A Particle Swarm Optimization – Least Mean Squares Algorithm for Adaptive Filtering

D. J. Krusienski and W. K. Jenkins
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA

ABSTRACT

A particle swarm optimization-least mean squares (PSO-LMS) algorithm is presented for adapting various classes of filter structures. The LMS algorithm is widely accepted as the preeminent adaptive filtering algorithm because of its speed, efficiency, and provably convergent local search capabilities. However, for multimodal error surfaces, a global search algorithm, such as PSO or the genetic algorithm (GA), is required. The proposed PSO-LMS hybrid algorithm combines the advantageous properties of the two conventional algorithms to provide enhanced performance characteristics.

Introduction

In many adaptive signal processing problems the mean squared error surface that exists as a hyper-surface in the multidimensional parameter space may be ill-conditioned in the sense that it is a non-quadratic surface that possesses local minima and saddle points, in addition to a global minimum that represents the optimal solution. It is well known that such surfaces inhibit efficient optimization for certain classes of adaptive systems, such as infinite impulse response (IIR) adaptive filters, nonlinear polynomial adaptive filters, and neural networks.

Stochastic optimization algorithms aim at increasing the probability of encountering the global minimum, without performing an exhaustive search of the entire parameter space. Unlike gradient descent techniques, the performance of stochastic optimization techniques in general is not dependent upon the filter structure. Therefore, these types of algorithms are capable of globally optimizing any class of adaptive filter structures or any types of objective functions by assigning the parameter estimates to represent filter tap weights, neural network weights, or any other possible parameters of the unknown system model (even the exponents of polynomial terms).

Congregational LMS

Gradient-based optimization techniques attempt to estimate the gradient of the error surface and proceed to an optimum by following the negative direction of this estimated gradient. These algorithms are well known, widely used, and proven simple, effective, and convergent local optimization techniques. The most notable of these algorithms is the least mean squares (LMS) algorithm [3]. The problem is that, being a local optimization technique, gradient decent is limited because it is unable to converge to the global optimum on a multimodal error surface if the algorithm is not initialized in the valley of the global optimum.

A variation of the LMS algorithm, the congregational LMS (CON-LMS) [1], attempts to locate the global optimum by running several LMS algorithms in parallel, initialized with different initial coefficients. The notion is that a larger, concurrent sampling of the error surface will increase the likelihood that one process will be initialized in the global optimum valley. This technique does have potential, but it is inefficient and may still suffer the fate of a standard gradient technique in that it will be unable to locate the global optimum if none of the initial estimates is located in the global optimum valley. By using a similar congregational scheme, but one in which information is collectively exchanged between estimates and intelligent randomization is introduced, structured stochastic search algorithms are able to hill-climb out of local minima. This enables the algorithms to achieve better, more consistent results using a fewer number of total estimates. There are several different structured stochastic search approaches in the adaptive filtering literature, most notably simulated annealing [8], and evolutionary algorithms such as the genetic [8][9][10] and particle swarm optimization [2].

Particle Swarm Optimization

Particle swarm optimization was first developed in 1995 by Eberhart and Kennedy [2] rooted on the notion of swarm intelligence of insects, birds, etc. The algorithm attempts to mimic the natural process of group communication of individual knowledge that occurs when such swarms flock, migrate, forage, etc. in order to achieve some optimum property such as configuration or location.

As with CON-LMS, PSO begins with a random population of individuals; here termed a swarm of particles. Again, each particle in the swarm is a different possible set of the unknown parameters to be optimized. Each particle represents a point in the solution space that has a relative fitness determined by evaluating the

0-7803-8622-1/04/$20.00

parameters with respect to a predetermined fitness function that has an extremum at the desired optimal solution. The goal is to efficiently search the solution space by swarming the particles toward the best fit solution encountered in previous iterations with the intent of encountering better solutions through the course of the process and eventually converging on a single minimum error solution.

The standard PSO algorithm begins by initializing a random swarm of M particles, each having R unknown parameters to be optimized. Unless there is prior knowledge about the parameter space, the initial particles are typically distributed uniformly about the presumed parameter space to facilitate a global search. At each iteration, the fitness of each particle is evaluated according to the selected fitness function. The algorithm stores and progressively replaces the most fit parameters of each particle ($pbest_i$, $i=1,2,\ldots,M$) as well as a single most fit particle (*gbest*) as better fit parameters are encountered. The parameters of each particle (p_i) in the swarm are updated at each iteration (n) according to the following equations:

$$\begin{aligned}\overline{vel_i}(n) &= w * \overline{vel_i}(n-1)\\ &+ acc_1 * diag\left[e_1, e_2, \ldots, e_R\right]_1 * (gbest - p_i(n-1))\\ &+ acc_2 * diag\left[e_1, e_2, \ldots, e_R\right]_2 * (pbest_i - p_i(n-1))\end{aligned} \tag{1}$$

$$p_i(n) = p_i(n-1) + \overline{vel_i}(n) \tag{2}$$

where $\overline{vel_i}(n)$ is the velocity vector of particle I, e_r is a vector of random values $\in (0,1)$, acc_1, acc_2 are the acceleration coefficients toward *gbest and* $pbest_i$, respectively, and w is the inertia weight.

It can be gathered from the update equations that the trajectory of each particle is influenced in a direction determined by the previous velocity and the location of *gbest* and $pbest_i$. The acceleration constants are typically chosen in the range 0-2 and serve dual purposes in the algorithm. For one, they control the relative influence toward *gbest* and $pbest_i$ respectively by scaling each resulting distance vector. Secondly, the two acceleration coefficients combined form what is analogous to the step size of an adaptive algorithm. Acceleration coefficients closer to 0 will produce fine searches of a region, while coefficients closer to 1 will result in lesser exploration and faster convergence. The random e_i vectors have R different components, which are randomly chosen in the range 0-1. This allows the particle to take constrained randomly directed steps in a bounded region between *gbest* and $pbest_i$. The acceleration coefficients should be chosen in conjunction with the random e_i components for a desired average step size.

The inertia weight controls the influence of the previous velocity. It is typically set to decay from 1 to 0 during some adequate interval in order to allow the algorithm to converge on *gbest*.

When a new *gbest* is encountered during the update process, all other particles begin to swarm toward the new *gbest*, continuing the directed global search along the way. The search regions continue to decrease as new $pbest_i s$ are found within the search regions. When all of the particles in the swarm have converged to *gbest*, the *gbest* parameters characterize the minimum error solution determined by the algorithm.

In adaptive filtering, the mean squared error (MSE) between the output of the unknown system and the output of the AF is the typical cost function, and will hence be used for the fitness evaluation of each particle in the on-line form of PSO. For an adaptive system identification configuration, the windowed MSE cost function is as follows:

$$J(n) = \min_i \left(\frac{1}{N} \left[\sum_{k=0}^{N-1} (d(n-k) - y_{k,i}(n))^2 \right] \right) \tag{3}$$

$$y_{k,i}(n) = f[x(n-k-1), x(n-k-2), \ldots, x(n-k-L) \tag{4}$$

where $d(n)$ is the desired signal, $y_{k,i}(n)$ is the adaptive filter output, $f(\cdot)$ is a linear or nonlinear operator, N is the length of the window that the error is averaged, and L is the amount of delay in the filter. The AF output $y_{k,i}(n)$ may also be a function of past values of itself if it contains feedback, or also a function of intermediate variables if the AF has a cascaded structure. When $J(n)$ is minimized the AF parameters provide the best possible representation of the unknown system.

Certain enhancements of the standard PSO algorithm to improve the overall efficiency and performance were presented in references [4] and [5]. The advantageous properties of these enhancements manifest themselves differently in different types of problems. However, it was found that the inclusion of three particular enhancements provides substantial performance improvements in virtually all variations of the PSO [5]. These three enhancements are i) mutation, ii) re-randomization about gbest, and iii) adaptive inertia operations. When these are added to the basic PSO algorithm, the result is referred to as the modified PSO (MPSO). The MPSO algorithm is designed to balance convergence speed and search quality tradeoffs, and by so doing provides significantly improved performance compared to the conventional PSO.

PSO-LMS Hybrid

By combining the advantageous features of search algorithms, hybrid algorithms can be derived that are sometimes superior to the original algorithms. The goal is to cleverly devise a combination of the dominant global search techniques with convergent local search techniques to give the greatest performance, without forsaking simplicity or computational complexity whenever possible.

One weakness of PSO is that its local search is not guaranteed convergent; its local search capability lies primarily in the swarm size and search parameters. Also, the PSO search tends to stagnate when the convergence speed is maximized, if corrective measures are not taken [2]. On the other hand, the problem with simply running a brute-force population of N independent algorithms is that there is no collective information exchange between population members, which makes the algorithm inefficient and prone to the local minimum problem of standard LMS. Therefore, it is desirable to combine the convergent local search capabilities of the LMS algorithm with the global search of PSO.

When initialized in the global optimum valley, the LMS algorithm can be tuned to provide an optimal rate of convergence with out fear of encountering a local minimum. Therefore, by using a structured stochastic search, such as PSO, to quickly focus the population on regions of interest, an optimally tuned LMS algorithm can take over and provide better results than standard LMS. A generalized form of this PSO-LMS Hybrid algorithm is presented here, which can easily be extended for IIR or LMS back-propagation updates for nonlinear structures. For a general adaptive filter structure, the LMS update takes the form:

$$w(n) = w(n-1) + \mu e(n-1)\nabla y(n-1) \qquad (5)$$

where $e(n)$ is the instantaneous error between the desired signal and filter output, $\nabla y(n)$ is the gradient of the output with respect to the filter parameters, and μ is the step size. This update can be considered a directional vector, similar to those used to generate the particle updates of PSO. The LMS step size μ should be chosen according to the guidelines given in [3], to provide stability and the desired convergence properties. To form the PSO-LMS hybrid, the LMS update from equ. (5) is combined with to the PSO particle update from equ. (2) to create the hybrid update:

$$\begin{aligned} p_i(n) = p_i(n-1) &+ c_1 \overline{vel_i}(n) \\ &+ c_2 \mu e(n-1)\nabla y(n-1) \end{aligned} \qquad (6)$$

where c_1 and c_2 are scaling factors that control the relative influence of the PSO and LMS directions, respectively. These scaling factors should be chosen such that $c_1 + c_2 = 1$ in order to control the stability of the algorithm. The principle is to decrease the influence of the more global PSO component as the algorithm progresses, which will act to increase the influence of the LMS component in order to provide the desired convergence properties.

There are several advantages to this hybrid approach. First, the LMS component of this algorithm is capable of tracking a dynamic plant, which could also be used to re-randomize the PSO search. Another advantage of this hybrid approach is that smaller population sizes can presumably be used compared to classical PSO or congregational LMS, with similar performance results for relatively compliant error surfaces. This is due to the fact that the local search capabilities of the PSO are directly related to the population size, which can be relaxed due to the convergent properties of LMS, without sacrificing the initial global search.

Simulation Examples

In the following examples, the properties of CON-LMS, PSO, and PSO-LMS are compared for several IIR and nonlinear system identification problems. All algorithms were initialized with the same population of real-valued parameters and allowed to evolve. The window length, *N*, was set to 100 in each case. For each simulation, the MSE is averaged over 50 successful trials in which the algorithm converged to the neighborhood of the global optimum. The specifics of each algorithm are as follows:

PSO: The classical PSO algorithm is implemented with both acceleration constants weighted equally at 1.2, giving an average step size of approximately 0.6. The acceleration constant was chosen to give a reasonable balance between the search quality and convergence speed for each case. The inertia weights are set to 0.5 initially and linearly decay to zero by the last trail.

PSO-LMS: The PSO-LMS algorithm is implemented as described by equ. (6). As with classical PSO, both acceleration constants are weighted equally at 1.2 and the inertia weights linearly decay from 0.5 to zero. The step sizes of the LMS contribution were selected to give the best performance. The weighting of the PSO/LMS contributions were chosen to provide a smooth transition at the point that PSO stagnates.

CON-LMS: The congregational LMS algorithm [1] is implemented where a population of independent LMS algorithms is adapted. The step sizes were selected to give the best performance.

IIR Identification

For this example the plant, a second order pole-zero filter taken from [11], is given as:

$$H_{PLANT}(z^{-1}) = \frac{1.25z^{-1} - 0.25z^{-2}}{1 - 0.3z^{-1} + 0.4z^{-2}}$$

The adaptive filter is a matched second order structure:

$$H_{AF}(z^{-1}) = \frac{p_i^1 + p_i^2 z^{-1} + p_i^3 z^{-2}}{1 + p_i^4 z^{-1} + p_i^5 z^{-2}}$$

The step size of the LMS contribution of PSO-LMS was selected as 0.01; the CON-LMS step size was selected as 0.01. The experimental results for this example using a population of 20 are shown in Figure 1.

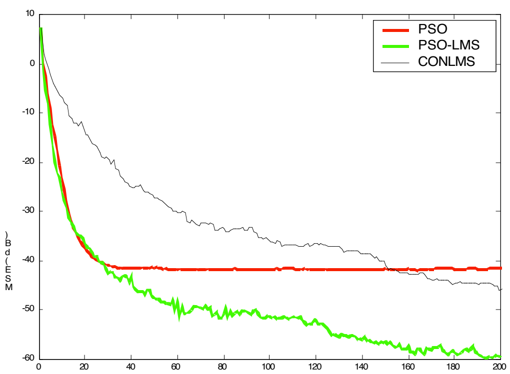

Figure 1. IIR system identification.

In this example, as well as in many other experimental IIR examples studied in [5], the CON-LMS algorithm exhibits the slowest convergence rate due to the fact that there is no information transfer between the estimates. The PSO-LMS hybrid provides the LMS algorithm with a better starting point, displaying a similar rate of convergence to CON-LMS as the PSO portion begins to stagnate. In several of the cases, the PSO-LMS hybrid achieved the fastest initial convergence due to the incorporated gradient descent. This makes the algorithm favorable for situations having a significant noise floor requiring fast convergence. Both of the LMS based algorithms are capable of eventually attaining the noise floor when the number of generations is increased, assuming that they are not trapped in a local minimum.

Matched Order Volterra Identification

In this example, a matched structure truncated Volterra AF is used to identify the truncated Volterra plant taken from [7]

$$y[n] = -0.64x[n] + x[n-2] + 0.9x^2[n] + x^2[n-1]$$

$$\begin{aligned}\hat{y}[n] = {} & p_i^1 x[n] + p_i^2 x[n-1] + p_i^3 x[n-2] \\ & + p_i^4 x^2[n] + p_i^5 x^2[n-1] + p_i^6 x^2[n-2] \\ & + p_i^7 x[n]x[n-1] + p_i^8 x[n]x[n-2] + p_i^9 x[n-1]x[n-2]\end{aligned} \tag{7}$$

The step size of the LMS contribution of PSO-LMS was selected as 0.1; the LMS step size was selected as 0.1. The results given in Figure 2 illustrate the learning curves with an SNR of 80dB.

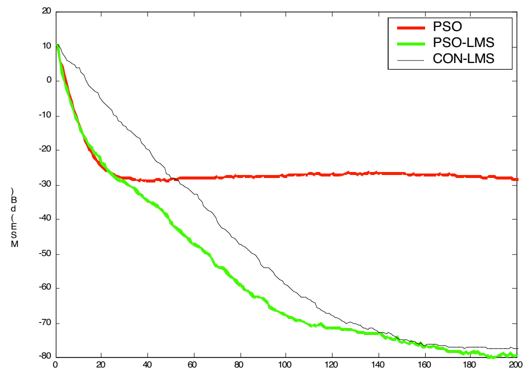

Figure 2. Matched order Volterra identification.

Black-box Volterra Identification

In this example, the identification of an LNL cascade system model taken from [7] is performed using a truncated Volterra adaptive filter. The LNL plant consists of a 4th order Butterworth lowpass filter followed by a 4th power memoryless nonlinear operator, followed by a 4th order Chebyshev lowpass filter, as shown in Figure 3. This system is a common model for

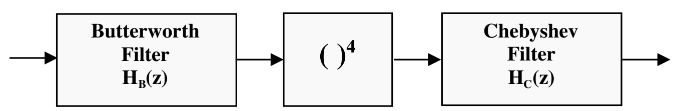

Figure 3. Black-box system.

satellite communication systems in which the linear filters model the dispersive transmission paths to and from the satellite, and the nonlinearity models the traveling wave tube (TWT) transmission amplifiers operating near the saturation region:

$$\hat{H}_B(z^{-1}) = \frac{(0.2851 + 0.5704z^{-1} + 0.2851z^{-2})(0.2851 + 0.5701z^{-1} + 0.2851z^{-2})}{(1 - 0.1024z^{-1} + 0.4475z^{-2})(1 - 0.0736z^{-1} + 0.0408z^{-2})}$$

$$\hat{H}_C(z^{-1}) = \frac{(0.2025 + 0.288z^{-1} + 0.2025z^{-2})(0.2025 + 0.0034z^{-1} + 0.2025z^{-2})}{(1 - 1.01z^{-1} + 0.5861z^{-2})(1 - 0.6591z^{-1} + 0.1498z^{-2})}$$

The truncated Volterra adaptive filter relation used to model this system is given in equ. (7). The learning curves for this example are given in Figure 4. Note that Figure 4 also includes results for the modified PSO (MPSO) and the genetic algorithm (GA) to provide further comparisons.

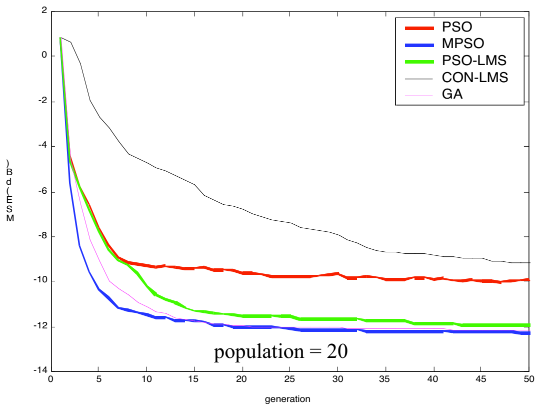

Figure 4. Black-box Volterra identification.

It is observed from Figures 2 and 4 that for these nonlinear system identification examples the conventional PSO stagnates, while the PSO-LMS continues to improve from the stagnation point of the conventional PSO at a rate similar to CON-LMS. The convergence speeds of the LMS based algorithms are more rapid in these cases because the coefficients are linear combinations of the ranked-ordered input terms, and do not suffer from the same nuances as the LMS-IIR algorithms. Again, MPSO is capable of achieving the best performance with relatively small population sizes.

The relatively high minimum MSE obtained in Figure 4 is a result of the adaptive filter being a poor model for the plant and has no reflection on the performance of the algorithms. The algorithms are functioning well on the error surfaces, blind to the actual structure of the filter.

REFERENCES

[1] Blackmore, K.L., "Williamson, R.C., Mareels, I.M.Y, and Sethares, W.A., "Online Learning via Congregational Gradient Descent," *Mathematics of Control, Signals*, and Systems 10, pp. 331-363, 1997.

[2] Eberhart, R. C. and Kennedy, J. "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan. pp. 39-43, 1995

[3] Haykin, S., *Adaptive Filter Theory,* 4th ed. Prentice Hall, 2001.

[4] Krusienski, D. J. and Jenkins, W. K., "Adaptive Filtering Via Particle Swarm Optimization", *Proc. of the 37th Asilomar Conf. on Signals, Systems, and Computers*, November 2003.

[5] Krusienski, D. J., "Enhanced Structured Stochastic Global Optimization Algorithms for IIR and Nonlinear Adaptive Filtering," Ph.D. dissertation, Department of Electrical Engineering, The Pennsylvania State University, State College, PA, August 2004.

[6] Krusienski, D. J. and Jenkins, W. K., "The Application of Particle Swarm Optimization to Adaptive IIR Phase Equalization," *Proc. of the 2004 ICASSP,* May 2004.

[7] Mathews, V. J., and Sicuranze, G. L., *Polynomial Signal Processing,* John Wiley & Sons, Inc., 2000.

[8] Nambiar, R. and Mars, P., "Genetic and Annealing Approaches to Adaptive Digital Filtering," *Proc. 26th Asilomar Conf. on Signals, Systems, and Computers*, vol. 2, pp. 871-875, Oct. 1992.

[9] Ng, S.C., Leung, S.H., Chung, C.Y., Luk, A., and Lau, W.H., "The Genetic Search Approach: A New Learning Algorithm for Adaptive IIR Filtering," *IEEE Signal Processing Magazine*, pp. 38-46, November 1996.

[10] Tang, K.S., Man, K.F., and He, Q. "Genetic Algorithms and their Applications," *IEEE Signal Processing Magazine*, pp. 22-37, November 1996.

[11] Yao L., and Sethares, W.A., "Nonlinear Parameter Estimation via the Genetic Algorithm, *IEEE Transactions on Signal Processing*, vol.42, April 1994.