EURASIP Journal on Applied Signal Processing 2003:8, 834–840

A Comparison of Evolutionary Algorithms for Tracking Time-Varying Recursive Systems

Michael S. White
Royal Holloway, University of London, Egham Hill, Egham, Surrey, TW20 0EX, UK
Email: mike@whitem.com

Stuart J. Flockton
Royal Holloway, University of London, Egham Hill, Egham, Surrey, TW20 0EX, UK
Email: s.flockton@rhul.ac.uk

Received 28 June 2002 and in revised form 29 November 2002

A comparison is made of the behaviour of some evolutionary algorithms in time-varying adaptive recursive filter systems. Simulations show that an algorithm including random immigrants outperforms a more conventional algorithm using the breeder genetic algorithm as the mutation operator when the time variation is discontinuous, but neither algorithms performs well when the time variation is rapid but smooth. To meet this deficit, a new hybrid algorithm which uses a hill climber as an additional genetic operator, applied for several steps at each generation, is introduced. A comparison is made of the effect of applying the hill climbing operator a few times to all members of the population or a larger number of times solely to the best individual; it is found that applying to the whole population yields the better results, substantially improved compared with those obtained using earlier methods.

Keywords and phrases: recursive filters, evolutionary algorithms, tracking.

1. INTRODUCTION

Many problems in signal processing may be viewed as system identification. A block diagram of a typical system identification configuration is shown in Figure 1. The information available to the user is typically the input and the noise-corrupted output signals, $x(n)$ and $a(n)$, respectively, and the aim is to identify the properties of the "unknown system" by, for example, putting an adaptive filter of a suitable structure in parallel to the unknown system and altering the parameters of this filter to minimise the error signal $\epsilon(n)$. When the nature of the unknown system requires pole-zero modelling, there is a difficulty in adjusting the parameters of the adaptive filter, as the mean square error (MSE) is a nonquadratic function of the recursive filter coefficients, so the error surface of such a filter may have local minima as well as the global minimum that is being sought. The ability of evolutionary algorithms (EAs) to find global minima of multimodal functions has led to their application in this area [1, 2, 3, 4].

All these authors have considered only time-invariant unknown systems. However in many real-life applications, time variations are an ever-present feature. In noise or echo cancellation, for example, the unknown system represents the path between the primary and reference microphones. Movements inside or outside of the recording environment cause the characteristics of this filter to change with time. The system to be identified in an HF transmission system corresponds to the varying propagation path through the atmosphere. Hence there is an interest in investigating the applicability of evolutionary-based adaptive system identification algorithms to tracking time-varying recursive systems. Previous work on the use of EAs in time-varying systems has been published in [5, 6, 7, 8, 9] but none of these deal with system identification of recursive systems. After explaining our choice of filter structure in Section 3, we go on in Section 4 to compare the performance of the EA introduced in [4] with that of the algorithm in [7]. We show that while both can cope reasonably well with slow variations in the system parameters, the approach of [7] is more successful in the case of discontinuous changes, but neither copes well where the variation is smooth but fairly rapid (the distinction between slow and rapid variation is explained quantitatively in Section 3.1). In Section 5, we propose a new hybrid algorithm which embeds what is in effect a hill-climbing operator within the EA and show that this new algorithm is much more successful for the difficult problem of tracking rapid variations.

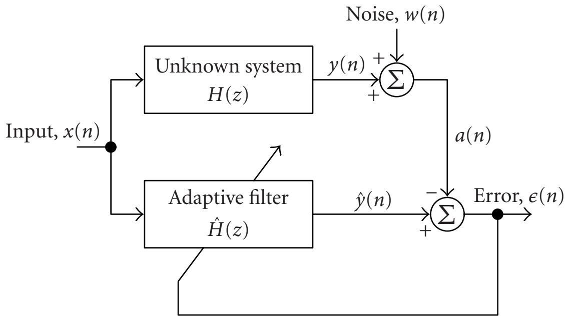

FIGURE 1: System identification.

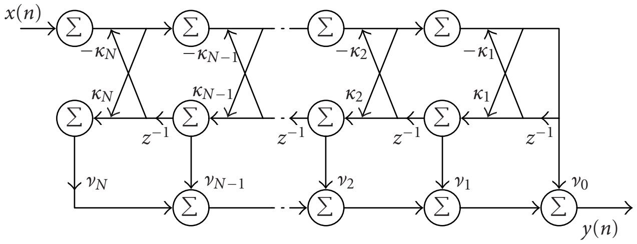

FIGURE 2: Pole-zero lattice filter.

2. GENETIC ALGORITHMS IN CHANGING ENVIRONMENTS

The standard genetic algorithm (GA), with its strong selection policy and low rate of mutation, quickly eliminates diversity from the population as it proceeds. In typical function optimization applications, where the "environment" remains static, we are not usually concerned with the population diversity at later stages of the search, so long as the best or mean value of the population fitness is somewhere near to an acceptable value. However, when the function to be optimized is nonstationary, the standard GA runs into considerable problems once the population has substantially converged on a particular region of the search space. At this point, the GA is effectively reliant on the small number of random mutations, occurring each generation, to somehow redirect its search to regions of higher fitness since standard crossover operators are ineffective when the population has become largely homogeneous. This view is borne out by Pettit's and Swigger's study [10] in which a Holland-type GA was compared to cognitive (statistical predictive) and random point-mutation models in a stochastically fluctuating environment. In all cases, the GA performed poorly in tracking the changing environment even when the rate of fluctuation was slow. An approach to providing EAs capable of functioning well in time-varying systems is the mutation-based strategy adopted by Cobb and Grefenstette [5, 6, 7]. In this approach, population diversity is sustained either by replacing a proportion of the standard GA's population with randomly generated individuals, the *random immigrants* strategy, or by increasing the mutation rate when the performance of the GA degrades (*triggered hypermutation*). Cobb's hypermutation operator is adaptive, briefly increasing the mutation rate when it detects that a degradation of performance (measured as a running average of the best performing population members over five generations) has occurred. However, it is easy to contrive categories of environmental change which would not trigger the hypermutable state. On continuously changing functions, the hypermutation GA has a greater variance in its tracking performance than either the standard or random immigrants GA. In oscillating environments, where the changes are more drastic, the high mutation level of the hypermutation GA destroys much of the information contained in the current population. Consequently, when the environment returns to its prior state, the GA has to locate the previous optimum from scratch.

3. CHOICE OF RECURSIVE FILTER STRUCTURE

One of the main difficulties encountered in recursive adaptive systems is the fact that the system can become unstable if the coefficients are unconstrained. With many filter structures, it is not immediately obvious whether any particular set of coefficients will result in the presence of a pole outside the unit circle, and hence instability. On the other hand, it is important that the adaptive algorithm is able to cover the entire stable coefficient space, so it is desirable to adopt a structure which will make this possible at the same time as making stability monitoring easy. It is for this reason that the pole-zero lattice filter [11] was adopted for this work. A block diagram of the filter structure is given in Figure 2.

The input-output relation of the filter is given by

$$y(n) = \sum_{i=0}^{N} \nu_i(n) B_i(n), \tag{1}$$

where $F_i(n)$ and $B_i(n)$ are the forward and backward residuals denoted by

$$\begin{aligned} B_i(n) &= B_{i-1}(n) + \kappa_i(n) F_i(n), && i = 1, 2, \dots, N, \\ F_i(n) &= F_{i+1}(n) - \kappa_{i+1}(n) B_i(n-1), && i = N, \dots, 1, \\ F_N(n) &= x(N). \end{aligned} \tag{2}$$

It can be shown that a necessary and sufficient condition for all of the roots of the pole polynomial to lie within the unit circle is $|k_i| < 1$, $i = 1, \dots, N$, so the stability of candidate models can be guaranteed merely by restricting the range over which the feedback coefficients are allowed to vary. Since this must be done when implementing the GA anyway, the ability to maintain filter stability is essentially obtained without cost.

3.1. Quantifying time variations in the system being tracked

Work on the tracking performance of LMS, detailed in [12], employs the concept of the *nonstationarity degree* to embody the notions of both the size and speed of time variations. The nonstationarity degree $d(n)$ is defined as

$$d(n) = \sqrt{\frac{E\left(|t(n)|^2\right)}{\sigma_{\min}(n)}}, \tag{3}$$

where $t(n)$ is the output noise caused by the time variations in the unknown system and $\sigma_{\min}(n)$ is the output noise power in the absence of time variations in the system.

Having devised a metric incorporating both the speed and size of time variations, Macchi [12] goes on to describe three distinct classes of nonstationarity. *Slow* variations are those in which the nonstationarity degree is much less than one, that is, the variation noise is masked by the measurement noise. For the LMS adaptive filter, slow changes to the plant impulse response are seen to be easy to track since the time variations need not be estimated very accurately. This class of time variations is further subdivided into two groups in which the "unknown" filter coefficients undergo deterministic or random evolution patterns. *Rapid* variations ($d(n)$ permanently greater than one), however, present a much greater problem to LMS and LS adaptive filters. In the case of time-varying line enhancement at low signal-to-noise ratio, where the frequency of the sinusoidal input signal is "chirped," Macchi et al. state that "... slow adaptation/slow variation condition implies an upper limit for the chirp rate ψ. This limit is the level above which the misadjustment is larger than the original additive noise. The noisy signal is thus a better estimate of the sinusoid than the adaptive system output. The "slow adaptation" condition is therefore required, in practice, to implement the adaptive system" [13, page 360].

In the case of LMS adaptive and inverse adaptive modelling, "adaptive filters cannot track time variations which are so rapid that $d(n)$ is permanently greater than one. Indeed within a single iteration, the algorithm cannot acquire the new optimal filter $\tilde{H}(n+1)$, starting from $\tilde{H}(n)$" [12, page 298].

As a consequence, only a special subset of rapid time variations is generally considered in the context of LMS filter adaptation. The *jump* class of nonstationarity produces scarce large changes in the unknown filter impulse-response. Hence the definition of jump variations is variations where occasionally

$$d(n) \geq 1, \tag{4}$$

but otherwise,

$$d(n) \ll 1. \tag{5}$$

In this case "occasionally" is defined as a period of time long enough for the algorithm to achieve the steady-state where the error is approximately equal to the additive noise.

4. RANDOM IMMIGRANTS AND BGA-TYPE ALGORITHMS

In this section, the performance of two genetic adaptive algorithms operating in a variety of nonstationary environments is investigated. The first algorithm is the modified genetic adaptive algorithm described in [4]. The lattice coefficients are encoded as floating-point numbers and the mutation operator used is that from the breeder genetic algorithm (BGA) described in [14]. This scheme randomly chooses, with probability 1/32, one of the 32 points $\pm(2^{-15}A, 2^{-14}A, \dots, 2^{0}A)$, where A defines the mutation range and is, in these simulations, set to $0.1 \times$ coefficient range. The crossover operator involved selecting two parent filter structures at random and generating identical copies. Two cut points were randomly selected and coefficients lying between these limits were swapped between the offspring. The newly generated lattice filters were then inserted into the population replacing the two parent structures.

A measure of fitness of the new filter was obtained by calculating the MSE for a block of current input and output data. A block length of 10 input-output pairs was used for the experiments reported below on a slowly varying system while a length of 5 input-output pairs was used for the rapidly varying system. Fitness scaling was used, as described in Goldberg [15, page 77], and fitness proportional selection was implemented using Baker's stochastic universal sampling algorithm [16]. Elitism was used to preserve the best performing individual from each generation. Crossover and mutation rates were set to 0.1 and 0.6, respectively, and the population contained 400 models. It was hoped that the use of the BGA mutation scheme would give this algorithm a greater ability to follow system changes than that of a GA using a more conventional mutation scheme, as the BGA algorithm retains, even when the population has comparatively converged, significant probability of making substantial changes in the coefficients if the system that it is modelling is found to have changed.

In competition with this genetic optimizer, the random immigrants mechanism of Cobb and Grefenstette, discussed above, was placed. For this set of simulation experiments, 20% of the population was replaced by randomly generated individuals every 10 generations. The same controlling parameters were used for both GAs.

4.1. The test systems

Deterministically varying environments were produced by making nonrandom alterations to the coefficients of a sixth-order all-pole lattice filter. In the case of slow and rapid time variations, the lattice coefficients were varied in a sinusoidal or cosinusoidal fashion taking in the full extent of the coefficient range (±1). Changes to the plant coefficients were effected at every sample instant with the precise magnitude of these variations reflected in the value of d for each environment. With measurement noise suitably scaled to give a signal-to-noise ratio of approximately 40 dB, the nonstationarity degrees of the slow and rapidly varying systems are 0.03 and 1.6, respectively.

Traditional (nonevolutionary) adaptive algorithms can run into problems when called upon to track rapid time variations (d permanently greater than one). When these changes occur infrequently, however, the well-documented

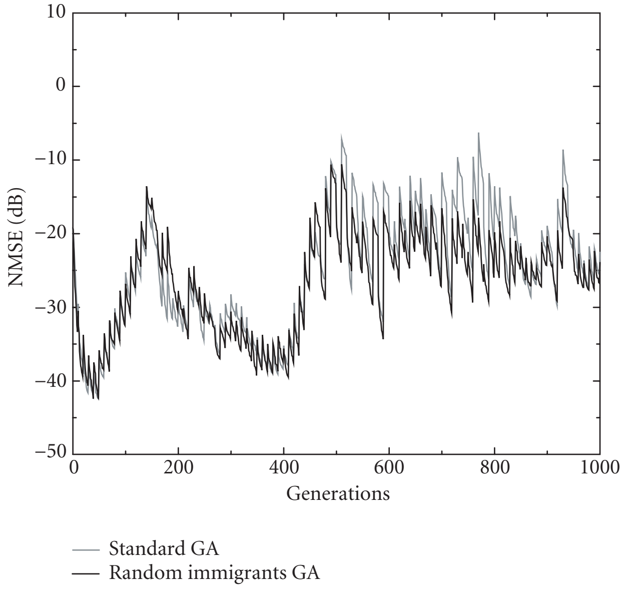

Figure 3: Performance of the genetic adaptive algorithm in a rapidly varying environment ($d = 1.6$).

transient behaviour of the adaptive algorithm can be used to describe the time to convergence and excess MSE that results. In order to investigate the performance of the genetic adaptive algorithm under such conditions, an environment was constructed in which the time variations of the plant coefficients are occasional and are often large in magnitude. The system to be modelled was once again a sixth-order all-pole filter. The infrequent time variations were introduced by periodically negating one of the plant lattice coefficients. As a consequence, for much of the simulation, the unknown system is time invariant ($d = 0$) with the nonstationarity degree greater than zero only during the occasional step changes.

4.2. Results

The performance of the BGA-based algorithm and random immigrants GA was evaluated in each of the three time-varying environments detailed. In each case, fifty GA runs were performed using the same environment (time-varying system).

In both the slowly changing and the jump environments, the behaviour was more or less as expected. In the slowly changing environment, both algorithms were able to reduce the error to near the −40 dB noise floor (set by the level of noise added to the system) and inspection of the parameters shows them to be following the changes in the system well. In the case of the step changes, the random immigrants algorithm exhibited better behaviour, recovering more quickly when the system changed. The tracking of rapid changes however is more difficult than either of these, and hence of more interest, and in this neither of the algorithms are particularly successful. The error reduction performance of the two adaptive algorithms is illustrated in Figure 3. In addition to rapid small-scale excursions resulting from the use of blocked input-output data, the extent to which the unknown system is correctly identified fluctuates on a more macroscopic scale. The normalised mean square error (NMSE) varies between the theoretical minimum of −40 dB and a maximum of around −8 dB, eventual settling down to a mean of around −20 dB.

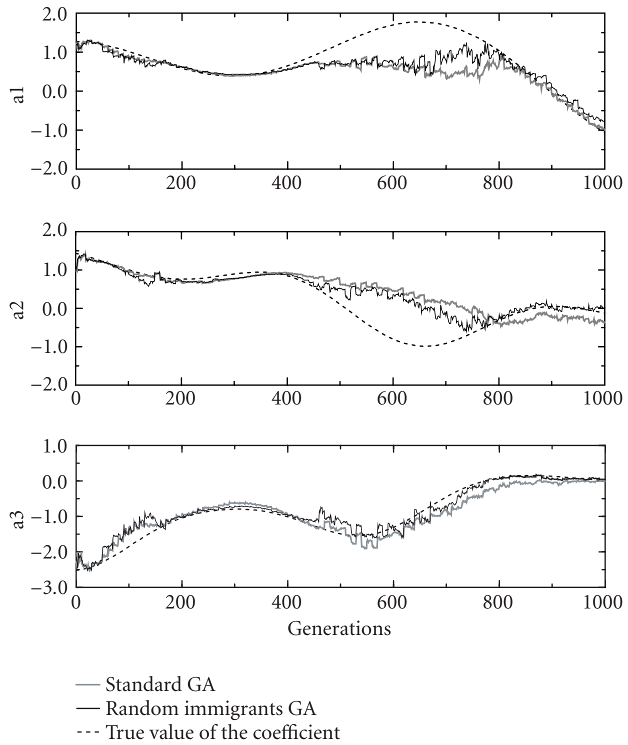

Figure 4: Genetic adaptive algorithm tracking performance in a rapidly varying environment ($d = 1.6$).

These phenomena can be explained when one looks at a graph of the coefficient tracking performance (Figure 4). The graph shows the time evolutions of the first three direct-form coefficients of the plant (represented by a dotted line) and the best adaptive filter in the population. The coefficients generated by the standard floating point GA are depicted by a gray line whilst those produced by the random immigrants GA are represented by a black line. Neither the standard floating-point GA nor the random immigrants GA were able to track the rapid variations in the plant coefficients throughout the entire run. The periods when the best adaptive filter coefficient values differed significantly from the optimal values correspond, in both cases, to the times when the identification was poor (see Figure 3).

5. HYBRID GENETIC ALGORITHMS

Clearly, an algorithm which would be better able to track rapid changes system parameters would be useful. A possible method is to devise a hybrid algorithm combining the global properties of the GA with a local search method to follow the local variations in the parameters. In this way, the two major failings of the individual components of the hybrid can be addressed. The GA is often capable of finding reasonable solutions to quite difficult problems but its characteristic slow finishing is legendary. Conversely, the huge array of gradient-based and gradientless local search techniques run the risk of becoming hopelessly entangled in local optima. In combining these two methodologies, the hybrid GA has been shown to produce improvements in performance over the constituent search techniques in certain problem domains [17, 18, 19, 20].

Goldberg [15, page 202] discusses a number of ways in which local search and GAs may be hybridized. In one configuration, the hybrid is described in terms of a batch scheme. The GA is run long enough for the population to become largely homogeneous. At this point, the local optimization procedure takes over and continues the search, from perhaps the best 5 or 10% of solutions in the population, until improvement is no longer possible. This method allows the GA to determine the gross features of the solution space, hopefully resulting in convergence to the basin of attraction around the global optimum, before switching to a technique better suited to fine tuning of the solutions. An alternative approach is to embed the local search within the framework of the GA, treating it rather like another genetic operator. This is the scheme adopted by Kido et al. [18] (who combine GA, simulated annealing, and TABU search), Bersini and Renders [20] (whose GA incorporates a hill-climbing operator), and Miller et al. [19] (who employ a variety of problem-specific local improvement operators). This second hybrid configuration is better suited to the identification of time-varying systems. In this case, the local search heuristic is embedded within the framework of the EA and is treated as another genetic operator. The local optimization scheme is enabled for a certain number of iterations at regular intervals in the GA run.

The hybrid approach utilizes a random hill-climbing technique to perform periodic local optimization. This procedure is ideally suited to incorporation in the EA since it does not require calculation of gradients or any other auxiliary information. Instead, the same evaluation function can be employed to determine the merit of the newly sampled points in the coefficient space. Since the technique is "greedy," the locally optimized solution is always at least as good as its genetic predecessor. In addition, once a change in the unknown system has occurred and is detected by a degradation of the model's performance, no *new* data samples are required. The hill-climbing method incorporated here into the GA is the random search technique proposed by Solis and Wets [21]. This algorithm randomly generates a new search point from a uniform distribution centred about the current coefficient set. The standard deviation of the distribution ρ_k is expanded or contracted in relation to the success of the algorithm in locating better performing models. If the first-chosen new point is not an improvement on the original point, the algorithm tests another point the same distance away in exactly the opposite direction.

In detail, the structure of the algorithm as used here is as follows. Firstly, the parameter ρ_k is updated, being increased by a factor of 2 if the previous 5 iterations have all yielded improved fitness, decreased by a factor of 2 if the previous 3 iterations have all failed to find an improved fitness, and left unchanged if neither of these conditions has been met. In the second step, a new candidate point in coefficient space is obtained from a normal distribution of standard deviation ρ_k centred on the current point. The fitness of this new point is then evaluated. If the fitness is improved, the new point is retained and becomes the current point; if the fitness is not improved, the point an equal distance in the opposite direction is tested; and if better, it becomes the current point. If neither yields an improvement, the current point is kept and the algorithm returns to the first step.

The use of this hybrid arrangement of EA and hill climber introduces further control parameters into the adaptive system, namely, the number of structures to undergo local optimization and the number of iterations in each hill-climbing episode. Two extremes were investigated. In the first, hybrid *A*, every model in the population underwent a limited amount of hill climbing. The other configuration, hybrid *B*, locally optimized only the best structure in the population at each generational step. In order to allow for direct comparison with the results in the previous section, the population size was reduced so that there would be approximately the same number of function evaluations in each case. For hybrid *A*, each model in a population of 100 underwent three iterations of the hill-climbing algorithm at every generational step while for hybrid *B* the population was set to 300 and then the best at each generation was optimized over approximately 100 iterations of the random hill-climbing procedure.

Simulation experiments indicated that both hybrids were able to track the slowly varying environment requiring less than two hundred generations to acquire near-optimal coefficient values. The smaller population size implemented in each case resulted in poorer initial performance, but this was offset by the increased rate of improvement brought about by the local hill-climbing operator. In the case of intermittent step changes in the unknown system characteristics, the performance of the two hybrids was observed to fall between that of the standard and random immigrants GAs. Figure 5 compares the tracking performance of these two hybrid GA configurations in a rapidly changing environment. Hybrid *A* (development of every individual) is represented by a gray line. The second hill-climbing/GA hybrid (development of the best individual) is shown by a black solid line. Although a slight bias in the estimated coefficients is sometimes in evidence, hybrid *A* is clearly able to track the qualitative behaviour of the plant coefficients. Development of the best individual, however, is not sufficient to induce reliable tracking and the performance of hybrid *B* suffers as a result.

The addition of individual improvement within the EA framework has resulted in an adaptive algorithm which is able to track the coefficients of a rapidly varying system ($d > 1$) with some success. This is a feat which poses considerable problems to conventional adaptive algorithms (see Section 3.1). Wholesale local improvement was observed to

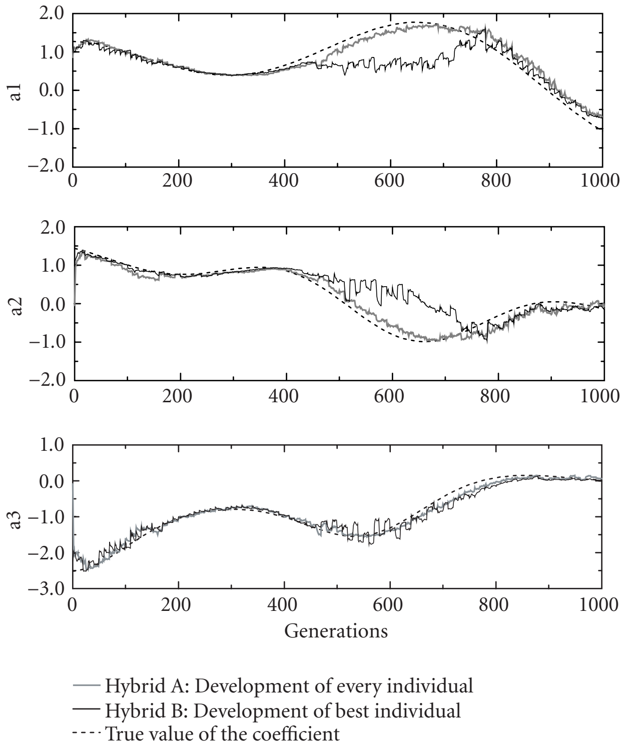

Figure 5: Genetic adaptive algorithm tracking performance in a rapidly varying environment ($d = 1.6$).

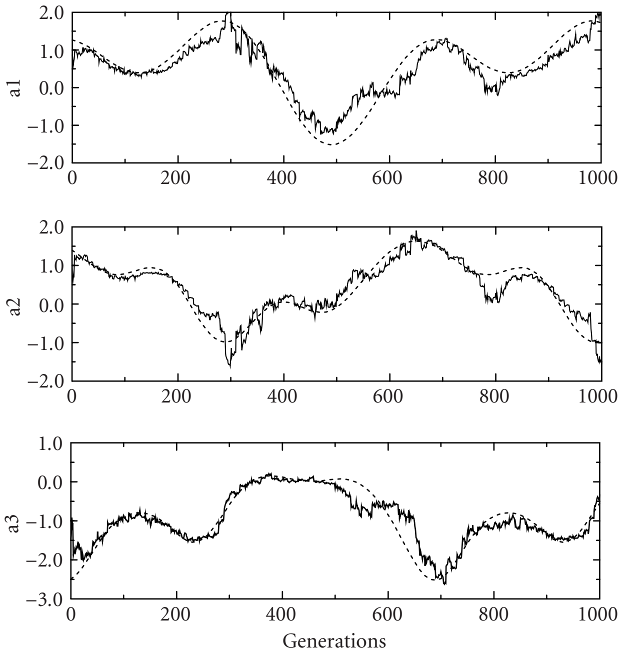

Figure 6: Genetic adaptive algorithm tracking performance in a rapidly varying environment ($d = 4.8$).

outperform the development of a single individual since this latter technique leaves the remainder of the population trailing behind the best structure. As the nonstationarity degree of the plant is increased, an adaptive algorithm relying solely upon evolutionary principles will lag further behind the time variations. This hybrid technique, however, permits the provision of greater local optimization flexibility (more iterations of the hill climber) when required.

Figure 6 illustrates the tracking performance of the hybrid GA subjected to a time-varying environment in which the nonstationarity degree was three times greater than in the previous experiment ($d = 4.8$). The population in this case contained 400 models, each one undergoing ten local optimization iterations at every generational step. The input-output block size was further reduced to just two samples in order that the plant coefficients would not vary substantially within the duration of a data block. This resulted in the coefficient estimates generated by the hybrid adaptive algorithm fluctuating about their trajectory to a greater extent. Individual evaluations of candidate models, however, required far less computation. The overall tracking performance of the hybrid was observed to be less accurate in this case but the mean estimates of the time-varying plant coefficients were observed to express the correct qualitative behaviour.

With emphasis shifting away from the role of evolutionary improvement in the hybrid adaptive algorithm as the time variations become more extreme, the balance of exploration versus exploitation (or global versus local search) is altered. This highlights that no single adaptation scheme is likely to outperform all others on every class of time-varying problem. On slowly varying systems, for example, a more or less conventional EA provided good performance. When the unknown system was affected by intermittent but large-scale time variations, the wider ranging search of the random immigrants operator was required. If the error surface is multimodal, hill-climbing operators are unlikely to provide the desired search characteristics. Conversely, with a rapidly changing system, the fast local search engendered by the hill-climbing operator provides the necessary response since only relatively minor changes to the optimal coefficients occur at each generational step. However, this classification assumes that the nature of the time variations affecting the unknown system is known in advance. When such information is not available or when more than one class of time variation is present, some combination of techniques may be desirable.

6. CONCLUSIONS

On system identification tasks where the plant coefficients are changing slowly ($d \ll 1$), both the floating-point GA and the random immigrants GA were able to track the time variations. However, when the time variations were infrequent but large in magnitude (jump variations), the standard GA was unable to react quickly to the changes in the coefficient values; but the random immigrants mechanism, on the other hand, produced sufficient diversity in the population to rapidly respond to such step-like time variations. Neither algorithm was able to successfully track the plant coefficients when the time variations were rapid and continuous ($d > 1$). In the final section of the paper, a hybrid scheme is introduced and shown to be more effective than either of the earlier schemes for tracking these rapid variations.

REFERENCES

[1] D. M. Etter, M. J. Hicks, and K. H. Cho, "Recursive adaptive filter design using an adaptive genetic algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '82)*, vol. 2, pp. 635–638, IEEE, Paris, France, May 1982.

[2] R. Nambiar, C. K. K. Tang, and P. Mars, "Genetic and learning automata algorithms for adaptive digital filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '92)*, pp. 41–44, IEEE, San Francisco, Calif, USA, March 1992.

[3] K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 5, pp. 1033–1046, 1992.

[4] M. S. White and S. J. Flockton, "Adaptive recursive filtering using evolutionary algorithms," in *Evolutionary Algorithms in Engineering Applications*, D. Dasgupta and Z. Michalewicz, Eds., pp. 361–376, Springer-Verlag, Berlin, Germany, 1997.

[5] H. G. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments," Tech. Rep. 6760, Navy Center for Applied Research in Artificial Intelligence, Washington, DC, USA, December 1990.

[6] J. J. Grefenstette, "Genetic algorithms for changing environments," in *Proc. 2nd International Conference on Parallel Problem Solving from Nature (PPSN II)*, R. Männer and B. Manderick, Eds., pp. 137–144, Elsevier, Amsterdam, September 1992.

[7] H. G. Cobb and J. J. Grefenstette, "Genetic algorithms for tracking changing environments," in *Proc. 5th International Conference on Genetic Algorithms (ICGA '93)*, S. Forrest, Ed., pp. 523–530, Morgan Kaufmann, San Mateo, CA, USA, July 1993.

[8] A. Neubauer, "A comparative study of evolutionary algorithms for on-line parameter tracking," in *Proc. 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., pp. 624–633, Springer-Verlag, Berlin, Germany, September 1996.

[9] F. Vavak, T. C. Fogarty, and K. Jukes, "A genetic algorithm with variable range of local search for tracking changing environments," in *Proc. 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., pp. 376–385, Springer-Verlag, Berlin, Germany, September 1996.

[10] E. Pettit and K. M. Swigger, "An analysis of genetic-based pattern tracking and cognitive-based component tracking models of adaptation," in *Proc. National Conference on Artificial Intelligence (AAAI '83)*, pp. 327–332, Morgan Kaufmann, San Mateo, CA, USA, August 1983.

[11] A. H. Gray Jr. and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 6, pp. 491–500, 1973.

[12] O. Macchi, *Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission*, John Wiley & Sons, Chichester, UK, 1995.

[13] O. Macchi, N. Bershad, and M. Mboup, "Steady-state superiority of LMS over LS for time-varying line enhancer in noisy environment," *IEE Proceedings F*, vol. 138, no. 4, pp. 354–360, 1991.

[14] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm I. Continuous parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.

[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing, Reading, Mass, USA, 1989.

[16] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Genetic Algorithms and Their Applications: Proc. 2nd International Conference on Genetic Algorithms (ICGA '87)*, J. J. Grefenstette, Ed., pp. 14–21, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, July 1987.

[17] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as a function optimizer," in *Proc. 4th International Conference on Genetic Algorithms (ICGA '91)*, R. K. Belew and L. B. Booker, Eds., pp. 271–278, Morgan Kaufmann, University of California, San Diego, Calif, USA, July 1991.

[18] T. Kido, H. Kitano, and M. Nakanishi, "A hybrid search for genetic algorithms: Combining genetic algorithms, TABU search, and simulated annealing," in *Proc. 5th International Conference on Genetic Algorithms (ICGA '93)*, S. Forrest, Ed., p. 641, Morgan Kaufmann, University of Illinois, Urbana-Champaign, Ill, USA, July 1993.

[19] J. A. Miller, W. D. Potter, R. V. Gandham, and C. N. Lapena, "An evaluation of local improvement operators for genetic algorithms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 23, no. 5, pp. 1340–1351, 1993.

[20] H. Bersini and J.-M. Renders, "Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways," in *Proc. 1st IEEE Conference on Evolutionary Computation (ICEC '94)*, D. B. Fogel, Ed., vol. I, pp. 312–317, IEEE, Picataway, NJ, USA, June 1994.

[21] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 19–30, 1981.

Michael S. White was a student at Royal Holloway, University of London, where he received the B.S. and Ph.D. degrees. He is currently employed by a New York-based hedge fund.

Stuart J. Flockton received the B.S. and Ph.D. degrees from the University of Liverpool. He is a Senior Lecturer at Royal Holloway, University of London. His research interests centre around signal processing and evolutionary algorithms.