Feature

Design and Performance of Adaptive Systems Based on Structured Stochastic Optimization Strategies

D. J. Krusienski and W. K. Jenkins

WARREN PHOTOGRAPHIC

Abstract

The theory and design of linear adaptive filters based on FIR filter structures is well developed and widely applied in practice. However, the same is not true for more general classes of adaptive systems such as linear infinite impulse response adaptive filters (IIR) and nonlinear adaptive systems. This situation results because both linear IIR structures and nonlinear structures tend to produce multi-modal error surfaces for which stochastic gradient optimization strategies may fail to reach the global minimum. After briefly discussing the state of the art in linear adaptive filtering, the attention of this paper is turned to IIR and nonlinear adaptive systems for potential use in echo cancellation, channel equalization, acoustic channel modeling, nonlinear prediction, and nonlinear system identification . Structured stochastic optimization algorithms that are effective on multimodal error surfaces are then introduced, with particular attention to the Particle Swarm Optimization (PSO) technique. The PSO algorithm is demonstrated on some representative IIR and nonlinear filter structures, and both performance and computational complexity are analyzed for these types of nonlinear systems.

1531-636X/05/$20.00©2005 IEEE

Introduction

Adaptive digital signal processing is a subject that has attracted increasing attention in recent years due to demands for improved performance in high data rate digital communication systems and in wideband image/video processing systems. Adaptive system identification, adaptive noise cancellation, adaptive linear prediction, and adaptive channel equalization are just a few of the important applications areas that have been significantly advanced with adaptive signal processing techniques. Much of the recent success in adaptive signal processing has been facilitated by improvements in VLSI digital signal processor (DSP) integrated circuit technology, which currently provides large amounts of digital signal processing power in a convenient and reliable form. Since improvements in integrated circuit technology continue to emerge it is likely that adaptive techniques will assume an even more important role in high performance electronic systems of the future.

While theory and design of linear adaptive filters based on FIR filter structures is a well developed subject, the same is not true for linear infinite impulse response adaptive filters (IIR) or nonlinear adaptive systems. Both linear IIR structures and nonlinear structures tend to produce multi-modal error surfaces for which stochastic gradient optimization strategies may fail to reach the global minimum. Also, with stochastic gradient algorithms, IIR adaptive filters must be carefully monitored for stability in case their poles move outside the unit circle during adaptation.

Linear FIR Adaptive Filters

An adaptive finite impulse response (FIR) filter consists of a digital tapped delay line with variable multiplier coefficients that are adjusted by an adaptive algorithm [1]. The adaptive algorithm attempts to minimize a cost function that is designed to provide an instantaneous on-line estimate of how closely the adaptive filter achieves a prescribed optimum condition. The cost function most frequently used is an approximation to the expected value of the squared error, $E\{|e|^2(n)\}$, where $e(n) = d(n) - y(n)$ is the difference between a training signal $d(n)$ (sometimes called the desired response) and the filter output $y(n)$, and $E\{\bullet\}$ denotes the statistical expected value. The training signal $d(n)$ is obtained by different means in different applications, and the acquisition of an appropriate training signal is often a limiting factor in performance.

The input vector and the coefficient weight vector of the adaptive filter at the *nth* iteration are defined as $\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ and $\mathbf{W}(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^t$, respectively, where the superscript t denotes vector transpose. The nth output is then given by

$$y(n) = \sum_{k=0}^{N-1} w_k x(n-k) = \mathbf{W^t}(n)\mathbf{X}(n). \tag{1}$$

In the following discussion, the training signal $d(n)$ and the input signal $x(n)$ are assumed to be stationary and ergodic. An adaptive filter uses an iterative method by which the tap weights $\mathbf{W}(n)$ are made to converge to the optimal solution $\mathbf{W}^*$ that minimizes the cost function. It is well known that $\mathbf{W}^*$, known in the literature as the Wiener solution, is given by

$$\mathbf{W}^* = \mathbf{R}_{\mathrm{xx}}^{-1}\,\mathbf{p}_{\mathrm{xd}}, \tag{2}$$

where $\mathbf{R}_{\mathrm{xx}} = E\{\mathbf{X}(n)\mathbf{X}^t(n)\}$ is the autocorrelation matrix of the input and $\mathbf{p}_{xd} = E\{\mathbf{X}(n)d(n)\}$ is the cross-correlation vector between the input and the desired response. The most common iterative approach is to update each tap weight according to a steepest descent strategy; i.e., the tap weight vector is incremented in proportion to the gradient $\nabla\mathbf{w}$ according to

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu\nabla\mathbf{w}, \tag{3}$$

where μ is the step size, $\nabla\mathbf{w} = [\nabla_{w0}(n), \ldots, \nabla_{wN-1}(n)]^t$, and $\nabla_{wk}(n) = \delta E\{e^2(n)\}/\delta w_k$ is the partial derivative of the cost function with respect to $w_k(n)$, for $k = 0, \ldots, N-1$. However, the precise value of the cost function is not known, nor is the gradient known explicitly, so it is necessary to make some simplifying assumptions that allow gradient estimates to be computed on-line. Different approaches to estimating the cost function and/or the gradient lead to different adaptive algorithms, such as the well known Least Mean Squares (LMS) and the Recursive Least Squares (RLS) algorithms. In particular, including the Hessian matrix in equation (3) to accelerate the steepest descent optimization strategy leads to the family of quasi-Newton algorithms that are characterized by rapid convergence at the expense of greater computational complexities.

The LMS algorithm [1] makes the simplifying assumption that the expected value of the squared error is approximated by the squared error itself, i.e., $E\{|e(n)|^2\} \sim |e(n)|^2$. In deriving the algorithm, the error squared is differentiated with respect to $\mathbf{W}$ to approximate the true gradient. In vector notation the LMS update relation becomes

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + 2\mu e(n)\boldsymbol{X}(n). \qquad (4)$$

The value of μ, usually determined experimentally, greatly affects both the convergence rate of the adaptive process and the minimum mean squared error after convergence. To ensure stability and guarantee convergence of both the coefficients and the mean squared error estimate, μ must satisfy the condition $0 < \mu < 1/NE\{x^2(n)\}$, where $E\{x^2(n)\} = (1/N)tr[\mathbf{R}_{xx}]$ is the average input signal power which can be calculated from $\mathbf{R}_{xx}$ if the input autocorrelation matrix is known. When μ is properly chosen, the weight vector converges to an estimate of the Wiener solution.

Linear IIR Adaptive Filters

The mean squared error (MSE) approximation that led to the conventional LMS algorithm for FIR filters has also been applied to infinite impulse response (IIR) filters [2].

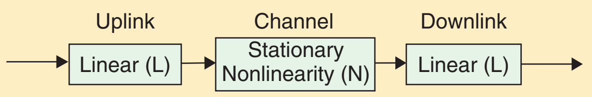

Figure 1. Model of a typical nonlinear communication channel.

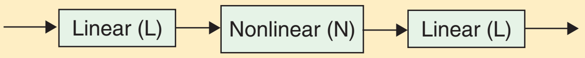

Figure 2. LNL model.

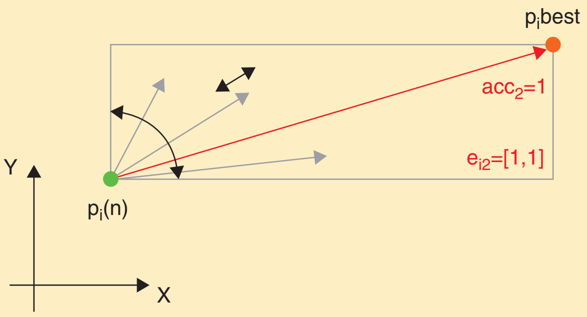

Figure 3. Example of scaling and random direction bounds for the vectors.

Recall that a direct form IIR digital filter is characterized by a difference equation

$$y(n) = \sum_{k=0}^{N_b-1} b_k x(n-k) + \sum_{j=0}^{N_a-1} a_j y(n-j), \qquad (5)$$

where the b_k's are the coefficients that define the zeros of the filter and the a_k's define the poles. The LMS adaptive algorithm for IIR filters is derived in a similar manner as in the FIR case, although the recursive relation (equation (5)) is used instead of the convolution sum (equation (1)) to characterize the input-output relationship of the filter. The IIR derivation is more complicated because the recursive terms on the right side of equation (5) depend on past values of the filter coefficients.

If derivatives of the squared error function are calculated using the chain rule, so that first order dependencies are taken into account and higher order dependencies are ignored, the result is

$$\nabla_{E[|e|^2]} = \left[-2e(n)\frac{\partial(y(n))}{\partial \mathbf{a}}, -2e(n)\frac{\partial(y(n))}{\partial \mathbf{b}} \right],$$

where

$$\frac{\partial y(n)}{\partial b_k} = x(n-k) + \sum_{j=1}^{N_a} a_j(n)\frac{\partial y(n-j)}{\partial b_k} \quad k = 0, \ldots, N_b - 1 \qquad (6a)$$

and

$$\frac{\partial y(n)}{\partial a_k} = y(n-k) + \sum_{j=1}^{N_a} a_j(n)\frac{\partial y(n-j)}{\partial a_k} \quad k = 0, \ldots, N_a - 1. \qquad (6b)$$

This procedure does not result in a closed form expression for the gradient but it does produce a recursive relation by which the gradients can be generated using equation (6).

It is well known that the use of the output error in the formulation of the cost function prevents bias in the solution due to noise in the desired signal. However, the effect of this recursion is to make the problem nonlinear in terms of the coefficient parameters. Also, the current filter parameters now depend directly upon previous filter coefficients, which are time-varying. This often leads to MSE surfaces that are not quadratic in nature. There are many examples in the literature for which IIR MSE surfaces contain local minima, in addition to the global minimum. In these cases gradient search techniques may become entrapped in local minimum, resulting in poor performance due to improper convergence of the adaptive filter [3]–[6].

Nonlinear Adaptive Systems

There are many the applications where voice signals, audio signals, images, or video signals are subjected to nonlinear processing, and which require nonlinear adaptive compensation to achieve the proper system identification and parameter extraction. For example, a generic nonlinear communication system is shown in Figure 1. The overall communication channel between the transmitter and the receiver is often nonlinear, since the amplifiers located in the (satellite) repeaters usually operate at/or near saturation in order to conserve power. The saturation nonlinearities of the amplifiers introduce nonlinear distortions in the signal they process. The path from the transmitter to the repeater as well as from the repeater to the receiver may each be modeled as a linear system. The amplifier characteristics are usually modeled using memoryless nonlinearities. In general, the static nonlinearity is comprised of a linear term and higher order polynomial terms; hence the output of such a system can be represented as the sum of a linear part and a nonlinear part. Such systems can be modeled by connecting nonlinear and linear filter modules into a series cascade configuration. In particular, many nonlinear systems can be represented by the LNL model shown in Figure 2 [7]. It is well known that a similar nonlinear model can be used for magnetic recording channels, where the interaction between the electronic bit stream and the magnetic recording medium via the read/write heads exhibits nonlinear behavior. As in the case of IIR adaptive structures, nonlinear adaptive filters can also produce multi-modal error surfaces on which stochastic gradient optimization strategies may fail to reach the global minimum due to premature entrapment. Neural networks and Volterra nonlinear adaptive filters are well known for their tendency to generate troublesome multimodal error surfaces [8]–[12].

Structured Stochastic Optimization Algorithms

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 based 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 objective functions by assigning the parameter estimates to represent filter tap weights, neural network weights, or any other possible parameter of the unknown system model (even the exponents of polynomial terms) [13].

A brute-force, purely stochastic search would consist of continuously evaluating random, independent parameter estimates relative to a suitable objective/cost/fitness

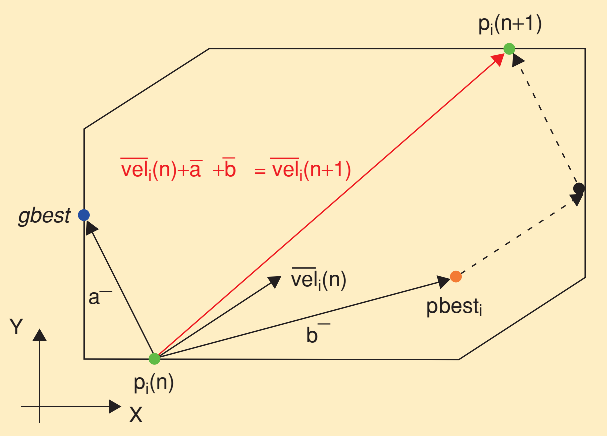
$p_i(n+1)$
$\overline{vel}_i(n)+\bar{a}\ +\bar{b}\ =\overline{vel}_i(n+1)$
gbest
$\bar{a}$
$vel_i(n)$
$pbest_i$
$\bar{b}$
Y
X
$p_i(n)$

Figure 4. Example of the possible search region for a single particle.

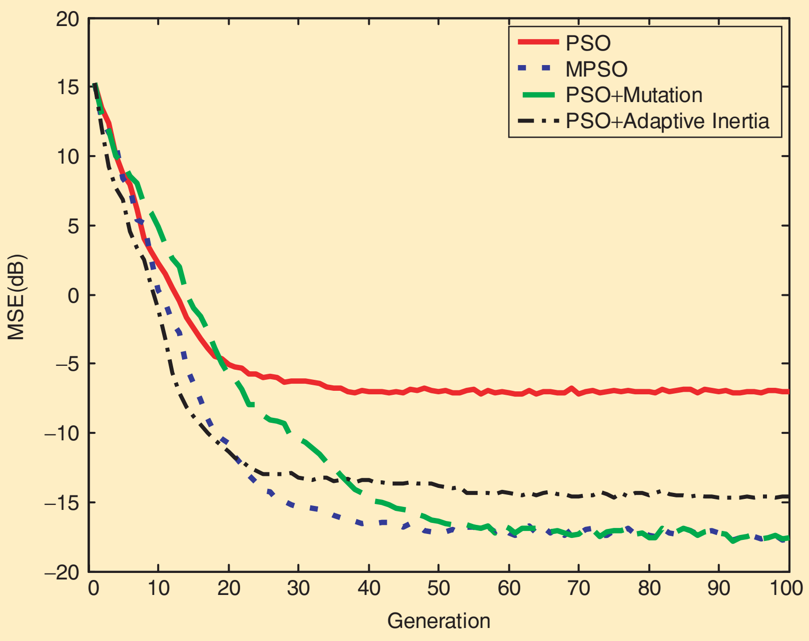

Figure 5. MPSO Enhancements.

function until some minimal error condition is satisfied. A slightly more sophisticated stochastic search would be to evaluate a set of parameter estimates generated by an appropriate random distribution with respect to the parameter space. Although these types of stochastic searches can potentially find the global optimum in some simple cases, they lack any real structure and fail to utilize other information available from the combined search that would enable them to be more efficient.

The foundation of a *structured* stochastic search is to *intelligently* generate and modify the randomized estimates in a manner that efficiently searches the error space, based on some previous or collective information generated by the search [14]. Several different structured stochastic optimization techniques can be found in adaptive filtering literature, most notably simulated annealing [15]–[19], evolutionary algorithms such as the genetic algorithm [20], [18] [21]–[24], and swarm intelligence algorithms such as particle swarm optimization [25]–[37]. One interesting item to note is that all of the prominent structured stochastic techniques are inspired by a natural or biological process. This can be attributed to the observance that such natural processes exhibit a sense of structure and stability achieved through some sort of randomness or chaos. This section reviews the two prominent population based structured stochastic optimization strategies, evolutionary algorithms and particle swarm optimization, due to their superior convergence properties for adaptive filtering applications. Special emphasis is placed on PSO due to its relative novelty.

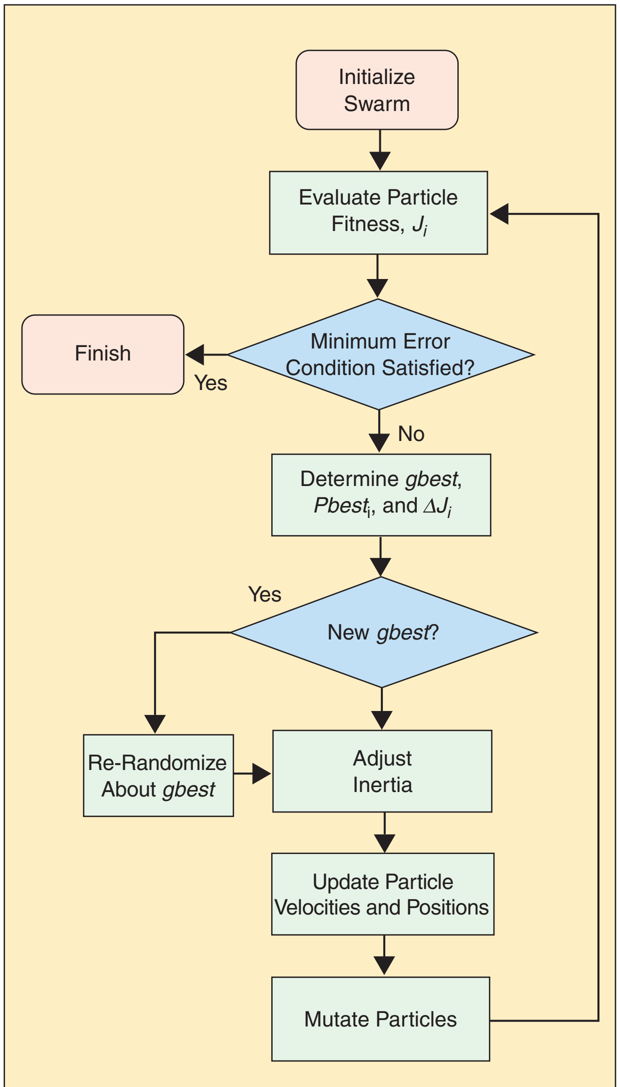
Initialize Swarm
Evaluate Particle Fitness, J_i
Minimum Error Condition Satisfied?
Finish
Yes
No
Determine *gbest*, $Pbest_i$, and ΔJ_i
Yes
New *gbest*?
Re-Randomize About *gbest*
Adjust Inertia
Update Particle Velocities and Positions
Mutate Particles

Figure 6. MPSO Flowchart.

Evolutionary Algorithms

Evolutionary algorithms (EA) [38] begin with a random set of candidate solutions (the unknown parameters to be optimized), termed the population. Each candidate solution in the population is termed an individual. Each individual's set of parameters is termed a chromosome or genome, and each parameter is termed a gene. Depending on the nature of the problem, the chromosomes may represent real numbers or can be encoded as binary strings.

At every generation, the fitness of each individual is evaluated by a predetermined fitness function that is assumed to have an extremum at the desired optimal solution. An individual with a fitness value closer to that of the optimal solution is considered better fit than an individual with a fitness value farther from that of the optimal solution. The population is then evolved based on a set of principles rooted in evolutionary theory such as natural selection, survival of the fittest, and mutation. Natural selection is the mating of the fittest individuals (*parents*) within the population to produce a new individual (*offspring*). This equates to randomly swapping corresponding parameters (*crossover*) between the parents to produce a new, potentially better fit individual. The new offspring then replace the least fit individuals in the population, which is the survival of the fittest facet of the evolution. A portion of the population is then randomly mutated in order to add new parameters to the search. The expectation is that only the offspring that inherit the best parameters from the parents will survive and the population will continually converge to the best possible fitness that represents the optimal or suitable solution. Several EA paradigms exist such as the genetic algorithm, evolutionary programming, and evolutionary strategies; each emphasizing only specific evolutionary constructs, encoding, and operators.

Previous work on EAs for adaptive filtering, specifically the GA, shows that the GA is capable of globally optimizing both IIR [20], [18] [22]–[24] [39] and nonlinear [23], [24] filter structures. A hybrid genetic-LMS approach is used in [39], which utilizes the GA to find suitable initial IIR filter coefficients for the LMS algorithm, resulting in an improved performance over implementing a standard GA or LMS. The performance of the GA is examined for general nonlinear recursive adaptive filters in [24], along with a proof of convergence for the estimation error.

Particle Swarm Optimization

Particle swarm optimization was first developed in 1995 by Eberhart and Kennedy [40]–[45] 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.

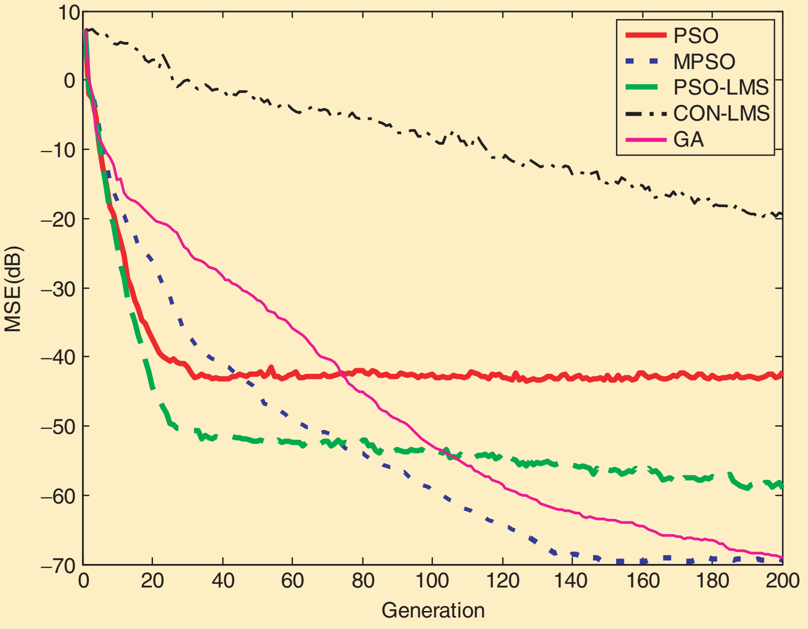

Figure 7. Population = 50. z

Similar to EAs, conventional PSO begins with a random population of individuals; here termed a swarm of particles. As with EAs, each particle in the swarm is a different possible set of the unknown parameters to be optimized. The parameters that characterize each particle can be real-valued or encoded depending on the particular circumstances. The premise is to efficiently search the solution space by swarming the particles toward the best fit solution encountered in previous epochs with the intent of encountering better solutions through the course of the process and eventually converging on a single minimum error solution.

The conventional PSO algorithm begins by initializing a random swarm of M particles, each having R unknown parameters to be optimized. At each epoch, 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 ($\mathrm{p_i}$) in the swarm are updated at each epoch (n) according to the following equations:

$$\begin{aligned}\overline{vel_i}(n) = w\,{}^{*}\overline{vel_i}(n-1) + acc_1\,{}^{*}diag\,[e_1, e_2, \ldots, e_R]_{i1} \\ \times\ {}^{*}(gbest - p_i(n-1)) \\ + acc_2\,{}^{*}diag\,[e_1, e_2, \ldots, e_R]_{i2}\ {}^{*}(pbest_i - p_i(n-1))\end{aligned} \tag{7}$$

$$p_i(n) = p_i(n-1) + \overline{vel_i}(n), \tag{8}$$

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 $gbest$, $pbest_i$ 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 $pbest_i$. Each particle's previous position ($pbest_i$) and the swarm's overall best position ($gbest$) are meant to represent the notion of individual experience memory and group knowledge of a "leader or queen," respectively, that emerges during the natural swarming process. The acceleration constants are typically chosen in the range $\in (0, 2)$ 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, as illustrated for a 2-dimensional case in Figure 3. 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. Setting the acceleration greater than 1 allows the particle to possibly overstep $gbest$ or $pbest$, resulting in a broader search. The random e_i vectors have R different components, which are ran-

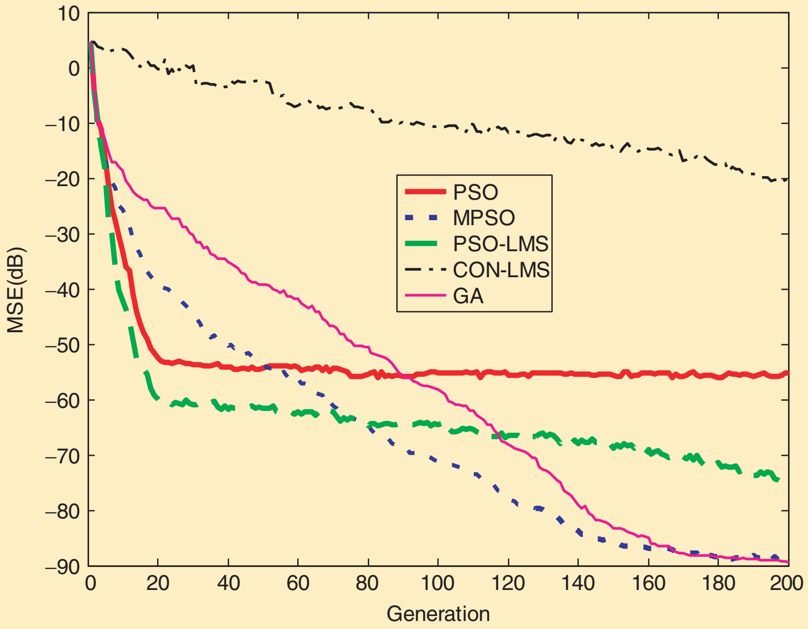

Figure 8. Population = 100.

domly chosen from a uniform distribution in the range ∈ (0, 1). This allows the particle to take constrained randomly directed steps in a bounded region between *gbest* and *pbest$_i$*, as shown in Figure 3.

A single particle update is graphically illustrated in two dimensions in Figure 4. The new particle coordinates can lie anywhere within the bounded region, depending upon the weights and random components associated with each vector. The particle update bounds in Figure 4 are basically composed of all of the bounded regions for each vector as shown in Figure 3, with the addition of the previous velocity component.

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 obtained by the algorithm.

One of the key advantages of PSO is the ease of implementation in both the context of coding and parameter selection. This is much simpler and intuitive to implement than complex, probability based selection and mutation operators required for evolutionary algorithms. Guidelines for selecting and optimizing the PSO parameters are detailed in [46]–[50].

The Particle Swarm–Least Mean Square (PSO-LMS) Hybrid Algorithm

One weakness of conventional PSO is that its local search is not guaranteed convergent; its local search capability lies primarily in the swarm size and search parameters. On the other hand, the problem with simply running a brute-force population of independent LMS algorithms [4] 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 [51] 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 without apprehension 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 be extended for IIR updates and back-propagation updates for nonlinear structures.

For a general adaptive filter structure, the LMS update takes the form:

$$w(n) = w(n-1) + \mu e(n-1)\nabla y(n-1), \qquad (9)$$

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. To form the PSO-LMS hybrid, the LMS update from equation (1) is combined with to the PSO particle update from equation (5) to create the hybrid update:

$$p_i(n) = p_i(n-1) + c_1\overline{vel_i}(n) + c_2\mu e(n-1)\nabla y(n-1), \quad (10)$$

where $c_1(n)$ and $c_2(n)$ are scaling factors that control the relative influence of the PSO and LMS directions, respectively. These scaling factors should be chosen such that $c_1(n) + c_2(n) = 1$ 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. In addition to the locally convergent properties, another advantage of this algorithm is that the LMS component enables tracking of a dynamic system.

Modified Particle Swarm Optimization

Although PSO-LMS has several desirable traits, the LMS component of the update can grow complex and tend to slow the convergence of certain IIR and nonlinear structures. Therefore, several simple enhancements can be added to conventional PSO that significantly boost the performance without much added complexity. The following enhancements address the two main weaknesses of conventional PSO: outlying particles and stagnation.

If the new *gbest* particle is an outlying particle with respect to the swarm, the rest of the swarm can tend to move toward the new *gbest* from the same general direction. This may potentially leave some critical region around the new minimum excluded from the search. To combat this problem, when a new *gbest* is encountered, a small percentage of randomly selected particles can be re-randomized about the new *gbest*. This does not affect the global search properties, and can actually improve the convergence rate.

Particles closer to *gbest* will tend to quickly converge on it and become stagnant (no longer update) while the other more distant particles continue to search. A stagnant particle is essentially useless because its fitness continues to be evaluated but it no longer contributes to the search. Stagnancy of particles can be eliminated by slightly varying the random parameters of each particle at every epoch, similar to mutation in EAs. This has little effect on particles distant from *gbest* because this random influence is relatively small compared to the

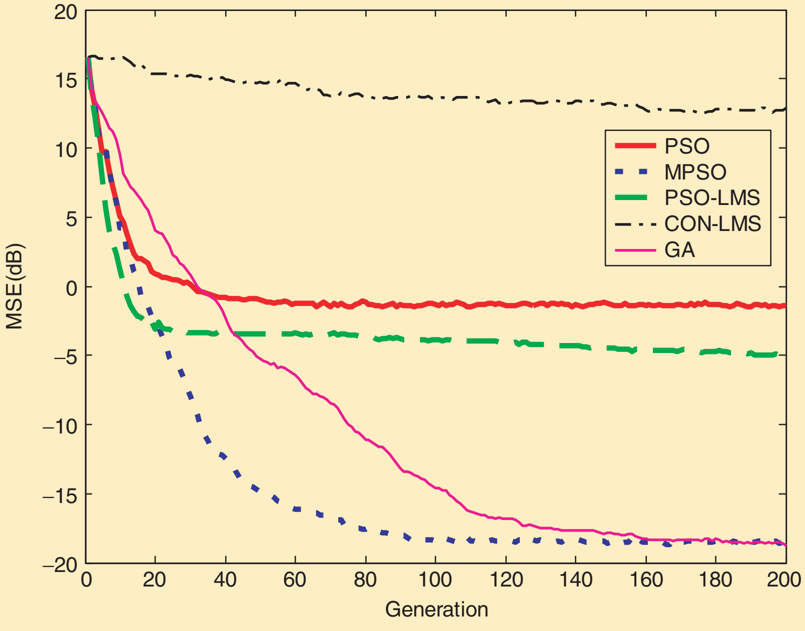

Figure 9. Population = 20.

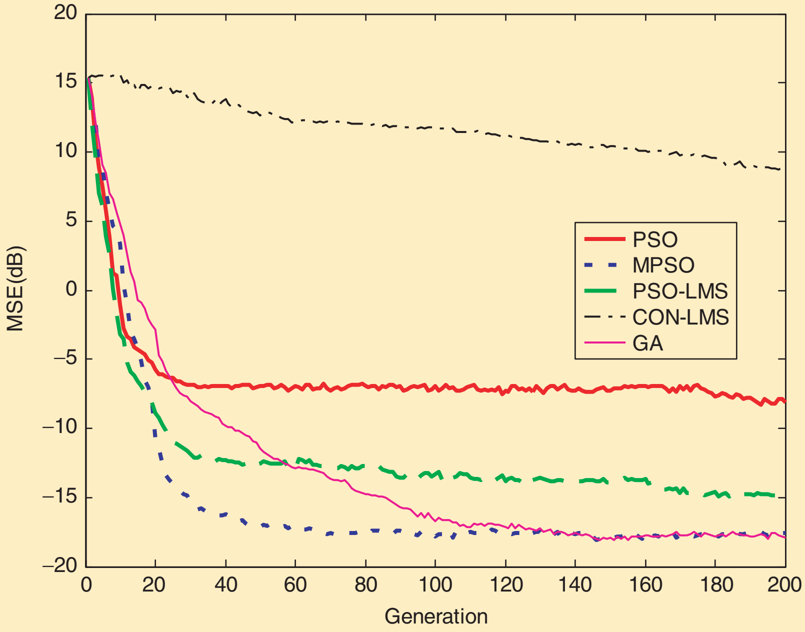

Figure 10. Population = 40.

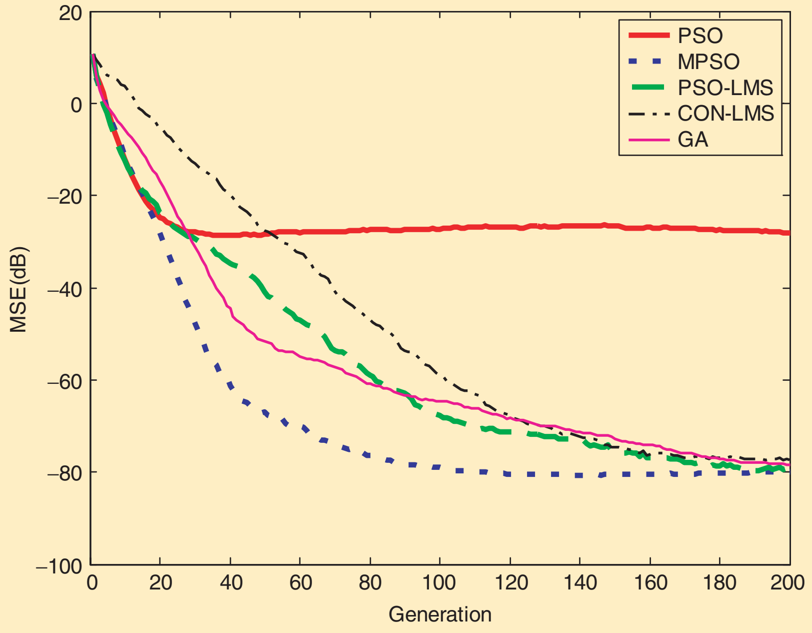

Figure 11. Population = 20.

random update of equation (2). However, this eliminates any stagnant particles by forcing a finer search about *gbest*.

Because the mutation operator tends to slow the optimal convergence rate of PSO in general, the following adaptive inertia operator is included to compensate:

$$w_i(n) = \frac{1}{\left(1 + e^{\frac{-\Delta J_i(n)}{S}}\right)}$$

where $w_i(n)$ is the inertia weight of the ith particle, $\Delta J_i(n)$ is the change in particle fitness between the current and last generation, and S is a constant used to adjust the transition slope based on the expected fitness range. The adaptive inertia automatically adjusts to favor directions that result in large increases in the fitness value, while suppressing directions that decrease the fitness value. This modification does not prevent the hill-climbing capabilities of PSO, it merely increases the influence of potentially fruitful inertial directions, while decreasing the influence of potentially unfavorable inertial directions.

The relative effects of the suggested enhancements are illustrated for a simple IIR system identification example in Figure 5. A flow chart for such an algorithm, referred to here as MPSO, is given in Figure 6. This algorithm is capable of providing desirable performance and convergence properties in most any context. In addition to eliminating the concerns of conventional PSO, the algorithm is designed to balance the convergence speed and search quality tradeoffs of a stochastic search. At a minimum, a more compact form of MPSO, using only the additional mutation operator, will provide a significant increase in performance of conventional PSO. When computational complexity is an issue, another major advantage of these algorithms is that they are able to achieve robust results with smaller populations; since the computational complexity is directly related to the population size for population based algorithms.

Experimental Examples

In the following examples, the properties of the aforementioned algorithms several IIR and nonlinear system identification problems, using a windowed mean squared error between the desired training signal and the adaptive filter output as the fitness function. All algorithms were initialized with the same population of real-valued parameters and allowed to evolve. The population sizes were selected to provide reasonable

Table 1.

Algorithm	Description
PSO	The conventional PSO algorithm.
MPSO	The modified PSO algorithm, with matching PSO parameters.
PSO-LMS	The PSO-LMS algorithm, with matching PSO parameters. A smooth transition between algorithms was selected to occur at the point that conventional PSO stagnates.
CON-LMS	The congregational LMS algorithm [52] is implemented where a population of independent LMS estimates is updated at each epoch.
GA	The genetic algorithm uses a ranked elitist strategy suggested in [20], where the 6 fittest members of the population are used to generate offspring, which replace the remaining least fit members of the population. For each offspring, two of the 6 parents are selected randomly and the crossover is performed by a random weighted average of each parent's coefficients. This scheme was chosen due to its accelerated convergence properties.

performance, while revealing the performance characteristics of each algorithm. The fitness is defined as the MSE error of the candidate solutions evaluated over a causal window of 100 input samples. Unless specified otherwise, the input signals are zero-mean Gaussian white noise with unit variance. For each simulation, the MSE is averaged over 50 independent Monte Carlo trials. The specifics of each algorithm are given in Table 1.

Matched high-order IIR Filter, White noise input

For this example the plant, given below is a fifth-order low-pass Butterworth filter example taken from [20]:

$$H_{PLANT}(z^{-1}) = \frac{0.1084 + 0.5419z^{-1} + 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}}{1 + 0.9853z^{-1} + 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}} \tag{11}$$

$$H_{AF}(z^{-1}) = \frac{p_i^1 + p_i^2 z^{-1} + p_i^3 z^{-2} + p_i^4 z^{-3} + p_i^5 z^{-4} + p_i^6 z^{-5}}{1 + p_i^7 z^{-1} + p_i^8 z^{-2} + p_i^9 z^{-3} + p_i^{10} z^{-4} + p_i^{11} z^{-5}}. \tag{12}$$

The learning curves for this example are given in Figures 7 and 8. The simulations having the larger population illustrate the case where the population size is sufficient and all of the algorithms rarely become trapped in a local minimum.

Table 2 lists the local minima statistics after swarm convergence for the decreased population IIR examples. One interesting observance is that population based searches are inherently less likely to produce unstable IIR filters due to the number of estimates available at any given epoch, which is not the case for sequential searches such as LMS.

Table 2.

Algorithm	% Trials in Local Minimum
MPSO	2
GA	6
PSO-LMS	6
CON-LMS	15
PSO	20

Reduced order IIR Filter, Colored noise input

For this example, the plant given below is an example taken from [20] and [53]:

$$H_{PLANT}(z^{-1}) = \frac{1}{(1 - 0.6z^{-1})^3} \tag{13}$$

$$H_{AF}(z^{-1}) = \frac{p_i^1}{1 + p_i^2 z^{-1} + p_i^3 z^{-2}} \tag{14}$$

$$H_{COLOR}(z^{-1}) = (1 - 0.6z^{-1})^2(1 + 0.6z^{-1})^2. \tag{15}$$

The adaptive filters use a colored input generated by filtering white noise by the FIR filter given in equation (15). This, in combination with the reduced order, creates a bimodal error surface. Again, the colored noise leads to smaller stable step sizes and slowed convergence rates. The learning curves for this example are given in Figures 9 and 10.

Matched Structure Volterra Identification

In this example, a matched structure truncated Volterra adaptive filter is used to identify the truncated Volterra plant taken from [54]:

$$y[n] = -0.64x[n] + x[n-2] + 0.9x^2[n] + x^2[n-1] \tag{16}$$

$$\begin{aligned} \hat{y}[n] = {} & p_i^1 x[n] + p_i^2 x[n-1] + p_i^3 x[n-2] \\ & + p_i^4 x^2[n] + p_i^5 x^2[n-1] + p_i^6 x^2[n-2] \\ & + p_i^7 x[n]x[n-1] + p_i^8 x[n]x[n-2] \\ & + p_i^9 x[n-1]x[n-2]. \end{aligned} \tag{17}$$

The learning curves for this example are given in Figure 11.

Nonlinear LNL Cascade Identification

In this example, the identification of an LNL cascade taken from [54] is performed using a unmatched LNL adaptive filter. The LNL plant consists of a 4th order Butterworth lowpass filter (equation 2), followed by a 4th power memoryless nonlinear operator, followed by a 4th order Chebyshev lowpass filter (equation 3), as shown in Figure 12.

This system is a common model for 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.

$$\hat{H}_B(z^{-1}) = \frac{(0.2851 + 0.5704z^{-1} + 0.2851z^{-2})(0.2851 + 0.5701z^{-1} + 0.2851z^{-2})}{(1 - 0.1024z^{-1} + 0.4475z^{-2})(1 - 0.0736z^{-1} + 0.0408z^{-2})} \tag{18}$$

$$\hat{H}_C(z^{-1}) = \frac{(0.2025 + 0.288z^{-1} + 0.2025z^{-2})(0.2025 + 0.0034z^{-1} + 0.2025z^{-2})}{(1 - 1.01z^{-1} + 0.5861z^{-2})(1 - 0.6591z^{-1} + 0.1498z^{-2})} \tag{19}$$

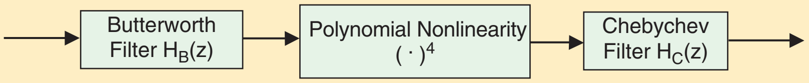
Butterworth Filter $H_B(z)$
Polynomial Nonlinearity $(\cdot)^4$
Chebychev Filter $H_C(z)$

Figure 12. Nonlinear LNL unknown system.

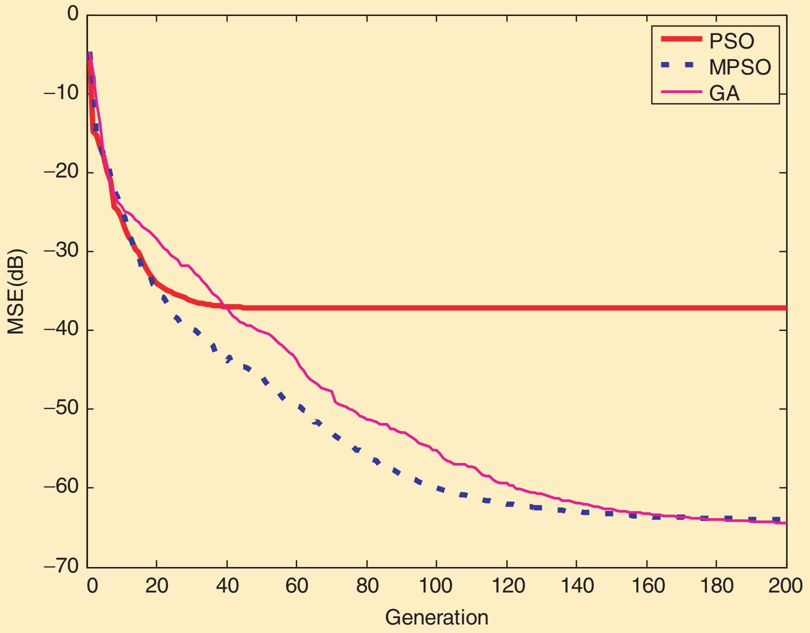

Figure 13. Population = 50.

The LNL adaptive filter structure is given as follows:

$$\hat{H}_B(z^{-1}) = \frac{p_i^1 + p_i^2 z^{-1} + p_i^3 z^{-2} + p_i^4 z^{-3}}{1 + p_i^5 z^{-1} + p_i^6 z^{-2} + p_i^7 z^{-3}} \tag{20}$$

$$\text{nonlinearity} = p_i^8 \left[\hat{H}_B(z^{-1})\right]^4 \tag{21}$$

$$\hat{H}_C(z^{-1}) = \frac{p_i^9 + p_i^{10} z^{-1} + p_i^{11} z^{-2} + p_i^{12} z^{-3}}{1 + p_i^{13} z^{-1} + p_i^{14} z^{-2} + p_i^{15} z^{-3}}. \tag{22}$$

The learning curves for this example are given in Figure 13.

Summary and Conclusions

The results presented demonstrate that the structured stochastic algorithms are capable of quickly and effectively adapting the coefficients of a wide variety of IIR and nonlinear structures. From the simulation results, it is observed that, with a sufficient population size, all of the structured stochastic algorithms are capable of converging rapidly to below −20 dB in most instances, which is an order of magnitude faster than most existing gradient based techniques.

In all cases, the congregational LMS algorithm exhibits the slowest convergence rate due to the fact that there is no information transfer between the estimates. It is observed from the learning curves that conventional PSO stagnates and PSO-LMS continues to improve from the stagnation point of conventional PSO at a rate similar to CON-LMS. 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.

Since the GA does not have an explicit step size, the convergence rate can only be controlled to a limited extent through the crossover and mutation operations, and the algorithm must evolve at its own intrinsic rate. Due to the nature of the algorithm, these GA operators become increasingly taxed as the population decreases, resulting in depreciating performance. On the other hand, as the population size increases, the performance gap between the GA and MPSO begins to diminish for large parameter spaces.

Though conventional PSO exhibits a fast convergence initially, it fails to improve further because the swarm quickly becomes stagnant, converging to a suboptimal solution in all instances. However, with the same set of algorithm parameters, the MPSO particles do not stagnate, allowing it to reach the noise floor. Smaller acceleration coefficients can be used with conventional PSO to allow it to approach the noise floor, forsaking the rapid convergence rate. With the introduction of a simple mutation type operator, adaptive inertia weights, and re-randomization, MPSO can retain the favorable convergence rate with a smaller population, while still achieving the noise floor and avoiding local minima. This can offer considerable savings in cases where computational complexity is an issue.

A topic that warrants further investigation is an assessment of the performance of these algorithms on modified error surfaces. It is evident that structured stochastic search algorithms perform better on surfaces exhibiting relatively few local minima, because the local attractors offer little interference to the search. By incorporating alternate formulations of the adaptive filter structures or performing data preprocessing, such as input orthognalization and power normalization, the error surface can be smoothed dramatically. These additions will likely result in improved performance of stochastic search algorithms.

Related to this notion is the assertion that structured stochastic algorithms, particularly versions of PSO, have a tendency to excel on lower order parameter spaces. This is due partly to the fact that lower dimensional parameter spaces tend to exhibit fewer local minima in general, and because the volume of the hyperspace increases exponentially with each additional parameter to be estimated. Therefore it is hypothesized that dimensionality reduction techniques, such as implementing alternate formulations of adaptive filter structures that are able to accurately model unknown systems with minimum number of parameters, could benefit the overall performance.

References

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

[2] J. J. Shynk, "Adaptive IIR Filtering," *IEEE ASSP Magazine*, pp. 4–21, April 1989.

[3] H. Fan, "New adaptive IIR filtering algorithms," Ph.D. Thesis, Dept. of Electrical and Computer Eng., University of Illinois, Urbana Champaign, 1985.

[4] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits and Systems*, vol. CAS-33, no. 10, pp. 939–947, Oct. 1986.

[5] S. Pai, "Global Optimization Techniques for IIR Phase Equalizers", M. S. Thesis, The Pennsylvania State University, University Park, PA, 2001.

[6] S. Pai, W. K. Jenkins, and D. J. Krusienski, "Adaptive IIR phase equalizers based on stochastic search algorithms," *Proc. 37th Asilomar Conf. Circuits, Systems, and Computers*, Nov. 2003.

[7] A. E. Nordsjo and L. H. Zetterberg, "Identification of certain time-varying nonlinear Wiener and Hammerstein systems," *IEEE Trans. on Sig. Proc.*, vol. 49, no. 3, pp. 577–592, March 2001.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1999.

[9] A. Ismail and A. P. Engelbrecht, "Training product units in feedforward neural networks using particle swarm optimization," *Proceedings of the International Conference on Artificial Intelligence*, Durban, South Africa. pp. 36–40, 1999.

[10] A. Ismail and A. P. Engelbrecht, "Global optimization algorithms for training product unit neural networks," *Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks* (IJCNN 2000), pp. 132–137, 2000.

[11] R. Mendes, P. Cortez, M. Rocha, and J. Neves, "Particle swarms for feedforward neural network training," in *Proc. Int. Joint Conf. Neural Networks* (IJCNN '02), vol. 2, pp. 1895–1899, 2002.

[12] F. van den Bergh and A. P. Engelbrecht, "Training product unit networks using cooperative particle swarm optimizers," *Proceedings of INNS-IEEE International Joint Conference on Neural Networks 2001*, Washington DC, USA. 2001.

[13] K. Y. Lee, and M. A. El-Sharkawi, ed., *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, IEEE Power Engineering Society, 2002.

[14] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons Ltd, 2002.

[15] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, 1989.

[16] N. Benvenuto, M. Marchesi, G. Orlandi, F. Piazza, and A. Uncini, "Finite wordlength digital filter design using an annealing algorithm," *Acoustics, Speech, and Signal Processing*, ICASSP-89. vol. 2, pp. 861–864, 1989.

[17] S. Kirkpatrick, C. D. Gelatt Jr., and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 498–516, 1983.

[18] R. Nambiar and P. Mars, "Genetic and annealing approaches to adaptive digital filtering," *Proc. 26th Asilomar Conf. on Signals, Systems, and Computers*, vol. 2, pp. 871–875, Oct. 1992.

[19] J. Radecki, J. Konrad, and E. Dubois, "Design of multidimensional finite-wordlength FIR and IIR filters by simulated annealing," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 6, June 1995.

[20] P. Mars, J. R. Chen, and R. Nambiar, *Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communications*, CRC Press, Inc., 1996.

[21] A. Neubauer, "Genetic algorithms for adaptive non-linear predictors," *1998 IEEE International Conference on Electronics, Circuits and Systems, Volume: 1*, vol. 1, pp. 209–212, 7–10 Sept. 1998.

[22] A. Neubauer, "Non-linear adaptive filters based on genetic algorithms with applications to digital signal processing," *Genetic Algorithms In Engineering Systems: Innovations And Applications*, 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446), pp. 180–185, 2–4 Sept. 1997.

[23] K. S. Tang, K. F. Man, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, pp. 22–37, Nov. 1996.

[24] L. Yao and W. A. Sethares, "Nonlinear parameter estimation via the genetic algorithm," *IEEE Transactions on Signal Processing*, vol. 42, April 1994.

[25] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences," *Evolutionary Programming VII: Proc. of the Seventh Annual Conference on Evolutionary Programming*, 1998.

[26] A. Carlisle and G. Dozier, "Adapting particle swarm optimization to dynamic environments," *Proceedings of International Conference on Artificial Intelligence* (ICAI 2000), Las Vegas, Nevada, USA. pp. 429–434, 2000.

[27] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC 1999), pp. 1951–1957, 1999.

[28] R. C. Eberhart and Y. Shi, "Evolving artificial neural networks," *Proceedings of International Conference on Neural Networks and Brain*, 1998, Beijing, P. R. China. pp. PL5–PL13, 1998.

[29] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, San Diego, CA. 1998.

[30] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, vol. 1, pp. 84–88, 2000.

[31] R. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Congr. Evolutionary Computation* (CEC 01), vol. 1, pp. 81–86, 2001.

[32] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC 2002), Honolulu, Hawaii USA. 2002.

[33] D. J. Krusienski, "Enhanced structured stochastic global optimization algorithms for IIR and nonlinear adaptive filtering," Ph.D. Thesis, Dept. of Electrical Eng., The Pennsylvania State University, University Park, PA, 2004.

[34] D. J. Krusienski and W. K. Jenkins, "A particle swarm optimization-LMS hybrid algorithm for adaptive filtering," *Proc. of the 38th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2004.

[35] D. J. Krusienski and W. K. Jenkins, "Particle swarm optimization for adaptive IIR filter structures," *Proc. of the 2004 Congress on Evolutionary Computation*, pp. 965–970, June 2004.

[36] D. J. Krusienski and W. K. Jenkins, "The application of particle swarm optimization to adaptive IIR phase equalization," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, pp. II-693–II-696, 17–21 May 2004.

[37] D. J. Krusienski and W. K. Jenkins, "Adaptive filtering via particle

swarm optimization," *Proc. of the 37th Asilomar Conf. on Signals, Systems, and Computers*, pp. 571–575, Nov. 2003.

[38] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[39] 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 Magazine*, pp. 38–46, Nov. 1996.

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

[41] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ. pp. 1942–1948, 1995.

[42] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108, 1997.

[43] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence,* San Francisco: Morgan Kaufmann Publishers, 2001.

[44] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. Congr. Evolutionary Computation* (CEC 02), vol. 2, pp. 1671–1676, 2002.

[45] J. Kennedy and J. M. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," in *Proc. IEEE World Congr. Computational Intelligence Evolutionary Computation*, pp. 78–83, 1998.

[46] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 58–73, Feb. 2002.

[47] A. I. El-Gallad, M.E. El-Hawary, A. A. Sallam, and A. Kalas, "Enhancing the particle swarm optimizer via proper parameters selection," *Canadian Conference on Electrical and Computer Engineering*, 2002, pp. 792–797, 2002.

[48] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Computational Intelligence Evolutionary Computation*, pp. 69–73, 1998.

[49] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evolutionary Computation* (CEC 99), vol. 3, pp. 1945–1950, 1999.

[50] F. van den Bergh and A. P. Engelbrecht, "Effects of swarm size on cooperative particle swarm optimizers," *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, San Francisco, USA, 2001.

[51] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2002.

[52] K. L. Blackmore, R.C. Williamson, I. M. Y. Mareels, and W. A. Sethares, "Online learning via congregational gradient descent," *Mathematics of Control, Signals, and Systems* 10, pp. 331–363, 1997.

[53] M. S. White and S. J. Flockton, Chapter in Evolutionary Algorithms in Engineering Applications, Editors: D. Dasgupta and Z. Michalewicz, Springer Verlag, 1997.

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

Dean J. Krusienski (S'01–M'04) received the B.S.E.E., M.S.E.E., and Ph.D.E.E. degrees from The Pennsylvania State University, University Park, PA, in 1999, 2001, and 2004 respectively.

From 2000–2004, he was a graduate teaching assistant at Penn State. He is currently a postdoctoral researcher at the New York State Department of Health's Wadsworth Center in Albany, NY. At The Wadsworth Center, he is researching the application of adaptive signal processing and pattern recognition techniques to a non-invasive brain-computer interface that allows individuals with severe motor disabilities to communicate or control a computer/device via mental imagery. His current research interests include global optimization algorithms for adaptive filtering, computational intelligence, neural networks, blind source separation, and biomedical signal processing.

Dr. Krusienski is a member of Eta Kappa Nu and the IEEE.

W. Kenneth Jenkins received the B.S.E.E. degree from Lehigh University and the M.S.E.E. and Ph.D. degrees from Purdue University. From 1974 to 1977 he was a Research Scientist Associate in the Communication Sciences Laboratory at the Lockheed Research Laboratory, Palo Alto, CA. In 1977 he joined the University of Illinois at Urbana-Champaign where he was a faculty member in Electrical and Computer Engineering from 1977 until 1999. From 1986-1999 Dr. Jenkins was the Director of the Coordinated Science Laboratory and Principal Investigator on the Joint Services Electronics Program (JSEP) at the University of Illinois. In 1999 he moved to his current position as Professor and Head of the Electrical Engineering Department at The Pennsylvania State University in State College, PA.

Dr. Jenkins' current research interests include digital filtering, signal processing algorithms, multidimensional array processing, computer imaging, one and two dimensional adaptive digital filtering, and biologically inspired optimization algorithms for signal processing. He co-authored a book entitled *Advanced Concepts in Adaptive Signal Processing*, published by Kluwer in 1996. He is a past Associate Editor for the *IEEE Transaction on Circuits and Systems*, and President (1985) of the CAS Society. He served as General Chairman of the 1988 Midwest Symposium on Circuits and Systems, as the Program Chairman of the 1990 IEEE International Symposium on Circuits and Systems, as the General Chairman of the Thirty Second Annual Asilomar Conference on Signals and Systems, and as the Technical Program Chair for the 2000 Midwest Symposium on Circuits and Systems. He is currently serving as President of the Electrical and Computer Engineering Department Heads Association (ECEDHA).

Dr. Jenkins is a Fellow of the IEEE and a recipient of the 1990 Distinguished Service Award of the IEEE Circuits and Systems Society. In 2000 he received a Golden Jubilee Medal from the IEEE Circuits and Systems Society and a 2000 Millennium Award from the IEEE. In 2000 was named a co-winner of the 2000 International Award of the George Montefiore Foundation (Belgium) for outstanding career contributions to the field of electrical engineering and electrical science, and in 2002 he was awarded the Shaler Area High School Distinguished Alumnus Award.