Regular Paper

Adaptive filtering of EEG/ERP through noise cancellers using an improved PSO algorithm

Mitul Kumar Ahirwal a, Anil Kumar a,*, Girish Kumar Singh b,c

a PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482011, MP, India
b Department of Electrical Engineering, Indian Institute of Technology, Roorkee, Uttarakhand 247667, India
c PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482011, MP, India

A R T I C L E   I N F O

Article history:
Received 21 June 2012
Received in revised form 1 October 2013
Accepted 3 October 2013
Available online 11 October 2013

Keywords:
Adaptive Filters
EEG/ERP
Detection
LMS
NLMS
RLS

A B S T R A C T

In this paper, event related potential (ERP) generated due to hand movement is detected through the adaptive noise canceller (ANC) from the electroencephalogram (EEG) signals. ANCs are implemented with least mean square (LMS), normalized least mean square (NLMS), recursive least square (RLS) and evolutionary algorithms like particle swarm optimization (PSO), bacteria foraging optimization (BFO) techniques, genetic algorithm (GA) and artificial bee colony (ABC) optimization technique. Performance of this algorithm is evaluated in terms of signal to noise ratio (SNR) in db, correlation between resultant and template ERP, and mean value. Testing of their noise attenuation capability is done on EEG contaminated with white noise at different SNR levels. A comparative study of the performance of conventional gradient based methods like LMS, NLMS and RLS, and swarm intelligence based PSO, BFO, GA and ABC techniques is made which reveals that PSO technique gives better performance in average cases of noisy environment with minimum computational complexity.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Recordable brain wave in form of electrical signals replicates the response of stimulation, known as evoked potential (EPs), or event-related potentials (ERPs). Stimulation for ERP generation has various types like visual, auditory, and motor movement, etc [1–3]. Since ERPs are weak signals and buried in signals of spontaneous EEG with very low signal-to-noise ratio (SNR) [1]. Typically, ERP responses are categorized in to two types, based on the way they respond to stimulation. First type represents the time-locked and phase-locked responses, also termed as event-related potentials (ERP), and they appear for short time interval. Second type is time-locked, but not phase-locked (induced) modulation, called event-related de-synchronization (ERDS), and these are measured as alterations in the functional connectivity within the specified cortex of brain. Different analytical methods have been used for analysis of these two types of EEG changes [4].

Evoked activities responses in EEG are enhanced and extracted by simple linear methods based on synchronized averaging, while induced activities are analyzed using nonlinear methods such as power spectral analysis and rectified averaging [4]. Today, ERP analysis has become a major part of brain research. These ERP play an important role in design and development of brain computer interface (BCI) [4–6]. Effectiveness of ERP analysis depends only on EEG signal of high SNR value. However, EEG signals are noisy and non-stationary due to its process of generation from group of neurons. EEG signals are contaminated by artifacts due to line noise, muscle movements, sometime with cardiac signals (ECG), eye blinking and eyeball movements also [1,7]. Therefore, during the past decades, several techniques have been developed for the artifact removal from EEG signals [1].

Now a day’s application based EEG researches are the motivation behind the need of estimation of perfect ERP signals. Point of interest (POI) based image retrieval system is implemented using rapid image triage (RIT), which is the latest example for utilization of ERP generated in response of different images. Performance of RIT system is also related with the SNR of recorded ERP which leads to need of de-noising method to improve ERP [8]. Filtering is always beneficial when it is performed before any processing on ERP. In [9], optimal filtering (OF) is designed with ICA which improves the overall results as well as helps in dimension reduction of EEG data. Variability and changes in ERP is some time necessary, but it is difficult to trace. In order to overcome these problem, a Bayesian method having two stages is proposed and tested on variability of P300 (type of visual ERP) [10]. Sources localization of ERP is also important as SNR improvement, because the number of channels reduces only when the location of generation of particular type of ERP is known, in case of multiple channels. Source localization with spatial notch filter is proposed...
in [11] which are able to localize source accurately within high noise environment. Particle filtering (PF) has been proposed in [12] in order to track the variations of P3a and P3b parameters of P300 ERP within consecutive time-locked trials of the stimulated EEGs. Since the aim is to track latency, amplitude, and width of the ERP subcomponents across the trials. Especially for mental fatigue analysis and delection of fatigue levels where the relative variability of P300 subcomponents is the key factor, a novel spatiotemporal filtering method for single trial estimation of event-related potential (ERP) subcomponents is proposed [13]. This method is able to estimate temporally the correlated ERP subcomponents such as P3a and P3b.

The simplest and the most widely applied method for the analysis of ERPs is averaging of the measurements over an ensemble of trials, also known as ensemble averaging (EA). It is the optimal way to improve signal-to-noise ratio (SNR), when the underlying model of the observations is assumed that the ERP is a deterministic signal independent of additive background noise. ERP signals and background EEG noise are assumed to be uncorrelated in this traditional technique of EA. Based on the theory of researches present which confers the reliability of EA, probably ERP component variability is related to the psycho-physiological factors of a particular human being. Thus, in early stage of the research, many researchers have focused on the ensemble reduct in extracting a template ERP, assuming that the stimulus-induced changes in the EEG signal are very small [14–15]. Major drawback of the averaging technique is its dependency on number of trials or more trials are needed for better results. Independent component analysis (ICA) has been also employed for the purpose of noise reduction and extraction of ERPs with assumption that EEG observations are generated by linear mixing of a number of source signals, which must be statistically independent, and this is mostly true with regard to brain and oculomotor components [16–19].

Several wavelet based algorithms [2,20–22] have been developed and employed for noise reduction and extraction of ERPs by decomposing signals into several levels. Generally, these algorithms are called Wavelet de-noising. These techniques are complex, and are having complicated nature. Filtering is also the most commonly used method for single trial analysis of ERP, with which the contamination due to on-going background activity can be attenuated from the evoked potential. A major difficulty in the filtering method is that it offers often very low SNR or performance of the filter in the detection of the signals depends on statistical properties of the signal which is to be processed. To overcome these problems, the concept of adaptive filters and its applications as noise canceller was introduced by Widrow et al. [23]. Since then, adaptive noise cancellation techniques have been used in many engineering applications [1,23].

Literature review explores that various types of algorithms or error estimation methods have been exploited in adaptive filters to adjust the weights of filter, and error estimation according to the EEG signals and noise property [1,24–28]. Most efficient gradient based algorithms are least mean square (LMS), recursive least square (RLS) and their different variants. Recently, evolutionary techniques have emerged as robust tool for solving linear and non-linear equations. These techniques use the concept of random population generation which acts as possible solutions. Among evolutionary techniques, the most famous and robust technique are particle swarm optimization (PSO), genetic algorithm (GA), artificial bee colony (ABC), and bacteria foraging optimization (BFO). There are very few references available in which PSO, BFO and ABC have been employed for adaptive noise cancellations, system identification and channel equalization, respectively [29–37]. Genetic algorithm is the earliest optimization algorithm and also the simplest to understand. GA with its basic and modified version has been applied to optimize solution of different problems [38–40].

In this paper, a PSO and other evolutionary techniques have been assimilated with adaptive filter to de-noise the ERP. Analysis of PSO with respect to growing range of particles can be correlated with increasing noise level which is only possible with the basic and most known PSO version. Reason and motivation behind the choice of PSO for this problem are given below, as theoretically compared to other computational intelligence techniques:

I. EEG signal by nature of generation is considered as non-linear function. Hence, to perform the same type of treatment, PSO algorithm having same non-linear and stochastic nature is applied for noise cancellation. Basically, adaptive filtering for linear modeling implemented by gradient based algorithm (conventional adaptive filtering) and for non-linear modeling, adaptive Volterra filters (kind of polynomial extension to the linear adaptive filter) or non-classical adaptive filters that do not rely on linear modeling techniques that are based on evolutionary algorithm (PSO) are the best option due to its simplicity and lack of applied research in the field of EEG processing.

II. Literature review on Neuro fuzzy systems reflects that they are more oriented towards the classification application to mimic nature of human decision making.

III. Ant colony optimization is also a well known optimization method but it is more difficult to theoretically analyze the concept of pheromone accumulation with respect to filtering problem.

A detailed review on behind the choice of PSO for this problem is given in these references and the references therein [41–42].

Compared to genetic algorithm (GA), the advantages of PSO are that it is easy to implement, and there are few parameters to adjust. GA has no guidance or directional concept (Gbest) like PSO, it only depends on crossover and mutation operation which simply define statically and only provide random change in solution. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas, where GA can be applied. Advantages of PSO over GA and simulated annealing (SA) are given in [29,41,43]. The detailed discussion is provided in [29,41,43] and the references therein. Advanced aspects of swarm intelligence and evolutionary algorithms are reviewed in [44–47]. Parameter tuning of swarm intelligence/evolutionary algorithms through methods like qualitative and quantitative parameters as high-level and low-level hierarchy among parameters has been described in [45]. Review capturing the entire horizon of the research on real-parameter evolutionary multimodal optimization and evolutionary optimization in dynamic environments is presented in [46,47]. Non-parametric procedures for pair-wise and multiple comparison for performance analysis was given in [48].

For the detailed investigation of different PSO versions under adaptive noise cancellation for EEG/ERP, work can be carried out with several hybrid and efficient version of PSO. Hybridization of PSO and differential evaluation is discussed in [49], to improve convergence by modifying the spatial characteristics. Adaptation in PSO according to problem is implemented using non-uniform mutation based method and an adaptive sub-gradient method. This leads to version of PSO known as multiple adaptive methods (PSO-MAM) [50]. Dynamic neighborhood learning particle swarm optimizer (DNLPSO) is based on the modification in selection of best historical value from its dynamic neighbor [51]. This helps in avoiding the premature convergence.

Due to popularity of PSO, several hybrid versions like particle swarm optimization with ant colony optimization (PACO) [52] and PSO with back propagation neural network (PSO-BP) [42] have been developed for application in the field of capacitated vehicle
2. Overview of adaptive filters

Adaptive filter is an algorithm that attempts to model the relationship between two signals in an iterative manner. An adaptive filter is defined by four aspects: the signals being processed, the structure that defines the input/output relation, the parameter, which can be iteratively varied to alter the filter's input/output relationship, and the adaptive algorithm, which describes how the parameters are adjusted [23,59–60]. Here, signals are the pair of input and reference signals, and structure is used for implementing digital filter (finite or infinite impulse response FIR or IIR filter). While parameters regulate the input to output mapping or computational relationship according to requirement of the designer, and adaptive algorithms act as the set of rules defined to update the filter coefficients in each iteration, and have close relation with the parameters [23,59–60].

An adaptive filter is more specifically understood as a self-learning digital filter, which is capable of adjusting its coefficients in order to minimize the error function, which is calculated as the distance measurement between the reference or desired signal and the output of the adaptive filter [59–60]. The basic structure of an adaptive filter is illustrated in Fig. 1.

Here, \( x(n) \) is the input signal, and \( d(n) \) is the reference signal or the desired output signal (some noise component are present in it). While \( y(n) \) is the output signal, and the error signal is computed by \( e(n) = d(n) - y(n) \). Adaptive algorithm exploits the error signal produced at every instant to update the adaptive filter coefficient vector \( w(n) \) in every iteration with the help of performance criterion. Commonly, the adaptation process tries to minimize the cost function of error signal, and aspires to approximate output signal as the reference signal in a statistical sense [23,59–60].

This basic structure used is modified according to several practical applications such as system identification, noise cancellation, channel equalization, and signal prediction [59]. In this work, noise canceling structure is used as descendent structure consisting of three noise cancellers. The concept that modifies the adaptive filter as adaptive noise cancellers is depicted in Fig. 2 [23,59].

In Fig. 2, \( s(n) \) is a signal of interest, which is corrupted by a noise component \( q(n) \), and pure version of \( s(n) \) is the desired signal, but cannot be obtained directly in practice. Then, the noisy signal \( s(n) + q(n) \) is employed as a reference signal for the adaptive filter whose input must be correlated version of \( q(n) \), represented as \( \tilde{q}(n) \). Adaptive algorithm adjusts the filter weights \( w(n) \) such that filter input signal \( \tilde{q}(n) \) is translated as output signal \( y(n) \), which is close estimation of \( q(n) \), if \( y(n) \) is equivalent to \( \tilde{q}(n) \). Then, the error signal \( e(n) \) is equivalent to \( s(n) \) in the form of \( \tilde{s}(n) \).

3. Swarm intelligence techniques

3.1. Overview of PSO technique

Particle swarm optimization is a member of wide category of swarm intelligence methods for solving the optimization problems. Originally, PSO technique was developed by Eberhart and Kennedy in 1995 inspired by the social behavior of bird flocking or fish schooling [61]. It is a population based stochastic optimization technique where the swarm is composed of some volume less particles with velocities, each of which represents a feasible solution in the solution space. The principal idea behind PSO is to combine self-experiences with social experiences. It uses a collection of flying particles (updating solutions per iteration) in a search space (current and possible solutions within defined range) as well as tries to move towards a promising area in order to get to a global optimum. Iterations and population are the two major pillars of PSO that are bounded with the power of dimensions and randomness of possible solutions. In every iteration, the global best and local best solutions are selected for updating the present solutions, which act as new population for next iteration [61].

In PSO formulation, there are ‘s’ number of computational agents called ‘particles’, and every particle is represented by \( 2 \times d \) parameters. Every particle flies through the search space to find the best solution with an adaptable velocity that is dynamically modified according to its own flying experience and also to the flying experience of the other particles. Let \( t \) be a time instant. The new position \( X_i(t+1) \) of \( i \)-th particle at time \( t+1 \) is computed by adding to the old position \( X_i(t) \) at time \( t \) and a velocity vector \( V_i(t+1) \). The updating process of particles as well as velocity associated with them is given in [62].

\[
V_i = V_i + U(0,\varphi 1) \times (P_i - X_i) + U(0,\varphi 2) \times (P_g - X_i) \tag{1}
\]

\[
X_i = X_i + V_i \tag{2}
\]
In Eq. (1), $U(0, \varphi_1)$ represents a vector of random numbers uniformly distributed in $[0, \varphi_1]$ and is randomly generated at each iteration and for every particle. The parameters $\varphi_1$ and $\varphi_2$ determine the magnitude of the random forces in direction of personal best $P_i$ and neighborhood best $P_g$. Decision to accept the updated particle value is taken on its fitness value or cost function which is to be minimized or maximized for optimal solution.

As the application of PSO has increased in solving the problems of engineering and other fields, different version of PSO has been developed [61]. Among these, most applied version of PSO is inertia weighted PSO in which, velocity is modified with the use of inertia as

$$ V_i = \text{Inr} \times V_i + U(0, \varphi_1) \times (P_i - X_i) + U(0, \varphi_2) \times (P_g - X_i) $$

where, $\text{Inr}$ is inertia, which decays with time, and is given by

$$ \text{Inr} = \frac{I_{\text{max}} - (I_{\text{max}} - I_{\text{min}})}{N} $$

In the above equation, $I_{\text{max}}$ and $I_{\text{min}}$ represent the upper and lower limit of inertia respectively. $N$ is the total number of iteration while $N$ is the current iteration.

### 3.2. Overview BFO technique

Bacteria foraging optimization (BFO) is an optimization technique based on the foraging nature of real bacteria [63]. Bacteria recognize the path of food based on the gradients of chemicals in their environment and secrete attracting and repelling chemicals into the environment in same path to perceive others on that path. Mechanisms like tumbling and spinning through which bacteria can move are referred as swimming. There are only two basic motions known as tumble and swim that bacteria follow to reach at good environment in same path to perceive others on that path.

These processes of tumble and swim are collectively known as chemotaxis which is followed by each bacterium. Elimination and reproduction are required to kill those bacteria, which are unable to get out from bad nutrient environment (noxious) and generate new bacteria to balance the population [63].

Optimization algorithm derived based on the bacteria survival strategy allows cell-to-cell (bacteria’s) stochastic process. Series of three main processes on a population of simulated cells are performed to mimic real bacteria behavior:

(a) **Chemotaxis**: In this case, the cost of cells is determined based on objective function, and cells move along the manipulated direction and distance (tumble and swim). The majority of the work of the algorithm resides in this process.

(b) **Reproduction**: In this process, only those cells that performed well in their lifetime may contribute to produce the next generation (new solution set).

(c) **Elimination-dispersal**: In this step, the cells are discarded and new random solutions are generated with a low probability to balance the size of population.

Basic terminology used in BFO algorithm with the main working equation (Eq. [5]) responsible for chemotaxis process is given below:

Let $j, k$, and $l$ be the index for the chemotactic step, reproduction step and elimination-dispersal step respectively. The following parameters are set before initializing the loops:

- $P$: dimension of the search space
- $S$: population of bacteria
- $N_L$: number of chemotactic steps
- $N_r$: swimming length (bacterial movements within one chemotactic cycle)
- $Sr$: number of bacteria reproduced
- $N_g$: number of reproduction steps
- $N_{ed}$: number of elimination-dispersal events
- $P_{ed}$: elimination-dispersal probability
- $C(i)$: step size taken in the random direction specified by the tumble.

Above notations are interpreted with iterations as $P(j, k, l) \{ (j, k, l) \mid I \in [1, 2, \ldots, S], l \in [q] \}$ is the position (solution) of each member in the population of the $S$ bacteria at the $j$th chemotactic step, $k$th reproduction step, and $l$th elimination-dispersal event. Here, let $j (i, j, k, l)$ denotes the cost at the location of the $i$-th bacterium. The main and majority work in chemotactic process is modeled in the form of Eq. (5),

$$ \theta(j + 1, k, l) = \theta(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^2(i) \Delta^{\prime}(i)}} $$

where, $\Delta$ indicates a vector in the random direction, the $i$th bacterium position as $\theta_j$ and $j, k$, and $l$ are the index of processes described above [33,63].

### 3.3. Overview of GA optimization

Genetic algorithms are the most famous and earliest search algorithms based on biological evolution. Global optimum solution through GA is conducted by moving from an old population of individuals (solutions or chromosomes) to a new population using genetics-like operators (crossover and mutation). An evaluation or cost function assigns a fitness value to each individual within the population. This fitness value is measured for the quality of an individual. The basic optimization strategy involves simply the selection of highly fit individuals in order to produce better individuals in next generation. A typical genetic algorithm involves four major steps of fitness evaluation, selection, crossover and mutation of a new population [38]. The use of real valued representation in the GA is also implemented to offer a number of advantages in numerical function optimization over binary encoding [39]. Conversion of chromosomes to binary is not needed and efficient floating-point internal computer representations can be used directly which increases the efficiency of the GA. This real valued GA is based on arithmetic crossover and mutation operations [39,64]. Crossover is modeled as

$$ C_{i}^{gen+1} = a \cdot C_{i}^{gen} + (1-a) \cdot C_{j}^{gen} \quad \text{and} \quad C_{j}^{gen+1} = (1-a) \cdot C_{i}^{gen} + a \cdot C_{j}^{gen} $$

where, $C_{i}^{gen}$ and $C_{j}^{gen}$ are two chromosomes, selected randomly for crossover, $C_{i}^{gen+1}$ and $C_{j}^{gen+1}$ are next generation individuals which are a linear combination of their parents, $a$ is a random number in the range of $[0, 1]$. Mutation involves randomly alteration of individuals by generating a random real value, multiplied to make a random change.

As per implementation in this paper, let the population of chromosomes is defined as $C_1, C_2, \ldots, C_r$, where $L'$ is the population size. After genetic operations like crossover and mutation are performed, population size becomes $2 \times L'$. Among the $2 \times L'$ values, selection of $L'$ values is done with the Roulette Wheel selection method. Fitness values after evaluation is normalized and cumulative distribution of the obtained normalized fitness values is obtained. Random number 'r' simulates rotation of the Roulette wheel.
3.4. Overview of ABC technique

Artificial bee colony algorithm (ABC) has been proposed by Karaboga in 2005 for optimizing the solutions of various engineering and science problems [35–37]. ABC algorithm is based on the principle inspired from the foraging behavior of a bee colony. ABC algorithm simulates the real world behavior of bees; and there are three types of artificial bees, known as employed bees, onlooker bees and scout bees. Number of employed bees and onlooker bees are equal in colony. Employed bees search the food in food source area bounded within the search space, after that mutual sharing of food source information with onlooker bees is done. Onlooker bees focus and tend to select possibly best food sources from those discovered by the employed bees, and then further search around the selected food source is performed. Bees whose food sources get abandoned are translated from a few employed bees to scout bees, assigned with new random food source [35–37].

The population initialization with random possible solutions of SN (colony size) is considered as food sources. Let, \( X_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,D}) \) represents the \( i \)th solution in population with \( D \)-dimension, each random solution is generated with minimum to maximum range as Eq. (7),

\[
X_{ij} = X_{\text{min},j} + \text{rand}(0,1) \times (X_{\text{max},j} - X_{\text{min},j}),
\]

(7)

where, \( i = 1, 2, \ldots, SN \), \( j = 1, 2, \ldots, D \), \( X_{\text{min}} \) and \( X_{\text{max}} \) are the lower and upper bounds with \( D \) dimension, respectively. Food sources are randomly assigned to \( SN \) number of employed bees and their fitness value are calculated. Bee initialization phase, each employed bee \( X_i \) generates a new food source \( V_j \) in the neighborhood of its present position by using Eq. (8),

\[
V_{ij} = X_{ij} + \phi_{ij} \times (X_{ij} - X_{ij}),
\]

(8)

where \( k \in [1, 2, \ldots, SN] \) and \( j \in [1, 2, \ldots, D] \) are randomly selected indexes, \( k \) must be different from \( i \). \( \phi_{ij} \) is a random number within the range \([-1, 1]\). When \( V_j \) is obtained, evaluation and comparison of \( X_i \) and \( V_j \) is done on the basis of fitness function. If \( V_j \) is equal to or better than that of \( X_i \), then \( X_i \) will replace \( X_i \) as a new member of the population, otherwise \( X_i \) remains unchanged. Here, a greedy selection is applied between the previous and new candidate solutions.

Selection of food sources by onlooker bees is based on probabilistic selection process. This probabilistic selection is based on the fitness values of solutions in the population; Roulette wheel selection is applied using Eq. (9),

\[
P_i = \frac{f_i}{\sum_{j=1}^{D} f_j},
\]

(9)

where, \( f_i \) is the fitness value of solution \( i \). Obviously, higher the \( f_i \) is, the more probability that the \( i \)th food source is chosen. Onlooker bees evaluate the nectar information taken from all the employed bees and then choose a food source \( X_i \) based on its probability value \( P_i \). After the selection of food source \( X_i \) is done by onlooker, a modification of \( X_i \) is performed by Eq. (8). Food sources that cannot be further enhanced within predefined trials (limit) assumed to be abandoned and bee declared as scout bee. Scout bees generate a food source randomly.

According to the above description of algorithms their pseudo codes (steps in algorithm) are presented in Section 4.3.

4. Methodology for adaptive noise cancellation

Consider the adaptive noise canceler scheme depicted in Fig. 2, \( s(n) \) is the EEG signal (ERP), is corrupted by \( q(n) \) at different noise level, and it is white Gaussian noise (WGN) signal. As mentioned above, \( q(n) \) is the correlated version of noise, generated through Matlab. It is assumed that the corrupted signal \( d(n) \) is composed of the desired \( s(n) \) and noise \( q(n) \), which is additive and not correlated with \( s(n) \). ANC works on estimation of desired output by minimization of error between the desired and actual output and is illustrated in Fig. 3.

In this work, adaptive filters with LMS, NLMS, RLS and PSO are constructed to filter out the ERP. Here, the first order FIR filter was used as order of filter related to its linearity of systems. If length of the filter is more, it provides more linearity. However, EEG signal is non-linear in nature; hence order of filter must be low to reduce linearity in the filter. Another support is provided by theory of non-linear Volterra filters. According to this; filter is constructed with the first term linear equation, the second term quadratic equation, and the third term cubic equations, and so on. Therefore, for more non-linear system, more will be the order of equations. Non-linear Volterra series have been also applied on EEG for artifacts reduction [65], but it becomes more complex in nature as its order is increased. For least complexity, only single term was used which was linear in nature.

For this work, white Gaussian noise is exploited as background noise because WGN is consisting of wide range of frequencies with equally distributed power; and also acts as the worst case for noise contamination. If ANC performs well on white noise reduction, then it can perform well with any type of noise.

4.1. Data specification

As mentioned in the database, different motor/imagery tasks are performed by subjects. While 64-channel EEG was recorded using the BCI2000 system with 160 Hz sampling frequency, 10 seconds length data were used having 1600 samples. Each subject performs 14 experimental runs with 4 tasks. In this case, data of 9 subjects is taken while performing tasks T1 (corresponds to left fist, onset of real or imagined motion) in trial 3, 4, 7, 8, 11, and 12. Only, channel C4 is selected to get the ERP related to hand moves, also known as sensorimotor evoked potential [66–67]. Performing the averaging on total of 54 trials, the pattern obtained is assumed as signature of left fist movement, referred as template ERP. Time at which task is performed is known, so samples after that time were zoomed to visualize the pattern in Fig. 4. White Gaussian noise is mixed with SNR-10 dB, –15 dB, and –20 dB to the signal obtained after the averaging, which is considered as recorded EEG. This data set is taken from the Physionet web database [68–70].

EEG signal by nature of generation is considered as non-linear function. Hence, to perform the same type of treatment, PSO algorithm having same non-linear and stochastic nature is applied for noise cancellation. Basically, adaptive filtering for linear modeling implemented by gradient based algorithm (conventional adaptive filtering) and for non-linear modeling, adaptive Volterra filters (kind of polynomial extension to the linear adaptive filter) or non-classical adaptive filters that do not rely on linear modeling techniques that are based on evolutionary algorithm (PSO) are the best option.

PSO in such application is superior to conventional adaptive filtering because the gradient based algorithm like LMS and RLS produces only one solution per iteration with their updated
equation, so it is more difficult to trace the changes in EEG signals due to its non-stationary nature. Power of guidance and many possible solutions per iteration make PSO more effective in this application as compared to conventional adaptive filtering.

4.2. ANC based on gradient algorithms

As mentioned in Section 1, most aspect of ANC is adaptive algorithm, which is responsible for overall working and quality of the output signal. This subsection describes the most popular gradient based adaptive algorithms known as LMS, NLMS and RLS, which are employed in the field of EEG noise cancellation and ERP extraction.

4.2.1. Least-mean-square (LMS) algorithm

LMS algorithm is the most widely used algorithm in adaptive filtering due to its low computational complexity [3,23,59–60]. In this algorithm, error estimation and weights updating are done through Eqs. (10) and (11), respectively.

\[
e(n) = d(n) - y(n) \tag{10}
\]

\[
w(n+1) = w(n) + 2\mu e(n)x(n) \tag{11}
\]

where, \(x(n)\) is the vector of inputs (signal correlated to noise), \(d(n)\) is denoted as the desired signal, \(y(n)\) is the estimate of the desired signal calculated as \(y(n) = x'(n)x(n)\) (theoretically desired output signal is denoted as \(y(n)\), but practically it is \(x\) the vector having output signal at each iteration) and \(e(n)\) is the estimation error, while \(w(n)\) is the weight vector at nth iteration, and the parameter \(\mu\) is the convergence factor that regulates the convergence rate.

4.2.2. Normalized LMS (N-LMS) algorithm

If the convergence speed of LMS algorithm is increased without carrying out more calculation, then normalized LMS algorithm, a variant of LMS algorithm is exploited. This algorithm utilizes a variable convergence factor aiming at the minimization of instantaneous output error [59]. In this case, error function and weights updating are performed with Eqs. (12) and (13), respectively.

\[
e(n) = d(n) - x'(n)x(n) \tag{12}
\]

\[
w(n+1) = w(n) + \frac{\mu n}{x' + \lambda x(n)x(n)} e(n)x(n) \tag{13}
\]

In Eq. (9), \(K\) is a constant having small value used in order to avoid large step sizes when \(x^2(n) x(n)\) becomes small, and \(\mu_n\) is the range of 0–2.

4.2.3. Recursive least-squares (RLS) algorithm

RLS algorithms are known to pursue fast convergence even when the Eigen value of input signal correlation matrix is high. This algorithm has excellent performance when working in time-varying environments [3,59–60]. All these advantages come at the cost of increased computational complexity and stability problems, which are not as critical in LMS based algorithms. In RLS algorithm, error function and weights updating are performed with Eqs. (14) and (15), respectively.

\[
e(n) = d(n) - x'(n)w(n) \tag{14}
\]

\[
w(n) = S_D(n)P_D(n) \tag{15}
\]

In Eq. (15), \(S_D\) is the deterministic correlation matrix of the input signal, and \(P_D\) is the deterministic cross correlation vector between the input and desired signals. These are calculated using Eqs. (16) and (17), respectively.

\[
S_D = \frac{1}{\lambda} [S_D(n-1) - \frac{S_D(n-1)x(n)x'(n)S_D(n-1)}{\lambda + x'(n)S_D(n-1)x(n)}] \tag{16}
\]

and

\[
P_D(n) = iP_D(n-1) + d(n)x(n) \tag{17}
\]

Initialization of \(S_D(−1)\) is done as \(\delta I\), where \(\delta\) is the inverse of the input signal power and \(I\) is the identity matrix, and \(\lambda\) is the forgetting factor. The output signal is given by Eq. (18).

\[
Y = w'(n)x(n) \tag{18}
\]

4.3. ANC based on evolutionary techniques

In this work, the inertia weighted PSO is employed to calculate optimum weights of ANC to minimize the mean square error (MSE). Due to practical constraints, the entire data (entire input signal) is not available to ANC, and optimum weight vector changes in each iteration. Therefore, a window or samples of the input signals are fetched in every iteration. In this method, the objective function (fitness function) represents an estimate of mean square error (MSE) over input samples used in that iteration [31]. At nth iteration, this estimate of MSE for ith particle is defined as Eq. (19).

\[
J_i(n) = \frac{1}{N} \sum_{j=1}^{N} [\varepsilon_j(n)]^2 \tag{19}
\]
where $N$ is the number of samples of input data and $e_{ji}(n)$ is the $j$th error for the $i$th particle. In PSO technique, it is also important to consider the balancing between global and local search as it is normal assumption that at starting, broader area with larger moves are needed, and for reaching to the best solution, narrow area and smaller moves are required [54,59]. This tradeoff between the global and local search is handled by the inertia weight, which is initialized by upper limit and decreased with time to lower limit. Stopping criteria for PSO is defined as a considerable minimum error based on the cost function (objective function) or as maximum iterations. In this implementation, maximum iteration is taken as the stopping criteria, which is equal to data length.

A summarized description of the steps to be followed in developing the computer algorithm to construct ANC with PSO, BFO, GA and ABC based on their principles discussed are presented in Algorithms 1–4 respectively:

**Algorithm 1.**

**Step 1:** Select the problem space, length of weights vectors (positions); define velocities and inertia within their maximum and minimum limits.

**Step 2:** Initialize particles with random positions (weight vector) and their corresponding velocities within the problem space.

**Step 3:** Check whether the present position is within the problem space or not. If not, then modify positions so as to be inside the problem space. If yes, follow the next step.

**Algorithm 2.**

1. Evaluate fitness value by Eq. (19) for each particle.
2. Compare, present fitness value with the particle’s previous best value, If better ?
3. Assign the present coordinates to $p_{best}$ coordinates.
4. Assign present global minimum to $g_{best}$
5. Update the velocities, particle and Inertia by Eqs. (3), (2), and (4)
6. If, Maximum iteration reached?
7. If yes, Stop

**Fig. 5.** Flow chart of adaptive noise canceller based on PSO technique.
Step 4: Evaluate objective function and find out fitness value by Eq. (19) for each particle.

Step 5: Compare the present fitness value with particles’ previous best value (pbest). If the present fitness value is better, then assign the present fitness value to pbest and assign the present coordinates to pbest coordinates.

Step 6: Find the present global minimum among particle’s best position.

Step 7: If present global minimum is better than gbest, then assign the present global minimum to gbest and assign the present coordinates to gbest coordinates.

Step 8: Update the velocities by Eq. (3) and update each particle to new position by Eq. (2), update inertia by Eq. (4) and return to Step 8.

Step 9: If the present coordinates to gbest, then go to Step 2; otherwise, end.

Flow chart of BFO with basic steps is shown in Fig. 6.

Algorithm 3.

Step 1: Generate the initial population of L chromosomes (weight vectors) within range.

Step 2: Select two chromosomes randomly (L times) as parents for crossover. Perform arithmetic crossover using Eq. (6), to generate two child chromosomes.

Step 3: Mutate all child chromosomes by random change.

Step 4: Evaluate entire 2 × L population, obtain cumulative distribution of normalize the fitness values.

Step 5: Apply Roulette wheel selection to select fittest L chromosomes as next generation.

Step 6: Find the best among chromosome population (final weight vector)

Step 7: Repeat Steps 2–6 until stopping criterion is not satisfied (maximum iteration).

Flow chart of GA with basic steps is shown in Fig. 7.

Algorithm 4.

Step 1: Generate the initial population of SN solutions (weight vectors) within range.

Step 2: Generate new solutions Vi for the employed bees by Eq. (8) and determine the fitness.

Step 3: Apply the greedy selection process for the generated new solutions.

Step 4: Determine the probability values Pi for the solutions Xi by Eq. (9).

Step 5: Generate the new solutions for the onlookers from the solutions Xi with probabilities Pi using roulette wheel selection.

Step 6: Apply the greedy selection process for the onlookers and find out scout bees.

Step 7: Determine the new random solution for the scout bee, find fittest among population (final weight vector).

Step 8: Repeat Steps 2–7 until stopping criterion is not satisfied (maximum iteration).

Flow chart of ABC with basic steps is shown in Fig. 8.

5. Results and discussions

In this section, ANCs developed using gradient based algorithms and proposed PSO technique have been used for event related potential (ERP) detection. To examine the efficacy of proposed PSO algorithm with ANC, three trials with different noise contamination level are done. For fidelity parameter analysis, several parameters such as signal to noise ratio (SNR) in dB, correlation between resultant and template ERP, and mean value are computed, while Kurtosis and Skewness, also termed as statistical measure of shape are determined to examine the quality of ERP as it is more important when classification is performed after filtering. SNR ratio in dB is computed by Eq. (20),

$$\text{SNR}_{\text{dB}} = 10\log_{10}\left(\frac{\text{EEG}_{\text{input}} - \text{EEG}_{\text{output}}}{\text{EEG}_{\text{input}}}ight)^2.$$

(20)
If SNR at output is reached to zero, then perfect reconstruction is achieved. Correlation ($r$) and mean are computed by Eqs. (21) and (22):

$$r = \frac{N\sum X Y - (\sum X)(\sum Y)}{\sqrt{N\sum X^2 - (\sum X)^2} \times N\sum Y^2 - (\sum Y)^2}$$ (21)

and

$$\text{Mean} = \frac{1}{N} \sum_{i=0}^{N-1} X_i$$ (22)

respectively. Kurtosis ($k$) is a measure of whether data are peaked or flat relative to a normal distribution and computed by Eq. (23),

$$k = \frac{E(x-\mu)^4}{\sigma^4}$$ (23)

where $\mu$ is the mean and $\sigma$ is the standard deviation of $x$. The data set with high Kurtosis tends to have a distinct peak near to mean.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>WGN (dB)</th>
<th>SNR</th>
<th>Correlation</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>10</td>
<td>-0.529705025651460</td>
<td>0.805098930479366</td>
<td>-6.38072335670224</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.623603104422147</td>
<td>0.789454981259423</td>
<td>-6.32670931056368</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.529239165567772</td>
<td>0.790342554798739</td>
<td>-6.4712654018497</td>
</tr>
<tr>
<td>RLS</td>
<td>10</td>
<td>-0.288599020534435</td>
<td>0.841655856859675</td>
<td>-6.615805184251312</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.157276027859427</td>
<td>0.432459837492407</td>
<td>-7.0087719318254</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.274515270926343</td>
<td>0.690044020869897</td>
<td>-6.07802954313322</td>
</tr>
<tr>
<td>NLMS</td>
<td>10</td>
<td>-0.174667215298034</td>
<td>0.599178998969432</td>
<td>-6.19814056838491</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.175440765315636</td>
<td>0.529479569451894</td>
<td>-6.0823004709568</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.183352627030900</td>
<td>0.521266934470591</td>
<td>-6.54278075486164</td>
</tr>
<tr>
<td>PSO</td>
<td>10</td>
<td>-0.707725677797578</td>
<td>0.56782584646417</td>
<td>-6.38447215915598</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-1.20242556384678</td>
<td>0.54855653324003</td>
<td>-6.6486248115516</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.33478876968283</td>
<td>0.470410663969425</td>
<td>-6.62017733536979</td>
</tr>
<tr>
<td>BFO</td>
<td>10</td>
<td>-0.44193579701729</td>
<td>0.926618363082569</td>
<td>-6.3136470923726</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.596452349047635</td>
<td>0.912272098727396</td>
<td>-6.5457621404735</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.524218407352458</td>
<td>0.80926591453533</td>
<td>-6.5457621404735</td>
</tr>
<tr>
<td>GA</td>
<td>10</td>
<td>-0.27615310101723</td>
<td>0.783917743099631</td>
<td>-6.0968422550516</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.159883940734720</td>
<td>0.97115590944443</td>
<td>-6.32978678874667</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.20798197204018</td>
<td>0.805075246659080</td>
<td>-6.5657970545736</td>
</tr>
<tr>
<td>ABC</td>
<td>10</td>
<td>-0.029919267075556</td>
<td>0.582406918429626</td>
<td>-8.58591087243210</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-0.082346451778546</td>
<td>0.596560658596612</td>
<td>-5.8934605325759</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-0.10474269489797</td>
<td>0.808396172680060</td>
<td>-6.07019310586807</td>
</tr>
</tbody>
</table>
Table 1 (continued)

<table>
<thead>
<tr>
<th>WGN (dB)</th>
<th>SNR</th>
<th>Correlation</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-0.626727141764385</td>
<td>0.755834546458429</td>
<td>-6.335985453714038</td>
</tr>
<tr>
<td></td>
<td>0.0322726652289304</td>
<td>0.800500716622007</td>
<td>-5.93742104311879</td>
</tr>
<tr>
<td></td>
<td>0.585556454492652</td>
<td>0.790127273849435</td>
<td>-6.354521493526876</td>
</tr>
<tr>
<td>-20</td>
<td>-0.0160242306657579</td>
<td>0.977206817346148</td>
<td>-6.435584813564383</td>
</tr>
<tr>
<td></td>
<td>0.942942308926634</td>
<td>0.816501640137035</td>
<td>-5.92281917539656</td>
</tr>
<tr>
<td></td>
<td>0.469413929214284</td>
<td>0.945365520670307</td>
<td>-6.36983268157323</td>
</tr>
</tbody>
</table>

Fig. 10. Illustration of averaged fidelity parameters, (A) average SNR of three trials corresponding to different noise levels, (B) averages correlation of three trials corresponding to different noise levels, and (C) mean difference between template ERP and average of three trials.
decline rather rapidly, and has heavy tails. While in the case of low kurtosis, it tends to have a flat top near the mean rather than a sharp peak. A uniform distribution would be an extreme case.

Skewness ($s$) is the measure of symmetry and is defined as Eq. (24)

$$s = \frac{E(x-\mu)^3}{\sigma^3}$$  

(24)

A distribution, or data set, is symmetric if it looks same to left and right of the center point.

Several tests were conducted to find out the proper step size ($\mu$), value of forgetting factor ($\lambda$), and particles limits. In case of LMS algorithm, different tests with step sizes 0.005, 0.001, 0.05, and 0.01 were conducted, and it was found that LMS algorithm gives better results with step size 0.05, while NLMS algorithm gives better performance with step size 0.1. RLS algorithm was also tested for different values of forgetting factor such as 0.5, 0.25, 1, 1.25 and 1.5, and inversion matrix coefficient ($\delta$) = 0.1. It was found that RLS algorithm with forgetting factor = 1 gives better performance for the given value of inversion matrix coefficients. The proposed PSO technique was simulated for value 1 to 80 having movement ± 3 for each noise level, velocity limit was set to ± 0.5, and maximum to minimum value of inertia is selected 0.4 and 0.1 respectively, after performing the sensitivity analysis of parameter (velocity and Inertia). Sensitivity of parameters is tested by simulating different combination of inertia and velocity. Variation in inertia is produced by changing maximum and minimum value of inertia ($I_{max}$ = 0.05:0.05:1, in 20 intervals and $I_{max}$ = 0.01:0.01:0.2, in 20 intervals) and velocity ($V$ = 0.02:0.02:1, in 50 intervals). Correlation matrix is plotted in Fig. 7 to observe the effect of various combinations of inertia and velocity with population ($P$) of 50 particles. Finally, it is verified that the selected value of inertia and velocity comes under higher correlation area as illustrated in Fig. 9.

BFO algorithm have several numbers of parameters ($S, N_s, N_c, N_i, N_{ed}, C(i), P_{ed}$) to choose suitable combination of parameters. BFO

Fig. 11. Illustration of finding particles limit in ANC based on PSO technique for different noise contamination level. (A) In case of –10 dB noise. (B) In case of –15 dB noise. (C) In case of –20 dB noise.
is tested with three types of setting \( (N_c = 10, 20, \text{ and } 30), N_e = \{2, 4, \text{ and } 6\}, N_w = \{2, 4, \text{ and } 6\}, N_{ed} = \{2, 4, \text{ and } 6\}, P_{ed} = [0.25, 0.50, \text{ and } 0.75]\), and \( C(i) = [0.01, 0.1, \text{ and } 0.5]\), where \( l = 1, 2, 3, \ldots, S \). S is fixed to 50 to make similarity with PSO population. Among these settings, best values of parameters are selected as \( N_c=20, N_e=4, N_w=4, N_{ed}=4, P_{ed}=0.50 \) and \( C(i)=0.1 \).

GA algorithm is simple to implement with arithmetic crossover and mutation as these are dependent on linear combination of random value weighted parents. Only parameter for arithmetic crossover and mutation is the nature of random generation. Random numbers can be generated with normal or uniform distributions; and in this implementation, uniform distribution is used with population of 50 chromosomes.

The simulation results obtained in each case are tabulated in Table 1 and are graphically shown in Fig. 10. As can be seen from Table 1, the proposed PSO technique gives improved performance as compared to gradient based adaptive algorithms. The average SNR obtained with the proposed ANC based on PSO, BFO, GA and ABC technique is 0.1364, 0.1109, 0.3141, and 0.3095 respectively, while it is obtained with the proposed ANC based on PSO, BFO, GA and ABC compared to gradient based adaptive algorithms. The average SNR/\( C_0 \) levels by simulation of growing range from 1 to 80 having movement of keeping Table 1 and are graphically shown in Fig. 10. A scheme e shows freedom in using with population of 50 chromosomes.

As mentioned above, in signal detection or classification, shape of signal is also an important factor. Therefore, Kurtosis and skewness are computed after the proper range is not known, the algorithm can not give the output ERP and correlation coefficients are plotted to get the intersection point at corresponding parameters limit. From Fig. 11, it is found that the proposed PSO technique yields best performance with particles limits 28 ± 3, 44 ± 3, and 78 ± 3 for noise contamination levels −10 dB, −15 dB, and −20 dB respectively.

Analysis of the proposed method is oriented towards finding of particles range in each of noise level with bounded increment range of particles. SNR and correlation are monitored during increment in range. Establishment of proposed method is done as variable or as non-stationary nature of cost of fitness function. In every iteration, the desired input sample changes through which error is calculated and weight vector (coefficients) are updated by PSO. This method is helpful to figure out how the noise level affects the range of particles (solutions), because until the proper range is not known, the algorithm can not give the desired results. In same fashion, the range for bacteria in case of BFO algorithm is determined, and 25 ± 3, 44 ± 3, 77 ± 3 are found to be the perfect match for noise levels −10 dB, −15 dB and −20 dB respectively. This range selection procedure is also repeated with GA and ABC.

As mentioned above, in signal detection or classification, shape of signal is also an important factor. Therefore, Kurtosis and skewness are computed after filtering of ERP, and their difference from template ERP is listed in Table 2. Fig. 12 illustrates the shape of filtered ERP, in which row (A) is the noisy ERP at −10 dB, −15 dB, and −20 dB of WGN. Plots 10(B), 10(C), 10(D), 10(E), 10(F), 10(G) and 10(H) is the output ERPs through the ANC based on LMS, NLMS, RLS proposed PSO, BFO, GA and ABC technique corresponding to noisy ERP signal.

As can be seen from the simulation results given in Table 2 and from Fig. 10, the shape of filtered ERP is distorted with very large difference as the noise level increases in case of gradient based approaches, while it is less distorted in case of proposed PSO technique. The standard Kurtosis (K) and skewness (S) value of template ERP calculated through Eqs. (19) and (20) are 2.8418 and 0.3181 respectively. The average difference in Kurtosis value (\( K_{ave} \)) obtained with LMS, NLMS, RLS, PSO, BFO, GA and ABC techniques is −23.372, −5.0643, −33.7996, −0.0304, 0.1653, −0.1140, and 0.0634 respectively. The average difference in skewness value (\( S_{ave} \)) obtained with gradient based adaptive algorithm, PSO, BFO, GA and ABC technique is −2.06623, −0.3411, −2.58373, 0.038367, 0.723367, 0.0683 and 0.0372.

Comparison among different evolutionary techniques can be done based on the basic of their complexity of execution process, as it is clear from the three major process of BFO, which are executed on every bacterium in population that makes BFO execution much more complicated and huge. Single execution of BFO needs at least 3-dimensional array operations for each bacterium (pop × elimination × reproduction × chemotactic). BFO algorithm also fails to retain the original amplitude of signal, and therefore, mean value difference increases which is not acceptable. GA needs two major steps to be performed on each chromosome, crossover and mutation. During crossover process, population becomes double which increases the overhead in mutation. ABC also has additional process of finding scout bees and regenerating their food sources. On the other hand, PSO only needs at least 1-dimensional array operation (pop × update of particles × velocity) which is simple and much faster. During simulations, it is observed that GA implementation as adaptive noise canceler is not stable enough (if noise property slightly changes, it affects performance with huge difference). Computational complexity can also be measured as searching complexity and time complexity [37]. Searching complexity in finding neighborhoods solutions depending on their quality. Since the numbers of particles, bacteria, chromosomes and total bees (employ and onlooker) are equal to P, S, L and SN respectively with maximum iteration (maximum cycles number MCN). In each cycle P, S, L and SN searches are conducted, when the MCN is over, total \( P \times MCN, S \times MCN, L \times MCN \) and SN × MCN searches are carried out which is estimation of their search complexity.

Parameter setting or tuning method with qualitative approach to find suitable range of solutions was adapted by adding a new parameter “C” to control the range “R”. By taking a new constant parameter “C” to improve the SNR and correlation in presented study, range declaration becomes a qualitative parameter.

### Table 2: Shape measures of ERP.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>WN (dB)</th>
<th>( K_{ave} )</th>
<th>( S_{ave} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>−10</td>
<td>−0.5023</td>
<td>0.0536</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>−8.0179</td>
<td>−5.08</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>−61.3958</td>
<td>−5.7443</td>
</tr>
<tr>
<td>NLMS</td>
<td>−10</td>
<td>−3.9716</td>
<td>−0.4932</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>−5.9933</td>
<td>−0.7083</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>−5.2504</td>
<td>−0.2718</td>
</tr>
<tr>
<td>RLS</td>
<td>−10</td>
<td>−0.0468</td>
<td>−0.1113</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>−0.305</td>
<td>−0.1275</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>−101.047</td>
<td>−7.6124</td>
</tr>
<tr>
<td>PSO</td>
<td>−10</td>
<td>−0.0005</td>
<td>0.503</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>−0.0404</td>
<td>0.2215</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>−0.0503</td>
<td>0.0433</td>
</tr>
<tr>
<td>BFO</td>
<td>−10</td>
<td>0.0825</td>
<td>0.7760</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>0.2065</td>
<td>0.7460</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>0.2070</td>
<td>0.6481</td>
</tr>
<tr>
<td>GA</td>
<td>−10</td>
<td>−0.0203</td>
<td>0.0112</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>0.0156</td>
<td>0.1239</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>−0.3374</td>
<td>0.0698</td>
</tr>
<tr>
<td>ABC</td>
<td>−10</td>
<td>−0.0367</td>
<td>−0.0028</td>
</tr>
<tr>
<td></td>
<td>−15</td>
<td>0.1769</td>
<td>0.0686</td>
</tr>
<tr>
<td></td>
<td>−20</td>
<td>0.0500</td>
<td>0.0457</td>
</tr>
</tbody>
</table>
For comparing among algorithm Wilcoxon signed-ranks test is adapted with respect to average fidelity parameter obtained as results in case of –10 db and –20 db SNR to provide ranking for each algorithm. This test provides ranks for algorithm on the basis of their sensitivity with respect to noise level (considering results of lowest and highest noise SNR level used in the paper). Tables 3 and 4 are listed with signed-ranks test calculations.
In Table 3 negative rank appears because SNR and MD have their optimum values as minimum as possible (zero), which in case of correlation value optimizes as maximum (up to 1). Table 4 concludes final ranking based on absolute averaging of SNR, correlation and mean difference. On the basis of signed rank test, ranking of algorithm is arranged in descending order as PSO, BFO, correlation and mean difference. On the basis of signed rank test, ABC algorithm has been proposed for ERP filtering from noisy EEG signal. Simulation results clearly show the key advantageous features of the ANC based on PSO and other evolutionary technique over others in the field of biomedical signal processing. It is evident that the proposed ANC yields improved fidelity parameters as compared to gradient based ANCs. Shape related measures for detection of ERP is also found more closely to the template ERP in case of proposed PSO and other evolutionary technique.

References


Table 3
Signed Rank test for sensitivity analysis.

<table>
<thead>
<tr>
<th>SNR based signed rank test</th>
<th>SNR</th>
<th>Corr</th>
<th>MD</th>
<th>Signed rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5608</td>
<td>0.1778</td>
<td>0.3830</td>
<td>4</td>
<td>LMS</td>
</tr>
<tr>
<td>0.7691</td>
<td>0.3461</td>
<td>0.4229</td>
<td>6</td>
<td>NLMS</td>
</tr>
<tr>
<td>0.4908</td>
<td>0.0723</td>
<td>0.4185</td>
<td>5</td>
<td>RLS</td>
</tr>
<tr>
<td>0.2072</td>
<td>0.0475</td>
<td>0.0597</td>
<td>1</td>
<td>PSO</td>
</tr>
<tr>
<td>0.0324</td>
<td>0.2036</td>
<td>–0.1712</td>
<td>3</td>
<td>BFO</td>
</tr>
<tr>
<td>0.2510</td>
<td>0.3981</td>
<td>–0.1471</td>
<td>2</td>
<td>GA</td>
</tr>
<tr>
<td>0.0376</td>
<td>0.4761</td>
<td>–0.4385</td>
<td>7</td>
<td>ABC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Correlation based signed rank test</th>
<th>Corr</th>
<th>Corr</th>
<th>Corr</th>
<th>Signed rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7949</td>
<td>0.5499</td>
<td>0.2449</td>
<td>6</td>
<td>LMS</td>
</tr>
<tr>
<td>0.5289</td>
<td>0.3396</td>
<td>0.1893</td>
<td>5</td>
<td>NLMS</td>
</tr>
<tr>
<td>0.9195</td>
<td>0.6624</td>
<td>0.2571</td>
<td>7</td>
<td>RLS</td>
</tr>
<tr>
<td>0.9905</td>
<td>0.9564</td>
<td>0.0341</td>
<td>1</td>
<td>PSO</td>
</tr>
<tr>
<td>0.8893</td>
<td>0.8368</td>
<td>0.0525</td>
<td>2</td>
<td>BFO</td>
</tr>
<tr>
<td>0.9897</td>
<td>0.8939</td>
<td>0.1048</td>
<td>4</td>
<td>GA</td>
</tr>
<tr>
<td>0.9917</td>
<td>0.9130</td>
<td>0.0787</td>
<td>3</td>
<td>ABC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MD based signed rank test</th>
<th>MD</th>
<th>MD</th>
<th>MD</th>
<th>Signed rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0052</td>
<td>0.4761</td>
<td>–0.4708</td>
<td>7</td>
<td>LMS</td>
</tr>
<tr>
<td>0.0968</td>
<td>0.2871</td>
<td>–0.1903</td>
<td>5</td>
<td>NLMS</td>
</tr>
<tr>
<td>0.0673</td>
<td>0.4479</td>
<td>–0.3806</td>
<td>6</td>
<td>RLS</td>
</tr>
<tr>
<td>0.0353</td>
<td>0.0051</td>
<td>0.0302</td>
<td>1</td>
<td>PSO</td>
</tr>
<tr>
<td>1.6577</td>
<td>1.7336</td>
<td>–0.0758</td>
<td>2</td>
<td>BFO</td>
</tr>
<tr>
<td>0.0201</td>
<td>0.1233</td>
<td>–0.1032</td>
<td>3</td>
<td>GA</td>
</tr>
<tr>
<td>0.0034</td>
<td>0.1451</td>
<td>–0.1417</td>
<td>4</td>
<td>ABC</td>
</tr>
</tbody>
</table>

Table 4
Final rank based on fidelity parameters.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SNR</th>
<th>CORR</th>
<th>MD</th>
<th>Average rank</th>
<th>Final rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>5.6</td>
<td>6</td>
</tr>
<tr>
<td>NLMS</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5.3</td>
<td>5</td>
</tr>
<tr>
<td>RLS</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>6.6</td>
<td>7</td>
</tr>
<tr>
<td>PSO</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BFO</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2.3</td>
<td>3</td>
</tr>
<tr>
<td>GA</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>ABC</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>4.6</td>
<td>4</td>
</tr>
</tbody>
</table>

6. Conclusions

In this work, several adaptive noise cancellation techniques based on gradient methods such as LMS, NLMS and RLS algorithms have been examined, and an improved ANC based on PSO, BFO, GA and ABC algorithm has been proposed for ERP filtering from noisy signal. Simulation results clearly show the key advantageous features of the ANC based on PSO and other evolutionary technique over others in the field of biomedical signal processing. It is evident that the proposed ANC yields improved fidelity parameters as compared to gradient based ANCs. Shape related measures for detection of ERP is also found more closely to the template ERP in case of proposed PSO and other evolutionary technique.


