Scalable and Invertible PMNN Model for MagnetoRheological Fluid Dampers

M. Cao, K.W. Wang and Kwang Y. Lee

Journal of Vibration and Control 2008; 14; 731
DOI: 10.1177/1077546307083988

Scalable and Invertible PMNN Model for Magneto-Rheological Fluid Dampers

M. CAO
General Motors R&D, REB, 30500 Mound Road, Warren, MI 48071, USA
(ming.cao@gm.com)

K. W. WANG
Department of Mechanical and Nuclear Engineering, Pennsylvania State University, 157E Hammond Building, University Park, PA 16802, USA

KWANG Y. LEE
Department of Electrical and Computer Engineering, Baylor University, One Baylor Place #97356 Waco, TX 76798-7356, USA

(Received 26 November 2006; accepted 31 July 2007)

Abstract: To advance the state of the art of physical-principle-enhanced hybrid artificial neural network (ANN) modeling, network configurations with parallel modules (PMNN) reflecting the structural information of the physical principles have been developed (Cao, 2001). In this paper, the PMNN configuration is applied to develop a scalable and invertible dynamic magneto-rheological (MR) fluid damper model. To advance the state of the art and address issues of the current ANN-based MR damper models found in the open literature, two ANN-based MR damper models are developed in this study. The first one is a conventional first-principle-enhanced hybrid neural network model (defined as the baseline model in the current study) that improves upon the previous ANN-based MR damper models by introducing feedback loops to represent the dynamic behaviors of the MR damper. A PMNN-based MR damper model is then derived to further improve the control-input-output scalability and realize the invertible model concept. "Input-output scalability" refers to the model's capability to accurately estimate the system response with input profiles significantly different from the training data. "Invertible model" means that the resultant forward model can be directly transformed into an inverse model through a simple algebraic operation. The network training/testing results indicate that while both models provide satisfactory performance, the PMNN model outperforms the baseline model by showing superior control-input-output scalability. The candidacy of PMNN as a control-oriented actuator modeling tool is further strengthened by the fact that it is invertible, in other words, the inverse model with desired force as input and the control signal–voltage as output can be easily established by algebraically manipulating the forward model. This study indicates that PMNN, as a scalable and invertible dynamic modeling tool, is feasible for developing system-design-oriented models of vibration control purposes.

Key words: Actuator, parallel modulated neural network (PMNN), vibration control, dynamic modeling.

Journal of Vibration and Control, **14(5):** 731–751, 2008 DOI: 10.1177/1077546307083988

Figures 3, 5 appear in color online: http://jvc.sagepub.com

1. BACKGROUND: MR DAMPER AND SCALABLE/REVERTIBLE ANN MODELING TECHNIQUE

1.1. MR Damper

Magneto-rheological (MR) fluids have been investigated by many researchers. One important feature of such fluids is that their material properties can be modulated through an applied electro-magnetic field. More specifically, they are capable of reversibly changing from a linear Newtonian fluid to a semi-solid within milliseconds, and the yield strength of this semi-solid is controllable. This characteristic makes the MR fluid-based devices attractive candidates for vibration control and isolation. Therefore, MR fluid dampers have been developed for damping augmentation of various mechanical systems (Carlson et al., 1996; Hiemenz and Wereley, 1999; Marathe et al., 1999; Gandhi et al., 2001; Lai and Liao, 2002; Choi and Wereley, 2003; Kavlicoglu et al., 2006; Koo et al., 2006), where promising results have been shown.

1.2. Previous MR Damper Modeling Efforts

Significant efforts have been made to model the behaviors of the MR (or electro-rheological (ER) fluids which have similar characteristics) fluids. Peel et al. (1996) and Kamath et al. (1996) focused on establishing models based on mechanisms and physical principles. Developed on the basis of an idealized nonlinear Bingham plastic shear flow mechanism, the quasi-steady dashpot damper model by Kamath et al. (1996) reveals the insight mechanisms of the MR damper:

$$\tau = \tau_y \operatorname{sgn}\left(\frac{du}{dr}\right) + \mu\frac{du}{dr}, \quad |\tau| > |\tau_y| \tag{1a}$$

$$\frac{du}{dr} = 0, \quad |\tau| < |\tau_y| \tag{1b}$$

where τ_y is the dynamic yield stress and is assumed to be a polynomial function of the applied field, u is the velocity, r is the radial coordinate, and μ is the plastic viscosity and is assumed to be independent of the applied field for simplification. Newtonian shear flow can be viewed as a special case of Bingham plastic shear flow with zero dynamic yield stress. Comparison with experimental results confirmed that this Bingham plastic model could provide modestly accurate prediction of the damping force. Since ideal assumptions were made to facilitate the mathematical analysis, these models are not accurate enough for simulation and controller design purposes, although they can provide some design guidelines.

Spencer et al. (1997) presented a numerically tractable yet accurate MR phenomenological model. The rationale behind this approach is to combine the Bouc-Wen model (Wen, 1976), which is capable of exhibiting a wide variety of hysteretic behavior, with the spring-linear damper-Coulomb friction component model. The total force generated by the MR damper is formulated as follows (Spencer et al., 1997):

$$F = \alpha Z + c_0(\dot{x} - \dot{y}) + k_0(x - y) + k_1(x - x_0) = c_1\dot{y} + k_1(x - x_0) \tag{2}$$

where z is the revolutionary variable in Bouc-Wen (Wen, 1976) model, k_1 represents the stiffness of the accumulator, c_1 intends to simulate the roll off at low velocities when the velocity is decreasing, c_0 stands for the viscous damping coefficient observed at large velocities, and k_0 is the stiffness for large velocities. The forces applied on either sides of the rigid bar are in balance:

$$c_1\dot{y} = \alpha z + k_0(x - y) + c_0(\dot{x} - \dot{y}). \tag{3}$$

The constant parameters α, c_1, c_0 in equations (2) and (3) are all linearly correlated with the applied field strength u. Comparisons with empirical results obtained from a prototype MR damper showed that this model could accurately predict the response of the MR damper over a wide range of operating conditions.

The model developed by Spencer et al. (1997) can be easily implemented in dynamic simulations, so it is a convenient tool for studying system response. However, its complexity makes it hard to use for developing MR-damper-based control strategies. Therefore, more efforts followed for deriving accurate yet simple MR damper models. One of the promising approaches is to utilize tools such as artificial neural networks (ANN), as investigated by Chang and Roschke (1998), Chang and Zhou (2002) and Du et al. (2006).

1.3. Dynamic ANN Modeling Methodologies

Initially inspired by the observed structure of biological neural processing systems, neural networks have gained wide applications in many fields, including classification, signal processing, data compression, pattern recognition, feature detection, system modeling and control. The application of neural networks as system modeling tools is rooted in their ability to approximate functions (Cybenko, 1989; Hornik et al., 1989). The idea of incorporating physical knowledge into the ANN model has been extensively explored, aiming at providing more accurate description of the system characteristics. It has been demonstrated that a gray-box hybrid approach, combining physical principles and the neural networks, can significantly improve the system performance and outperform the traditional black-box ANN model (Cao et al., 2004a,b; Aceves et al., 2006). While promising, the major bottle neck of neural network as a design tool is that it cannot guarantee its accuracy outside the training range. It was pointed out (Cao, 2001; Cao et al., 2005) that one way to address this issue and improve the model scalability (defined as the ANN model's capability to account for input feature space outside the training data range) is to introduce the first-principle structural information into the neural network configuration, if this information is available. Depending on the available physics and the system, two different approaches were proposed: the dynamic parallel modulated neural network (PMNN) concept (Cao et al., 2005) and the non-dimensional artificial neural network (NDANN) structure (Cao et al., 2006). While both approaches demonstrate promising input-output scalability and are thus good tools for system design, the PMNN has greater potential for developing control-oriented actuator models, as showcased by the PMNN model for automotive friction components (Cao, 2001; Wang and Cao, 2001; Cao et al., 2005).

2. PROBLEM STATEMENT AND OVERALL APPROACH

Chang and Roschke (1998) investigated the performance of an MR damper model based on ANN. The results demonstrated that the well-trained black-box neural network model can accurately predict the output force of an MR damper. In Chang and Roschke's (1998) approach, the measured damper forces of previous time steps are used as inputs to the forward network models. While interesting, this model cannot be applied for simulation and design purposes when the measured force is not available during these stages. More importantly, with static mapping, the dynamic characteristics of the system cannot be well represented. Therefore, the first objective of this investigation is to derive an MR damper model based on recurrent network architecture with feedback loops, which takes care of the dynamic characteristics of the MR damper. This model will be applied as a representative of conventional ANN approaches, and compared with the PMNN-based model aiming at demonstrating the superior input-output scalability of PMNN.

Inverse MR damper models for controller design purpose have also been proposed (Chang and Zhou, 2002). The inverse MR damper model is a standalone model based on a recurrent neural network architecture. Therefore, a complete two-way mapping is achieved by two models: a forward model and an inverse model. The input and output of the forward model are the control signal (voltage for MR damper) and the damper force, respectively. The inverse model outputs control voltage using the desired force response as input. Du et al. (2006) applied a similar approach for capturing the forward and inverse dynamic behaviors of an MR damper using an evolving radial basis function (RBF) network. In this investigation, genetic algorithm (GA) was combined with other learning algorithms for network training and satisfying performance was demonstrated. While these approaches have opened up a good path for MR damper modeling, there are still rooms for improvement. One issue is that since two models (forward and inverse) are required for one process, training of the networks could be numerically intensive. The other issue is that the ANN cannot account for input feature space outside the training data range, that is, the input-output-scalability is not guaranteed.

To further enhance the model's applicability for controller design and to improve voltage scalability of the recurrent ANN based model, the second and the major objective of this study is to develop a more advanced MR damper model utilizing the PMNN concept. Two major advantages of PMNN are:

(a) It is a control-oriented approach providing promising input-output scalability, and thus benefits system design.
(b) The forward model can be easily transformed into an inverse model form through simple algebraic operation, i.e., only one model is needed for both forward and inverse mapping. Subsequently, the network training only needs to be done once.

The organization of the paper is as follows: The basic concept of the physical principle-based neural network with parallel modules (PMNN) is discussed in the next section (Section 3). In Section 4, the dynamic MR damper model based on a recurrent network configuration is presented. This model serves as the baseline model versus the more advanced PMNN model. Section 5 discusses the derivation of the PMNN MR damper model and compares its performance with the baseline model. The procedure to algebraically derive the

inverse model from the forward PMNN model is also briefly discussed in Section 5. The study is concluded in Section 6.

3. PMNN – A SCALABLE CONTROL-ORIENTED APPROACH

This section starts with a simple system demonstrating the basic idea of PMNN. We assume the system can be physically formulated as:

$$\begin{aligned} Y(\bar{X}) &= Fc_1(\bar{x}_1) + Fc_2(\bar{x}_2) + ... + Fc_{n_m}(\bar{x}_{n_m}) \\ \bar{X} &= \bar{x}_1 \cup \bar{x}_2 \cup ... \cup \bar{x}_{n_m} \end{aligned} \tag{4}$$

where Fc_i (i = 1, 2, ..., n_m) represent the components of the system output, and n_m is the number of modules. $\bar{x}_i$'s are the input variable vectors, corresponding to the respective system output components. Each subset $\bar{x}_i$ (i = 1, 2, ..., n_m) consists of a few elements from $\bar{X}$. Therefore, they might share vector elements with each other, since an input might be influential for several different components.

The conventional network model architecture for the system formulated in equation (4) is shown in Figure 1. This configuration mixes all the input signals and processes them through one multi-layer neural network. With feedback loops added, the ANN depicted in Figure 1 can be easily transformed into a recurrent multi-layer-perceptron (MLP) and can thus be applied to model dynamic systems. Figure 2 shows the newly derived neural-network paradigm, where the resultant neural network model is

$$Y = \sum_{i=1}^{n_m} NN\_out_i \tag{5}$$

i.e., the activation function of the output neuron is

$$H(y_1, y_2, ..., y_{n_m}) = \sum_{i=1}^{n_m} y_i. \tag{6}$$

The activation function of the output neuron is chosen to be linear to realize the additive characteristic of the physical system (equation (4)). By applying n_m sub-neural-networks (parallel modules) to capture the flavor of the n_m components of the system output Y (Figure 2), the effects of the n_m output components are separated explicitly. This approach, as a very simple form of PMNN, results in a more accurate model including the structure of the physical principles. Similar to the conventional network structure, the PMNN configuration can be applied for modeling dynamic systems through simply adding feedback loops.

Obviously, this modulated neural network idea can be expanded to include other correlation architectures, for instance, if a system to be modeled can be written as:

$$Y = \prod_{i=1}^{n_m} F_i(\bar{x}_i) \tag{7}$$

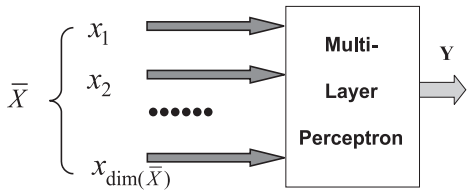

Figure 1. Conventional neural network model.

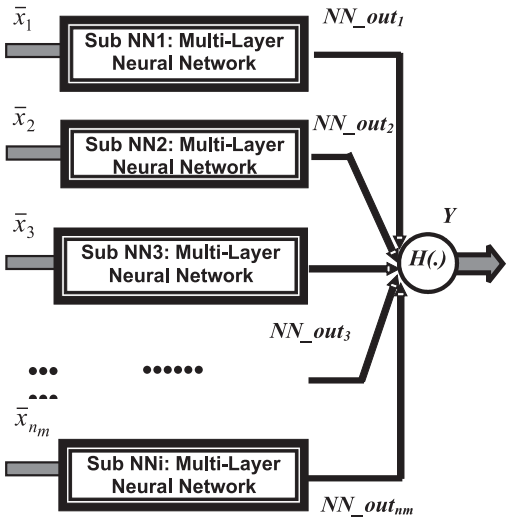

Figure 2. Parallel modulated neural network model

then the resultant neural network model will become:

$$Y = \prod_{i=1}^{n_m} NN\_out_i \tag{8}$$

where NN_out_i represents the output of the *i-th* sub network.

Based on the above discussions, the success of PMNN in system modeling relies on the fact that the system can be physically formulated as separate parallel modules.

For dynamic system modeling, the general formulas for the PMNN model have been derived as (Cao, 2001; Wang and Cao, 2001; Cao et al., 2005):

$$\begin{aligned} Y_i(k) &= H_i\{\bar{W}^i_{hid\_out}{}^T \bullet \bar{F}_i[\tilde{W}^i_{inp\_hid} \bullet \bar{x}_i^T]\}, \quad i = 1, 2, ..., n_m \\ Y(k) &= H\{Y_1(k), Y_2(k), ..., Y_{n_m}(k)\} \qquad (9a) \\ \bar{x}_i^T &= (\bar{x}_i^{s\,T}, Y(k-1), Y(k-2), ..., Y(k-n_o), u(k), u(k-1), ..., u(k-n_u)) \end{aligned}$$

$$
\begin{aligned}
Y_i(k) &= H_i\{\bar{W}^i_{hid\_out}{}^T \bullet \bar{F}_i[\tilde{W}^i_{inp\_hid} \bullet \bar{x}_i^T]\}, \quad i = 1, 2, ..., n_m \\
Y(k) &= H\{Y_1(k), Y_2(k), ..., Y_{n_m}(k)\} \qquad (9b) \\
\bar{x}_i^T &= (\bar{x}_i^{sT}, Y(k-1), Y(k-2), ..., Y(k-n_o), u(k), u(k)^2, ..., u(k)^{n_p}).
\end{aligned}
$$

In this paper, the default vector orientation is column. In equation (9), Y is the system output, Y_i's are the outputs of the individual modules, i is the module index, $\bar{x}_i^s$ is the state variables for the i^{th} module, $\bar{x}_i$ is the input vector for the i^{th} sub-network (module), $\bar{W}^i_{hid\_out}$ (vector dimension: N_hid_i, column vector) is the network weighting vector from the hidden layer to the output layer for the i^{th} module, N_hid_i is the number of hidden layer neurons for the i^{th} module; $\tilde{W}^i_{inp\_hid}$ (dimension: $N\_hid_i \times N\_inp_i$) is the network weighting matrix from the input layer to the hidden layer for the i^{th} module, N_inp_i is the number of NN input for the i^{th} module; $\bar{F}_i$ is the activation function vector of the hidden layer neurons for the i^{th} module, and u is the control signal, n_o is the order of the output-delay, n_u is the order of the input-delay. The operator "$\bullet$" is the inner multiplication of vector/vector or matrix/vector pairs. The activation function of the output neuron H is derived based on the structural information of the available first-principle-based or empirical correlations. Equation (9b) is a variation of equation (9a) with n_u equals zero. Therefore, only the current control signal, as well as the higher order polynomial terms of it (with n_p as the highest polynomial order), are applied as the inputs for the individual modules. Considering that the time history of the output already contains the information concerning previous time steps' control input, equations (9a) and (9b) can actually be regarded as equivalent with suitably chosen network configurations and sizes. In equation (9), the current output is related with the output of previous time steps, which makes it a dynamic formula.

The PMNN correlation between the system output and the control input/state variables can be formulated in a way more explicitly reflecting structural information of physical principles, provided this kind of information is available (Cao, 2001; Wang and Cao, 2001):

$$
\begin{aligned}
Y_i(k) &= H_i\{\bar{W}^i_{hid\_out}{}^T \bullet \bar{F}_i[\tilde{W}^i_{inp\_hid} \bullet (\bar{x}_i^{sT}, Y(k-1), Y(k-2), ..., Y(k-n_o)]\}, \\
i &= 1, 2, ..., n_m \\
Y(k) &= H_p\{Y_1(k), Y_2(k), ..., Y_{n_m}(k), \bar{x}_s^T, u(k)\}, \qquad (10a) \\
\bar{x}_s &= \bar{x}_1^s \cup \bar{x}_2^s ... \cup \bar{x}_{n_m}^s \\
& H_p - A \text{ polynomial function.}
\end{aligned}
$$

Equation (10a) is the foundation for the PMNN-based MR damper model to be presented in Section 5 of this paper. In this formula, the control input does not show up in the input layer of the sub-networks. Instead, it only appears in the output layer of the PMNN, correlating with the system output as a polynomial function. As will be shown later, this enables a single model applicable as a forward as well as an inverse model, which therefore benefits controller designs. It also improves the input-output scalability, which is often lacking in the conventional hybrid ANN modeling methodologies.

As shown by equation (10a), the input state variables ($\bar{x}_i^s$, $i = 1, 2, \ldots, n_m$) may also be inputs for the polynomial output formula (H_p) if scalability exists between them and the final PMNN output. Obviously, depending upon the system to be modeled, the input state variables can exclusively serve as inputs for the sub-networks (modules), i.e., they will not show up at all in the polynomial output formula:

$$Y(k) = H_p\{Y_1(k), Y_2(k), ..., Y_{n_m}(k), u(k)\}, \tag{10b}$$

The physics-based structure enables PMNN to provide accurate prediction even when the input is significantly different from the training data in both the amplitude and pattern. This is obviously desired for modeling control actuators, noticing that the design space for the controller does not necessarily fall in the range of the training data, which is usually collected under nominal operating conditions.

During the training process of PMNN, it makes sense to rearrange the weighting vectors and matrices of all the modules into one overall weighting vector. The overall PMNN weighting vector is formulated as:

$$\begin{aligned}
\bar{\mathbf{W}}_{PMNN} &= \{\bar{W}_1^T, \bar{W}_2^T, ..., \bar{W}_{n_m}^T\}^T, \\
\bar{W}_i &= \begin{pmatrix} \bar{W}^i_{inp_hid} \\ \bar{W}^i_{hid_out} \end{pmatrix}, \quad i = 1, 2, ..., n_m \\
\bar{W}^i_{inp_hid} &= \{\tilde{W}^i_{inp_hid}(1,1), \tilde{W}^i_{inp_hid}(1,2), ..., \tilde{W}^i_{inp_hid}(1, No_hid_i), \\
&\quad \tilde{W}^i_{inp_hid}(2,1), \tilde{W}^i_{inp_hid}(2,2), ..., \tilde{W}^i_{inp_hid}(2, No_hid_i), \\
&\quad \ldots, \\
&\quad \tilde{W}^i_{inp_hid}(No_inp_i, 1), \tilde{W}^i_{inp_hid}(No_inp_i, 2), ..., \\
&\quad \tilde{W}^i_{inp_hid}(No_inp_i, No_hid_i)\}^T.
\end{aligned} \tag{11}$$

Therefore, the dimension of the overall weighting vector is:

$$\dim_{PMNN} = \sum_{i=1}^{n_m}[(No\_inp_i + 1)No\_hid_i. \tag{12}$$

For a conventional dynamic MLP, the overall weighting vector is:

$$\begin{aligned}
\bar{\mathbf{W}}_{NN} &= \{\tilde{W}_{inp_hid}(1,1), \tilde{W}_{inp_hid}(1,2), ..., \tilde{W}_{inp_hid}(1, No_hid), \\
&\quad \tilde{W}_{inp_hid}(2,1), \tilde{W}_{inp_hid}(2,2), ..., \tilde{W}_{inp_hid}(2, No_hid), \\
&\quad \ldots, \\
&\quad \tilde{W}_{inp_hid}(No_inp, 1), \tilde{W}_{inp_hid}(No_inp, 2), ..., , \\
&\quad \tilde{W}_{inp_hid}(No_inp, No_hid)\bar{W}^T_{hid_out}\}^T.
\end{aligned} \tag{13}$$

In equation (13), the superscript i is dropped since now there is only one processing module, versus multiple ones of PMNN as shown in equation (11). The neural network correlation for the conventional approach is thus formulated as:

$$\begin{aligned} Y(k) &= H\{\bar{W}_{hid\_out}{}^{T} \bullet \bar{F}[\tilde{W}_{inp\_hid} \bullet (\bar{x}_s^T, Y(k-1), Y(k-2), ..., Y(k-n_o), \\ &\quad u(k), u(k-1), ...), u(k-n_u)]\} \end{aligned} \tag{14a}$$

where $\bar{W}_{hid\_out}$ is the weighting vector from the hidden layer to the output layer, $\bar{F}$ is the activation function operator of the hidden layer, $\tilde{W}_{inp\_hid}$ is the weighting matrix from the input layer to the hidden layer, $\bar{x}_s$ is the input state vector, $Y(k-jj),\ jj = 1, 2, ...n_o$ are the time-delayed output feedbacks. One possible variation of equation (14a) has n_u equals zero, and includes higher polynomial orders of the control input:

$$\begin{aligned} Y(k) &= H\{\bar{W}_{hid\_out}{}^{T} \bullet \bar{F}[\tilde{W}_{inp\_hid} \bullet (\bar{x}_s^T, Y(k-1), Y(k-2), ..., Y(k-n_o), \\ &\quad u(k), u(1)^2, ..., u(k)^{n_p}]\} \end{aligned} \tag{14b}$$

where n_p represents the highest polynomial order of the control input. The variations shown in equations (14a) and (14b) parallel with those demonstrated by equations (9a) and (9b) for PMNN.

Obviously, the dimension of the overall weighting vector for a conventional single MLP approach is:

$$\dim_{NN} = (No\_inp + 1)No\_hid. \tag{15}$$

Since the sizes of sub-networks of PMNN are usually much smaller than the conventional single MLP approach, the PMNN will not necessarily increase the numerical burden of network training. Actually, it has been shown many times that the PMNN converges faster than a conventional hybrid network model (Cao, 2001; Cao et al., 2005).

In the next section (Section 4) of this paper, the conventional recurrent MLP-based MR damper model will be synthesized using equation (14).

4. MR DAMPER MODEL BASED ON CONVENTIONAL HYBRID NETWORK APPROACH

The previous neural network MR damper model (Chang and Roschke, 1998), while demonstrating promising accuracy, cannot be directly applied for simulation purpose because the measured force is normally not available during these stages and the dynamic characteristics are thus not well represented. To address the dynamics of the MR damper, in this section, network outputs (estimated damper force) of previous time steps are used as NN inputs (instead of the measured damper force) to derive a conventional-hybrid-network-based model. The analytical data calculated using Spencer's model (1997) is applied for network training and testing.

It has been demonstrated that the hybrid network architecture integrated with physical principles can dramatically improve the performance of network models (Cao et al., 2004a,b; Aceves et al., 2006). The physical principles (equation (1)) suggest that the velocity is the dominant factor in determining the damping force of the MR fluid. Equation (1) also reveals that the output force of the MR damper strongly relies not only on the velocity but also on the sign of the velocity, indicating that correlation between the velocity and the damping force is strongly nonlinear. By only choosing the displacement of the piston head as network input, Chang and Roschke's model (1998) did not explicitly realize this relationship. Therefore, new network inputs reflecting the physical principles, such as the velocity of the piston head, are introduced to enhance the model performance. The square and cubic terms of the velocity are also used as neural network inputs to help capture the nonlinear characteristics of the force–velocity correlation. Another important relationship that needs to be reflected by the model is the voltage-dependence of the damping force. Since the MR damper is frequently used as a device for vibration control, accurately modeling this characteristic is critical for the success of controller design. As Spencer et al. (1997) pointed out, the system response saturates when the applied voltage is larger than a specified value (2.25 V for the damper tested by Spencer). That is, the relationship between the voltage and output force is also strongly nonlinear. Hence, linear and square terms of the applied voltage are introduced as network inputs to approximate the nonlinear force–voltage correlation. With all these improvements and new features, Figure 3 shows the MR damper model based on the conventional neural network architecture. The configuration shown in Figure 3 is indeed the same as that of Figure 1 except for the feedback loops. This network is a realization of the conventional recurrent network formula (equation 14b) with output and control input delay order n_o and n_u at two and zero respectively, and polynomial order of the control input n_p equals 2:

$$
\begin{aligned}
F_k &= H\{\bar{W}_{hid\_out}{}^T \bullet \bar{F}[\tilde{W}_{inp\_hid} \bullet \bar{x}^T]\} \\
\bar{x}^T &= (\bar{x}_s^T, Y(k-1), Y(k-2), u(k), u(k)^2) \\
\bar{x}_s^T &= [\dot{x}_k \ \dot{x}_k^2 \ \dot{x}_k^3 \ -1], \quad u(k) = v_k, \\
[Y(k) \ \ Y(k-1) \ \ Y(k-2)] &= [F_k \ \ F_{k-1} \ \ F_{k-2}].
\end{aligned} \tag{16}
$$

This ANN MR damper model is trained and tested based on the analytical data calculated using Spencer's model (1997). The analytical model provides a fast and easy way to create any kind of operating combination for network training and testing, which is particularly beneficial for validating the input-output scalability requiring some quite outrageous voltage (the control input for this case) profile. Once a model has been validated by legitimate analytical results, it usually performs well using real experimental data, as has been demonstrated by Cao et al. (2005). The system parameters used are illustrated in Table 1. Since the network model is a dynamic system, the dynamic Levenberg-Marquardt (Press et al., 1986) algorithm, a robust and fast scheme commonly used in neural network training, is used for network training. In the Levenberg-Marquardt algorithm, the Hessian matrix is approximated by a 1st order gradient, and the updating step is adaptively adjusted.

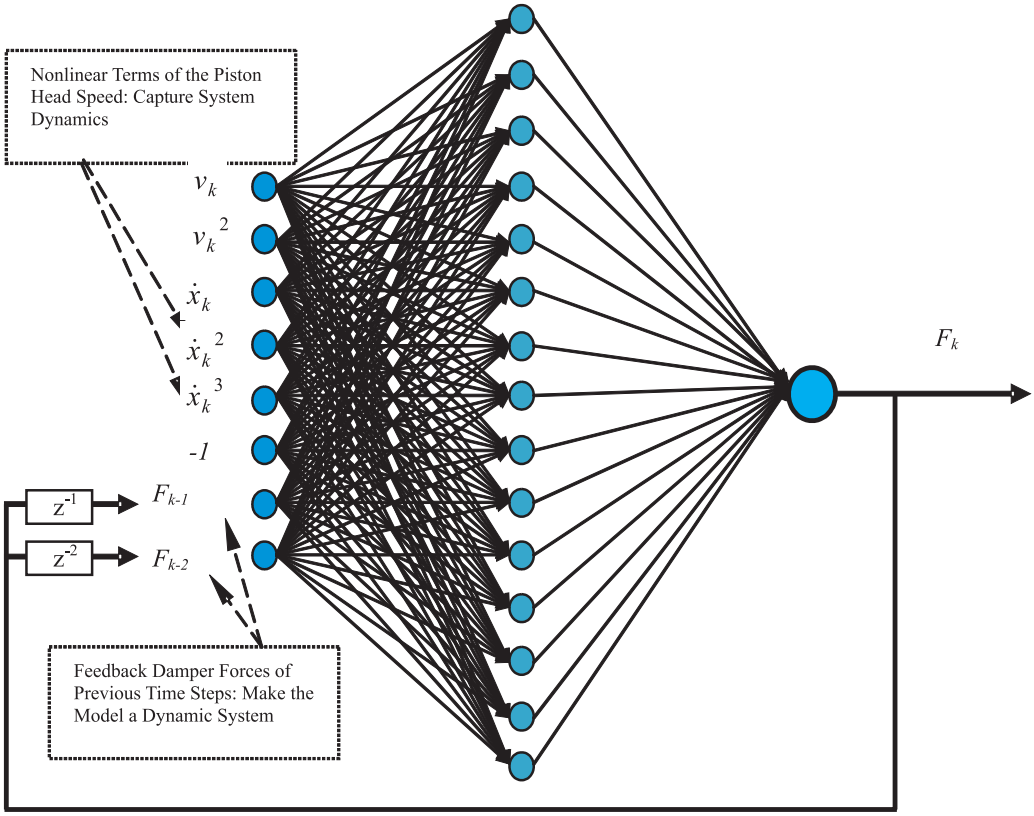

Figure 3. Dynamic MR damper model based on the conventional network approach.

A dynamic regression correlation to calculate the gradient of the network output with respect to the network weighting vector is applied to reflect the dynamic nature of this model:

$$\begin{aligned}
\frac{dF_k}{d\bar{\mathbf{W}}_{NN}} &= \frac{\partial F_k}{\partial \bar{\mathbf{W}}_{NN}} \quad if \quad k = 1 \\
\frac{dF_k}{d\bar{\mathbf{W}}_{NN}} &= \frac{\partial F_k}{\partial \bar{\mathbf{W}}_{NN}} + \frac{dF_k}{dF_{k-1}}\frac{dF_{k-1}}{d\bar{\mathbf{W}}_{NN}} \quad if \quad k = 2 \\
\frac{dF_k}{d\bar{\mathbf{W}}_{NN}} &= \frac{\partial F_k}{\partial \bar{\mathbf{W}}_{NN}} + \frac{dF_k}{dF_{k-1}}\frac{dF_{k-1}}{d\bar{\mathbf{W}}_{NN}} + \frac{dF_k}{dF_{k-2}}\frac{dF_{k-2}}{d\bar{\mathbf{W}}_{NN}} \quad if \quad k > 2
\end{aligned} \tag{17}$$

where F_k, F_{k-1}, F_{k-2} are the current, previous-time-step, and 2-time-step delayed system outputs, with k as the current time step. The definition of the overall weighting vector ($\bar{\mathbf{W}}_{NN}$) for this conventional dynamic MLP has been defined in equation (13). The difference between $\frac{\partial F_k}{\partial \bar{\mathbf{W}}_{NN}}$ and $\frac{dF_k}{d\bar{\mathbf{W}}_{NN}}$ is due to the fact that the system is a dynamic system with the feedback loops (that is, F_k is a function of network weightings, as well as the outputs of previous time

Table 1. MR Damper model parameters [Spencer et al., 1997].

Parameter	Value	Parameter	Value
c_{0a}	21.0 N.sec/cm	k_1	5.0 N/cm
c_{0b}	3.50 N.sec/(cm.v)	x_0	14.3 cm
c_{1a}	283.0 N.sec/cm	Γ	363 cm^{-2}
c_{1b}	2.95 N.sec/(cm.v)	B	363 cm^{-2}
α_a	140.0 N/cm	A	301
α_b	695 N/(cm.v)	n	2
k_0	46.9 N/cm	η	190 sec^{-1}

steps also correlating with the network weightings). For instance, to calculate the derivative of the output with respect to the hidden-output layer weight vector ($No\_hid \times 1$), we have partial differential as (from equation (16)):

$$\frac{\partial Y(k)}{\partial \bar{W}_{hid\_out}} = \frac{\partial F_k}{\partial \bar{W}_{hid\_out}} = \frac{\partial H(Z)}{\partial Z}\Big|_{z=Y_o(k)} \bar{F}[\bar{X}_o(k)] \tag{18a}$$

$$Y_o(k) = \bar{W}_{hid\_out}{}^T \bullet \bar{F}[\bar{X}_o(k)] \tag{18b}$$

$$\begin{aligned} \bar{X}_o(k) &= [X_o^1(k) X_o^2(k) \ldots X_o^{No\_hid}(k)]^T \\ &= \tilde{W}_{inp\_hid} \bullet [\bar{x}_s^T, Y(k-1), Y(k-2), u(k)u(k)^2]^T \end{aligned} \tag{18c}$$

$$\bar{F}[\bar{X}_o(k)] = \left\{ \begin{array}{l} F^1[X_o^1(k)] \\ F^2[X_o^2(k)] \\ \ldots \\ F^{No\_hid}[X_o^{No\_hid}(k)] \end{array} \right\}. \tag{18d}$$

For a static system:

$$\frac{dF_k}{d\bar{W}_{hid\_out}} = \frac{\partial F_k}{\partial \bar{W}_{hid\_out}} \tag{19}$$

but since the current conventional-network-based MR damper model contains inputs that are actually time-delayed system outputs (equation (16)), which also depends on the network weightings, regression terms have to be added for accurate gradient calculation (equation (18c)):

$$\begin{aligned} \frac{dY(k)}{d\bar{W}_{hid\_out}} &= \frac{\partial Y(k)}{\partial \bar{W}_{hid\_out}} + \frac{dY(k)}{dY(k-1)} \frac{dY(k-1)}{d\bar{W}_{hid\_out}} + \frac{dY(k)}{dY(k-2)} \frac{dY(k-2)}{d\bar{W}_{hid\_out}} \\ &= \frac{\partial Y(k)}{\partial \bar{W}_{hid\_out}} + \frac{dH(Z)}{dZ}\Big|_{Z=Y_o(k)} \bar{W}_{hid\_out}{}^T \bullet \left\{ \left[\frac{d\bar{F}[\bar{X}]}{d\bar{X}} \Big|_{\bar{X}=\bar{X}_o(k)} \right. \right. \end{aligned}$$

$$\left.\left.\frac{d\bar{X}_o(k)}{dY(k-1)}\right]\frac{dY(k-1)}{d\bar{W}_{hid\_out}}+\left[\frac{d\bar{F}[\bar{X}]}{d\bar{X}}\Big|_{\bar{X}=\bar{X}_o(k)}\frac{d\bar{X}_o(k)}{dY(k-2)}\right]\frac{dY(k-2)}{d\bar{W}_{hid\_out}}\right\}$$

$$where \quad \left[\frac{d\bar{F}[\bar{X}]}{d\bar{X}}\Big|_{\bar{X}=\bar{X}_o(k)}\frac{d\bar{X}_o(k)}{dY(k-kk)}\right]$$

$$= \begin{pmatrix} \frac{dF^1[X^1]}{dX^1}\Big|_{X^1=X_0^1(k)}\ \tilde{W}_{inp\_hid}(1,i_{kk}) \\ \frac{dF^2[X^2]}{dX^2}\Big|_{X_2=X_0^2(k)}\ \tilde{W}_{inp\_hid}(2,i_{kk}) \\ \cdots \\ \frac{dF_1[X^{No\_hid}]}{dX^{No\_hid}}\Big|_{X^{No\_hid}=X_0^{No\_hid}(k)}\ \tilde{W}_{inp\_hid}(No\_hid,i_{kk}) \end{pmatrix}, \ kk=1,\ 2$$

i_{kk} is the position number of $Y(k-kk)$ in the input layer (Fig. 3) :

$$i_1 = 7; i_2 = 8. \tag{20}$$

Similarly, the partial differential of the output with respect to the input-to-hidden-layer weighting matrix, which is a 2-dimensional tensor, can be calculated as:

$$\frac{\partial F_k}{\partial \tilde{W}_{inp\_hid}} = \frac{\partial H(Z)}{\partial Z}\Big|_{Z=Y_o(k)}\ \bar{W}_{hid\_out}{}^T \bullet \frac{\partial \bar{F}[\bar{X}]}{\partial \bar{X}}\Big|_{\bar{X}=\bar{X}_o(k)}\frac{\partial \bar{X}_o(k)}{\partial \tilde{W}_{inp\_hid}}$$

$$i.e.,$$

$$\frac{\partial F_k}{\partial \tilde{W}_{inp\_hid}(ii,jj)} = \frac{\partial H(Z)}{\partial Z}\Big|_{Z=Y_o(k)}\ \bar{W}_{hid\_out}(ii)\frac{\partial \bar{F}^{ii}[X^{ii}]}{\partial X^{ii}}\Big|_{X^{ii}=X_0^{ii}(k)}\ \bar{x}(jj)$$

$$ii = 1,2,...,No\_hid,\ jj = 1,2,...,No\_inp. \tag{21}$$

In equation (21), $\bar{x}(jj)$ is the jj^{th} input of the network as defined in equation (16). A regression formula similar to equation (20) completes the calculation of the dynamic derivative with respect to the input-layer-hidden-layer weighting matrix.

Constant applied voltage, and sinusoidal velocity signal with combinations of different frequencies and magnitudes are used to derive the training data sets for network training. The well-trained MR damper model is then tested by analytical results obtained based on operating conditions different from the training sets.

As Figure 4(a) shows, the testing voltage profile remains at zero for the first 0.5 seconds, then starts to increase until the saturation voltage (2.25 volt.) is reached. This voltage pattern is different from the constant voltage applied in the training cases, while the magnitude of the voltage remains the same as the training data sets.

The network testing results are illustrated by Figures 4(b), (c) and (d). The comparison between the time history of the neural network output and the analytical result is demonstrated in Figure 4(b). As this figure shows, the neural network output agrees well with the analytical result. Figure 4(c) shows the relationship between the damping force and the dis-

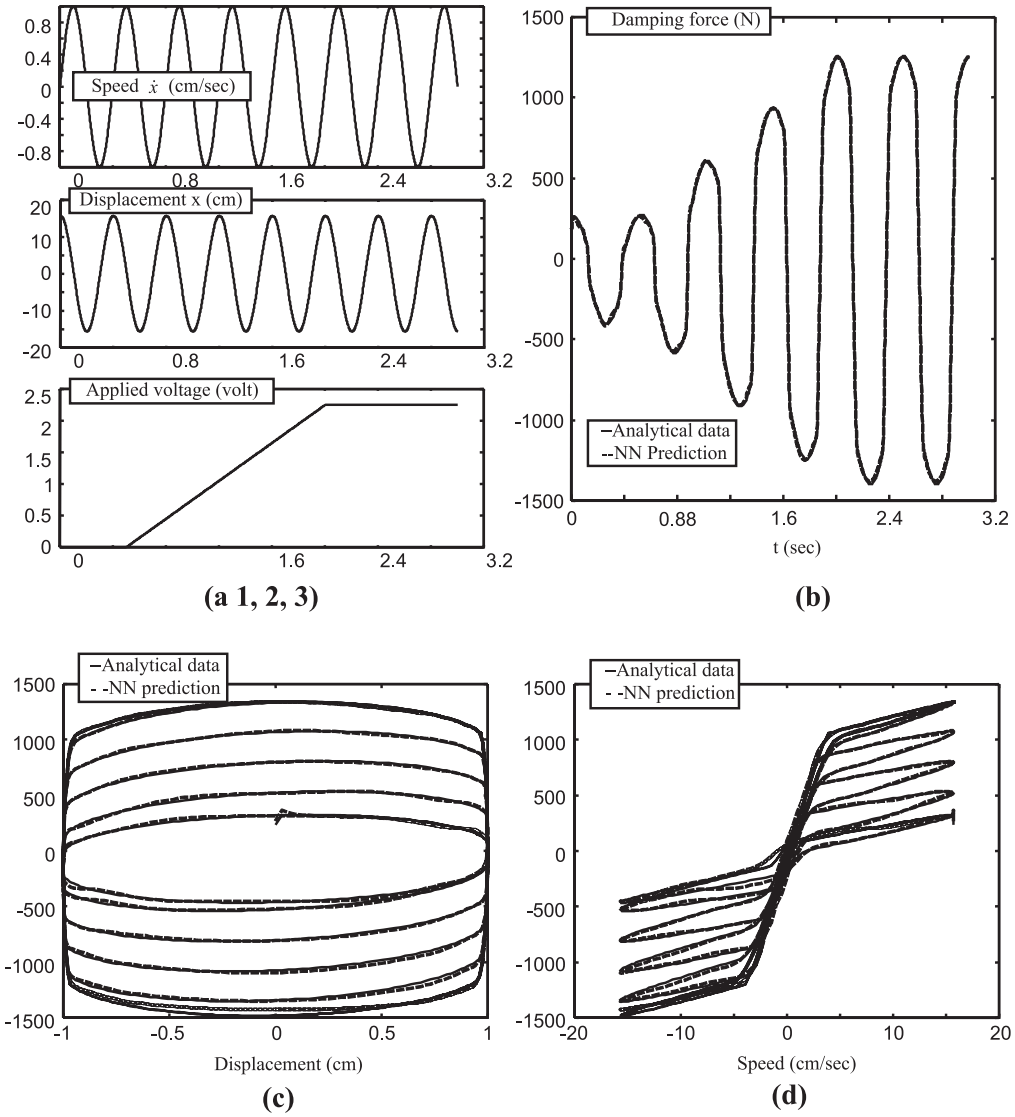

Figure 4. Performance of the conventional network-based dynamic MR damper model: network testing. (a) Operating conditions; (b) Comparison between the network output and analytical prediction: time Domain; (c) Comparison between the network output and analytical prediction: force-displacement relationship; (d) Comparison between the network output and analytical prediction: force-speed relationship.

placement, while Figure 4(d) illustrates the damping force's dependence on the speed of the piston head. As shown by Figures 4(b), (c) and (d), the conventional network-based dynamic MR damper model provides promising results for the tested scenario and correctly captures the system dynamics. It is worth pointing out here that the damping hysteresis phenomenon (Figure 4(d)) is effectively reflected by the ANN model.

The MR damper model presented in the current section, while providing accurate results under this specific testing scenario, does not necessarily guarantee satisfying results for other voltage input profiles with more significant deviations from the training data (as will be shown in the next section). Therefore, it is not sufficient for system design, which demands better input-output scalability that can only be achieved by including the structural information of the physical principles. This is indeed a restatement of the major motivation behind the PMNN-based system modeling.

5. CONTROL-ORIENTED MR DAMPER MODEL BASED ON PMNN – SCALABLE AND INVERTIBLE

5.1. Scalability Validation – Testing with Voltage Profile Significantly Different from the Training Data

Although physical principle flavors have been included in the conventional-network-based MR damper model (Section 4), this approach does not realize the input-output scalability desired for the controller design. Aiming at improving voltage scalability and thus enhancing the model's applicability for controller design, the modulated network concept is utilized to develop a more advanced MR damper model (Figure 5). The same simulated data for developing the conventional network model in Section 4, derived using Spencer's model (1997), is applied for training and testing the newly developed PMNN MR damper model.

As Figure 5 shows, the model consists of three parallel sub-neural networks. The first module represents the dynamic behavior of the MR damper when the applied voltage is zero, i.e., this component is not controllable. The second module is responsible for capturing the linear dependence of the damping force on the applied voltage. Since the correlation between the damping force and the applied voltage is highly nonlinear, a third module, aimed at reflecting the nonlinear relationship between the applied voltage and output damping force, is added. In addition, the speed-dependence of the 2nd and 3rd modules, the controllable components, is also explicitly implemented into the network. One issue worth noting here is that a "cut-off" filter is applied on the control signal (the input voltage) in this PMNN model. That is, when the voltage magnitude becomes higher than the saturation value – 2.25 volts, the input voltage amplitude applied to the network remains at 2.25 volts.

Applying the PMNN formula of equation (10a), the model in Figure 5 can be written as (noticing now the number of modules n_m is 3):

$$
\begin{aligned}
&Y_i(k) = H_i\{\bar{W}^i_{hid\_out}{}^T \bullet \bar{F}_i[\tilde{W}^i_{inp\_hid} \bullet (\bar{x}_i^{sT}, Y(k-1), Y(k-2)]\},\\
&\bar{x}_i^{sT} = [\dot{x}_k \ \dot{x}_k^2 \ \dot{x}_k^3], i = 1, 2, 3\\
&[Y(k) \ \ Y(k-1) \ \ Y(k-2)] = [F_k \ \ F_{k-1} \ \ F_{k-2}],\\
&Y(k) = H_p\{Y_1(k), Y_2(k), Y_3(k), \bar{x}_s{}^T, u(k)\}\\
&= \alpha_1 Y_1(k) + \alpha_2 Y_2(k)\dot{x}_k u(k) + \alpha_3 Y_3(k)\dot{x}_k u(k)^2\\
&\text{where} \quad u(k) = v_k.
\end{aligned}
\tag{22}
$$

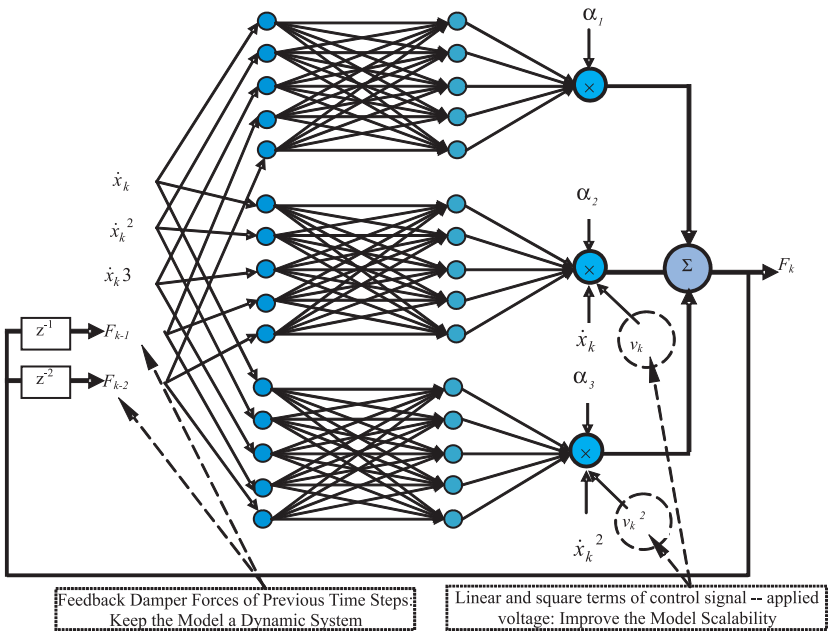

Figure 5. Dynamic MR damper model based on the parallel modulated neural network

Applying more explicit notations, we have:

$$\begin{aligned} F_k &= F[f_k^{NN1}, f_k^{NN2}, f_k^{NN3}] = \alpha_1 f_k^{NN1} + \alpha_2 f_k^{NN2} \dot{x}_k v_k + \alpha_3 f_k^{NN3} \dot{x}_k v_k^2 \\ f_k^{NNi} &= f_i[F_{k-1}, F_{k-2}, \dot{x}_k, \bar{W}_i], \quad i = 1, 2, 3 \end{aligned} \tag{23}$$

where f_k^{NNi} represent the outputs of the ith sub-neural networks, $\dot{x}_k$ is the speed at time k, v_k is the voltage at time k, $\bar{W}_i$ is the weighting vector of the ith sub-network, parameters $\alpha_i (i = 1, 2, 3)$ are determined by trial and error. Similar to the training of the conventional network based-MR damper model, the following chain rule-based correlations are derived to calculate the dynamic gradient information during the network training process:

$$\frac{dF_k}{d\bar{\mathbf{W}}_{PMNN}} = \alpha_1 \frac{df_k^{NN1}}{d\bar{\mathbf{W}}_{PMNN}} + \alpha_2 \frac{df_k^{NN2}}{d\bar{\mathbf{W}}_{PMNN}} \dot{x}_k v_k + \alpha_3 \frac{df_k^{NN3}}{d\bar{\mathbf{W}}_{PMNN}} \dot{x}_k v_k^2 \tag{24a}$$

$$\begin{cases} \dfrac{df_k^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} = \dfrac{\partial f_k^{NNi}}{\partial \bar{\mathbf{W}}_{PMNN}}, & if \quad k = 1 \\ \dfrac{df_k^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} = \dfrac{\partial f_k^{NNi}}{\partial \bar{\mathbf{W}}_{PMNN}} + \dfrac{df_k^{NNi}}{df_{k-1}^{NNi}} \dfrac{df_{k-1}^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} & if \quad k = 2 \\ \dfrac{df_k^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} = \dfrac{\partial f_k^{NNi}}{\partial \bar{\mathbf{W}}_{PMNN}} + \dfrac{df_k^{NNi}}{df_{k-1}^{NNi}} \dfrac{df_{k-1}^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} + \dfrac{df_k^{NNi}}{df_{k-2}^{NNi}} \dfrac{df_{k-2}^{NNi}}{d\bar{\mathbf{W}}_{PMNN}} & if \quad k > 2 \end{cases}$$

$$i = 1, 2, 3 \tag{24b}$$

where

$$\frac{\partial f_k^{NNi}}{\partial \bar{\mathbf{W}}_{PMNN}} = \left[\frac{\partial f_k^{NNi}}{\partial \bar{W}_1} \delta(i,1) \frac{\partial f_k^{NNi}}{\partial \bar{W}_2} \delta(i,2) \frac{\partial f_k^{NNi}}{\partial \bar{W}_3} \delta(i,3) \right]^T \tag{24c}$$

$\bar{W}_1$, $\bar{W}_2$ and $\bar{W}_3$ are the weighting vectors of the first, second and the third sub neural networks (modules), as defined in equation 11; $\delta(i, j)$ equals one if $i = j$, otherwise it is zero. Partial derivatives of each sub-network output with respect to the network weightings are similar as those in equations (18) and (21).

Network training and testing are successfully performed based on the same simulated data applied for developing the conventional network model. In the previous section (Section 4), the MR damper model based on the conventional neural network approach has been successfully tested with a piecewise-linear magnetic profile. To demonstrate the performance improvement, network testing based on a voltage profile with large-magnitude and high frequency sinusoidal perturbation is carried out for the new networks – the PMNN model described in the current section, as well as for the conventional network MR damper model described in the previous section.

Noticing the difference between the training and testing data is much more significant; the network testing based on the sinusoidal-perturbed magnetic field illustrates the improved model scalability concerning the magnetic field. Figures 6(a) and (b) show the applied voltage and speed profiles, respectively. The testing results of the new network approach and the conventional network model, together with the simulated data, are illustrated in Figure 6(c). The analytical simulation result exhibits high frequency oscillation behavior due to the high frequency component of the applied voltage profile. While the prediction of the conventional network model shows a clear mismatch with the analytical result, the PMNN network result closely follows the analytical prediction, i.e., the modulated approach provides a more accurate estimation and demonstrates better voltage scalability.

5.2. PMNN MR Damper Model – An Invertible Model

The testing results clearly indicate the superior control input-output scalability of the proposed PMNN over the conventional hybrid ANN for modeling MR damper. This feature alone qualifies the PMNN MR damper model as a strong candidate for controller design, since it performs well in a much larger operational space than that covered by the training data, a necessity for system design. As mentioned earlier in this paper, PMNN has another outstanding feature attractive for system design: it can be transformed into an inverse model through simple algebraic operations. In other words, instead of two models (Chang and Zhou, 2002; Du et al., 2006), now only one model is needed for the forward mapping as well as the inverse mapping.

Based on equation (23), the desired voltage can be obtained by solving the quadratic equation:

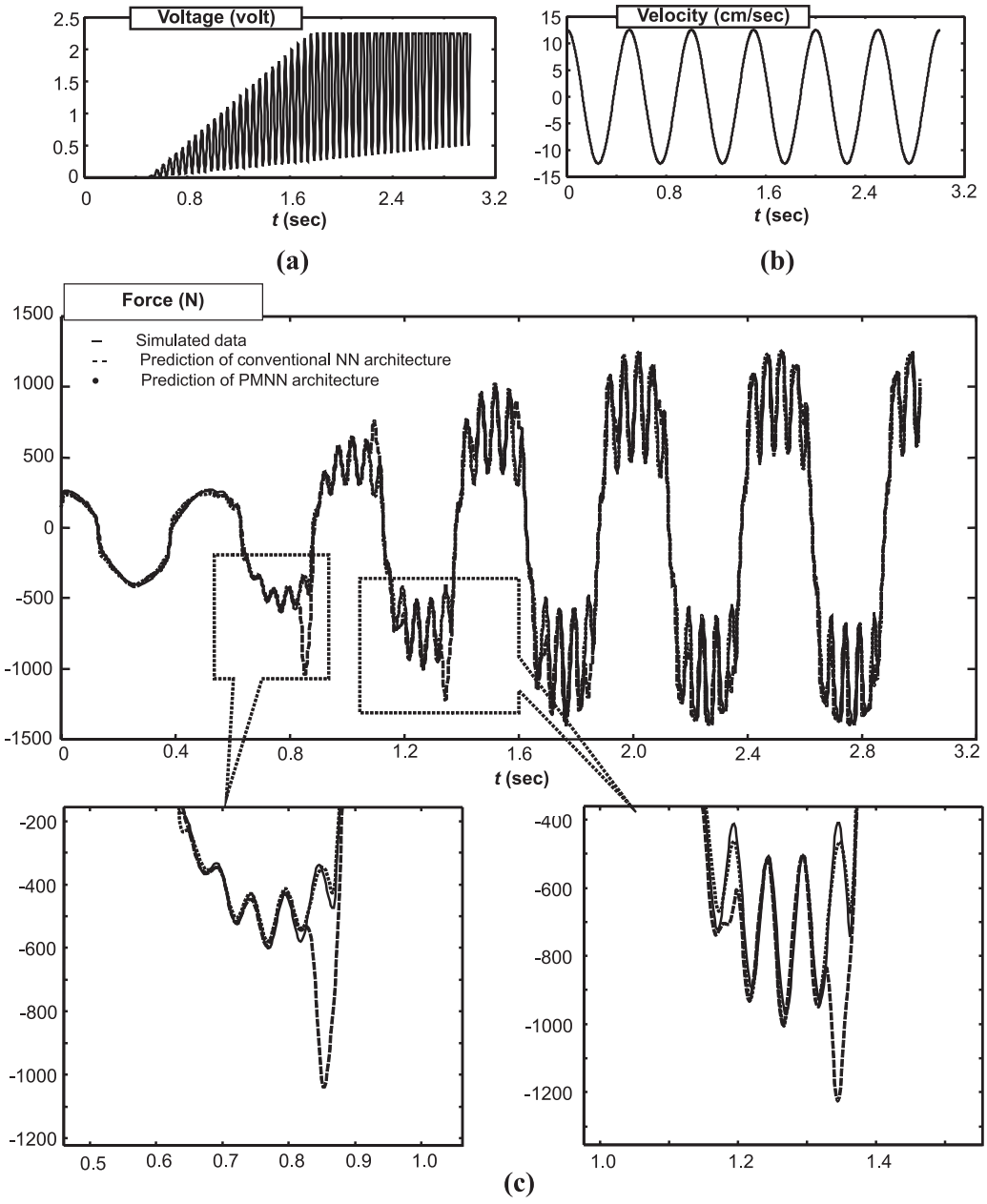

Figure 6. Performance comparison between the parallel modulated and conventional network-based MR damper model: network testing. (a) Applied voltage profile; (b) Speed profile; (c) Testing comparison between the PMNN and conventional ANN-based MR damper model.

$$v_k^d = -\frac{\alpha_2 f_k^{NN2}}{2\alpha_3 f_k^{NN3}} + \frac{\sqrt{(\alpha_2 f_k^{NN2}\dot{x}_k)^2 - 4\alpha_3 f_k^{NN3}\dot{x}_k(\alpha_1 f_k^{NN1} - F_k^d)}}{2\alpha_3 f_k^{NN3}\dot{x}_k}$$

$$if \quad f_k^{NN3} \neq 0, \quad \dot{x}_k \neq 0 \tag{25a}$$

$$v_k^d = \frac{F_k^d - \alpha_1 f_k^{NN1}}{\alpha_2 f_k^{NN2}\dot{x}_k} \quad if \quad f_k^{NN3} = 0, \quad f_k^{NN2} \neq 0, \quad \dot{x}_k \neq 0. \tag{25b}$$

As demonstrated by equation (25), if $f_k^{NN3} = 0$, the model becomes a linear model, and the desired voltage (the control signal) is subsequently a linear function of the desired force. It is obvious that for a solution to exist, the desired force has to be larger than the output of the first sub-network $\alpha_1 f_k^{NN1}$, which reflects the term not directly related with the velocity. Otherwise, the desired force cannot be further modulated.

Under most circumstances, $f_k^{NN3} > 0$. The desired voltage is thus a nonlinear function of the desired force (equation (25a)), and the unique solution exists when the following condition is satisfied:

$$F_k^d > \alpha_1 f_k^{NN1} - \frac{(\alpha_2 f_k^{NN2})^2 \dot{x}_k}{4\alpha_3 f_k^{NN3}}, \quad if \quad \dot{x}_k > 0$$

$$F_k^d < \alpha_1 f_k^{NN1} - \frac{(\alpha_2 f_k^{NN2})^2 \dot{x}_k}{4\alpha_3 f_k^{NN3}}, \quad if \quad \dot{x}_k < 0. \tag{26}$$

As illustrated by equations (25)–(26), the PMNN approach can serve as a forward model as well as an inverse model through realizing a polynomial input-output correlation. Not only does it provide a simple way to back-calculate the open-loop control input based on a desired output value, but it also possesses better input-output scalability. In other words, the model can perform well even when the input space is outside that provided by the training data sets. This is especially beneficial for controller design, the design space of which can inevitably fall outside the training space.

6. CONCLUSION

Two ANN-based MR damper models are developed in this study: a conventional hybrid neural network model enhanced by physical principles and feedback loops realizing system dynamics, and a PMNN-based MR damper model. The first approach (defined as the baseline model in this paper) improves upon the previous ANN-based MR damper modeling efforts through the inclusion of system dynamics. The second one further improves the control-input-output scalability and realizes the invertible model concept, which makes it a control-oriented approach. The network training and testing results clearly indicate that both models provide satisfying performance, while the second one, the PMNN model, outperforms the baseline model by showing superior control-input-output scalability. The candidacy of PMNN as a control-oriented actuator modeling tool is further strengthened by the fact that it is invertible; in other words, the inverse model with desired force as input and the control signal–voltage as output can be easily established by algebraically manipulating the forward model.

This study demonstrates the feasibility of developing scalable and invertible MR damper models, which can be easily applied for system design purposes. It should be pointed out here that like all the other modeling techniques, PMNN is not a "universal" approach; a successful implementation of PMNN demands knowledge about the structure of physical principles.

REFERENCES

Aceves, S. M., Flowers, D. L., Chen, J.-Y., and Babajimopoulos, A., 2006, "Fast prediction of HCCI combustion with an artificial neural network linked to a fluid mechanics code," *Society of Automative Engineers Powertrain and Fluid Systems Conference and Exhibition*, Toronto, Canada, October 16–19.

Cao, M., 2001, "Grey box neural network and its application to system modeling," MS Thesis in Electrical Engineering, Pennsylvania State University.

Cao, M., Wang, K. W., DeVries, L., Fujii, Y., Tobler, W. E., Pietron, G. M., Tibbles, T., and McCallum, J., 2004a, "Steady state hydraulic valve fluid field estimator based on non-dimensional artificial neural network," *ASME Journal of Computing and Information Science in Engineering* **4(3)**, 257–270.

Cao, M., Wang, K. W., Fujii, Y., and Tobler, W. E., 2004b, "A hybrid neural network approach for the development of friction component dynamic model," *ASME Journal of Dynamic Systems, Measurement and Control* **126(1)**, 144–153.

Cao, M., Wang, K. W., Fujii, Y., and Tobler, W. E., 2005, "Development of friction component model for automotive powertrain system analysis and shift controller design based on parallel modulated neural networks," *ASME Journal of Dynamic Systems, Measurement and Control* **127(3)**, 382–405.

Cao, M., Wang, K. W., DeVries, L., Fujii, Y., Tobler, W. E., and Pietron, G. M., 2006, "Experimental investigation and grey-box modeling of spool-type automotive VFS valves with circular flow ports and notches," *ASME Journal of Dynamic Systems, Measurements and Control* **128(3)**, 636–654.

Carlson, J. D., Catanzarite, D. M., and St. Clair, K. A., 1996, "Commercial magneto-rheological devices," in *Proceedings of the 5th International Conference on Electro-Rheological, Magneto-Rheologocal Suspensions and Asscociated Technology* W. A. Bullough (Singapore: World Scientific), pp. 20–28.

Chang, C. C. and Roschke, P., 1998, "Neural network modeling of a magnetorheological damper," *Journal of Intelligent Material Systems and Structures* **9**, 755–764.

Chang, C. C. and Zhou, L., 2002, "Neural network emulation of inverse dynamics for a magnetorheological damper," *Journal of Structural Engineering* **128(2)**, 231–239.

Choi, Y.-T. and Wereley, N. M., 2003, "Vibration control of a landing gear system featuring electrorheological/magnetorheological fluids," *Journal of Aircraft* **40(3)**, 432–439.

Cybenko, G., 1989, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems* **2**, 303–314.

Du, H., Lam, J. and Zhang, N., 2006, "Modeling of a magneto-rheological damper by evolving radial basis function networks," *Engineering Applications of Artificial Intelligence* **19(8)**, 869–881.

Gandhi, F., Wang, K. W., and Xia, L., 2001, "Magnetorheological fluid damper feedback linearization control for helicopter rotor application," *Smart Materials and Structures* **10**, 96–103.

Hiemenz, G. J. and Wereley, N. M., 1999, "Seismic response of civil structures utilizing semi-active MR and ER bracing systems," *Journal of Intelligent Material Systems and Structures* **10(8)**, 646–651.

Hornik, K., Stinchcombe, M., and White, H., 1989, "Multilayer feedforward networks are universal approximators," *Neural Networks* **2**, 359–366.

Kamath, G. M., Hurt, M. K. and Wereley, N. M., 1996, "Analysis and testing of binham plastic behavior in semi-active electrorheological fluid dampers," *Smart Materials and Structures* **5**, 576–590.

Kavlicoglu, B., Gordaninejad, F., Evrensel, C., Fuchs, A. and Korol, G., 2006, "A semi-active, high-torque, magnetorheological fluid limited slip differential clutch," *ASME Journal Vibration and Acoustics* **128(5)**, 604–610.

Koo, J.-H., Ahmadian, M., and Setareh, M., 2006, "Experimental robustness analysis of magneto-rheological tuned vibration absorbers subject to mass off-tuning," *ASME Journal of Vibration and Acoustics* **128(1)**, 126–131.

Lai, C.-Y. and Liao, W. H., 2002, "Vibration control of a suspension system via a magnetorheological fluid damper," *Journal of Vibration and Control* **8(4)**, 527–547.

Marathe, S., Gandhi, F., and Wang, K. W., 1999, "Helicopter blade response and aeromechanical stability with a magnetorheological fluid based lag damper," *Journal of Intelligent Material Systems and Structures* **9,** 272–282.

Peel, D. J., Stanway, R., and Bullough W. A., 1996, "Dynamic modeling of an ER vibration damper for vehicle suspension applications," *Smart Materials and Structures* **5**, 591–606.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge.

Spencer, B. F., Dyke, S. J., Sain, M. K. and Carlson, J. D., 1997, "Phenomenological model for magnetorheological dampers," *Journal of Engineering Mechanics* **123**, 230–238.

Wang, K. W and Cao, M., 2001, *PMNN-based Friction Component Model – Technical Report to Ford Motor Company*, Department of Nuclear and Mechanical Engineering., Pennsylvania State.

Wen, Y. K., 1976, "Method of random vibration of hysteretic systems," *ASCE Journal of the Engineering Mechanics Division* **102(2)**, 249–263.