Parallel genetic algorithm based unsupervised scheme for extraction of power frequency signals in the steel industry

P.K. Nanda, B. Ghose and T.N. Swain

Abstract: A steel industry has different types of loads, and so the incoming supply voltage of some units becomes distorted thus affecting those systems that depend on a distortionless supply. A novel unsupervised scheme named the recursive hybrid parallel genetic algorithm based line enhancer (RHPGABLE) scheme is proposed, to track the desired power frequency signal from the corrupted one. The RHPGABLE scheme is based on a proposed new crossover operator known as the generalised crossover (GC) operator. The delay and the filter coefficients are estimated recursively to yield optimal solutions. In the recursion of the proposed RHPGABLE algorithm, a parallel genetic algorithm (PGA) based on a coarse-grained approach is employed to estimate the delay, while the filter coefficients are estimated by PGA and a least mean squares (LMS) algorithm. RHPGABLE is an unsupervised scheme in the sense that no *a priori* knowledge of delay or filter coefficients and the associated training signal component is assumed to be available. The proposed scheme has been tested successfully on both synthetic data and data obtained from the Steel Melting Shop of Rourkela Steel Plant, Orissa, India.

1 Introduction

Different types of load, such as linear, nonlinear, switching and inductive, occur in a steel plant. The load weighing system provided on the cranes determines the true weight of loads on these cranes. The major cranes utilised in the steel melting shop (SMS) of Rourkela Steel Plant are of 250 tons capacity since molten hot metal and molten steel weighing up to 250 tons or more is handled. The adopted weighing system is controlled through programmable logic controllers (PLC) designed by ABB Ltd. The accuracy of the load weighing system depends greatly on the incoming power supply to the system. However, the incoming power supply to the load weighing system suffers from noise due to: (i) the proximity of power supply cables; (ii) the effect of different loads; (iii) random load fluctuations and switching actions; (iv) frequent switching of inductive and passive elements such as contactors and timers; (v) switching of motors.

The above mentioned interfering signals greatly affect the weighing system. Interference may damage the equipment associated with the weighing system and necessitate frequent recalibrations. The problem of interference is viewed as noise interference with periodic signals. The input supply becomes distorted, with variable amounts of distortion that depend on the load of a given shop floor or the connected shops. Since there is a continuous change in loading conditions in a steel melting shop, the levels of contaminating noise also change. Parameters of active and passive filters need to be redesigned to meet the changing conditions.

The proposed scheme adapts to the changing system requirements, and hence is more suitable for the problem described than filters with fixed parameters. The scheme suppresses noise and, thus, protects any frequency-sensitive devices connected to the load weighing system.

The problem of extracting noise-free signals from corrupted signals has been pursued for at least three decades [1–3]. Many adaptive filtering techniques have been proposed to obtain viable solutions to this problem. The problem can be considered in two forms: (i) when a training signal is available; and (ii) when no training signal is available. Whenever a training signal is available, the usual approach is to obtain the optimal filter coefficients using a suitable adaptive algorithm [1–3]. Often, in practice, the training signal or the desired component may not be available. In such situations, one would employ the notion of adaptive line enhancement to eliminate the periodic interference with a broadband component [1, 2]. The performance of the adaptive line enhancer (ALE) depends upon a proper choice of the delay element and the adaptive algorithm. Usually, the delay is selected on an *ad hoc* basis, while the filter coefficients are estimated by a suitable adaptive algorithm. In the proposed RHPGABLE scheme, the delay is estimated together with the filter coefficients. Adaptive algorithms, based on the notion of gradient descent, suffer from the problem of local minima trapping and, thus, can yield suboptimal solutions. This inherent problem of local minima can be overcome by employing random search algorithms. Recently, the basic genetic algorithm (BGA) and its variants have been

© IEE, 2002

IEE Proceedings online no. 20020501

DOI: 10.1049/ip-vis:20020501

Paper first received 19th December 2000 and in final revised form 13th May 2002

P.K. Nanda and B. Ghose are with the Department of Electrical Engineering, Regional Engineering College, Rourkela, Orissa 769 008, India

T.N. Swain is with the Steel Melting Shop, Rourkela Steel Plant, Rourkela, Orissa 760 994, India

employed to determine an optimal set of filter coefficients [4, 5]. Because of the large population size, the computational burden is very high when implementing the BGA [6, 7]. The processing time is reduced by employing a parallel genetic algorithm (PGA) [8, 9] instead of the BGA and it also yields better solutions than the BGA. We have estimated the filter coefficients using the PGA and the least mean squares (LMS) algorithm, and the delay using the PGA.

In this paper, a new scheme called the recursive hybrid parallel genetic algorithm based line enhancer (RHPGABLE) is proposed. The efficacy of an adaptive line enhancement scheme depends on a proper choice of the delay that helps to separate the broadband component from the periodic signal. Since the choice of delay is crucial for devising an efficient scheme, we consider the problem of joint estimation of delay and filter coefficients. Joint estimation is difficult because the optimal estimate of filter coefficients depends upon the optimal estimate of the delay and *vice versa*. Hence, we formulate the problem so that the delay and the filter coefficients are estimated recursively. This recursive scheme yields partial optimal solutions instead of optimal solutions. The RHPGABLE scheme works in an unsupervised framework, where no *a priori* knowledge of the training signal, delay, and filter coefficients is assumed to be available. In the proposed scheme, the delay estimation step employs PGA, while the filter coefficient estimation step uses both PGA and the LMS algorithm. The PGA is based on a new crossover operator known as the generalised crossover (GC) operator. The proposed algorithm is validated with a tapped delay finite impulse response (FIR) filter. This algorithm successfully extracted the power frequency component corrupted by additive white Gaussian noise of varying strengths. The algorithm was also validated with practical data obtained from the steel melting shop of Rourkela Steel Plant. Since the proposed scheme is based on a parallel genetic algorithm, it is suitable for real-time implementation in extracting desired components from corrupted ones. The results presented in this paper correspond to a serial implementation of the proposed parallel algorithm.

2 Problem statement

2.1 Adaptive line enhancer

Fig. 1 shows a typical block diagram representation of an adaptive line enhancer (ALE), where the input denotes the corrupted signal and the output is the estimated signal. The delay element Δ denotes the amount by which the input signal is delayed before passing through the adaptive filter. A reference signal is not used in the ALE, as shown in Fig. 1. The delay element helps to separate the signal component from the noise component. The signal tracking capability depends greatly upon the proper choice of delay element. The adaptive filter considered is the FIR filter.

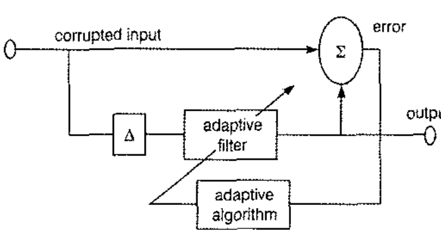

Fig. 1 *Adaptive line enhancer*

The delay and the optimal filter coefficients, $\boldsymbol{h}^* = [h_1^*, h_2^*, \ldots, h_N^*]$, have to be determined, where N denotes the order of the filter.

2.2 RHPGABLE scheme

To overcome the problem of *ad hoc* choice of delay, we propose a RHPGABLE scheme in which a simultaneous estimate of the delay as well as the filter coefficients are obtained. The RHPGABLE scheme is shown in Fig. 2. The following optimality criterion is adopted for the joint estimation:

$$(\boldsymbol{h}_{opt}, \Delta_{opt}) = \min_{(\boldsymbol{h},\Delta)} e(\boldsymbol{h}, \Delta) \tag{1}$$

where $e(\boldsymbol{h}, \Delta)$ is the error at the output of the adaptive filter, $\boldsymbol{h}_{opt}$ denote the optimal filter coefficients, and Δ_{opt} is the optimal delay. Solving (1) is difficult because the optimal value of the filter coefficients and the optimal value of the delay are interdependent and unknown. A set of optimal values of $\boldsymbol{h}^*$ depends upon the proper choice of Δ, and in the sequel, the optimal value of the delay depends upon the optimal value of $\boldsymbol{h}^*$. Hence, the above problem is split into the following:

$$\boldsymbol{h}^* = \min_{\boldsymbol{h}} e(\boldsymbol{h}, \Delta^*) \tag{2}$$

$$\Delta^* = \min_{\Delta} e(\boldsymbol{h}^*, \Delta) \tag{3}$$

In (2), determination of $\boldsymbol{h}^*$ is dependent on Δ^*, while in (3), determination of Δ^* is dependent on $\boldsymbol{h}^*$. $\boldsymbol{h}^*$ and Δ^* are known as 'partial optimal' solutions. Since $\boldsymbol{h}^*$ and Δ^* are not available, solving (2) and (3) is also a difficult task. For such problems in a deterministic framework [10], it has been shown that parameters can be recursively estimated to eventually converge to a partial optimal solution $\boldsymbol{h}^*$ and Δ^*. In the limiting case, these recursive estimates converge to partial optimal solutions. In the same spirit, we propose to adopt the following recursive scheme to find the partial optimal solutions $\boldsymbol{h}^*$ and Δ^* in a stochastic framework. Let k denote the kth iteration of the proposed algorithm. The following recursion is adopted for the above problem:

$$\hat{\boldsymbol{h}}(k+1) = \min_{\boldsymbol{h}} e(\boldsymbol{h}, \hat{\Delta}(k)) \tag{4}$$

$$\hat{\Delta}(k+1) = \min_{\Delta} e(\hat{\boldsymbol{h}}(k+1), \Delta) \tag{5}$$

The combined recursion consists of estimation of $\hat{\boldsymbol{h}}(k+1)$ using $\hat{\Delta}(k)$, and estimation of $\hat{\Delta}(k+1)$ using $\hat{\boldsymbol{h}}(k+1)$. These recursive estimates, $\hat{\Delta}(k)$ and $\hat{\boldsymbol{h}}(k)$, will eventually converge to partial optimal solutions. The PGA and the LMS algorithm are employed to obtain $\hat{\boldsymbol{h}}(k+1)$, while only the PGA is employed to obtain the estimate of $\hat{\Delta}(k+1)$. Thus, $\hat{\boldsymbol{h}}(k+1)$ and $\hat{\Delta}(k+1)$ are recursively obtained in a

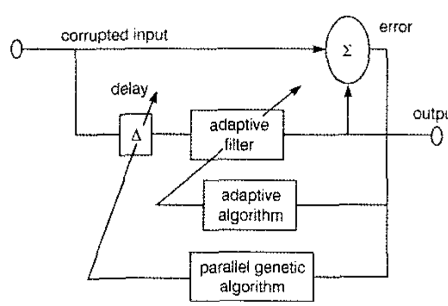

Fig. 2 *Recursive hybrid parallel genetic algorithm based adaptive line enhancer (RHPGABLE)*

stochastic framework until the convergence criterion is met. We considered the following convergence criterion. The algorithm stops when $|\hat{\Delta}(k) - \hat{\Delta}(k-1)| < threshold$. The value of the threshold is 2. This indicates that two consecutive estimates are equal.

3 Parallel genetic algorithm

In GAs, the population size is one of the parameters that determines the quality of solution. As population size increases, the GA has a better chance of finding the global solution. The increase in population size results in a heavy computational burden. Hence with a serial GA, one has to choose between obtaining a good result with high confidence and paying a high computational cost, or reducing the confidence requirement and getting a possibly poor result more quickly. In contrast, parallel GAs can keep the quality of the results high and find them rapidly, because larger populations can be processed in less time using parallel machines. Parallel genetic algorithms (PGAs) have been used to find solutions to many complex problems [8, 9]. The motivation behind the use of PGAs is twofold: (i) to reduce the processing time needed to reach an acceptable solution; (ii) to obtain, in some sense, better solutions than those from comparably sized serial GAs. GAs can be parallelised using either a coarse- or fine-grained approach. In fine-grained parallel GAs, the evaluation of individuals and the application of genetic operators are explicitly parallelised so that every individual has a chance to mate with all the rest. The speedup gained is proportional to the number of processors used.

In the coarse-grained approach, the population is divided into a few sub-populations that are kept relatively isolated from each other. This method of parallelisation introduces a migration operator and a migration policy which help to send some individuals from one sub-population to another. In the coarse-grained approach, the two genetic models of population structures used extensively are: (i) the island model; and (ii) the stepping stone model. The population in the island model is partitioned into small sub-populations by geographic isolations where individuals can migrate to any other sub-population. In the stepping stone model, the population is partitioned into small sub-populations, but migration is restricted to neighbouring sub-populations. Our algorithm is implemented based on a coarse-grained approach incorporating the island model. There are different migration policies and the solutions depend upon the choice of migration policy. We have used a migration policy in which good migrants replace bad individuals of a sub-population.

4 Generalised crossover (GC) operator

In this paper, we introduce a new crossover operator known as the generalised crossover (GC) operator which, when applied to two parents, produces one offspring instead of two as in the basic genetic algorithm. The GC operator is shown in Fig. 3. The operator can be described as follows. Two parents p_1 and p_2 are selected at random and two crossover points are also selected at random. In between the two crossover points, two bits of the respective positions of the two selected parents are now passed through a nonlinear function (switching function) to produce one output. If x and y are two switching variables, then the possible switching functions $f(x, y)$ are 0, $x'y'$, $x'y$, x', xy', y', $x'y+xy'$, $x'+y'$, xy, $xy+x'y'$, y, $x'+y$, x, $x+y'$, $x+y$, 1. Of the above 16

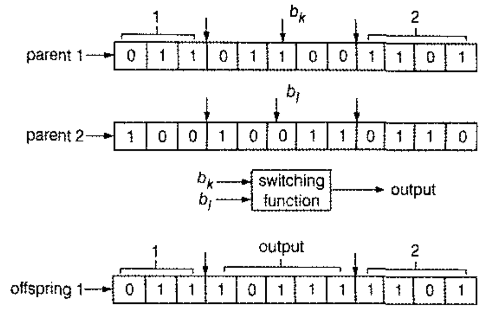

Fig. 3 *Generalised crossover operator*

functions, 0 and 1 are not used because they correspond to inconsistent functions. For the two-variable case, a switching function is selected at random from the above-mentioned functions and the two bits are impressed as the input. The corresponding output is stored in the same bit position as one of the parents. Analogously, all other bits are generated by selecting the other respective bits from the two parents and passing them through the randomly selected switching function. So, a stream of bits between the two crossover points is generated that replaces one of the parents to generate one offspring. The motivation is twofold: (i) it helps to explore the diversity of solutions in the solution space; and (ii) this model is more plausible from the evolutionary sense in that two parents produce one offspring at a time. The same GC operator is applied again to the same two parents with the two new randomly chosen crossover points to produce one more offspring. As a result of this operation, two offspring are produced from the two parents by applying the GC operator twice. Thus, N offsprings are generated from N parents and the total population is maintained constant by employing a suitable selection mechanism.

5 Implementation

Fig. 2 shows the implementation of the proposed RHPGABLE scheme in which the delay as well as the filter coefficients are estimated recursively. Since the estimates of the filter coefficients are dependent on the choice of the delay, both filter coefficients and delay are estimated recursively. Estimates of the delay and the filter coefficients eventually converge to the desired solution after several combined iterations. One estimation of the delay and estimation of one set of filter coefficients constitute one combined iteration. This process is repeated until the convergence criterion is met, that is, the error is below the pre-selected level. The algorithm starts with an arbitrary choice of delay, and using this delay, the filter coefficients are estimated using the PGA and the LMS algorithm. In the filter coefficient estimation step, the PGA is run for a sufficient number of iterations, such that the output error is below a threshold, or in other words, the average fitness is above 0.6. These estimated coefficients are used as the initial coefficients for the LMS algorithm. The fitness function used is $fit_i(k) = 1/(1 + (e_i(k))^2)$, where $fit_i(k)$ denotes the fitness of the ith parent in kth generation of a sub-population and $e_i(k)$ denotes the error for the ith parent at kth iteration of a sub-population. The fitness of an individual might grow and the maximum value of the fitness can be unity, which indicates that the string is the best fit. The current implementation of the RHPGABLE algorithm is serial, and attempts are being made at its parallel implementation. Salient steps of the algorithm are given below.

RHPGABLE algorithm

Step 1. Initialise the filter coefficients to random values from a uniform distribution. Select an arbitrary delay.

Step 2. Using the delay and the randomly selected filter coefficients estimate the filter coefficients using the PGA and the LMS algorithm.

(i) Initially the population is partitioned into a number of sub-populations.

(*a*) In each sub-population at *k*th generation of the GA,

$$\hat{x}(k) = \sum_{n=0}^{(N-1)} \boldsymbol{h}_n(k)y(k-n-\Delta)$$
$$e(k) = y(k) - \hat{x}(k)$$

where N denotes the order of the filter. After every 10 generations, 20% of the individuals of a sub-population migrate to other sub-populations with a migration probability P_{mig}. The individuals are ranked in descending order according to their fitness. The 20% of individuals having high fitness are selected for migration. The sub-populations are connected so that the selected individuals are migrated to every other sub-population.

(*b*) Step (*a*) is repeated until the average fitness of the total population elements is above a threshold value. The threshold value is 0.6. Thus an estimate of $\Delta\hat{\boldsymbol{h}}(k)$ is obtained by minimizing $e^2(k)$.

(ii) The filter coefficients, $\boldsymbol{h}(k)$ are updated to $\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + \Delta\hat{\boldsymbol{h}}(k)$.

(iii) Steps (i) and (ii) are repeated until the average fitness is above a selected threshold.

(iv) The filter coefficients obtained from step (iii) are used as the initial coefficients for the LMS algorithm. At the *k*th step of the LMS algorithm,

$$x(k) = \sum_{n=0}^{(N-1)} \boldsymbol{h}_{G_n}(k)y(k-n-\Delta)$$
$$e(k) = y(k) - x(k)$$
$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + 2\mu(k)e(k)y(k-\Delta)$$

where N is the order of the filter, μ is the convergence coefficient, and $\boldsymbol{h}_{G_n}$ denotes the filter coefficients updated by the PGA. The filter coefficients are updated until the error is below a pre-selected threshold.

Step 3. Using the estimated filter coefficients $\boldsymbol{h}(k+1)$ from step 2, the estimated delay $\hat{\Delta}(k+1)$ is obtained by employing the PGA. The population is divided into a number of sub-populations. In each sub-population, at the *k*th iteration of the PGA,

(*a*)

$$x(k) = \sum_{n=0}^{(N-1)} \boldsymbol{h}_n(k+1)y(k-n-\Delta)$$
$$e(k) = y(k) - x(k).$$

The estimate of the delay is obtained for a few generations while minimising $e^2(k)$. Subsequently, the selected number of individuals are migrated to other sub-populations with migration probability P_{mig}.

(*b*) Step (*a*) is repeated until the average fitness of the total population elements is above a threshold and thus $\hat{\Delta}(k+1)$ is obtained while minimising $e^2(k)$.

Step 4. Steps 2 and 3 are repeated until the convergence criterion is met. The convergence criterion is that the estimated delay does not change for three consecutive iterations.

6 Results and discussion

In simulations, we have considered both synthetic data and practical data obtained from the load weighing system at the steel melting shop of Rourkela Steel Plant. In the case of synthetic data, we have considered the following signal models:

$$y(t) = A\sin(wt) + n(t) \quad (6)$$
$$y(t) = A\sin(wt) + \frac{A}{3}\sin(3wt) + \frac{A}{5}\sin(5wt) + n(t) \quad (7)$$

where $y(t)$ is the corrupted signal $A\sin(wt)$ is the desired component, A is the amplitude of the signal, and $n(t)$ is zero mean white Gaussian noise. In (6), $y(t)$ corresponds to a simple sinusoidal signal, and in (7), $y(t)$ is a periodic signal consisting of a fundamental component plus odd harmonics. Signal-to-noise ratio (SNR) is defined as $SNR_{dB} = 10\log_{10}(SP/NP)$, where SP and NP denote the signal and noise power respectively. The filter considered in both models is an FIR filter having order $N = 16$. Although both models could be tested for different corrupted signals with varying noise strength, we present only a few typical cases.

Fig. 4 shows the results obtained for a corrupted input with SNR = 10 dB while employing the RHPGABLE scheme. The signal model given by (6) is considered for

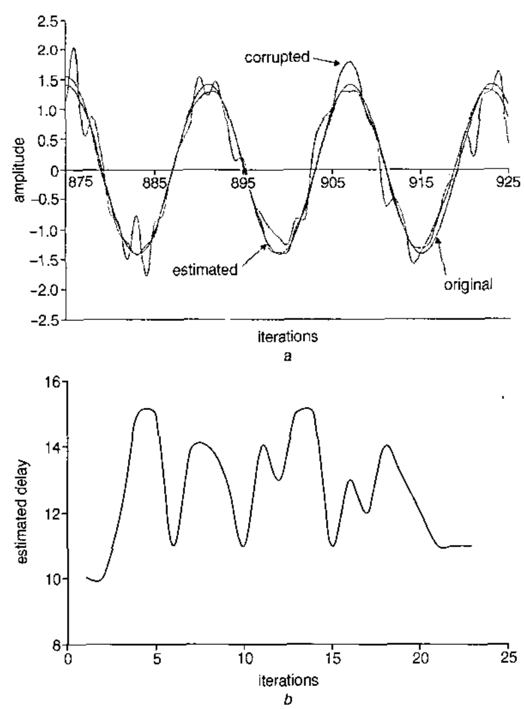

Fig. 4 *RHPGABLE scheme*
a Time domain signals showing the corrupted input of SNR = 10 dB, simple sinusoidal desired signal and estimated signal
b Estimated delay

simulation. The initial value of the delay is chosen to be 10. Each filter coefficient is binary coded, of 10 bits length, and one chromosome consists of 16 coefficients with binary coding. Using the above selected delay, the PGA is run for a sufficient number of iterations that the output error is below a threshold, or equivalently, the fitness is above a threshold. We have fixed the threshold value of the fitness to be 0.6. The weights obtained from PGA are used as the initial weights for the LMS algorithm and, thus, the filter coefficients are estimated for an arbitrarily chosen delay.

The parameters used for the PGA are: number of population elements, $M=60$; probability of crossover, $p_c=0.85$; probability of mutation, $p_m=0.006$; filter order, $N=16$; number of bits per filter coefficient, $B=10$ bits; probability of migration, $p_{mig}=0.9$; number of sub-populations, $d=4$; and migration rate, $\gamma=20\%$. It is seen from Fig. 4*b* that the algorithm started from an arbitrary delay of 10 and converged to a delay of 11 after 22 combined iterations. Using this converged delay of 11, the filter coefficients are estimated using the PGA and the LMS algorithm. These estimated coefficients are able to track the desired components even under high noise conditions, as shown in Fig. 4*a*. The corresponding squared error settles at around −50 dB after 800 iterations.

Fig. 5 shows the results obtained using the signal model given by (7) with input SNR = 20 dB. The proposed scheme started with an arbitrary delay of 12 and converged to 12 after 9 combined iterations. The delay is estimated by the PGA algorithm. This converged value of the delay is used to estimate the filter coefficients and, thus, extract the desired signal components. It is observed from Fig. 5*a*, that for a complex periodic signal, the algorithm extracted the desired components only after a larger number of iterations compared to the results obtained using the first model. The parameters used in the PGA are the same as those used in the signal model given by (6). The convergence coefficient $\mu(k)$ of the LMS algorithm is selected to be 0.06. μ is allowed to decrease with iteration in accordance with an exponentially decaying function.

6.1 Practical data of SMS of Rourkela Steel Plant

The RHPGABLE scheme was also tested using a set of real data, voltage signals obtained from the R, Y and B phases of the load weighing system. The input supply voltage was recorded by a recorder, and subsequently, the digital data were collected from the recorder. As observed in Fig. 6*a*, the input signal of the R-phase signal is quite distorted. The noise in this phase is modelled as Gaussian noise and the algorithm successfully tracked the desired component, as shown in Fig. 6*a*. For the sake of comparison with the estimated signal, the desired signal is also plotted in Fig. 6*a*. The algorithm started with an arbitrary delay of 13 and converged to a delay of 11 after 13 combined iterations. The corresponding error in dB is shown in Fig. 6*b*, where it is seen that the error has reduced to a level of −60 dB in approximately 100 iterations of the LMS algorithm. The parameters used for the PGA are the same as those used in the case of synthetic data. The initial value of μ is 0.09. It is clearly observed that the algorithm successfully extracted the signal components, while the noise was modelled as Gaussian noise.

Encouraging results are also obtained in case of Y- and B-phase signals. Although the Y-phase data is more distorted than the R-phase data, the algorithm tracked the desired component satisfactorily. The algorithm started with an arbitrary delay of 12 and converged to a delay of 13 after six combined iterations. The corresponding error is shown in Fig. 7*b*. The parameters for the PGA are

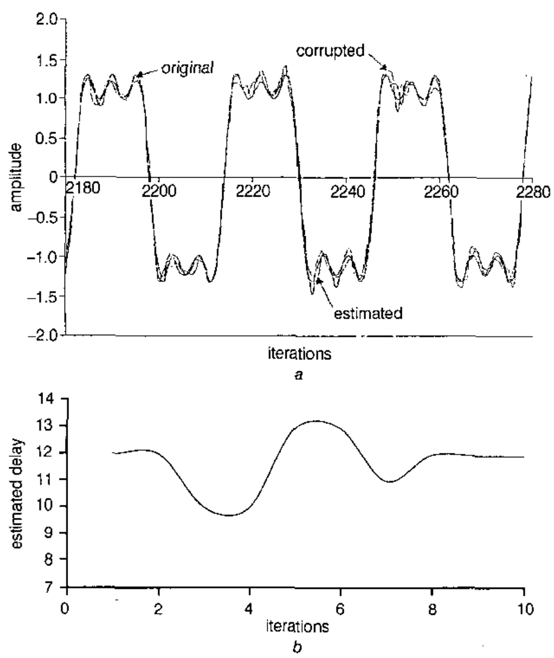

Fig. 5 *RHPGABLE scheme*
a Time domain signals showing the corrupted input of SNR = 20 dB, complex sinusoidal desired signal and estimated signal
b Estimated delay

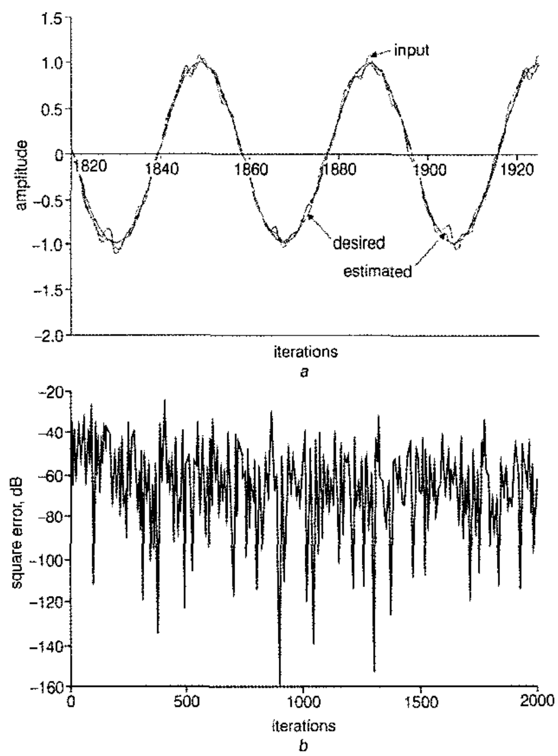

Fig. 6 *RHPGABLE scheme*
a Practical data (corrupted signal) of R-phase, normalised voltage signal obtained from the SMS of Rourkela Steel Plant, extracted signal and corresponding desired signal
b Corresponding squared error

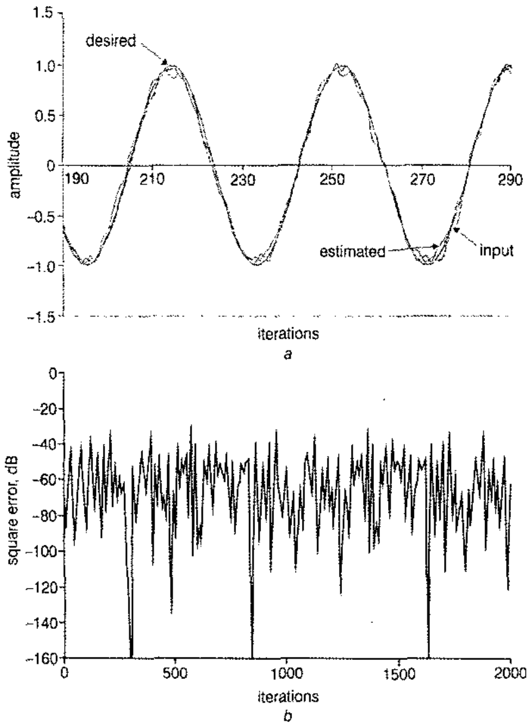

Fig. 7 *RHPGABLE scheme*
a Practical data (corrupted signal) of Y-phase, normalised voltage signal obtained from the SMS of Rourkela Steel Plant, extracted signal and corresponding desired signal
b Corresponding squared error

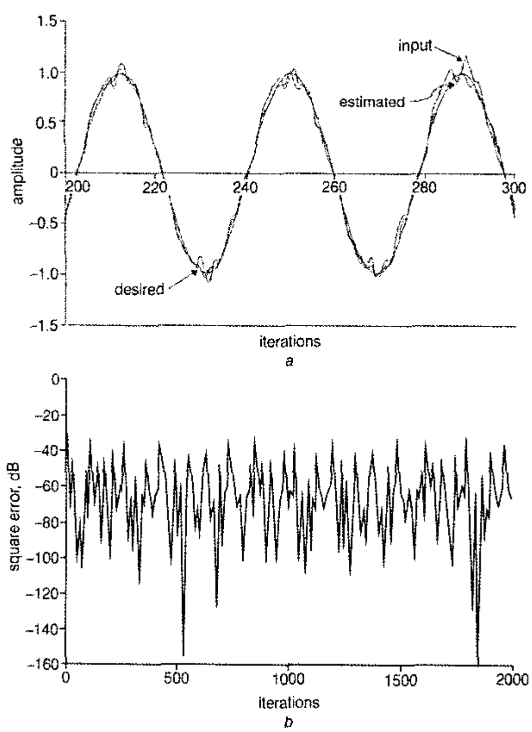

Fig. 8 *RHPGABLE scheme*
a Practical data (corrupted signal) of B-phase, normalised voltage signal obtained from the SMS of Rourkela Steel Plant, extracted signal and corresponding desired signal
b Corresponding squared error

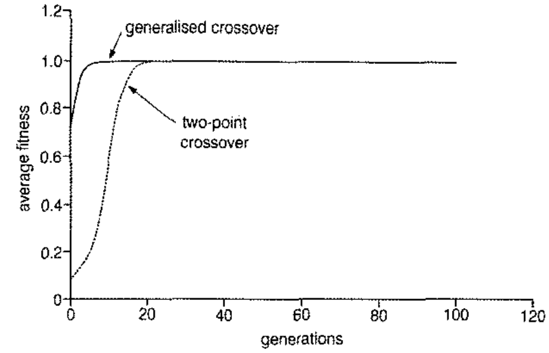

Fig. 9 *Average fitness corresponding to input signal of SNR = 10 dB, with generalised crossover operator and two-point crossover*

unchanged and the initial value of μ is 0.05. Using the converged value of 13, the algorithm tracked the desired component as shown in Fig. 7*a*.

Similarly, results obtained for the B-phase signal are shown in Fig. 8. The noise in the corrupted input is modelled as Gaussian. The algorithm started with an arbitrary delay of 10, and finally converged to the same delay after seven combined iterations. Even though the input signal of the B-phase is the worst of the three cases, the algorithm extracted the desired component. With the converged delay of 10, the PGA and the LMS algorithm are used to estimate the filter coefficients and, thus, extract the signal. The desired component is tracked within 300 iterations of the LMS algorithm as shown in Fig. 8*a*. The corresponding error is shown in Fig. 8*b*, where the error is below −40 dB after 50 iterations.

The efficacy of the proposed scheme with the GC operator is compared with that of using two-point crossover as shown in Fig. 9. It is observed in Fig. 9 that the GC operator based scheme converges faster than the scheme using two-point crossover. Thus, the proposed scheme performed well on practical data in an unsupervised framework.

7 Conclusions

A new adaptive line enhancement scheme named as the RHPGABLE scheme is proposed for extraction of periodic signals when training signals are not available. The scheme uses a novel crossover operator and the parallel genetic algorithm, thus, making it suitable from a practical standpoint. This is an unsupervised scheme in the sense that one need not have *a priori* knowledge of the delay and the filter coefficients. Here, the delay is estimated rather than selected on an *ad hoc* basis. The scheme was successfully tested on synthetic data as well as practical data obtained from the load weighing system of the SMS, Rourkela Steel Plant. The practical data obtained from Rourkela Steel Plant were for noisy load conditions. Application of this scheme could prevent damage to the costly equipment associated with the load weighing system. Furthermore, frequent calibration of the load weighing system may not now be required. The performance of the scheme with the new operator is found to be superior to that of a scheme using a two-point crossover operator. The results presented in this paper are for serial implementation of the RHPGABLE scheme. Currently, attempts are being made to obtain results for parallel implementation.

8 Acknowledgment

This research was supported by the Ministry of Human Resource Development (MHRD), India, Grant on Biomedical Image Processing Using Genetic Algorithm.

9 References

1 WIDROW, B., GLOVER, J.R., Jr, MCCOOL, J.M., KAUNITZ, J., WILLIAMS, C.S., HEARN, R.H., ZEIDLER, J.R., DONG, E., and GOODLIN, R.C.: 'Adaptive noise canceling: principles and applications', *Proc. IEEE*, 1975, **63**, (12), pp. 1692–1714
2 WIDROW, B., and STEARNS, S.D.: 'Adaptive signal processing' (Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1985)
3 HAYKIN, S.: 'Adaptive filter theory' (Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1986)
4 TANG, K.S., MAN, K.P., KWONG, S., and HE, Q.: 'Genetic algorithms and their Applications', *IEEE Signal Process. Mag.*, 1996, (11), pp. 22–37
5 NANDA, P.K., SAHU, P.K., SWAIN, S.K., and JAGANNATH, B.K.: 'Genetic Algorithm based adaptive line enhancement schemes for extraction of power frequency signal'. Proceedings of International Conference on Evolutionary computing for computer, communication, control and power, ECCAP-2000, India, Jan2000, pp. 91–98
6 GOLDBERG, D.E.: 'Genetic algorithm in search optimization machine learning' (Addison Wesley, 1989)
7 ATMAR, W.: 'Notes on simulation of evolution', *IEEE Trans. Neural Netw.*, 1994, **5**, (1), pp. 10–148
8 CANTU-PAZ, E.: 'A survey of parallel genetic algorithms', *Calculateurs Paralleles*, 1998, **10**, (2), pp. 141–171
9 CANTU-PAZ, E.: 'Designing efficient and accurate parallel gentic algorithms'. PhD Dissertation, Illinois Genetic Algorithm Laboratory, UIUC, USA, 1999
10 WENDELL, R.E., and HURTER, A.P., Jr.: 'Minimization of non separable objective functions subject to disjoint constraints', *Oper. Res.*, 1976, **24**, (4), pp. 643–657