Intelligent Learning Algorithms for Active Vibration Control

Intelligent Learning Algorithms for Active Vibration Control

A. Madkour, M. A. Hossain, *Member, IEEE*,
K. P. Dahal, *Member, IEEE*, and H. Yu, *Member, IEEE*

***Abstract*—This correspondence presents an investigation into the comparative performance of an active vibration control (AVC) system using a number of intelligent learning algorithms. Recursive least square (RLS), evolutionary genetic algorithms (GAs), general regression neural network (GRNN), and adaptive neuro-fuzzy inference system (ANFIS) algorithms are proposed to develop the mechanisms of an AVC system. The controller is designed on the basis of optimal vibration suppression using a plant model. A simulation platform of a flexible beam system in transverse vibration using a finite difference method is considered to demonstrate the capabilities of the AVC system using RLS, GAs, GRNN, and ANFIS. The simulation model of the AVC system is implemented, tested, and its performance is assessed for the system identification models using the proposed algorithms. Finally, a comparative performance of the algorithms in implementing the model of the AVC system is presented and discussed through a set of experiments.**

***Index Terms*—Adaptive systems, fuzzy neural network, intelligent control, recursive estimation, system identification, vibration control.**

I. Introduction

This research investigates the design and development of intelligent learning algorithms for an active vibration control (AVC) system. It is reported earlier that the conventional online system identification schemes are, in essence, local search techniques [1]–[5]. These techniques often fail in the search for the global optimum if the search space is not differentiable or linear in the parameters. There have been many attempts to overcome this limitation using artificial intelligence [4]–[9]. This investigation employed recursive least squares (RLS), evolutionary genetic algorithms (GAs), general regression neural networks (GRNNs), and an adaptive neuro-fuzzy inference system (ANFIS) to identify the characteristics of the AVC algorithm for flexible beam control. A comparative performance of the algorithms is also explored to demonstrate their capabilities and merits in implementing an AVC system.

AVC is not a new concept. It is based on the principles that were initially proposed by Lueg in the early 1930s for noise cancellation [10]. Many attempts have been made since then at devising methods for tackling the problems arising due to unwanted structural vibrations (disturbances) consisting of passive and active control [2]–[5], [11]–[19]. Traditional methods of vibration suppression include passive control, which consist of mounting passive material on the structure [15]. On the other hand, AVC consists of artificially generating the cancelling source(s) to destructively interfere with the unwanted source, and thus, results in a reduction in the level of the vibration (disturbances) at desired location(s). This is realized by detecting and processing the vibration by a suitable electronic controller so that, when superimposed on the disturbances, cancellation occurs. Due to the broadband nature of the disturbances, it is required that the control mechanism in an AVC system realizes suitable frequency-dependent characteristics so that cancellation over a broad range of frequencies is achieved. In practice, the spectral contents of the disturbances as well as the characteristics of system components are, in general, subject to variation, giving rise to the time-varying phenomena. This implies that the control mechanism is further required to be intelligent enough to track these variations so that the desired level of performance is achieved and maintained [2].

A model of a flexible beam system in transverse vibration is considered in this correspondence. Such a system has an infinite number of modes, although in most cases the lower modes are the dominant ones requiring attention. The unwanted vibrations in the structure are assumed to be the result of a single-point disturbance of the broadband nature. First-order central finite difference (FD) methods are used to study the behaviors of the beam and develop a suitable test and verification platform. An AVC system is designed utilizing a single-input single-output control structure to yield optimum cancelation of broadband vibration at a set of observation points along the beam. The controller design relations are formulated so as to allow online design and implementation [15].

The flexible beam system, as discussed later, is considered as the plant model. The traditional RLS filter, evolutionary GAs, GRNN, and the ANFIS algorithms of the MATLAB tool boxes are used to estimate the AVC system canceling signal based on the input and corresponding output of the plant model. This is realized by minimizing the prediction error of the actual plant output and the corresponding model output. The AVC system is designed for optimum cancellation of broadband vibration along the beam. It is then implemented and tested using RLS, GAs, GRNN, and ANFIS algorithm. The performances of the proposed algorithms in implementing the simulation model of the AVC system are assessed in the suppression of vibration along the beam. These are presented and discussed through a set of simulation results.

II. Flexible Beam System

Flexible structure systems are known to exhibit an inherent property of vibration when subjected to disturbance forces leading to component and/or structural damage. Such systems include a wide range of engineering applications, e.g., skyscrapers and bridges in civil engineering applications; propellers, aircraft fuselage and wings, satellite solar panels, and helicopter blades in aerospace structures; and turbo generator shafts, engines, gas turbine rotors, and electric transformer cores in electromechanical systems [14].

In this correspondence, we consider a simulation model for a cantilever beam 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. A schematic diagram of this cantilever beam system is shown in Fig. 1.

The motion of the beam in transverse vibration is, thus, governed by the well-known fourth-order partial differential equation (PDE) [14], [18]

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

where μ is a beam constant given by $\mu^2 = EI/\rho A$, with ρ, 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

Manuscript received September 30, 2005; revised June 8, 2006. This correspondence was recommended by Associate Editor G. Feng.

A. A. Madkour, M. A. Hossain, and K. P. Dahal are with the Department of Computing, University of Bradford, Bradford BD7 1DP, U.K. (e-mail: Madkour@hotmail.co.uk; m.a.hossain1@bradford.ac.uk; k.p.dahal@bradford.ac.uk).

H. Yu is with the Faculty of Computing, Engineering and Technology, Staffordshire University ST18 0DG, U.K. (e-mail: h.yu@staffs.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2007.900640

1094-6977/$25.00 © 2007 IEEE

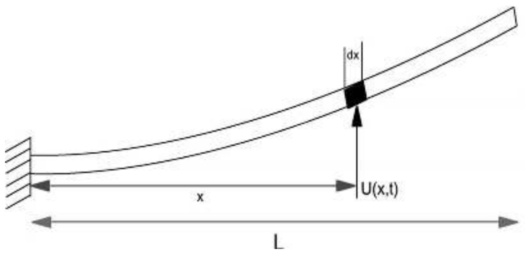


Fig. 1. Schematic diagram of the cantilever beam system.

free ends of the beam are given by

$$y(0,t)=0 \qquad \frac{\partial y(0,t)}{\partial x}=0$$
$$\frac{\partial^2 y(L,t)}{\partial x^2}=0 \qquad \frac{\partial^3 y(L,t)}{\partial x^3}=0. \tag{2}$$

It is to be noted that the model incorporates no damping. To construct a suitable platform for test, a method of obtaining numerical solution of the PDE in (1) is required. This can be achieved by using the FD method. This involves a discretization of the beam into a finite number of equal-length sections (segments), each of length Δx, and considering the beam motion (deflection) for the end of each section at equally spaced time steps of duration Δt. Thus, first-order central FD methods are used to approximate the partial derivative terms in (1) and (2) yielding [1]

$$Y_{j+1} = -Y_{j-1} - \lambda^2 S Y_j + (\Delta t)^2 U(x,t)\frac{1}{m} \tag{3}$$

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 discretization steps Δt and Δx, and $\lambda^2 = (\Delta t)^2(\Delta x)^{-4}\mu^2$. Equation (3) is the required relation for the simulation algorithm, characterizing the behavior 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 $0 < \lambda^2 \leq 0.25$ [1].

III. AVC System

A schematic diagram of an AVC structure is shown in Fig. 2. An unwanted (primary disturbance) point source emits broadband disturbance into the structure. This is detected by a detector, processed by a controller of suitable transfer characteristics, and fed to a canceling (secondary) point actuator. The secondary (control) signal thus generated interferes with the disturbance to achieve a reduction in the level of vibration at an observation point along the structure.

To allow the development of an AVC algorithm, consider the system in Fig. 2 with the detected signal U_M as input, and the observed signal Y_O as output. For complete cancellation of the disturbance to be achieved at the observation point, the signal $Y_O = 0$ must be forced to become zero. This is equivalent to the minimum variance design criterion in a stochastic environment. This requires the primary and secondary signals at the observation point to be equal in amplitudes and have a phase difference of 180° relative to one another. The work by Tokhi and Leitch of synthesizing the controller on the basis of this objective will yield the required controller transfer function as [1], [15].

$$C = [1 - Q_1/Q_0]^{-1} \tag{4}$$

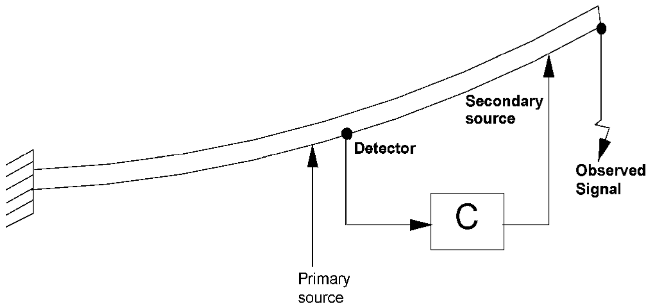


Fig. 2. Active vibration control structure.

where Q_0 and Q_1 represent the equivalent transfer functions of the system (with input at the detector and output at the observer) when the secondary source is *off* and *on*, respectively. Equation (4) is the required controller design rule, which can easily be implemented online. This will involve estimating Q_0 and Q_1, using a suitable system-identification algorithm, designing the controller using (4), and implementing the controller to generate the control signal. Equation (4) reflects that the order of the controller is twice of the original plant model which leads to estimate a large number of parameters. In this investigation, the RLS, GAs, GRNN, and ANFIS algorithms are used as online system identification algorithms to estimate the AVC system canceling signal itself, based on the input and corresponding output of the plant model. The designed AVC system can be tuned to keep it stable in case of any uncertainty, noise, or disturbance, using the self-tuning mechanism reported in [15]. These algorithms are briefly described in the following sections to demonstrate the computational merits and complexity.

IV. RLS Algorithm

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 [20]

$$\hat{\theta}(k) = f(\hat{\theta}(k-1), D(k), k) \tag{5}$$

where $D(k)$ denotes the 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 [21], [22]. For a multiparameter system, this form can be represented as

$$Y(z) = \frac{1 + b_1(z^{-1}) + \cdots + b_m(z^{-m})}{1 + a_1(z^{-1}) + \cdots + a_m(z^{-m})} U(z) \tag{6}$$

where Y, U, and m are the system input, output, and order, respectively. or

$$y(k) = \psi^T(k)\,\theta \tag{7}$$

where

$$\psi(k) = [-y(k-1), \ldots, -y(k-m), u(k), \ldots, u(k-m)]^T \tag{8}$$

and

$$\theta = [a_1, \ldots, a_m, b_1, \ldots, b_{m+1}]^T. \tag{9}$$

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

$$J(r) = \sum_{k=1}^{r} \left(y(k) - \psi(k)^T \hat{\theta}(r)\right)^2. \tag{10}$$

Equation (10) can be written in a recursive form as [17]

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\Psi(k)(y(k) - \mathbf{\Psi}^T(k)\hat{\theta}(k-1)) \tag{11}$$

where

$$P(k) = \left[P(k-1) - \frac{P(k-1)u(k)\psi^T(k)P(k-1)}{1+\psi^T(k)P(k-1)\psi(k)}\right]. \tag{12}$$

V. Intelligent Algorithms

The conventional online system identification schemes are, in essence, local search techniques. These techniques often fail in the search for the global optimum if the search space is not differentiable or linear in the parameters. On the other hand, these techniques do not iterate more than once on each datum received. In contrast, as mentioned earlier, real-time estimation scheme requires an updated parameter within the time span between successive samples [23]. An alternative strategy using artificial intelligence algorithm could provide a better solution. To achieve this goal, four most commonly used algorithms are explored to demonstrate their capabilities. These are described as follows.

A. GA and Operators

Over the last decade, GAs have been extensively used as search and optimization tools in various problem domains. The GA simultaneously evaluates many points in the parameter space and converges toward the global solution. The GA differs from other search techniques in the use of concepts taken from natural genetics and evolution theory [24], [25]. According to Goldberg [26], GAs are different from the more normal optimization and search procedures in the following 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 [25]. The GA iteratively uses selection, crossover, and mutation operators for the population evolution. The selection operator selects solutions to be parents based on their fitness. A 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 the new population for the next generation.

Here, we report the applications of two GAs. The first one uses a traditional crossover, which is referred to as a traditional crossover GA (TCGA); and the second one uses a new crossover operator as described in Section V-A1.

The GAs consider the same multiparameter system given by (7). Binary strings are used to represents a solution (individual). The fitness function for the system can be defined as [4]

$$f(e) = \sum_{k=1}^{r} |y(k) - \hat{y}(k)| \tag{13}$$

where $y(k)$ is the measured output, $\hat{y}(k)$ is the estimated model output, and r is the number of sets of measurement considered. Equation (13) may be written in vector form as

$$f(e) = \sum_{k=1}^{r} \left|y(k) - \hat{\theta}_0\hat{\psi}(k)\right|. \tag{14}$$

TABLE I
Example of Two Parents Having an Eight-Gene Chromosome

Gene no.	1	2	3	4	5	6	7	8
Parent 1	0	1	1	0	0	1	1	0
Parent 2	0	0	1	1	0	1	0	1

TABLE II
Reproduction in Table I Using RCGA

Gene no.	1	2	3	4	5	6	7	8
Child 1	0	1	1	1	0	1	0	0
Child 2	0	1	1	0	0	1	1	0
Child 3	0	0	1	0	0	1	0	1
Child 4	0	0	1	1	0	1	1	1

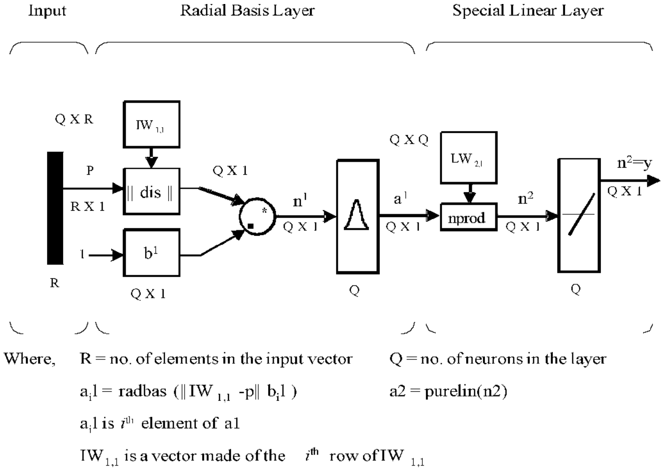

Input
Radial Basis Layer
Special Linear Layer
Q X R
IW 1,1
P
R X 1
|| dis ||
1
b1
R
Q X 1
Q X 1
n1
Q X 1
a1
Q X 1
Q
Q X Q
LW 2,1
nprod
n2
Q X 1
Q
n2=y
Q X 1
Where, R = no. of elements in the input vector
Q = no. of neurons in the layer
a_i1 = radbas (||IW 1,1 -p|| b_i1)
a2 = purelin(n2)
a_i1 is i^th element of a1
IW 1,1 is a vector made of the i^th row of IW 1,1

Fig. 3. Basic GRNN structure.

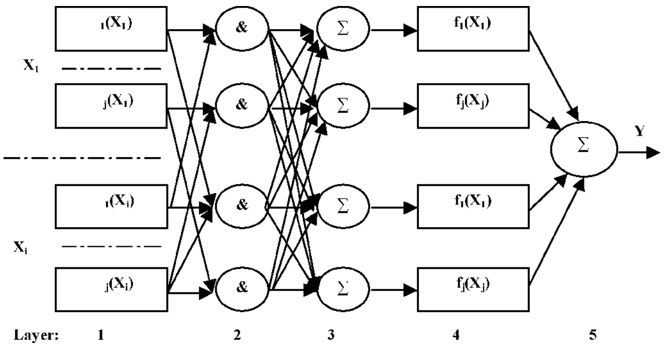


Fig. 4. Basic ANFIS structure.

The TCGA in our application uses multipoint crossover operator. The TCGA works on randomly selected pairs of solutions from the mating pool with a certain crossover rate. This operation exchanges the genes between the two selected solutions. This operation is uncertain, and may take time to converge to the global minimum [26]–[28]. This is the main motivation to develop a new crossover to use with the recessive trait crossovers genetic algorithm (RCGA).

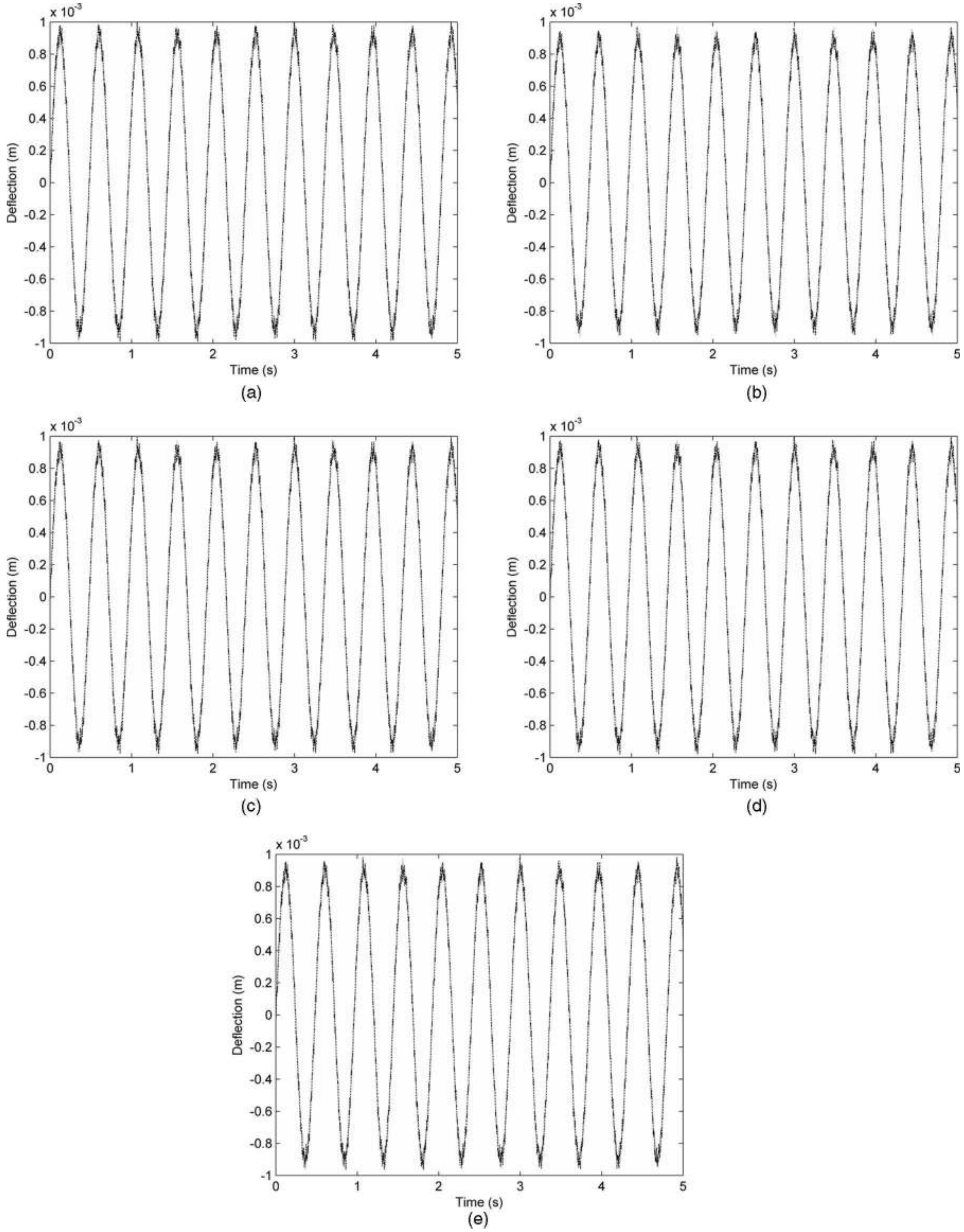


Fig. 5. (a) Performance of the RLS algorithm. (b) Performance of the TCGA. (c) Performance of the RCGA algorithm. (d) Performance of the GRNN algorithm. (e) Performance of the ANFIS algorithm.

1) RCGA: In the Nineteenth Century, Darwin originated his theory of evolution [29]–[31]. Darwin suggested that in the universal struggle for life, nature “selects” those individuals who are best suited (fittest) for the struggle, and these individuals in turn reproduce more than those who are less fit, thus changing the composition of the population.

There are three methods of population inheritance: dominant, recessive, and sex linked [32]. The expression of sex-linked properties depend on the person’s sex. For dominant properties, only one genetic trait is needed for this property to be expressed. However, if a genetic trait is recessive, a person needs to inherit two copies of the gene for

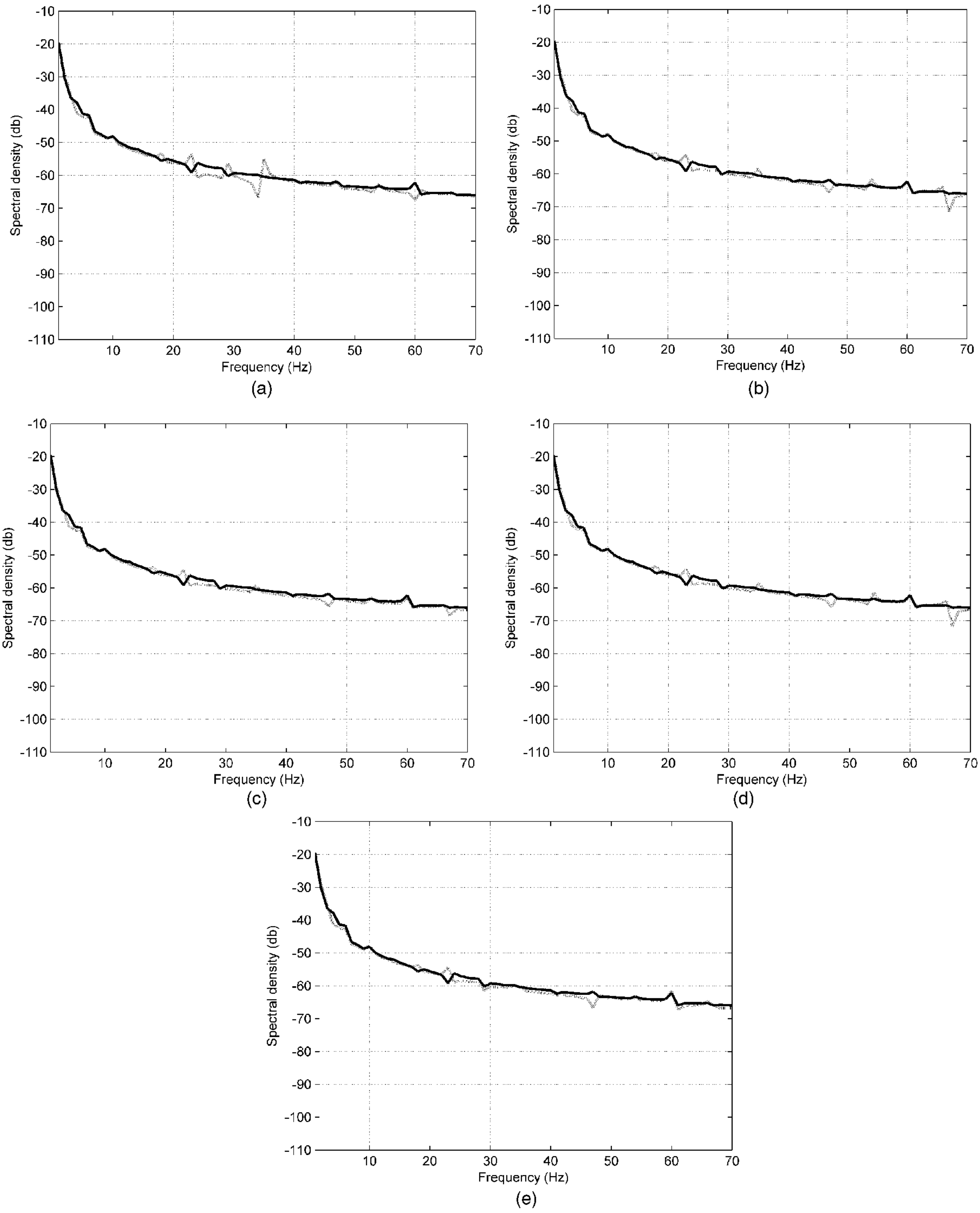


Fig. 6. (a) Performance of the RLS algorithm in auto-power spectral density. (b) Performance of the TCGA in auto-power spectral density. (c) Performance of the RCGA algorithm in auto-power spectral density. (d) Performance of the GRNN algorithm in auto-power spectral density. (e) Performance of the ANFIS algorithm in auto-power spectral density.

the trait to be expressed. Thus, both parents have to be carriers of a recessive trait in order for a child to express that trait. If either of the parents is a carrier, there is a 25% chance of each child showing the recessive trait, and it becomes 100% if both have that recessive trait.

Using the concepts taken from the recessive property inheritance, a crossover operator has been developed. Here, the GA with this operator is called RCGA. The RCGA produces children by selecting the common genes between parents, and choosing the remaining genes randomly. The main difference between the TCGA and RCGA is the way of how the new population is inherited from the previous generations. To use the proposed population inheritance approach through the recessive trait crossover we assume that the complementary of all of the chromosome parts makes its survival fitness for a fixed length of the chromosomes.

Now, let us assume that two parents have the eight genes chromosome as shown in Table I. It is worth noting that those parents have common genes at (1, 3, 5, and 6). According to the Darwin theory, those two parents' struggle for the fittest depends on these common genes; thus, we can keep these genes without any change when we produce their children to fit in the same struggle, and try to make their children of better fitness by the crossover of different genes using the four possible binary combinations randomly. This is the only random operation in our TCGA. The new solutions will be as shown in Table II.

Referring to recessive trail behavior, the selection of the parents for mating is very important. We have used a selection operation, which sorts the old population according to their fitness, and then, mate the first parent with the second to generate four new children, as shown in the above example, and so on.

The overall algorithm can be written as follows.

Step 1) Create a random population of N individuals.

Step 2) Evaluate their fitness.

Step 3) Sort the individuals in the population according to their fitness.

Step 4) Choose the best $N/2$ individuals as mating pool to generate new population.

Step 5) Generate four new individuals by reproducing the nearest two parents from the mating pool, keeping the common genes and randomly swapping the different genes. This creates a new population of N individuals.

Step 6) Apply mutation operation with a probability.

Step 7) Repeat Steps from 2 to 6 for the best fitness value.

B. GRNN

Use of neural network (NN) techniques to solve the problems in guidance began in the late 1990s by Song and Tahk [33]. The basic idea of NN midcourse guidance is to train NNs 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 [34].

The GRNN is the NN architecture that can solve any function approximation problems in the sense of estimating a probability distribution function [35]. It is a powerful memory-based network that could estimate continuous variables, and converges to the underlying regression surface [36].

The main advantage of the GRNN approach is its 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 [37].

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 [38].

GRNN is a feed-forward NN established on the theory of nonlinear regression (Fig. 3). It is a three-layer network with one hidden layer [35]. Each layer has entirely different roles: 1) the input layer, where the inputs are applied; 2) 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); 3) the linear output layer, where the outputs are produced.

C. ANFIS

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 as that of the NNs. 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 [39]–[41].

TABLE III
ERROR CONVERGENCE AND CORRESPONDING EXECUTION TIME IN IMPLEMENTING THE ALGORITHMS

Algorithm	Error	Time (s)
RLS	0.2465	0.301
TCGA	0.2412	0.475
RCGA	0.2174	0.425.
GRNN	0.2395	0.481
ANFIS	0.2383	0.432

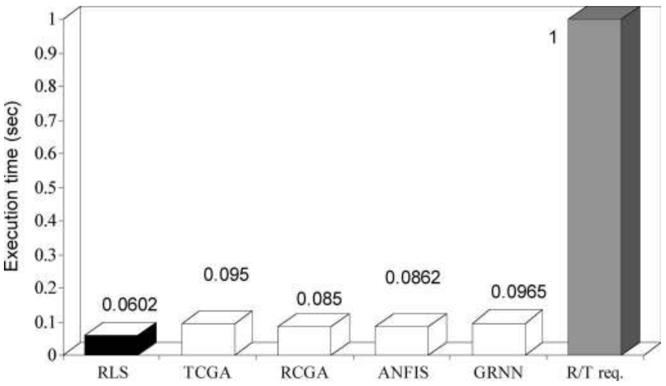


Fig. 7. Execution time of the algorithms relative to the real-time requirement.

Fig. 4 shows the basic structure of the ANFIS algorithm for a first-order Sugeno-style fuzzy system. It is worth noting that Layer-1 consists of membership functions described by the generalized bell function

$$\xi(X) = (1 + ((x - c)/a)^{2b})^{-1} \tag{15}$$

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 Layer-4 is comprised of a linear combination of the inputs multiplied by the normalized firing strength w

$$Y = w(pX + r) \tag{15}$$

where p and r are adaptable parameters. Layer-5 is a 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.

VI. SIMULATION RESULTS

To demonstrate the algorithms and to study the behavior of the system, we will use a simulation model of an aluminum-type cantilever beam in transverse vibration as a platform for the investigation. This flexible beam and its AVC system has been used as a platform

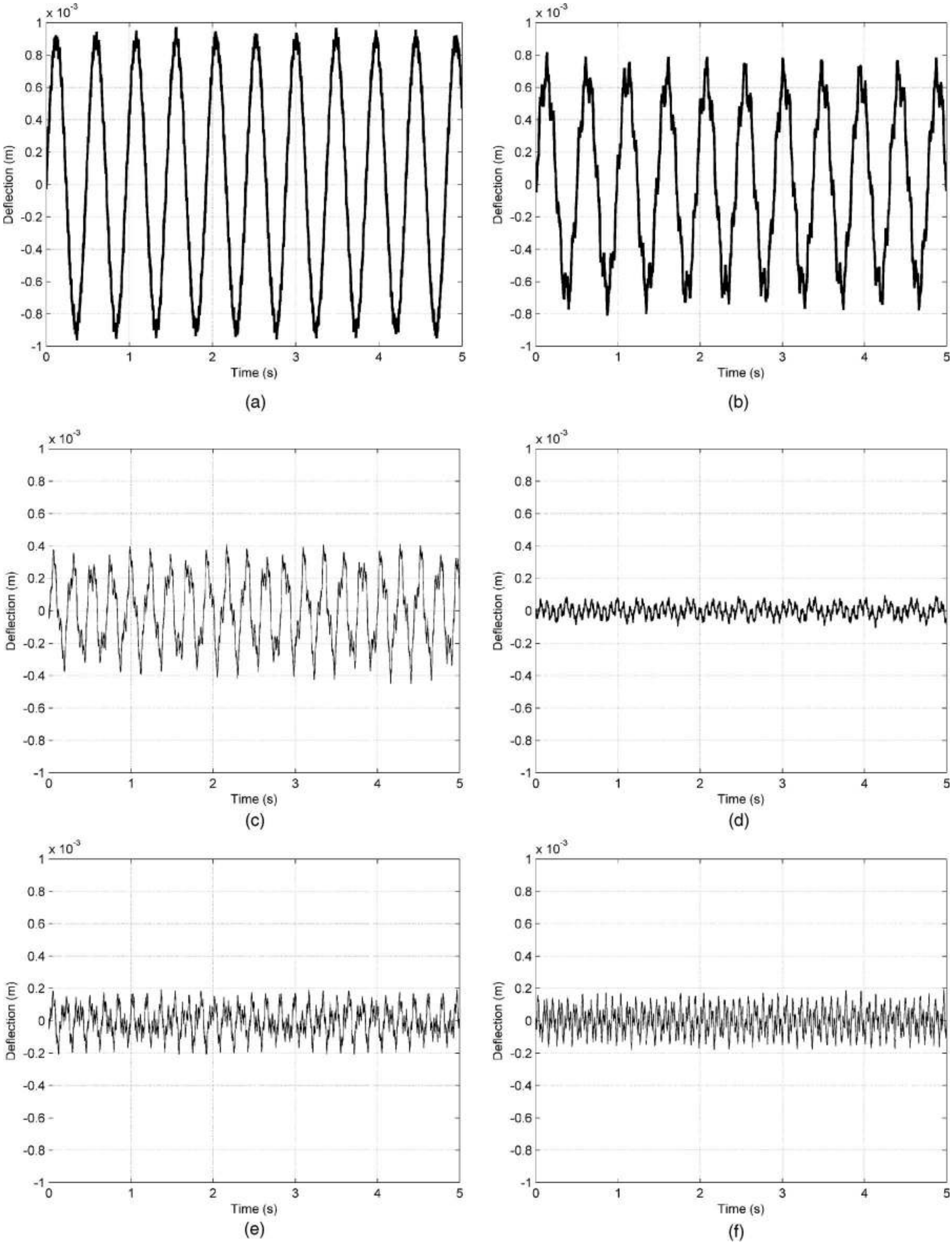


Fig. 8. (a) Beam fluctuation at the end point before cancellation. (b) Beam fluctuation at the end point after cancellation in implementing AVC using RLS. (c) Beam fluctuation at the end point after cancellation in implementing AVC using TCGA. (d) Beam fluctuation at the end point after cancellation in implementing AVC using RCGA. (e) Beam fluctuation at the end point after cancellation in implementing AVC using GRNN. (f) Beam fluctuation at the end point after cancellation in implementing AVC using ANFIS.

for many case studies using RLS and/or TCGAs [1]–[6], [15]–[19]. A comparative performance for using the RLS and TCGA has been reported earlier [19].

Our aluminum beam has the specifications of length as $L = 0.635\,\mathrm{m}$, mass $m = 0.037$ kg, beam constant $\mu = 1.351$. The first five resonance modes of this beam, as obtained through simulation exercise and verified by theoretical analysis, are located at 1.875, 11.751, 32.902, 64.476, and 106.583 Hz, respectively, with the first two modes being the dominant ones. The beam is divided into 20 small segments, and a sample period was selected as $\Delta t = 0.3\,\mathrm{ms}$ that satisfies the stability

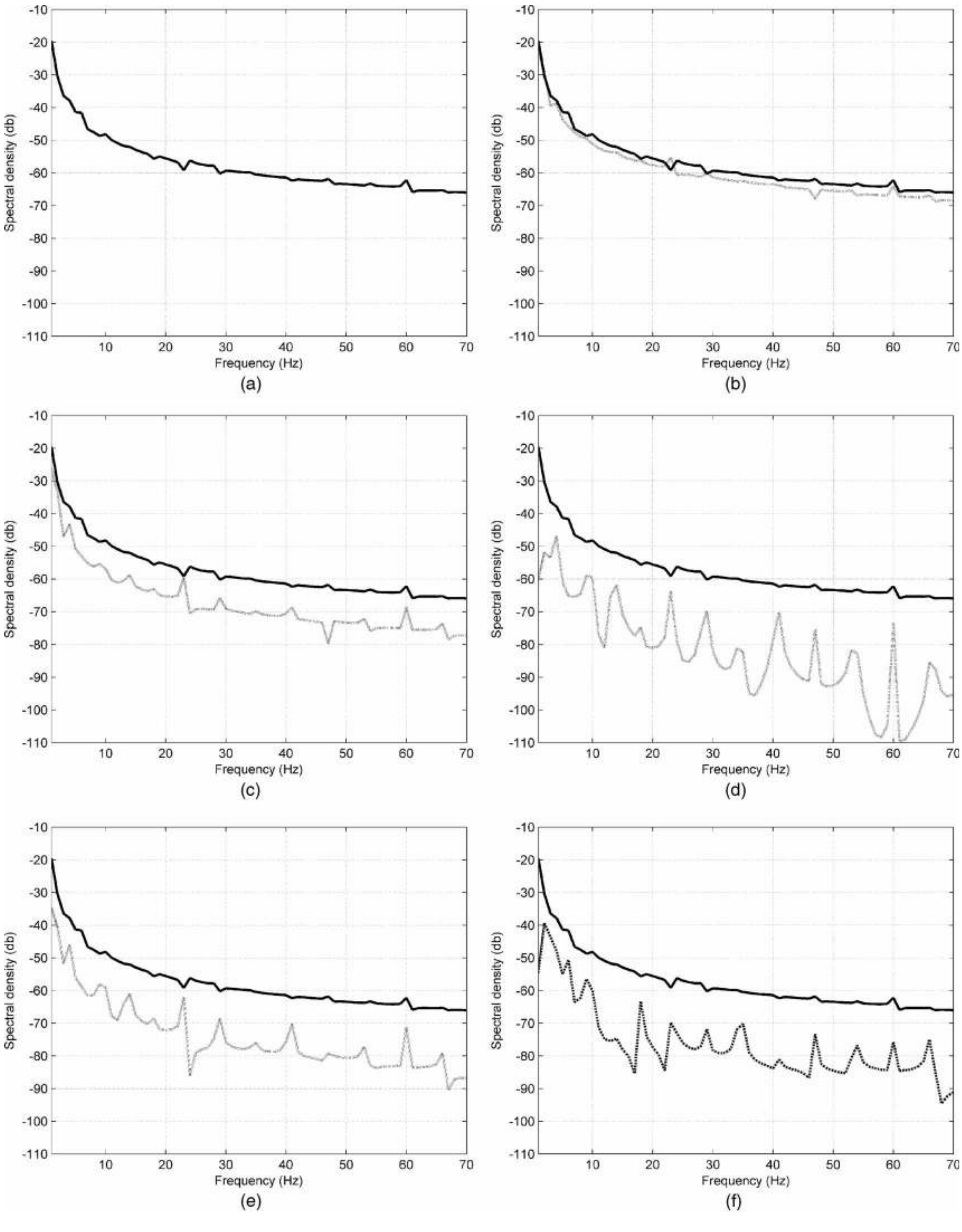


Fig. 9. (a) Auto-power spectral density at the end point before cancellation. (b) Auto-power spectral density at the end point (solid line represents before cancellation and dotted line represents after cancellation in implementing the AVC system using RLS). (c) Auto-power spectral density at the end point (solid line represents before cancellation and dotted line represents after cancellation in implementing the AVC system using TCGA). (d) Auto-power spectral density at the end point (solid line represents before cancellation and dotted line represents after cancellation in implementing the AVC system using RCGA). (e) Auto-power spectral density at the end point (solid line represents before cancellation and dotted line represents after cancellation in implementing the AVC system using GRNN). (f) Auto-power spectral density at the end point (solid line represents before cancellation and dotted line represents after cancellation in implementing the AVC system using ANFIS).

requirements of the FD simulation algorithm (3) and is sufficient to cover all the resonance modes of vibration of the beam. For the algorithm to be stable, this gives a value of $\lambda = 0.3629$. To allow dominant modes of vibration of the beam to be excited, a step disturbance force (0.1 N) of finite duration (0.3 ms) was applied as a primary force at grid point 16, and the control actuator signal is applied at grid point 20. The detector (sensor) and observer are placed at grid points 12 and 20 as input and output samples of the plant, respectively.

The traditional RLS filter, TCGA, RCGA, GRNN, and the ANFIS algorithms were used to estimate the parameters of the AVC system (4) based on the input and corresponding output of the plant model.

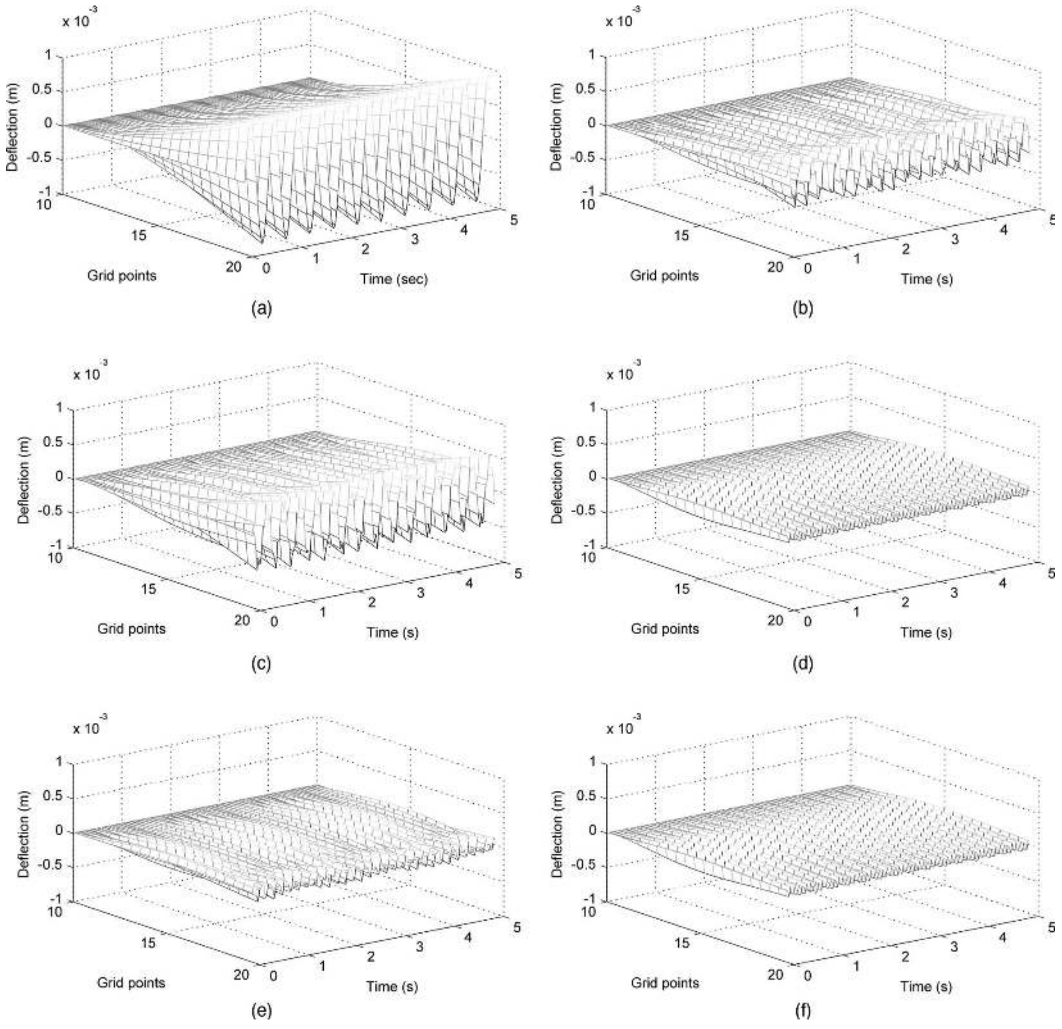


Fig. 10. (a) Beam fluctuation along its length before cancellation. (b) Beam fluctuation along its length after cancellation in implementing the AVC system using RLS. (c) Beam fluctuation along its length after cancellation in implementing AVC using TCGA. (d) Beam fluctuation along its length after cancellation in implementing the AVC system using RCGA. (e) Beam fluctuation along its length after cancellation in implementing the AVC using GRNN. (f) Beam fluctuation along its length after cancellation in implementing AVC using ANFIS.

A. System Identification Algorithms

A linear discrete second-order model was estimated using the RLS, TCGA, RCGA, GRNN, and ANFIS algorithms

$$Y(z) = \frac{1 + b_1(z^{-1} + b_2(z^{-2})}{1 + a_1(z^{-1} + a_2(z^{-2})} U(z). \qquad (16)$$

Investigations were carried out using MATLAB Genetic Algorithm Toolbox version 1.2 for TCGA, NN Toolbox version 4.0.4 for GRNN, and MATLAB Fuzzy Logic Toolbox version 2.2 for ANFIS.

The system identification algorithms were carried out for about 5 s (16 700 iterations) using the linear discrete second-order model (16) with grid points 12 and 20 as input and output samples of the plant, respectively, with a set of input-output data simulated using (3).

RLS, TCGA, and RCGA are used to estimate the parameters of the model of (16). 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, eight-bit representation, 10% mutation rate, and ten population sizes are the most suitable parameters of the TCGA and RCGA for best convergence. This also offers an acceptable time period for online system identification [3].

Fig. 5 shows the time-domain performance of the RLS, TCGA, RCGA, GRNN, and ANFIS algorithm, where the solid line represents actual output, and dotted line represents the estimated output of the model. It is observed that a significant level of convergence leads to almost overlapping of the two signals in each case. It is also noted that the ANFIS offers similar level of performance for convergence as compared to the other algorithms. Corresponding auto-power spectral density is shown in Fig. 6, which further demonstrated the level of convergence of the actual and estimated models. As in Fig. 5, the solid line in Fig. 6 represents actual output, and dotted line represents the estimated output of the model.

Table III summarizes the average error convergence for the 16 700 samples (about 5 s). It is worth noting that the error has been calculated based on the differences between the absolute value of the original and the estimated signal. From Table III, it is observed that the TCGA performs relatively better as compared to the RLS. ANFIS performance is relatively better as compared to that of TCGA and GRNN. The best performance for convergence among the algorithms is offered by the RCGA.

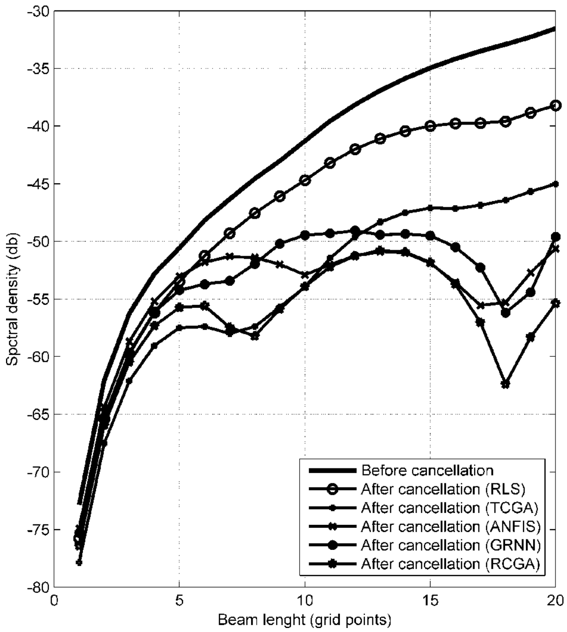


Fig. 11. Beam auto-power spectral density along its length before and after cancellation in implementing AVC using the five algorithms.

Table III also demonstrates the computation time of the algorithms for their corresponding convergence. It is noted that all the algorithms achieved real-time performance (5 s). Among the algorithms, RLS achieved the best real-time performance; in contrast, GRNN achieved the worst real-time performance, although the overall time difference is not very significant.

Fig. 7 shows the relative real-time performance of all the algorithms. This is calculated based on the ratio of the execution time of an algorithm to the real-time requirement (R/T req.). It is noted that RLS execution time is only 6.02% of the real-time requirement; in contrast, GRNN requires 9.65%. It is also observed that RCGA requires about 8.5%, in contrast TCGA requires 9.5%. This reflects that the new crossover approach RCGA is 1% faster than the traditional TCGA approach. From Table III and Fig. 7, it is concluded that the RCGA offers better and faster convergence as compared to the TCGA.

B. AVC System

As discussed earlier, the RLS, TCGA, and RCGA algorithms were used to estimate the parameters of the AVC system (4) based on the input and corresponding output of the plant model, while the ANFIS- and GRNN-based AVC system models were developed based on the input and the cancellation signal required for destructive interference at the control point. It is worth noting that the output signal with 180° phase shift is considered as the cancellation signal. The location of the input sensor and the output actuator has been selected for the best performance based on the previous investigation, as discussed in Section VI-A. The simulation model of the AVC system is then tested and validated through a set of experiments.

Fig. 8 shows the time-domain response at the end point of the beam before and after cancellation. Fig. 8(a) shows the vibration before suppression. Corresponding vibration after cancellation using different algorithms are shown in Fig. 8(b)–(f), respectively.

It is observed that the peak-to-peak amplitude at the end point before cancellation was +973.3 to 973.3 μm and reduced to +95.9 to −95.9 μm by RCGA, +192 to −192 μm by ANFIS, +194.7 to −194.7 μm by GRNN, +412.8 μm to −412.8 μm by TCGA, and +885.1 to −885.1 μm by RLS-based AVC system. It is noted in the identification section that RCGA offers the best convergence among all the algorithms. This, in turn, helps to identify the more accurate controller parameters and results best performance among all the algorithms in implementing the AVC system.

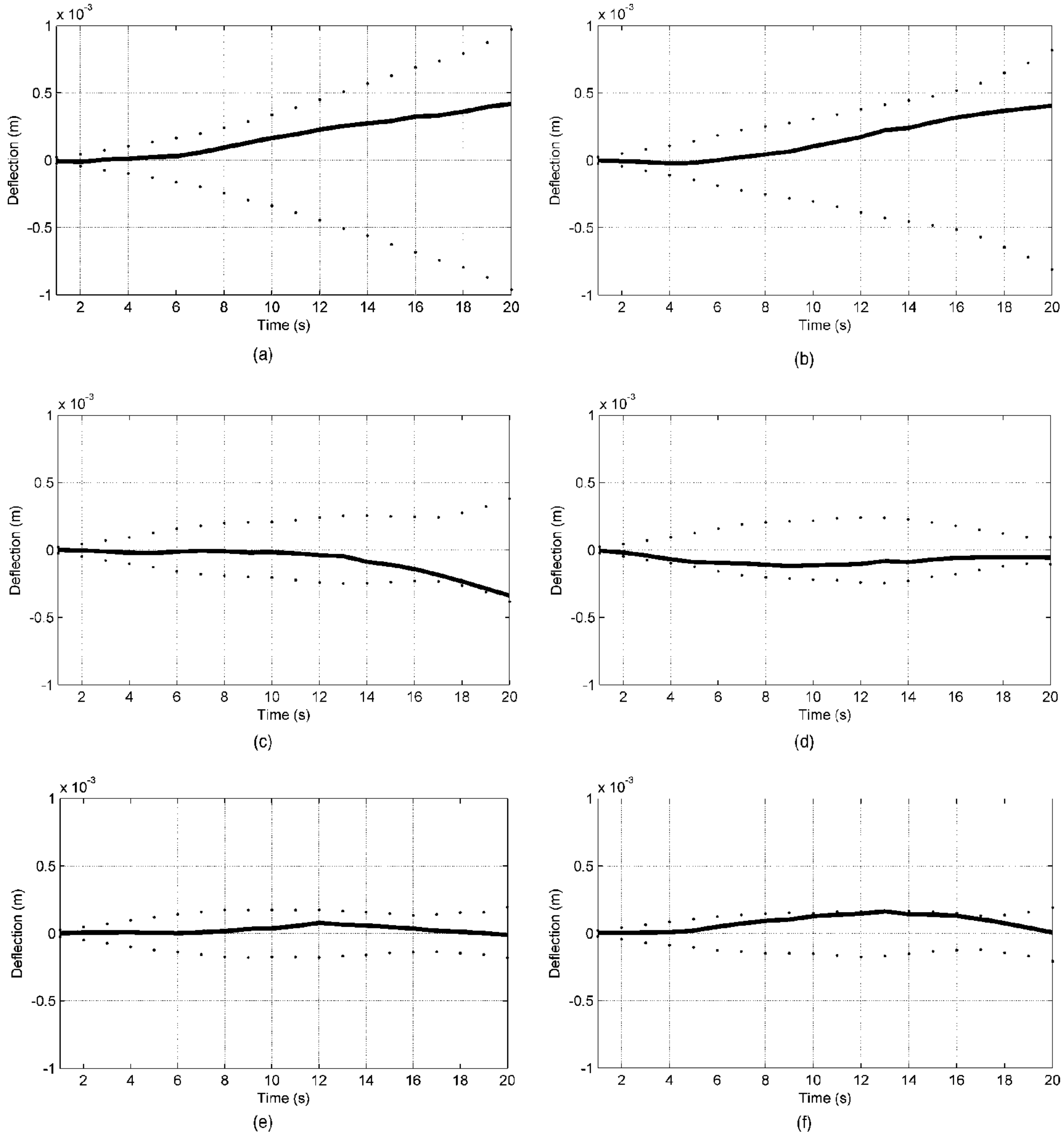
Fig. 9 further demonstrates the auto-power spectral density (in decibels) at the end point of the beam. Fig. 9(a) shows the auto-power spectral density before vibration suppression. Vibration cancellation in implementing the AVC system are shown in Fig. 9(b)–(f), respectively. The solid lines in these figures depict the auto-power spectral density before cancellation, and the dotted lines depict the auto-power spectral density after cancellation. It can be noted that a significant level of reduction is achieved by RCGA for the first resonant frequency. A phenomenon of reasonable spillover impact [42] has been observed with all the algorithm as shown in Fig. 9. The spillover effect is high particularly at high frequencies due to noncollocated control scheme. This effect is generally less in the dominating low resonant frequencies for cancellation. However, the overall energy level after cancellation for all resonant frequencies using AVC system has been reduced. ANFIS and GRNN achieved similar level of performances along the range of frequencies. However, spillover effect at higher frequencies is relatively lower as compared to the RCGA. The RLS offers worst performance among the algorithms. This further reflects the worst system identification performance of the RLS algorithm.

Fig. 10 shows the 3-D time-domain performance along the length of the beam. Fig. 10(a) shows vibration suppression before cancellation. The corresponding vibration suppression in implementing the AVC system are shown in Fig. 10(b)–(f), respectively. It is noted that a significant level of vibration suppression is achieved by RCGA and ANFIS algorithm along the beam length. A minor fluctuation is visible in the middle of the beam. Again, RLS achieved the worst performance.

Fig. 11 further demonstrates the relative beam auto-power spectral density performances along its length before and after cancellation in implementing AVC using the five algorithms. It shows that the RCGA offered the best performance among all the algorithms. It is observed that the performances of the algorithms are not linear along the length of the beam length. This may be due to the location of the disturbance force and control actuator. It is worth noting that the resonance frequencies at some nodes are more dominant than others [16]. Therefore, the cancellation of the energy level may vary and could cause a nonlinear shape of energy level along the beam length.

Fig. 12 demonstrates the overall range of fluctuations for the beam in the time domain along the grid points in 2-D form. The dotted line in the figure depicts the maximum and minimum amplitude of the deflection along the grid points after cancellation, and the solid line depicts the amplitude of deflection of each algorithm at a random sampling period. It is further evidenced that overall performance of the RCGA and ANFIS are better as compared to the other algorithms.

The simulation results and convergence analysis show that the RCGA and ANFIS performed consistently better than the other algorithms. We believe that the learning aspect of these two algorithms play a significant role during the simulation process. Using a given input/output data set, ANFIS constructs FIS whose membership function parameters are tuned (adapted) using either a backpropogation algorithm alone or in combination with least square type of method. The proposed RCGA, using a new population inheritance approach through the recessive trait crossover, offers the propagation of good building blocks of genes to subsequent generations (iterations). These features of the ANFIS and RCGA, we believe, offered a significant impact in providing better results and convergence.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 12. (a) Beam amplitude after cancellation. (b) Beam amplitude before cancellation using RLS. (c) Beam amplitude before cancellation using TCGA. (d) Beam amplitude before cancellation using RCGA. (e) Beam amplitude before cancellation using GRNN. (f) Beam amplitude before cancellation using ANFIS.

VII. Concluding Remarks

This correspondence has presented an investigation into the development of intelligent adaptive learning algorithms for an AVC system using the evolutionary RLS, TCGA, RCGA, GRNN, and ANFIS. A model of a flexible beam system in transverse vibration was considered as the plant for testing and validation of the AVC system. MATLAB Toolboxes were used for AVC system design based on RLS, TCGA, RCGA, GRNN, and ANFIS. The simulation model of the AVC system has been implemented and verified to demonstrate the merits and capabilities of the learning algorithms through a set of experiments. A comparative performance in implementing the AVC system using RLS, TCGA, RCGA, GRNN, and ANFIS has been presented and discussed.

It is noted that the RCGA- and ANFIS-based learning algorithms offer relatively better performance as compared to the other algorithms for system identification. This leads to a significant level of vibration cancellation by RCGA- and ANFIS-based AVC systems at the lower resonant mode. However, the GRNN-based AVC system shows relatively better performance at higher resonant modes. In general, RCGA- and ANFIS-based AVC systems offered better performance as compared to the other algorithms.

Finally, it is worth noting that this investigation has given attention to cancel the vibration of the dominant low resonant frequencies. However, it is observed that there is a spillover effect [42] at higher frequencies for a number of AVC algorithms. This requires an attention to improve the performance further, which can be considered as a dimension for future research. Subsequently, real-rig-based implementation is also considered as one of the main future investigations.

Acknowledgment

The authors would like to thank the EPSRC research council (research grand EP/E025250/1) for the support of this research.

References

[1] M. O. Tokhi and R. R. Leitch, *Active Noise Control*. Oxford, U.K.: Clarendon, 1992.

[2] M. O. Tokhi and M. A. Hossain, "Self-tuning active vibration control in flexible beam structures," *Proc. IMechE-I: J. Syst. Control Eng.*, vol. 208, pp. 263–277, 1994.

[3] M. O. Tokhi, M. A. Hossain, and M. H. Shaheed, *Parallel Computing for Real-time Signal Processing and Control*. Berlin, Germany: Springer, 2002, pp. 141–150.

[4] M. A. Hossain and M. O. Tokhi, "Evolutionary adaptive active vibration control," *Proc. IMechE-I: J. Syst. Control Eng.*, vol. 211, pp. 183–193, 1997.

[5] A. A. M. Madkour, M. A. Hossain, K. P. Dahal, and H. Yu, "Real-time system identification using intelligent algorithms," in *Proc. IEEE Conf. Intell. Cybern. Syst.*, Northern Ireland, U.K., 2004, pp. 236–241.

[6] K. Yuasa, K. Takao, T. Yamamoto, and T. Hinamoto, "System modeling using GA and control for nonlinear systems," in *Proc. 2003 Congr. Evol. Comput.*, vol. 22003, pp. 1178–1185.

[7] Z. Hou, Q. Shen, and H. Li, "Nonlinear system identification based on ANFIS," in *Proc. Int. Conf. Neural Netw. Signal Process., 2003*, vol. 1, pp. 510–512.

[8] A. Sakaguchi and T. Yamamoto, "A study on system identification using GA and GMDH network," in *Proc. IEEE 29th Annu. Conf. 2003*, vol. 3, no. 2–6, pp. 2387–2392, Nov.

[9] K. Abed-Meraim, W. Qiu, and Y. Hua, "Blind system identification," *Proc. IEEE*, vol. 85, no. 8, pp. 1310–1322, Aug. 1997.

[10] P. Lueg, "Process of silencing sound oscillations," U.S. Pat. Off., Washington, DC, U.S. Patent 2,043,416,1936.

[11] S. Elliott, I. M. Stothers, and P. A. A. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 10, pp. 1423–1434, Oct. 1987.

[12] S. J. Elliott, P. A. Nelson, and C. C. Fuller, *The Active Control of Vibration*. London, U.K.: Academic, 1997.

[13] A. Baz and S. Poh, "Performance of an active control system with piezoelectric actuators," *J. Sound Vibr.*, vol. 126, pp. 327–343, 1988.

[14] P. K. Kourmoulis, "Parallel processing in the simulation and control of flexible beam structure systems" Ph.D. thesis, Dept. Autom. Control Syst. Eng., Univ. Sheffield, U.K., 1990.

[15] M. O. Tokhi and R. R. Leitch, "Design and implementation of self-tuning active noise control systems," *Proc. Inst. Electr. Eng.-D*, vol. 138, pp. 421–430, 1991.

[16] M. A. Hossain, "Digital signal processing and parallel processing for real-time adaptive active vibration control" Ph.D. thesis, Dept. Autom. Control Syst. Eng., Univ. Sheffield, U.K., 1995.

[17] M. O. Tokhi and M. A. Hossain, "A unified adaptive active control mechanism for noise cancellation and vibration suppression," *J. Mech. Syst. Signal Process.*, vol. 10, no. 6, pp. 667–682, 1996.

[18] S. Z. M. Hashim, M. O. Tokhi, and I. Z. Mat Darus, "Genetic adaptive active vibration control of flexible structures," in *Proc. IEEE SMC UK-RI Chap. Conf. Intell. Cybern. Syst.*, 2004, pp. 166–171.

[19] M. A. Hossain, A. A. M. Madkour, and K. P. Dahal, "Comparative performance of intelligent identification and control algorithms for a flexible beam vibration," in *Proc. Inst. Syst. Technol. Inf., Control Commun.*, Barcelona, Spain, 2005, vol. IV, pp. 364–367.

[20] H. Chen and J. Zhang, "Identification and adaptive control for systems with unknown orders, delay, and coefficients," *IEEE Trans. Autom. Control*, vol. 35, no. 8, pp. 866–877, Aug. 1990.

[21] L. Xia and J. B. Moore, "Recursive identification of over parameterized systems," *IEEE Trans. Autom. Control*, vol. 34, no. 3, pp. 327–331, Mar. 1989.

[22] L. C. Sing, W. K. Wo, S. P. Fai, and C. L. Wan, "On-line training and pruning for recursive least square algorithms," *Electron. Lett.*, vol. 32, no. 23, pp. 2152–2153, Nov. 7, 1996.

[23] K. Kristinsson and G. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 5, pp. 1033–1046, Sep./Oct. 1992.

[24] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: World Scientific, 1999.

[25] W. K. Lennon and K. M. Passino, "Genetic adaptive identification and control," *Eng. Appl. Artif. Intell.*, vol. 12, pp. 185–200, 1999.

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

[27] 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," in *Proc. IEE Semin. Target Track.: Algorithms Appl.*, Birmingham, U.K., 2006, pp. 99–104.

[28] A. A. M. Madkour, M. A. Hossain, and K. P. Dahal, "Modified crossover operator approach for evolutionary optimization," in *Proc. Colloq. Optimization Control*, Univ. Sheffield, U.K., Apr. 2006, to be published.

[29] G. Himmelfarb, *Darwin and the Darwinian Revolution*. New York: Doubleaday, 1959.

[30] D. B. Fogel, *Evolutionary Computation, Toward a New Philosophy of Machine Intelligence*. Piscataway, New Jersey: IEEE, 1995.

[31] Muscular Dystrophy Canada. [Online]. Available: http://www.muscle.ca/content/index.php?id=315, accessed Dec. 2002.

[32] E. Richard and M. D. Robert, *Nelson Essentials of Paediatrics*, 3rd ed. Philadelphia, PA: Saunders, 1998, pp. 129–146.

[33] M. J. Tahk, (May 2000). [Online]. Available: http://fdcl.kaist.ac.kr/supra/research/s_nn_guidance.html

[34] E. J. Song, H. Lee, and M. J. Tahk, "On-line suboptimal midcourse guidance using neural networks," in *Proc. 35th SICE Annu. Conf.*, Tottori Univ., Japan, 1996, pp. 1313–1318.

[35] W. Kanbuaa, S. Supharatidb, and I. Tan, "Ocean wave forecasting in the Gulf of Thailand during Typhoon Linda 1997: WAM and neural network approaches," *J. Sci. Soc. Thailand, ScienceAsia*, vol. 31, no. 4, pp. 243–250, 2005.

[36] S. Kanmani, V. R. Uthariaraj, V. Sankaranarayanan, and P. Thambidurai, "Object oriented software quality prediction using general regression neural networks," *ACM SIGSOFT Softw. Eng. Notes*, vol. 29, no. 5, pp. 1–6, 2004.

[37] P. D. Wasserman, *Advanced Methods in Neural Computing*. New York: Van Nostrand, 1993, pp. 155–161.

[38] D. F Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[39] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[40] J. S. R. Jang and N. Gulley, *Fuzzy Logic toolbox Users Guide*. Natick, MA: Mathworks, 1995.

[41] J. S. R. Jang, N. Gulley, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. London, U.K.: Prentice-Hall, 1997, pp. 335–360.

[42] J. Hong and D. S. Bernstein, "Bode integral constraints, colocation, and spillover in active noise and vibration control," *IEEE Trans. Control Syst. Technol.*, vol. 6, no. 1, pp. 111–120, Jan. 1998.