

A Neural Fuzzy System for Vibration Control in Flexible Structures

Xiaoxu Ji¹, Wilson Wang²

¹Control Engineering, Lakehead University, Thunder Bay, Canada

²Mechanical Engineering, Lakehead University, Thunder Bay, Canada

E-mail: xji@Lakeheadu.ca, wilson.wang@Lakeheadu.ca

Received June 5, 2011, revised June 26, 2011, accepted July 4, 2011

Abstract

An adaptive neural fuzzy (NF) controller is developed in this paper for active vibration suppression in flexible structures. A recurrent identification network (RIN) is developed to adaptively identify system dynamics of the plant. A novel recurrent training (RT) technique is suggested to train the RIN so as to optimize nonlinear input-output mapping and to enhance convergence. The effectiveness of the developed controller and the related techniques has been verified experimentally corresponding to different control scenarios. Test results show that the proposed RIN can effectively recognize the time-varying dynamics of the plant. The RT-based hybrid training technique can improve the adaptive capability of the control system to accommodate different system conditions and enhance the training convergence. The developed NF controller is a robust and stable vibration suppression system, and it outperforms other related NF controllers.

Keywords: Adaptive NF Controller, Active Vibration Control, Recurrent Training Technique, Flexible Structures, Recurrent System Identification

1. Introduction

Vibration suppression is important in many engineering applications such as aerospace systems, robots, buildings, and so on. Vibration suppression can be undertaken either passively or actively. The passive vibration suppression employs some supplementary elements (e.g., dampers and springs) to adjust the characteristics of controlled structures to reduce vibrations [1]. Although the passive suppression is relatively simple in principle, it is difficult to apply in some applications where frequencies are low or extra weights are undesirable such as in aerospace vehicles. Active vibration suppression employs actuating mechanisms to reduce the vibration; it usually provides higher control performance than most passive controls, which is applied in this work to suppress the vibration especially in flexible structures.

The classical controllers such as PID and PD have been widely used for vibration suppression in flexible structures [2,3]. These linear controllers, however, are sensitive to operating conditions; furthermore, it is usually difficult to adjust controller's gains to effectively tackle the overshoot and load disturbance problems simultaneously. Sometimes, it is difficult to derive accu-

rate analytical models in many real engineering systems especially when the plants to be controlled are complex in structure and operate under noisy environments [4,5]. An alternative is the use of intelligent controls based on soft computing schemes such as fuzzy logic (FL) and neural networks (NNs) [6-8]. FL systems, however, lack training capability to adapt themselves under new system conditions, whereas the NN-based reasoning is opaque to users. A solution to solve these problems is to use their synergetic systems such as neural fuzzy (NF) schemes. NF controllers have been utilized in several flexible manipulator applications. For example, a NF paradigm was proposed in [9] for controlling a flexible manipulator with variable payload; the weighting factor of the fuzzy logic was trained by a gradient algorithm. By using NF dynamic-inversion, a discrete-time adaptive tracking method was proposed in [10] for vibration control in a robotic manipulator. However, the parameters in these NF controllers were tuned using the classical training methods, such as gradient algorithm and least squares estimator (LSE); training convergence was limited due to the trapping of local minima.

Although most NF controllers outperform those based on the classical FL and NNs, a typical NF controller is

usually more complex in architecture [11]. This will result in low sampling frequencies and some difficulties in implementation. Furthermore, in some practices, NF controllers may need more control requests than some other classical controllers (e.g., PD and input shaping) in order to achieve the required control performance. To tackle these problems, the objective of this work is to develop a more efficient adaptive NF paradigm for active vibration control especially in the flexible structures. The new aspects of this work include: 1) a novel NF controller with a specific recurrent identification network (RIN) is developed for adaptive vibration suppression; 2) A new recurrent training (RT) technique is suggested to optimize the RIN scheme and to improve the convergence and control performance; and 3) a unique workstation is built for analysis and active vibration suppression for flexible beams.

2. Design of the NF Controller

2.1. Experimental Setup

To facilitate illustration, the vibration suppression workstation developed for this research work is illustrated in **Figure 1**. The tested flexible beam can be with different materials, structures, and orientations; a steel beam with a dimension of $1.5 \times 110 \times 440$ mm is used in this study. Since the objective of this work is to reduce the lateral vibration only, the beam is fixed at one end and placed in a vertical configuration to reduce twisting effects. The vibration in the flexible beam is attenuated by the actuating system equipped at the top of the flexible beam, which consists of an actuating beam and a servo motor. The actuating beam is much stiffer than the flexible beam and is treated as a rigid member. The servo motor drives the actuator through a gear train with a gear ratio of 70:1 to suppress vibration in the flexible beam. The position of the rigid beam θ is measured by an encoder using a 1024 count disc which in quadrature results in 4096 counts/rev. A pair of extra mass blocks is attached to the flexible beam to simulate variable system dynamics of the test beam. The extra mass blocks (about 2×150 grams) take about 20% of the mass of the plant. The disturbance can be provided manually or automatically. In automatic excitation, given a pulse signal, the motor drives the rigid beam, though a gearbox, to generate a disturbance over a specified displacement (e.g., 15 deg) to make the flexible beam in free vibration; then the vibration will be suppressed by the related controllers. The vibration deflection of the flexible beam ε is measured by strain gauges attached close to the fixed end the flexible beam. Based on the relationship between the deflections at the tip of the flexible beam (*i.e.*, ε) and the deformation

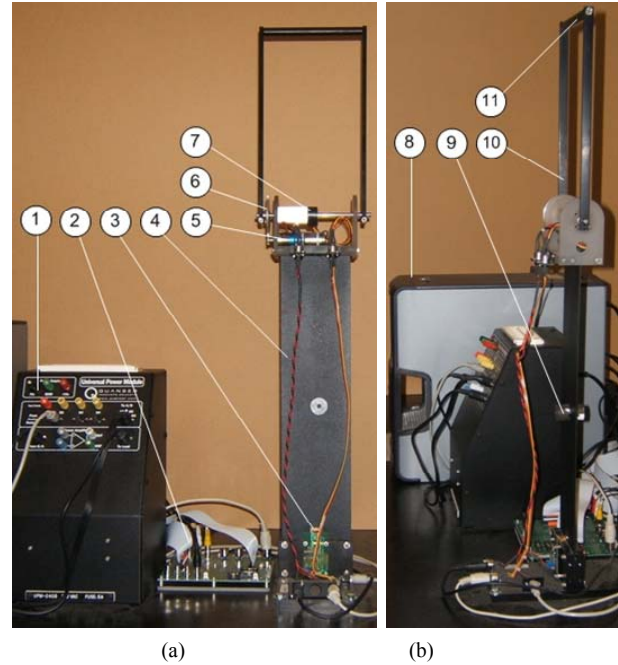


Figure 1. Experimental setup for vibration suppression in flexible beams: (a) the front view; (b) the side view. 1-power supply, 2-DSP board, 3-strain gauges, 4-flexible beam, 5-drive motor, 6-gear train, 7-encoder, 8-computer, 9-extra mass blocks, 10-rigid beam, 11-cross beam.

at the location of measurement (using strain gauges), the strain gauges are calibrated to generate 1 volt per 2.54 cm. The power is supplied by a universal power unit (UPM-2405). The measured signals and control signals are communicated with the computer through a specific DSP board [12].

A simple model of the experimental setup is shown in **Figure 2**, where ε and θ are the deflection of the flexible beam and the rotation angle of the rigid beam, respectively. The related system parameters are listed in **Table 1**.

2.2. The Adaptive Neural Fuzzy Controller

In the developed adaptive NF control, the FL provides a high-level IF-THEN control reasoning framework, whereas the controller parameters are optimized using an appropriate training algorithm. Suppose the control system has n input variables $\{x_1, x_2, \dots, x_n\}$ and one output v , the fuzzy reasoning rules can be represented in the following general form [13],

$$\begin{aligned}
 R_j : & IF(x_1 \text{ is } A_1^g) .AND(x_2 \text{ is } A_2^g) . \\
 & AND \dots AND(x_n \text{ is } A_n^g) \\
 & THEN(v = b_0^j + b_1^j x_1 + \dots + b_n^j x_n)
 \end{aligned} \tag{1}$$

where A_i^g is a membership function (MF), $i = 1, 2, \dots, n$,

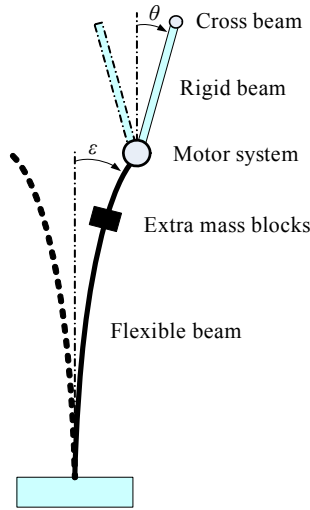


Figure 2. Simplified model of the experimental setup. θ and ϵ are assumed to be positive in the clockwise rotation as shown in the graph.

Table 1. Main parameters of the experimental setup.

Physical parameters	Symbol	Values
Mass of the motor and its fixture	M_e	0.6 kg
Cross beam mass	m_p	0.05 kg
Rigid beam inertia	I	0.0039 kg/m ²
Rigid beam length	L_r	0.285 m
Rigid beam mass	m_b	0.072 kg
Effective stiffness of the flexible beam	K_e	30N/m
Flexible beam length	L_f	0.44 m
Flexible beam mass	m_f	0.22 kg
Motor Torque constant	K_m	0.0767 nm/amp
Motor Armature resistance	R_m	2.6 Ohm

and $g = 1, 2, \dots, G$, G is the number of MFs for each input variable; $j = 1, 2, \dots, m$, and m is the total number of rules, and b^j are constants. In fact, fuzzy reasoning in (1) is a TS1 paradigm [14]. Test results have shown that the TS1 NF control provides more smooth control effects than a TS0 paradigm because it contains more linear consequent parameters than in the TS0 controller. Consequently, the TS1-based NF scheme will be used in this work.

The network architecture of the proposed NF control is schematically shown in Figure 3. It is a four layer network in which each node performs a particular activation function on the incoming signals. The links represent the flow direction of signals between nodes. Unless

specified, the links have unity weights. The input nodes in layer 1 transmit the inputs $\{x_1, x_2, \dots, x_n\}$ to the next layer directly. Each node in layer 2 acts as a membership function (MF). By simulation tests, three MFs are selected for each input variable: sigmoid functions for *small* and *large* MFs, and a Gaussian function for *medium* MF. MFs can be either a single node that performs a simple activation function or multilayer nodes that perform a complex function. The nodes in layer 3 perform the fuzzy T -norm operations. If a product operator is used, the firing strength of rule \mathfrak{R}_j will be

$$\mu_j = \mu_{A_1^g}(x_1) \mu_{A_2^g}(x_2) \cdots \mu_{A_n^g}(x_n) \quad (2)$$

where $\mu(\cdot)$ are the MF grades.

After normalization of the rule firing strengths in layer 4, if a centroid defuzzification method is used, the overall output becomes

$$v = \frac{\sum_{j=1}^m \mu_j (b_0^j + b_1^j x_1 + \cdots + b_n^j x_n)}{\sum_{j=1}^m \mu_j} \quad (3)$$

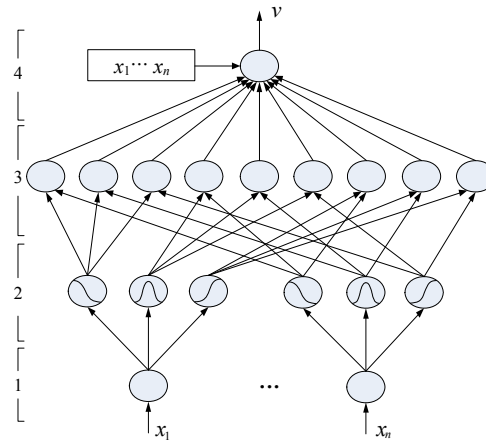


Figure 3. Network architecture of the NF controller.

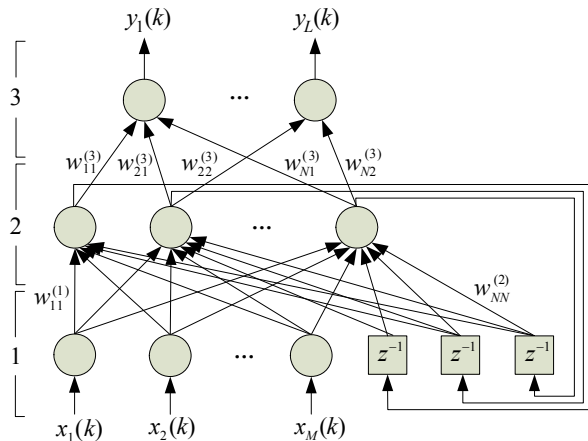


Figure 4. Network architecture of the RIN.

Once the NF control paradigm is established, its parameters should be trained properly so as to achieve optimal control performance. In this case, in each training epoch, the nonlinear system parameters of the MFs are trained by the gradient method in the backward pass, whereas the linear consequent parameters are fine-tuned by the LSE in the forward pass [14].

3. System Identification and Recurrent Training (RT)

3.1. The Recurrent System Identification Network

System identification is the process to recognize the model of the tested plant automatically. One of the advantages of the intelligent control over the classical controls is that system models can be identified by training instead of analytical equations [15]. Real-time system identification is especially important for the plants with time-varying dynamics. In this work, a recurrent identification network (RIN) is developed to adaptively recognize the system dynamics; its network architecture is illustrated in **Figure 4**. It is a 3-layer network in which each node performs a particular activation function on the incoming signals. Layer 1 is the input layer. Layer 2 is the recurrent layer, in which each node has a weighted feedback link to deal with time explicitly as opposed to representing temporal information spatially. Each feedback unit copies the activation output of the corresponding node from the previous time step for context processing; its purpose is to allow the network to memorize cues from the past so as to improve modeling accuracy.

Let $x(k)$ denote the $M \times 1$ external input vector applied to the RIN at the discrete time instant k , $x_i(k)$, $i=1,2,\dots,M$. Assume that there are N recurrent neurons in the hidden layer; Let $y_j^{(2)}(k)$ denote the output of the j th neuron in layer 2, $j=1,2,\dots,N$. Then $y_j^{(2)}(k+1)$ is the forecasted output of the j th neuron generated after one step at time step $k+1$. The signal in each feedback link represents the neuron output in the previous time step (*i.e.*, $k-1$)

$$y_j^{(2)}(k) = \sum_{r=1}^N w_{rj}^{(2)} y_r^{(2)}(k-1) + \sum_{i=1}^M w_{ij}^{(1)} x_i(k) + b_j^{(2)} \quad (4)$$

$w_{ij}^{(1)}$ is the weight of the link from the i th input $x_i(k)$ in layer 1 to the j th neuron in layer 2; $w_{rj}^{(2)}$ is the weight of recurrent link from the r th neuron to the j th neuron in layer 2; $b_j^{(2)}$ is the bias of the j th neuron in layer 2.

The network outputs $y_l(k)$ in layer 3 are computed as the summation of incoming signals:

$$y_l(k) = \sum_{j=1}^N w_{jl}^{(3)} y_j^{(2)}(k) \quad (5)$$

where l is the number of network outputs, $l=1,2,\dots,L$; $w_{jl}^{(3)}$ is the link weight between layer 2 and layer 3, $j=1,2,\dots,N$

3.2. Recurrent Training (RT) Technique

The RIN parameters should be tuned properly to improve the adaptive capability of the controller to accommodate different system conditions. Most of the currently used system identification networks are trained by the classical training methods such as gradient algorithm and LSE [15]. A novel training technique, recurrent training (RT), will be suggested to fine-tune system parameters. Different from other real-time recurrent training, the proposed RT technique can be used to optimize not only recurrent links, but also the general feedforward links in the recurrent NN. It aims to match the outputs of certain neurons in a processing layer to their desired values at specific time instants. The gradients in the RT technique will be recursively computed at each time instant instead of waiting until the end of the presented sequence as with the general gradient-based algorithms. If the desired (target) data sets are $\{\theta_d(k), \varepsilon_d(k)\}$, the objective function $E_o(k)$ at time instant k will be defined as

$$E_o(k) = \sum_k E(k) = \frac{1}{2} \sum_{l=1}^L [y_{ld}(k) - y_l(k)]^2 \quad (6)$$

where $y_{ld}(k)$ and $y_l(k)$ are the l th desired output and the corresponding RIN output from Equation (5), respectively.

To minimize this objective function, we compute the gradients of $\mathbf{w}_{\omega j} = \{w_{\omega j}^{(1)}, w_{\omega j}^{(2)}\}$

$$\nabla_{\mathbf{w}} E_o = \frac{\partial E_o}{\partial \mathbf{w}_{\omega j}} = \sum_k \frac{\partial E(k)}{\partial \mathbf{w}_{\omega j}} = \sum_k \nabla_{\mathbf{w}} E(k) \quad (7)$$

where $\nabla_{\mathbf{w}} E(k)$ is the gradient of the instantaneous error $E(k)$ with respect to the weights $\mathbf{w}_{\omega j}$.

In order to implement the proposed RT technique to train the recurrent networks in real time, we will use an instantaneous estimate of the gradient $\nabla_{\mathbf{w}} E(k)$ at time step k . If $y_j^{(2)}(k)$ is the output of the j th neuron in layer 2, the error-propagation at time instant k will be

$$\begin{aligned} \frac{\partial^+ E(k)}{\partial \mathbf{w}_{\omega j}} &= \frac{\partial E(k)}{\partial y_j^{(2)}(k)} \frac{\partial^+ y_j^{(2)}(k)}{\partial \mathbf{w}_{\omega j}} \frac{\partial^+ y_j^{(2)}(k)}{\partial \mathbf{w}_{\omega j}} \\ &= \frac{\partial y_j^{(2)}(k)}{\partial \mathbf{w}_{\omega j}} + \frac{\partial y_j^{(2)}(k)}{\partial y_j^{(2)}(k-1)} \frac{\partial^+ y_j^{(2)}(k-1)}{\partial \mathbf{w}_{\omega j}} \end{aligned} \quad (8)$$

To minimize $E(k)$, the gradients $\frac{\partial^+ E(k)}{\partial \mathbf{w}_{\omega j}}$ in the

RT technique will be recursively computed at each time instant; there is no need to wait until the end of the presented sequence as in the classical gradient algorithms [16]. The weights \mathbf{w}_{ω_j} is updated by

$$\mathbf{w}_{\omega_j}(k+1) = \mathbf{w}_{\omega_j}(k) - \eta_{\omega_j} \frac{\partial^+ E(k)}{\partial \mathbf{w}_{\omega_j}} \quad (9)$$

where the term $\frac{\partial^+ E(k)}{\partial \mathbf{w}_{\omega_j}}$ is an approximator of the original gradient. The learning rate η_{ω_j} should be selected properly to improve training convergence ($\eta_{\omega_j} = 0.001$ in this case).

3.3. The Hybrid Training Technique

The RIN will be optimized by a hybrid training technique based on the RT and LSE. The purposes of using a hybrid training technique include: 1) a hybrid training process possesses randomness that may aid in escaping local minima [13]; 2) it is necessary for real-time applications, especially for time-varying systems. In training, in the forward pass of each training epoch, the bias parameters, link weights between the inputs and recurrent neurons, as well as the recurrent links are trained by the use of the RT technique; the link weights $w_{j_l}^{(3)}$ between the recurrent neurons in Layer 2 and the output neurons are fine-tuned using LSE method in the backward pass of each training epoch.

4. Control Performance Comparison

4.1. System Implementation

To verify the effectiveness of the developed adaptive NF controller and the related techniques, a series of tests will be conducted with the experimental setup as shown in **Figure 1**. In implementation, two control input variables and one output variable will be utilized in this case. The first input x_1 is selected as the rotating angle error between the desired actuator (rigid) beam position (*i.e.*, 0 deg in this case) and the real position of the actuator beam θ ; in this case, $x_1 = \theta$. The second input x_2 is chosen as the deflection error of the flexible beam whose desired value is 0, that is, $x_2 = \varepsilon$. Both θ and ε are assumed positive in the clockwise rotation as specified by solid lines in **Figure 2**. The control output variable v is selected as the feedback voltage that comes out of the controller and feeds into the drive motor.

For system identification, as illustrated in **Figure 4**, the suggested RIN has three inputs in the input layer: $x(k) = \{\theta_d(k), \varepsilon_d(k), v(k)\}$, where θ and ε are the control input variables (*i.e.*, x_1 and x_2); and v is the control output

variable. By simulation tests, 6 neurons are selected in the recurrent layer, each having a recurrent link. The output layer consists of two output neurons that are to forecast θ and ε (*i.e.*, $L = 2$)

$$\theta(k+1) = y_1(k) = \sum_{j=1}^6 w_{j1}^{(3)} y_j^{(2)}(k) \quad (10)$$

$$\varepsilon(k+1) = y_2(k) = \sum_{j=1}^6 w_{j2}^{(3)} y_j^{(2)}(k) \quad (11)$$

where $w_{j1}^{(3)}$ and $w_{j2}^{(3)}$ ($j=1, 2, \dots, 6$) are the link weights between layer 2 and layer 3.

The disturbance is provided automatically in this test over a specified displacement (e.g., 15 deg). With the consideration of the properties of the DSP board and spectral characteristics of the flexible beam, by tests, a time period of 0.44 sec is selected in this case for the excitation as illustrated in **Figure 5**. The sampling time interval is selected as 0.001 sec.

4.2. Performance of the RT-based Training of the RIN

Firstly, the effectiveness of the proposed RT-based training technique is examined in the RIN. With regards to control requests of the motor, **Figure 6(a)** shows the desired (*i.e.*, real) rotating angle $\theta_d(k)$ of the rigid (actuator) beam and the predicted rotating angle $\theta(k+1)$ by the RIN trained by the proposed RT-LES over five external disturbances; the prediction error $\theta_d(k) - \theta(k+1)$ is shown in **Figure 6(c)**. As a comparison, **Figure 6(b)** illustrates corresponding results as the RIN is trained by the gradient-LES, whereas the prediction error is demonstrated in **Figure 6(d)**. It is clear that the RIN trained by the RT technique can effectively predict the control requests, and outperforms the RIN trained by the gradient method (the prediction errors can be reduced about 50% in the predicted rotating angle of the rigid beam).

On the other hand, **Figures 7(a)** and **(c)** demonstrate the desired (*i.e.*, real) deflection of the flexible beam and the predicted deflection of the flexible beam with

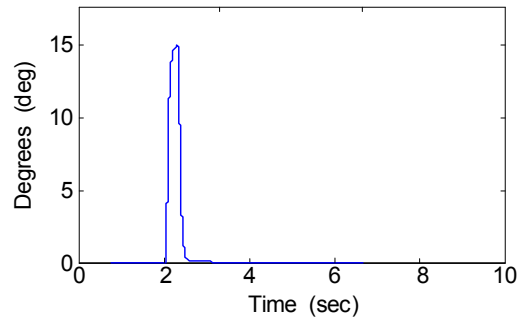


Figure 5. The disturbance signal for control testing.

the RIN trained by the RT-LSE and the gradient-LSE, respectively. The corresponding prediction errors $\varepsilon_d(k) - \varepsilon(k+1)$ are shown in **Figures 7(c) and (d)**, respectively. Apparently, the developed RIN can effectively capture the plant's dynamic behavior. The proposed RT technique can significantly improve the performance of system identification. The prediction error

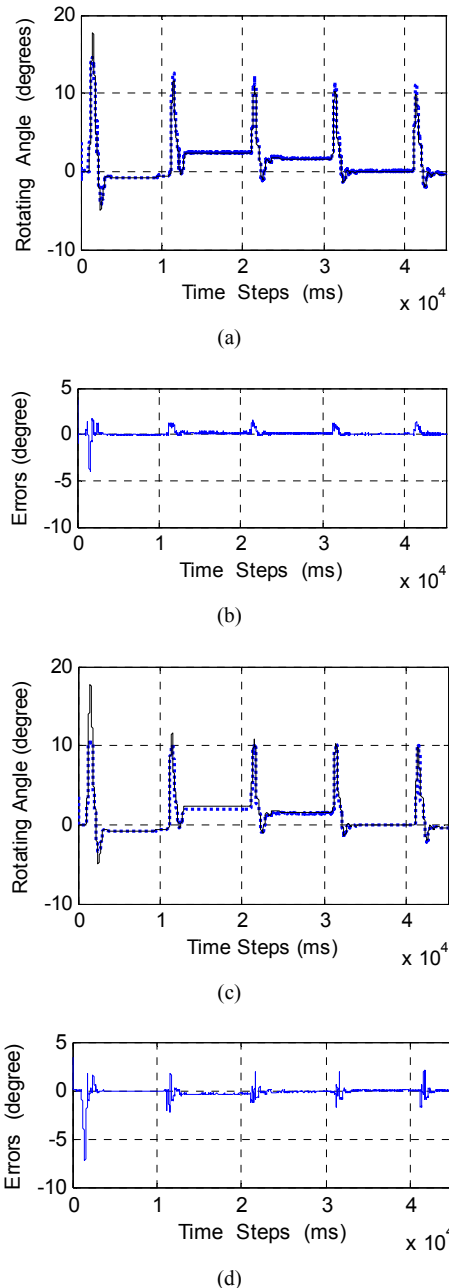


Figure 6. Comparison of the desired (solid lines) and predicted (dotted lines) rotation of the rigid beam (control efforts): (a) RIN is trained by RT-LSE, (b) error of rotating angle; (c) RIN is trained by gradient-LSE; (d) error of rotating angle.

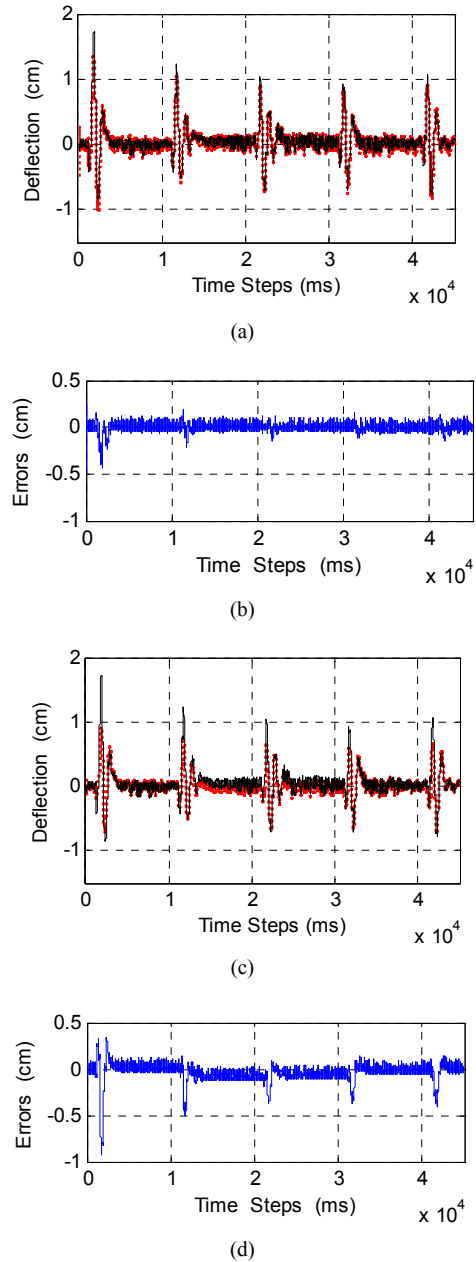


Figure 7. Comparison of the desired (solid lines) and predicted (dotted lines) deflection of the flexible beam: (a) RIN is trained by RT-LSE; (b) Error of deflection; (c) RIN is trained by gradient-LSE; (d) Error of deflection.

of the deflection of the flexible beam can be reduced up to 70% compared with the classical gradient algorithm.

4.3. Control Performance Comparison

To make a comparison, test results from the related NF controllers with different scenarios will be listed:

1) Controller-1: using the same NF scheme as illustrated in **Figure 3** without system identification. Its pur-

pose is to examine the efficiency of the system identification process, especially for time-varying systems.

2) Controller-2: using the same NF control scheme; but the system identification is based on a feedforward NN which has the same number of network parameters and is trained by the gradient algorithm.

3) Controller-3: using the same NF control scheme; but the system identification is performed by the developed RIN which is trained by gradient-LSE method.

4) Controller-4: using the same NF control scheme; but the system identification is performed by the developed RIN which is trained by the suggested RT-LSE technique.

All the related controllers are implemented in MATLAB Simulink. Firstly, we run the tests without extra mass blocks attached. **Figure 8** shows the control performance of the related controllers over 10 seconds. It is clear that the developed NF controller with the proposed RIN and RT-based training (Controller-4) outperforms other related controllers. The only difference between Controller-1 and Controller-2 is related to the system identification. A NF control without system identification takes longer time to converge especially when the beam is very flexible. A controller with efficient system identification can facilitate the recognition of the plant's dynamics so as to improve control outperform. The difference between Controller-2 and Controller-3 is related to the system identification strategy. It is clear that the developed RIN can recognize the dynamics of the plant more effectively and implement it for control operations. The RIN outperforms the feedforward NN in system identification since the recurrent links in the RIN can store context information so as to improve mapping between

the input space and the output space. The training convergence of the NN control, however, is slow due to its application of black box reasoning.

The only difference between Controller-3 and Controller-4 is related to the suggested RT training (*i.e.*, RT-LSE versus gradient-LSE). Compared with the classical gradient algorithm, the RT-related training can significantly enhance training convergence and improve the control performance. The training efficiency of the RT technique is associated with its error-propagation which is calculated at each time step instead of minimizing the overall error function at the end of each sequence as in gradient algorithm.

To verify the robustness of the developed controller against time-varying conditions, a pair of adhesive mass blocks is attached to the flexible beam in three different locations during the tests: a top position about 5 cm below the top end of the flexible beam, a middle position (as shown in **Figure 1**), and a bottom position about 5 cm above the bottom end of the flexible beam. When the mass blocks are attached to the beam, the dynamic properties of the plant (e.g., the mass distribution and natural frequencies) will change, which will result in different responses when a similar disturbance is applied to the flexible beam. **Figures 9-11** show the control performance comparison from different controllers as the extra mass blocks are placed at three different positions of the flexible beam. It is seen that the developed controller (Controller-4) outperforms other controllers among these control scenarios due to effective system identification and training; Controller-4 is robust for time-varying conditions. The proposed RT technique enables the controller to effectively recognize and accommodate the new

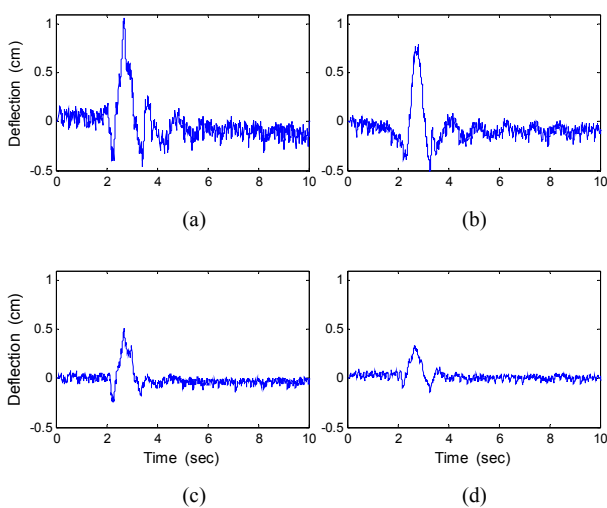


Figure 8. Performance comparison without extra loads, by (a) Controller-1, (b) Controller-2, (c) Controller-3, (d) Controller-4.

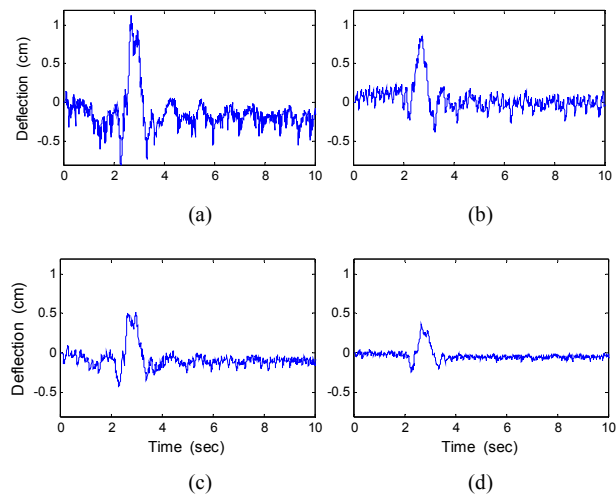


Figure 9. Performance comparison when extra mass blocks are placed at a top position of the flexible beam: by (a) Controller-1, (b) Controller-2, (c) Controller-3, and (d) Controller-4.

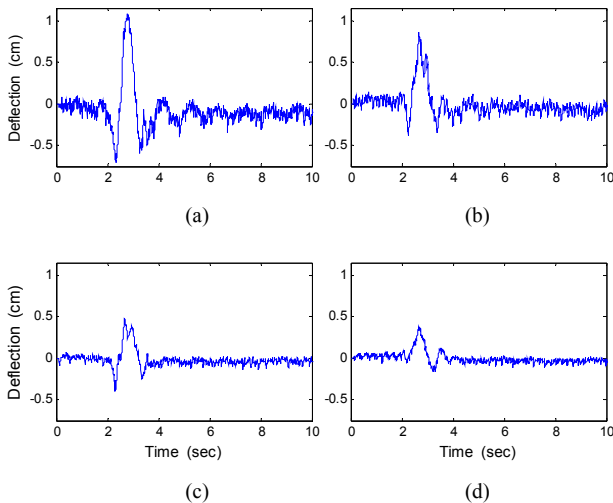


Figure 10. Performance comparison when extra mass blocks are placed in a middle position of the flexible beam: by (a) Controller-1, (b) Controller-2, (c) Controller-3, and (d) Controller-4.

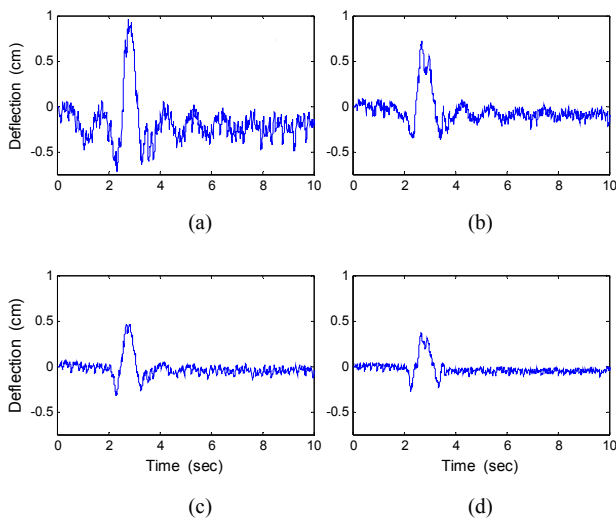


Figure 11. Performance comparison when extra mass blocks are placed at a bottom position of the flexible beam: by (a) Controller-1, (b) Controller-2, (c) Controller-3, and (d) Controller-4.

system dynamics. Furthermore, the hybrid training technique (e.g., RT-LSE) can reduce the sensitivity to the initial conditions of the related parameters and improve training convergence by reducing the possible trapping due to local minima during the training process.

Controller-2 trained by the gradient method performs reasonably well in these tests although it is prone to being trapped by local minima. The main reason is related to the gradient-based searching algorithm.

It is seen that as extra masses are placed at different positions of the flexible beam, there is no apparent con-

trol performance difference in terms of the overshoot, undershoot, and settling time. It means that the NF paradigm is a universal approximation [13] which contains some adaptive capability to accommodate variance of system dynamics. On the other hand, the RIN can more effectively recognize new system dynamic conditions and perform vibration suppression operations.

5. Conclusions

A novel NF controller is developed in this paper for active vibration suppression in flexible structures. A novel recurrent network, RIN, has been adopted to identify system dynamics in real-time. A new RT-based hybrid training technique is suggested to improve the convergence of the RIN scheme. The effectiveness of the developed NF controller and the related techniques has been verified by experimental tests on the developed experimental setup. The comprehensive test investigation has demonstrated that the developed NF controller is an effective strategy for active vibration suppression. The RIN scheme can effectively recognize system dynamic properties for real-time control operations. The RT-based training technique can enhance convergence of training and improve adaptive capability of the controller to accommodate time-varying dynamics of the plants. The developed NF controller is robust and outperforms other related NF paradigms in terms of the control requests, overshoot, undershoot, and settling time.

6. References

- [1] D. Younesian, E. Esmailzadeh and R. Sedaghati, "Passive Vibration Control of Beams Subjected to Random Excitations with Peaked PSD," *Journal of Vibration and Control*, Vol. 12, No. 9, 2006, pp. 941-953. [doi:10.1177/1077546306068060](https://doi.org/10.1177/1077546306068060)
- [2] G. Genta, "Vibration Dynamics and Control," Springer, New York, 2009. [doi:10.1007/978-0-387-79580-5](https://doi.org/10.1007/978-0-387-79580-5)
- [3] Z. Mohamed, J. Martins, M. Tokhi, J. Costa and M. Botto, "Vibration Control of a Very Flexible Manipulator System," *Control Engineering Practice*, Vol. 13, No. 3, 2005, pp. 267-277. [doi:10.1007/978-0-387-79580-5](https://doi.org/10.1007/978-0-387-79580-5)
- [4] K. Melhem and W. Wang, "Global Output Tracking Control of Flexible Joint Robots via Factorization of the Mass Matrix," *IEEE Transactions on Robotics*, Vol. 25, No. 2, 2009, pp. 248-257. [doi:10.1007/978-0-387-79580-5](https://doi.org/10.1007/978-0-387-79580-5)
- [5] M. Ishikawa, Y. Minami and T. Sugie, "Development and Control Experiment of the Trident Snake Robot," *IEEE/ASME Transactions on Mechatronics*, Vol. 15, No. 1, 2010, pp. 9-16. [doi:10.1109/TMECH.2008.2011985](https://doi.org/10.1109/TMECH.2008.2011985)
- [6] R. Jamil, "Inverse Dynamics Based Tuning of a Fuzzy Logic Controller for a Single-Link Flexible Manipulator," *Journal of Vibration and Control*, Vol. 13, No. 12, 2007, pp. 1741-1759. [doi:10.1177/1077546307076282](https://doi.org/10.1177/1077546307076282)

- [7] T. Li and S. Tsai, "T-S Fuzzy Bilinear Model and Fuzzy Controller Design for a Class of Nonlinear Systems," *IEEE Transactions on Fuzzy Systems*, Vol. 15, No. 3, 2007, pp. 494-506. [doi:10.1109/TFUZZ.2006.889964](https://doi.org/10.1109/TFUZZ.2006.889964)
- [8] C. Ting, "An Observer-Based Approach to Controlling Time-Delay Chaotic Systems via Takagi-Sugeno Fuzzy Model," *Information Sciences*, Vol. 177, No. 20, 2007, pp. 4314-4328. [doi:10.1109/TFUZZ.2006.889964](https://doi.org/10.1109/TFUZZ.2006.889964)
- [9] L. Tian and C. Collins, "A Dynamic Recurrent Neural Network-Based Controller for a Rigid-Flexible Manipulator System," *Mechatronics*, Vol. 14, No. 5, 2004, pp. 471-490. [doi:10.1109/TFUZZ.2006.889964](https://doi.org/10.1109/TFUZZ.2006.889964)
- [10] M. Caswara and H. Unbehauen, "A Neurofuzzy Approach to the Control of a Flexible-Link Manipulator," *IEEE Transaction on Robotics and Automation*, Vol. 18, No. 6, 2002, pp. 932-944. [doi:10.1109/TRA.2002.805660](https://doi.org/10.1109/TRA.2002.805660)
- [11] L. Tian and C. Collins, "Adaptive Neuro-Fuzzy Control of a Flexible Manipulator," *Mechatronics*, Vol. 15, No. 10, 2005, pp. 1305-1320. [doi:10.1016/j.mechatronics.2005.02.001](https://doi.org/10.1016/j.mechatronics.2005.02.001)
- [12] W. Wang and O. Jianu, "A Smart Monitor for Machinery Fault Detection," *IEEE/ASME Transactions on Mechatronics*, Vol. 15, No. 1, 2010, pp. 70-78. [doi:10.1109/TMECH.2009.2016956](https://doi.org/10.1109/TMECH.2009.2016956)
- [13] W. Wang, "An Intelligent System for Machinery Condition Monitoring," *IEEE Transactions on Fuzzy Systems*, Vol. 16, No. 1, 2008, pp. 110-122. [doi:10.1109/TFUZZ.2007.896237](https://doi.org/10.1109/TFUZZ.2007.896237)
- [14] Z. Sun, K. Au and T. Choi, "A Neuro-Fuzzy Inference System through Integration of Fuzzy Logic and Extreme Learning Machines," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 37, No. 5, 2007, pp. 1321-1331. [doi:10.1109/TSMCB.2007.901375](https://doi.org/10.1109/TSMCB.2007.901375)
- [15] P. Mastorocostas and J. Theocharis, "A Recurrent Fuzzy-Neural Model for Dynamic System Identification," *IEEE Transactions on System, Man and Cybernetics-Part B: Cybernetics*, Vol. 32, No. 2, 2002, pp. 176-190.
- [16] K. Christodoulou and C. Papadimitriou, "Structural Identification Based on Optimally Weighted Modal Residuals," *Mechanical System and Signal Processing*, Vol. 21, No. 1, 2007, pp. 4-23. [doi:10.1016/j.ymsp.2006.05.011](https://doi.org/10.1016/j.ymsp.2006.05.011)