13th International Conference on
*AEROSPACE SCIENCES & AVIATION TECHNOLOGY*,
*ASAT- 13,* May 26 – 28, 2009, E-Mail: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel : +(202) 24025292 – 24036138, Fax: +(202) 22621908

# System Identification Using Intelligent Algorithms

Zaki B. Nossair[*], A. A. Madkour[**], M. A. Awadalla[***] , M. M. Abdulhady[†]

**Abstract:** This research presents an investigation into the development of system identification using intelligent algorithms. A simulation platform of a flexible beam vibration using finite difference (FD) method is used to demonstrate the capabilities of the identification algorithms. A number of approaches and algorithms for system identifications are explored and evaluated. These identification approaches using (a) traditional Recursive Least Square (RLS) filter, (b) Genetic Algorithms (GAs) (c) Adaptive Neuro_Fuzzy Inference System (ANFIS) model (d) General Regression Neural Network (GRNN) and (e)Bees Algorithm (BA). The above algorithms are used to estimate a linear discrete second order model for the flexible beam vibration. The model is implemented, tested and validated to evaluate and demonstrate the merits of the algorithms for system identification. Finally, a comparative performance of error convergence of the algorithms is presented and discussed.

**Keywords:** System identification, adaptive control, intelligent identification, recursive least squares algorithm, Genetic algorithm, ANFIS, Bees Algorithm.

## 1 Introduction

This paper presents an investigation into the development of a discrete time model based on the observation of the input and output signals. Such models can be used for control system design, adaptive guidance or fault detection [1]. Parameter estimation, in turn system identification is a common criterion for control system, in particular for sensitive or adaptive control system design. In fact, a closed loop control system may be unstable or exhibit unacceptable transient response characteristics if the estimated parameters used in the system model for controller design do not coincide with the actual process parameters. Therefore, accurate and reliable parameters estimation technique is critical for the design and development of high-performance control systems in which the estimated parameters are often used in the field orientation, motion control, self-sensing, and other advanced algorithms.

The main objective of this paper is to identify a linear discrete second order model using RLS filter, GAs ANFIS, GRNN and BA. A simulation platform of a flexible beam system in transverse vibration using FD method [4] is considered to demonstrate the capabilities of the algorithms for system identification. The proposed second order model is implemented using the RLS, GAs, ANFIS, GRNN, and BA. It is then tested and validated for system identification within the simulation framework of a flexible beam system. Finally, a

[*] Department of Computer Science, University of Helwan, Helwan, Egypt, Faculty of Engineering, Helwan University, znosssair@yahoo.com
[**] a.a.m.madkour@bradford.ac.uk
[***] awadalla_medhat@yahoo.co.uk
[†] mida.xx@gmail.com

comparative performance of the fore algorithms are presented and discussed to demonstrate the capabilities of the algorithm in implementing system identification.

## 2 Traditional RLS Algorithms

This is a traditional adaptive filter algorithm. It estimates the current parameter vector $\hat{\theta}(k)$ based on the previous estimated vector $\hat{\theta}(k-1)$, as follows [2], [3]:

$$\hat{\theta}(k) = f(\hat{\theta}(k-1), D(k), k) \qquad (1)$$

where, $D(k)$ denotes data available at time $(k)$, and f(.,.,.) denotes an algebraic function, the form of which determines the specific algorithm. In the case of dynamic system, data $D(k)$ normally consider the form of present and past observation of the system outputs and inputs. For multi-parameter system, this form can be represented as follows:

$$y(k) = \psi^T(k)\theta \qquad (2)$$

where,

$$\begin{aligned}\psi(k) = [&-y(k-1), \cdots, -y(k-m),\\ &u(k), \cdots, u(k-m)]^T\end{aligned} \qquad (3)$$

$$\theta = [a_1, \cdots, a_m, b_1, \cdots, b_{m+1}]^T \qquad (4)$$

The estimation of the parameters vector $\theta$ is performed in a way such that the estimated $\hat{\theta}_r$ minimizes the cost index $J(r)$, where $r$ denotes the number of sets of measurement,

$$J(r) = \sum_{k=1}^{r} (y(k) - \psi(k)^T \hat{\theta}(r))^2 \qquad (5)$$

Equation (5) can be written in a recursive form as:

$$\begin{aligned}\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\psi(k)\\ (y(k) - \Psi^T(k)\hat{\theta}(k-1))\end{aligned} \qquad (6)$$

$$P(k) = \left[ P(k-1) - \frac{P(k-1)u(k)\psi^T(k)P(k-1)}{1+\psi^T(k)P(k-1)\psi(k)} \right] \qquad (7)$$

## 3 Intelligent Identification Algorithms

The conventional system identification schemes are in essence local search techniques. These techniques often fail in the search for the global optimum if the search apace is not differentiable or linear in the parameters. On the other hand, these techniques do not iterate more than once on each datum received. An alternative strategies using artificial intelligence algorithm could provide better solution. To achieve this goal four most commonly used intelligence algorithms are used to demonstrate the capabilities. These are described below.

## 3.1 Genetic Algorithm

Over the last decade, genetic algorithms (GAs) have been extensively used as search and optimization tools in various problem domains. GA simultaneously evaluates many points in the parameter space and converges towards the global solution. The genetic algorithm differs from other search techniques by the use of concepts taken from natural genetics and evolution theory [5], [6]. According to Goldberg [7] GAs are different from more normal optimization and search procedures in four ways:

- GAs work with a coding of the parameter set, not the parameters themselves.
- GAs search from a population of points, not a single point.
- GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules.

The basic GA evolution can be summarized as follows: create a population of individuals (solutions), evaluate their fitness, generate a new population by applying genetic operators, and repeat this process for a number of times [6-8]. The GA iteratively uses selection, crossover and mutation operators for the population evolution. Selection operator selects solutions to be parents based on their fitness. Crossover operator is used on these parent strings to obtain the new solutions that inherit the good and bad properties of their parent solutions. There are many traditional crossover operators such as uniform, single point and multipoint crossovers. Mutation operator is then applied to produce new characteristics, which are not present in the parent solutions. The newly created solutions form new population for the next generation.

The GAs consider the same multi parameter system given by equation (2). Binary strings are used to represents a solution (individual). The fitness function for the system can be defined as [2],

$$f(e) = \sum_{k=1}^{r} \left| y(k) - \hat{y}(k) \right| \tag{8}$$

where, $y(k)$ is measured output, $\hat{y}(k)$ is estimated model output, and $r$ is the number of sets of measurement considered. Equation (8) may be written in vector form as:

$$f(e) = \sum_{k=1}^{r} \left| y(k) - \hat{\theta}_0 \hat{\psi}(k) \right| \tag{9}$$

## 3.2 Adaptive Neuro-Fuzzy Inference System

The ANFIS techniques provide a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input-output data. This learning method works in a similar form of the neural networks. There is a MATLAB© function in the Fuzzy Logic Toolbox that accomplishes this membership function parameter adjustment called ANFIS. This hybrid adaptive Neuro-fuzzy function ANFIS is used for system identification which is the major training routine for Sugeno-type fuzzy inference systems (FIS). ANFIS has been reported to produce good results as a function approximation tool [9-11] .

Figure 1 shows the basic structure of the ANFIS algorithm for a first order Sugeno-style fuzzy system. It is worth noting that the Layer-1 consists of membership functions described by generalized bell function:

$$\zeta(X) = (1 + ((X - c)/a)^{2b})^{-1} \tag{10}$$

where $a, b$ and $c$ are adaptable parameters. Layer-2 implemented the fuzzy AND operator, while Layer-3 acts to scale the firing strengths. The output of the Layer-4 is comprised of a linear combination of the inputs multiplied by the normalized firing strength $w$.

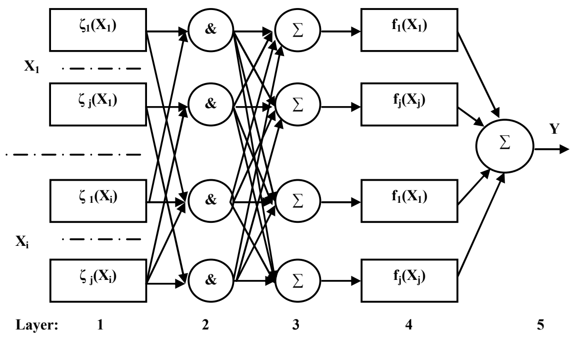


**Fig. 1 Basic ANFIS structure.**

$$Y = w(pX + r) \tag{11}$$

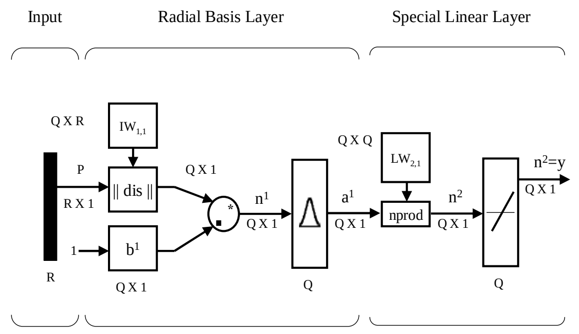
where, $p$ and $r$ are adaptable parameters. Layer-5 is simple summation of the outputs of layer-4. The adjustment of modifiable parameters is a two step process. First, information is propagated forward in the network until Layer-4, where the parameters are identified by a least-squares estimator. Then the parameters in Layer-2 are modified using gradient descent. The only user specified information is the number of membership functions in the universe of discourse for each input and the input-output is considered as training data.

## 3.3 The General Regression Neural Network

Use of neural network (NN) techniques to solve problems in guidance began in the late 1990s by E. J. Song, and L. H. Tahk [12]. The basic idea of neural network midcourse guidance is to train neural networks to learn the functional form of the optimal guidance command in terms of the current states and terminal conditions, and use them for real-time guidance [13].
The GRNN is the NN architecture that can solve any function approximation problems in the sense of estimating a probability distribution function [14]. It is a powerful memory based network that could estimates continuous variables and converges to the underlying regression surface [15].

The main advantage of the GRNN approach is simplicity. It is noted that the adjustment of only one parameter is sufficient for determining the network. GRNN approximates any arbitrary function between input and output vectors, drawing the function estimate directly from the training data. Furthermore, as the training set size becomes large, the estimation error approaches zero, with only mild restrictions on the function [16].

GRNN was firstly developed by Specht (1991), who claims that the algorithm in GRNN is able to provide a smooth transition from one observed value to another, even with sparse data in a multidimensional measurement space [17].



Where, R = no. of elements in the input vector Q = no. of neurons in the layer

$a_i l$ = radbas (|| $IW_{1,1}$ -p|| $b_i l$ ) a2 = purelin(n2)

$a_i l$ is $i^{th}$ element of a1

$IW_{1,1}$ is a vector made of the $i^{th}$ row of $IW_{1,1}$

**Fig. 2 Basic GRNN structure.**

GRNN is a feed-forward NN established on the theory of non-linear regression. It is a three-layer network with one hidden layer [14]. Each layer has entirely different roles.

The input layer: where the inputs are applied.

The hidden layer: where a nonlinear transformation is applied on the data from the input space to the hidden space, in most applications the hidden space is of high dimensionality.

The linear output layer: where the outputs are produced.

## 3.4 Bees Algorithm

The Bees Algorithm is an optimization algorithm inspired by the natural foraging behaviour of honey bees to find the optimal solution. It belongs to the category of “intelligent” optimization tools as it also simultaneously evaluates many points in the parameter space and converges towards the global solution and is also based on the method of minimization of the prediction error. Bees Algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2 by equation (8) also like The GA In step 4, bees that have the highest fitnesses are chosen as “selected bees” and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the best e sites which represent more promising solutions are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population – representatives from each selected patch and other scout bees assigned to conduct random searches.[18]

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) - format new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While

## 4 The flexible beam system

Consider a cantilever beam as a plant model of length $L$, fixed at one end and free at another, with a force $U(x,t)$ applied at a distance $x$ from its fixed (clamped) end at time $t$, resulting a deflection $y(x,t)$ of the beam from its stationary (fixed) position at the point where the force has been applied. The motion of the beam in transverse vibration is, thus, governed by the well known fourth-order partial differential equation (PDE) [19].

$$\mu^2 \frac{\partial^4 y(x,t)}{\partial x^4} + \frac{\partial^2 y(x,t)}{\partial t^2} = \frac{1}{m} U(x,t) \tag{12}$$

where $\mu$ is a beam constant given by $\mu^2 = \frac{EI}{\rho A}$, with $\rho$, $A$, $I$ and $E$ representing the mass density, cross-sectional area, moment of inertia of the beam and the Young modulus respectively, and $m$ is the mass of the beam. The corresponding boundary conditions at the fixed and free ends of the beam are given by

$$\begin{gathered} y(0,t) = 0 \quad \text{and} \quad \frac{\partial y(0,t)}{\partial x} = 0 \\ \frac{\partial^2 y(L,t)}{\partial x^2} = 0 \quad \text{and} \quad \frac{\partial^3 y(L,t)}{\partial x^3} = 0 \end{gathered} \tag{13}$$

Note that the model thus utilised incorporates no damping. To construct a suitable platform for test, a method of obtaining numerical solution of the PDE in equation (12) is required. This can be achieved by using the finite difference (FD) method. This involves a discrimination of the beam into a finite number of equal-length sections (segments), each of length $\Delta n$, and considering the beam motion (deflection) for the end of each section at equally-spaced time steps of duration $\Delta t$. Thus, first-order central FD methods is used to approximate the partial derivative terms in equations (12) and (13) yields.

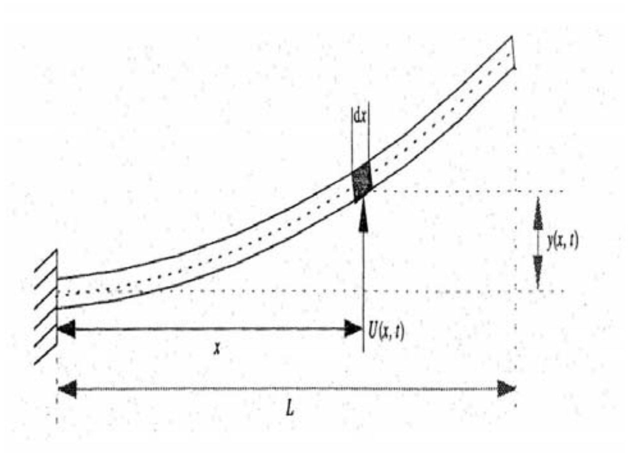


**Fig. 3 Schematic diagram of the cantilever beam system.**

$$Y_{j+1} = -Y_{j-1} - \lambda^2 S Y_j + (\Delta t)^2 U(x,t)\frac{1}{m} \tag{14}$$

where, $Y_k$ $(k = j-1, j, j+1)$ is an $n \times 1$ matrix representing the deflection of grid-points 1 to $n$ of the beam at time step $k$, S is a matrix, given in terms of characteristics of the beam and the discrimination steps $\Delta t$ and $\Delta x$, and $\lambda^2 = (\Delta t)^2 (\Delta x)^{-4} \mu^2$. Equation (14) is the required relation for the simulation algorithm, characterising the behaviour of the cantilever beam system, which can be implemented on a digital computer easily. It has been shown that a necessary and sufficient condition for stability satisfying this convergence requirement is given by $0 < \lambda^2 \leq 0.25$ [20].

## 5 Implementation and results

A cantilever beam in transverse vibration of length L = 0.635 meter, mass m = 0.037 kg, was considered as plant for investigation. The beam was discretised into 19 small segments. To allow dominant modes of vibration of the beam to be excited, a step disturbance force (0.1N) of finite duration was applied to a suitable node of the beam. The input and output samples of the plant was collected from two suitable nodes of the beam. Moreover, sample period was selected as $\Delta t = 0.3$ms which is sufficient to cover all the resonance modes of vibration of the beam [21].

A linear discrete second order model was estimated using the RLS, GA, RCGA, ANFIS, GRNN and BA.

$$Y(z) = \frac{1 + b_1(z^{-1}) + b_2(z^{-2})}{1 + a_1(z^{-1}) + a_2(z^{-2})} U(z) \tag{15}$$

Investigations were carried out using MATLAB© Genetic Algorithm Toolbox Version 2.0.2 for TCGA, Neural Network Toolbox Version 5.0.1 for GRNN, and MATLAB© Fuzzy Logic Toolbox Version 2.2.4 for ANFIS.

The system identification algorithms were carried out for about 5 s (16,700 iterations) using the linear discrete second order model (15) with grid points 16 and 20 as an input and output samples of the plant respectively with a set of input output data simulated using (14) .
RLS, GA, and BA are used to estimate the parameters of the model of (15). On the other hand, the GRNN and ANFIS are used to estimate the equivalent model using plant input and corresponding output.

It is observed that 100 generations, 8- bit representation, 10% mutation rate, and 10 population sizes are the most suitable parameters of the GA for best convergence [22].

Table 1 shows the summary of the error convergence performances of the algorithms. The error has been calculated based on the differences between absolute value of the original and the estimated signal. On the other hand, the execution time of the algorithms was measured for 16,700 iterations. It is noted that error convergences for all the algorithms are in similar level. It is also noted that the GAs perform better as compared to the RLS and the BA offers the best performance among the five algorithms, although the overall performance variation are not very significant.

**Table 1: Performance in implementing the five algorithms**

| **Algorithm** | **Error** |
|---|---|
| RLS | 0.7856 |
| GA | 0.6530 |
| ANFIS | 0.6559 |
| GRNN | 0.6388 |
| BA | 0.6079 |

A comparative performance for system identification using the RLS and GAs has been reported earlier [2]. Figure 4 shows the time domain performance of the (a) RLS, (b) GA, (c) ANFIS, (d) GRNN, and (e) BA, where the solid signal represents actual output and doted one represents the estimated output of the model. It is observed that a significant error convergence leads almost overlapping of the two signals in each case. It is also noted that the BA offers similar level of performance for error convergence as compared to the other four algorithms. Corresponding auto-power spectral density is shown in Fig. 5, which further demonstrated the similarity and level of error convergence. As shown in Fig. 4, the solid signal in Fig. 5 represents actual output and doted one represents the estimated output of the model.

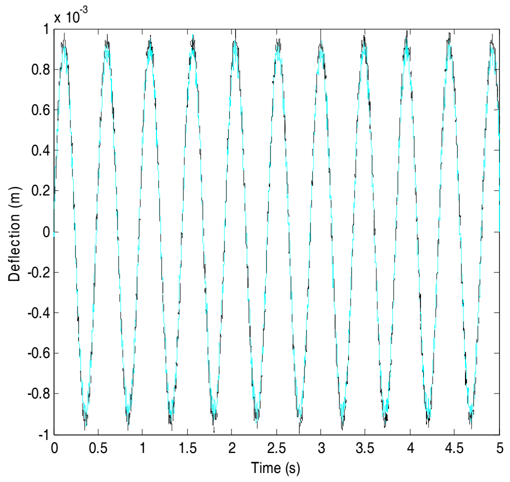


**Fig. 3.a   Performance of the RLS algorithm.**

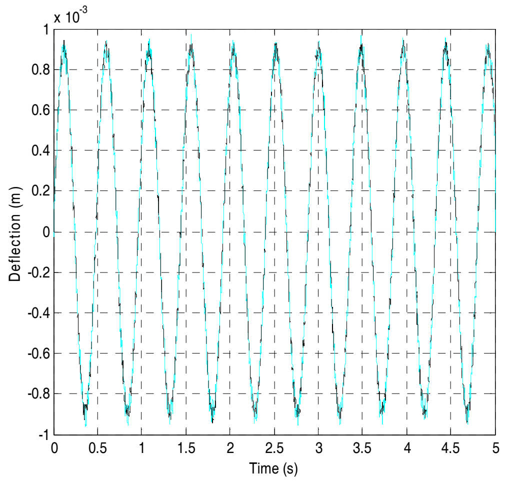


**Fig. 3.b   Performance of the GA.**

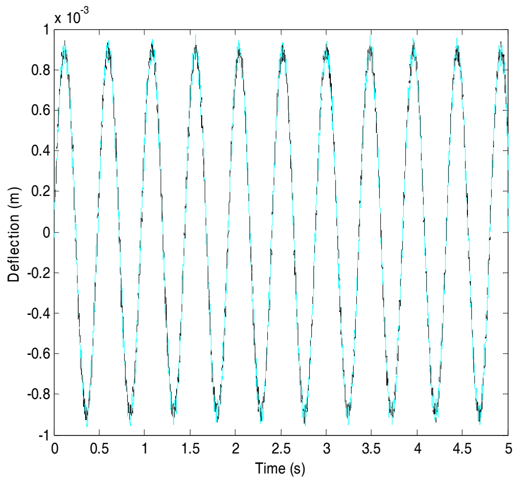


**Fig. 3.c   Performance of the GRNN algorithm.**

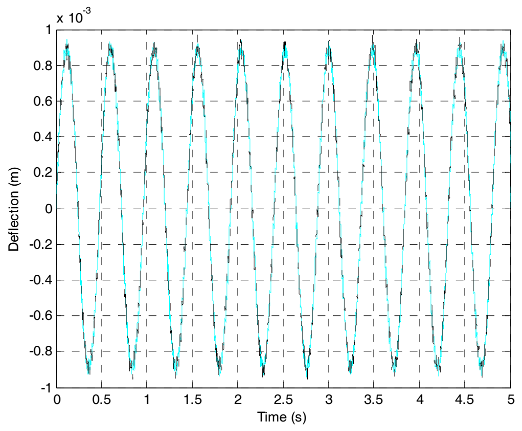


**Fig. 3.d Performance of the ANFIS algorithm.**

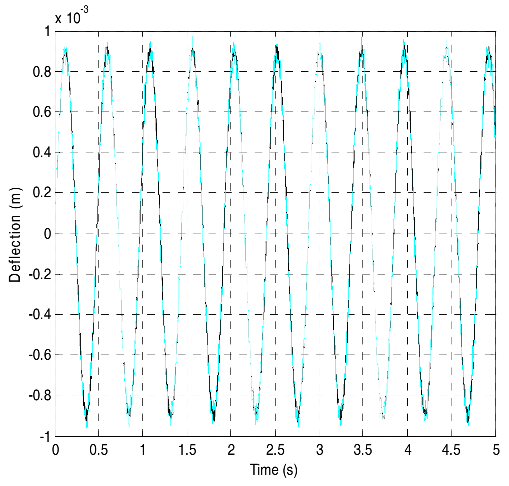


**Fig. 3.e Performance of the BA algorithm.**

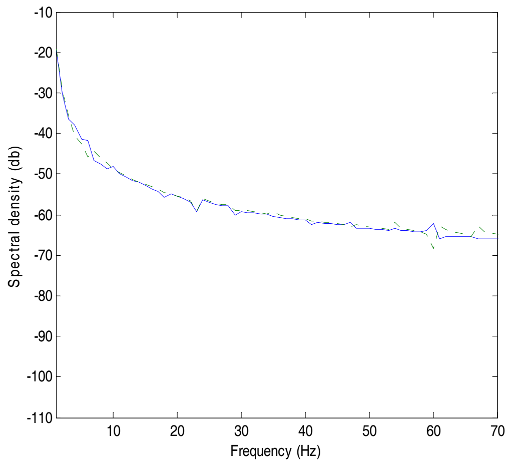


**Fig. 4.a Performance of the RLS algorithm in auto-power spectral density**

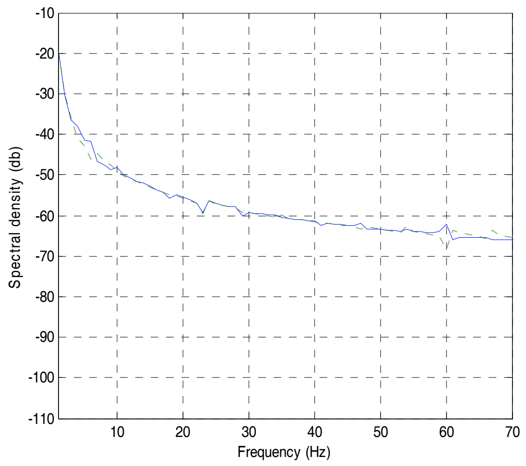


**Fig. 4.b Performance of the GA in auto-power spectral density.**

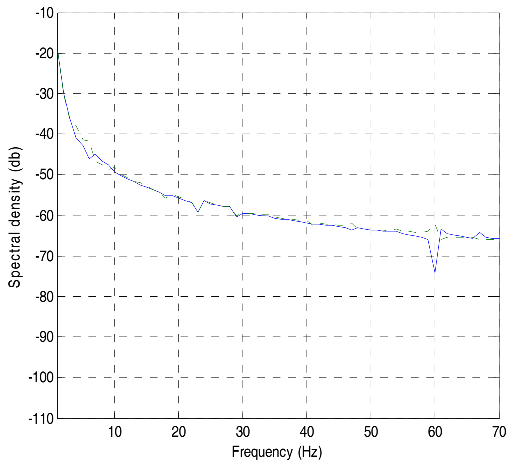


**Fig. 4.c Performance of the ANFIS algorithm in auto-power spectral density.**

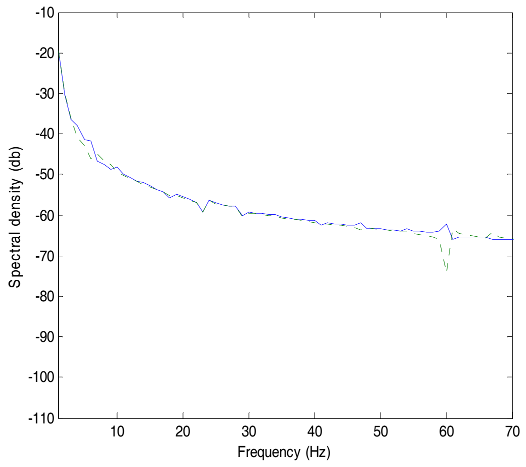


**Fig. 4.d Performance of the GRNN algorithm in auto-power spectral density.**

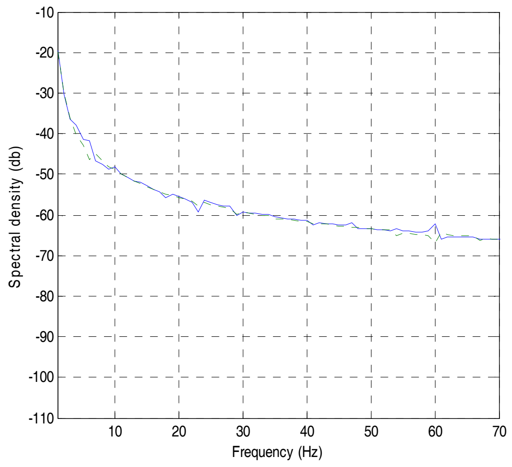


**Fig. 4.e Performance of the BA auto-power spectral density.**

## 6 Conclusion

This paper has presented the intelligent system identification of a flexible beam system for adaptive active vibration control. A plant model for a flexible beam system was considered to demonstrate the merits of the algorithms. A comparative performance of the algorithms has been presented and discussed through a set of experiments. BA shows relatively better error convergence for the same number of iterations.

Finally, a comparative performance has been provided to demonstrate the capability of the algorithms for further work on adaptive active control system design.

## 7 References

[1]. K. Ogata, "Discrete-time control systems", Prentice-Hall, Inc., 2nd edition, 1995

[2]. M. A. Hossain and M. O. Tokhi, "Evolutionary adaptive active vibration control" Proc Inst. Mechanical Eng., Vol. 211(part 1), pp. 183-193, 1997.

[3]. H. Chen and J. Zhang, "Identification and adaptive control for systems with unknown orders, delay, and coefficients", IEEE Transaction on Automatic Control, Vol. 35, No. 8, pp.866-877, 1990.

[4]. L. Xia and J .B. Moore "Recursive Identification of over parameterized systems", IEEE Transaction on Automatic Control, No. 34, 1989.

[5]. K. Kristinsson and G. Dumont, “System identification and control using genetic algorithms”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, pp: 1033-1046,1992.

[6]. D. A. Coley, "An Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific, 1999.

[7]. W. K. Lennon, K.M. Passino, “Genetic adaptive identification and control”, Engineering Applications of Artificial Intelligence, Vol.12, pp: 185-200, 1999.

[8]. D. E. Goldberg, ‘Genetic Algorithms for Search, Optimization, and Machine Learning’, Reading, MA: Addison-Wesley, 1989.

[9]. J. S. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp: 665-685, 1993.

[10]. J. S. R. Jang , N. Gulley, “Fuzzy Logic toolbox User™s Guide”, The Mathworks Inc., 1995.

[11]. J. S. R. Jang, N. Gulley, E. Mizutani, “Neuro-Fuzzy and soft computing, A computational approach to learning and Machine intelligence”, Prentice-Hall, Inc.UK, pp: 335-360,1997.

[12]. M. J. Tahk, http://fdcl.kaist.ac.kr/supra/research /s_nn_guidance.html, May, 2000

[13]. E. J. Song, H. Lee, and M. J. Tahk, “On-line suboptimal midcourse guidance using neural networks”, In Proceedings of the 35th SICE Annual Conference, Tottori University, Japan, pp: 1313—1318, 1996.

[14]. W. Kanbuaa, S. Supharatidb, and I. Tan “Ocean Wave Forecasting in the Gulf of Thailand during Typhoon Linda 1997 : WAM and Neural Network approaches”, Journal of the Science Society of Thailand, ScienceAsia Vol. 31 No.4, pp. 243-250,2005.

[15]. S. Kanmani, V. R. Uthariaraj, V. Sankaranarayanan and P. Thambidurai “Object oriented software quality prediction using general regression Neural Networks”, ACM SIGSOFT Software Engineering Notes ,Vol. 29 , No. 5, pp. 1-6, 2004.

[16]. P.D. Wasserman, “Advanced Methods in Neural Computing”, New York, Van Nostrand Reinhold, pp: 155-161, 1993.

[17]. D. F Specht, “A General Regression Neural Network”, IEEE Transactions on Neural Networks, Vol. 2, No. 6, pp: 568-576 , 1991

[18]. D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri , S. Rahim , M. Zaidi "Rahim , S. and Zaidi, M. The Bees Algorithm – A novel tool for complex optimisation problems", the 2nd Virtual International Conference on Intelligent Production Machines and Systems, Cardiff, UK, 2006, 454-459.

[19]. P. K. Kourmoulis, "Parallel processing in the simulation and control of flexible beam structure systems", PhD thesis, Dept. of Automatic Control & Systems Engineering, The University of Sheffield, 1990.

[20]. M. O. Tokhi and M. A. Hossain, “Self-tuning active vibration control in flexible beam structures", Proceedings of IMechE-I: Journal of Systems and Control Engineering, Vol. 208, pp. 263-277, 1994.

[21]. M. O. Tokhi, M. A. Hossain, and M. H. Shaheed, "Parallel Computing for Real-time Signal Processing and Control", Springer, 2002

[22]. A. A. M. Madkour, M. A. Hossain, and K. P. Dahal, “Evolutionary Optimization Approach for Missile Guidance Algorithms to Achieve Accurate Interception of Maneuvering Targets in 3D”,The IEE Seminar on Target Tracking: Algorithms and Applications, Birmingham, UK, pp: 99-104, 2006.