Modified Firefly Optimization for IIR System Identification

Mehrnoosh Shafaati, Hamed Mojallali

Electrical Engineering Department, Faculty of Engineering, University of Guilan
Rasht, Iran, (e-mails: mehr.shafaati@gmail.com, mojallali@guilan.ac.ir)}

Abstract: Because of the nonlinear and recursive nature of the physical systems, system identification is a challenging and complex optimization problem. Infinite impulse response (IIR) and nonlinear adaptive systems are widely used in modeling real-world systems. IIR models due to their reduced number of parameters and better performance are preferred over finite impulse response (FIR) systems. In the past few decades, meta-heuristic optimization algorithms have been an active area of research for solving complex optimization problems. In the present paper a modified version of a recently introduced population-based firefly algorithm (MFA) is used to develop the learning rule for identification of three benchmark IIR and nonlinear plants. MFA's performance is compared with standard firefly algorithm (FA), GA and three versions of PSO. The results demonstrate that MFA is superior in identifying dynamical systems.

Keywords: System Identification, IIR Structure, Adaptive filtering, Firefly optimization.

1. INTRODUCTION

Adaptive filtering is one of the most important issues in the field of digital signal processing and, has been extensively applied in areas like noise cancellation, channel equalization, linear prediction, system identification and etc. In the latter the adaptive filter is used to provide a model that represents the best fit to an unknown system. Selecting the correct order and estimating the parameters of the adaptive filter is a fundamental issue in system identification. In practice, most of the systems are nonlinear to some degree and recursive in nature, hence nonlinear system identification has attracted much attention in the field of science and engineering (Panda et al., 2007). While theory and design of linear adaptive filters based on FIR structures is well developed and widely applied in practice, the same is not true for more general classes of adaptive systems such as linear infinite impulse response adaptive filters (IIR) and nonlinear adaptive systems (Krusienski and Jenkins, 2005). In the last decades, substantial effort has been done to use IIR adaptive filtering techniques as an alternative to adaptive FIR filters. The main advantages of IIR filters are that, firstly, due to the pole-zero structure they are more suitable to model physical systems and hence, the problem of system identification can also be viewed as a problem of adaptive IIR filtering (Venayagamoorthy, 2010). Secondly, they require less number of coefficients than FIR filters to achieve the same level of performance, so they involve less computational burden compared to FIR structures (Netto et al., 1995). Alternatively, with same number of coefficients, an adaptive IIR filter performs better than an adaptive FIR filter (panda et al., 2011). However, these good characteristics come along with some possible drawbacks: both linear IIR structures and nonlinear structures tend to produce multi-modal error surfaces with respect to the parameters for which conventional derivative based learning algorithms may fail to converge to the global optima. These algorithms such as least mean square (LMS) try to find the global minimum of the search space by moving toward the direction of negative gradient and hence can easily get trapped in local minima (Widrow et al., 1976; Krusienski and Jenkins, 2005). Furthermore, slow convergence and instability of the algorithms are two major concerns in IIR filter structures. IIR Filters especially those with high order must be carefully monitored for bounded input bounded output (BIBO) stability since their poles can easily move outside the unit circle during the learning adaption. In order to alleviate these problems several new structures and algorithms have been proposed (Regalia, 1992; David, 1981). Traditionally least square techniques have been well studied for the identification of static and linear systems (Widrow et al., 1976). For nonlinear system identification, different algorithms have been used in the past including neural networks (Hongway and Yanchun, 2005; Brouwn and Krijsman, 1994) and gradient based search techniques. Also, a number of adaptive system identification techniques have been reported in the literature (Ljung, 1987; Ljung and Gunnarsson, 1990; Astriim and eykhoff, 1971). (Shynk 1989) presented a tutorial which provides an overview of methods, filter structures and recursive algorithms used in adaptive IIR filtering. Another tutorial was brought forward by (Johnson 1984) on adaptive IIR filtering. This work was the first attempt to unify the concepts of adaptive control and adaptive filtering. A general framework of adaptive IIR filtering algorithms for system identification is reported in (Netto et al. 1995). Population based stochastic optimization algorithms like GA and PSO have been also applied to the problem of IIR and nonlinear system identification. Unlike gradient based techniques, they are not dependent upon the filter structure. Due to this fact and their evolutionary searching ability, they are capable of globally optimizing any class of adaptive filter structures by assigning their resultant estimated parameters to filter coefficients, neural network weights or any other possible parameter of the unknown system model (even the exponents of polynomial terms) (Lee and El-Sharkawi, 2002). In (Kumon, et al. 2000; Kristinson and Dumont, 1992) GA is used to improve the identification performance of nonlinear systems. GA based methods have been combined with many different classic methods and as a result, hybrid evolutionary algorithms with faster convergence and better local search ability have been created. The combination of GA with LMS, recursive least square (RLS) and simulated annealing is presented in (Warwick et al. 1999; NG et al. 1996), respectively. Application of PSO in the system identification can be found in (Panda et al. 2007; Lee et al. 2006; Shen and Zeng 2007). Recently (Luitel and Venayagamoorthy 2010) presented PSO with quantum infusion for adaptive IIR identification. In (Majhi and Panda 2010) identification of complex nonlinear dynamic plants using PSO and Bacterial foraging Optimization (BFO) is proposed. Several other investigators utilized meta-heuristic algorithms for system identification and IIR filter design e.g. artificial bee colony (Karaboga, 2009), seeker optimization (Dai et al., 2010), cat swarm optimization (Panda et al. 2011)

The goal of this paper is to introduce a modified firefly algorithm which shows enhanced performance in solving the complex optimization task of adaptive system identification.

By gradually reducing the randomness we tried to enhance the exploitation quality of the algorithm and by adding a social dimension to each firefly (global best), the chance of fireflies for global exploration was increased. In order to verify the validity of the above statements, the proposed algorithm was used as an adaptive learning rule for parameter identification of a benchmark IIR system, a nonlinear IIR system and a typical nonlinear communication channel. The results prove that FA and MFA convergence behaviour in terms of speed and accuracy is much better than standard GA and PSO. Most important of all is that, MFA outperforms FA in nearly all cases. Also, it can be seen that MFA is more robust to the increase in dimension of the problem than the other algorithms used in this paper.

The rest of this paper is organized as follows. Section 2 considers a brief introduction to IIR system identification. Section 3 discusses the basic firefly algorithm and our proposed approach in detail. The proposed MFA based system identification method is explained in Section 4. Simulation results and discussions are given in Section 5. The sensitivity analysis of key parameters of FA and MFA are presented in Section 6. The paper ends with conclusions in Section 7.

2. IIR SYSTEM IDENTIFICATION

The problem of determining a mathematical model for an unknown system by monitoring its input-output data is known as system identification. This mathematical model then could be used for control system design purposes. The block diagram of the adaptive system identification is shown in Fig. 1. When the plant behaviour is not known, it can be modelled using an adaptive filter and then its identification task is done using adaptive algorithms. The adaptive algorithm tries to tune the adaptive filter coefficients such that the error between the output of the unknown system and the estimated output is minimized. In other words, it matches the adaptive model's input-output relationship as close as possible to that of the unknown system which is the goal of system identification. As said, IIR systems, having both poles and zeros give us a better representation of the real-world systems over FIR structures. An IIR system is described as:

$$Y(z) = H(z) \;\; U(z) \tag{1}$$

where,

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_m z^{-m}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}} \tag{2}$$

is the IIR transfer function, $Y(z)$ is the z-transform of the output $Y(n)$, and $U(z)$ denotes the z-transform of the input $u(n)$. a_i , $i = 0,1,2,...,\ m$ and b_i , $i = 0,1,2,...,\ n$ are the feed-forward and feed-back coefficient of the IIR system, respectively. The IIR filter can also be formulated as a difference equation:

$$y(n) = \sum_{k=1}^{n} b_k\, y(n-k) + \sum_{k=0}^{m} a_k\, u(n-k) \tag{3}$$

As illustrated in Fig. 1, $v(n)$ is the additive noise in the output of the system. Combining $v(n)$ and $y(n)$ we get the overall output of the system $d(n)$. Additionally, for the same set of inputs, the adaptive IIR filter block produces $\hat{y}(n)$. The estimated transfer function can be represented as:

$$\hat{H}(z) = \frac{\hat{a}_0 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + ... + \hat{a}_m z^{-m}}{\hat{b}_0 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + ... + \hat{b}_n z^{-n}} \tag{4}$$

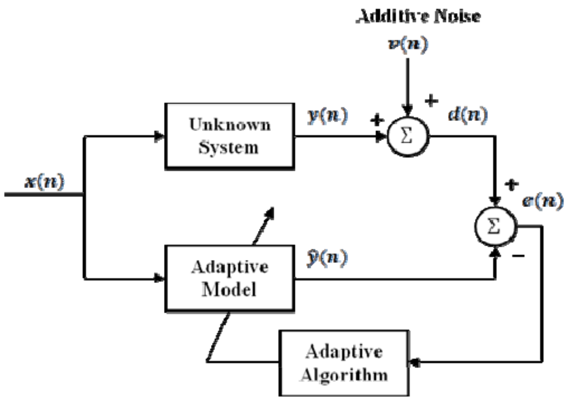

Fig. 1. Block diagram of adaptive system identification.

where, $\hat{a}_i$ and $\hat{b}i$ signify the approximated coefficients of the IIR system. In other words, the transfer function of the actual system is to be identified using the transfer function of the adaptive filter. The difference between $d(n)$ and $\hat{y}(n)$ produces the input to the adaptive algorithm .The adaptive algorithm uses this residual to adjust the parameters of the IIR system. It can be recognized from the figure that:

$$d(n) = y(n) + v(n) \tag{5}$$

$$e(n) = d(n) - \hat{y}(n) \tag{6}$$

The cost function to be minimized by the adaptive identification algorithm is the mean square error between $d(n)$ and $\hat{y}(n)$ given by:

$$J = E\left[\left(d(n) - \hat{y}(n)\right)^2\right] \cong \frac{1}{N}\sum_{n=1}^{N} e^2(n) \qquad (7)$$

where, N denotes the number of input samples and $E(.)$ is the statistical expectation operator. The optimization algorithms employed in this paper, search the solution space to find those values of parameters which contribute to the minimization of (7).

3. FIREFLY OPTIMIZATION

Firefly algorithm is a novel meta-heuristic optimization algorithm which was first developed by Xin-She Yang at Cambridge University in 2008. As a nature-inspired algorithm it imitates the Social behaviour of fireflies and the phenomenon of bioluminescent communication. Since its emergence in (Yang 2008), it has been successfully applied to many engineering optimization problems. Digital image compression, multi-objective and non-convex economic dispatching are two real-world applications of firefly algorithm (Horng and Jiang, 2010; Apostolopoulos and Vlachos, 2011; Yang et al. 2012). Recent studies show that it is a powerful tool for discrete time problems. Contrary to the most meta-heuristic algorithms such as Ant Colony firefly can efficiently deal with stochastic test functions (Yang, 2010).

3.1 Behaviour of fireflies

The flashing light of fireflies which is produced by a bioluminescence process constitutes a signalling system among them for attracting mating partners or potential preys. It is interesting to know that there are about two thousand species of fireflies around the world. Each, has its own pattern of flashing. As we know, the light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \propto 1/r^2$. Furthermore, the air absorbs light. These two combined factors make most fireflies visual to a limit distance.

3.2 Firefly algorithm

The flashing light can be formulated in such a way that is associated with the objective function. We will first discuss the basic formulation of the firefly algorithm (FA) and then a modified version will be proposed in order to improve the performance of the algorithm.

By idealizing some of the flashing characteristics of fireflies, firefly-inspired algorithm was presented by Xin-She Yang. This algorithm uses the following three idealized rules: 1) all fireflies are unisex which means that they are attracted to each other regardless of their sex; 2) the degree of attractiveness of two fireflies is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move toward the brighter one and more brightness corresponds to less distance between two fireflies. If there is no brighter one than a particular firefly, it will move randomly; 3) the brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness can simply be proportional to the value of the objective function. Based on these three rules, the basic steps of firefly algorithm (FA) can be summarized as the pseudo code shown in Fig. 2.

Firefly Algorithm

```
Generate initial population of fireflies x_i (i = 1,2, ..., n)
Light intensity I_i at x_i is determined by f(x_i)
Define light absorption coefficient γ
while (t < MaxGeneration)
for  i = 1:n  all n fireflies
    for  j = 1:n  all fireflies (inner loop)
    if(I_i > I_j ), Move firefly i toward j ; end if
       Vary attractiveness with distance r via exp [−γr]
       Evaluate new solutions
     end for j
end for i
Rank the fireflies and find the current global best g_best
end while
Postprocessing & visualization
```

Fig. 2. Pseudo code of the firefly algorithm.

3.3 Light Intensity and Attractiveness

The variation of light intensity and formulation of the attractiveness are two important issues in the firefly algorithm. The attractiveness β is proportional; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus it will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness will also vary with the degree of absorption. The combined effect of both inverse square law and absorption can be approximated as the following Gaussian form as (8). Hence the attractiveness function $\beta(r)$ can be any monotonically decreasing function such as the following generalized form:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \qquad (m \geq 1) \qquad (8)$$

Where r is the distance between two fireflies, β_0 is the attractiveness at $r = 0$, and γ is a fixed light absorption coefficient which can be used as a typical initial value. In theory, $\gamma \in [0, \infty)$ but in practice γ is determined by the characteristic length of the system to be optimized. In most applications it typically varies from 0.1 to 1. Characteristic distance Γ is the distance over which the attractiveness changes significantly. A typical scale Γ should be associated with the scale concerned in our optimization problem. For a given length scale, the parameter γ can be chosen according to:

$$\gamma = \frac{1}{\Gamma^m} \qquad (9)$$

The distance between any two fireflies i and j at x_i and x_j, respectively is the Euclidean distance as follows:

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2} \tag{10}$$

Where $x_{i,k}$ is the k 'th component of the i 'th firefly (x_i).

The movement of a firefly i that is attracted to another more attractive (brighter) firefly j, is determined by the following equation which shows the relation between the new position of the firefly i (x_i') and its old position (x_i):

$$x_i' = x_i + \beta_0 e^{-\gamma r^2} \left(x_j - x_i \right) + \alpha \varepsilon_i \tag{11}$$

where, the second term is due to the attraction. The third term is randomization, with $\alpha \in [0,1]$ being the randomization parameter, and ε_i a vector of numbers drawn from a Gaussian distribution or uniform distribution.

3.4 Modified Firefly Algorithm (MFA)

The basic firefly algorithm is very efficient. It is suitable for parallel implementation because different fireflies can work almost independently. Furthermore, the fireflies aggregate more closely around each optimum; so that it is possible to adjust its parameters such that it can outperform the basic PSO (Yang, 2010). But we can see from the simulation results that the solutions are still changing as the optima are approaching. To improve the solution quality we tried to reduce the randomness so as the algorithm could converge to the optimum more quickly. We defined the randomization parameter α as the following form (12), as u can see α decreases gradually as the optima are approaching.

$$\alpha = \alpha_\infty + \left(\alpha_0 - \alpha_\infty \right) \; e^{-t} \tag{12}$$

where, $t \in [0, t_{\max}]$ is the pseudo time for simulation and $t_{\max}$ is the maximum number of generations. α_0 is the initial randomization parameter while is the final value. In addition we added an extra term $\lambda \varepsilon_i (x_i - g_{best})$ to the updating formula. In the simple version of the firefly algorithm (FA), the current global best g_{best}, is only used to decode the final best solutions. The modified updating formula is shown in (13).

$$x_i' = x_i + \beta_0 e^{-\gamma r^2} \left(x_j - x_i \right) + \alpha \varepsilon_i + \lambda \varepsilon_i \left(x_i - g_{best} \right) \tag{13}$$

λ is a parameter similar to α Since all PSO-based algorithms uses this term, this modification can be also viewed as a combination of FA with PSO.

4. MFA BASES SYSTEM IDENTIFICATION

The parameter updating rule can be summarized in the following steps:

1) The coefficients of the model are initially chosen from a random population of M fireflies in search space. For example, in MFA, each firefly corresponds to P number of parameters and each parameter refers to one coefficient of the adaptive filter.

2) Optimization algorithm parameters are defined in this step. For example β_0, γ and λ in MFA . Max.Iteration is set to 1000.

3) For a set of given input $\{x_i\}_{i=1}^{N}$, N =1000, with a known distribution, the output set is computed as: $Y = \left[y(1) \; y(2) ... \; y(N) \right]^T$ .Then the measurement noise is added to the output of the plant and the resultant signal serves as the desired signal.

4) Each of the input samples is also passed through the model using each firefly as model parameters. In this way M sets of N estimated outputs are obtained.

$\hat{Y}_k = \left[\hat{y}_k(1) \; \hat{y}_k(2) ... \; \hat{y}_k(N) \right]^T$, $k = 1,2,...M$

5) In this step, each of the desired output is compared with corresponding estimated output and k errors are produced. The mean square error corresponding to kth firefly is determined by using the relation:

6) $$MSE(k) = \frac{1}{N} \left[\left(Y - \hat{Y}_k \right)^T \left(Y - \hat{Y} \right) \right] \tag{14}$$

Since the objective is to minimize $MSE(k)$, GA, PSO based, FA and MFA minimization methods are used.

7) In GA the mutation and crossover are carried out to evolve the solutions.

8) In PSO based algorithms velocity and position of each particle is updated using (15) and (16).

9)

$$V_i(t+1) = wV_i(t) + c_1 \times rand(.) \times \left(X_i - pbest_i \right) + c_2 \times rand(.) \times \left(X_i - Gbest \right) \tag{15}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{16}$$

where, V_i is the velocity vector of particle i . X_i and $pbest_i$ represents the position and the best previous position of the ith firefly, respectively. $Gbest$ shows the best position among all particles. c_1 and c_2 are called acceleration constants which control the maximum step size and w is known as inertia weight that controls the impact of previous velocity of particles on its current one.

10) In FA the position of each firefly is updated using (11).

11) In MFA each firefly moves according to (13).

12) In each generation the minimum MSE (MMSE) among all the fireflies, is obtained and plotted against number of iterations to show the learning characteristics.

13) The learning process will stop when a predefined MSE level or the maximum number of iterations is reached. The gene, the particle or the firefly that corresponds to the least amount of fitness (best attainable match between Y and $\hat{Y}_k$ in the sense of MSE) shows the estimated parameters.

#

5. SIMULATION RESULTS

In this section, three benchmark systems are considered to confirm the efficiency of our proposed method in parameter identification of IIR and nonlinear systems. For the sake of a more comprehensive comparison, in addition to FA and MFA simulations are done using three versions of PSO, and a standard version of GA, as well. GA and PSO are two of the most known evolutionary optimization algorithms and have been an active area of research during the past decade (Kennedy and Eberhart, 1995; Holland, 1975; Goldberg, 1989). Two performance criterion mean square error (MSE) and mean square deviation (MSD) are used to compare the performance of the aforementioned approaches. MSE is the mean square error between the desired signal and adaptive filter's output as given in (7) in section 2 and MSD corresponds to mean square deviation between the actual coefficients and the estimated coefficients which is defined as:

$$MSD = \frac{1}{p}\sum_{i=0}^{p-1}\left(\Phi(i) - \hat{\Phi}(i)\right)^2 \tag{17}$$

where, Φ is the desired parameter vector, $\hat{\Phi}$ is the estimated parameter vector and P is the total number of parameters to be estimated.

In order to ensure the validity of the results, each experiment is repeated in 20 consecutive trials and the resultant average for the best MSE's of each run, is plotted in corresponding figures. The (min, mean, and standard deviation) values for MSE and MSD, and the computational time, over 20 runs are given in table1 to 3 respectively, at the end of this section. Each simulation study is carried out in MATLAB v.7.5 on a personal computer with Intel® Core™ 2 Duo CPU 2 GHz and 1 GB of RAM. In all cases the population size is set to 50 and random numbers take values between 0 and 1. Unless specified otherwise, the input data is a Gaussian white noise with zero mean and unit variance and the output data is contaminated with a Gaussian random noise with zero mean and a variance of 0.001 which corresponds to SNR value of 60dB. The simulation parameters for the six algorithms are as the following: for PSO with linear adaptive inertia weight (PSOLW): acceleration constants c_1 and c_2 are set to 2, inertia weight is linearly decreased from 0.9 to 0.4 with increasing iterations and the maximum velocity $v_{\max}$ is set at 15% of the dynamic range of the variable on each dimension (Shi and Eberhart, 1999). In PSOW the adaptive inertia weight factor w, is determined as follow:

$$w = \begin{cases} w_{\min} + \dfrac{(w_{\max} - w_{\min})(f - f_{\min})}{f_{avg} - f_{\min}} & f \le f_{avg} \\ w_{\max} & f \succ f_{avg} \end{cases} \tag{18}$$

where, $w_{\min}$ and $w_{\max}$ denote the maximum and minimum of w, respectively. f is the current objective function value of the particle, f_{avg} and $f_{\min}$ are the average and minimum objective values of all particles, respectively. In this approach, w is varied based on the objective function value so that particles with low objective values can be protected while particles with objective value greater than average will be disrupted. Hence, it provides a good way to maintain population diversity and to sustain good convergence capacity. The simulation parameters for PSOW are: $w_{\max} = 1.2$, $w_{\min} = 0.2$, $c_1 = c_2 = 2$, and $v_{\max}$ is limited to the 15% of the search space (Liu et al., 2005). SPSO2011 is a standard version of PSO and has a constant value of the inertia weight.

Parameter values for this algorithm are available online and the address is given through references. In the GA algorithm the bit number per dimension is set to 16, mutation probability is 0.1, and crossover step is of single point type with a probability of 0.7. The parameters of firefly algorithms are given next. For FA: γ =2 based on the scale of search space. $\alpha = 0.1$, , $\beta_0 = 1$, and ε_i =rand-1/2, where rand is a random number generator uniformly distributed in [0,1]. And for MFA: $\beta_0 = 1$, $\gamma = 1$, $\lambda = 0.1$ and α exponentially decreases from 0.25 to 0.05. It should be noted that for IIR cases we considered two models, one with the same order as the actual system , and the other with a reduced order IIR structure. As the number of coefficients of the model decreases, the degree of freedom reduces and multimodality of the error surface increases. Thus, it imposes a challenge to the optimization problem.

5.1 Example 1

The transfer function of the plant is given by, (Shynk,1989a)

$$H(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314_1 z^{-1} + 0.25z^{-2}} \tag{19}$$

Two IIR structures are used for the identification purposes which are given below:

$$H(z) = \frac{a_0 + a_1 z^{-1}}{b_0 - b_1 z^{-1} - b_2 z^{-2}} \tag{20a}$$

$$H(z) = \frac{a_0}{b_0 - b_1 z^{-1}} \tag{20b}$$

Simulation results related to the actual order model (20a) and reduced order model (20b) are given in Figs. 4a and 4b, respectively. It is clear from the figures that the utilization of MFA has resulted in greater estimation accuracy than other algorithms. In Fig.4a GA and SPSO2011 have converged to a suboptimal solution since their final error level is much higher than other algorithms. Table 1 gives the attained MSE values over 20 simulations regarding the actual order and reduced order model, respectively. The MSD for actual order model is shown in Table 2. As it is observed, the best values are given in bold. The respective computational time of the algorithms related to the actual and reduced model are listed in Table 3, as well. The results demonstrate that, MFA provides the best results in terms of MSE and MSD.

5.2 Example 2

Consider the following nonlinear IIR system taken from (Bai et al. 2007).

$$y = f\left(y(k-1), u(k-1)\right) = 0.2y(k-1) - 0.5y(k-1)^2 u(k-1) + u(k-1)^3 \quad (21)$$

System (12) is identified using the following structures:

$$y = ay(k-1) + by(k-1)^2 u(k-1) + cu(k-1)^3 \quad (22a)$$

$$\frac{y(k)}{u(k)} = H(z) = \frac{a_0 + a_1 z^{-1}}{b_0 - b_1 z^{-1} - b_2 z^{-2}} \quad (22b)$$

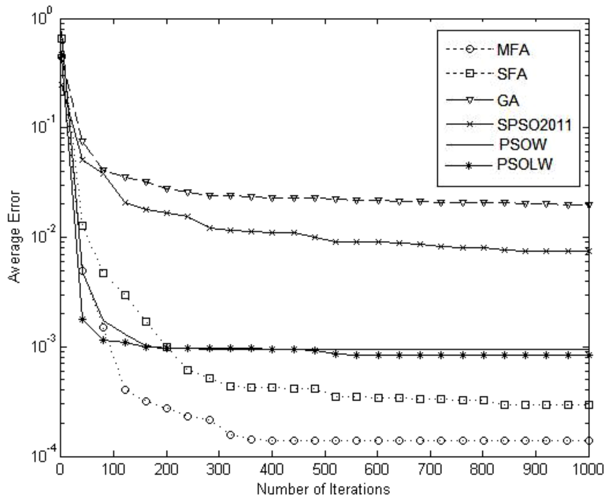

Fig. 4a. Learning curves of all algorithms for example-1 modelled using a 2nd order system.

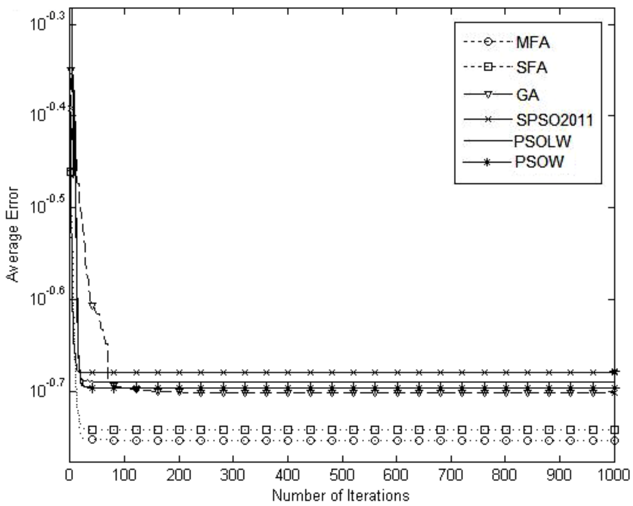

Fig. 4b. Learning curves of all algorithms for example-1 modelled using a 1st order system.

In (22a), it is assumed that the structure of the plant is known, therefore, the system identification problem reduces to parameter estimation of the nonlinear plant. In (22b) a second order IIR structure is used to model the unknown nonlinear plant. The input's are i.i.d. uniformly distributed in [-1,1] and the noise is i.i.d. uniformly in [-0.05, 0.05]. The error graphs for the two different models are shown in Figs. 5a and 5b, respectively. Like example 1, the performance measures are shown in tables 1 and 2. As seen in figures, SPSO2011 gets stuck in a local minimum since its MSE value remains steady from the beginning. FA and MFA outperform the other algorithms with smaller MSE and MSD average value and higher or equivalent convergence speed. Table 3 indicates that MFA and FA need less computational time in comparison to PSO based and GA algorithms. In this study, same as example 1, MFA shows superior performance in terms of model matching and convergence speed.

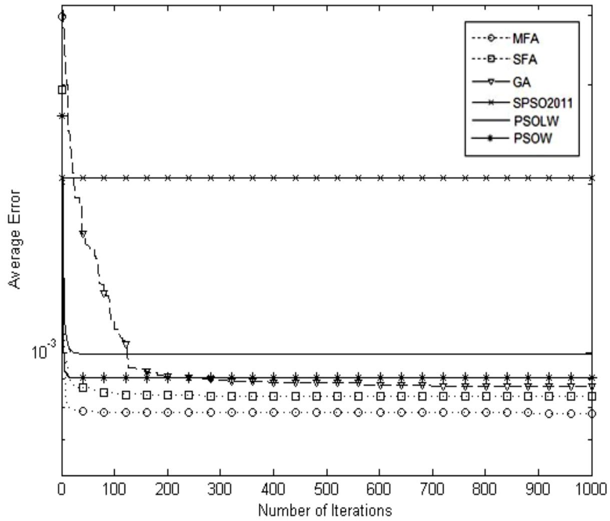

Fig. 5a. Learning curves of all algorithms for example-2 modelled using nonlinear structure.

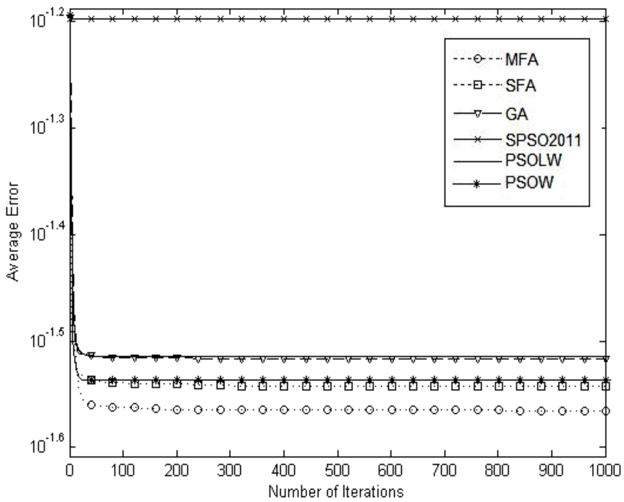

Fig. 5b. Learning curves of all algorithms for example-2 modelled using a 2nd order system.

5.3 Example 3

There are many applications where audio signals, or video signals are subjected to nonlinear processing, and which require nonlinear adaptive compensation to achieve the proper system identification and parameter extraction (Lee and El-Sharkawi, 2002). For example a generic communication system is shown in Fig.6. Many nonlinear systems can be represented by the Wiener-Hammerstein model of fig.6. In this example an LNL cascade system taken from (Krusienski and Jenkins, 2005; Mathews and Sicuranze, 2000), is considered. The plant consists of a 4th order Butterworth low-pass filter (23), followed by a memoryless nonlinear operator, followed by a 4th order Chebyshev lowpass filter (24), as illustrated in Fig.7, this system is a common model for satellite communication systems in which the linear filters model the transmission paths to and from the satellite and the nonlinearity models the travelling wave tube (TWT) transmission amplifiers operating near saturation region.

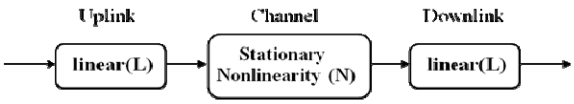

Fig. 6. .Model of a typical nonlinear channel.

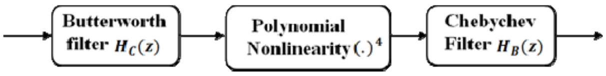

Fig. 7. LNL Unknown System

$$\hat{H}_B(z) = \frac{(0.2851+0.5704z^{-1}+0.2851z^{-2})\ \ (0.2851+0.5701z^{-1}+0.2851z^{-2})}{(1-0.1024z^{-1}+0.4475z^{-2})\ \ (1-0.0736z^{-1}+0.0408z^{-2})} \tag{23}$$

$$\hat{H}_c(z) = \frac{(0.2025+0.288z^{-1}+0.2025z^{-2})\ \ (0.2025+0.00341z^{-1}+0.2025z^{-2})}{(1-1.01z^{-1}+0.5861z^{-2})\ \ (1-0.6591z^{-1}+0.01498z^{-2})} \tag{24}$$

Two cases are considered for the LNL (Weiner-Hammerstein) adaptive filter structure, in case 1 the linear and nonlinear parts have the same order as the actual system and in case 2, the reduced order structure is given as follows:

$$\hat{H}_c(z) = \frac{d_0+d_1z^{-1}+d_2z^{-2}+d3z^{-3}}{1+e_1z^{-1}+e_2z^{-2}+e_3z^{-3}} \tag{25}$$

$$\hat{H}_B(z) = \frac{a_0+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}}{1+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}} \tag{26}$$

$$nonlinearity = c\left[\hat{H}_B(z)\right]^4 \tag{27}$$

The input is a pseudo-random binary sequence shown in Fig.8. The learning curves for both cases are given in Fig.9a and 9b. Same as the first two examples, Table 1 and 2 provides a quantitave measure of the six algorithms in terms of MSE and MSD. The convergence characteristic of algorithms is clear from the figures, where in both cases MFA reaches its minimum MSE level in a less number of iterations than the other algorithms. Besides the higher speed of convergence, MFA encompasses the best average MSE and MSD values that demonstrate its superior performance in finding the global minimum. After MFA, FA has the best results. In GA, the chromosomes and in SPSO2011 and PSOW particles will become stagnate leading to a suboptimal solution. Stochastic optimization algorithms, particularly versions of PSO, have a tendency to excel on lower order parameter spaces, This is because of the fact that higher dimentional parameter spaces tend to exhibit more local minima in general (Krusienski and Jenkins, 2005). It can be understood from the tables 1 and 2 that most of the algorithms fail to fine tune their solutions around a global minimum. The standard deviation values have considerably increased (even more than the average value). In both cases MFA has the smallest standard deviation. As a result we can say that our proposed method is more robust to the dimension of the problem. MSD Table results reveals that the overal ability of algorithms in fine adjusting the 19 parameters of the actual order model is considerably decreased in comparision with the last two examples. The computational time required for FA and MFA, as shown in Table 3 is less than the other algorithms.

All in all, the extensive simulation results given in this section certified that MFA, FA, PSOW and PSOLW in contrast to conventional PSO and GA have a minor chance of pre-mature convergence. Both MFA and FA are capable of attaining the noise floor but MFA can perform this job with greater accuracy and faster convergence.

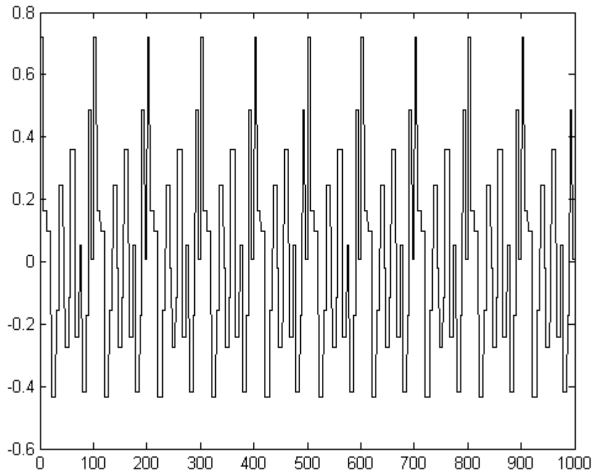

Fig. 8. Input pseudo-random binary sequence.

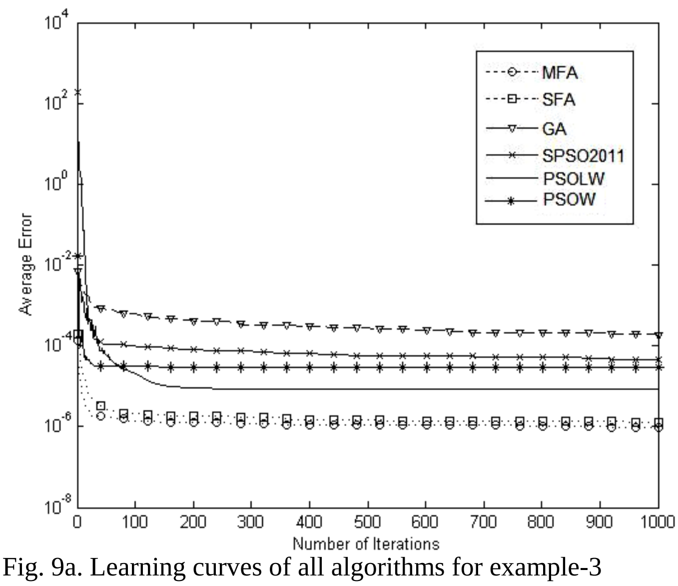

Fig. 9a. Learning curves of all algorithms for example-3 modelled using actual order structure.

6. SENSITIVITY ANALYSIS

To study the effect of parameters and population size on our proposed methods, three experiments are carried out in this section.

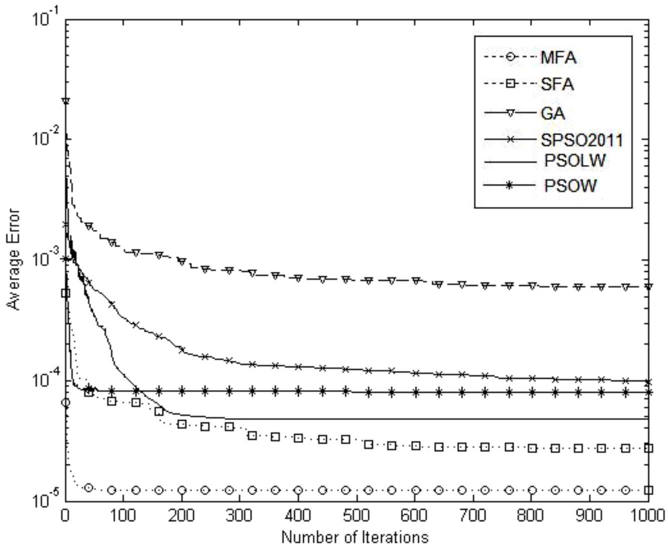

Fig. 9b. Learning curves of all algorithms for example-3 modelled using reduced order structure.

6.1 Experiment 1: variation of population size

In this experiment, Simulations are repeated with 5, 25, and 100 fireflies and the results for each example are provided in table 4.

In example 1 and 2 population size of 100 fireflies, and in example 3, size of 50, corresponds to the best average MSEvalue. It is clear from the table that, average MSE values for 50 and 100 numbers of fireflies are relatively close.

Furthermore, we know that, in the same number of iterations (1000), the total number of fitness function evaluation will increase with the population size. Therefore, to reduce the computational complexity, population size of 50 can be a reasonable choice for optimization problems.

6.2 Experiment 2: variation of α in FA

In this experiment α is varied (α =0.1, 0.3, 0.5 and 0.9) while the other parameters are kept unchanged. Results for the three examples are given in corresponding table. In all cases except one (example 2), It can be seen that the best average results with respect to MSE is obtained using a value of 0.1 for α . Hence, this value is preferred in our simulations and in modified firefly algorithm it sounds to be reasonable to decrease α from 0.25 to 0.05.

Table 1. MSE

Examples	MSE	MFA	FA	SPSO2011	PSOW	PSOLW	GA
1 (2nd Order IIR)	Best	4.7408e-5	4.9576e-5	0.0031	7.1567e-4	8.8742e-4	9.7564e-4
	Average	**1.3853 e-4**	2.9318e-4	0.007	8.3594e-4	9.5719e-4	0.0195
	Std. dev.	3.0581e-5	1.1404e-4	0.0031	3.5959e-4	2.1835e-4	0.0136
1 (1st Order IIR)	Best	0.1635	0.1708	0.1953	0.1930	0.1941	0.1062
	Average	**0.1776**	0.1810	0.2102	0.2010	0.2012	0.1985
	Std. dev.	0.0082	0.0079	0.0092	0.0045	0.0060	0.0242
2 (Nonlinear)	Best	7.5040e-4	7.5594e-4	1.3986e-3	8.5867e-4	9.4805e-4	8.5842e-4
	Average	**7.8023e-4**	8.3626e-4	1.966 e-3	8.9683e-4	9.9582e-4	8.6946e-4
	Std. dev.	2.1070e-5	2.6030e-5	5.8656e-4	2.8878e-5	1.9684e-5	1.1691e-5
2 (2nd Order IIR)	Best	0.0258	0.0250	0.0396	0.0280	0.0289	0.0282
	Average	**0.0272**	0.0287	0.0635	0.0291	0.0305	0.0304
	Std. dev.	0.0011	0.0018	0.0186	8.8168e-4	0.0012	0.0017
3 (Actual Order)	Best	6.3312e-7	6.4407e-7	3.0011e-5	5.9814e-6	1.0064e-6	4.2699e-5
	Average	**9.3079e-7**	1.1744e-6	6.6718e-5	2.8865e-5	8.7322e-6	1.8414e-4
	Std. dev.	2.7981e-7	5.0960e-7	6.3202e-5	5.3127e-5	8.9497e-6	7.4256e-5
3 (Reduced Order)	Best	7.1403e-7	6.1310e-6	3.6668e-5	1.1303e-5	2.3216e-6	1.9479e-4
	Average	**1.1093e-5**	4.8427e-5	9.4331e-5	8.0449e-5	4.4544e-5	6.0501e-4
	Std. dev.	7.0102e-6	6.2210e-5	3.8596e-5	8.4725e-5	7.8793e-5	3.6791e-4

Table 2. MSD

Examples	MSD	MFA	FA	SPSO2011	PSOW	PSOLW	GA
1 (2nd Order IIR)	Best	3.1425e-6	1.0250e-6	9.7541e-4	1.3750e-7	1.2500e-8	3.1109e-4
	Average	**4.6392e-5**	6.6524e-5	0.0044	7.5330e-5	7.9323e-5	0.0196
	Std. dev.	5.3854e-5	7.4581e-5	0.0032	2.1903e-4	2.6667e-4	0.0132
2 (Nonlinear)	Best	8.5999e-7	2.1800e-6	0.0027	1.0066e-6	4.8133e-6	1.9360e-5
	Average	**1.0047e-5**	1.4573e-5	0.0061	3.8469e-5	7.9675e-5	1.6351e-4
	Std. dev.	8.0897e-6	1.3455e-5	0.0044	4.6887e-5	1.1677e-4	2.2962e-4
3 (Actual Order)	Best	0.0028	0.0012	0.0341	0.0232	0.0166	0.1171
	Average	**0.0062**	0.0072	0.0585	0.0421	0.0284	0.1647
	Std. dev.	0.0051	0.0110	0.0112	0.0095	0.0086	0.0315

Table 3. Computational Time(in second)

Example	time	MFA	FA	SPSO2011	PSOW	PSOLW	GA
1	Best	9.6921	11.3306	6.9663	7.7660	6.0719	99.2362
(2nd Order IIR)	Average	10.1982	11.9361	**7.2421**	7.9313	**6.2099**	1.0203e+2
	Std. dev.	0.3404	0.3864	0.2365	0.1038	0.0867	1.6149
1	Best	3.3583	3.2734	5.3330	6.7254	5.0447	99.3974
(1st Order IIR)	Average	3.4574	**3.3915**	5.5708	6.9379	5.2868	1.0149e+2
	Std. dev.	0.0665	0.0874	0.2453	0.1328	0.1323	1.1300
2	Best	17.1116	16.4762	19.8583	20.4498	18.6293	361.8733
(Nonlinear)	Average	17.2752	**16.9559**	20.6419	21.0125	18.8453	363.6922
	Std. dev.	0.2369	0.51195	1.13718	0.41903	0.17340	2.2848
2	Best	4.0286	3.7758	6.7679	7.4658	6.6034	3.4089e+2
(2^{nd} Order IIR)	Average	4.1159	**3.8250**	6.8628	8.4828	6.7033	3.4914e+2
	Std. dev.	0.0624	0.0500	0.0698	0.4635	0.0679	7.4158
3	Best	36.2201	34.8042	39.4432	42.4370	40.1864	2.3139e+2
(Full Order)	Average	36.8455	**35.5898**	42.8841	44.0122	41.9984	2.3371e+2
	Std. dev.	0.4689	0.4706	3.5035	1.5265	1.6044	2.0094
3	Best	32.3354	31.5373	33.1299	39.9935	34.4355	2.2778e+2
(Reduced Order)	Average	33.1410	**31.8433**	34.2243	41.5944	36.0785	2.2895e+2
	Std. dev.	0.7970	0.2126	1.0661	1.1204	1.1718	1.03104

Table 4. MSE result of experiment 1

Example	MSE	Pop.Size =5	Pop.Size =25	Pop.Size =50	Pop.Size =100
	Best	5.511e-5	6.830e-5	4.740e-5	8.390e-5
1	Average	1.512e-4	1.431e-4	1.385e-4	**1.362e-4**
	Std. dev.	8.664e-5	6.357e-5	3.058e-5	4.329e-5
	Best	7.659e-4	7.504e-4	7.535e-4	7.339e-4
2	Average	7.872e-4	7.802e-4	**7.752e-4**	7.868e-4
	Std. dev.	2.214e-5	2.107e-5	1.573e-5	2.261e-5
	Best	7.456e-7	7.640e-7	6.431e-7	6.551e-7
3	Average	1.302e-6	1.477e-6	**9.407e-7**	1.267e-6
	Std. dev.	5.772e-7	6.761e-7	2.798e-7	6.018e-7

Table 5. MSE result of experiment 2

Example	MSE	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.9$
	Best	4.957e-5	4.408e-4	0.001	0.007
1	Average	**2.931e-4**	0.001	0.004	0.026
	Std.dev.	1.140e-4	5.921e-4	0.002	0.011
2	Best	8.053e-4	7.559e-4	7.932e-4	8.252e-4
IIR	Average	8.401e-4	**8.362e-4**	8.400e-4	8.505e-4
	Std.dev.	2.092e-5	2.603e-5	2.411e-5	2.603e-5
2	Best	0.0267	0.025	0.028	0.028
Nonlinear	Average	0.0293	**0.028**	0.029	0.029
	Std. dev.	0.0017	0.001	9.838e-4	0.001
	Best	6.440e-7	6.945e-7	6.820e-7	6.989e-7
3	Average	**1.174e-6**	1.975e-6	2.225e-6	2.414e-6
	Std. dev.	5.096e-7	1.332e-6	1.648e-6	2.290e-6

6.3 Experiment 3: variation of λ in MFA

Effect of different values of λ (λ = 0.1, 0.3, 0.5 and 0.9) on MFA is investigated in this experiment. Like experiment 2, results show that value of 0.1 is a reasonable choice for λ. This could be predictable, because α and λ have the same function in firefly algorithm.

Table 6. MSE result of experiment 3

Example	MSE	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.9$
	Best	4.740e-5	1.171e-4	9.353e-5	1.102e-4
1	Average	**1.385 e-4**	1.972e-4	2.113e-4	2.289e-4
	Std. dev.	3.058e-5	5.281e-5	8.153e-5	9.521e-5
2	Best	0.025	0.025	0.027	0.028
IIR	Average	0.027	**0.027**	0.028	0.028
	Std. dev.	0.001	0.001	7.989e-4	5.303e-4
2	Best	7.473e-4	7.504e-4	7.607e-4	7.618e-4
Nonlinear	Average	7.817e-4	**7.802e-4**	7.854e-4	7.878e-4
	Std.dev.	2.185e-5	2.107e-5	1.622e-5	2.089e-5
	Best	6.312e-7	8.994e-7	9.268e-7	1.067e-6
3	Average	**9.307e-7**	1.619e-6	2.138e-6	2.503e-6
	Std. dev.	2.798e-7	7.987e-7	1.263e-6	1.682e-6

7. CONCLUSIONS

The Firefly optimization algorithm has the advantages of being easy to understand and simple to implement, so that, it can be used for wide variety of optimization tasks. To our knowledge, this is the first report of applying firefly algorithm in adaptive system identification task. In order to enhance the searching quality of the algorithm we have performed two modifications, first by reducing the randomness and second by hybridizing the algorithm with PSO by adding the global search component of PSO to the updating formula. The proposed algorithm was used for identification of three benchmark systems: a linear IIR, a nonlinear IIR and an LNL system, respectively.

The performance of MFA is compared with that of a standard version of firefly (FA), three versions of PSO and a standard version of GA. The lower values of mean square error between the actual and estimated system for MFA, in all studies show that it is a promising candidate for adaptive system identification. In most cases the computational complexity of MFA and FA is less than the other four algorithms. In addition, the simulation results show that MFA has better or at least equivalent, convergence speed than FA, GA and three PSO's. Therefore the proposed method can be applied in real-time applications. To further verify its robustness, MFA needs to be applied to various benchmarks and real world dynamical systems. In this paper, it was assumed that the structure of the model e.g. the order of the IIR model is known as a priori. Hence, finding the proper order for an IIR model, which brings about a trade-off between computational complexity and model matching ability of the algorithm, can be a good point for further research. Hardware implementation is another point that is considered by authors for their future work.

REFERENCES

Apostolopoulos, T. and Vlachos, A., (2011). Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem, *International Journal of Combinations*,, Article ID 523806

Astriim, K.J. and Eykhoff, P., (1971). System identification – a survey. *Automatica*, Vol. 7, pp. 123-162.

Bai, E. W., Tempo, R. and Liu, Y., (2007.). Identification of iir nonlinear systems without prior structural information. *IEEE Trans. on Automatic Control*, Vol.52, No. 3.

Brouwn, G.G., Krijgsman, A.J., (1994), Hybrid neural networks for nonlinear system identification. *IEEE International Conference on control*, pp. 21-24.

Dai, C., Chen, W., Zhu, Y. (2010) Seeker optimization algorithm for digital iir filter design. *IEEE transactions on industrial electronics*, Vol.57, No.5, pp. 1710-1718.

Goldberg, D.E., (1989). Genetic algorithms in search. Optimization and Machine Learning, Reading, Mass., Addison wesley.

Holland, J.,(1975) Adaption in natural and artificial systems, University of Micchigan Press, Ann anbor.

Horng M.-H. and Jiang T. W., (2010). The codebook design of image vector quantization based on the firefly algorithm, In: *Computational Collective Intelligence, Technologies and Applications,* LNCS, Vol. 6423, pp. 438-447.

Johnson, C. R. Jr. (1984). Adaptive iir filtering: Current results and open issues," *IEEE Trans. on Info. Theory*, Vol. 30, pp. 237-250.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital iir filters. *Journal of the Franklin Institute*, 346, 328-348.

Kristinsson, K., Dumont, G.A. (1992) System identification and control using genetic algorithms. *Systems, IEEE Transactions on Systems, Man and Cybernetics*, 22, 1033-1046.

Krusienski, D.J., Jenkins, W.K., (2005). Design and performance of
adaptive systems based on structured stochastic optimization strategies. *IEEE Circuits and Systems Magazine*. 5, 8-20.

Kumon, T., Iwasaki, M., sizuki, T., Hashiyama, T., Matsui, N.,

Okuma, S., (2000). Nonlinear system identification using genetic algorithm. Industrial Electronics Society, IECON 2000. *26th Annual Conference of the IEEE*. 4, pp. 2485-2491, Hall, Englewood Cliffs, 1987.

Lee, B.H., Kim, S., Seok, J., Won, S., (2006). Nonlinear system identification based on support vector machine using particle swarm optimization. In: *International Joint Conference, SICE-ICASE*, October 2006, pp. 5614-5618.

Lee, K.Y., El-Sharkawi, M.A., ed., (2002). Tutorial on modern heuristic optimization techniques with applications to power systems, *IEEE power engineering society.*

Liu, Bo., Wang, L., Jin, Y.H., Tang, F., Huang, D.X., (2005). Improved particle swarm optimization with chaos. *Chaos, Solitons & Fractals*, pp. 1261-1271.

Ljung, L. and Gunnarsson, S., (1990). Adaptation and tracking in system identification- A Survey. *Automatica*, Vol. 26, pp. 7-22.

Luitel, B. & Venayagamoorthy, G. K. (2010). Particle swarm optimization with quantum infusion for system identification. *Engg. applications of Artificial Intelligence*, Vol. 23, No. 10, pp. 635-649.

Majhi, B. & Panda, G. (2010). Development of efficient identification scheme for nonlinear dynamic systems using swarm intelligence techniques. *Expert Systems with Applications*, Vol. 37, pp. 556-566.

Mathew, V.J, & Sicuranze, (2000). Polynomial signal processing, John Wiley & Sons, Inc.

Netto, S.L., Diniz, P.S.R., Agathoklis, P., (1995). Adaptive IIR filtering algorithms for system identification: a general framework, *IEEE Transaction on education* 38, 54-66.

NG, S.C., Leung, S.H., Chung, C.Y., Luk, A., Lau, W.H., (1996). The genetic search approach. A new learning algorithm for adaptive IIR filtering. *IEEE Signal Process. Magazine*,38-46.

Panda, G., Mohanty, D., Majhi, B., Sahoo, G., (2007). Identification of nonlinear systems using particle swarm optimization technique. In: *IEEE Conference on Evolutionary Computation*, September 2007, pp. 3253-3257.

Panda, G., Pradhan, P.M., Majhi, B. (2011). IIR system identification using cat swarm optimization. *Expert Systems with Applications*, Vol. 38, No. 10,12671-12683.

Shen, Y., Zeng, C., (2007) A self organizing particle swarm optimization algorithm and application. In: *Third International Conf. on Natural Computation*, Vol. 4, pp. 668-672.

Shi, Y., and Eberhart, R., (1999). Empirical study of particle swarm optimization. in *Proc.Congr.Evol.Comput.,* pp.1945-50.

Shynk, J.J., (1989). Adaptive iir filtering. *IEEE Trans. on Acoustic Speech and Signal Processing*, pp. 4-21, Apr.

Standard PSO 2011 (SPSO2011) on the particle swarm central, Programs section. [Online]. Available: http://www.particleswarm.info

Theofilatos, K., Beligiannis, G.,and Likothanassis, S., (2009). Combinig evolutionary and stochastic gradient techniques for system identification, *Journal of Computation and Applied Mathematics*, 227,147-160.

Warwick, K., Kang, Y.H., Mitchell, R.J., (1999). Genetic least square for system identification, *Soft Computing*, 3, 200-205.

Widrow, B., McCool, J.M, Larimore, M.G, Johnson, C.R. JR., (1976). Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proceedings of the IEEE*, 64, pp. 1151-1162.

Yang, X-S. (2008). Nature-Inspired Metaheuristic Algorithm Luniver Press.

Yang, X.-S.,(2010). Firefly algorithm, stochastic test functions and design optimization, *Int. J. Bio-inspired Computation*, Vol. 2, pp.78
78-84

Yang, X. S., (2010). Engineering Optimization: An introduction with Metaheuristic Applications," John Wiley & Sons.

Yang, X.S., Hosseini, S.S., Gandomi, A.H., (2012). Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, *Applied Soft Computing*, Vol. 12(3), 1180-1186.