Journal of Solid Mechanics Vol. 10, No. 2 (2018) pp. 387-399

Vibration Suppression of Simply Supported Beam under a Moving Mass using On-Line Neural Network Controller

S. Rezaei [1, *], M. Pourseifi [2]

[1]University of Applied Science and Technology, Center of Mammut, Tehran, Iran
[2]Faculty of Engineering, The University of Imam Ali, Tehran, Iran

Received 12 March 2018; accepted 11 May 2018

ABSTRACT

In this paper, model reference neural network structure is used as a controller for vibration suppression of the Euler–Bernoulli beam under the excitation of moving mass travelling along a vibrating path. The non-dimensional equation of motion the beam acted upon by a moving mass is achieved. A Dirac-delta function is used to describe the position of the moving mass along the beam and its inertial effects. Analytical solution the equation of motion is presented for simply supported boundary condition. The hybrid controller of system includes of a controller network and an identifier network. The neural networks are multilayer feed forward and trained simultaneously. The performance and robustness of the proposed controller are evaluated for various values mass ratio of the moving mass to the beam and dimensionless velocity of a moving mass on the time history of deflection. The simulations verify effectiveness and robustness of controller.

Keywords: Vibration control; Neural network controller; Euler–Bernoulli beam theory; Moving mass.

1 INTRODUCTION

DYNAMIC and vibration control of an elastic beam structure under the influence of a moving mass or load is a vital engineering application. Kononov and de Borst [1] investigated the dynamic stability of beams under the influence of inertial forces and high velocities of moving loads. Fryba [2] offered solutions to problems of vibration of beams and plates subjected to moving loads. Bridges and railway bridges are characteristic structures that carry moving masses and loads. Bilello et al. [3] experimentally investigated a small-scale bridge model under a moving mass and demonstrated that there is excellent agreement between analytical and experimental results without significant inertial influence at different mass velocities. Sung [4] investigated the dynamic modeling and piezoelectric control of a simply supported beam subjected to a moving mass. Nikkhoo et al. [5] studied the dynamic behavior and modal control of beams under moving mass. Prabakar et al. [6] studied optimal semi-active control of a half-car model using a magneto-rheological damper. Pisarski and Bajer [7] explored semi-active control of a string supported by a viscous damper subjected to a moving load. Ryu and Kong [8] investigated the dynamic response and active vibration control of beams subjected to a traveling mass. Good agreement was observed between the experimental results of the dynamic deflection of a beam traversed by a moving mass and their

*Corresponding author. Tel.: +98 09190212306.
E-mail address: *sara_r759@yahoo.com* (S.Rezaei).

simulation results. Recent research on neural networks have found them useful for identification, signal processing, and processing control applications. The use of neural networks for vibration control has been studied by Flanders et al. [9], who used a RBF network to minimize beam vibration. Chen et al. [10] applied neural networks as state estimators when using a modified independent modal space control algorithm for vibration control of a cantilever beam. Smyser and Chandrashekhara [11] used neural networks to imitate an LQG/LTR controller for vibration control of composite beams. Valoor et al. [12] applied recurrent neural networks for vibration control of smart composite beams. Qiu et al. [13] used a BP neural network controller for vibration suppression of a flexible piezoelectric beam. In order to robustness performance many researches use recurrent neural network (RNN) to the identification and control of nonlinear dynamic systems [14-19].

In this paper model reference controller based neural network structure is considered for vibration suppression of the Euler–Bernoulli beam under the excitation of moving mass. In the proposed controller, a feed forward identifier network is used to provide the sensitivity information. This information is then used to train the controller network. The initial weights of identifier network are trained offline using data generated by process model. Then, the coupled neural networks were used for control and identification plant. These networks are trained simultaneously and the updating of the neural networks weights continue during the operation of the system. The controller network is feed forward neural network which uses of the sensitivity information of the plant provided by identifier network in order to update weights and produce the proper control signal to drive plant output to desired output. The performance of the proposed neural network controller is verified for various values mass ratio of the moving mass to the beam and dimensionless velocity of a moving mass on the time history of deflection. The neural network controller with two actuators shows excellent performance in suppressing the dynamic deflection of the beam under different constant velocities and various moving masses.

2 DYNAMIC MODELING

The mathematical model of a simply supported Euler-Bernoulli beam subjected to a moving mass is considered. It is assumed that the mass travels in a straight line in the horizontal direction, that this movement is known, and that the beam only vibrates in the y direction. Let $w(x,t)$ denote the transverse displacement of the beam and x and y denote the axial and the transverse coordinates, respectively. The initial conditions for an uncontrolled dynamic beam are $w(x,0)=f_1(x)$ and $\partial w(x,0)/\partial t = f_2(x)$, in which $f_1(x)$ and $f_2(x)$ are continuous functions. For the Euler-Bernoulli beam model under a moving mass of weight mg and velocity v, suppose that p actuators are used in the system at positions $x_i, i=1,2\dots p$ to apply control force $u_i(t)$ as shown in Fig. 1. The equation of transverse motion of the controlled system can be written as:

$$\rho_b A_b \frac{\partial^2 w(x,t)}{\partial t^2} + E_b I_b \frac{\partial^4 w(x,t)}{\partial x^4} = [mg + m\frac{d^2\xi(t)}{dt^2}]\delta(x-vt) + \sum_{i=1}^{p} u_i(t)\delta(x-x_i) \tag{1}$$

where ρ_b the density of the beam, A_b the cross-sectional area, E_b the Young's modulus of elasticity, I_b the moment of inertia and $\delta(\cdot)$ the Dirac delta function. Also, $\xi(t)$ denotes the vertical displacement of the moving mass that acts upon a moving coordinate, which is denoted as vt in this case. As a result

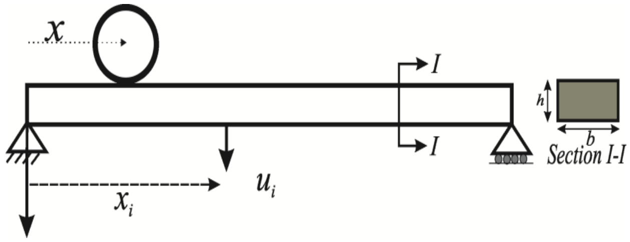


Fig.1
A mathematical model of a simply-supported beam subjected to a moving mass.

$$\xi(t) = w(vt,t), \frac{d\xi}{dt} = \frac{\partial w}{\partial t} + v\frac{\partial w}{\partial x}, \frac{d^2\xi(t)}{dt^2} = \frac{\partial^2 w}{\partial t^2} + 2v\frac{\partial w}{\partial t}\frac{\partial w}{\partial x} + v^2\frac{\partial^2 w}{\partial x^2} \tag{2}$$

Using Eqs. (1) and (2), the governing equation of the system can be expressed as:

$$\rho_b A_b \frac{\partial^2 w}{\partial t^2} + E_b I_b \frac{\partial^4 w}{\partial x^4} + m(\frac{\partial^2 w}{\partial t^2} + 2v\frac{\partial w}{\partial t}\frac{\partial w}{\partial x} + v^2\frac{\partial^2 w}{\partial x^2})\delta(x-vt) = mg\ \delta(x-vt) + \sum_{i=1}^{p} u_i(t)\delta(x-x_i) \tag{3}$$

Employing the assumed mode method, the displacement of a beam, $w(x,t)$, can be expressed in modal expansion as:

$$w(x,t) = \sum_{n=1}^{\infty} \phi_n(x) q_n(t) \tag{4}$$

where $q_n(\mathrm{t})$ are the unknown time-dependent generalized coordinates and $\phi_n(x)$ is the n^{th} mode shape for the undamped beam. For the simply supported beam, the assumed mode shape may be written as:

$$\phi_n(x) = \sin(\frac{n\pi x}{L}) \tag{5}$$

Substituting Eq. (5) into Eq. (4) and then substituting the resulting equation into Eq. (3) and multiplying both sides by $\phi_j(x)$ yields

$$\begin{aligned} &\sum_{n=1}^{\infty} \frac{d^2 q_n}{dt^2} \int_0^L \rho_b A_b \phi_n \phi_j dx + \sum_{n=1}^{\infty} q_n \int_0^L E_b I_b \frac{d^4 \phi_n}{dx^4}\phi_j dx + m[\sum_{n=1}^{\infty} (\frac{d^2 q_n}{dt^2}\int_0^L \phi_n\phi_j dx + 2v\frac{dq_n}{dt}\int_0^L \frac{d\phi_n}{dx}\phi_j dx \\ &+ v^2 q_n \int_0^L \frac{d^2\phi_n}{dx^2}\phi_j dx)]\delta(x-vt) = mg\int_0^L \phi_j \delta(x-vt)dx + \sum_{i=1}^{p}\int_0^L \phi_j u_i(t)\delta(x-x_i)dx \end{aligned} \tag{6}$$

Substituting the derivative of Eq. (5) into Eq. (6) produces

$$\begin{aligned} &\sum_{n=1}^{\infty} \frac{d^2 q_n}{dt^2}\int_0^L \rho_b A_b \sin(\frac{n\pi x}{L})\sin(\frac{j\pi x}{L})dx + \sum_{n=1}^{\infty} q_n \int_0^L E_b I_b (\frac{n\pi}{L})^4 \sin(\frac{n\pi x}{L})\sin(\frac{j\pi x}{L})dx \\ &+ m\{\sum_{n=1}^{\infty}[\frac{d^2 q_n}{dt^2}\int_0^L \sin(\frac{n\pi x}{L})\sin(\frac{j\pi x}{L})dx + 2v\frac{dq_n}{dt}\int_0^L (\frac{n\pi}{L})\cos(\frac{n\pi x}{L})\sin(\frac{j\pi x}{L})dx \\ &- v^2 q_n \int_0^L (\frac{n\pi}{L})^2 \sin(\frac{n\pi x}{L})\sin(\frac{j\pi x}{L})dx]\}\delta(x-vt) = mg\int_0^L \sin(\frac{j\pi x}{L})\delta(x-vt)dx + \sum_{i=1}^{p}\int_0^L \sin(\frac{j\pi x}{L})u_i(t)\delta(x-x_i)dx \end{aligned} \tag{7}$$

According to the following condition of orthogonality

$$\int_0^L \phi_n \phi_j\, dx = \begin{cases} \dfrac{L}{2} & n = j \\ 0 & n \neq j \end{cases} \tag{8}$$

Based on the Dirac-delta function for the load terms

$$\int_{x_1}^{x_2} g(x)\delta(x-x_0)\,dx=\begin{cases} g(x_0) & x_1<x_0<x_2 \\ 0 & otherwise \end{cases} \tag{9}$$

Eqs. (7), (8) and (9) can be used to produce the differential equation of motion as:

$$\begin{aligned}&\frac{d^2q_n}{dt^2}+\frac{2m}{\rho_bA_bL}\sum_{j=1}^{\infty}\sin(\frac{n\pi v}{L}t)\sin(\frac{j\pi v}{L}t)\frac{d^2q_j}{dt^2}+\frac{4m\pi v}{\rho_bA_bL^2}\sum_{j=1}^{\infty}j\sin(\frac{n\pi v}{L}t)\cos(\frac{j\pi v}{L}t)\frac{dq_j}{dt}\\&+\omega_n^2q_n-\frac{2m}{\rho_bA_bL}\sum_{j=1}^{\infty}(\frac{j\pi v}{L})^2\sin(\frac{n\pi v}{L}t)\sin(\frac{j\pi v}{L}t)q_j=\frac{2mg}{\rho_bA_bL}\sin(\frac{n\pi v}{L}t)\\&+\frac{2}{\rho_bA_bL}\sum_{i=1}^{p}u_i(t)\sin(\frac{n\pi x_i}{L})dx\qquad n=1,2,\ldots\end{aligned} \tag{10}$$

where ω_n is the fundamental frequency of the simply supported beam as:

$$\omega_n=\sqrt{\frac{E_bI_b}{\rho_bA_b}}(\frac{n\pi}{L})^2 \tag{11}$$

By introducing the following dimensionless parameters and variables

$$\Phi_n=\frac{q_n}{\mu},\eta_i=\frac{x_i}{L_b},v_0=\frac{v}{v_{cr}}=\frac{\pi v}{\omega_1L_b},\tau=\frac{\pi}{L_b^2}\sqrt{\frac{E_bI_b}{\rho_bA_b}}t,\mu=\frac{4\rho_bA_bgL_b^4}{E_bI_b\pi^5},\Upsilon=\frac{m}{\rho_bA_bL_b},U=\frac{\pi^3u}{2\rho_bA_bLg} \tag{12}$$

Eq. (10) can be expressed in dimensionless form as:

$$\begin{aligned}&\frac{d^2\Phi_n}{d\tau^2}+2\Upsilon\sum_{j=1}^{\infty}\sin(n\pi v_0\tau)\sin(j\pi v_0\tau)\frac{d^2\Phi_j}{d\tau^2}+4\Upsilon\pi v_0\sum_{j=1}^{\infty}j\sin(n\pi v_0\tau)\cos(j\pi v_0\tau)\frac{d\Phi_j}{d\tau}+n^4\pi^2\Phi_n\\&-2\Upsilon\sum_{j=1}^{\infty}(j\pi v_0)^2\sin(n\pi v_0\tau)\sin(j\pi v_0\tau)\Phi_j=\frac{\Upsilon\pi^3}{2}\sin(n\pi v_0\tau)+\sum_{i=1}^{p}U_i(\tau)\sin(n\pi\eta_i)dx\qquad n=1,2,\ldots\end{aligned} \tag{13}$$

Considering the first r vibrational modes of the beam, Eq. (13) is a set of r coupled ordinary differential equations that can be expressed in matrix form as:

$$[M(\tau)]\{\ddot{\Phi}_n(\tau)\}+[C(\tau)]\{\dot{\Phi}_n(\tau)\}+[K(\tau)]\{\Phi_n(\tau)\}=\{f(\tau)\}+[D(\tau)]\{U(\tau)\} \tag{14}$$

In which $M(\tau),C(\tau)$ and $K(\tau)$ are the $r\times r$ matrixes with their elements defined as:

$$M_{nj}=\delta_{nj}+2\Upsilon\sin(n\pi v_0\tau)\sin(j\pi v_0\tau)\qquad n,j=1,2,\ldots r \tag{15}$$

$$C_{nj}=4\Upsilon\pi v_0j\sin(n\pi v_0\tau)\cos(j\pi v_0\tau) \tag{16}$$

$$K_{nj}=n^4\pi^2\delta_{nj}-2\Upsilon(j\pi v_0)^2\sin(n\pi v_0\tau)\sin(j\pi v_0\tau) \tag{17}$$

The elements of matrix $f_{r\times 1}$ and $D_{r\times p}$ are

$$f_j = \begin{cases} \dfrac{\Upsilon\pi^3}{2}\sin\left(j\pi v_0\tau\right) & for \quad 0 \le \tau \le \tau_f \\ 0 & for \quad \tau > \tau_f \end{cases} \tag{18}$$

$$D_{ji} = \sin\left(j\pi\eta_i\right) \qquad i = 1,2,\ldots,p \tag{19}$$

And $\tau_{\mathrm{f}} = 1/vL_b\sqrt{E_bI_b/\rho_bA_b}$ is the time at which the moving mass reaches the end of the beam. By defining the state vector as $X = \begin{bmatrix}\Phi_n & \dot{\Phi}_n\end{bmatrix}^T_{1\times 2r}$, the state-space form of this equation becomes

$$\dot{X}(\tau) = A(\tau)X(\tau) + B(\tau)U(\tau) + F(\tau) \tag{20}$$

where

$$A(\tau) = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}_{2r\times 2r}$$

$$B(\tau) = \begin{bmatrix} 0 \\ -M^{-1}D \end{bmatrix}_{2r\times p}$$

$$F(\tau) = \begin{bmatrix} 0 \\ M^{-1}f \end{bmatrix}_{2r\times 1}$$

3 NEURAL NETWORK CONTROLLER DESIGN

In this section, we use neural model reference controller to control the beam vibration. The neural model reference controller consists of two neural networks: controller network and identifier network, as shown in Fig.2. Identifier network is trained to model the dynamic behavior of the process to be controlled. The controller network makes use of the plant model network and is trained to produce the proper control signal in order that plant output follows the desired output. The inputs to the identifier network are control forces $u(t)$ .The Identifier output is prediction of the process output $\hat{y}(t)$. The input to controller network is process output $y(t)$. Its outputs are control forces.

3.1 Identifier network

In the practical applications, the real plant dynamics may not be available for the controller design. Thus, the neural network process model is used to provide the needed information about the plant. This information is then used to train the controller network. Now, In order to enhance the networks (controller and identifier) convergence rate, the initial weights of identifier network are trained off line. The most common method of offline identification is called forward modeling. During training, the process (beam) and neural network model receive the same input, the outputs from neural network model and process are compared, and this error signal is used to update the weights of the network so that the output of the network will be similar to the process. In order to, generate some experimental input/output data from the process; we use proper inputs to the plant. In the case of the beam system, this would be the input forces and the output deflection. This network uses the back-propagation learning rule to update the weights.

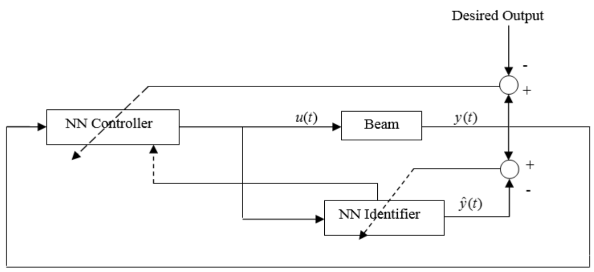


Fig.2
Block diagram of the neural network controller.

The following is brief outline of back propagation theory. Training a network by back propagation involves three stage: the feed forward of input training pattern, the back propagation of associated error and the adjustment of the weights [20]. The neural network architecture is given in the Fig.3. The neural network weights are initialized to small random value to start the training. In a general feed-forward network, neuron *j* computes a weighted sum of its inputs of the form

$$net_j = \sum_{i=1}^{N} w_{ji} x_i \tag{21}$$

where x_i is the *i th* neuron in previous layer, that sends a connection to neuron *j*, w_{ji} is the weight associated with that connection and *N* is total number of neurons in previous layer. The sum in (21) is transformed by a nonlinear activation function $f(.)$ to give the output of *j th* neuron in the form

$$z_j = f_j\left(net_j\right) \tag{22}$$

The output layer of identifier uses a linear activation function and the hidden layer neurons use a log- sigmoid function defined as:

$$z_j = \frac{1}{1+e^{-net_j}} \tag{23}$$

The error signal at the output of neuron o is defined by

$$e_o = y_o - y_{od} \tag{24}$$

where neuron *o* is an output node, y_o is output and y_{od} is the desired output for neuron *o*. The squared error of all neurons in the output layer is defined as:

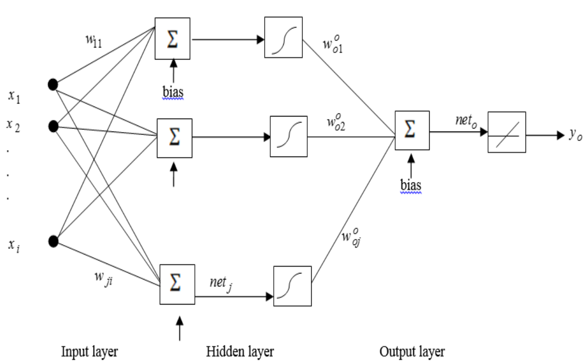


Fig.3
Neural network architecture.

$$E(w)=\frac{1}{2}\sum_{o=1}^{M_0}(y_o-y_{od})^2 \tag{25}$$

where M_0 is the number of neurons in output layer. Training of this network is realized by adapting the network weights such that this error function is minimized. We now introduce a useful notation

$$\delta_j=-\frac{\partial E}{\partial net_j} \tag{26}$$

where δ_j is error signal of j *th* neuron. According to the Steepest Descent Method, the successive adjustments applied to the weight vector are in direction of steep descent, that is in a direction opposite to the gradient vector $\nabla E(w)$ Thus

$$w(k+1)=w(k)-\eta\nabla E(w) \tag{27}$$

The weights of the o *th* neuron in output layer for neural network with M_0 output neuron and N_h hidden layer neuron is updated as follow

$$w_{oj}^{o}(k+1)=w_{oj}^{o}(k)+\eta\delta_o z_j \tag{28}$$

where $o=1,\ldots,M_0, j=1,\ldots,N_h,\eta$ is learning rate, δ_o is error signal of o *th* neuron of output layer and z_j is output of j *th* neuron of hidden layer. after the δ_o for all the output neuron was Evaluated, the δ_o is Backpropagated in order to obtain δ_j for each hidden unit in the network.thus the weight for j *th* neuron of hidden layer with N_h neuron and N_i input is adjusted as following

$$w_{ji}(k+1)=w_{ji}(k)+\eta f_j'(net_j)x_i\sum_{o=1}^{M_0}\delta_o w_{oj} \tag{29}$$

where $j=1,\ldots,N_h, i=1,\ldots,N_i$. The weights are iteratively updated by this method until the error function (Eq.(25)) tends to zero.

3.2 Controller network

The controller network is trained to produce the proper control signal in order that minimize the difference between the reference model and the actual system output. Fig.3 shows a MLP network which used in controller network. The neural network weights are initialized to small random value to start the training. The output error for the controller network at iteration *k* is the difference between the plant output and the desired output, as following [21]

$$e_c(k)=y(k)-r_d(k) \tag{30}$$

Training this network is obtained by adjusting the network weights such that the following loss function is minimized

$$E_c(W_c)=\frac{1}{2}e_c^T e_c \tag{31}$$

According to the Steepest Descent Method, the controller weights are updated as following

$$W_c\left(k+1\right)=W_c\left(k\right)-\eta\nabla E_c\left(W_c\right) \tag{32}$$

where W_c represent hidden layer and output layer weights and ∇E_c is the gradient vector of cost function, we can write

$$\nabla E_c=\frac{\partial E_c}{\partial W_c}=\left(\frac{\partial e_c\left(k\right)}{\partial W_c\left(k\right)}\right)^T e_c\left(k\right) \tag{33}$$

The controller network gradient is defined as:

$$\nabla e_c\left(k\right)=\frac{\partial e_c\left(k\right)}{\partial W_c\left(k\right)}=\frac{\partial e_c\left(k\right)}{\partial u\left(k-1\right)}\frac{\partial u\left(k-1\right)}{\partial W_c\left(k\right)}=\frac{\partial y\left(k\right)}{\partial u\left(k-1\right)}\frac{\partial u\left(k-1\right)}{\partial W_c\left(k\right)} \tag{34}$$

Now, Identifier neural network is used to obtain an estimate of the sensitivity of the plant with regard to its inputs, Thus if Identifier neural network has N_h neuron in hidden layer, then

$$\frac{\partial y\left(k\right)}{\partial u\left(k-1\right)}=\sum_{j=1}^{N_h}\frac{\partial\hat{y}\left(k\right)}{\partial y_j^h\left(k\right)}\frac{\partial y_j^h\left(k\right)}{\partial u\left(k-1\right)} \tag{35}$$

where $y_j^h\left(k\right)$ is the output of *j th* neuron in hidden layer of identifier neural network at iteration k . Substituting (35) into (34), we have

$$\nabla e_c\left(k\right)=\sum_{j=1}^{N_h}\left(\frac{\partial\hat{y}\left(k\right)}{\partial y_j^h\left(k\right)}\frac{\partial y_j^h\left(k\right)}{\partial u\left(k-1\right)}\right)\frac{\partial u\left(k-1\right)}{\partial W_c\left(k\right)} \tag{36}$$

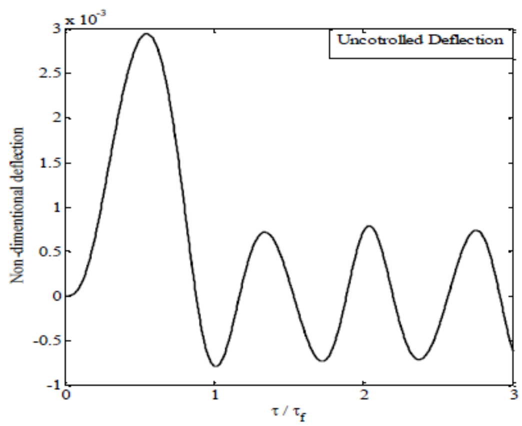
Substituting (36) into (32), the controller weights are adjusted as following

$$W_c\left(k+1\right)=W_c\left(k\right)-\eta e_c\left(k\right)\nabla e_c\left(k\right) \tag{37}$$

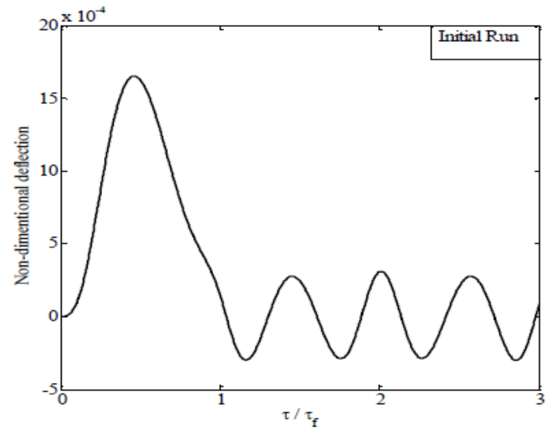
4 RESULTS AND DISCUSSION

This section presents results of the vibration suppression of a simply supported beam under excitation of a moving mass based on neural model reference controller. The identifier network was trained offline using 1500 input-output pairs generated by process model. The identifier was trained until the mean square error between identifier output and process output was below 0.001. Now, the coupled neural networks were used for control and identification plant. Which these networks were trained simultaneously. In this work two independent control forces (actuators) are considered at equal distance along the beam. In controller network, we used MLP network which consist of one hidden layer with 12 neurons and log-sigmoid activation function and an output layer with two neurons. The neural network weights are initialized to small random value in the range of ± 0.1 to start the training. The identifier network consists of one hidden layer with 10 neurons and an output layer with one neuron. The hidden layer contains log-sigmoid activation functions and the output layer contains a linear function. Identifier network weights are initialized with offline value. In this study, the following parameters are used in computing the numerical results: $E=6.5\times10^{10}Pa, L=1m, t=4\times10^{-3}m, b=32\times10^{-3}m, \rho_b=2500kg/m^3$. The performance of the hybrid control system was tested for various mass ratio of the moving mass to the beam and dimensionless velocity of a moving mass on the time history of deflection.

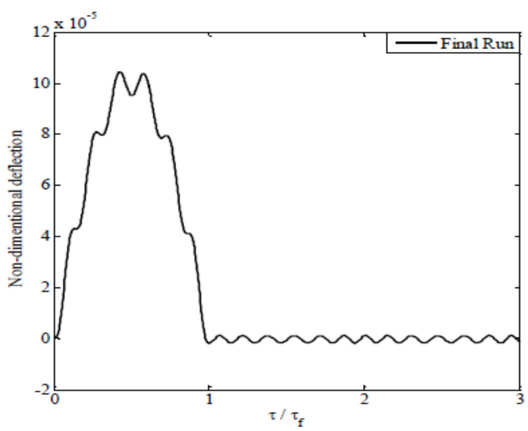
Figs.4 (a-c) show the time history of the controlled and uncontrolled dynamic deflections at mid-span of the beam with $v_0 = 0.3$ and $\Upsilon = 0.3$. Fig.4(c) shows the controller performance for the vibration suppression of the first vibrational mode of the beam by employing the neural control method. Comparing the control results with the uncontrolled case, it can be seen that the vibration is significantly suppressed. Additionally, Fig. 4(b) and 4(c) show the controlled dynamic deflection during training proposed neural network controller. These figures do indeed demonstrate significant improvement in the responses of the system.



(a)



(b)

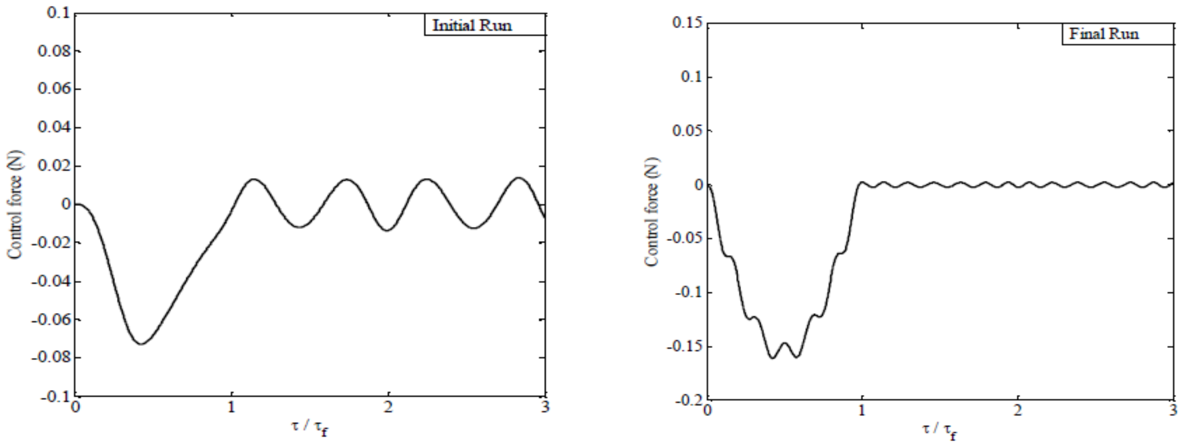


(c)

Fig.4
Controller performance as learning progresses for $v_0 = 0.3$ and $\Upsilon = 0.3$ at (1/3) spans.

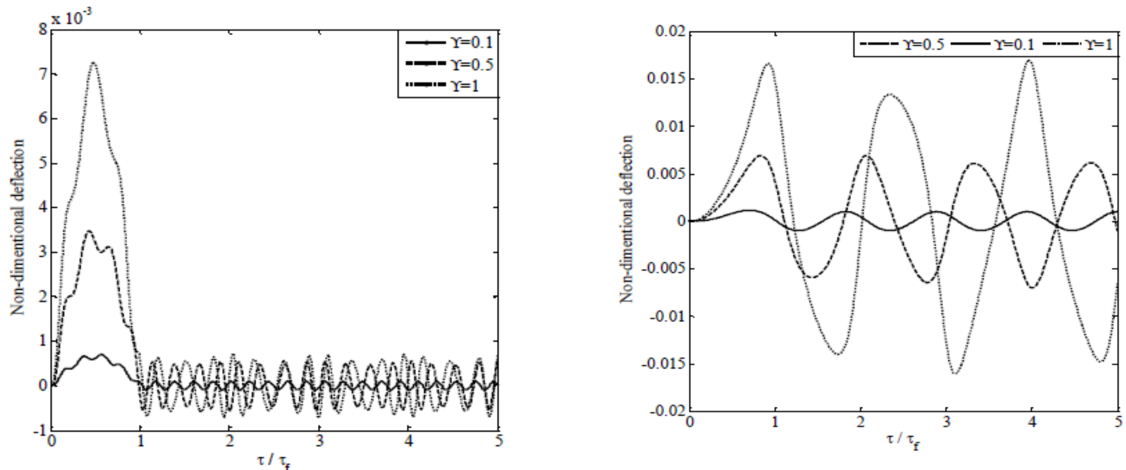
The time-domain control input forces applied to the actuators of the first vibrational mode of the beam are shown in Figs.5 (a) and 5(b). These figures show the corresponding actuator forces at each stage of training. Also because of the symmetrical placement of forces, their values are the same.

Now, the designed controller performance is evaluated for various values mass ratio of the moving mass to the beam and dimensionless velocity of a moving mass. The results are shown in Figs.6 and 7. Figs. 6(a-c) show a comparison of uncontrolled dynamic deflections of the beam for various values of the mass ratio of the moving mass to the beam $\Upsilon = 0.1, 0.5, 1$, when the dimensionless velocity of a moving mass is $v_0 = 0.1, 0.5, 1$. These figures demonstrate that, the dynamic deflections of beam rise as the mass ratio increases. Additionally, it is clear that the occurrence of the maximum dynamic deflection moves from the first phase of the motion (while the mass is still on the beam) to the second one (the mass has passed through the end point of the beam, and there is a free vibration) for all the velocity ranges from $v_0 = 0.5$ to the critical velocity as the velocity of a moving mass increases.

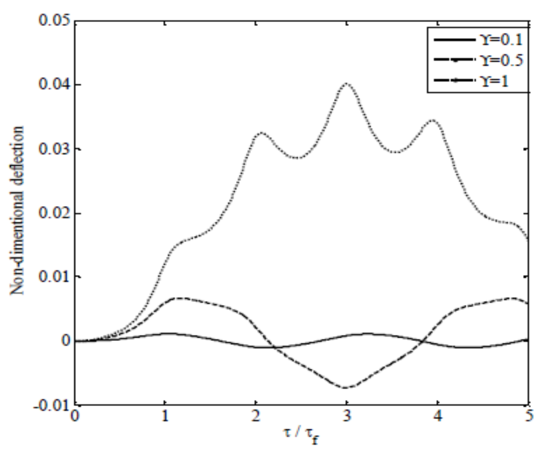


(a) (b)

Fig.5
Required control force for $v_0 = 0.3$ and $\Upsilon = 0.3$ at (1/3) spans.



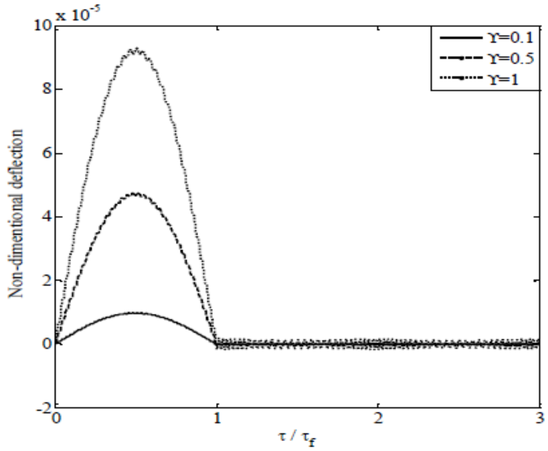
(a) (b)



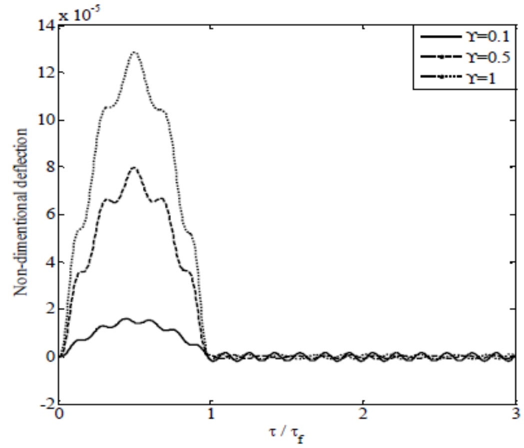
(c)

Fig.6
The time history of the uncontrolled dynamic deflection at midspan of the beam with $v_0 = 0.1, 0.5, 1$ and $\Upsilon = 0.1, 0.5, 1$.

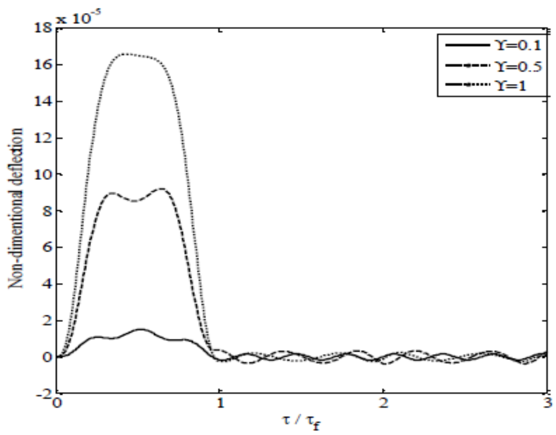
Figs.7 (a-c) show a comparison of controlled dynamic deflections of the beam for various values of the mass ratio of the moving mass to the beam Υ, when the dimensionless velocity of a moving mass is $v_0 = 0.1, 0.5, 1$. In all cases it can be seen that the neural network controller is successful in the vibration suppression. These figures confirm that the maximum dynamic responses of the beam in the first phase of the motion has decreased. Moreover, the neural network controller quickly and efficiently suppresses dynamic deflection of the beam in the second phase of the motion in comparing with its uncontrolled response and strongly counteracts the effect of the moving mass.



(a)



(b)



(c)

Fig.7
The time history of the controlled dynamic deflection at mid-span of the beam with $v_0 = 0.1, 0.5, 1$ and $\Upsilon = 0.1, 0.5, 1$.

Plots of time history of two controllers (actuators) at equal distance along the beam are provided in Figs. 8 and 9 to further investigate the effect of change in moving mass weight and velocity of moving mass on control forces. In Fig.8, for a constant velocity, the control input forces show that the magnitude is dependent upon the moving mass. As expected, the required control forces are increased by rising the moving mass weight.

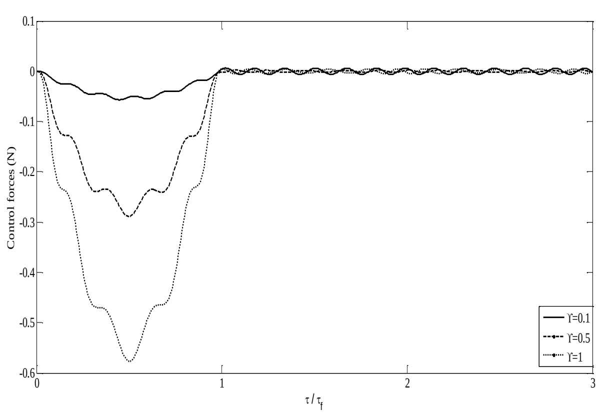


Fig.8
Required control force for $v_0 = 0.5$ and $\Upsilon = 0.3$ at (1/3) spans.

In Fig. 9, for a constant moving mass weight, the comparison of the control input forces show increasing the velocity of moving mass has not considerable effect on the amount of control forces.

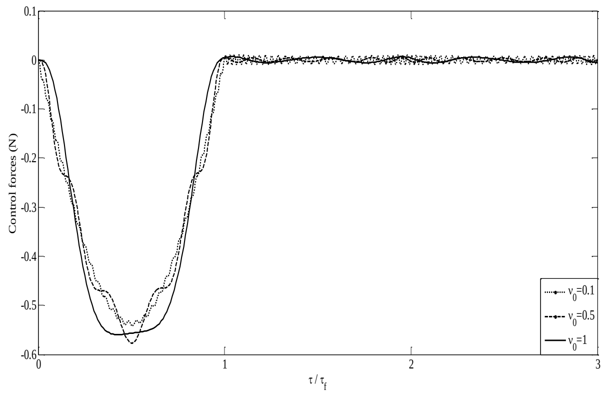


Fig.9
Required control force for $\Upsilon = 0.5$ and $v_0 = 0.1, 0.3, 1$ at (1/3) spans.

5 CONCLUSIONS

The present study examined vibration control of an Euler–Bernoulli beam under excitation of a moving mass using the model reference neural network controller. The non-dimensional equation of motion upon the beam for a moving mass was extended to a simply supported boundary condition. The NN-identifier was trained offline using data generated by process model. The controller network was trained online while; the identifier network itself learns changes in system. Numerical simulation was carried out to confirm the efficiency of the proposed neural network controller in suppressing the dynamic deflection of the beam for different mass ratios of moving mass to beam and the dimensionless velocity of a moving mass on the time history of deflection.

REFERENCES

[1] Kononov A.V., De Borst R., 2002, Instability analysis of vibrations of a uniformly moving mass in one and two-dimensional elastic systems, *European Journal of Mechanics-A/Solids* **21**(1): 151-165.

[2] Frýba L., 2013, *Vibration of Solids and Structures under Moving Loads*, Springer Science & Business Media.

[3] Bilello C., Lawrence A.B., Daniel K., 2004, Experimental investigation of a small-scale bridge model under a moving mass, *Journal of Structural Engineering* **130**(5): 799-804.

[4] Sung Y-G., 2002, Modelling and control with piezo actuators for a simply supported beam under a moving mass, *Journal of Sound and Vibration* **250**(4): 617-626.

[5] Nikkhoo A., Rofooei F. R., Shadnam M. R., 2007, Dynamic behavior and modal control of beams under moving mass, *Journal of Sound and Vibration* **306**(3): 712-724.

[6] Prabakar R. S., Sujatha C., Narayanan S., 2009, Optimal semi-active preview control response of a half car vehicle model with magnetorheological damper, *Journal of Sound and Vibration* **326** (3): 400-420.

[7] Pisarski D., CzesŁaw I.B., 2010, Semi-active control of 1D continuum vibrations under a travelling load, *Journal of Sound and Vibration* **329**(2): 140-149.

[8] Ryu B.J., Yong-Sik K., 2012, *Dynamic Responses and Active Vibration Control of Beam Structures under a Travelling Mass*, INTECH Open Access Publisher.

[9] Flanders S. W., Laura I.B., Melek Y., 1994, *Alternate Neural Network Architectures for Beam Vibration Minimization*, ASME-PUBLICATIONS-AD.

[10] Chen Ching I., Marcello R.N., James E.S., 1994, Active vibration control using the modified independent modal space control (MIMSC) algorithm and neural networks as state estimators, *Journal of Intelligent Material Systems and Structures* **5**(4): 550-558.

[11] Smyser C.P., Chandrashekhara K., 1997, Robust vibration control of composite beams using piezoelectric devices and neural networks, *Smart Materials and Structures* **6**(2): 178.

[12] Valoor Manish T., Chandrashekhara K., Sanjeev A., 2001, Self-adaptive vibration control of smart composite beams using recurrent neural architecture, *International Journal of Solids and Structures* **38**(44): 7857-7874.

[13] Qiu Zh., Xiangtong Zh., Chunde Y., 2012, Vibration suppression of a flexible piezoelectric beam using BP neural network controller, *Acta Mechanica Solida Sinica* **25**(4): 417-428.

[14] Ku Chao Ch., Kwang Y.L., 1995, Diagonal recurrent neural networks for dynamic systems control, *IEEE Transactions on Neural Networks* **6**(1): 144-156.

[15] Li X., Wen Y., 2002, Dynamic system identification via recurrent multilayer perceptrons, *Information Sciences* **147**(1): 45-63.

[16] Lin F-J., Hsin-Jang Sh., Po-Huang Sh., Po-Hung Sh., 2006, An adaptive recurrent-neural-network motion controller for XY table in CNC machine, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **36**(2): 286-299.

[17] Lin F-J., Hsin-Jang Sh., Li-Tao T., Po-Huang Sh., 2005, Hybrid controller with recurrent neural network for magnetic levitation system, *IEEE Transactions on Magnetics* **41**(7): 2260-2269.

[18] Pearlmutter Barak A., 1989, Learning state space trajectories in recurrent neural networks, *Neural Computation* **1**(2): 263-269.

[19] Yu W., 2004, Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, *Information Sciences* **158**: 131-147.

[20] Haykin S., 1998, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR.

[21] Kasparian V., Celal B., 1998, Model reference based neural network adaptive controller, *ISA Transactions* **37**(1): 21-39.