ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Sound and Vibration 266 (2003) 1009–1023

JOURNAL OF
SOUND AND
VIBRATION

www.elsevier.com/locate/jsvi

An inverse model of MR damper using optimal neural network and system identification

Pin-Qi Xia*

College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China

Received 5 February 2002; accepted 4 October 2002

Abstract

Magnetorheological (MR) damper is one of the more promising new devices for vibration control of structures. External energy required by the adjustable fluid damper is minuscule while speed of its response is in the order of milliseconds. The MR damper is a semi-active control device and has been characterized by a set of non-linear differential equations which represent a forward model of the MR damper, i.e., the model can generate a force to a given displacement and applied voltage.

This paper presents an inverse model of the MR damper, i.e., the model can predict the required voltage so that the MR damper can produce the desired force for the requirement of vibration control of structures. The inverse model has been constructed by using a multi-layer perceptron optimal neural network and system identification, which are Gauss–Newton-based Levenberg–Marquardt training algorithm, optimal brain surgeon strategy and autoregressive with exogenous variables (ARX) model. Based on the data from numerical simulation of the MR damper, the trained optimal neural networks can accurately predict voltage. If the inverse model is used in a control system, the semi-active vibration control can be implemented easily by using the semi-active MR damper.

1. Introduction

The non-linear phenomenological model of magnetorheological (MR) damper has been developed by Spencer et al. [1] The model is based on a Bouc–Wen hysteresis model and is accurately and numerically tractable for the characteristic of the MR damper. By using this mathematical model, the force of the MR damper is directly solved to a given displacement and applied voltage. The model is called forward model. In an active control system, the control force

*Corresponding author. Fax: +86-25-489-1422.
E-mail address: xiapq@nuaa.edu.cn (P.-Q. Xia).

0022-460X/03/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0022-460X(02)01408-6

needs to be known to meet a required vibration level. If the MR damper is used for semi-active vibration control, it may be desirable to have an inverse model to predict preliminarily the applied voltage in order to generate the required control force. However, solving the non-linear equations describing the performance of the MR damper may be difficult or time consuming to predict a required voltage, which will lead to complex and difficult controller design. Therefore, it is valuable to establish an inverse model, instead of the complex mathematical model, of the MR damper which can directly predict voltage.

Generally, it is difficult to develop a mathematical model for a complex non-linear dynamic system such as the MR damper. Recently, the artificial neural network [2] has been effectively applied to model complex systems because of its great learning (training process) and creation (emulator) functions. It is possible to model the MR damper and identify the performance of the complex dynamic system by using neural network. The technology of neural network includes constructing network architectures and training them. Before construction of the network architecture, two important problems must be considered: how many hidden layers should be selected and how many neurons should be used in each layer? The questions are one of the most difficult aspects of developing neural network models. Currently, to make a priori determination of the appropriate size of the network is still difficult. Increasing the number of hidden layers will make the neural network possess a better capability in learning more non-linear mappings. However, too many hidden layers will deteriorate the performance of neural networks. Studies [2] have shown that one or two hidden layers are sufficient for most problems. The number of neurons in the hidden layers determine the capacity of neural networks and the complexity of the underlying knowledge base in the data used for training neural networks. The network architecture is not unique. The size of the neural network cannot be well determined for a good performance. In order to construct an ideal network, this paper presents a practical strategy: (1) using appropriately more neurons in the hidden layers so that, firstly, the neural network can acquire adequate knowledge from the training data; (2) using a so-called optimal brain surgeon (OBS) strategy [3] to remove the superfluous weights from the network so that an optimal network architecture can be determined by pruning the network.

Network learning or training is the process of weighting adjustment based on the input/output data. Training algorithms are generally based on the minimization of an energy or error function that is the function of the vector of weights to be adjusted. Generally, these algorithms include the gradient descent algorithm [4], the back propagation algorithm [5], the least-squares algorithm [6], the recursive estimation method [7] and the Gauss–Newton-based Levenberg–Marquardt method [8].

It is difficult to determine the input and output signals when using neural network to model a non-linear dynamic system. This procedure is crucial for a successful identification of the complex system. System identification theory [7] provides a mathematical framework for analysis and design of dynamic systems of various types, regardless of their special physical nature and functions. System identification is the procedure to develop a dynamic model of a structural system to replicate the behavior of the system by measuring input/output relationships of the system. Hence, using system identification may be useful for determination of input/output signals for the neural network.

In this paper, the methodologies for construction of an inverse model of the MR damper are presented using optimal neural network and system identification. The data for training and validating neural network were produced from the simulation of the mathematical model for the MR damper proposed by Spencer et al. [1]. The system identification based ARX model and a multi-layer perception network were selected to determine the model structure. To construct an inverse model of the MR damper, the displacement, voltage and force were selected as the input variables, while the voltage was selected as the output variable. The fully connected networks with one hidden layer and 12 neurons in the hidden layer were selected to train by using the Gauss–Newton-based Levenberg–Marquardt method. The OBS strategy was used to retrain the network and prune the superfluous weights from the network to determine of an optimal network. The optimal network was verified by using two different sets of data.

2. Simulated MR damper performance

Spencer et al. [1] proposed a new phenomenological model based on a Bouce–Wen hysteresis model for a prototype MR damper developed by the Lord Corporation. The simple mechanical model of the MR damper is shown in Fig. 1 and it is governed by the following seven equations:

$$f = c_1\dot{y} + k_1(x - x_0), \tag{1}$$

$$\dot{y} = \frac{1}{c_0 + c_1}[\alpha z + c_0\dot{x} + k_0(x - y)], \tag{2}$$

$$\dot{z} = -\gamma|\dot{x} - \dot{y}|z|z|^{n-1} - \beta(\dot{x} - \dot{y})|z|^n + A(\dot{x} - \dot{y}), \tag{3}$$

$$\alpha = \alpha_a + \alpha_b u, \tag{4}$$

$$c_1 = c_{1a} + c_{1b}u, \tag{5}$$

$$c_0 = c_{0a} + c_{0b}u, \tag{6}$$

$$\dot{u} = -\eta(u - v). \tag{7}$$

A total of 14 model parameters were obtained using a constrained non-linear optimization algorithm to characterize the damper. Optimized parameters were determined to fit the

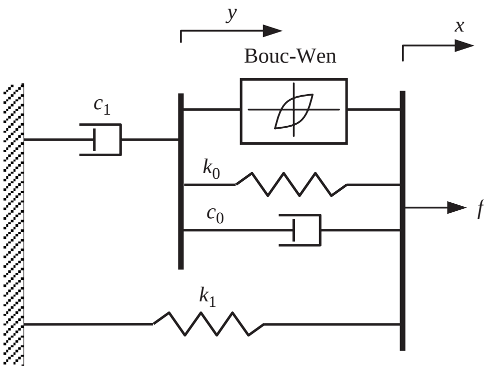

Fig. 1. Model of MR damper.

generalized model to the experimental data in a variety of tests. The resulting parameters are given in Table 1. According to the model shown in Fig. 1, the force f of the prototype damper is obtained if the patterns of displacement x and voltage v are prescribed. The model portrayed effectively the behavior of the prototype MR damper and was numerically tractable. From the model given in Eqs. (1)–(7), a set of data including given displacement x and voltage v, and force f generated by the MR damper can be obtained. Fig. 2(a) shows the displacement, which is a 2.5 Hz sinusoid with amplitude of 1.5 cm. The applied voltage shown in Fig. 2(b) has the straight-line type with expression of $v = 1.25 + t$ (Volt). Fig. 2(c) shows the force generated by the MR damper. The force–displacement loops are shown in Fig. 3 and progress along a clockwise path with increasing time, whereas the force–velocity loops are shown in Fig. 4 and progress along a counterclockwise path with increasing time. These displacement, voltage and force representing the performances of the MR damper were used to train the neural network.

3. Modelling techniques for MR damper performance

To construct the neural network for representing the performance of the MR damper, the modelling techniques including a multi-layer perceptron (MLP) neural network, a Gauss–Newton-based Levenberg–Marquardt training algorithm and an ARX model are presented as follows.

3.1. MLP network

The MLP network is the most often used member in the neural network family due to its ability to model simple as well as very complex functional relationships. A fully connected two layer feedforward MLP-network with p inputs, m outputs, q hidden neurons and one bias in the input and output layers, respectively, is shown in Fig. 5. For those MLP-networks having only one hidden layer, only sigmodial activation functions f in the hidden layer and one linear activation function F, the output $\hat{y}_i$ can be expressed as the function of weights and inputs:

$$\hat{y}_i = F_i\left[\sum_{i=1}^{q} W_{ij} f_i\left(\sum_{k=1}^{p} w_{jk} u_k + w_{j0}\right) + W_{i0}\right]. \tag{8}$$

Table 1
Parameter values of MR damper model

Parameter	Value	Parameter	Value
c_{0a}	21.0 N s/cm	α_a	140.0 N/cm
c_{0b}	3.50 N s/cm V	α_b	695.0 N s/cm V
k_0	46.9 N/cm	γ	363.0 cm^{-2}
c_{1a}	283.0 N s/cm	β	363.0 cm^{-2}
c_{1b}	2.95 N s/cm V	A	301.0
k_1	5.0 N/cm	n	2
x_0	14.3 cm	η	190 s^{-1}

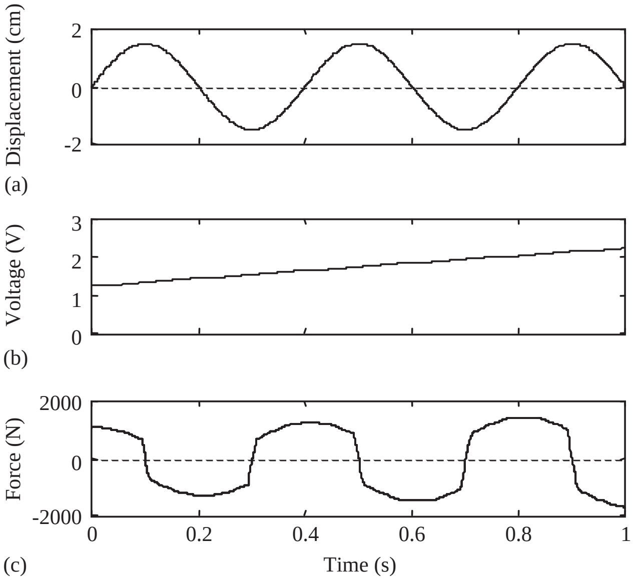

Fig. 2. Signals for training neural network.

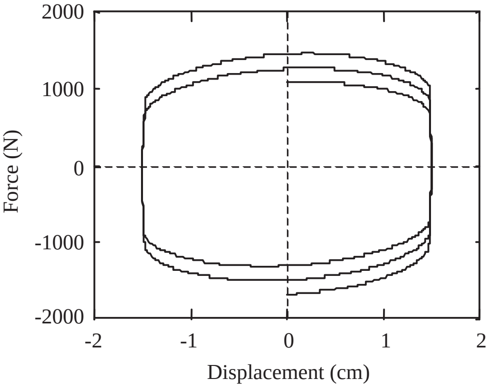

Fig. 3. Force–displacement loops of MR damper.

The weights W_{ij} and $w_{jk}(i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, q)$ specified by vector θ are the adjustable parameters of the network and they are determined from a set of data through the process called training. The training data are a set of inputs $\{u(s)\}$ and the corresponding desired outputs $\{y(s)\}$. The hyperbolic tangent function is selected as the sigmodial action function f in the hidden layer. Its curve is shown in Fig. 6 and it is expressed by

$$f(x) = 1 - \frac{2}{1 + e^{2x}}. \tag{9}$$

The training data set is specified with N points as

$$Z^N = \{[u(s), y(s)]\big|_{s=1,\ldots,N}\}. \tag{10}$$

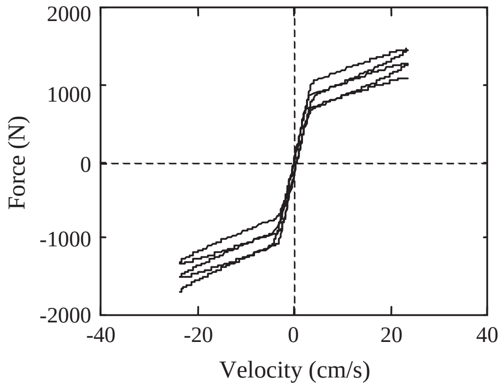

Fig. 4. Force–velocity loops of MR damper.

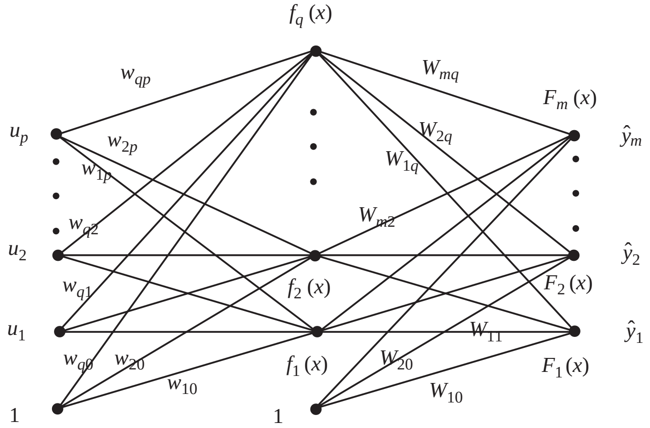

Fig. 5. A fully connected two-layer feedforward network.

Then the training objective is to determine a mapping from a set of training data Z^N to a set of possible weights $\hat{\theta}$:

$$Z^N \to \hat{\theta}, \tag{11}$$

so that the network can produce prediction $\hat{y}(s)$, close to the true outputs $y(s)$.

The prediction error approach is based on a measure of closeness of a mean square error criterion augmented with a regularization term

$$V_N(\theta, Z^N) = \frac{1}{2N}\sum_{s=1}^{N}[y(s) - \hat{y}(s|\theta)]^{\mathrm{T}}[y(s) - \hat{y}(s|\theta)] + \frac{1}{2N}\theta^{\mathrm{T}}D\theta, \tag{12}$$

where D is the regularization matrix. For a simple weight decay, D is a diagonal matrix which is selected to be $D = \varepsilon I$, in which I is a unit matrix, and ε is a small number which represents a

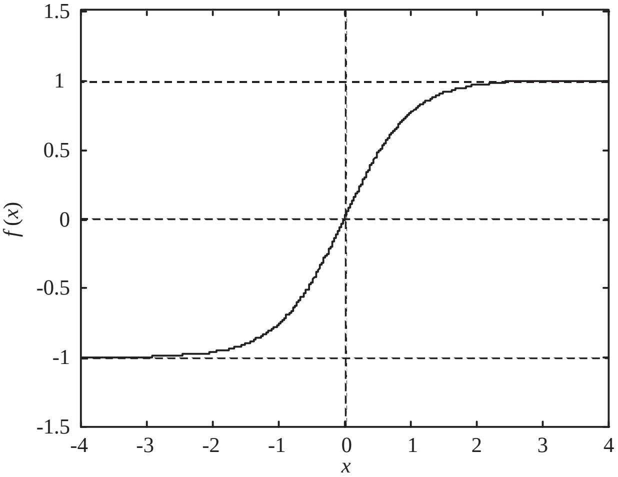

Fig. 6. Curve of hyperbolic tangent function.

weight decay. Criterion (12) is also called a regularized criterion. If $D = 0$, then criterion (12) is called an unregularized criterion. The weights vector is then found to be

$$\hat{\theta} = \arg_{\theta} \min V_N(\theta, Z^N) \tag{13}$$

by an iterative minimization scheme

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} + \mu^{(i)} f^{(i)}, \tag{14}$$

where $\hat{\theta}^{(i)}$ denotes the ith iteration, $f^{(i)}$ is a search direction based on information about $V_N(\theta, Z^N)$ acquired at previous iterations, and $\mu^{(i)}$ is the step size determined so that an appropriate decrease in the value of $V_N(\theta, Z^N)$ is obtained.

3.2. Levenberg–Marquardt training algorithm

The Gauss–Newton-based Levenberg–Marquardt method was used to minimize the mean-square error, due to its rapid convergence properties and robustness. In terms of the method, the following equation is used to determine the search direction $f^{(i)}$:

$$f^{(i)} = -[V''_N(\hat{\theta}^{(i)}, Z^N) + \lambda^{(i)} I]^{-1} V'_N(\hat{\theta}^{(i)}, Z^N), \tag{15}$$

where λ is a small positive scalar. $V'_N(\hat{\theta}^{(i)}, Z^N)$ and $V''_N(\hat{\theta}^{(i)}, Z^N)$ are the matrices of the first and second partial derivatives of $V_N(\hat{\theta}^{(i)}, Z^N)$ with respect to $\hat{\theta}^{(i)}$. If $\lambda = 0$, then Eq. (15) will degenerate to the Gauss–Newton method.

By adding a scalar λ to the diagonal elements in the matrix of second partial derivatives or the so-called Hessian matrix, the Gauss–Newton-based Levenberg–Marquardt method overcomes the non-positive definite problems arising in the Hessian matrix. The scalar λ is adjusted according to the size of ratio $r^{(i)}$ between actual reduction in $V_N(\theta, Z^N)$ on the ith step and the corresponding predicted reduction as:

$$r^{(i)} = \frac{V_N(\theta^{(i)}, Z^N) - V_N(\theta^{(i)} + \mu^{(i)} f^{(i)}, Z^N)}{V_N(\theta^{(i)}, Z^N) - L_N(\theta^{(i)} + \mu^{(i)} f^{(i)})}, \tag{16}$$

where $L_N(\theta^{(i)} + \mu^{(i)} f^{(i)})$ is the resulting quadratic approximation for ith iteration which can be obtained from a truncated Taylor series expansion of $V_N(\theta^{(i)}, Z^N)$ about $\theta^{(i)}$ based on the Newton's method, and can be written:

$$L_N(\theta^{(i)} + \mu^{(i)} f^{(i)}) = V_N(\theta^{(i)}, Z^N) + f^{(i)\mathrm{T}} V'_N(\theta^{(i)}, Z^N) + \tfrac{1}{2} f^{(i)\mathrm{T}} V''_N(\theta^{(i)}, Z^N) f^{(i)}. \tag{17}$$

The ratio $r^{(i)}$ measures the accuracy indicating how $L_N(\theta^{(i)} + \mu^{(i)} f^{(i)})$ approximates $V_N(\theta^{(i)} + \mu^{(i)} f^{(i)}, Z^N)$. The closer $r^{(i)}$ is to unity, the better is the agreement. The ith iteration takes the following steps:

(1) taking initial values of weight vector $\theta^{(0)}$ and an initial value $\lambda^{(0)}$ ($\theta^{(0)}$may be a set of small (<1) non-zero random numbers and $\lambda^{(0)}$ may be a small (e.g. 1) positive number);
(2) determining the search direction $f^{(i)}$ from Eq. (15);
(3) calculating $r^{(i)}$ by Eq. (16), if $r^{(i)} > 0.75$, then setting $\lambda^{(i)} = \frac{1}{2}\lambda^{(i)}$ (if predicted reduction is close to actual reduction ($r^{(i)} \to 1$), then let the search direction approach the Gauss–Newton search direction and increase the step size $\mu^{(i)}$);
(4) if $r^{(i)} < 0.25$, then setting $\lambda^{(i)} = 2\lambda^{(i)}$ (if predicted reduction is far from the actual reduction ($r^{(i)} \to 0$), then let the search direction approach the gradient direction and decrease the step size $\mu^{(i)}$);
(5) if $0.25 \leqslant r^{(i)} \leqslant 0.75$, then setting $\lambda^{(i)} = \lambda^{(i)}$;
(6) if $r^{(i)} \leqslant 0$; then accepting $\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} f^{(i)}$ as a new iteration and letting $\lambda^{(i+1)} = \lambda^{(i)}$ and $i = i + 1$;
(7) if the stop criterion ($i \leqslant$ maximum number of iterations or $V_N(\theta^{(i)}, Z^N) <$ error bound) is not satisfied, then go to step (2).

The constants 0.25, 0.75, etc. are arbitrary because the algorithm is quite insensitive to their change. In practice, a more sophisticated iteration [9] can be used. If $r^{(i)} < 0.25$, then $\lambda^{(i)}$ is chosen in an interval $(0.1 - 0.5)\left|\mu^{(i)} f^{(i)}\right|$ on the basis of a polynomial interpolation. Other possible changes may include a more sophisticated extrapolation strategy.

3.3. ARX model

After determining the global architecture of the neural network and the training algorithm, it is necessary to choose a set of regressors (or an appropriate model structure) so as to determine a set of control signals (or input signals) and output signal for the MR damper identification. ARX model is the most basic model describing the relationship between the input and output with the linear difference equation

$$y(s) = \varphi^{\mathrm{T}}(s)\theta, \tag{18}$$

$$\varphi(s) = [y(s-1) \ldots y(s-n_a) u(s-n_k) \ldots u(s-n_b-n_k+1)]^{\mathrm{T}}, \tag{19}$$

where $\varphi(s)$ is a regression vector containing the regressors; θ is a vector containing the weights as adjustable parameters; $u(s)$ and $y(s)$ are the input and output signals, respectively; n_a and n_b are the model orders which determine the structure of the regression vector; n_a represents the number of past output signals and n_b represents the number of past input signals; n_k is the time delay which is equal to one usually. ARX model means that the next output value of the model structure is determined by the previous observations of output and input. The predictor for Eq. (19) gives

$$\hat{y}(s, \theta) = g(\varphi(s), \theta), \tag{20}$$

where g is the function realized by the neural network. Here, the predictor is a non-linear predictor because it is not a scalar product between a known regression vector and the parameter vector θ.

After selecting an ARX model structure, an important procedure is to select the physical variables as signals and the number of past signals as regressors. For the MR damper described previously, there are three physical variables: displacement, voltage and force. In order to predict a required voltage to a desired force, the applied voltage was selected as the desired output from the network and the force generated by the MR damper was selected as the input signal into the network. Such a model is called an inverse model because the desired network output is the plant control input, and the network inputs are the plant measured outputs suitably delayed. Furthermore, the predicted outputs may be related to the past desired outputs. To obtain one of the best trained networks, the desired output of the MR damper was also used as input signal in this investigation. Hence, the input signals into the network were the displacement x, voltage v and force f, while the predicted output signal was the voltage $\hat{v}$. The neural network adjusts the weights to reduce the error between the predicted output and the desired output.

Determination of the model orders n_a and n_b is very difficult and complicated for a non-linear system. He and Asada [10] presented a method for identifying orders of input–output models of non-linear dynamic systems by calculating the index of the order. The index is interpreted such that an insufficient lag space (n_a and n_b) leads to a large index. Increasing the lag space, the index will decrease until a sufficiently large lag space structure is reached. Further increasing the lag space will not change the index significantly. In practical application, the curve of the index versus the number of n_a and n_b can be plotted to find out the knee-point, where the order index flattens out. The knee-point corresponds to the number of n_a and n_b. According to He and Asada's method, the number of n_a and n_b for the data used to identify the MR damper is two. Therefore, the model structure for identifying the MR damper was a second order model. Up to here, the determined model structures were composed of six input signals (two past displacements, two past voltages and two past forces) and one predicted signal (current force for the forward model or current voltage for the inverse model). Fig. 5 shows the fully connected neural network with 12 neurons in the hidden layer.

3.4. Optimal network architecture

Because the model selection including regressor and network selection is subjective to some extent, the constructed networks are not optimal. The so-called OBS strategy [3] gives a possibility for automatic optimization of the network architecture. The OBS is an important strategy and is a unique method that has been implemented for models of dynamic systems. According to the OBS strategy, the optimal network architecture can be determined by pruning the superfluous weights from the network. The OBS strategy considers that if a network has the smallest training error, then it is an optimal network. Implementation of the optimization process is to retrain each of the intermediate network and calculate the training error each time when a weight has been eliminated, then select the network with the smallest training error as the final network. Pruning of a weight deletes the corresponding link. Pruning of all links that connect a neuron to the rest of the topology deletes the neuron, which is within hidden neurons. Hence, pruning of weights simplifies the complexity of initial network topologies and improves the comprehension of data processing and topological parts in the network.

In order to implement the optimization process, the network is trained to a local minimum of training error E and the neighborhood of the local minimum is quadratic. When a weight is eliminated, the change in training error E is estimated through the following steps:

(1) expanding the criterion E to second order around an extremum θ_*;
(2) finding the changes of the remaining weights resulting from elimination of a weight, and retraining the remaining weights in the quadratic approximation to the new minimum of E; and
(3) computing the associated changes in training error.

For convenience of formularization, criterion (12) is rewritten by

$$V_N(\theta, Z^N) = E_N(\theta, Z^N)\tfrac{1}{2}\theta^{\mathrm{T}} D\theta, \tag{21}$$

where $V_N(\theta, Z^N)$ represents the regularized criterion and $E_N(\theta, Z^N)$ represents the unregularized criterion. When the jth weight is eliminated, the change of training error E is expressed by

$$\delta E_j = \gamma_j \theta_*^{\mathrm{T}} D H^{-1}(\theta_*) e_j + \tfrac{1}{2}\gamma_j^2 e_j^{\mathrm{T}} H^{-1}(\theta_*) E_N''(\theta_*, Z^N) H^{-1}(\theta_*) e_j. \tag{22}$$

The changes of all remaining weights are

$$\delta\theta = -\gamma_j H^{-1}(\theta_*) e_j. \tag{23}$$

In expressions (22) and (23), $H(\theta_*)$ is the Hessian matrix of the regularized criterion

$$H(\theta_*) = E_N''(\theta_*, Z^N) + \frac{1}{N} D, \tag{24}$$

where $E_N''(\theta_*, Z^N)$ is the matrix ($\theta = \theta_*$) of second partial derivative of $E_N(\theta_*, Z^N)$ with respect to θ, e_j is the jth unit vector and γ_j is the Lagrange multiplier which can be determined by

$$\gamma_j = \frac{\theta_*^{\mathrm{T}} e_j}{e_j^{\mathrm{T}} H^{-1}(\theta_*) e_j}. \tag{25}$$

3.5. *Estimation and validation of neural network*

Based on the developed methodologies, the neural network shown in Fig. 7 was trained by using the data shown in Fig. 2. The fully connected neural network had 12 neurons in the hidden layer. Using the pruning technology described previously, the optimal network was

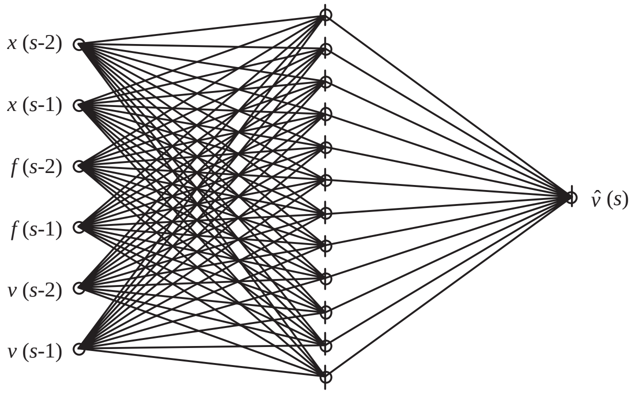

Fig. 7. Fully connected neural network.

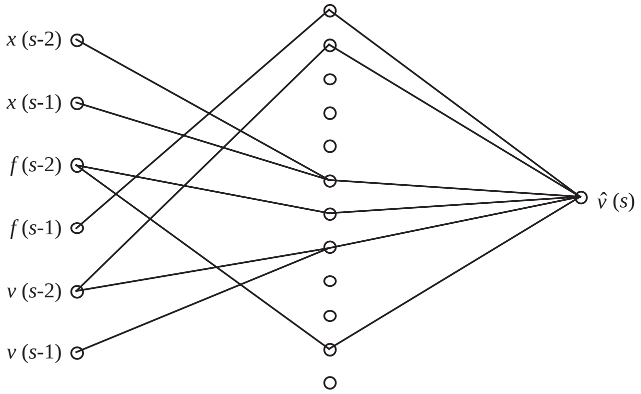

Fig. 8. Optimal neural network.

obtained. Fig. 8 shows the optimal network in which 83 weights were eliminated, about 85.5% of the total weights.

The trained optimal network was used to predict signals. After the displacement and force shown in Fig. 2 are input into the optimal model, the predicted voltage and its comparison with the desired (or required) voltage to the MR damper are shown in Fig. 9. The graphical comparison of the predicted and desired signals was very well. The optimal network was trained successfully.

Another way to verify the optimal network is to study quantitatively the errors between predicted and desired signals. The following expressions represent the errors:

$$E_v = \sqrt{\sum\nolimits_{s=1}^{N} [v_d(s) - v_p(s)]^2} \Big/ \sqrt{\sum\nolimits_{s=1}^{N} [v_d(s) - \mu_v]^2}, \tag{26}$$

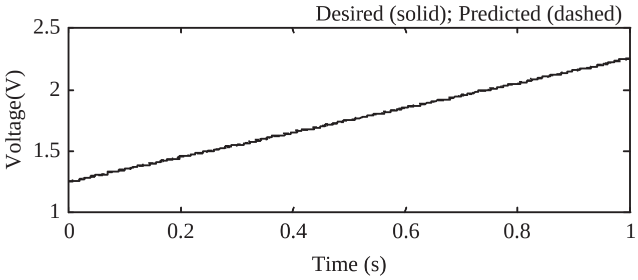

Fig. 9. Comparison of desired and predicted voltage.

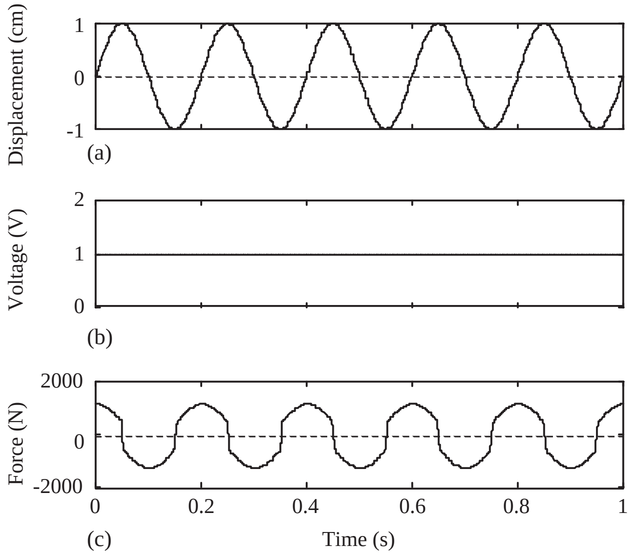

Fig. 10. First set of signals for validating optimal network.

where v_d is the desired voltage applied to the MR damper, v_p is the corresponding voltage predicted by the optimal network, and μ_v is the mean value of the desired voltage. The normalized error was calculated as $E_v = 0.0044$ which is very small.

Although the optimal network was trained very well, it still needs to be validated by using other signals. According to the characteristic of the operating conditions for the MR damper, two sets of signals were used to verify the optimal network. The first set of signal was composed of a sinusoidal displacement with 1.0 cm amplitude and 5.0 Hz frequency and a constant voltage with 1.0 V level. The force generated by the MR damper was periodic with time increasing. Fig. 10 shows these signals. The second set of signal consisted of a random displacement and a random step voltage. The force generated by the MR damper was also random. Fig. 11 shows the random signals.

Applying the validating displacements and forces to the optimal model, the voltages predicted by the network and their comparison with the desired voltages are shown in Figs. 12 and 13, corresponding to the first and second set of signal, respectively. The curves of the predicted and

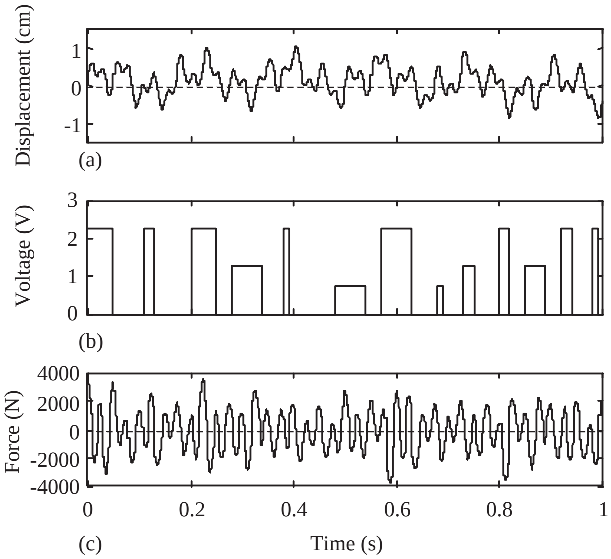

Fig. 11. Second set of random signals for validating optimal network.

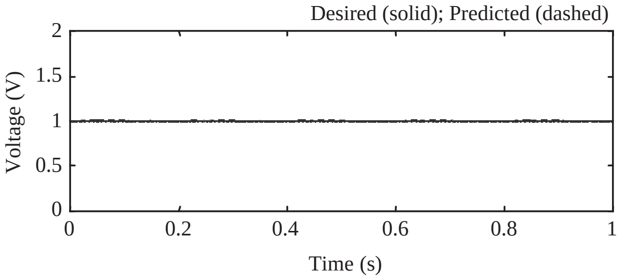

Fig. 12. Comparison of desired and predicted voltage signals using first set of validating data.

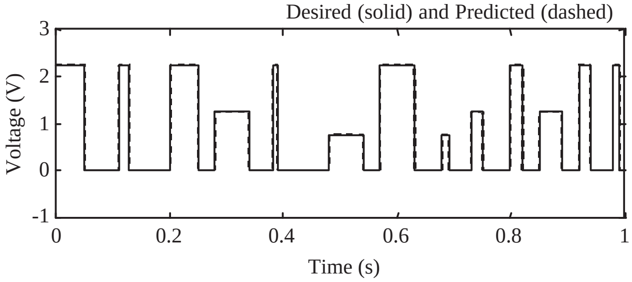

Fig. 13. Comparison of the desired and predicted voltage signals using second set of validating data.

desired voltages agreed very well. The normalized errors calculated by using formula (26) were only $E_v = 0.0018$ and 0.0020, corresponding to the first and second set of the validating signals, respectively.

The direct inverse modelling is the simplest approach conceptually. The MR damper output (force) was used as an input to the network. The network output was compared with the MR damper input (voltage) and this error was used to train the network. This structure tended clearly to force the network to represent an inverse model of the MR damper. However, it should be noted that there is a drawback to the inverse modelling approach: the learning procedure is not 'goal-oriented' [11]. In other words, the training signal must be chosen to sample over a wide range of system inputs, and the actual operational inputs may be hard to define a priori. The actual goal in the control context is to make the system output behave in a desired way, and thus the training signal in direct inverse modelling may not correspond to the explicit goal. To overcome the problem, a specialized inverse learning approach [12], which is based on the error between desired system outputs and actual outputs, can be used for inverse modelling.

4. Conclusions

(1) An inverse model of a MR damper was developed using optimal neural network and system identification techniques. The trained optimal network can accurately predict the required voltage for the desired force. The developed inverse model overcomes the limitation due to the complexity in solving the non-linear mathematical equations describing the performance of MR damper. If the optimal neural network is used in a control system, it is easy to implement semi-active vibration control by using the MR damper.

(2) An artificial neural network together with the system identification theory is a very effective and reliable methodology for identifying non-linear dynamic systems without knowing their inherent physical nature and functions. In order to obtain the best identifier, the network architecture including a set of regressor must be selected carefully.

(3) The OBS strategy is an effective technology to optimize neural network. The strategy can overcome the difficulty in determining the number of neurons in the hidden layer during the construction of a neural network. The OBS technology can simplify the complexity of initial network topologies and improve the comprehension of data processing and the topological parts in the network by pruning the superfluous weights from the network.

References

[1] B.F. Spencer Jr., S.J. Dyke, M.K. Sain, J.D. Carlson, Phenomenological model for magnetorheological dampers, Journal of Engineering Mechanics, American Society of Civil Engineers 123 (3) (1997) 230–238.

[2] J. Ghaboussi, A. Joghataie, Active control of structures using neural networks, Journal of Engineering Mechanics 121 (4) (1995) 555–567.

[3] L.K. Hansen, M.W. Pedersen, Controlled growth of cascade correlation nets, in: M. Marinaro, P.G. Morasso (Eds.), Proceedings of ICANN'94, Sorrento, Italy, 1994, pp. 797–800.

[4] R.A. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks 1 (1988) 295–307.

[5] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing: Explorations in Microstructure of Cognition, 1: Foundations, MIT Press, Cambridge, MA, 1986.

[6] T. Soderstrom, P. Stoica, System Identification, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[7] L. Ljung, System Identification—Theory for the user, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[8] R. Fletcher, Practical Methods of Optimization, Wiley, New York, 1987.

[9] X. He, H. Asada, A new method for identifying orders of input–output models for nonlinear dynamic systems, Proceedings of the American Control Conference, S.F., California, 1993.

[10] B. Hassibi, D.G. Stork, in: S.J. Hanson, et al. (Eds.), Second Order Derivatives for Network Pruning: Optimal Brain Surgeon, NIPS 5, Vol. 164, Morgan Kaufmann, San Mateo, 1993.

[11] M.I. Jordan, D.E. Rumehart, Forward models: supervised learning with a distal teacher, Cognitive Science 16 (3) (1992) 307–354.

[12] D. Psaltis, A. Sideris, A.A. Yamamura, A multiayered neural network cotroller, IEEE Control Systems Magazine 8 (1988) 17–21.