An Evolutionary Computation Embedded IIR LMS Algorithm

Deependra Talla[1], Sathyanarayan S. Rao[2] and Lizy K. John[1]

deepu@ece.utexas.edu, rao@ece.vill.edu, ljohn@ece.utexas.edu

[1] Department of Electrical and Computer Engineering, The University of Texas at Austin, TX

[2] Department of Electrical and Computer Engineering, Villanova University, PA

Abstract – *An improved Infinite Impulse Response (IIR) Least Mean Squares (LMS) algorithm using parallel filters and evolutionary programming techniques is introduced. IIR filters have the attractive property that they require fewer computations than a corresponding FIR filter, but they are prone to instability and local minimum problems. Evolutionary algorithms are good in global optimization scenarios, but are computationally very expensive. Adaptive filter weights for a given step size and initial weight vectors may not lead to optimal solutions. In this paper we extend the IIR LMS algorithm by embedding an evolutionary computation, and also simultaneously implement multiple filters (different initial weight vectors) to achieve optimal solutions.*

I. INTRODUCTION

Adaptive filters are now commonly used in a wide range of Digital Signal Processing (DSP) systems. Commercial systems that rely on adaptive filtering in one way or another include high-speed modems, echo-cancellation in speakerphones, interference removal in medical imaging, active noise control applications, on-line system identification in chemical plants, linear prediction, and beamforming in radio astronomy.

For practical reasons, first generation adaptive systems generally employed the basic Finite Impulse Response (FIR) linear filter structures along with simple gradient descent or least squares algorithms such as Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithms. However, they have certain performance limitations. Some applications require a more general Infinite Impulse Response (IIR) filtering structure as in the exact restoration of a received signal corrupted by multipath distortion. Moreover, the IIR filter offers potential performance improvement by proving to be less computationally expensive than equivalent FIR filters. A recursive IIR filter generally provides better performance than a FIR filter that has the same number of coefficients. This is because the desired response can be approximated more effectively by the output of a filter that has both poles and zeros compared to one that has only zeros [1]. To achieve a specified level of performance, an IIR filter generally requires considerably fewer coefficients than a corresponding FIR filter.

However, some practical problems still exist in the use of adaptive IIR filters. As the error surface of IIR filters is usually non-quadratic and multimodal with respect to the filter coefficients, learning algorithms for IIR filters can easily be stuck at local minima and cannot converge to the global minimum [2]. Whereas the error surface of FIR filters is quadratic and unimodal for linear system problems. Besides the local minima problem, the stability of an IIR filter during its adaptation is another issue. IIR filters will become unstable if the poles move outside the unit circle during the adaptation process. Therefore, stability monitoring is of vital importance, especially when adapting high-order IIR filters.

Evolutionary algorithms [3], such as genetic algorithms (Gas), evolutionary programming (EP) and evolutionary strategies (ES) have recently received much attention for global optimization problems such as the error surfaces of IIR filters. These evolutionary algorithms are heuristic population-based search procedures that incorporate random variation and selection. But as the search space for these algorithms is extremely large, the randomization process may lead to time wasted in searching along incorrect directions. This leads to slow convergence and high computational complexity.

Several contributions have directed their interests in employing evolutionary algorithms for IIR filters [4][5][6][7][8] in one way or another. In [4], a new learning algorithm for adaptive IIR filtering using a genetic search approach is presented. Using both the LMS algorithm for IIR filters and genetic programming techniques, the authors demonstrate faster convergence and global search capability by comparing with pure LMS and genetic algorithm implementations. In [5][6] and [7], the authors introduce evolutionary digital filtering for IIR adaptive digital filters. In this approach, instead of using any gradient-based algorithms such as the LMS, several digital filters (order of 1000) are used and the best output among them is selected. Adapting the filter coefficients is done in regular intervals using evolutionary programming techniques. This is a very computationally complex algorithm, especially if implemented on a single processor.

Initiated by the merits and shortcomings of the IIR LMS algorithm and pure evolutionary computations, embedding an evolutionary computation in the pure LMS algorithm will help the LMS algorithm to escape the local minima problem. Also since the LMS is a directed search, evolutionary computation will benefit from escaping incorrect direction searches. In this paper, we present an improved IIR LMS algorithm implementing multiple filters which exposes more parallelism at the algorithmic level and also using some evolutionary computation when adapting filter weights during each iteration of the algorithm.

II. ALGORITHM

The IIR LMS algorithm is an extension of the FIR LMS algorithm. A direct-form implementation of the recursive IIR filter is preferred over the parallel and lattice forms in view of the computational complexity associated with the latter structures. The IIR direct-form filter can be constructed as:

$$y(n)=\sum_{i=1}^{L}a_i y(n-i)+\sum_{j=0}^{M}b_j x(n-j) \qquad (1)$$

where x(n) is the current input value, y(n) is the current output value. The learning algorithm of the adaptive IIR filter is used to adjust the feedback and feedforward coefficients, a_i and b_j respectively for a particular input and output to optimize a performance criterion that generates a suitable estimate based on a desired response d(n).

Let the weight vector θ and data vector X(n) be defined as:

$$\theta=[a_1...a_L b_0...b_M]^T \qquad (2)$$

$$X(n)=[y(n-1)..y(n-L)x(n)..x(n-M)]^T \qquad (3)$$

The LMS algorithm can be represented as:
For each n:

$$y(n)=\theta^T(n)X(n) \quad \underline{\text{filter output}} \qquad (4)$$

$$\nabla_\theta y(n)=X(n)+\sum_{i=1}^{L}a_i\nabla_\theta y(n-i) \quad \underline{\text{gradient estimate}} \qquad (5)$$

$$\theta(n+1)=\theta(n)+\mu[d(n)-y(n)]\nabla_\theta y(n) \quad \underline{\text{coefficient update}} \qquad (6)$$

where μ is a constant step size.

The instantaneous estimation error e(n) is:

$$e(n) = d(n) - y(n) \qquad (7)$$

For each value of n, Eq. (4) is to produce the filter output and Eqs. (5) and (6) are then used to compute the next set of coefficients θ (n+1).

In this paper, we extend the original IIR LMS algorithm that was discussed above by simultaneously implementing multiple filters each with different and unique initial weight vectors and gradient estimate and coefficient vectors that are updated during each iteration. We define several digital filters with filter weight vectors θ_1, θ_2 .., θ_n. The IIR LMS algorithm is applied to each filter separately with the same input signal and desired signal as shown in Fig. 1 below. The convergence of the LMS algorithm very much depends on the choice of the step size and the choice of initial values for the filter coefficients. Having multiple filters guarantees better convergence and smaller mean squared error than when using a single filter. The best output value is selected from a summation of a block of mean squared error values for each filter.

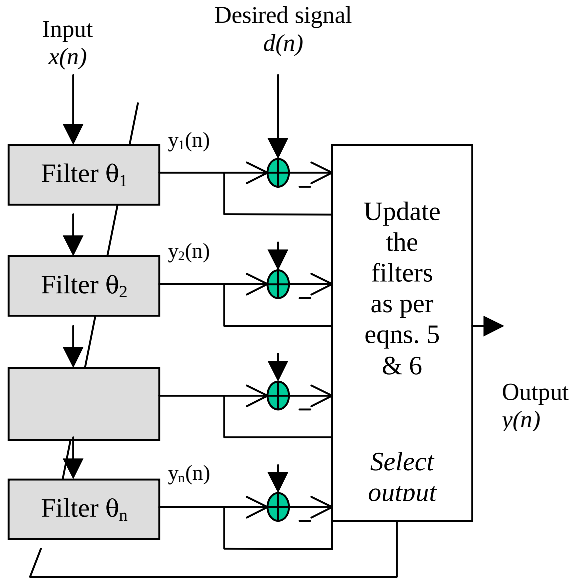


Fig. 1. Block diagram of IIR LMS algorithm extended with multiple parallel filters.

III. STABILITY MONITORING AND EVOLUTIONARY COMPUTATION

IIR filters become unstable if the poles move outside the unit circle during the adaptation process. As a result, the output can grow without bounds and the filter breaks down. Stability monitoring is essential when adapting IIR filters. During each adaptation of the filter coefficients, stability of the filter needs to be ascertained before computing the output signal value.

Two methods were investigated for checking the stability during the perturbation of the adaptive IIR filter. The first method is based on Kharitonov's theorem [9], which requires the testing of four related polynomials to ascertain the stability of the filter. This method is general and can be used to stabilize any filter that has a proper rational transfer function. Application of the Kharitonov's theorem with additional constraints yields optimal coefficient bounds. Overall the complexity of this method is high [4] and it has been shown that it also restricts the size of the coefficient space.

In the second implementation, the direct-form IIR filter is converted into its equivalent lattice form. The requirement for stability transforms into verifying if each of the reflection coefficients has an absolute magnitude less than 1. Only the direct-form feedback coefficients need to be converted into their equivalent feedback reflection coefficients for stability monitoring.

After testing the poles for stability, the filter coefficients can be updated by using several approaches [1]. The simplest approach is to use the previous stable coefficients if the current set of filter coefficients is found to be unstable. However, this method is not deemed to be robust for some applications. Another approach is to set the value of the reflection coefficients that are outside the permissible range to a value close to the nearest bound [4] and convert them back to the direct-form coefficients.

In our approach, whenever the LMS algorithm has a slow convergence or the filter coefficients are unstable, we embed an evolutionary computation by randomly perturbing the values of the previous stable filter coefficients. The coefficients are produced in a certain controlled manner; particularly, they are tested if the stability criteria are satisfied. If found unstable again after producing new filter coefficients, we use the previous stable filter coefficients as the current new filter coefficients during that iteration. Depending on whether the convergence is too slow or the filter is unstable again in the next generation, we continue to evolve new filter coefficients once each per iteration.

Alternatively, the evolutionary computation can be repeated multiple times per each generation/iteration for robustness. However, this increases the serial computation in the algorithm significantly. Moreover by implementing several parallel filters, even if one filter is not robust, another filter can be used for output generation. Over several iterations, but without increasing serial computation, the unstable or slowly converging filter can be made to adapt in a correct direction. Each set of filter coefficients are updated as follows when using evolutionary computation:

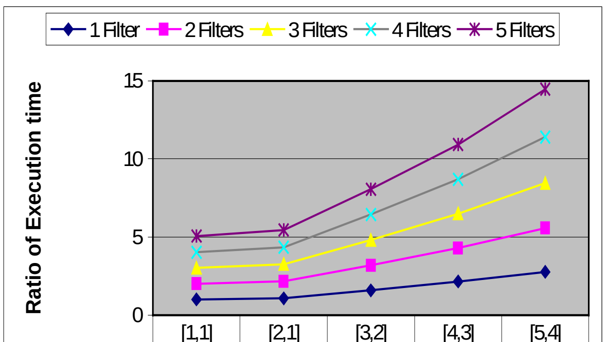
$$\vec{\theta}_{new} = \vec{\theta}_{old} + \sigma D \quad (8)$$

where σ is a random number in the range [-1, 1] and D is the allowable offset range for each evolution.

IV. COMPUTATIONAL COMPLEXITY OF MULTIPLE PARALLEL FILTERS

Using several parallel filters for the IIR LMS algorithm increases the computationally complexity linearly with the number of filters. However, each filter can be operated on a different processor thereby keeping the serial computation of the algorithm the same as that of one filter. For slow sampling rates, several filters can be implemented on a single processor given today's superscalar and very long instruction word (VLIW) processors.

To obtain the computational complexity, we obtained the number of clock cycles consumed by a Pentium II processor when implementing our algorithm. Fig. 2 shows the results. Five different filter orders for [L, M] as per equation (1) – [1,1], [2,1], [3,2], [4,3] and [5,4] are studied. The number of filters is varied from 1 to 5. The baseline case is a single filter with a [1,1] configuration given an execution time value of 1. The remaining 24 configuration's execution times are ratios with the baseline filter. The experiment is conducted for 10,000 iterations of the IIR LMS algorithm.



	[1,1]	[2,1]	[3,2]	[4,3]	[5,4]
1 Filter	1	1.08	1.59	2.14	2.76
2 Filters	2.01	2.16	3.18	4.3	5.57
3 Filters	3.01	3.26	4.8	6.5	8.46
4 Filters	4.03	4.35	6.44	8.69	11.4
5 Filters	5.07	5.46	8.06	10.93	14.48

Filter Configuration

Fig. 2. Computational complexity of multiple filters with different configurations.

V. CONVERGENCE RESULTS

The convergence behavior of the LMS algorithm depends very much on the choices of step size and the initial values of the filter coefficients. Fig. 3 below shows the effect of different initial values of the filter coefficients for three parallel filters operating on the same input and desired signals. A unit signal with white noise was used as the input signal. The filter configuration was M=3 and L=2 with the step size being 0.01. The results shown below are an average of 100 independent runs.

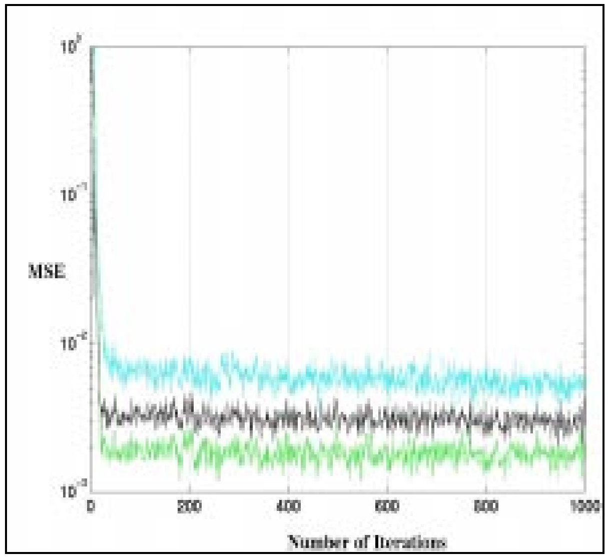


Fig. 3. Different convergence characteristics with different initial filter coefficients

Embedding an evolutionary computation in the IIR LMS algorithm with parallel filters helps in stability monitoring. Having only one filter, if the IIR LMS filter goes unstable, the output value is distorted heavily until the learning algorithm finds the right direction to converge. But by having multiple filters, even if one filter becomes unstable, another filter can generate the output while the unstable filter is made to follow the right direction for convergence. Fig. 4 below shows the result of one filter going unstable during adaptation while the other two filters converge normally. The unstable filter slowly converges as the number of iterations increase. The filter configuration was M=3 and L=2, with a step size of 0.04. Each of the sets of filter coefficients was randomly initialized with one set however, always being initialized to all zeros.

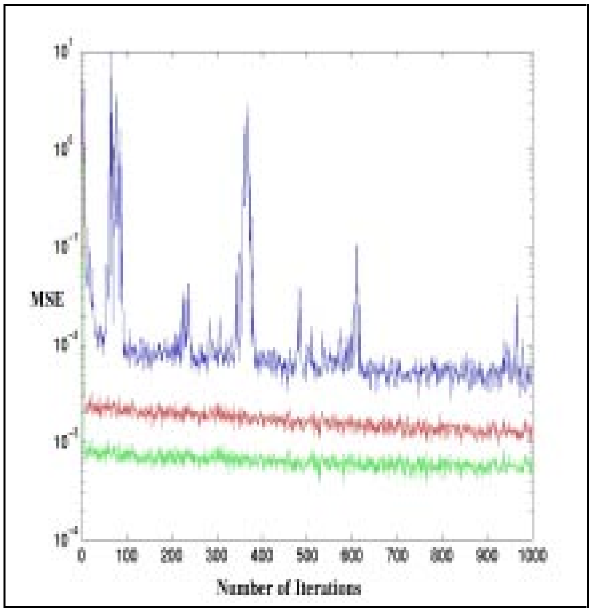


Fig. 4. Learning curves (mean square error) of three parallel filters

VI. CONCLUDING REMARKS

In this paper, we extended the original IIR LMS algorithm in two ways. First, we implemented multiple parallel filters each initialized with a different set of coefficients. Complexity of different order filters with different number of parallel filters was measured. Next, we embedded an evolutionary computation into the IIR LMS algorithm mainly for stability monitoring. We used minimal computations to increase parallelism in the algorithm. Results show that even if one filter goes unstable, evolutionary computation can redirect the search in correct directions while parallel filters that are not unstable contribute to the output.

ACKNOWLEDGMENTS

This work is supported in part by the State of Texas Advanced Technology Program grant #403, the National Science Foundation under Grants CCR-9796098 and EIA-9807112, and by Dell, Intel, Microsoft and IBM.

REFERENCES

[1] J.J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, pp. 4-21, Apr. 1989.

[2] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[3] D.B. Fogel and K. Chellapilla, "Revisiting evolutionary programming," *AeroSense'98: Aerospace/Defense Sensing and controls*, Orlando, Apr. 1998.

[4] S.C. Ng, S.H. Leung, C.Y. Chung, A. Luk and W.H. Lau, "The genetic search approach: A new learning algorithm for adaptive IIR filtering," *IEEE Signal Processing Mag.*, vol. 13, no. 6, pp. 38-46, Nov. 1996.

[5] M. Abe and M. Kawamata, "Evolutionary digital filtering for IIR adaptive digital filters based on the cloning and mating reproduction," *IEICE Trans. Fundamentals*, vol. E81-A, no. 3, pp. 398-406, Mar. 1998.

[6] M. Abe and M. Kawamata, "A single DSP implementation of evolutionary digital filters," Proc. *IEEE Intl. Workshop on Intelligent Sig. Processing and Comm. Systems*, pp. 253-257, Nov. 1998.

[7] M. Abe, M. Kawamata and T. Higuchi, "Convergence behavior of evolutionary digital filters on a multiple-peak surface," *Proc. IEEE Intl. Sym. on Circuits and Systems*, vol. 2, pp. 185-188, May 1996.

[8] S. Sundaralingam and K.C. Sharman, "Genetic evolution of adaptive filters," *Proc. DSP UK*, pp. 47-53, Nov. 1997.

[9] T. Brennan, "Bounding adaptive filter poles using Kharitonov's theorem," *Proc. 22nd Asilomar Conf. Signals, systems, Computers*, pp. 658-662, Nov. 1988.

[10] D. Talla, S.S. Rao and K. Chellapilla, "TMS320C40 based implementation of a real-time adaptive IIR filter," *Proc. Intl. Conf. Signal Proc. Appl. and Tech. ICSPAT-97*, pp. 133-137, Sep. 1997.

[11] S.S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1986.

[12] S.D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. Circuits & Syst.*, vol. CAS-28, no. 6, pp. 603-606, Jun. 1981.

[13] S.D. Stearns, R.A. David and D.M. Etter, "A survey of IIR adaptive filtering algorithms," *Proc. IEEE Intl. Conf. on Acoustics, speech and signal processing*, pp. 635-638, 1982.