Journal of
Marine Science and Technology

Volume 24 | Issue 3 Article 17

SOFT COMPUTING TECHNOLOGIES IN DESIGN OF FUZZY CONTROLLER FOR ACTIVE VIBRATION ISOLATION SYSTEMS

inn-Tong Chiu
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University., cjt7725@gmail.com

Chih-Chung Fang
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University.

Recommended Citation
Chiu, inn-Tong and Fang, Chih-Chung (2016) "SOFT COMPUTING TECHNOLOGIES IN DESIGN OF FUZZY CONTROLLER FOR ACTIVE VIBRATION ISOLATION SYSTEMS," *Journal of Marine Science and Technology*: Vol. 24: Iss. 3, Article 17.
DOI: 10.6119/JMST-015-1026-2
Available at: https://jmstt.ntou.edu.tw/journal/vol24/iss3/17

Journal of Marine Science and Technology, Vol. 24, No. 3, pp. 519-529 (2016)
DOI: 10.6119/JMST-015-1026-2

SOFT COMPUTING TECHNOLOGIES IN DESIGN OF FUZZY CONTROLLER FOR ACTIVE VIBRATION ISOLATION SYSTEMS

Jinn-Tong Chiu and Chih-Chung Fang

Key words: soft computing, artificial neural networks, fuzzy control theory, active vibration isolation systems.

ABSTRACT

In the paper, we develop a two-phase design approach for a fuzzy logic controller (FLC) of an active vibration isolation system, which adopts two components of soft computing techniques, namely fuzzy logic control theory and neural networks. First, we design a fuzzy logic controller for an active vibration isolation system using fuzzy logic control theory. We conduct numerical simulations of an active vibration isolation system with three kinds of exciting loads. The results indicate that the performance of the fuzzy logic controller is very good, but its control rule surface does not measure up in practical terms. Secondly in the neural network, trained by Quick-prop with Newton's method, is used to model an approximation of this fuzzy logic controller, termed the neural network fuzzy controller (NNFC). The control performance indicates that the NNFC has the same performance as the FLC designed in the first phase and its control rule surface is improved and more suitable for actual use. The results indicate that the proposed design approach provides a robust, controllable, practical, and low-cost solution for a FLC of an active vibration isolation system.

I. INTRODUCTION

The vibration isolation systems with higher performance are in great demand in many scientific and industrial fields such as integrated circuit (IC) manufacturing, engine mounts for automobiles and ships, etc. For precision machining and measuring, precision instruments and vehicle safety and comfort, it is essential that mechanical systems have strict constraints on vibration and shock in their environment. For this purpose, vibration control technologies are of great importance and have become a significant research topic.

Currently, vibration isolators and vibration absorbers are commonly employed in dynamic systems to reduce vibration (Cheung et al., 2015). This paper focuses on the vibration isolation method, i.e. an isolator that is installed between the ground and the main system with vibration excitation designed to reduce the transmission of the excitation loads. Passive vibration isolation is a typical example and the most common method used to satisfy this requirement. It uses springs, dampers, and other dynamic parameters to provide a controlled damping system and has the advantages of simple structure, effectiveness, reliability, and no additional equipment. However, its performance regarding isolation of low frequency and ultra-low frequency vibrations is not satisfactory. Moreover, given the fixed eigenvalues of dynamic systems, relatively slow response time, and lack of a flexible control method, this approach is not suitable for vibration control that requires high precision. In contrast, active vibration isolation enables closed-loop control of dynamic systems in random, external, and exciting load conditions, achieving a strong anti-interference ability, and therefore, a more robust isolation effect.

The vibration isolation controller is the kernel of an active vibration isolation system (AVIS). It not only receives the signal of the sensor, transports the signal of the actuator unit, but must also select the control method for the different actuator. So far, there are many frequently used classical control methods: Narendra et al. (1997), proposed the multiple model switching tuning (MMST) control method to tackle the problem of remarkable and rapid variation in plant parameters. Bai et al. (2002) studied control design of active vibration isolation using linear quadratic Gaussian (LQG) control and μ-synthesis. Engels et al. (2006), examined the performance of centralized and decentralized feedback controllers. These control methods have common characteristics: the controller is designed based on mathematical models and performance requirements of the controlled system, and the control method is described analytically using mathematics. In real words, it is not easy to establish a clear mathematical model for complex structural dynamic vibration systems with factors such as structural nonlinearities, uncertainties of structural models, and time-variant dynamic parameters. The established mathe-

Paper submitted 03/23/15; revised 07/08/15; accepted 10/26/15. Author for correspondence: Jinn-Tong Chiu (e-mail: cjt7725@gmail.com).
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University.

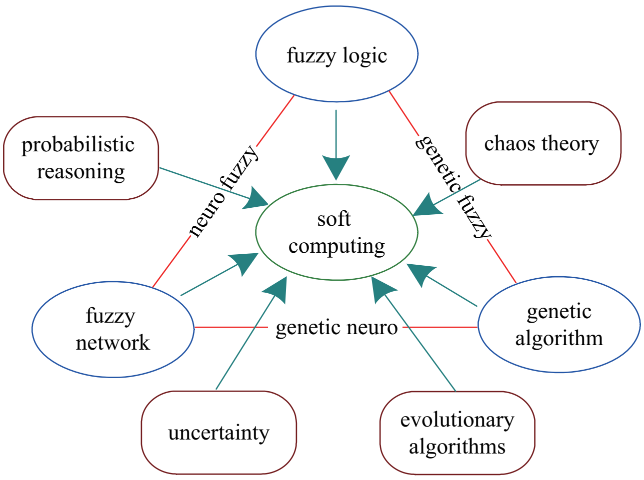


Fig. 1. Framework of soft computing.

matical models are often too complex and do not enable effective isolation control. These motivate us to consider intelligent control theory, also known as soft computing techniques, to propose an effective and feasible control technique for active vibration isolation systems.

Soft computing is a concept proposed by Zadeh (1994). It is not a single computing method, but rather a collection of various hybrid algorithms that simulate biological behaviour and thinking activities, as shown in Fig. 1. Elements of soft computing include artificial Neural Networks (NN), Fuzzy Systems (FS), Evolutionary Computation (EC), Swarm Intelligence and Chaos Theory.

Fuzzy logic control theory was proposed by Zadeh (1965) in his work on fuzzy set theory. Zadeh (1973) researched the soft computing technique. Mamdani (1976) applied fuzzy logic control theory in a study of simple control systems. The main steps for developing a fuzzy controller are: (1) fuzzification, (2) selection of the suitable linguistic control rule base, (3) design of a suitable control algorithm, and (4) defuzzification. Extensive research and application of fuzzy set theory and fuzzy logic controller has been conducted, and good results have been achieved. Song and Chen (2010) designed the fuzzy active vibration isolation control algorithm base on neural networks to forecast the vibration of next moving state. Ricardo et al. (2014) studied the fuzzy logic controller of non-singleton type for automatic design and tested the controller by using the trajectory tracking of an autonomous mobile. Lin et al. (2015) applied the design process of fuzzy logic-based algorithm to the seismic isolator system for structural control.

Early study on neural networks includes research by Rumelhart et al. (1986) and Hebb (1989). Hornik et al. (1990) proposed multi-layer feed-forward networks; they used neural networks to describe a model mapping input to output, which does not need a physical or mathematical model and is a type of universal approximator. Neural networks is an information processing paradigm that mimics the way the densely interconnected, parallel structure of the human brain processes information. In this domain, a popular fast learning algorithm known as back-propagation learning algorithm (Backprop) is trained by the steepest descent method (Rumelhart et al. 1986). Fahlman and Lebiere (1988) followed with a new learning process called Quickprop, a modification of Backprop that uses Newton's method, a second order weight update function, to accelerate the convergence over simple first-order (steepest) gradient descent.

Nastac (2008) developed a fuzzy logic controller for a vibration isolation device driver. In his work, the final remark he made was that the biggest difficulty in developing a fuzzy logic controller was setting the initial values for the linguistic rules base, which depends on the empirical experience of the designers; the computational complexity and the number of rules increase exponentially as the number of system variables increases. As a result, hybrid soft computing technology may be an important approach to improving control performance and reducing difficulties in developing the linguistic rules base of the FLCEker and Torun (2006) proposed PID-Fuzzy logic control in nonlinear industrial systems. Kim et al. (2006) used the genetic algorithm (GA) for optimization of the fuzzy logic controller. Their purpose in employing a GA was to determine appropriate fuzzy control rules and to adjust parameters of the membership functions. Cheong et al. (2007), applied differential evolution (DE) to the automatic design of a hierarchical fuzzy logic controller. The cost in computer time is currently the biggest concern for designers in using evolutionary computations (ECs) such as GAs, DEs, etc. to automatically develop the FLC.

In this paper, a two-phase design approach is proposed, which uses fuzzy logic control theory and neural networks to develop a fuzzy logic controller for active vibration isolation systems. In the first phase of the present approach, fuzzy logic control theory is adopted to develop a feasible FLC by using expert experiences through trial and error with limited time cost. In the second phase, we use neural networks to build an approximate model of the FLC (developed in the first phase) to ensure the necessary adjustment is proposed so that the imprecise prediction of the NN can be used to achieve a robust, controllable, practicable, and low-cost solution. The approximator of this fuzzy logic controller is called the neural network fuzzy controller (NNFC). In this paper, the Quickprop learning algorithm, which has good convergence speed, is used to train the neural networks. And in conclusion, the control performances of FLC and NNFC are highlighted and discussed using a numerical simulation of an active vibration isolation system.

II. FUZZY LOGIC CONTROLLER

There is increasing demand for precision machining and measuring, precision instruments and vehicle safety and comfort. The fuzzy logic controller can play an important role in meeting this need because knowledge-based control rules can easily be implemented in systems, and it is going to be a conventional control method since the control design strategy is simple and practical and is based on linguistic information.

Fuzzy logic control theory uses IF-THEN clauses, similar as in human experience, to express events. It has the following characteristics: (1) the use of fuzzy linguistic variables rather than numerical values and mathematical models to describe a system, (2) the use of conditional propositions to represent the characteristics of the system and its actions, and (3) the use of fuzzy inference methods to express control algorithms. The fuzzy logic controller (FLC) has four main parts: fuzzifier, fuzzy rule base, fuzzy inference engine and defuzzifier. These are set up as described:

(1) Select a fuzzifier (fuzzification): The fuzzifier maps crisp values of input variables selected from the controlled system into fuzzy sets, i.e. the input variables have to be transformed into linguistic variables, i.e. variables whose values are words or sentences in a natural language, and not a number. A fuzzy set is characterized by a linguistic variable. Every linguistic variable has to be expressed by a suitable membership function. The most commonly used functions are the triangular, Gaussian, trapezoidal and piecewise linear. The main function of the fuzzifier is to activate rules associated (through linguistic variab1es) with fuzzy sets.
(2) Set-up a fuzzy knowledge base: The fuzzy knowledge base consists of rule bases and databases. These bases are built using the expert experiences of the designer through trial and error.
(3) Design a fuzzy inference (control algorithm) based on the controlled system variables: Fuzzy inference is expressed in terms of linguistic variables. Depending on the input values, linguistic variables become active and the inference engine produces a fuzzy set for the output linguistic variables. The inference engine is the kernel of the fuzzy logic controller and it handles the manner in which rules are combined, representing the knowledge base of the system.
(4) Choose a defuzzifier (Defuzzification): The output fuzzy set inferred from fuzzy inference engine is given as input to a defuzzyfier, which transforms the set into crisp values to the controlled system.

The framework diagram of design a fuzzy logic controller is shown in Fig. 2.

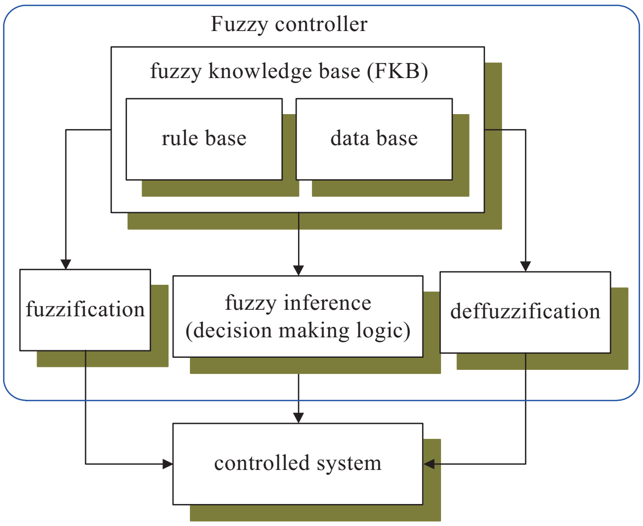


Fig. 2. Framework of fuzzy logic controller.

III. NEURAL NETWORKS

Neural networks (Rumelhart et al., 1986, Hebb, 1989) are intended to simulate the human brain which consists of a large number of neurons (10^{11}) interconnected via a larger number of synapses (10^{14}), resulting in a highly non-linear, fault-tolerant, and parallel processing system. Neural networks can be seen as information processing systems that simulate biological neural networks. To date, many types of neural net work architectures have been proposed (Hornik et al.,1990). In this paper, two popular neural networks, Back-propagation and Quickprop, are briefly described.

1. Back-Propagation Networks (BPN)

Back-propagation Networks (BPN) (Rumelhart et al., 1986), also known as Multi-Layer Perceptrons (MLP), are multilayer feed-forward networks. They are formed by an input layer, an output layer, and several hidden layers in between. A back-propagation algorithm is used to adjust the values of the network nodes. In the first phase, signals are fed from the input layer and passed through the hidden layers to the output layer in the feed-forward way. The output value is then calculated. The second phase is the back-propagation of adjustments. The values of network nodes are adjusted according to the delta rule, such that the network output value will converge to the expected value. This iterative process for updating node values is called network learning. The corresponding network architecture is shown in Fig. 3.

In a neural network, multiple hidden layers are used to increase its capacity for processing non-linear problems. In theory, if there are enough neurons in the hidden layers of the perceptron, a two-layer structure will be sufficient to enable the output of the perceptron to approximate any continuous function, and thus it becomes a "universal approximator" (Hornik et al., 1990).

An output of a neuron is a linear or nonlinear transfer function $\varphi_j(\cdot)$. Usually in the hidden layers it is a sigmoid function $y_j = \dfrac{1}{1+\exp(-v_j)}$, and in the output layer it is a linear transfer function.

The back-propagation algorithm is described below:

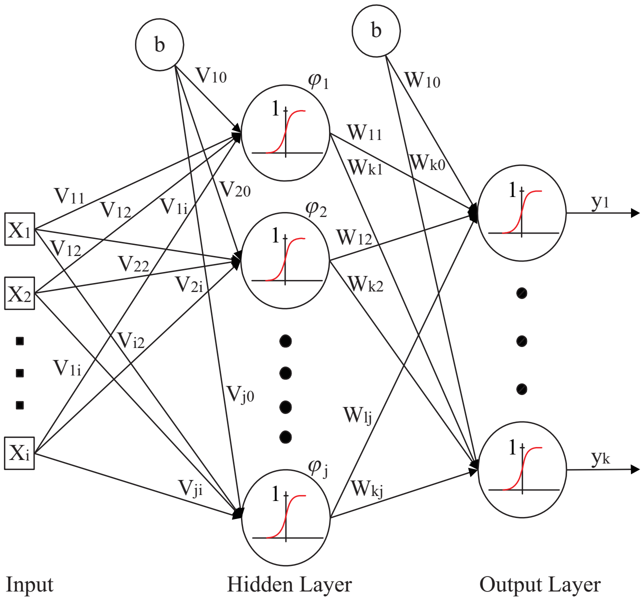


Fig. 3. Structure of back-propagation neural network.

1. Initial value setting: the setting of initial network structure (number of layers, number of neurons), transfer function, node weights, learning rate, learning cycles (Epochs) and the threshold.
2. Input of the sample set: samples are divided into training samples and test samples, used for training and testing the network, respectively.
3. Feed-forward computation: for each neuron, calculate in order the output value for the input vector of a training sample.

$$y_j(n) = \varphi_j(v_j(n)) \tag{1}$$

If the j-th neuron is in the first hidden layer, then $y_i(n) = x_i(n)$, otherwise, $y_i(n)$ is the output of the previous layer at the n-th epoch, where

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{2}$$

where $v_j(n)$ is called the transfer function. m is the neuron's input dimension and $w_{ij}(n)$ is the node weight.

4. Back-propagation: updating node weights.
   First, the definition of error is

$$e_j(n) = d_j(n) - y_j(n) \tag{3}$$

where $d_j(n)$ is the expectation.

Thus, in the n-th epoch, the total error function E or the mean square error, MSE, is expressed as

$$E(n) = \frac{1}{2} \sum_{j \in c} e_j^2(n) \tag{4}$$

or

$$MES(n) = \frac{E(n)}{N}$$

where c is the set of the output resulting from the training samples and N is the number of samples in the collection c.

Using the steepest descent method to adjust the node weights by $\Delta w_{ji}(n)$, the error gradient $\partial E(n) / \partial w_{ji}(n)$ can be expressed according to the chain rule:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{5}$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = \frac{\partial \left[\sum_{i=0}^{m} w_{ji}(n) y_i(n) \right]}{\partial w_{ji}(n)} = y_i(n) \tag{6}$$

The definition of error correction is

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} \tag{7}$$

Adjustment of weight $\Delta w_{ji}(n)$ is then

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{8}$$

where η is the learning rate.

Thus, the new weight on the network is

$$\begin{aligned} w_{ji}(n+1) &= w_{ji}(n) + \eta w_{ji}(n) \\ &= w_{ji}(n) + \eta \delta_j(n) y_i(n) \end{aligned} \tag{9}$$

5. Iteration computations: repeat steps 3 and 4 until the MSE is smaller than the convergence threshold or the number of training epochs meets the setting maximum value. In the paper, the convergence threshold is set to 0.001.

2. Quickprop Algorithm

In the back-propagation network, its nonlinear transfer function is differentiable, and the steepest descent method is used to update the node weights. However, back-propagation learning is too slow and it scales up poorly as tasks become larger and more complex. Selection of the back-propagation learning parameters is something of a black art, and small differences in these parameters can lead to large differences in learning times (Fahlman and Lebiere, 1988). In order to let the error quickly converge to a local minimum, there are two commonly used methods. One is to dynamically adjust the learning rate, and the other is the second-order gradient method (Newton's method). Quickprop is a BPN using the latter. In training the networks, Quickprop uses the second-order gradient to calculate the adjustment of the weights (Fahlman and Lebiere, 1988). The adjustment of weights is as follows.

$$\Delta w_{ji}(n) = \frac{S(n)}{S(n-1)-S(n)} \Delta w_{ji}(n-1) \tag{10}$$

where

$$S(n) = \frac{\partial E}{\partial w_{ji}(n)};\ S(n-1) = \frac{\partial E}{\partial w_{ji}(n-1)} \tag{11}$$

The coefficient in the Eq. (10) is based on two assumptions: (1) the error function is a parabolic approximation and it opens upwards; (2) the weight can be adjusted independently, i.e. the weights will not affect each other in the learning process of the gradient value of the error function. The update rules of weights are:

(1) If $S(n)S(n-1) < 0$, Quickprop obtains the minimum of the function, which means that the search direction obtained from Eq. (10) is right.
(2) Since $S(n)S(n-1) > 0$, this means that the nth search direction is the same as the previous one. If $S(n)S(n-1) > S(n)^2$, then the weight should be updated in the same search direction. Therefore, Eq. (10) should be used or reinforced by:

$$\Delta w_{ji}(n) = \Delta w_{ji}(n) - \mu S(n) \tag{12}$$

where the parameter μ, which is called the maximum growth factor, in this paper, is set to 1.75 (Fahlman and Lebiere, 1988).

(3) If $S(n)S(n-1) > 0$ and $S(n)S(n-1) < S(n)^2$, then Eq. (10) does not preserve the sign of the search direction and it should adopt the steepest descent method, i.e.

$$\Delta w_{ji}(n) = -\mu S(n) \tag{13}$$

IV. ACTIVE VIBRATION ISOLATION SYSTEMS

In this paper, the active vibration isolation system shown in Fig. 4 is a two-mass dynamic system in which the main structure is placed on the isolator. The main structure, which undergoes the exciting load, $f_0(t)$, consists of the mass, m_1, stiffness, k_1 and damping coefficient, c_1. The isolator has mass, m_2, stiffness, k_2 and damping coefficient, c_2. An auxiliary force, known as the control force, $f_a(t)$, is imposed on the vibration isolator.

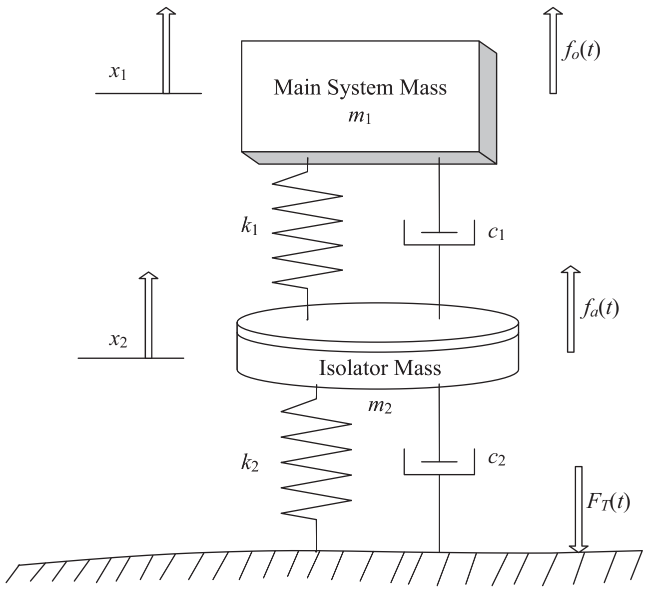


Fig. 4. Active vibration isolation system.

Assuming that the active vibration isolation system is a linear time-invariant dynamic system, its motion equation is as follows.

$$\begin{cases} m_2\ddot{x}_2 + (c_2+c_1)\dot{x}_2 + (k_2+k_1)x_2 - c_1\dot{x}_1 - k_1x_1 = f_a \\ m_1\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 - c_1\dot{x}_2 - k_1x_2 = f_0 \end{cases} \tag{14}$$

where x_i, $\dot{x}_i$, and $\ddot{x}_i$, are the displacement, velocity, and acceleration of the main structure, $i = 1$, and the isolator, $i = 2$.

As the exciting load and the control force impose on the two-mass dynamic system, the system produces a force, which is the transmissibility force, F_T, imposed on the ground, composed of the spring force, F_{spring_I} and the damping force, $F_{damping_I}$ on the isolator:

$$F_T = F_{spring_I} + F_{damping_I} \tag{15}$$

In this paper, the control performance of the active vibration isolation system is defined as the transmissibility force, $F_T(t)$. For the active vibration isolation system, we examine the control performance of (1) the fuzzy controller in the first phase, and (2) the neural network fuzzy controller in the second phase. The observations of the controlled system that are displacement, x_2 and velocity, $\dot{x}_2$ of the isolator are input to the fuzzy logic rules base for analysis of control force. The fuzzy logic control model for the active vibration isolation system is shown in Fig. 5. In the flowchart, the BPN algorithm is used to construct the rule surface of fuzzy logic controller shown in Fig. 6.

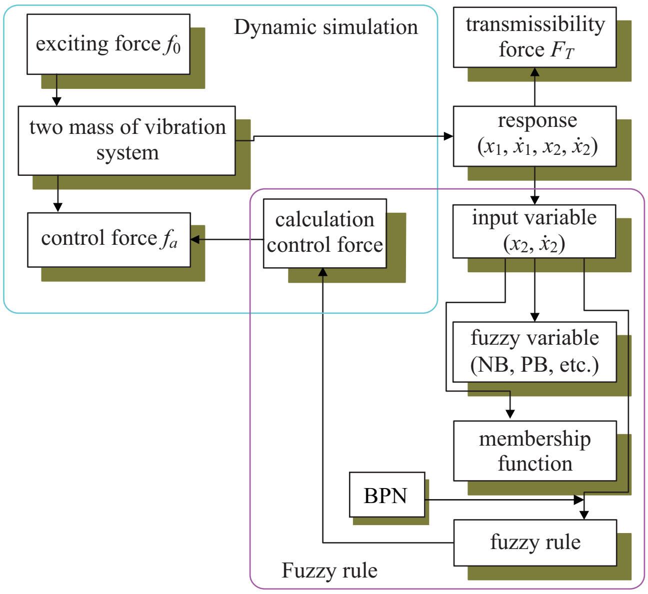


Fig. 5. The schematic diagram of the fuzzy logic control system.

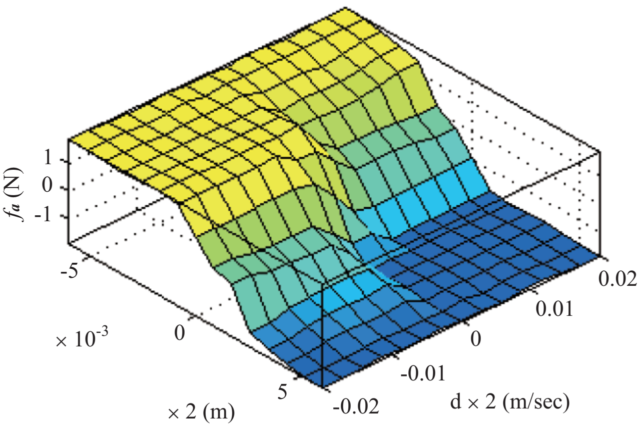


Fig. 6. Rule surface of the fuzzy logic controller.

V. SIMULATIONS OF ACTIVE VIBRATION ISOLATION CONTROL

Simulation of the active vibration isolation control for the AVIS of the two-mass dynamic system is conducted to assess the control performance of the approach presented in the paper. The dynamic parameters of the numerical example of AVIS are set as follows:

Main structure: mass m_1 = 15 kg; stiffness k_1 = 60 N/m; damping c_1 = 3 N·s/m.

Isolator: m_2 = 3 kg; stiffness k_2 = 60 N/m; damping c_2 = 3 N·s/m.

The exciting loads applied to the main structure have three load types, namely, shock, periodic and stochastic, as follow:

(1) Shock load shown in Fig. 7(a): Exciting force is 10 N, in the downward direction, in effect for 0.01 *s*.

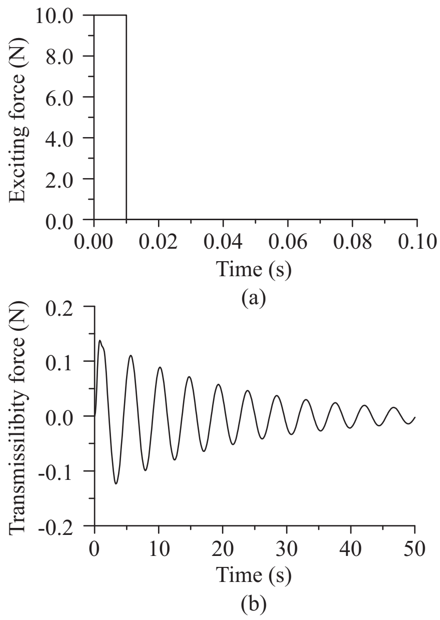


Fig. 7. The time histories of (a) shock exciting load and (b) transmissibility force in the vibration isolation system (without control force).

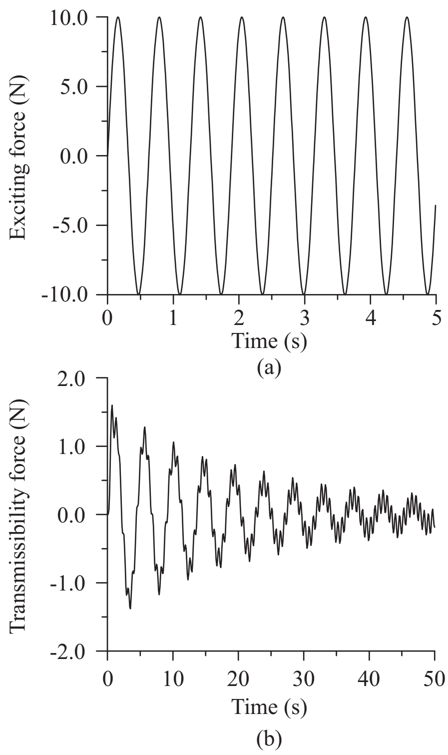


Fig. 8. The time histories of (a) periodic exciting load and (b) transmissibility force in the vibration isolation system (without control force).

(2) Periodic load shown in Fig. 8(a): Amplitude of the exciting force is f_0 = 10 N, frequency ω = 10 rad/s and sinusoidal waveform is represented by $f_0(t) = f_0 \sin(10t)$ N.

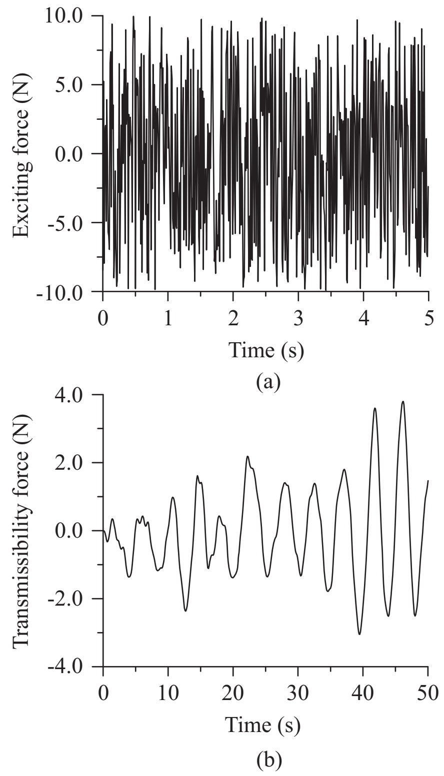


Fig. 9. The time histories of (a) stochastic exciting load and (b) transmissibility force in the vibration isolation system (without control force).

(3) Stochastic load shown in Fig. 9(a): The range of the exciting force is $-10\ N \le f_0(t) \le 10\ N$.

When shock, periodic and stochastic loads are applied to the main structure as an exciting load, the time histories of the transmissibility force of the dynamic vibration system are as shown in Figs. 7(b)-9(b), respectively.

Figs. 7(b)-9(b) show that the transmissibility force response of the dynamic vibration system has large values, not close to zero. Of the three exciting cases, the largest value of amplitude the transmissibility force $F_T(t)$ achieves is 4 N at 48 second for the stochastic exciting case; other values are 1.6 N at 1 second or so for the periodic case and 0.14 N at 1 second or so for the shock case. In the paper, the proposed two-phase design approach aims to designate a vibration isolation controller to achieve a better performance for the transmissibility force response. The numerical simulations for control performances of a vibration isolation controller designated in the two-phase procedure are described as follows:

1. The Fuzzy Logic Controller

1) Design of Fuzzy Logic Controller

Design of a fuzzy logic controller has the following steps:

(a) The choice of input and the input variables:

Table 1. Fuzzy linguistic variables and their values.

Linguistic variables	Meaning	Membership function and ranges
PB	positive big	[0.284,0.833,1,1]
PM	positive medium	[0.042,0.284,0.833]
PS	positive small	[0,0.042,0.284]
ZR	Zero	[-0.042,0,0.042]
NS	negative small	[-0.284,-0.042,0]
NM	negative medium	[-0.833,-0.284,-0.042]
NB	negative big	[-1,-1,-0.833,-0.284]

Table 2. Fuzzy control rules base.

control Force f_a(N)		displacement of isolator x_2 (m)						
		PB	**PM**	**PS**	ZR	**NS**	**NM**	**NB**
speed of isolator $\dot{x}_2$ (m/sec)	PB	NB	NB	NB	NM	NS	ZR	PB
	PM	NB	NB	NB	NM	NS	PS	PB
	PS	NB	NB	NM	NS	ZR	PM	PB
	ZR	NB	NB	NS	ZR	PS	PB	PB
	NS	NB	NM	ZR	PS	PM	PB	PB
	NM	NB	NS	PS	PM	PB	PB	PB
	NB	NB	ZR	PS	PM	PB	PB	PB

To solve this problem, a fuzzy logic control method is designed which uses the isolator displacement x_2 and velocity $\dot{x}_2$ as multi-input variables while the control force fa is the single output. In the paper, the values of isolator's displacement, x_2 and velocity, $\dot{x}_2$ are set to $|x_2| \le 0.006\ m$ and $|\dot{x}_2| \le 0.002$ m/s, respectively. The range of output control force is set to $-2.5\ \text{N} \le fa(t) \le 2.5$ N. Sampling time is set to 0.01 s.

(b) Segmentation of the domain and choice of the membership function:

Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZR), Positive Small (PS), Positive Medium (PM), and Positive Big (PB) are used to represent the fuzzy value of the controller input and output as linguistic variables. In the paper, their values are between -1 and 1. The membership function of the linguistic variables adopts the triangular function, except in the case of NB and PB, which use the trapezoid ones; and their corresponding values are listed in Table 1.

(c) Fuzzy control rule base:

The rules of fuzzy control and the method of consequence are regular. The rules of fuzzy control are listed in Table 2. In the fuzzy logic control system, the consequence methods used are as such: the method is either minimum or maximum, the implication is minimum, aggregation maximum, and the defuzzification is centroid. The rule surface of the controller is illustrated in Fig. 6.

2) Evaluation of the Control Performance of FLC:

Three exciting loads, such as shock, periodic and stochastic

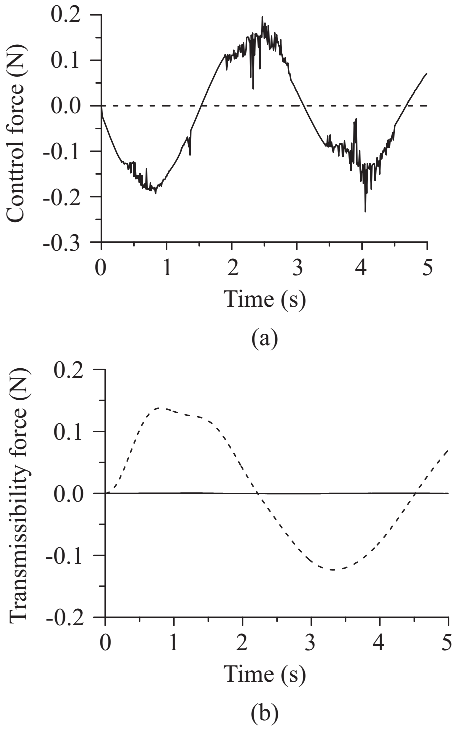


Fig. 10. The time histories of (a) control force of FLC and (b) transmissibility force in the AVIS under shock exciting load without control force (dashed line) and with control force (solid line).

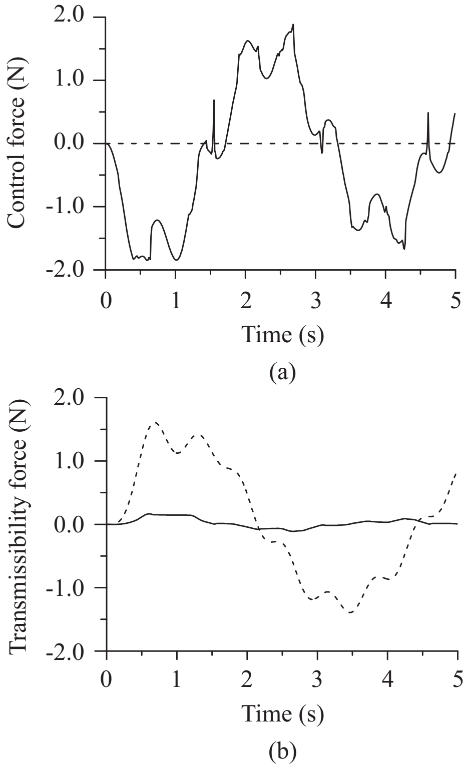


Fig. 11. The time histories of (a) control force of FLC and (b) transmissibility force in the AVIS under periodic exciting load without control force (dashed line) and with control force (solid line).

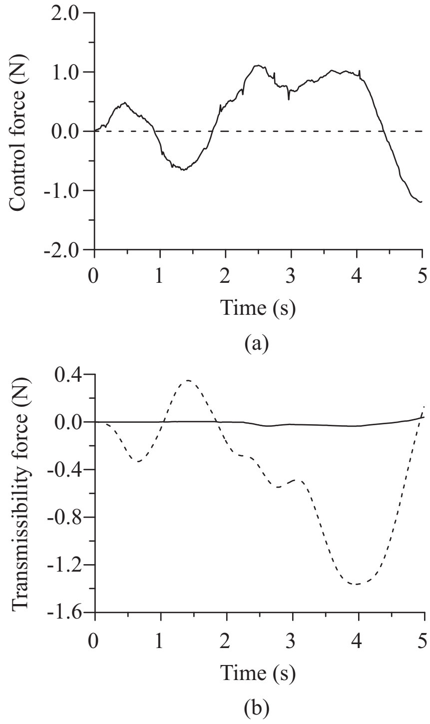


Fig. 12. The time histories of (a) control force of FLC and (b) transmissibility force in the AVIS under random exciting load without control force (dashed line) and with control force (solid line).

loads, and control force inferred from FLC, the AVIS are carried out, respectively. The time histories of the control force and transmissibility force of the AVIS are presented in Figs. 10-12, both with and without control force. Figs. 10(b)-12(b) show that the FLC designed in the paper achieves good control performance when the main structure is under either of the three kinds of exciting load. The three maximum transmissibility forces are 0.0005 N at 1.5 second or so for the shock exciting case, 0.15 N at 0.6 second or so for the periodic case, and 0.04 N at 5 second or so for the stochastic case. The control effects of FLC for three cases are close to 100% for the shock exciting case, 91.0% for the periodic case, and 99% for the stochastic case, respectively. From Figs. 10(a)-12(a), the maximum amplitudes of control force are far lower than 10 N, i.e. the amplitude of the exciting force, ad also lower than the maximum given value of $f_a(t)$, which is determined in designing the FLC. The maximum values are 0.3 N at 4 second or so for the shock exciting case, 1.8 N at 2.7 second or so for the periodic case, and 1.1 N at 1.5 second or so for the stochastic case, respectively.

From Figs. 10(a)-12(a), however, the histories of control force inferred from FLC have several large jumps during the controlling process and these could weaken its practical application even for a mechanical actuator. Fig. 6 shows that the control rule surface too has a large change around the isolator displacement, x_2, and velocity, $\dot{x}_2$, near the ZR. These jumps are due to empirical results of the tests and experimental experiences of the designer. Hence, it is expensive and difficult

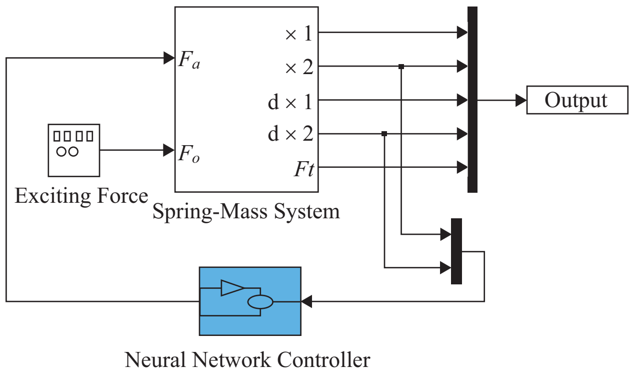


Fig. 13. Neural network fuzzy controller.

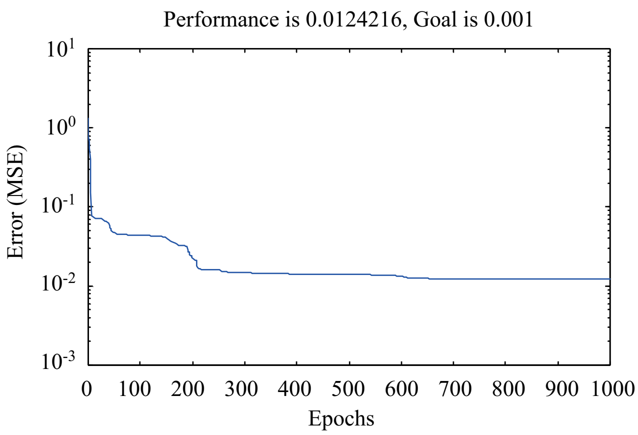


Fig. 14. Convergence of Quickprop learning error.

to improve the rule surface by modifying the membership functions and rules base of FLC.

In view of this, based on this FLC, the neural networks approach adopted in the second phase of the proposed approach will improve the weakness and achieve robust, controllable, and low-cost solutions.

2. The Neural Network Fuzzy Controller

In this phase, we use the Quick-prop algorithm to train the back-propagation neural network model of the fuzzy logic controller (also termed the neural network fuzzy controller (NNFC)). The trained model is applied to the active vibration isolation system as shown in Fig. 13. The parameter settings for training the neural network model are as follows: the number of node in the input layer is 2, the number of nodes in the hidden layer is 6, the learning rate η is 0.05, the active function adopts Tagent-sigmoid, the convergence criteria (denoted by ε) is 0.001, the number of training samples is 500, and the maximum number of training epochs is 1000. The convergence diagram of the mean square error (MSE) is presented in Fig. 14, and it finally converges to the value 0.0124 as the number of training epochs meets the maximum setting value, 1000. The rule surface of the neural network fuzzy

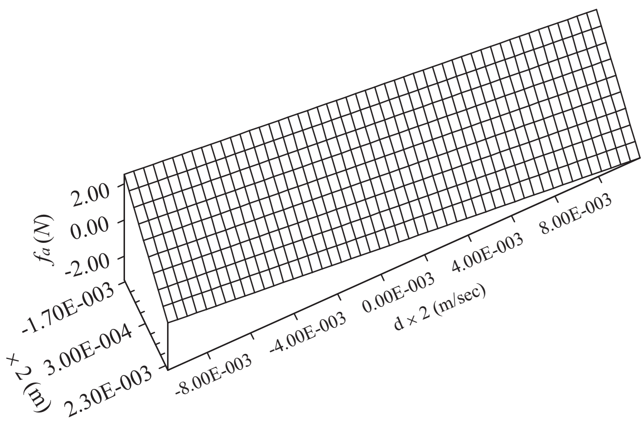


Fig. 15. The control surface of neural network controller.

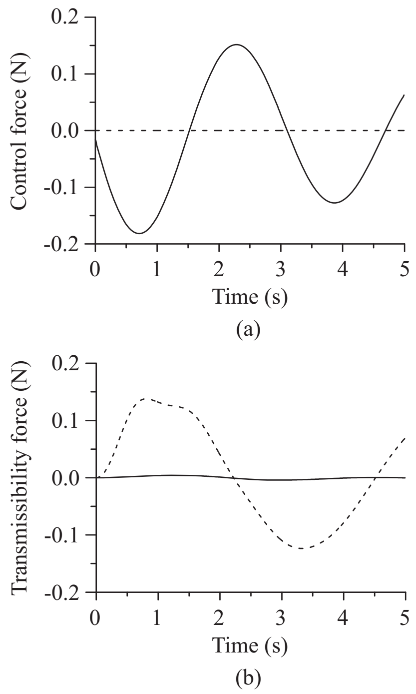
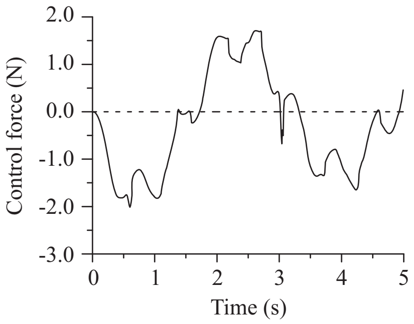


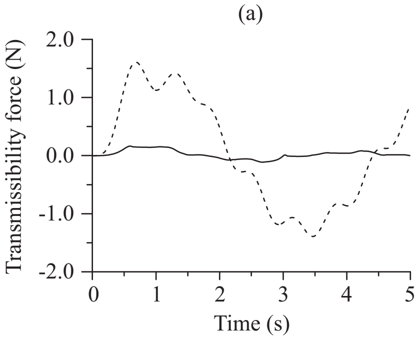
Fig. 16. The time histories of (a) control force of neural network fuzzy controller and (b) transmissibility force in the AVIS under shock exciting load without control force (dashed line) and with control force (solid line).

controller is shown in Fig. 15. The rule surface at the isolator displacement, x_2, velocity, $\dot{x}_2$ and in the vicinity of ZR has no jumps and is quite smooth.

With the three exciting loads applied on the main structure and the control force obtained from the neural network fuzzy controller, the numerical simulations have been conducted. The history diagrams of the control force and the transmissibility force are presented in Figs. 16-18. Figs. 16(a)-18(a) reveal that the control force during controlling process has no large jumps. From Figs. 16(a)-18(a), the maximum control forces are the same as those of FLC in the previous phase, except for the value 0.15 N at 0.8 second or so for the shock

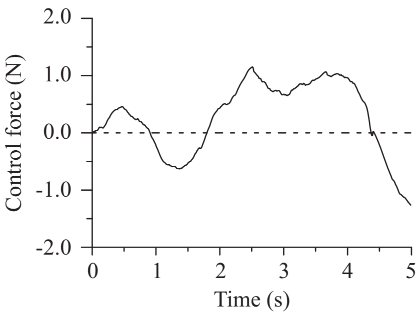


(a)

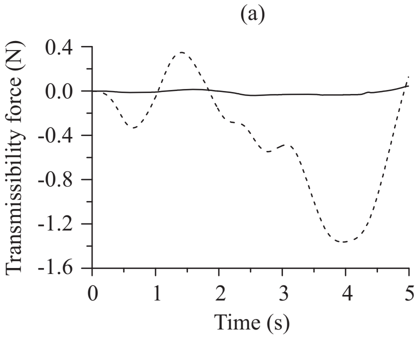


(b)

Fig. 17. The time histories of (a) control force of the neural network fuzzy controller and (b) transmissibility force in the AVIS under periodic exciting load without control force (dashed line) and with control force (solid line).



(a)



(b)

Fig. 18. The time histories of (a) control force of the neural network fuzzy controller and (b) transmissibility force in the AVIS under stochastic exciting load without control force (dashed line) and with control force (solid line).

exciting case. Figs. 16(b)-18(b) indicate their control performances are also good and correspond to that of the fuzzy logic controller developed in the previous phase. From Figs. 16(b)-18(b), the maximum transmissibility forces are also the same as those of the FLC, except for the value of 0.005 N at 1.5 second or so that is greater than that of the FLC. The results also indicate that the neural network approach adopted in the second phase of the proposed approach completely eliminates the problem of the large jumps of the FLC during the controlling.

With the above simulations of ACVIS in the two phases, the results demonstrate that the two-phase design approach proposed in the paper provides robust, controllable, practicable, and low-cost solutions for FLC design.

VI. CONCLUSION

In this paper, we propose a two-phase design approach, which integrates the soft computing technologies of fuzzy logic control theory and the neural networks, to design a fuzzy logic controller for active vibration isolation systems that achieves a very good control performance and is practical.

In fuzzy logic control theory, the control rules are based on expert knowledge and experiences using linguistic variables with specified membership functions to describe the characteristics of the controlled system. To construct the rule bases, the greater the understanding of the characteristics of the controlled system, the better will be the performance the fuzzy logic controller.

Control performance of the fuzzy logic controller developed in the first phase of the proposed approach is very good in simulations of an active vibration isolation system with the two-mass dynamic system under three kinds of external exciting loads, but it has a weak point in that the control force imposed on the isolator often has large jumps during controlling.

The proposed approach, in the second phase, aims to improve this drawback of the FLC, by using the Quick-prop algorithm with Newton's method to train a neural network model of FLC, termed a neural network fuzzy controller. Simulation results show that the control performance is of the same good quality as the FLC designed in the first phase, the control surface of the FLC is improved, and the jumps have been eliminated. It is clearly demonstrated in this paper that neural networks trained by Quick-prop algorithm with Newton's method can achieve a robust and fault-tolerant solution to the FLC of AVIS.

REFERENCES

Bai, M. R. and W. Liu (2002). Control design of active vibration isolation using, μ-synthesis. Journal of Sound and Vibration 257, 157-175.

Cheong, F. and R. Lai (2007). Designing a hierarchical fuzzy logic controller using the differential evolution approach. Applied Soft Computing 7(2), 481-491.

Cheung, Y. L., W. O. Wong and L. Cheng (2015). A subsystem approach for analysis of dynamic vibration absorbers suppressing broadband vibration. Journal of Sound and Vibration 342, 75-89.

Eker, I. and Y. Torun (2006). Fuzzy logic control to be conventional method. Energy Conversion and Management 47(4), 377-394.

Engels, W. P., O. N. Baumann and S. J. Elliott (2006). Centralized and decentralized control of structural vibration and sound radiation. Journal of Acoustical Society of America 119, 1487-1495.

Fahlman, S. E., M. Lebiere (1988). An Empirical Study of Learning Speed in Back-Propagation Networks, Tech Report CMU-CS-88-162, Carnegie Mellon University.

Hebb, D. O. (1989). The Organization of Behavior: A Neuro-psychological Theory, Wiley, New York.

Hornik, K., M. Stinchcombe and H. White (1990). Universal approximation of an unknown mapping and its derivatives using multiplayer feed forward networks. Neural Networks 3, 551-560.

Kim, H.-S. and P. N. Roschke (2006). Design of fuzzy logic controller for smart base isolation system using genetic algorithm. Engineering Structures 28(1), 84-96.

Lin, T.-K., L.-Y. Lu and H. Chang (2015). Fuzzy logic control of a stiffness-adaptable seismic isolation system. Structural Control and Health Monitoring 22(1), 177-195.

Mamdani, E. H. (1976). Application of fuzzy algorithms for control simple dynamic plants. Proc. IEEE 121(12), 1585-1588.

Narendra, K. S. and J. Balakrishnan (1997). Adaptive control using multiple models. IEEE Transaction Automatic Control 42(2), 171-187.

Nastac, S. (2008). About vibration isolation performances and fuzzy logic techniques. SISOM 2008 and Session of the Commission of Acoustics, Bucharest, 100-104.

Ricardo, M.-S., O. Castillo and J. R. Castro (2014). Genetic algorithm optimization for type-2 non-singleton fuzzy logic controllers. In: Recent Advances on Hybrid Approaches for Designing Intelligent Systems Studies in Computational Intelligence 547, 3-18.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986). Learning internal representations by error propagation. In: Parallel Distributed Processing 1, Cambridge, MA: MIT Press, 318-362.

Song, R. and L. Chen (2010). A Study about Fuzzy Control Algorithm Base on Neural Networks Forecast Model in Active Vibration Isolation Control System. 2010 International Conference on Computational Aspects of Social Networks.

Zadeh, L. A. (1965). Fuzzy Sets. In: Information and Control 8, 338-353.

Zadeh, L. A. (1973). Outline of a New Approach to The Analysis of Complex Systems and Decision Processes. IEEE Trans. on Systems, Man and Cybernetics, SMC-3(1), 28-44.

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. Communications of the ACM 37(3), 77-84.