Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Sound and Vibration 287 (2005) 759–784

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

Vibration control of building structures using self-organizing and self-learning neural networks

Alok Madan*

Department of Civil Engineering, Indian Institute of Technology, Delhi, Hauz Khas, New Delhi 110016, India

Received 3 February 2004; received in revised form 26 August 2004; accepted 29 November 2004
Available online 2 February 2005

Abstract

Past research in artificial intelligence establishes that artificial neural networks (ANN) are effective and efficient computational processors for performing a variety of tasks including pattern recognition, classification, associative recall, combinatorial problem solving, adaptive control, multi-sensor data fusion, noise filtering and data compression, modelling and forecasting. The paper presents a potentially feasible approach for training ANN in active control of earthquake-induced vibrations in building structures without the aid of teacher signals (i.e. target control forces). A counter-propagation neural network is trained to output the control forces that are required to reduce the structural vibrations in the absence of any feedback on the correctness of the output control forces (i.e. without any information on the errors in output activations of the network). The present study shows that, in principle, the counter-propagation network (CPN) can learn from the control environment to compute the required control forces without the supervision of a teacher (unsupervised learning). Simulated case studies are presented to demonstrate the feasibility of implementing the unsupervised learning approach in ANN for effective vibration control of structures under the influence of earthquake ground motions. The proposed learning methodology obviates the need for developing a mathematical model of structural dynamics or training a separate neural network to emulate the structural response for implementation in practice.

*Tel.: +91 011 2659 1237; fax: +91 011 2658 1117.
E-mail address: madan@civil.iitd.ernet.in (A. Madan).

0022-460X/$ - see front matter
doi:10.1016/j.jsv.2004.11.031

1. Introduction

The modern control theory has been extensively and successfully implemented in electrical engineering systems. The past two decades has witnessed intensive research activity in the application of control theory for vibration control in civil engineering structures. A number of innovative methods have been proposed for active control of vibrations in building structures. The more popular methods for active structural control are discussed in detail by Soong [1] and Leipholtz et al. [2]. However, most of proposed control algorithms are based on a mathematical model of the structural dynamics. As a result, they cannot be applied for the vibration control of a structure with unknown dynamics. Moreover, the dynamic response of practical structures cannot be exactly predicted by their mathematical models [3]. The implementation of the available structural control methods presents several other practical problems [1], such as uncertainty in system parameters of the structural system, inherent time delay in the control loop, limited observability of the structure (due to limited number of sensors for measurement of structural response), noise in response measurements, nonlinearity and time-varying characteristics of the structure. Neural networks display a number of desirable attributes for application in structural control that are not found in conventional symbolic serial computations using von Neumann computer architectures including robust performance when presented with noisy or incomplete input patterns, high rates of parallel computations, large fault tolerances, significant ability for generalization and adaptive learning [4]. Intuitively, therefore, neural networks offer considerable potential in addressing the practical problems that challenge the effective implementation of active control of vibrations in building structures.

The development of artificial neural networks (ANN) was inspired by the characteristics and capabilities of the human nervous system. Thus, an ANN, in general, consists of interconnected computing units that are geometrically organized in one, two or three dimensions. The individual computing units can be called ‘neurons’ after the complex cells in the central nervous system that are the basic computing elements of the human brain. The multi-layer feed-forward (MLFF) neural networks with the back-propagation (BP) training algorithm are one of the most popular ANN architectures as they are simple in implementation and are capable of performing any arbitrary mapping to a desired accuracy, provided a sufficient number of hidden units are implemented and a set of connection weights that perform the desired mapping exists [4]. Details of the various architectures and training algorithms in MLFF neural nets are extensively documented in the literature [4,5]. Pioneering research studies by Chen et al. [6], Ghaboussi and Joghataie [3] and Bani-Hani and Ghaboussi [7] on the application of MLFF neural networks in simulated vibration control of structures suggest that neural networks may have the potential of evolving into powerful adaptive controllers for practical vibration control of building structures in the future.

The basic idea of using neural networks in active structural control is that the neural network acts as the controller and replaces the control algorithm [3]. Chen et al. [6] trained a MLFF neural network (neural-action network) using the BP algorithm for active control of multi-story buildings under strong ground motions. An error function was selected by Chen et al. [6] that can be expressed as the mean square of the differences between the desired response value and actual response value of various state variables. The algorithm for back-propagating the errors was formulated by minimizing this error function in the weight space of the neural network. The generalized delta learning rule [8] was adapted for updating the connection weights of the network during training. The training of the neural-action network (controller network) was performed off-line in the batch mode (i.e. training data sets were presented to the network in batches). A neural emulator network was implemented to map the selected error function to the error in the control command output by the neural action network.

Ghaboussi and Joghataie [3] trained a MLFF neural network (neural controller) for controlling the vibrations of multi-degree-of-freedom (mdof) structural models subjected to earthquake ground excitations using the "quickprop" algorithm, which is a faster version of the BP training algorithm based on the generalized delta rule. The control criterion is defined in this study by Ghaboussi and Joghataie [3] with the objective of reducing the average of the predicted responses over a few time steps in the future below some specified limit (usually a small value). An emulator neural network was trained and employed to predict the response values in the future time steps [3]. In a subsequent study, Bani-Hani and Ghaboussi [7] extended the proposed application [3] of MLFF neural networks for active structural control to nonlinear structures with hysteretic force–displacement behaviour. The neural controller in both the studies by Ghaboussi and co-workers [3,7] was trained off-line in the batch mode. More recently, Liut et al. [9] reported a BP scheme for training MLFF neural networks in active structural control that does not rely on the emulation of the structural response by an additional (emulator) network. The training scheme is based on a force-matching procedure that directly utilizes the dynamic data to characterize the structural response. Kim et al. [10] proposed a BP algorithm for training a MLFF neural network with one hidden layer in computing the control force required for optimal active control of a structure subjected to earthquake excitations. The algorithm retains the idea of utilizing an emulator network for representing the relationship between the control signal and the structural response. The BP training algorithm proposed by Kim et al. [10] can be implemented in the batch as well as pattern mode. However, the application of the algorithm was demonstrated for a single-degree-of-freedom (sdof) structure only.

The available research results on the application of MLFF neural networks with BP training in active structural control are groundbreaking achievements in the relatively new area of artificial neural control of earthquake-induced vibrations in building structures. Due to their simplicity, the MLFF network with BP training algorithm is the most popular neural network in structural control applications. However, BP training takes a long time because all the interconnection weights of the network are updated after an input training sample is presented to the network. But, the most significant limitation of MLFF neural networks with BP training algorithm is that they can learn only under the supervision of a teacher (supervised learning).

2. Unsupervised learning in artificial neural networks

Many researchers in the area of artificial intelligence believe that ANN offer one of the most promising models for development of truly intelligent microprocessors. Since ANN emulate the biological neural networks, the idea of using an ANN as an adaptive controller is immediately appealing for application in active control of structural engineering systems. The objective of a neural-network-based controller in an active structural control system is to output control signals for actuation of control forces that can effectively reduce the dynamic response of the structure within realistic desirable limits [9] under the influence of a randomly varying external disturbance (e.g. earthquake ground motion). The input pattern presented to the network typically consists of past history of structural response (for e.g. values of selected structural response variables such as displacements, velocities, accelerations measured at representative locations on the structure in the previous sampling steps) and the past history of external disturbance (for e.g. values of ground accelerations recorded at the previous sampling steps in case of an earthquake). The outputs of the network are the digital signals for communication to the actuators that apply the control forces on the structure.

All the previously reported research efforts on implementation of ANN in active structural control have employed neural networks with the BP training algorithm. However, BP learning in neural networks intrinsically requires teacher signals in the form of desired or target outputs, which are necessary for determining the errors in the network outputs, back-propagating these errors through the hidden layers and updating the connection weights between the interconnected layers of neurons. Thus, a back-propagation neural (BPN) network can learn only with the aid of teacher signals (target outputs). In essence, a BPN network learns under the supervision of a teacher. Therefore, in order to train a BPN network controller for active control of earthquake-induced vibrations in structures, target control forces must be supplied to the network. In a practical control environment, however, the control forces that are required to produce the desired response of the structure subjected to an earthquake ground motion are unknown in advance. The determination of the control forces that would result in a desired structural response is equivalent to solving the inverse dynamics problem of the structural control system [9]. As a result, the target outputs (teacher signals) of the BPN network cannot be specified in a straightforward manner. Researchers have resolved this issue using different strategies such as utilizing an emulator neural network for mapping the errors in the controlled structural response to the errors in the control forces that are output by the controller network [3,6,7] or employing a force-matching procedure to translate the dynamic structural response to the BP training error in the network output without the use of an emulator network [9].

In contrast to a BPN network that can learn only under the supervision of a teacher (supervised learning), the central nervous system (CNS) of the human body can learn the motor functions of the body without the aid of a teacher (unsupervised learning). The CNS learns to control the dynamics of the human body under the influence of a random external disturbance without any direct feedback on the correctness of the motor response of the body to the disturbance. There is no clue or indication of whether the output motor response is right or wrong except the sensed results of the applied motor control actions on the body dynamics. A possible mode of learning the appropriate motor response to the environmental stimuli is learning by trial and error from the control environment of the body with progressively decreasing magnitude of error. As an example, an infant learns to balance its body on two feet and stand upright by stressing various muscles in the body to different magnitudes in response to the sensory inputs from the environment at each instant of time. During the learning process, the brain receives continuous valued analog inputs from the sensory cells through the afferent (sensory) nerves in the peripheral nervous system (PNS), which are converted into a sequence of discrete sets of sensory inputs. The process of transforming a continuous variable to a discrete variable is termed as quantization. When a discrete set of sensory inputs is presented to the biological neural network at any instant of time, it responds by computing the output activations at that instant, which are transmitted almost instantaneously through the efferent (motor) nerves in the PNS to the effector organs such as muscles, tendons and ligaments for generating the motor response of the body to the environmental stimuli. There will be an inevitable time lag between the external stimuli and the motor response. The discrete set of sensory inputs received by the network at the current time instant and the sets of sensory inputs received at the previous few time instants along with the computed network outputs at the immediately preceding time instant may form an input data sample for example. Since no target outputs are available to the network, the network must somehow learn without the aid of teacher signals by exploiting any structure or pattern among the input data samples [4].

In the absence of a teacher, the biological neural network can learn from the environment to categorize closely related patterns into a pattern class (group or cluster) on the basis of some distinguishable features and output the motor response at the current instant of time by seeking and recognizing the pattern classes (groups or clusters) that best match the pattern presented at the current time instant. Interestingly, the feature-sensitive ANN with the unsupervised competitive learning paradigm are capable of learning an effective and generalized form of vector quantization (VQ) to encode and classify the sensory input patterns from the environment [4].

A limited number of research studies have been reported in the literature in the recent past on the applications of feature-sensitive neural networks in the area of earthquake engineering. However, all these studies are based on structural damage detection and health monitoring of building structures [11–13]. There is a need for exploring the feasibility of employing feature-sensitive neural networks in neural-network-based control of vibrations in building structures. Moreover, the theory and practice of artificial neural control of structures is still in its infancy. There is a need for performing additional computer-simulated case-studies to collect research data on the capabilities and limitations of different neural network models with different learning schemes for vibration control of structures before neural network can be implemented for structural control in practice. The present study is based on the application of feature-sensitive ANN as self-organizing and self-learning controllers for active control of structural vibrations under the influence of an earthquake-induced base excitation. A counter-propagation neural network is trained to output the control forces that are required to reduce the earthquake-induced vibrations of the structure without the aid of teacher signals (target control forces). The study shows that, in principle, the counter-propagation networks (CPN) can undergo unsupervised learning in vibration control of structures.

3. Feature-sensitive neural networks

A feature-sensitive ANN is a self-organizing and self-learning network that can categorize stochastic vectorial data. It is a generalized version of the Vector Quantization (VQ) networks [4] based on the principle of competitive unsupervised learning. The feature-sensitive ANN derive their name from the biological neural networks in the cerebral cortex of the human brain that are one of the most complex biological systems and are responsible for mapping the features of the sensory inputs to the associated spatial regions of body functions in the brain. Each spatial region in the brain contains a number of similar neurons that cooperate with one another when performing the specialized function that they have been trained to handle. Thus, the neurons in a localized spatial region respond collectively to the stimuli from the sensory cells that they service. For example, the neurons in the visual cortex of the human brain respond to light patterns received by the retina. The feature-sensitive ANN are modelled to emulate the biological feature-to-localized-spatial-region mapping of the human nervous system. The self-organizing feature map (SOFM) artificial neural network developed by Kohonen [14] is a simplified model of the biological feature mapping neural network of the brain. The SOFM networks self-adapt to input stimuli patterns described by some unknown probability distribution. As a result of the adaptation, the SOFM network maps the input patterns to output patterns in a topologically coherent manner [4]. The SOFM networks have been implemented in a number of applications including vector quantization, pattern recognition, speech recognition, data compression and robotic control.

3.1. Counter-propagation neural networks

The counter-propagation neural network (CPN) introduced by Hecht-Nielsen [15] comprises of three layers of neurons: (i) an input layer, (ii) a hidden layer of feature-sensitive neurons that are capable of classifying the input pattern presented to them on the basis of selected distinguishable features and (iii) an output layer of neurons that interpolate the network outputs from the outputs of the hidden layer using the specified interpolation scheme. The hidden layer of neurons essentially functions as a classifying or clustering device that groups the presented input patterns into classes with similar features. The metric distance from some point may be considered as a feature, for example [11]. The hidden layer of the CPN is a 1D model of the SOFM developed by Kohonen [16] and will, therefore, be referred to as the Kohonen layer in this study. The output layer of the CPN acts as an averaging device developed by Grossberg [17] and thus termed as the Grossberg layer. The input layer neurons are fully connected to each of the Kohonen layer neurons through adaptable weight vectors. The feature-sensitive neurons in the Kohonen layer undergo competitive unsupervised learning during the training phase in which all the neurons in the layer receive the same input. Every neuron in the layer inhibits all the other neurons within the layer to compete for the winning position in a ‘winner-takes-all’ competition [16]. The neuron with the weight vector that is closest to the presented input feature vector wins the competition in the layer. Thus, different input patterns end up firing different feature-sensitive neurons.

The weight update or learning is based on the minimum disturbance principle in which the weight vector of only the ‘winner’ neuron is modified [11]. In one common learning scheme, the weight vector of the winning neuron is shifted towards the input pattern vector. Fig. 1(a) illustrates in a 2D space how the weight vector of a winner neuron is updated during the learning process. The figure shows that the weight vector w of the ‘winning’ neuron is shifted towards the input feature vector d by adding a fraction of the difference $(d-w)$ to the weight vector. In case, the input feature vector presented to the vector is not similar to the weight vectors of any of the existing Kohonen layer neurons according to a pre-defined measure of similarity, a new neuron with a weight vector equal to the presented input vector is created. As the training progresses, the network classifies the input training patterns presented to the network into different clusters. When the training is completed, the network reaches a steady state in which all the input training patterns in the input vector space have been classified into one of the clusters. The output layer of

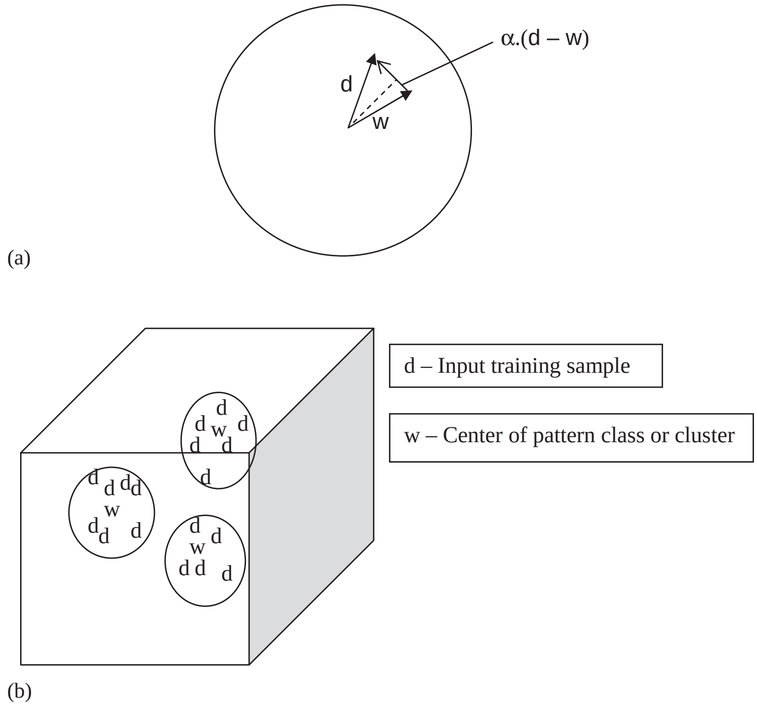


Fig. 1. Mechanism of unsupervised competitive learning in feature-sensitive neural networks. (a) Illustration of weight update during unsupervised competitive learning by neurons in the Kohonen layer. (b) Example of clustering of training patterns in three-dimensional input sample space by the Kohonen layer.

the CPN network is an averaging device that computes the network outputs during the operational phase of the network using a suitable averaging scheme.

4. Counter-propagation neural networks for vibration control in building structures

The feature-sensitive neural network implemented for structural control in this study is an improved variant of the CPN introduced by Scewczyk and Hajela [11] and was termed as the modified CPN. The architecture of the modified CPN is shown in Fig. 2. The computational algorithm of the network [11] is presented in the appendix. There are two types of connection weights in the network: (a) the weights on the connections between the neurons in the input layer and neuron '*i*' in the hidden (Kohonen) layer that are organized in a vector denoted as w_i, and, (b) the weights on the connections between the neuron '*i*' in the hidden layer to the neurons in the output layer that are organized in another vector denoted as z_i. In comparison to the earlier version of the CPN [15], the modified CPN offers additional attributes that enhance the performance of the network [11]. A network resolution parameter δ_r was incorporated that defines the measure of similarity according to which the presented input feature vector is classified into one of the existing clusters. The network resolution parameter provides the flexibility of determining the size of the Kohonen layer as well as controlling the accuracy of the

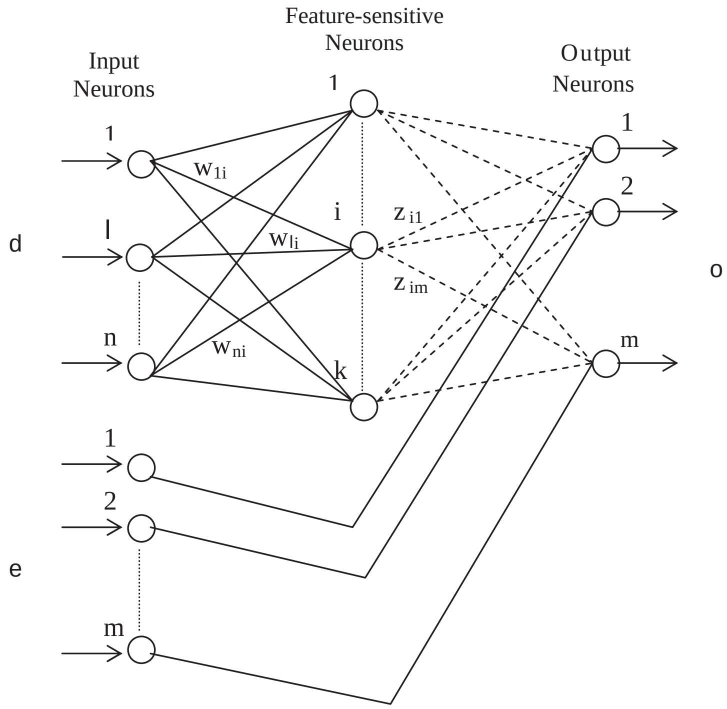


Fig. 2. Architecture of a counter-propagation neural (CPN) network.

approximations. An arithmetic mean is used during training to update the vector of connection weights z_i [11] to the output layer. The most significant enhancement in the modified CPN is the introduction of a nonlinear interpolation scheme to increase the accuracy of the network outputs during the operation of the network.

The neurons in the hidden (Kohonen) layer of the modified CPN classify the input patterns presented to them into clusters according to the metric distances between them. The classifying procedure can be explained by considering an example of clustering input patterns in the 3D space [Fig. 1(b)]. Assuming that the input patterns are presented to the network as (3×1)-dimensional feature vectors, the feature vector d_j that represents the input pattern j may be visualized as a point in a 3D space with Cartesian coordinates x_j, y_j and z_j. The weight vector w_i of any Kohonen layer neuron i will be also a (3×1)-dimensional vector and represents the centre of a cluster i in the 3D space with coordinates x_i, y_i and z_i. The metric distance between an input pattern j and centre of a cluster i may be calculated as

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{1}$$

At the start of learning, when the first input pattern is presented to the CPN, the centre of the first cluster $i = 1$ is located to coincide with the input pattern $j = 1$. In the 3D space, the cluster may be visualized as a sphere with its centre located at coordinates $x_{i=}x_j$, $y_i = y_j$ and $z_i = z_j$, where x_j, y_j and z_j are the coordinates of the point representing the input feature vector j. The radius δ_r

the sphere represents the size of the cluster and is determined by the resolution that is specified for the learning of the network. When additional input patterns are presented to the network, if the metric distance between the presented input pattern and the centre of an existing cluster is less than the network resolution δ_r, the input pattern belongs to that cluster and the centre of the cluster is shifted towards the presented input pattern according to the specified learning rule in order to account for all the patterns in the cluster. Otherwise, if the presented input pattern does not belong to any of the existing clusters, a new cluster is created centred at the point that represents the feature vector corresponding to the presented input pattern. The learning process is repeated for each subsequent input patterns that are presented to the network until all the input patterns are classified into one of the clusters as shown in Fig. 1(b). To generalize the classifying procedure, if the input patterns are presented to the network as n-dimensional feature vectors, then the input vector space is n-dimensional. During training, the input vector space is partitioned into n-dimensional clusters with the size of any cluster determined by the network resolution specified for the CPN. The n-dimensional weight vector of any Kohonen layer neuron represents the centre of a cluster and the input training patterns are classified into one of the clusters according to the Euclidean distance between the centres of the cluster and input feature vector (input pattern).

A training pattern is presented to the modified CPN in two parts that are in the form of two input vectors denoted as d and e (Appendix). In the context of structural engineering applications, generally, the input array d contains parameters that represent the effect on the structure (for e.g. the structural response or changes in response), whereas the input array e is composed of variables that are representative of the cause (e.g. the structural damage, external forces, etc.). Together, the input arrays d and e constitute the complete pattern of cause and effect required by the network to learn pattern classification and recognition. It is intuitive that both input arrays d and e are required for training the network in order to modify the weights on the connections to the hidden (Kohonen) layer w_i and the output layer z_i, while only the array d is used in the operational phase of the network. The connection weights to both the hidden and output layer are updated simultaneously during the training phase. In the proposed application of the modified CPN network for active control of structures subjected to seismic excitation, the input array 'e' may consist of the current values of the external inputs that affect the current structural response such as the current control actions on the structure and possibly the current earthquake ground acceleration, while the input array 'd' may comprise a combination of the current and past values of appropriate response variables that uniquely represents the effect of the external inputs on the structural response.

5. Application of modified CPN as a neural controller

The modified CPN network was implemented as a neural network controller in simulated active control of multi-story buildings under earthquake excitations and will be referred to as a 'CPN controller' or 'neural controller' in this study. The algorithm for training and operation of CPN controller presented in the appendix was programmed on the Matlab platform, while the dynamic response of the controlled structure was simulated using the Simulink toolbox in Matlab. Case studies were performed to investigate different combinations of input data variables for the purpose of identifying appropriate configurations of the input training pattern that result in effective training of the CPN network in active control of structural vibrations. The present section of the paper describes the details of the modelling and simulation of the structural control system, training and operation of the CPN controller and the simulated case studies.

5.1. Mathematical model of the structural control system

An eight-story building with identically constructed story units [18] was considered for illustrating the feasibility of unsupervised learning by the CPN controller for vibration control of multi-story buildings with modest control requirements. The structural control is accomplished by an active mass driver (AMD) installed at the top of the building. Fig. 3 shows the idealized structural model of the selected building with an AMD on the roof. As shown in the figure, the building is modelled as a planar frame structure with identical structural properties at each story. Based on the originally reported structural model [18], the lumped mass m assumed to be concentrated at each story is taken as 3.456×10^5 kg. The elastic stiffness k and natural damping coefficient c of each story unit are taken as 3.404×10^8 N/m and 2.937×10^6 kg/s respectively. Assuming that the floors at each story act as rigid diaphragms and the neglecting the axial deformations in the columns, each floor has 1 dof i.e. the relative lateral displacement of the floor with reference to the ground. The idealized structural model, therefore, has 8 dof corresponding to the relative displacements of the eight floors. The structural properties of the AMD were selected so that the damper mass is 2% of the generalized mass corresponding to the first mode of free vibration of the building, the damper frequency is 98% of the first mode natural frequency of the structure and the damping ratio of the damper is 7.3% [1]. Thus, the mass, natural damping constant and stiffness of the AMD system installed at the roof were adopted as 2.963×10^4 kg, 2.5×10^4 kg/s and 9.572×10^5 N/m, respectively. For purposes of simulating the dynamic response of building with the AMD at the top, the AMD was modelled as a hypothetical story unit with the aforementioned structural properties located above the top story (level 8) of the idealized structural model.

The dynamic response of the mdof structure is simulated by step-by-step integration of the differential equations that govern the motion of the structural model subjected to earthquake excitation at the base as well as the control forces at specified locations. The earthquake excitation is represented by a time-history of uniaxial ground accelerations $\ddot{x}_g(t)$ acting at the base of the structural model in the lateral direction. The equations of motion of the structural model are based on the state-space representation of the structural control system. The equations of motion in the state-space are integrated using the Simulink toolbox adopting a variable time step for integration.

5.2. Training of the CPN controller

The modified CPN algorithm presented in Appendix A was implemented for training the neural controller on-line with the simulation of the structural control system (pattern mode of training) under consideration. Fig. 3 shows a schematic block diagram for training and operation of the neural controller. The proposed scheme allows the training and operation of the neural controller concurrently. A training switch is incorporated in the block diagram that can be used in real-time

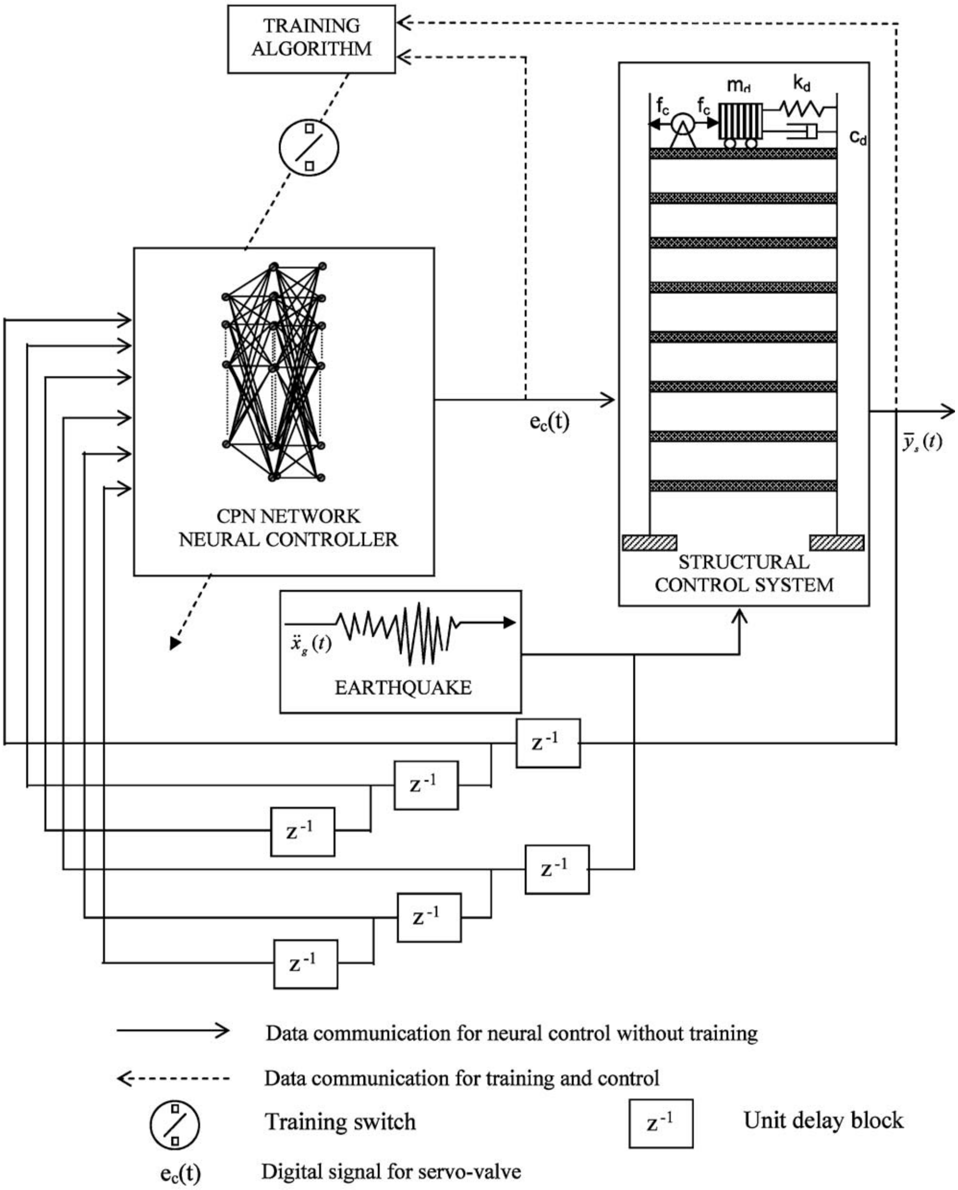


Fig. 3. Block diagram for simulation of training and operation of CPN network neural controller for vibration control of structures.

applications to discontinue the on-line training once the CPN network is trained in controlling the vibration amplitudes within the desired limits for a set of earthquake ground motions and resuming the training process again if the network needs to be re-trained for, e.g. in case of a different earthquake. During the training process, in general, the CPN network is provided at each sampling step with input data variables such as the current and delayed (sampled in the past sampling steps) signals of the simulated structural response, earthquake ground motion and control forces as the input training data. A time interval of 0.02 s (sampling interval) is employed for sampling the training data variables in this study. Each sample of the input data variables constitutes a training pattern that is presented to the network by the control environment. The network outputs are the control signals at the current sampling step that are applied as control forces on the structure at their respective specified locations.

The unsupervised learning of the CPN network begins with a single neuron in the Kohonen layer. As the training samples are input to the network, a new neuron is created in the Kohonen layer for each distinct pattern that is encountered by the network in the presented input samples. The new neuron is associated with the new pattern class or cluster to which the pattern belongs. The vector of connection weights to the new Kohonen layer neuron represents the centre of the new cluster. In case the pattern in an input sample is similar to a pattern that has already been encountered by the network according to some pre-defined measure of similarity (specified as the network resolution), then the pattern is considered as familiar and categorized into an existing cluster of similar patterns. The weight vector of the Kohonen layer neuron associated with the cluster is updated to account for the pattern most recently classified into the cluster. No new neuron is created in the Kohonen layer in this case. Eventually, in the course of training, a stage is reached when the CPN network has experienced a sufficient number of distinct patterns in the control environment and all the patterns in the sample space become familiar to the network. The number of neurons in the Kohonen layer does not increase any further upon the presentation of additional input samples. The size of the Kohonen layer stabilizes at a number of neurons that depends on the network resolution specified for the learning as well as the scatter in the training samples. The network learns to recognize all the patterns in the input data space by classifying them into one of the existing clusters (pattern classes). However, continued training of the CPN network may be necessary for further refinement in the connection weights (i.e. further learning) of the network. The speed and convergence of the learning process of the CPN will depend on the learning rate specified for the CPN.

5.3. Simulated case studies

The neural controller is trained in pattern mode using the modified CPN training algorithm during the simulation of the dynamic response of the idealized structural model of the building under the influence of earthquake ground accelerations and control actions in some cases and only the control actions in other cases. A number of cases with different configurations of the input training pattern to the CPN network and different permutations of network parameters were evaluated using the simulated structural control system in order to identify the minimum number of data variables in the input training sample and the optimum network parameters required for training the network in effective vibration control of the selected multi-story building model subjected to strong ground motions. The various cases considered in the present study can be classified into two fundamentally different categories: (a) case studies of Type A in which the array d of the input training sample presented to the CPN network consists of the history of structural response variables as well as ground accelerations, and (b) case studies of Type B wherein the array d consists of the structural response history only. The observations that emerged from the case studies on the sensitivity of the performance of the CPN controller to the configuration of the input training pattern and network parameters are summarized in the following sub-sections.

5.3.1. Influence of configuration of input training pattern on the efficacy of training

The feature-sensitive neural network (CPN) controller was assumed to be effectively trained in vibration control of the structural model if: (1) the controller learns to output control actions that can reduce the displacement at the top (eighth) level of the structure by the maximum percentage of the corresponding uncontrolled displacement that is possible using any configuration of the input training pattern and any choice of network parameters, and (2) the maximum possible percentage reduction in the structural displacements is satisfactory for applications with modest control requirements. Obviously, a training pattern configuration with the minimum number of input data variables that results in effective training is the most desirable. The case studies led to the following observations on the influence of training pattern configurations:

(i) The case studies showed that the neural controller can be effectively trained to reduce the structural displacements by considering only the displacement response history of the structure in the input array d of the training sample (training pattern). Inclusion of the velocity and/or acceleration response history of the structure in the training sample in addition to the displacement response history did not improve the performance of the trained neural controller. It is worthwhile to mention that since the seismic damage in the structural elements is directly correlated to the inelastic structural displacements, reduction of peak structural displacements is a crucial control objective in structural control.

(ii) The neural controller could be effectively trained by providing the structural displacements recorded in the current sampling step and the previous three or four sampling steps only in the input array d of the training sample. Inclusion of additional earlier past displacements in the training sample did not improve the performance of the trained neural controller. For effective training of the network, however, the input training pattern must include the past history of structural displacements up to at least three previous sampling steps. The requirement may be physically explained by the reasoning that the structural displacements sampled in the three previous steps uniquely represent the structural velocity and acceleration at the immediate past sampling step.

(iii) In case studies of Type A where the ground acceleration history was included in the input array d of the training sample, the information of ground acceleration sampled in the immediate past one step was observed to be sufficient for effectively training the neural controller. Inclusion of ground accelerations sampled in the earlier past steps in the input training sample was not found to improve the effectiveness of training. The physical reasoning of this observation is that structural displacement history provided in the training patterns already represents the effect of the earlier past ground motion.

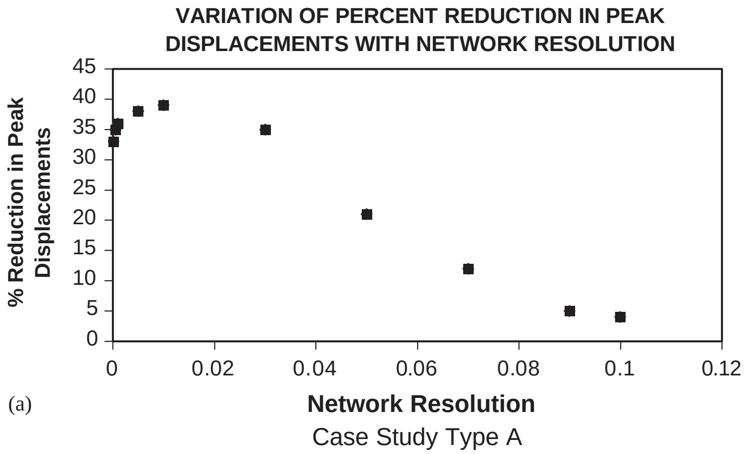
(iv) For effective training of the neural controller, only the current control force (i.e. control force computed at the current step) needs to be provided in the array e of the input training sample. Inclusion of past control force history in the training pattern did not improve the effectiveness of training. The observation is intuitively reasonable since the current structural response is uniquely determined by the past response history and the external inputs at the current instant of time.

Based on the above observations, it is proposed that the input training sample with the minimum number of data variables that results in the effective training of the CPN controller consists of the structural displacements measured in the current and past three sampling steps, the ground acceleration at the immediate past one sampling step (in case studies of Type A when the ground accelerations are considered as input data variables in the training pattern) and the control force at the current step. It may be noted that inclusion of any additional relevant data variables in the input training pattern did not result in any meaningful increase or decrease in the performance of the neural controller.

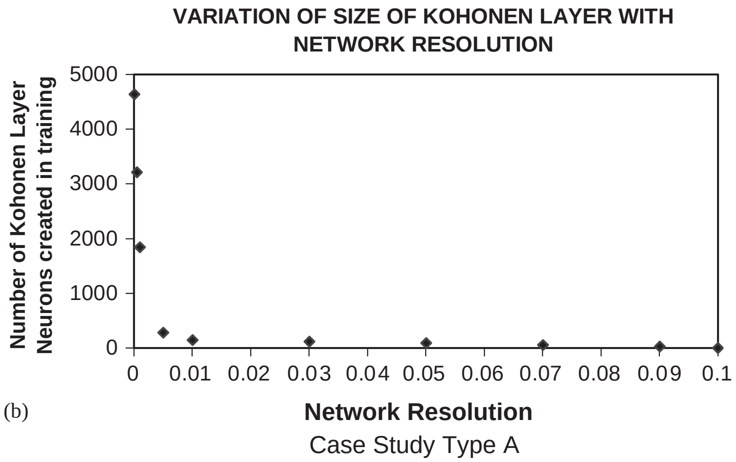
5.3.2. Influence of network parameters on the effectiveness of training

In the case studies, the performance of the trained CPN controller was observed to be most sensitive to the network resolution δ_r (Appendix) that was specified for training of the CPN network. As explained in an earlier section, the network resolution parameter defines the size of the pattern classes or clusters into which the input training patterns are classified by the Kohonen layer of the network. Since each neuron in the Kohonen layer represents a distinct pattern class (cluster), the parameter determines the number of neurons that will be created in the Kohonen layer during the training process. Larger the network resolution parameter, fewer the number of Kohonen layer neurons that will be created during the training with each neuron representing a pattern class of a large size. Increasing the network resolution parameter will thus reduce the computational time taken by the network to output the control force. However, if the network resolution parameter is too large, the performance of the neural controller may be impaired since very different input patterns may be classified into the same pattern class thus resulting in same control actions for dissimilar input patterns. Decreasing the network resolution parameter will enhance the accuracy of the neural controller up to a certain limit, beyond which the neural network tends to memorize the input patterns thus losing its ability for generalization. Moreover, decreasing the network resolution increases the number of neurons in the Kohonen layer, thus increasing the computational time and effort required for the network to calculate the control actions.

Fig. 4(a) shows the plot of variation of the percentage reduction in the top story displacement achieved by the trained neural controller with change in the network resolution specified for the controller network in case studies of Type A. It is apparent from the plot that the performance of the trained neural controller improves with decrease in the network resolution parameter used in training till an optimum value of the parameter is reached. Further decrease in the parameter beyond this optimum value does not improve the performance. Fig. 4(b) displays the variation in the number of Kohonen layer neurons generated with respect to the network resolution parameter specified in the case study. The figure indicates that too small a value of the network resolution parameter will result in an inordinately large number of neurons in the Kohonen layer thus reducing the computational efficiency of the network. In the limiting case when the network resolution is specified an extremely small value, a new neuron will be created for every training pattern that is presented to the network. The selection of the optimum value of the network resolution parameter is problem dependent and needs to be determined for the individual structural control system under consideration from parametric case studies. The order of magnitude of the parameter can be estimated by examining the ranges in the input training data



(a)



(b)

Fig. 4. Influence of the network resolution parameter on the training effectiveness of the CPN controller with unsupervised learning scheme.

variables. The optimum value of the parameter can then be determined by numerical experimentation through case studies. Based on the observations from the case studies, it is proposed that the size of the neighbourhood δ (Appendix) of the Kohonen layer neurons should be specified as half the optimum value of the network resolution parameter.

The learning rate parameter α (Appendix) specified for the CPN governs the speed and convergence of the training process of the CPN controller. Hecht-Nielsen [15] suggested a range of $0.0<\alpha<0.8$ for the parameter. As in case of the network resolution parameter, the value of the learning rate parameter to be specified for efficient training of the CPN controller is application dependent and needs to be determined for the particular structural control system through numerical trials. In the case studies performed on the structural control system considered, it was observed that the training of the CPN was most efficient with an initial learning rate of 0.3 until the size of the Kohonen layer stabilizes (i.e. the network learns to recognize all the training samples in the input sample space by classifying them into the existing clusters) and then continuing the training for further refinement in connection weights with a small learning rate of 0.1 (in order to fine-tune the weight vectors of the Kohonen layer neurons to the centres of their respective clusters).

For sake of brevity, the results of only two conceptually representative case studies in which the feature-sensitive neural network was trained with the minimum number of data variables in the input training sample and optimum network parameters are reported in this paper as follows. The first case study was of Type A wherein the input training sample includes the ground accelerations while the second was of Type B in which input training sample excludes the ground accelerations. It is intuitive that in the case study of Type A, the optimum value of the network resolution parameter would be dependent on the structural control system as well as ground excitation, whereas in the case study of Type B, the optimum value would depend only on the structural control system.

5.4. *Results and interpretations of case studies*

5.4.1. *Case study 1 (Type A)*

The input training sample considered in the case study consists of the lateral displacement at the eighth (top) level $x_8(t)$ of the structure observed at the current sampling step, delayed signals of the displacement at the top level $x_8(t-\Delta t)$, $x_8(t-2\Delta t)$, $x_8(t-3\Delta t)$ sampled in the previous three steps respectively, recorded ground acceleration $\ddot{x}_g(t-\Delta t)$ at the previous sampling step and the control force $f_c(t)$ measured at the current sampling step. Since the orders of magnitude of the structural displacement and ground acceleration are quite different, all the input data variables in the sample were normalized to the same order of magnitude by dividing the variables with their respective maximum possible values. The input arrays d and e, which are the two parts of the training sample presented to the CPN controller, may be expressed in vector notation as

$$d = \{x_8(t)\, x_8(t-\Delta t)\, x_8(t-2\Delta t)\, x_8(t-3\Delta t)\, \ddot{x}_g(t-\Delta t\}^{\mathrm{T}}, \tag{2a}$$

$$e = \{f_c(t)\}. \tag{2b}$$

The training samples of input arrays d and e were generated by simulating the dynamic response of the structural control system subjected to 30 s of the El-Centro ground motion. For a sampling interval of 0.02 s, thus, 1500 training samples were presented to the CPN controller in a single run of the simulation. In general, the simulation of the structural control system needs to be run repeatedly with the same earthquake ground motion for progressive refinement of the network weights in the training phase. The training of the CPN controller in one run of the simulation with a given ground motion is termed as an episode of training for that ground motion in the present study. Typically, the neural controller is trained for several episodes of the same earthquake. Since the CPN controller is completely untrained at the beginning of the training phase, it will not be able to output any meaningful control actions. Therefore, random control forces generated by the band-width-limited white noise source available in the Simulink toolbox were applied on the structural model during the first few episodes of training. After a few episodes, when the CPN network learns to compute control actions rationally, the random control forces may be discontinued and the control actions output by the network may be applied in further episodes.

The proposed learning scheme is instinctively consistent with the analogy of an infant learning the task of balancing the human body on the two feet. The initially untrained biological neural network of the infant will at first have to generate random motor actions in response to the environmental stimuli until the network discovers a minimum number of patterns among the input training samples constituted by the sensory inputs of the body dynamics and the motor actions that cause the dynamics. The case study indicated that the unsupervised learning by the CPN is most effective and efficient when the CPN was trained with random control forces for the first twenty episodes of the El-Centro earthquake with a learning rate of 0.3 and the training was further continued under the action of the control forces output by the CPN itself for the next 30 episodes of the El-Centro earthquake with a learning rate of 0.1. A total of 75,000 training samples (1500 in each episode) were presented to the CPN network in the course of training. The unsupervised learning by the network resulted in the creation of 144 neurons in the Kohonen layer of the network each of which represents a distinct class or cluster of similar patterns in the presented samples. After the completion of the training phase, the trained neural controller is implemented in the operational mode to compute the control forces to be actuated by the AMD in the simulated vibration control of the structural system subjected to El-Centro earthquake ground motion.

A fundamental difference in the implementation of CPN controller in the operational phase vis-à-vis the training phase is that the current displacement $x_8(t)$ (at the present sampling step) is assumed to be unavailable during the operation of neural controller, because in a practical structural control environment the current response is not known in advance. And, by the time the current response is sensed, it is too late to control it. In the absence of any information on the current structural response, the problem of supplying the complete input pattern to the neural controller in the operational phase was resolved in the present study by specifying the current displacement $x_8(t)$ as a fraction of the displacement measured in the immediately previous sampling step $x_8(t-\Delta t)$ in the input array d [Eq. (2a)]. The rationale underlying the strategy is that the CPN controller in the operational mode would then seek the pattern classes or clusters in which the response history shows a pattern of reduced displacement in the current sampling step in comparison to the previous step and output the control force corresponding to the cluster that best matches the pattern. Specifying a value of the fraction in the range 0.0–0.9 was observed to result in control actions that reduce the structural displacements in comparison to uncontrolled displacements provided the selected network parameters and configuration of input training patterns were optimal. Based on the experience with the case studies, a value of 0.7 is recommended for maximum reduction in structural displacements.

The training switch is turned off in the operational phase of the CPN controller to prevent any modifications in the architecture of the network or adjustments in the connection weights of the network during the operation of the controller. Fig. 5 shows the controlled response in comparison to the uncontrolled response (in the absence of the AMD) of the structural model under the influence of the El-Centro ground motion. The figure includes the time-history of the active control force in the AMD. It can be observed from the figure that the CPN controller with the unsupervised learning scheme implemented in this case study (Type A) reduces the top story displacements by as much as 39%. The figure also includes the optimal values of network parameters that were specified for training and operation of the neural controller. In order to evaluate the efficacy of the trained CPN controller in controlling the vibrations of the structural

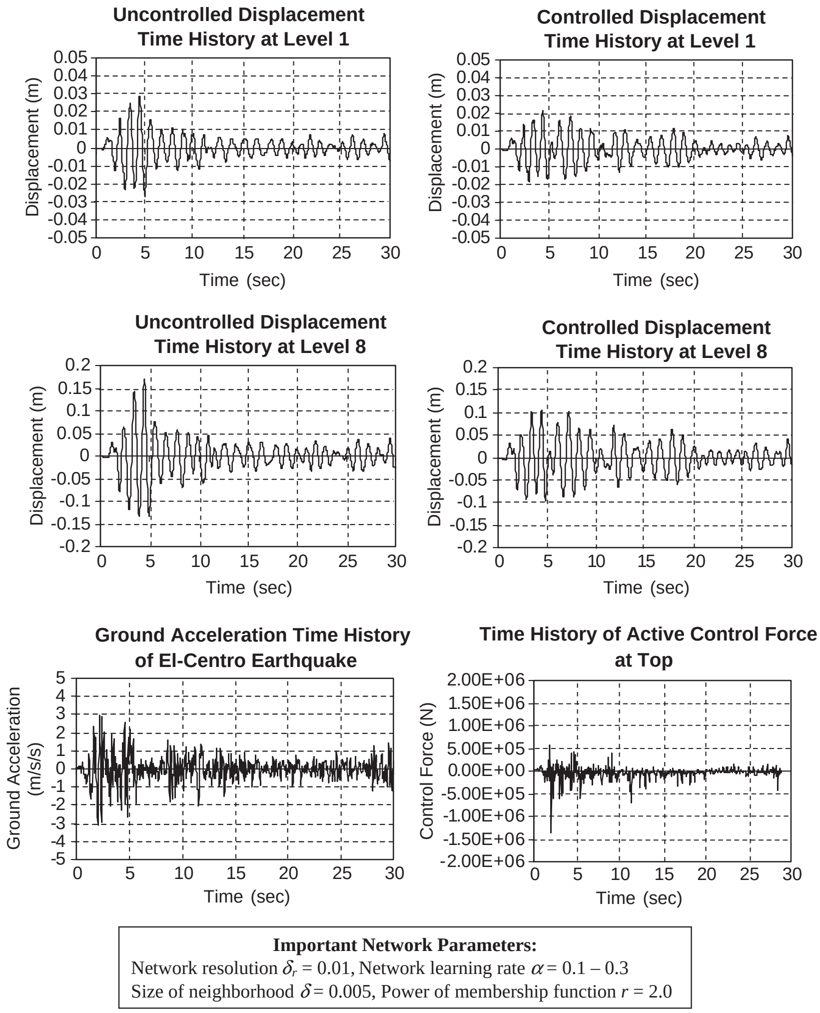


Fig. 5. Comparison of uncontrolled vs. controlled response under El-Centro earthquake using CPN network controller with the learning scheme in case study 1.

model under the influence of a different earthquake (other than the one for which the CPN controller is trained), the simulation of the structural control system was repeated with the recorded ground motion of the Taft earthquake implementing the neural controller that is trained with the El-Centro ground motion. The comparison of the controlled response and uncontrolled response at the top story of the structural model under the Taft earthquake is illustrated in Fig. 7(a). The figure shows that the trained CPN controller that has been trained in controlling the vibrations of the structural control systems due to El-Centro earthquake is also successful in reducing the peak displacement at the top level of the structural model under the influence of Taft earthquake ground motion by 27 percent.

5.4.2. Case study 2 (Type B)

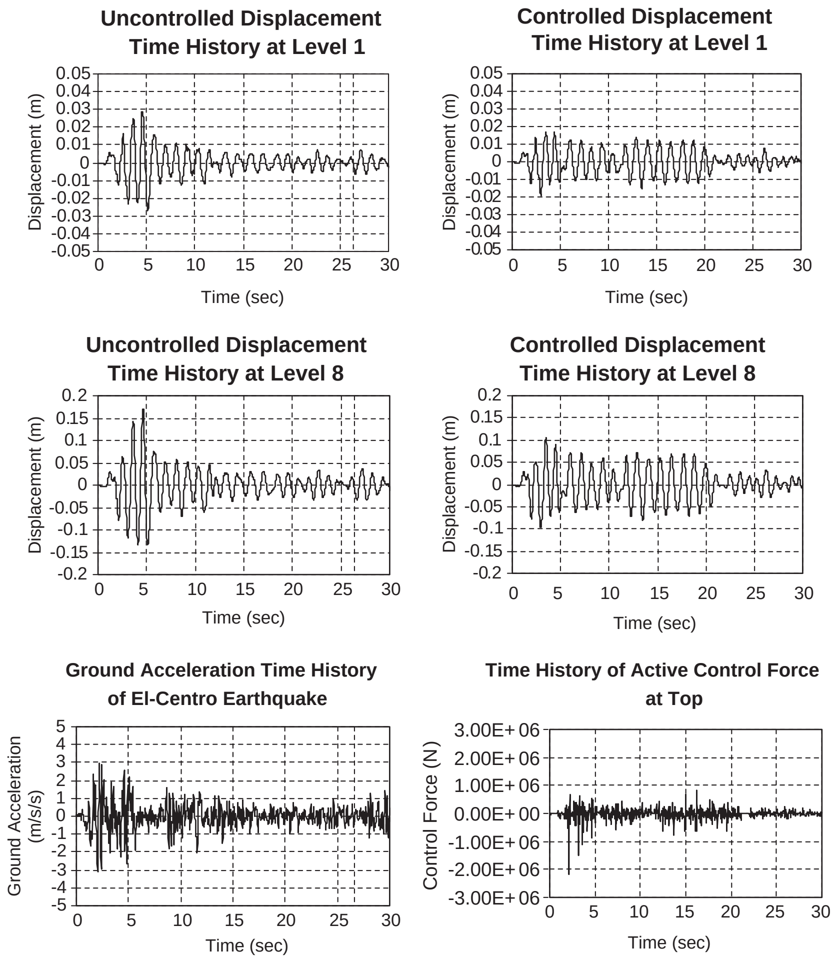
The input training sample considered in this case study does not include the ground accelerations. The rationale for the exclusion of the ground accelerations is based on the argument that biological neural networks do not receive any direct feedback on the magnitude of the environmental stimuli such as external forces due to ground shaking or gravity. The only indications of such external stimuli are the effects of the stimulus on the human body that are sensed by the nervous system. It may be hypothesized that the biological neural network concerned with computing the motor response of the human body to such environmental stimuli can only learn to predict the results of its own motor actions on the body dynamics and thus learn to compute corrective control forces that rectify the undesirable dynamic behaviour sensed in the past. Based on this reasoning, the neural controller in this case study was trained on-line with the simulation of the structural control system under the effect of random control forces in the AMD only (i.e. without the input of earthquake ground motions). The CPN network in this case was trained to classify and recognize input patterns comprising of control forces in the AMD and the resulting displacement response history at the top level of the structural model. The input training sample in the present case study was thus configured to consist of the lateral displacement at the eighth (top) level $x_8(t)$ of the structure observed at the current sampling step, delayed signals of the displacement $x_8(t-\Delta t)$, $x_8(t-2\Delta t)$, $x_8(t-3\Delta t)$ and $x_8(t-4\Delta t)$ at the top level sampled in the previous four steps respectively and the control force $f_c(t)$ measured at the current sampling step. Since all the data variables in the input array d have the same order of magnitude, there is no need to normalize the input training data. The input arrays d and e in the case study may be expressed in vector notation as

$$d = \{x_8(t)\, x_8(t-\Delta t)\, x_8(t-2\Delta t)\, x_8(t-3\Delta t)\, x_8(t-4\Delta t)\}^{\mathrm{T}}, \tag{3a}$$

$$e = \{f_c(t)\}. \tag{3b}$$

The training samples of input arrays d and e in this case were obtained by simulating the dynamic response of the structural control system subjected to random control forces generated by a band-width-limited white noise source for a time span of 30 s. The simulation of the structural control system was repeated with different noise powers of the white noise in the range 0.00004–0.00006 and different gains ranging from 1.0e07 to 1.5e07 (Newtons) with the purpose of presenting a wide variety of training patterns to the CPN network. Around a total 1,50,000 input training samples were presented to the network in all the simulations that were performed in the training phase. The number of neurons created in the Kohonen layer at the end of the training phase was 1042. It may be pertinent to mention that the simulations in the training phase in the present case study were completed within a few hours of simulation of the structural control system on the MATLAB-Simulink platform installed on a personal computer. It may be noted that the optimum values of the network parameters determined in the case study 1 (Type A) were found to yield the best results in this case study (Type B) also. Thus, it is proposed that, the optimum value of network resolution parameter is excitation-independent for the configurations of input training patterns used in the presented case studies if the data variables in the training sample are normalized to the same order of magnitude.

At the end of the training phase, the trained CPN controller was run in the operational mode to output the control forces to be applied by the AMD in simulation of the structural control system under the influence of recorded ground motions of the El-Centro and Taft earthquakes. The displacement at the current sampling step $x_8(t)$ in the input array d (Eq. (3a)) was specified as 0.7 times the displacement measured in the immediately previous sampling step $x_8(t - \Delta t)$ in the operational phase of the network for reasons mentioned earlier. The comparison of the controlled response with the uncontrolled response at the top story of the structural model (in the absence of the AMD) subjected to the El-Centro ground motion is illustrated by Fig. 6. The figure also shows the time-history of the active control force in the AMD. The optimum values of network parameters specified for training and operation of the neural controller are also included in the



Important Network Parameters:
Network resolution δ_r = 0.01, Network learning rate α = 0.1 – 0.3
Size of neighborhood δ = 0.005, Power of membership function r = 2

Fig. 6. Comparison of uncontrolled vs. controlled response under El-Centro earthquake using CPN network controller with the learning scheme in case study 2.

figure. The figure shows that the CPN controller with the unsupervised learning scheme implemented in this case study (Type B) reduces the top story displacements by at least 38%. Thus, the unsupervised learning scheme used in this case study (Type B) is as effective as the unsupervised learning scheme used in case study 1 (Type A) for training the CPN network in controlling the structural vibrations. Further, the unsupervised learning scheme used in this case study (Type B) study is independent of the earthquake ground motion characteristics, since the input training patterns presented to the network are based only on the response history of the structure and the associated control forces. Fig. 7(b) displays the comparison of the controlled response achieved by the CPN controller with the unsupervised learning scheme used in the case

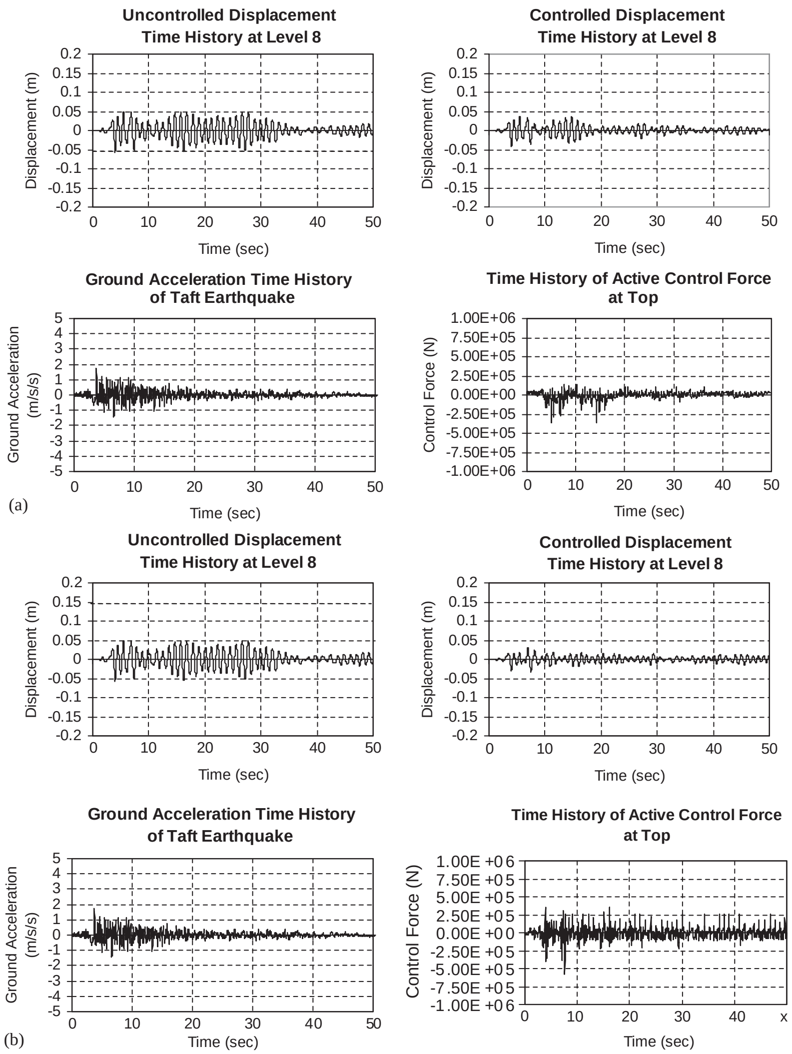


Fig. 7. Comparison of uncontrolled vs. controlled response under Taft earthquake using CPN network controller with unsupervised learning. (a) Case study 1; (b) case study II.

study with respect to the uncontrolled response at the top story of the structural model under the Taft earthquake. The figure indicates that the CPN controller trained with the unsupervised learning scheme used in the case study reduces the peak displacement at the top level of the structure by as much as 43 percent. The reduction in structural vibrations is significantly higher than that achieved using the unsupervised learning scheme in case study 1 (Type A). Thus, the results of the presented case studies indicate that the unsupervised learning scheme used in case study 2 (Type B) is more effective and efficient than the unsupervised learning scheme used in case study 1 (Type A) for training the CPN controller in active structural control.

It is pertinent to note here that based on the results of previously reported simulations of the active structural control system (AMD installed on the roof of the eight story building model) that were performed by other researchers [18], the classical closed–open loop optimal control algorithm developed by Yang et al. [18] was found to reduce the top story displacement by 60 percent, while the instantaneous optimal control algorithm [18] was found to reduce the top story displacement by 62 percent in a simulated structural control environment. In comparison, the present neural-network-based approach using a CPN controller with the unsupervised learning scheme can achieve a 43 percent reduction of the top story displacement. The lower control effectiveness of the CPN controller in simulations may be attributed to the fact that the neural controller uses the feedback of delayed signals of the structural response variables to compute the control forces, while the application of optimal control algorithms utilizes the feedback on the current instantaneous values to compute the control forces at the current time step. The formulations of the optimal control algorithms reported in the past [18] are based on the assumption that all the operations in the control loop can be performed instantaneously. In a practical structural control environment, however, a small but finite time of the order of a few milliseconds is consumed in performing the operations in the control loop. As a result, there is a time delay between the structural response and actuation of control forces. The time delay causes unsynchronized application of the control forces, which cannot only render the control ineffective but may also cause instability in the control system. Different techniques have been proposed in the past to compensate for this unavoidable time delay in practical applications. In the present study, since the trained CPN controller requires the feedback of state variables sampled at the previous sampling steps (delayed signals measured at least 0.02 s earlier) as input data for computing the current control forces in the operational phase, the proposed unsupervised learning schemes inherently account for the time delay effects. Due to their inherent capability for generalization and extrapolation of patterns, neural controllers can partially compensate for the loss of accuracy resulting from the use of delayed signals for computing the current control actions in the operational phase. Since all the operations in the control loop can be practically executed in 0.02 s, the time delay compensation is inherently integrated in the proposed learning schemes for neural networks in vibration control of structures.

6. Concluding remarks

The paper presents a possible alternative approach of learning in artificial neural networks (ANN) for active control of structures without the supervision of a teacher (unsupervised learning). The study is based on the application of feature-sensitive ANN as self-organizing and self-learning controllers that can learn to output the control forces required to reduce the structural vibrations without the aid of teacher signals (target control forces). The study shows that in the absence of a teacher, feature-sensitive neural networks can learn from the control environment to categorize closely related input samples into a pattern class (group or cluster) on the basis of some distinguishable features and output the appropriate current control signal by seeking and recognizing the pattern classes (groups or clusters) that best match the pattern of the input sample presented at the current time instant. Computer-simulated case studies are presented to demonstrate the feasibility of training a counter-propagation network (CPN) for active control of vibrations in structures under the influence of an earthquake base excitation. The competitive unsupervised learning mechanism of the CPN for active structural control was tested with different configurations of the input training samples and different combinations of network parameters. Results of two conceptually distinctive case studies performed with the optimal configurations of input training samples and optimum combinations of network parameters are presented. The results indicate that the CPN controller with the unsupervised learning paradigm can be effectively trained to reduce the earthquake-induced vibrations in building structures with modest control requirements. Comparisons of the controlled response using the CPN controller with the uncontrolled response indicates that the configuration of the input training sample that considers only the history of structural response and associated control forces (and does not consider the earthquake ground motion characteristics) results in more effective and efficient learning by the CPN for active structural control in general, since the training patterns are independent of the earthquake ground motion.

The present study hypothesises that the proposed unsupervised competitive learning approach of the CPN for active structural control is closer to the natural learning of motor control by biological neural networks in comparison to the supervised learning approach of the back-propagation neural (BPN) networks. A BPN network can learn to output the control actions required to control the structural vibrations only when the network is provided with the feedback on the desired or target control actions (teacher signals) for each input pattern presented to the network during training. On the other hand, the central nervous system (CNS) in humans can learn to control the dynamics of the human body under the influence of a random external disturbance without any feedback on whether the output motor response of the body to the disturbance is right or wrong except the sensed effects of the applied motor control actions on the body dynamics. Therefore, the supervised learning approach of BPN networks is not consistent with the process of learning motor control by the biological neural networks. Moreover, a major limitation of BPN networks is that they have a very slow rate of learning and thus take a long time to train, while a CPN network is easily trained. Furthermore, a CPN is capable of modelling a probablistic functional relation for a large domain of data and works with incomplete and/or noisy data. The CPN is a feature-sensitive neural net that functions as an associative memory device with a content-addressable memory, which stores and retrieves information based on the mutual relation among features rather than their location in the memory. The advantage of using associative memory is that even a partial knowledge of certain features may suffice an acceptable recall.

Appendix A

A.1. Training algorithm for the CPN network

In the training phase of the network, it is assumed that M training samples are presented to the network each in the form of two vectors, d_j and e_j, $j = 1,2,\ldots\ M$. The vectors, d_j and e_j constitute two parts of the training sample. The local variables and constants of the training algorithm are:

M = number of training samples presented to the network.
k = number of neurons in the hidden (Kohonen) layer = size of Kohonen layer of the network.
s_i = number of activations of the ith neuron in the hidden layer where $i = 1,\ldots, k$.
α = learning rate of the network and δ_r = network resolution.

Denoting the weight vector connecting the ith neuron in the hidden layer to the components of input array d as w_i and the weight vector connecting the ith neuron to the components of input array e as z_i, where $i = 1,\ldots.\ k$ is the index for numbering the hidden layer neurons

START
Initialize $k = 1$, $i = 1$ and $s_j = 0.0$; $j = 1,\ M$;
Set $w_i = d_1$ and $z_i = e_1$; $i = 1$;
For each $j = 2$, M
  if for all i, $\sum_{l=1}^{n} |d_{lj} - w_{li}| > \delta_r$; where $1 \leqslant i \leqslant k$, then $k = k + 1$, $w_k = d_j$ and $z_k = e_j$;
  else there exists i^* such that $\sum_{l=1}^{n} |d_{lj} - w_{li^*}| > \sum_{l=1}^{n} |d_{lj} - w_{li}|$; where $1 \leqslant i \leqslant k$
    $w_{i^*}^{\text{new}} = w_{i^*}^{\text{old}} + \alpha(d_j - w_{i^*}^{\text{old}})$; where $0 < \alpha < 1$
    $z_{i^*}^{\text{new}} = (s_{i^*}.z_{i^*}^{\text{old}} + e_j)/(s_{i^*} + 1)$;
    $s_{i^*} = s_{i^*} + 1$;
  endif
end
STOP

Outputs of the algorithm are k, w_i and z_i, $i = 1,\ldots, k$, which are stored as global variables.

A.2. Algorithm of averaging procedure for interpolation of the outputs

In the operational phase of the network, it is assumed that only an input array $\hat{d}$ is presented to the network. The approximation of corresponding output vector $\hat{o}$ is computed using a suitable averaging scheme. The algorithm of the averaging procedure uses the weight vectors w_i and z_i, $i = 1,\ldots, k$ to generate the network response $\hat{o}$. The local variables and constants of the averaging algorithm for the operation of the network are:

δ = size of the neighbourhood of the Kohonen layer neuron.
r = power of membership function.
w = index for numbering the winner neurons.
$S(w)$ = index set to remember the location of the winner neurons.

START
Initialize $w = 0$
for all i, where $1 \leqslant i \leqslant k$
  there exists i^* such that $\delta_0 = \sum_{l=1}^{n} |\hat{d}_l - w_{li^*}| \leqslant \sum_{l=1}^{n} |\hat{d}_l - w_{li}|$;
end
for all i, where $1 \leqslant i \leqslant k$
  if $\sum_{l=1}^{n} |\hat{d}_l - w_{li}| \leqslant \delta_0 + \delta$
    $w = w + 1$;
    $\delta(w) = \sum_{l=1}^{n} |d_l - w_{li}|$;
    $S(w) = i$;
  endif
end
for all s, $1 \leqslant s \leqslant w$
  $d(s) = |\delta_0 - \delta(s)|/\delta$; a normalized distance
  summation = summation + $[1 - d(s)^r]$;
end
for all s, $1 \leqslant s \leqslant w$
  $h(s) = [1 - d(s)^r]/$summation; a normalized contribution
  $i = S(s)$; location of the winner neuron in the network
  $\hat{o} = \hat{o} + h(s).z_i$
end
STOP

References

[1] T. T. Soong, *Active Structural Control: Theory and Practice*, Longman Scientific and Technical, Longman Group UK Limited, Essex, England, 1990.

[2] H.H. Leipholtz, M. Abdel-Rohman, *Control of structures*, Martinus Nijhoff Publishers, The Hague, The Netherlands, 1986.

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

[4] D.W. Patterson, *Artificial Neural Networks: Theory and Applications*, Prentice-Hall, Simon and Schuster (Asia), Singapore, 1996.

[5] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall Inc, Englewoods Cliff, NJ, 1997.

[6] H.M. Chen, K.H. Tsai, G.Z. Qi, J.C.S. Yang, F. Amini, Neural network for structure control, *Journal of Computing in Civil Engineering, ASCE* 9 (2) (1995) 168–176.

[7] K. Bani-Hani, J. Ghaboussi, Nonlinear structural control using neural networks, *Journal of Engineering Mechanics, ASCE* 124 (2) (1998) 319–327.

[8] D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing, Vol I: Foundations*, MIT Press, Cambridge, MA, 1986.

[9] D.A. Liut, E.E. Matheu, M.P. Singh, D.T. Mook, Neural-network control of building structures by a force-matching training scheme, *Earthquake Engineering and Structural Dynamics* 28 (1999) 1601–1620.

[10] J. Kim, H. Jung, I. Lee, Optimal structural control using neural networks, *Journal of Engineering Mechanics, ASCE*, (2), (2000) 201–205.

[11] Z. Szewczyk, P. Hajela, Damage detection in structures based on feature-sensitive neural networks, *Journal of Computing in Civil Engineering, ASCE* 8 (2) (1994) 163–179.

[12] J. Zhao, J.N. Ivan, J.T. DeWolf, Structural damage detection using artificial neural networks, *Journal of InfraStructure Systems, ASCE* 4 (3) (1998) 93–101.
[13] S.F. Masri, A.W. Smyth, A.G. Chassiakos, T.K. Caughey, N.F. Hunter, Application of neural networks for detection of change in nonlinear systems, *Journal of Engineering Mechanics, ASCE* 126 (7) (2000) 666–683.
[14] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43 (1982) 59–69.
[15] R. Hecht-Nielsen, Counterpropagation networks, *Journal of Applied Optics* 26 (23) (1987) 4979–4984.
[16] T. Kohonen, *Self-organization and Associative Memory*, Springer, New York, 1984.
[17] S. Grossberg, *Studies of Mind and Brain*, Reidel, Boston, MA, 1982.
[18] J.N. Yang, A. Akbarpour, P. Ghaemmaghami, New control algorithms for structural control, *Journal of Engineering Mechanics, ASCE* 113 (1987) 1369–1386.