processes

Gül Aydín, Orhan Aríkan, and A. Enis Çetin

***Abstract*— A new class of algorithms based on the fractional lower order statistics is proposed for finite-impulse response adaptive filtering in the presence of α-stable processes. It is shown that the normalized least mean p-norm (NLMP) and Douglas' family of normalized least mean square algorithms are special cases of the proposed class of algorithms. A convergence proof for the new algorithm is given by showing that it performs a descent-type update of the NLMP cost function. Simulation studies indicate that the proposed algorithms provide superior performance in impulsive noise environments compared to the existing approaches.**

***Index Terms*— Adaptive filtering, alpha-stable random processes, impulsive signals, LMS algorithm.**

I. Introduction

In many signal processing applications, the noise is modeled as a Gaussian process with significant simplification in the required processing. To justify this assumption, the central limit theorem (CLT) is usually quoted. However, a large class of physical observations exhibits non-Gaussian behavior, such as low frequency atmospheric noise, man-made noise, and underwater acoustic noise [1]–[4]. Typical realizations of such random signals contain a large number of outliers. Due to this reason, the Gaussian noise model for these signals cannot be justified. Recently, α-stable processes, which are the limiting distributions for a more general CLT, were proposed to model this type of impulsive noise environment [5].

The α-stable distributions do not have closed form probability density functions except the cases $\alpha = 1$ (Cauchy distribution) and $\alpha = 2$ (Gaussian distribution). However, they have closed form characteristic functions given by

$$\phi(t) = \exp\{iat - \gamma|t|^{\alpha}[1 + i\beta\ \mathrm{sign}(t)w(t,\alpha)]\} \tag{1}$$

where α $(0 < \alpha \leq 2)$ is the characteristic exponent, a is the location parameter, β $(-1 \leq \beta \leq 1)$ is the index of skewness, $\gamma > 0$ is the dispersion parameter, $\mathrm{sign}(\cdot)$ denotes the signum function, and

$$w(t,\alpha) = \begin{cases} \tan\left(\dfrac{\alpha\pi}{2}\right), & \text{if } \alpha \neq 1 \\ \dfrac{2}{\pi}\log|t|, & \text{if } \alpha = 1. \end{cases} \quad (2),$$

The distribution is called symmetric α-stable (SαS), if $\beta = 0$. The parameter α controls the tails of the distribution. For $0 < \alpha < 2$,

Manuscript received September 16, 1996; revised January 28, 1998. This paper was recommended by Associate Editor C.-Y. Wu.

The authors are with the Department of Electrical Engineering, Bilkent University, Ankara, Turkey.

Publisher Item Identifier S 1057-7130(99)01759-0.

the distributions have algebraic tails which are significantly heavier than the exponential tail of the Gaussian distribution. The smaller the value of the α, the heavier the tails of the distributions. This property makes the α-stable distributions an appealing model for impulsive noise environments. It is well known that algorithms developed under the Gaussian assumption may produce unacceptable results [5], [6], if the noise is impulsive.

Due to the heavy tails, stable distributions do not have finite second- or higher-order moments, except the limiting case of $\alpha = 2$. More precisely, for X, an α-stable random variable with $0 < \alpha < 2$

$$\boldsymbol{E}[|X|^p] = \infty, \quad \text{if } p \geq \alpha. \tag{3}$$

However, for $0 \leq p < \alpha$, the fractional lower order moment (FLOM) is finite, i.e.,

$$\boldsymbol{E}[|X|^p] < \infty, \quad \text{if } 0 \leq p < \alpha. \tag{4}$$

If $\alpha = 2$,, then

$$\boldsymbol{E}[|X|^p] < \infty, \quad \text{for all } p \geq 0. \tag{5}$$

The fractional pth-order moment [5] of an SαS random variable with zero location parameter, $a = 0$, is given by

$$\boldsymbol{E}[|X|^p] = C(p, \alpha)\gamma^{p/\alpha}, \quad \text{for } 0 < p < \alpha \tag{6}$$

where

$$C(p, \alpha) = \frac{2^{p+1}\Gamma\left(\dfrac{p+1}{2}\right)\Gamma(-p/\alpha)}{\alpha\pi\Gamma(-p/2)}. \tag{7}$$

In (7), $\Gamma(\cdot)$ denotes the gamma function.

Since the variance of an α-stable process is infinite for $\alpha < 2, \alpha$-stable processes cannot be treated in a Hilbert space framework which requires the existence of L_2 norm. However, it is possible to use a Banach space framework for the geometrical treatment of the α-stable processes, where only the existence of L_p norm for $p < \alpha$ is required [5], [7].

In this paper, we introduce new algorithms for adaptive filtering under additive α-stable noise with finite mean corresponding to the case of $1 \leq \alpha < 2$. These adaptive algorithms are based on fractional lower order statistics (FLOS). The performance of the algorithms is compared to those of the normalized least mean square (NLMS)-type algorithms which are developed under the Gaussian assumption and the NLMP norm algorithm which is based on FLOS.

II. Some Related Adaptive Filtering Algorithms

The NLMS algorithm, also known as the projection algorithm [9], has the following update equation:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu \frac{e_k}{\displaystyle\sum_{m=0}^{M-1} x_{k-m}^2} \boldsymbol{X}_k \tag{8}$$

where $\boldsymbol{W}_k = [w_{0,k} \cdots w_{M-1,k}]^T$ are the tap weights of the adaptive filter at time $k, \boldsymbol{X}_k = [x_k \cdots x_{k-M+1}]^T$ are the M samples of the input data in filter memory at time k, $e_k = d_k - \boldsymbol{W}_k^T \boldsymbol{X}_k$ is the error between the adaptive filter output and the desired signal d_k, and μ is the step size which should be appropriately determined.

Recently, Douglas [11] proposed a family of algorithms of the form

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu e_k F_q(\boldsymbol{X}_k)$$
$$[F_q(\boldsymbol{X}_k)]_i = \begin{cases} \dfrac{|x_{k-i}|,^{q-1} \ \mathrm{sign}(x_{k-i})}{\displaystyle\sum_{m=0}^{M-1} |x_{k-m}|^q}, & \text{if } 1 \leq q < \infty \\ \dfrac{1}{x_{k-n}}\delta_{i-n}, & \text{if } q = \infty \end{cases} \tag{9}$$

where $[F_q(\cdot)]_i$ denotes the ith element of the vector-valued function $F_q(\cdot), n$ is any one of the integers $0, 1, \cdots, M-1$ such that $|x_{k-n}| = \max_{0 \leq j \leq M-1} |x_{k-j}|$, and δ_j is the Kronecker delta function. The above update equation can be rewritten more compactly as

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu \frac{e_k}{\|\boldsymbol{X}_k\|_q^q} \boldsymbol{X}_k^{\langle q-1 \rangle} \tag{10}$$

where $\boldsymbol{X}_k^{\langle q-1 \rangle} = [x_k^{\langle q-1 \rangle} \cdots x_{k-M+1}^{\langle q-1 \rangle}]^T$ and $\langle \cdot \rangle$ operator corresponds to $z^{\langle b \rangle} \triangleq |z|^b \ \mathrm{sign}(z)$ for any real number z and $b \geq 0$. For $q = 2$, this algorithm reduces to the NLMS algorithm of (8). Also, for $q = 1$, it reduces to the well known normalized sign algorithm which was proposed to decrease the computational complexity of the NLMS algorithm [10]. Using any valid q and taking $\mu = 1$ the family of algorithms in (9) is shown to be the solution of the following optimization problem [11]:

$$\text{minimize} \quad \|\boldsymbol{W}_{k+1} - \boldsymbol{W}_k\|_p \tag{11}$$
$$\text{subject to} \quad d_k - \boldsymbol{W}_{k+1}^T \boldsymbol{X}_k = 0 \tag{12}$$

where $\|\cdot\|_p$ denotes the L_p norm, and p satisfies $1/p + 1/q = 1$. Therefore, the adaptation algorithm of (10) provides the minimum change in the L_p-norm sense of the tap weights to exactly satisfy the filtering relationship between the input data and the desired response at time k, similar to the projection in the L_2-norm case.

It can be shown that in the presence of α-stable distribution, the variance of the update term of (9) is not finite. We also experimentally observe this fact as discussed in Section III-A.

Another gradient descent algorithm which can be used in the presence of α-stable distributions has been derived to minimize the following p-norm cost function (hence known as least mean p-norm (LMP) or LMP algorithm)

$$J_k = \boldsymbol{E}[|e_k|^p] \tag{13}$$

and it has the following update equation [5]:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu e_k^{\langle p-1 \rangle} \boldsymbol{X}_k \tag{14}$$

for $1 \leq p < \alpha$. The normalized version of the LMP algorithm, which will be referred to as NLMP, has the following update equation [7]:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu \frac{e_k^{\langle p-1 \rangle}}{\|\boldsymbol{X}_k\|_p^p + \lambda} \boldsymbol{X}_k. \tag{15}$$

In the next section, we introduce a more general adaptation algorithm and then compare its performance to that of the algorithms summarized in this section.

III. A Family of Adaptation Algorithms Based on FLOS

As a generalization of the NLMP update equation, we propose the following update equation:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu \frac{e_k^{\langle a \rangle}}{\|\boldsymbol{X}_k\|_{qa}^{qa} + \lambda} \boldsymbol{X}_k^{\langle (q-1)a \rangle} \tag{16}$$

which reduces to the NLMP update if a and q are chosen as $p-1$ and p/a, respectively. In the presence of an α-stable process, the above update equation can be used with $0 < a \leq \alpha - 1$ and $1 \leq q$. Also, the update in (10) can be obtained as a special case with the choice of $a = 1$ and $\lambda = 0$.

The proposed update can be further motivated by observing that it corresponds to a gradient descent adaptation approach to the following cost function:

$$J_k = \boldsymbol{E}[|e_k|^{a+1}], \quad \text{for } 0 < a < \alpha - 1. \tag{17}$$

A proof of this observation can be obtained by showing that the inner product of the weight update vector in (16) and the instantaneous gradient estimate of the above cost function is nonpositive [8]. This will naturally lead to the conclusion that for sufficiently small μ, the update is of gradient descent type. It can be shown easily that instantaneous gradient estimate of the cost function in (17) at $\boldsymbol{W}_k$ is

$$\nabla_{\boldsymbol{W}} J_k = -(a+1)e_k^{\langle a\rangle}\boldsymbol{X}_k. \tag{18}$$

The inner product of the instantaneous gradient estimate and the weight update vector in (16) is

$$\begin{aligned}&\{\nabla_{\boldsymbol{W}} J_k\}^T \mu \frac{e_k^{\langle a\rangle}}{\|\boldsymbol{X}_k\|_{qa}^{qa}+\lambda}\boldsymbol{X}_k^{\langle (q-1)a\rangle}\\ &\quad = -(a+1)\mu \frac{e_k^{\langle a\rangle 2}}{\|\boldsymbol{X}_k\|_{qa}^{qa}+\lambda}\sum_{i=0}^{M-1}|x_{k-i}|^{(q-1)a+1}\end{aligned} \tag{19}$$

where the right-hand side is nonpositive as claimed. Therefore, although the update directions of the NLMP [where $p=(a+1)$] and the proposed algorithm are not the same, both of them correspond to the descent direction of the cost. Hence, for a sufficiently small μ, the proposed iterative update converges as well [7].

The algorithms of (15) and (16) can also be investigated in terms of their computational efficiency. The only difference between the two algorithms is the nonlinear transformation of the input vector in (16). Since the vector $\boldsymbol{X}_k^{\langle (q-1)a\rangle}$ can be rewritten as

$$\begin{aligned}\boldsymbol{X}_k^{\langle (q-1)a\rangle} &= \left[\frac{x_k^{\langle qa\rangle}}{x_k}\cdots\frac{x_{k-M+1}^{\langle qa\rangle}}{x_{k-M+1}}\right]^T\\ &= \left[\frac{x_k^{\langle qa\rangle}}{x_k}\boldsymbol{X}_{k-1}^{\langle (q-1)a\rangle}(1\colon M-1)\right]^T\end{aligned} \tag{20}$$

only the term $x_k^{\langle qa\rangle}/x_k$ is needed for recursive evaluation of $\|\boldsymbol{X}_k\|_{qa}^{qa}$ at each time step. The term $x_k^{\langle qa\rangle}/x_k$ can be computed by power series expansion, and it can be closely approximated by using a few multiplications independent of the filter length M. Therefore, the complexity of both NLMP- and FLOS-based algorithms are the same.

A. Simulation Studies

In the following set of simulations, we compare the performances of the adaptation algorithms considered in this paper in a prediction problem where the input sequence is an AR(M) α-stable process, which satisfies the following all-pole model

$$x_k = \sum_{i=1}^{M} a_i x_{k-i} + u_k \tag{21}$$

where a_i's are deterministic coefficients and u_k is an i.i.d. noise process with symmetrical α-stable (SαS) distribution. It has been shown that if $\{a_i\}$ is an absolutely summable sequence, then random variable x_k is also an SαS with the same parameters of u_k [5], [12]. Hence, the input sequence in (21) is a sequence of correlated α-stable random variables. In the following prediction simulations, the desired signal sequence d_k is set to x_{k+1}. Also, it is assumed that exact AR model order M of the input sequence is known. Therefore, in the case of a successful adaptation, the adaptive filter weights should converge to the coefficient vector $[a_1 a_2 \cdots a_M]^T$.

Here, we will first present the comparison study between FLOS-based adaptive filtering algorithms of (9) and (16). The input sequence is chosen as an $AR(2)$ sequence with coefficients $a_1 = 0.99$ and $a_2 = -0.1$. In order to investigate the dependence on the exponent of the α-stable process, we provide the results obtained with the three different exponents 1.1, 1.2, and 1.5, respectively. For more reliable results, throughout the paper all of the algorithms are run over 100 independent realizations of the input process. In Fig. 1, average tap weights of both algorithms are shown as a function of time. Subfigures correspond to the exponents 1.1, 1.2, and 1.5, respectively. For $\alpha = 1.1$, the value of the parameter p in (11) is taken as 12/11 and for $\alpha = 1.2$, and 1.5, p is taken as 7/6. We also show the performance of the FLOS-based algorithm of (16) on the same plot. As can be seen, the performance of the NLMS algorithm of (9) is far from satisfactory and gets worse for the lower values of α, while the FLOS-based algorithm of (16) converges to the optimum values around 2000 time steps.

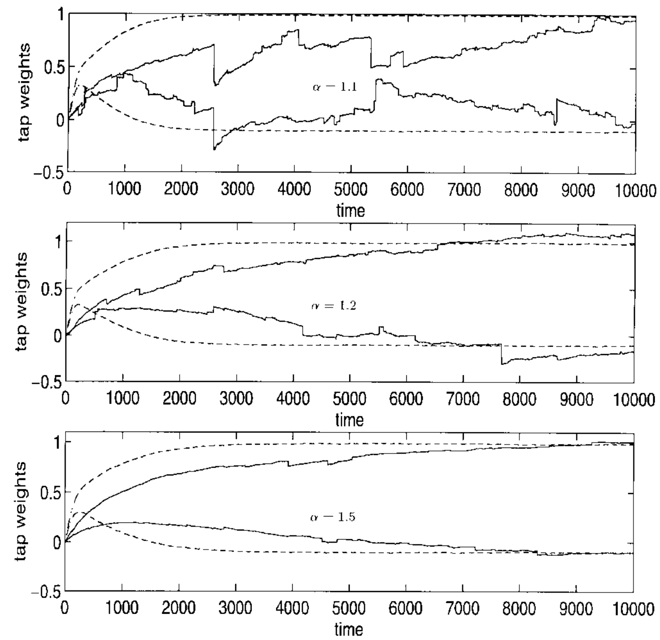

Fig. 1. Transient behavior of tap weight adaptations of (9) and the proposed FLOS-based algorithm of (16) (dashed line) for $\alpha = 1.1, \alpha = 1.2$, and $\alpha = 1.5$, respectively. The $AR(2)$ process parameters are $a_1 = 0.99$ and $a_2 = -0.1$.

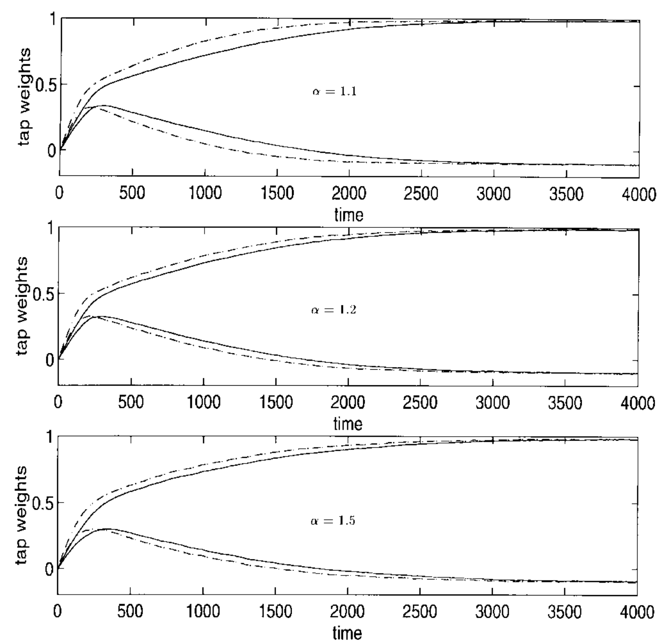

Fig. 2. Transient behavior of tap weight adaptations for the proposed FLOS-based algorithm (dashed line) of (16), and the NLMP algorithm (solid line) of (15) for $\alpha = 1.1, \alpha = 1.2$, and $\alpha = 1.5$. The $AR(2)$ process parameters are $a_1 = 0.99$ and $a_2 = -0.1$.

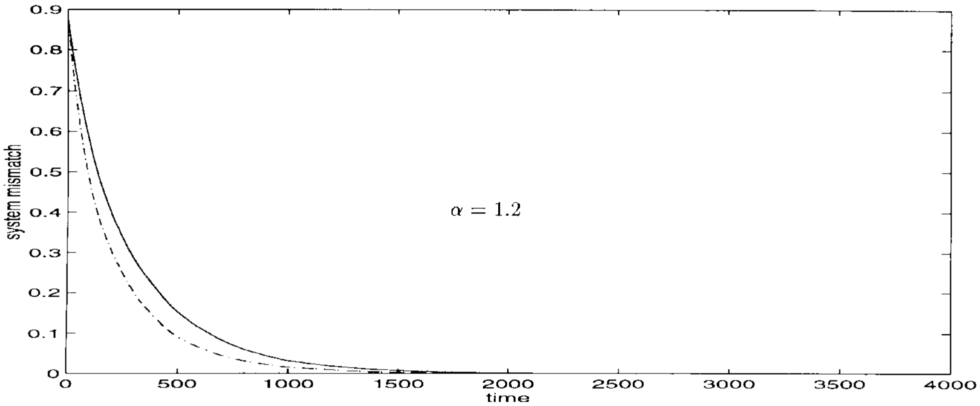

Fig. 3. The system mismatch, $\|\boldsymbol{W}_k - \boldsymbol{W}_*\|_2^2$, for the "momentum" FLOS-based algorithm of (22) (dashed line), and the "momentum" NLMP algorithm of (25) (solid line) for $\alpha = 1.2$. The $AR(5)$ process parameters are $a_1 = 0.89, a_2 = -0.152, a_3 = 0.1, a_4 = -0.197$, and $a_5 = 0.097$.

The same system identification problem is considered for the simulation studies of the proposed algorithm and NLMP algorithm of (15) in Fig. 2. We plot the performance of the proposed FLOS-based algorithm of (16) and the NLMP algorithm of (15) for $\alpha = 1.1, 1.2,$ and 1.5, respectively. As can be seen from the figure, the proposed algorithm converges to the optimum value around 3000 time steps while the NLMP algorithm converges around 4000 time steps. To get a fair comparison between the algorithms, the step size of the algorithms is adjusted so that the steady state variances of the tap weights are equal.

The proposed algorithms are all based on the instantaneous value of the gradient vector. When the signals are Gaussian and stationary, ignoring the past and using only the instantaneous gradient at time k is a reasonable approximation [13]. In impulsive noise environments, the current observation may be an outlier resulting in a significant deviation from the actual gradient of the cumulative cost function. The use of more than one term to estimate the gradient improves the robustness of convergence behavior of the algorithm. Therefore, we also investigated the following "momentum"-type update:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu f(\boldsymbol{t}_{k-j}, \cdots, \boldsymbol{t}_k) \tag{22}$$

where

$$\boldsymbol{t}_n = \frac{e_n^{\langle a \rangle}}{\|\boldsymbol{X}_n\|_{qa}^{qa} + \lambda} \boldsymbol{X}_n^{\langle (q-1)a \rangle} \tag{23}$$

and

$$f(\boldsymbol{t}_{k-j}, \cdots, \boldsymbol{t}_k) = \sum_{n=k-j}^{k} \boldsymbol{t}_n. \tag{24}$$

The above update is compared to similarly obtained "momentum"-type NLMP algorithm given by

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mu f(\boldsymbol{z}_{k-j}, \cdots, \boldsymbol{z}_k) \tag{25}$$

where

$$\boldsymbol{z}_n = \frac{e_n^{\langle p-1 \rangle}}{\|\boldsymbol{X}_n\|_p^p + \lambda} \boldsymbol{X}_n, \quad \text{for } k - j \le n \le k \tag{26}$$

and $f(\cdot)$ is given in (24). These "momentum" versions of the update equations requires additional memory to store the past estimates of the gradient, and an additional scalar vector multiplication. Therefore, their use can be justified easily, if their performance is better than the instantaneous gradient-based approaches.

In Fig. 3, we plot the system mismatch for $\alpha = 1.2$ by generating an $AR(5)$ α-stable random process for the "momentum" FLOS-based algorithm of (22) and for the "momentum" NLMP algorithm of (25) to compare their relative performances. The $AR(5)$ α-stable process parameters are $a_1 = 0.89, a_2 = -0.152, a_3 = 0.1, a_4 = -0.197,$ and $a_5 = 0.097$. We take $j = 2$ in this plot. It is clear that the proposed "momentum" FLOS-based algorithm of (22) performs better in this case. In Fig. 4, we plot for the system mismatch of the $AR(5)$ process defined above for various j values of the algorithms (22) and (25), respectively. For $j = 1$, the proposed FLOS-based algorithm of (22) converges around 2500 time steps, whereas for $j = 5$, it converges around 1000 time steps. Similarly, the algorithm of (25) converges in 3000 time steps for $j = 1$ and 1000 time steps for

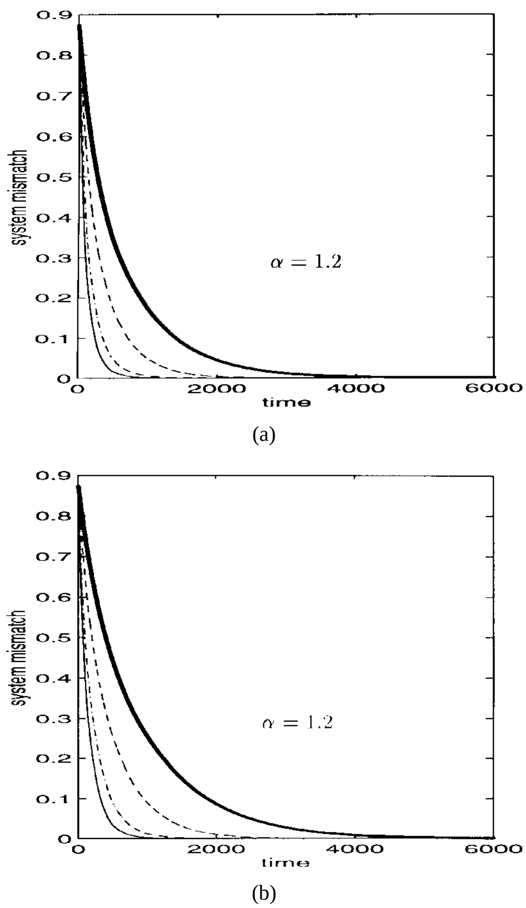

Fig. 4. Algorithms for $j = 0$ (heavy solid line), $j = 1$ (dashed line), $j = 3$ (dash-dot line), and $j = 5$ (solid line). (a)"Momentum" FLOS-based algorithms of (22). (b) "Momentum" NLMP algorithm of (25).

$j = 5$. However, increasing j means also decreasing the space in the memory. From the plots it is observed that there is a great improvement in results when j is increased from 1 to 3. However, there is not much difference when j is increased from 3 to 5. We reach a point of diminishing returns at $j = 3$, and the use of too many past values of the gradient vector does not improve the convergence further. In this section, all the simulation studies are the average of 100 independent trials as well.

IV. Conclusion

In this paper, new adaptive filtering algorithms for impulsive noise environments are introduced. These algorithms are obtained as a generalization of the NLMP algorithm by using FLOS. Under the same assumptions of the LMS convergence, it is shown that the new class of algorithms converge for small enough adaptation step size. The computational complexity of the proposed algorithms is in the same order with that of the NLMP algorithm. It is observed that the new algorithms demonstrate superior performance compared to the previous methods. Also, it is demonstrated that speedup in convergence can be achieved by using a "momentum" version of the update with tolerable increase in the memory requirements.

References

[1] B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises, and applications," *SIAM Rev.*, vol. 10, pp. 422–437, 1968.

[2] S. S. Pillai and M. Harisankar, "Simulated performance of a DS spread spectrum system in impulsive atmospheric noise," *IEEE Trans. Electromag. Compat.*, vol. EMC-29, pp. 80–82, 1987.

[3] M. Bouvet and S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 621–626, 1989.

[4] D. Middleton, "Statistical physical models of electromagnetic interference," *IEEE Tran. Electromag. Compat.*, vol. EMC-19, pp. 106–127, 1977.

[5] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, pp. 986–1009, July 1993.

[6] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. New York: Springer-Verlag, 1988.

[7] O. Aríkan, A. E. Çetin, and E. Erzin, "Adaptive filtering for non-Gaussian stable processes," *IEEE Trans. Signal Processing Lett.*, vol. 1, pp. 1–3, Nov. 1994.

[8] A. L. Peressini, F. E. Sullivan, and J. J. Uhl, *The Mathematics of Nonlinear Programming*. New York: Springer-Verlag, 1988.

[9] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[10] S. C. Douglas, "A family of normalized LMS algorithms," *IEEE Signal Processing Lett.*, vol. 1, pp. 49–51, Mar. 1994.

[11] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, pp. 282–287, June 1967.

[12] Y. Hosoya, "Discrete time stable processes and their certain properties," *Ann. Prob.*, vol. 6, no. 1, pp. 94–105, 1978.

[13] J. R. Treichler, C. R. Johnson, and M. G. Larimore, *Theory and Design of Adaptive Filters*. New York: Wiley, 1986.

A New Division Algorithm Based on Lookahead of Partial-Remainder (LAPR) for High-Speed/Low-Power Coding Applications

Hyung-Joon Kwon and Kwyro Lee

***Abstract*—A new polynomial division algorithm in finite field GF(2^m) based on the lookahead of partial-remainder (LAPR) is proposed. Since our algorithm is based on partial division on group basis and lookahead technique exploiting the linearity in finite field arithmetic, it is possible to completely eliminate polynomial multiplications leading to highly increased throughput per unit time. The inherent regularity and feedforward nature of our algorithm make it possible to be fully pipelined. When pipelined, its throughput is one quotient and one remainder per clock cycle, regardless of the degree of dividend polynomial, which is orders of magnitude faster than the conventional architecture using linear feedback shift register. An area-efficient sequential architecture based on LAPR is also presented. Although the throughput rate of sequential architecture is lower than that of the pipelined one, it is still higher than that of any division architecture ever reported. Those will be shown to be efficient, regular, and easily expandable, and hence, naturally suitable for very large scale integration implementation. In systems requiring modest speed, the high-speed nature of our proposed architecture can be traded for low-power consumption by reducing clock rate. We verified the general validity of the division algorithm based on LAPR by mathematical manipulation and simulation. The superiority of our proposed architecture compared with other reported ones is demonstrated with regard to its throughput, latency delays, and power.**

***Index Terms*— Author, please supply index terms. E-mail keywords@ieee.org for info.**

I. Introduction

Division in the finite field $GF(2^m)$ is the most important building block in coding systems such as BCH (Bose–Chaudhuri–Hocquenghem) and RS (Reed–Solomon) codes for applications to communications, optical disks, portable equipment, and control and computer systems, since these coding systems are based on long polynomial divisions. The conventional architecture for division in finite fields uses linear feedback shift register (LFSR), which consists of k stage feedback shift register, where k is the degree of the divisor polynomial. A diagram for such a division architecture is shown in Fig. 1. The quantities m_0, m_1, $\cdots$, m_k are the coefficients of the divisor polynomial. However, as the high-speed requirement for real-time audio or video coding capability and the low-power requirement for portable equipment increase, this division architecture using LFSR has shown several limitations as described below.

In some very high-speed applications, the presence of a global feedback signal imposes severe constraints on the switching speed. The fact that the input to all k stages is the feedback signal forces all k stages to be synchronous, necessitating the use of a global clock. The need to distribute the global clock and the feedback signal to all stages of the architecture can seriously restrict the maximum switching speed achievable in a practical implementation. To remedy this, an alternative configuration was suggested by Tong [1], where the feedback path is pipelined such that the feedback signal goes through one delay unit before it is fed back to each shift register

Manuscript received July 12, 1996; revised December 27, 1997. This paper was recommended by Associate Editor B. W. Lee.

The authors are with Korea Advanced Institute of Science and Technology, Yusung-Gu, Taejun 305–71, Korea.

Publisher Item Identifier S 1057-7130(99)01820-0.