International Journal of Innovative Computing, Information and Control
Volume **9**, Number **4**, April **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1573–1592

THE ACTIVE VIBRATION CONTROL OF A CENTRIFUGAL PENDULUM VIBRATION ABSORBER USING A BACK-PROPAGATION NEURAL NETWORK

Chi-Hsiung Liang[1] and Pi-Cheng Tung[2,*]

[1]Department of Vehicle Engineering
Army Academy R.O.C.
Jungli City, Taoyuan County 320, Taiwan
chibear@so-net.net.tw

[2]Department of Mechanical Engineering
National Central University
Jhongli City, Taoyuan County 32054, Taiwan
*Corresponding author: t331166@ncu.edu.tw

Received August 2011; revised July 2012

Abstract. *This study investigates a back-propagation (BP) neural network learning rule for control and system identification of an active pendulum vibration absorber (APV A) and develops an approach to find the bounds of learning rates based on the Lyapunov function. The use of adaptive learning rates guarantees convergence so the optimal learning rates were found. The objective of the BP algorithm was trained for tuning the system parameters in an APVA by suppressing vibration of the carrier. The simulation results for the BP neural network algorithm APVA are compared with the fuzzy BP neural network with non-neuroidentifier algorithm. The simulation results demonstrate the absorbing effectiveness of the proposed adaptive learning rates of BP neural network APVA to reduce carrier vibrations.*

Keywords: Neural network, Active vibration control, Centrifugal pendulum

1. Introduction.

A great number of rotating mechanical structures are subject to cyclical forces that cause undesirable torsional oscillatory motions during operation. The essential moving elements of a reciprocating engine are the piston, the crank, and the connecting rod. Vibration in reciprocating engines arises due to periodic variations of the gas pressure in the cylinder and inertial forces associated with the moving parts. Although a steady rotational speed is desirable in order to extend the life of the components and to reduce vibration and noise, it is generally impossible to maintain a precise constant rotational speed. In general, the reduction of torsional vibrations in a rotating shaft, such as automotive crank shafts, is achieved using different classes of dampers or absorbers [1,2].

Various methods have been proposed to suppress these vibrations, including the addition of torsional friction dampers and flywheels. However, neither of these methods is ideal. Friction dampers waste energy and generate heat, while flywheels not only increase the total mass of the system, but also reduce its responsiveness. Centrifugal Pendulum Vibration Absorbers (CPVAs) are also used to suppress torsional vibrations in rotating machinery. They are widely employed in light aircraft engines and helicopter rotors, as well as some high-performance automotive racing engines and diesel camshafts. The CPVA, however, is essentially a tuned absorber whose natural frequency varies in direct proportion to the rotational speed of the crankshaft. It also introduces no additional weight increase to the design of the system and thus can be used in many applications. This makes the CPVA an attractive alternative for the reduction of torsional vibrations in engines and a detailed study of its dynamic performance worthwhile.

The CPVA is comprised of a simple pendulum mounted on a vibrating rotor, as shown in Figure 1. The pendulum suppresses the effects of torsional disturbances in the linear range. Furthermore, the pendulum adds only a small amount of mass to the system and hence does not increase the rotating inertia. Additionally, the pendulum dissipates an insignificant amount of energy in the form of heat. The CPVA absorber was first introduced in 1929 [3] and has since been widely applied to eliminate torsional vibrations in geared radial aircraft-engine propeller systems [4]. Den Hartog and Newland [5,6] conducted a linear analysis of the dynamic response of the CPVA and briefly discussed the characteristics of its large amplitude nonlinear motion. Sharif-Bakhtiar and Shaw [7] studied the effects of damping on a CPVA with moderate amplitude motion and motion-limiting stops. Sharif-Bakhtiar and Shaw [8] also reported on the effects of nonlinearities and damping on the dynamic response of a centrifugal pendulum vibration absorber. Cronin [9] considered the problem of shake reduction in automobile engines using a CPVA. Subsequently, various kinds of CPVA have been employed. Schtter et al. [10] introduced a cam-based centrifugal pendulum (CBCP). The CBCP can completely balance the torque of a purely inertial mechanism for any speed, using an ordinary disc cam. Haddow and Show [11] demonstrated a rigid rotor fitted with several point mass absorbers that move along prescribed paths relative to the rotor. The rotor is subjected to an applied torque that fluctuates at a given order of rotation and the absorber paths are selected such that their motion counteracts the applied torque. Sung et al. [12] designed a well-weighted pendulum to harvest energy from a rotating wheel, which consisted of a pendulum and one or more weights. The conventional CPVA is a passive absorber optimized to suppress disturbance vibrations with a specific frequency. However, if the frequency of the disturbance varies with time, the efficiency of the absorber is reduced and the CPVA will fail to provide a satisfactory performance. To overcome this limitation of passive vibration absorbers, an active dynamic absorber can be developed to retain the advantages of the conventional CPVA. An active dynamic vibration absorber can be tuned according to the system characteristics to meet desired requirements [13,14]. The active vibration absorption technique uses the centrifugal delayed resonator (CDR) for active and passive device synthesis of the centrifugal pendulum absorber and the delayed resonator [15,16]. A pendulum with a spinning base is capable of neutralizing the vertical, harmonic excitation, the oscillation of the primary structure and the up-down swing of the pendulum. The proposed scheme is suitable for an active control system where energy consumption is a concern and there is a constraint on the power or force capacity of a linear actuator [17].

The active pendulum vibration absorber (APVA) is simply a CPVA to which is appended a torque motor that acts as an actuator to drive the pendulum at an adequate oscillatory amplitude. It has a very wide range of operating frequencies. Genetic algorithms (GAs) are based on the mechanism of natural selection and evolution and they have been applied to search for the global optimum in the APVA. Genetic algorithm applications in controls that are performed in real time are limited because of random solutions and convergence [18]. The fuzzy BP neural network controller systems with non-neuroidentifier architectures suppress the vibration of the carrier in the APVA. The algorithms are viewed as conventional fuzzy algorithms for coarse tuning and the BP algorithm is applied for fine tuning with a fixed learning rate. However, the fixed learning tends to be inefficient [19]. The derivations of the CPVA and APVA equations are presented according to Newton's second law of motion. The purpose of this study is to construct an APVA with a BP neural network mechanism. The simulation results verifying the effectiveness of the active vibration control are discussed, and finally, some conclusions are offered.

2. **APVA System Modeling.** Figure 1 depicts a schematic view of the centrifugal pendulum vibration absorber. The CPVA is comprised of a simple pendulum mounted on a vibrating carrier. The carrier wheel is driven by a rotary machine, with the torque T_c applied to the carrier as shown. The primary structure is equipped with a rotational pendulum which is driven by a motor. For simplicity, the rotational pendulum is assumed to have a length l, $l = 0.0375$ m and a lumped mass m, $m = 0.24288$ kg, at the tip.

The CPVA motion equation is analyzed in this section, as shown in Figure 1. The carrier in the CPVA rotates around a fixed axis normal to the X-Y plane through O. A rotating frame i-j is attached to the pendulum with its origin at point P and represents the transformation of the vector between the rotating and nonrotating axes. The i and j-axes of the coordinate system are aligned parallel and normal to l, respectively. Θ is the absolute angular displacement of the carrier. r, $r = 0.067$ m is the radius of the carrier and ϕ is the relative angular displacement of the pendulum with respect to the carrier.

The absolute acceleration of m can be written as

$$a_m = [-l(\dot{\phi} + \dot{\Theta})^2 - r\dot{\Theta}^2 \cos\phi + r\ddot{\Theta}\sin\phi]\vec{i} + [l(\ddot{\phi} + \ddot{\Theta}) + r\dot{\Theta}^2 \sin\phi + r\ddot{\Theta}\cos\phi]\vec{j}. \tag{1}$$

The moment equation about point P can be expressed as

$$ml[l(\ddot{\phi} + \ddot{\Theta}) + r\dot{\Theta}^2 \sin\phi + r\ddot{\Theta}\cos\phi] = -d_p.\dot{\phi}, \tag{2}$$

where d_p is the damping coefficient of the pendulum, $d_p = 0.0001$.

The equation of the moments about O becomes

$$\begin{aligned} I_c\ddot{\Theta} + m(l + r\cos\phi)[l(\ddot{\phi} + \ddot{\Theta}) + r\dot{\Theta}^2 \sin\phi + r\ddot{\Theta}\cos\phi] \\ + mr\sin\phi[-l(\dot{\phi} + \dot{\Theta})^2 - r\dot{\Theta}^2 \cos\phi + r\ddot{\Theta}\sin\phi] = T_c - d_c\dot{\Theta} \end{aligned} \tag{3}$$

where I_c is the moment of inertia of the carrier, $I_c = 0.01280$ kg.m^2, d_c is the damping coefficient of the carrier, $d_c = 0.01$.

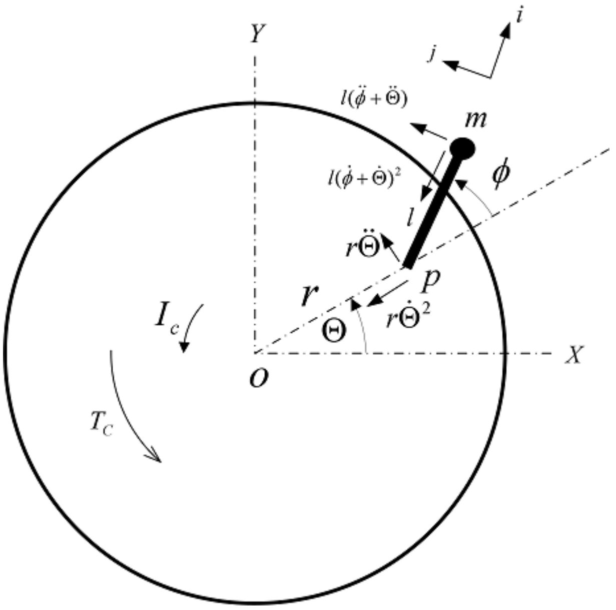

FIGURE 1. Schematic representation of a centrifugal pendulum vibration absorber (CPVA)

Hence, from Equations (2) and (3), the equations of motion for the centrifugal pendulum vibration absorber system can be written as:

$$\begin{gathered}
ml^2\ddot{\phi} + (ml^2 + m.lr\cos\phi)\ddot{\Theta} + d_p.\dot{\phi} + m.lr\dot{\Theta}^2\sin\phi = 0,\\
[I_c{+}m.l(l + r\cos\phi) + m(l + r\cos\phi)r\cos\phi + mr^2\sin^2\phi]\ddot{\Theta} + [ml(l + r\cos\phi)]\ddot{\phi}\\
-m.rl\dot{\phi}^2\sin\phi - 2mrl\dot{\phi}\dot{\Theta}\sin\phi + d_c\dot{\Theta} = T_{c.}
\end{gathered}\tag{4}$$

In the current example, the torque that acts on the carrier T_c is a constant torque T_{dc} plus a harmonic disturbance, T_a, $T_a = 2.5\sin(\omega t)$, i.e., $T_c = T_{dc} + T_a$. Since the system generally undergoes a gross rotational motion, Θ can be expressed as follows:

$$\Theta(t) = \Omega t + \theta(t); \quad \dot{\Theta}(t) = \Omega + \dot{\theta}(t); \quad \ddot{\Theta}(t) = \ddot{\theta}(t). \tag{5}$$

That is, the carrier angular displacement is composed of a nominal steady rotation and an oscillating part. The first and second time derivatives of $\theta(t)$ are the angular velocity and acceleration of the carrier, respectively.

Substituting Equation (5) into Equation (4), the two equations of the system can be expressed as

$$\begin{gathered}
ml^2\ddot{\phi} + (ml^2 + m.rl\cos\phi)\ddot{\theta} + d_p.\dot{\phi} + m.rl(\Omega + \dot{\theta})^2\sin\phi = 0,\\
(I_c + m.r^2 + ml^2 + 2mrl\cos\phi)\ddot{\theta}{+}(ml^2{+}m.rl\cos\phi)\ddot{\phi}{-}m.rl\dot{\phi}^2\sin\phi\\
-2mrl\dot{\phi}(\Omega + \dot{\theta})\sin\phi + d_c(\Omega + \dot{\theta}) = T_{dc} + T_a.
\end{gathered}\tag{6}$$

Therefore, eliminating the effect of the constant torque term of the mean component of the resistance torque, which arises from carrier damping, is completely counteracted by the constant torque term, $T_{dc} = d_c.\Omega$, where Ω is the angular velocity of the carrier, $\Omega =$ 188.5 rad/s (1800 rpm).

The dynamics of the CPVA system are given by the following differential equations:

$$\begin{gathered}
ml^2\ddot{\phi} + (ml^2 + m.rl\cos\phi)\ddot{\theta} + d_p.\dot{\phi} + m.rl(\Omega + \dot{\theta})^2\sin\phi = 0,\\
(I_c + m.r^2 + ml^2 + 2mrl\cos\phi)\ddot{\theta} + (ml^2 + m.rl\cos\phi)\ddot{\phi} - m.rl\dot{\phi}^2\sin\phi\\
-2mrl\dot{\phi}(\Omega + \dot{\theta})\sin\phi + d_c\dot{\theta}) = T_a.
\end{gathered}\tag{7}$$

In this study, T_p is the torque developed by the DC motor shaft to actuate the pendulum and is expressed as $T_p = K_p.\phi$ (like torsional spring).

The dynamics of the nonlinear APVA system can be given by the following differential equations:

$$\begin{gathered}
ml^2\ddot{\phi} + (ml^2 + m.rl\cos\phi)\ddot{\theta} + d_p.\dot{\phi} + m.rl(\Omega + \dot{\theta})^2\sin\phi = T_p,\\
(I_c + m.r^2 + ml^2 + 2mrl\cos\phi)\ddot{\theta} + (ml^2 + m.rl\cos\phi)\ddot{\phi} - m.rl\dot{\phi}^2\sin\phi\\
-2mrl\dot{\phi}(\Omega + \dot{\theta})\sin\phi + d_c\dot{\theta}) = T_a.
\end{gathered}\tag{8}$$

The APVA is simply a CPVA to which a torque motor is appended. The motor acts as an actuator to drive the pendulum at an adequate oscillatory amplitude. The APVA has a very wide range of operating frequencies.

For numerical solutions to the nonlinearly coupled equations, the fourth and fifth order Runge-Kutta formulas will be applied to solve Equation (8). By defining $x_a = \theta$, $x_b = \dot{\theta}$, $x_c = \phi$, and $x_d = \dot{\phi}$, Equation (8) can be written as four first-order differential equations. According to the fourth and fifth order Runge-Kutta method in MATLAB, the recurrence sequence can be used to find the values of $[\theta, \dot{\theta}, \phi, \dot{\phi}]$ in each time step.

In the current APVA, T_p is the torque needed to actuate the pendulum. Figures 2 and 3 show the frequency response curves for the carrier and the pendulum, respectively,

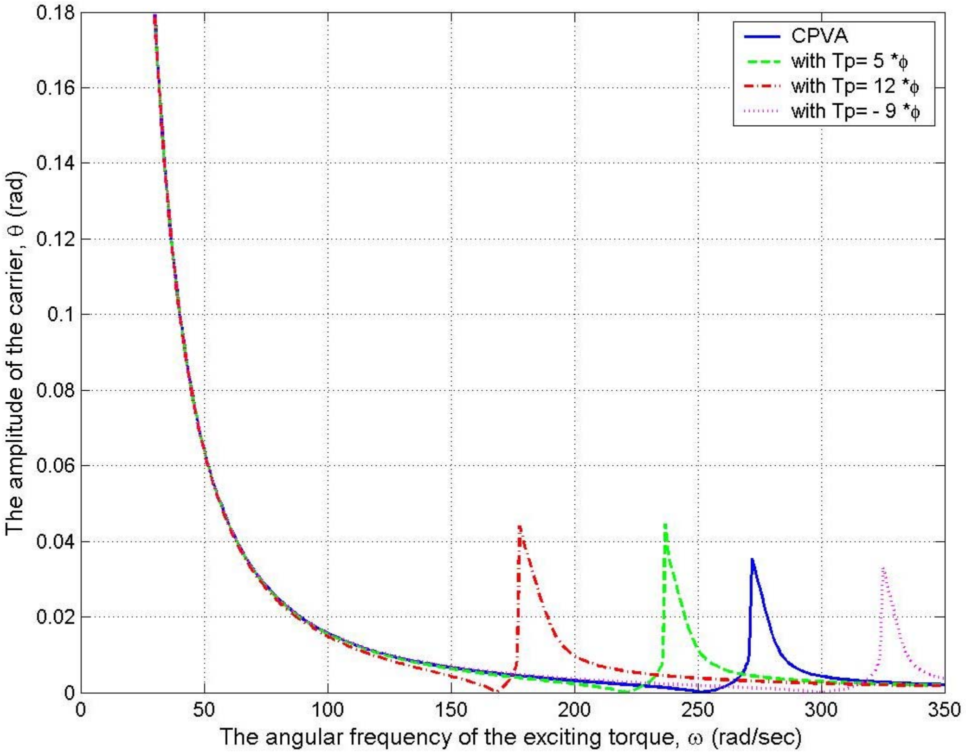

Figure 2. Frequency response of the carrier with varying $T_p = K_p.\phi$

with different K_p values for the torque, $T_p = K_p.\phi$. From the curves with excitation frequencies from 30 rad/sec to 350 rad/sec, we can observe that at this frequency, the carrier amplitude of the oscillation is reduced to approach zero, and the absorber has a finite amplitude of oscillation. This frequency is called the nonlinear anti-resonance frequency for the damped, nonlinear system. The solid lines indicate the nonlinear CPVA when $K_p = 0$; the different types of dashed lines indicate different K_p values for the torque T_p. Superimposed on these plots are the results from the simulations of the nonlinear equations of motion. It is found that altering the K_p values of the torque T_p will shift the anti-resonance frequency in the nonlinear APVA.

The purpose of this study is to construct a nonlinear APVA with control algorithms to shift the anti-resonance frequency, for the improvement of the vibration performance.

3. **APVA Using a Back-Propagation Neural Network.** The neural network model has often been applied in industry purposes. Neural network algorithms have also been used successfully for a widely variety of applications [20-23]. In this study, a back-propagation neural network learning rule is used to tune an active pendulum vibration absorber. The back-propagation algorithm is the most commonly used type of the neural network architecture for supervised learning because it is based on the weight error correction rules. The back-propagation algorithm has the advantages of simplicity and relatively simple implementation. An approach for control and system identification is presented in this section. An unknown plant is identified by a system identifier that provides information about the plant to a controller. The neurocontroller is used to drive the unknown dynamic system such that the error between plant and desired output is

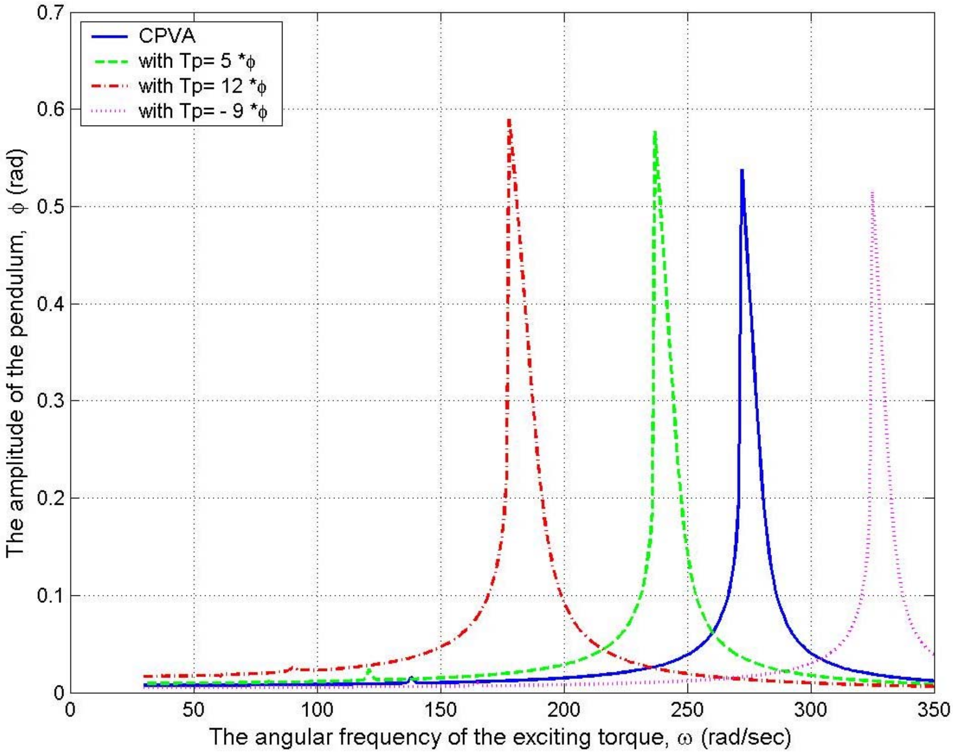

Figure 3. Frequency response of the pendulum with varying $T_p = K_p.\phi$

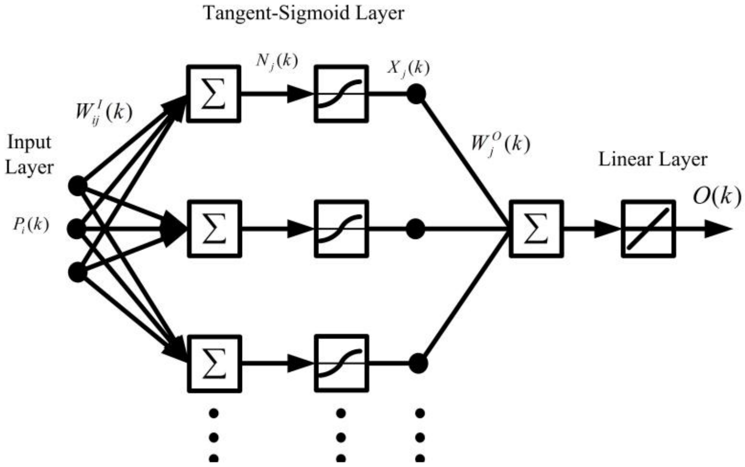

Figure 4. Multilayered feedforward neural network architecture

minimized. A generalized algorithm, called the dynamic backpropagation (DBP), is developed to train both the neurocontroller and neuroidentifier [24]. Both neurocontroller and neuroidentifier use the same neural network architecture, as shown in Figure 4.

From Figure 4, it can be seen that the input layer distributes inputs to the first hidden layer. The inputs then propagate forward through the network and each neuron computes its output according to

$$X_j(k) = g_j\left(\sum_{i=1}^{m} P_i(k) W_{ij}^I(k)\right), \tag{9}$$

where for each iteration k, $P_i(k)$ is the ith input and $W_{ij}^I(k)$ is the input weight of the connection between the ith neuron of the input layer and the jth neuron of the hyperbolic tangent sigmoid layer. For neurons in the hidden layers, the activation function is often chosen to be

$$g_j(N_j(k)) = \left(e^{Nj(k)} - e^{-Nj(k)}\right) / \left(e^{Nj(k)} + e^{-Nj(k)}\right), \tag{10}$$

where $g_j(N_j(k))$ is a hyperbolic tangent sigmoid function and $N_j(k)$ is the sum of inputs to the jth recurrent neuron,

$$N_j(k) = \sum_{i=1}^{m} P_i(k) W_{ij}^I(k) \tag{11}$$

The output nodes of the linear layer are said to be linear neurons. The output node performs a weighted sum of its inputs as follows:

$$O(k) = \sum_{j=1}^{n} W_j^O(k) X_j(k), \tag{12}$$

where $O(k)$ is the output of the network, $X_j(k)$ is the output of the jth recurrent neuron, and $W_j^o(k)$ is the output weight of the jth recurrent neuron.

The block diagram of the neural network based control system for the APVA is shown in Figure 5.

The inputs to the neurocontroller are the reference input u_r, the previous output of the neurocontroller $u(k-1)$, and the previous responses of the APVA plant $y_p(k-1)$. The output of the neurocontroller $u(k)$ is the control signal to the plant. The DBP algorithm developed in this study is used to adjust the weights of the neurocontroller such that the error between the output of the plant and the desired output from a reference model approaches a small value after some training cycles.

The term $y_u(k) \equiv \partial y_p(k)/\partial u(k)$ represents the sensitivity of the plant with respect to its input. Since the plant is normally unknown, the sensitivity term $y_u(k)$ is also unknown. This unknown value can be estimated by using the neuroidentifier. When the neuroidentifier is trained, the dynamic behavior of the neuroidentifier is close to the unknown plant, then $y_u(k) \equiv \partial y_{id}(k)/\partial u(k)$, where the value of $y_u(k)$ is a parameter for the neurocontroller.

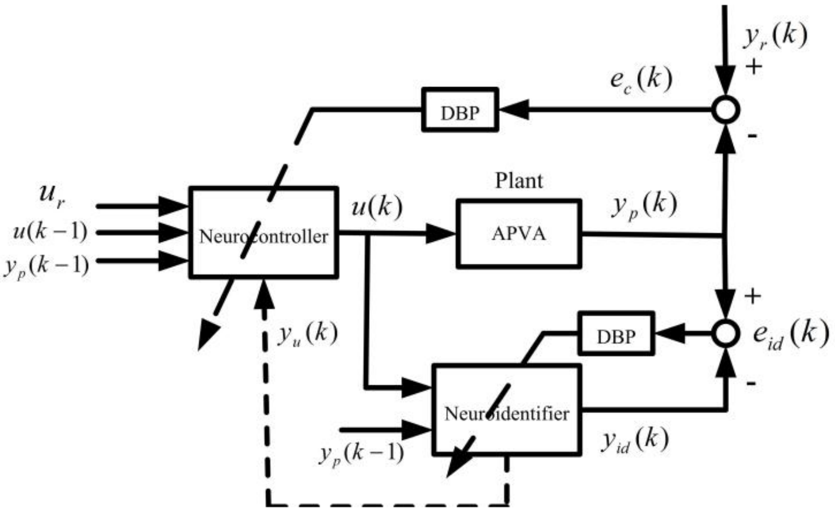

Figure 5. Block diagram of the neural network based control system for the APVA

3.1. **Dynamic backpropagation for the neuroidentifier.** This paper presents a backpropagation neural network approach for the APVA dynamic model identification. Identification is basically the process of developing or improving a mathematical representation of system dynamics. An approach for the identification of a nonlinear dynamic system using neural networks involves the dynamic differential equation into each of the neural network processing elements to create a new type of neuron called a dynamic neuron. From this paper the inputs of the neuroidentifier network are chosen to be $P_{id} = \{u(k), y_p(k-1)\}$; then the output $y_{id}(k)$ of the neural network should converge to the responses of the APVA plant $y_p(k)$ through the training process, as shown in Figure 5. The subscript *id* represents the neural network architecture for the neuroidentifier. The weight of the neural network is first initialized with small random numbers. The outputs of the neural network are computed by feeding forward the inputs through the network. The error $e_{id}(k)$ is calculated from the difference between the responses of the APVA plant $y_p(k)$ and the outputs of the neural network $y_{id}(k)$. By backpropagation of the error function to adjust the weights, the neural network can be trained to reach a desired accuracy for modeling the dynamic behavior of the APVA. An error function for a training cycle for the neuroidentifier can be defined as

$$E_{id}(k) = 1/2(y_p(k) - y_{id}(k))^2. \tag{13}$$

The weights can be adjusted following a gradient method, i.e., the updating rule for the weights becomes

$$W_{id}(k+1) = W_{id}(k) - \eta_{id}(\partial E_{id}(k)/\partial W_{id}(k)), \tag{14}$$

where η_{id} is the learning rate of the neuroidentifier.

The gradient of error in Equation (13) with respect to output weight is represented by

$$\begin{aligned}\partial E_{id}(k)/\partial W^O_{(id)j}(k) &= \partial E_{id}(k)/\partial y_{id}(k).\partial y_{id}(k)/\partial W^O_{(id)j}(k)\\ &= \partial E_{id}(k)/\partial y_{id}(k).\partial O_{id}(k)/\partial W^O_{(id)j}(k).\end{aligned} \tag{15}$$

The output of the neuroidentifier $y_{id}(k)$ is equal to $O_{id}(k)$, where $O_{id}(k)$ is the output of the neural network architecture.

From Equation (12), the output gradients for the neuroidentifier with respect to the output weights become

$$\partial O_{id}(k)/\partial W^O_{(id)j}(k) = X_{(id)j}(k). \tag{16}$$

The partial derivatives are given by

$$\partial E_{id}(k)/\partial y_{id}(k) = -(y_p(k) - y_{id}(k)) = -e_{id}(k), \tag{17}$$

where the neuron identification error is defined as $e_{id}(k) = y_p(k) - y_{id}(k)$.

Then,

$$\partial E_{id}(k)/\partial W^O_{(id)j}(k) = -e_{id}(k)X_{(id)j}(k). \tag{18}$$

The gradient of error in Equation (13) with respect to input weight is represented by

$$\begin{aligned}\partial E_{id}(k)/\partial W^I_{ij}(k) &= \partial E_{id}(k)/\partial y_{id}(k).\partial y_{id}(k)/\partial W^I_{ij}(k)\\ &= \partial E_{id}(k)/\partial y_{id}(k).\partial O_{id}(k)/\partial W^I_{ij}(k).\end{aligned} \tag{19}$$

From Equation (12), the output gradients for the neuroidentifier with respect to the input weights become $\partial O_{id}(k)/\partial W^I_{(id)ij}(k)$.

Then,

$$\begin{aligned}\partial O_{id}(k)/\partial W^{I}_{(id)ij}(k) &= \partial O_{id}(k)/\partial X_{(id)j}(k).\partial X_{(id)j}(k)/\partial W^{I}_{(id)ij}(k)\\ &= W^{O}_{(id)j}(k).\partial X_{(id)j}(k)/\partial W^{I}_{(id)ij}(k),\\ &= W^{O}_{(id)j}(k).\partial X_{(id)j}(k)/\partial N_{(id)j}(k).\partial N_{(id)j}(k)/\partial W^{I}_{(id)ij}(k)\\ &= W^{O}_{(id)j}(k)g'_j(N_{(id)j}(k))P_{id}(k)\end{aligned} \tag{20}$$

Thus,

$$\begin{aligned}g'_j(N_{(id)j}(k)) &= d((e^{N(id)j(k)} - e^{-N(id)j(k)})/(e^{N(id)j(k)} + e^{-N(id)j(k)}))/dN_{(id)j}(k)\\ &= 1 - (g_j(N_{(id)j}(k)))^2 = 1 - (X_{(id)j}(k))^2.\end{aligned} \tag{21}$$

Since inputs to the neuroidentifier are $P_{id} = \{u(k), y_p(k-1)\}$, Equation (11) becomes

$$N_{(id)j} = u(k)W^{I}_{(id)1j}(k) + y_p(k-1)W^{I}_{(id)2j}(k). \tag{22}$$

Thus,

$$\partial N_{(id)j}(k)/\partial W^{I}_{(id)ij}(k) = P_{id}(k). \tag{23}$$

Then,

$$\begin{aligned}\partial E_{id}(k)/\partial W^{I}_{(id)ij}(k) &= -\, e_{id}(k)g'_j(N_{(id)j}(k))W^{O}_{(id)j}(k)P_{id}(k)\\ &= -\, e_{id}(k)(1 - (X_{(id)j}(k))^2)W^{O}_{(id)j}(k)P_{id}(k).\end{aligned} \tag{24}$$

3.2. **Dynamic backpropagation for the neurocontroller.** An approach for control using a backpropagation neural network is presented in this section. The inputs to the neurocontroller are chosen as follows: $P_c = \{u_r, u(k-1), y_p(k-1)\}$, where inputs to the neurocontroller are the reference input, the previous output of the neurocontroller, and the previous responses of the APVA. Figure 5 shows a block diagram of the proposed neurocontroller for the APVA plant. The output of the neurocontroller $u(k)$ is the same as the control signal K_p to the plant. The output of the neurocontroller through the training process is used to tune the parameters of K_p. When the response of the nonlinear APVA is altered by K_p values in specific frequencies, the carrier amplitude of the oscillation will vary in the applicable range of K_p values. The design objective of the neural network control system is to regulate K_p values in an APVA, to suppress the vibration of the carrier.

The weights of the neurocontroller are first initialized with small random numbers. The outputs of the neurocontroller are computed by feeding forward the inputs through the network. The error $e_c(k)$ is calculated from the difference between the desired output of the reference model $y_r(k)$ and the responses of the APVA plant $y_p(k)$. The subscript c represents the neural network architecture for the neurocontroller. By using the DBP algorithm developed in this study, the weights of the neurocontroller are adjusted such that the error $e_c(k)$ approaches a small value after some training cycles. An error function for a training cycle for the neurocontroller can be defined as follows:

Let $y_r(k)$ and $y_p(k)$ be the desired and actual responses of the plant; then an error function for a training cycle for the neurocontroller can be defined as

$$E_c(k) = 1/2(y_r(k) - y_p(k))^2. \tag{25}$$

The weights can be adjusted following a gradient method, i.e., the updating rule of the weights becomes

$$W_c(k+1) = W_c(k) - \eta_c(\partial E_c(k)/\partial W_c(k)), \tag{26}$$

where η_c is the learning rate of the neurocontroller.

In the case of the neurocontroller, the gradient of error in Equation (25) with respect to output weight can be represented by

$$\begin{aligned}\partial E_c(k)/\partial W^O_{(c)j}(k) =& \partial E_c(k)/\partial y_p(k).\partial y_p(k)/\partial u(k).\partial u(k)/\partial O_c(k).\\ &\partial O_c(k)/\partial W^O_{(c)j}(k),\end{aligned} \tag{27}$$

where $O_c(k)$ is output of the neurocontroller.

The partial derivatives are given by

$$\partial E_c(k)/\partial y_p(k) = -(y_r(k) - y_p(k)) = -e_c(k), \quad \partial O_c(k)/\partial_{(c)j}(k) = X_{(c)j}(k),$$

where $e_c(k) = y_r(k) - y_p(k)$ is the error between the desired and output responses of the plant. Usually, a linear function is used for the output layer, so $\partial u(k)/\partial O_c(k) = 1$.

The term $y_u(k) \equiv \partial y_p(k)/\partial u(k)$ represents the sensitivity of the plant with respect to its input. Since the plant is normally unknown, the sensitivity term $y_p(k)$ is also unknown. This unknown value can be estimated by using the neuroidentifier. When the neuroidentifier is trained, its dynamic behavior is close to that of the unknown plant, i.e., $y_p(k) \approx y_{id}(k)$, where $y_{id}(k)$ is the output of the neuroidentifier.

Once the training process is done, we assume the sensitivity can be approximated by

$$y_u(k) \equiv \partial y_p(k)/\partial u(k) \approx \partial y_{id}(k)/\partial u(k), \tag{28}$$

where $u(k)$ is an input to the neuroidentifier and the plant.

Apply the chain rule to Equation (28), and note that $y_{id}(k) = O_{(id)}(k)$

$$\begin{aligned}\partial y_{id}(k)/\partial u(k) &= \partial O_{id}(k)/\partial u(k)\\ &= \sum_{j=1}^{n}[\partial(W^O_{(id)j}(k)X_{(id)j}(k))/\partial X_{(id)j}(k).\partial X_{(id)j}(k)/\partial u(k)]\\ &= \sum_{j=1}^{n}[W^O_{(id)j}(k).\partial X_{(id)j}(k)/\partial u(k)].\end{aligned} \tag{29}$$

Also,

$$\partial X_{(id)j}(k)/\partial u(k) = \partial X_{(id)j}(k)/\partial N_{(id)j}(k).\partial N_{(id)j}(k)/\partial u(k) \tag{30}$$

From Equation (22), $\partial N_{(id)j}(k)/\partial u(k)$ becomes $\partial N_{(id)j}(k)/\partial u(k) = W^I_{(id)1j}(k)$.

Thus,

$$\begin{aligned}y_u(k) \approx \partial y_{id}(k)/\partial u(k) &= \sum_{j=1}^{n}[W^O_{(id)j}(k)g'_j(N_{(id)j}(k))W^I_{(id)1j}(k)]\\ &= \sum_{j=1}^{n}[W^O_{(id)j}(k)(1-(X_{(id)j}(k))^2)W^I_{(id)1j}(k)].\end{aligned} \tag{31}$$

Then,

$$\begin{aligned}\partial E_c(k)/\partial W^O_{(c)j}(k) &= -e_c(k)y_u(k)X_{(c)j}(k)\\ &= -e_c(k)\sum_{j=1}^{n}[W^O_{(id)j}(k)(1-(X_{(id)j}(k))^2)W^I_{(id)1j}(k)]X_{(c)j}(k).\end{aligned} \tag{32}$$

In the case of the neurocontroller, the gradient of error in Equation (25) with respect to input weight is represented by

$$\begin{aligned}\partial E_c(k)/\partial W^I_{(c)ij}(k) &= \partial E_c(k)/\partial y_p(k).\partial y_p(k)/\partial W^I_{(c)ij}(k) \qquad (33)\\ &= \partial E_c(k)/\partial y_p(k).\partial y_p(k)/\partial u(k).\partial u(k)/\partial O_c(k).\partial O_c(k)/\partial W^I_{(c)ij}(k).\end{aligned}$$

The output gradients for the neurocontroller with respect to input weights are

$$\begin{aligned}\partial O_c(k)/\partial W^I_{(c)ij}(k) &= W^O_{(c)j}(k).\partial X_{(c)j}(k)/\partial W^I_{(c)ij}(k)\\ &= W^O_{(c)j}(k).\partial X_{(c)j}(k)/\partial N_{(c)j}(k).\partial N_{(c)j}(k)/\partial W^I_{(c)ij}(k)\\ &= W^O_{(c)j}(k).g'_j(N_{(c)j}(k))P_{(c)i}(k)\end{aligned} \tag{34}$$

where

$$g'_j(N_{(c)j}(k)) = 1 - (g_j(N_{(c)j}(k)))^2 = 1 - (X_{(c)j}(k))^2. \tag{35}$$

Then,

$$\begin{aligned}\partial E_c(k)/\partial W^I_{(c)ij}(k) &= -e_c(k)y_u(k)W^O_{(c)j}(k).g'_j(N_{(c)j}(k))P_{(c)i}(k)\\ &= -e_c(k).\sum_{j=1}^{n}[W^O_{(id)j}(k)g'_j(N_{(id)j}(k))W^I_{(id)1j}(k)]W^O_{(c)j}(k)g'_j(N_{(c)j}(k))P_{(c)i}(k).\end{aligned} \tag{36}$$

3.3. **Convergence and stability.** The updating rule for Equations (14) and (26) calls for a proper choice of learning rate η. For a small value of η, the convergence is guaranteed but the speed is very slow; on the other hand, if η is too big, the algorithm becomes unstable. This section develops a guideline for selecting the proper learning rate, which leads to an adaptive learning rate.

A discrete-type Lyapunov function can be given by

$$V(K) = 1/2[e^2(k)], \tag{37}$$

where $e(k)$ represents the error in the learning process.

Thus, the change of the Lyapunov function due to the training process is obtained by

$$\Delta V(k) = V(k+1) - V(k) = 1/2[e^2(k+1) - e^2(k)] \tag{38}$$

The error difference due to the learning can be represented by [25]

$$e(k+1) = e(k) + \Delta e(k) = e(k) + [\partial e(k)/\partial W]^T \Delta W, \tag{39}$$

where ΔW represents a change in an arbitrary weight vector. As shown in the Appendix, the learning rates η^O_{id} for the neuroidentifier weights W^O_{id} and η^I_{id} for the neuroidentifier weights W^I_{id} are chosen to be $0 < \eta^O_{id} < 2/h_{id}$, the optimal convergence rate $(\eta^O_{id})^* = 1/h_{id}$ and $0 < \eta^I_{id} < 2/(n_{id}.h_{id})[1/W^O_{id,\max}\ P_{id,\max}]^2$, the optimal convergence rate $(\eta^I_{id})^* = 1/(n_{id}.h_{id})[1/W^O_{id,\max}\ P_{id,\max}]^2$, respectively. The η^O_c and η^I_c be the learning rates for the neurocontroller weights W^O_c and W^I_c, respectively. The learning rates are chosen to be $0 < \eta^O_c < 2/(h_c S^2_{\max})$, the optimal convergence rate $(\eta^O_c)^* = 1/(S^2_{\max}h_c)$ and $0 < \eta^I_c < 2/(n_c.h_c.S^2_{\max})[1/W^O_{c,\max}\ P_{c,\max}]^2$, the optimal convergence rate $(\eta^I_c)^* = 1/(n_c.h_c.S^2_{\max})[1/W^O_{c,\max}\ P_{c,\max}]^2$.

3.4. **Response of the nonlinear APVA altering K_p values at specific frequencies.** Figure 6 shows the response curves for the APVA with varying K_p. From the curves for three specific excitation frequencies, 225 rad/sec, 260 rad/sec and 300 rad/sec, respectively, we can observe that at varying K_p, the carrier amplitude of the oscillation will alter within the range of $K_p = [-12, 15]$. It is found that at point A' the vibration amplitude of the carrier approaches zero, and at point B' the vibration amplitude of the carrier reaches a maximum.

Looking at the curves to the right of point B', we can observe that the curves never reach the minimum. In this study, we use a means that allows them to escape such a situation. From the curves to the right of point B', it can be seen that if the conditions $|\Delta y_p| < \varepsilon$ (i.e., $\varepsilon = 2.28 \times 10^{-5}$) and $y_p(k) > \gamma$ (i.e., $\gamma = 0.0024$) exist, then the reference input u_r and the input and output weights of the neurocontroller will be randomly rechosen to repeat the training session in order to escape the region of the curves to the right of point B'.

4. **Simulation Results.** The objective in this paper is to apply the back-propagation neural network controller, which has developed both the neurocontroller and neuroidentifier, and to set up adequate output variables that will control the vibration of the APVA at a minimum. In a neural network based control system, the plant is unknown in general; thus the sensitivity, which is required during training process, is normally not available. Some papers simply ignore this sensitivity and use the direct control approach. In [19], the chain rule is used to calculate a recurrence relationship in which the sensitivity of the preceding layer is computed from the sensitivity at the present layer with non-neuroidentifier. In this paper, a neuroidentifier is utilized to obtain the sensitivity, and, along with the neurocontroller, obtain the weight adjustment as well. The training parameters of the adaptive learning rates are also depicted in this section. In the simulation, $P_c = \{u_r, u(k-1), y_p(k-1)\}$ are the inputs to the neurocontroller, where u_r is the reference input to randomly generated between the range of $[-3, 9]$, $u(k-1)$ is the previous output of the neurocontroller (initial $u(0) = 0$) and $y_p(k-1)$ is the previous vibration amplitude of the carrier of the APVA (initial $y_p(0) = 0$). $P_{id} = \{u(k), y_p(k-1)\}$ are the inputs to the neuroidentifier where $u(k)$ is the output of the neurocontroller.

In the case of the neurocontroller, the input weight $W^I_{(c)}$ is a 3 by 20 matrix in which every element in the matrix is randomly generated in the range of $[0, 1]$. The output weight $W^O_{(c)}$ is a 1 by 20 matrix for which every element in the matrix is randomly generated in the range of $[0, 0.5]$. In the case of the neuroidentifier, the input weight $W^I_{(id)}$ is a 2

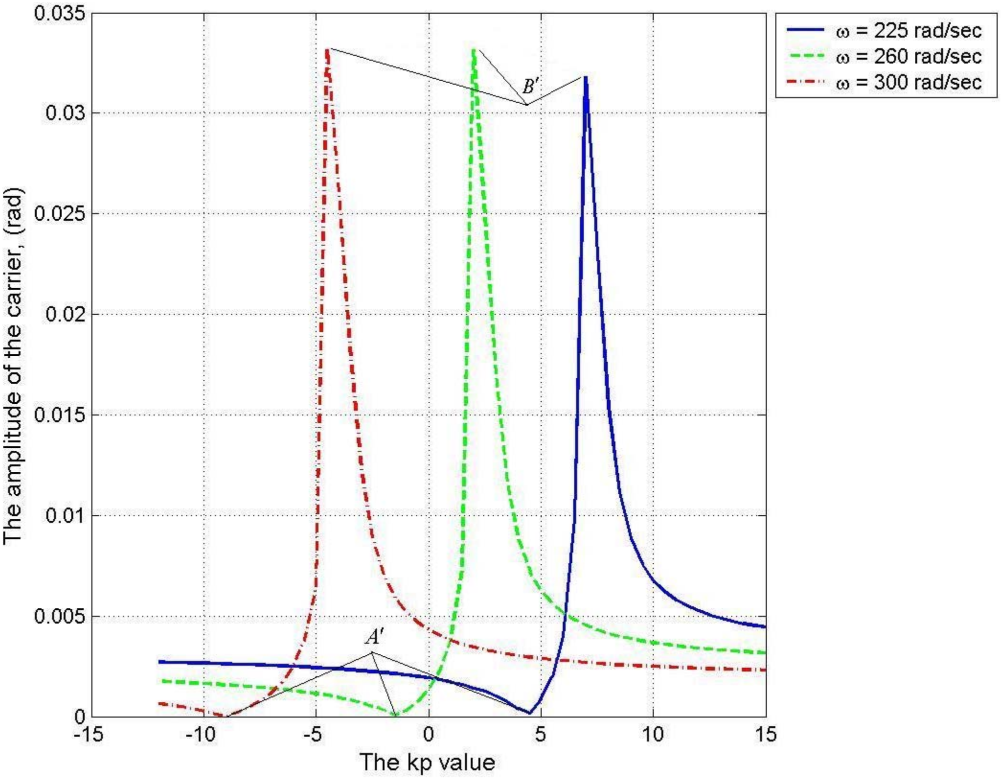

Figure 6. Response of the APVA with varying K_p

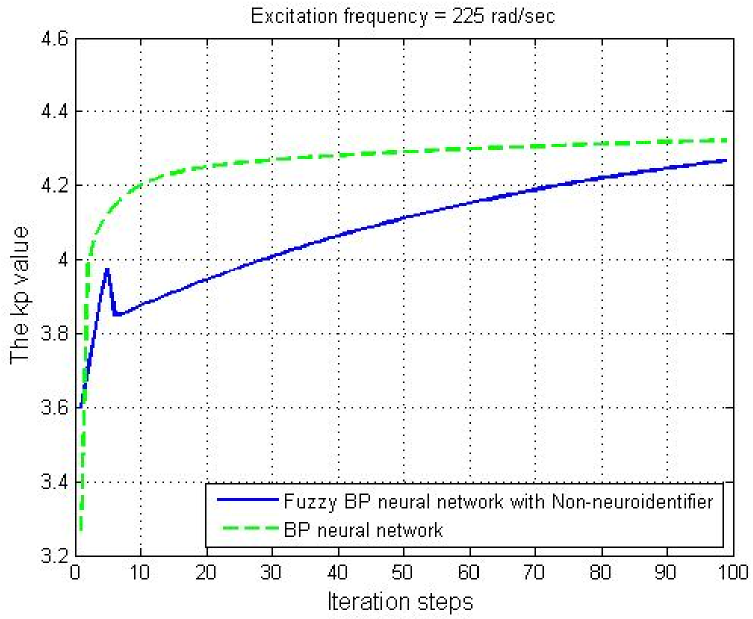

Figure 7. K_p values for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm

by 20 matrix in which every element in the matrix is randomly generated in the range of $[0, 0.01]$. The output weight $W^O_{(id)}$ is a 1 by 20 matrix in which every element in the matrix is randomly generated in the range of $[0, 0.01]$. The numbers of inputs to the neurocontroller and neuroidentifier are denoted by n_c and n_{id}, respectively, while h_c and h_{id} denote the numbers of neurons in the hidden layer for the neurocontroller and neuroidentifier, respectively; they are chosen to be $n_c = 3$, $n_{id} = 2$, $h_c = 20$, $h_{id} = 20$. The desired output of the reference model $y_r(k)$ is 0. The weights of the neurocontroller are adjusted such that the error $e_c(k)$ approaches a small value after some training cycles. In this section, the simulation results for the BP neural network algorithm APVA are compared with the fuzzy BP neural network algorithm with non-neuroidentifier. The fuzzy BP neural network controller systems with non-neuroidentifier architectures can be viewed as conventional fuzzy algorithms for coarse tuning and the BP algorithm is applied for fine tuning. The input and output range for the fuzzy algorithm are chosen from the specific intervals, and the membership functions in the fuzzification interface are chosen to be Gaussian membership functions. The BP neural network with non-neuroidentifier is used to fine-tune the system performance [19].

The K_p values for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm, with an excitation frequency of $\omega = 225$ rad/sec, are shown in Figure 7.

The vibration amplitude of the carrier for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm with an excitation frequency of $\omega = 225$ rad/sec are shown in Figure 8. As can be seen in the figure, the BP neural network converges faster than the fuzzy BP neural network with non-neuroidentifier algorithm. The disturbance excitation frequency is changed to 260 rad/sec, and the simulation results for the BP neural network are compared with the fuzzy BP neural network with non-neuroidentifier algorithm. The K_p values for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm, with an excitation frequency of $\omega = 260$ rad/sec, are shown in Figure 9.

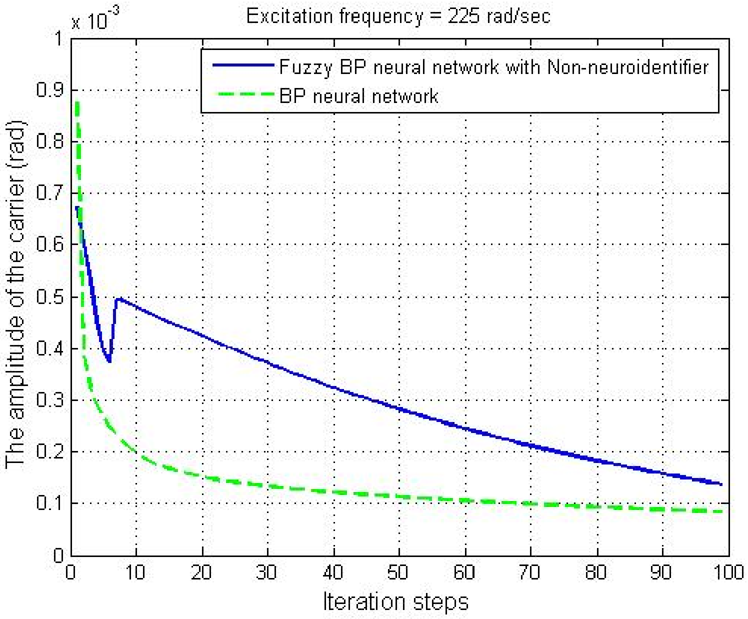

Figure 8. Vibration amplitude of the carrier for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm

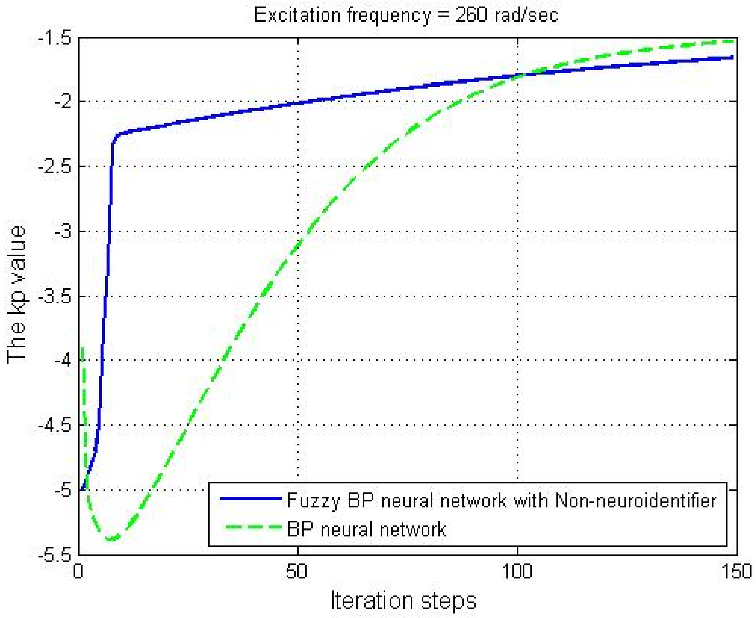

Figure 9. K_p value for the BP neural network algorithm and the fuzzy BP neural network algorithm with non-neuroidentifier

Figure 10 shows the carrier vibration of the APVA for the BP neural network algorithm and the fuzzy BP neural network with non-neuroidentifier algorithm with an excitation frequency of $\omega = 260$ rad/sec. As can be seen in the figure, the BP neural network algorithm converges faster than the fuzzy BP neural network algorithm with non-neuroidentifier.

From the fuzzy BP neural network with non-neuroidentifier algorithm, the output variable K_p for the fuzzy system in the specific interval needs more fuzzy rules to obtain precise output. The BP training algorithm of the fuzzy BP neural network with fixed

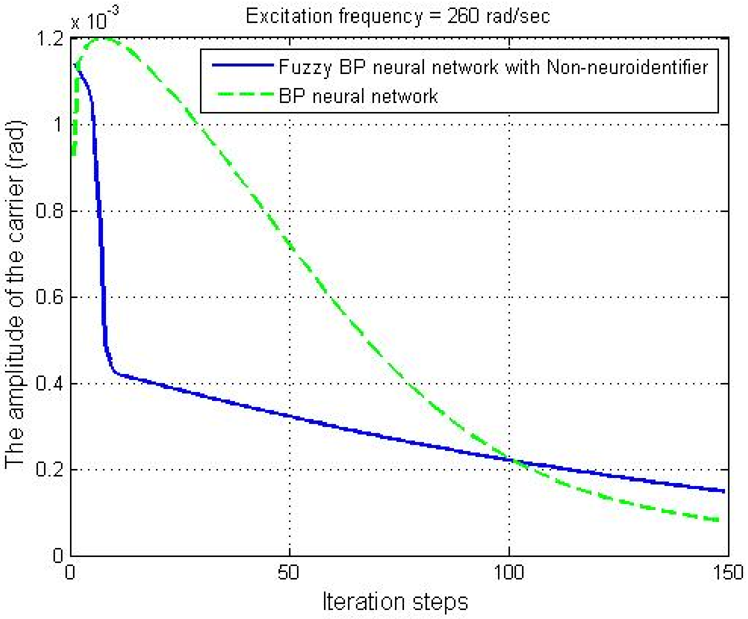

Figure 10. Vibration amplitude of the carrier for the BP neural network algorithm and the fuzzy BP neural network algorithm with non-neuroidentifier

learning rate tends to be inefficient. A large learning rate may make the system unstable while a small learning rate makes the training process too slow. However, in this paper, a neuroidentifier is utilized to obtain the sensitivity, in addition to the neurocontroller. The weight adjustment becomes smoother than the case without the sensitivity information. In this study, an approach to find the bounds on learning rates based on the Lyapunov function is developed. The use of adaptive learning rates guarantees convergence so the optimal learning rates are found. From the simulation results, we can observe that the absorbing efficiency is better with the adaptive learning rates of the BP neural network APVA than for the fuzzy BP neural network algorithm with non-neuroidentifier.

5. **Conclusions.** This study proposes the BP neural network algorithm, which includes the neurocontroller and neuroidentifier, for the control of an APVA. A theoretical dynamic model of the APVA was developed, and simulations were performed to verify the capability of the proposed control algorithms. The simulation results show that when the frequency of the disturbance fluctuates within a certain range, the control strategy can effectively suppress the vibrational amplitude of the carrier. In the BP neural network algorithm, the use of adaptive learning rates guarantees convergence so the optimal learning rates are found. From the simulation results, we can observe that the absorbing efficiency is better with the adaptive learning rates of the BP neural network APVA than for the fuzzy BP neural network algorithm with non-neuroidentifier. Further research should also be carried out on other aspects of the APVA. The nonlinear characteristics of the system must be considered in the design of the APVA. If ignored, the absorber's effectiveness could be reduced, perhaps even resulting in large oscillatory amplitudes in the carrier system. However, certain disturbances may lead to a chaotic response. If the behavior of a system can be correctly predicted, its operation will be rendered both safer and more reliable.

Acknowledgment. This work was supported by the National Science Council in Taiwan, under Project Number NSC 96-2628-E-008-075-MY3. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] W. T. Thomson, *Theory of Vibration with Applications*, 2nd Edition, Englewood Cliffs, Prentice-Hall, New Jersey, 1981.

[2] G. Genta, *Vibration Dynamics and Control*, Springer Science Business Media, New York, 2009.

[3] K. Wilson, *Practical Solution of Torsional Vibration Problems*, 3rd Edition, Chapman & Hall Ltd, London, 1968.

[4] E. S. Taylor, Eliminating crack shaft vibration in radial aircraft engines, *Transactions of the Society of Aeronautical Engineers*, vol.38, pp.81-87, 1936.

[5] J. P. Den Hartog, *Mechanical Vibrations*, 4th Edition, McGraw-Hill Book Company, Inc., New York, 1956.

[6] D. E. Newland, Nonlinear aspects of the performance of centrifugal pendulum vibration absorbers, *Journal Engineering for Industry, Transaction of the ASME*, vol.86, pp.257-263, 1964.

[7] M. Sharif-Bakhtiar and S. W. Shaw, The dynamic response of a centrifugal pendulum vibration absorber with motion-limiting stops, *Journal of Sound and Vibration*, vol.126, pp.221-235, 1988.

[8] M. Sharif-Bakhtiar and S. W. Shaw, Effects of nonlinearities and damping on the dynamic response of a centrifugal pendulum vibration absorber, *Journal of Vibration and Acoustics, Transaction of the ASME*, vol.114, pp.305-311, 1992.

[9] D. L. Cronin, Shake reduction in an automobile engine by means of crankshaft-mounted pendulums, *Mechanism and Machine Theory*, vol.27, no.5, pp.517-533, 1992.

[10] B. Demeulenaere, P. Spaepen and J. D. Schtter, Input torque balancing using a cam-based centrifugal pendulum: design procedure and example, *Journal of Sound and Vibration*, vol.283, pp.1-20, 2005.

[11] A. G. Haddow and S. W. Shaw, Centrifugal pendulum vibration absorber: An experimental and theoretical investigation, *Nonlinear Dynamics*, vol.34, pp.293-307, 2003.

[12] Y. J. Wang, C. D. Chen and C. K. Sung, Design of a frequency-adjusting device for harvesting energy from a rotating wheel, *Sensors and Actuators A: Physical*, vol.159, pp.196-203, 2010.

[13] S. G. Tewani, B. L. Walcott and K. E. Rouch, Active optimal vibration control using dynamic absorber, *IEEE International Conference on Robotics and Automation*, pp.1182-1187, 1991.

[14] Y. D. Chen, C. C. Fuh and P. C. Tung, Application of voice coil motors in active dynamic vibration absorbers, *IEEE Transactions on Magnetics*, vol.41, no.3, pp.1149-1154, 2005.

[15] M. Hosek, H. Elmali and N. Olgac, A tunable torsional vibration absorber: the centrifugal delayed resonator, *Journal of Sound and Vibration*, vol.205, pp.151-165, 1997.

[16] M. Hosek, N. Olgac and H. Elmali, Torsional vibration control of MDOF systems using the centrifugal delayed resonator, *IEEE International Conference on Control Applications*, pp.534-539, 1997.

[17] S. T. Wu, Active pendulum vibration absorbers with a spinning support, *Journal of Sound and Vibration*, vol.323, pp.1-16, 2009.

[18] C. H. Liang and P. C. Tung, Application of genetic algorithms to active vibration control of a centrifugal pendulum vibration absorber, *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol.224, no.4, pp.329-338, 2010.

[19] C. H. Liang and P. C. Tung, A fuzzy neural network for the active vibration control of a centrifugal pendulum vibration absorber, *International Journal of Modern Physics C*, vol.20, no.12, pp.1963-1979, 2009.

[20] H. Tamura, T. Gotoh, D. Okumura, H. Tanaka and K. Tanno, A study on the S-EMG pattern recognition using neural network, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4877-4884, 2009.

[21] B.-M. Chang, H.-H. Tsai and P.-T. Yu, The Dempster-Shafer theory combined with neural network in hand written character recognition, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2561-2573, 2009.

[22] Y. Mu and A. Sheng, Evolutionary diagonal recurrent neural network with improved hybrid ep-pso algorithm and its identification application, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp.1615-1624, 2009.

[23] F. Takeda, Y. Shiraishi and T. Sanechika, Alarm sound classification system of oxygen concentrator by using neural network, *International Journal of Innovative Computing, Information and Control*, vol.3, no.1, pp.211-222, 2007.

[24] C. C. Ku and K. Y. Lee, Diagonal recurrent neural networks for dynamic systems control, *IEEE Transactions on Neural Networks*, vol.6, no.1, pp.144-155, 1995.

[25] T. Yabuta and T. Yamada, Learning control using neural networks, *Proc. of the 1991 IEEE International Conference on Robotics and Automation Sacramento*, CA, USA, pp.740-745, 1991.

Appendix.

Convergence of the neuroidentifier. From the updating rule in Equations (14) and (15),

$$\Delta W_{id} = -\eta_{id}\partial E_{id}(k)/\partial W_{id} = \eta_{id} e_{id}(k)\partial O_{id}(k)/\partial W_{id}, \tag{A1}$$

where W_{id} and η_{id} respectively represent an arbitrary weight and the corresponding learning rate in the neuroidentifier, and $O_{id}(k)$ is the output of the neuroidentifier. Now we have the following general convergence theorem:

Theorem 1. *Let η_{id} be the learning rate for the weights of the neuroidentifier and let $G_{id,\max}$ be defined as $G_{id,\max}$: $\max_k ||G_{id}(k)||$, where $G_{id}(k) = \partial O_{id}(k)/\partial W$, and $||.||$ is the usual Euclidean norm. The convergence is guaranteed if η_{id} is chosen to be*

$$0 < \eta_{id} < 2/G^2_{id,\max}. \tag{A2}$$

Proof: From Equations (38) and (A1), $\Delta V(k)$ can be represented as

$$\begin{aligned}\Delta V(k) &= 1/2[e_{id}(k) + \Delta e^2_{id}(k) - e^2_{id}(k)] = \Delta e_{id}(k)[e_{id}(k) + 1/2\Delta e_{id}(k)]\\ &= [\partial e_{id}(k)/\partial W_{id}]^T \eta_{id} e_{id}(k).\partial O_{id}(k)/\partial W_{id}\{e_{id}(k)\\ &\quad + 1/2[\partial e_{id}(k)/\partial W_{id}]^T \eta_{id} e_{id}(k)\partial O_{id}(k)/\partial W_{id}\}.\end{aligned} \tag{A3}$$

Since for neuroidentifier $\partial e_{id}(k)/\partial W_{id} = -\partial y_{id}(k)/\partial W_{id} = -\partial O_{id}(k)/\partial W_{id}$, we obtain

$$\begin{aligned}\Delta V(k) &= -\eta_{id} e^2_{id}(k)||\partial O_{id}(k)/\partial W_{id}||^2 + 1/2\eta^2_{id} e^2_{id}(k)||\partial O_{id}(k)/\partial W_{id}||^4\\ &= -\lambda_{id} e^2_{id}(k),\end{aligned} \tag{A4}$$

where $\lambda_{id} = \eta_{id}||\partial O_{id}(k)/\partial W_{id}||^2 - 1/2\eta^2_{id}||\partial O_{id}(k)/\partial W_{id}||^4$.

Let $G_{id}(k) = \partial O_{id}(k)/\partial W_{id}$, $G_{id,\max}$: $\max_k||G_{id}(k)||$, and $\eta_1 = \eta_{id} G^2_{id,\max}$. Then,

$$\begin{aligned}\lambda_{id} &= 1/2||G_{id}(k)||^2\eta_{id}(2 - \eta_{id}||G_{id}(k)||^2)\\ &= 1/2||G_{id}(k)||^2\eta_{id}(2 - \eta_1||G_{id}(k)||^2/G^2_{id,\max})\\ &\geq 1/2||G_{id}(k)||^2\eta_{id}(2 - \eta_1) > 0.\end{aligned} \tag{A5}$$

From Equation (A5), $\eta_{id}(2 - \eta_1) > 0$, we obtain $0 < \eta_1 < 2 => 0 < \eta_{id} G^2_{id,\max} < 2 => 0 < \eta_{id} < 2/G^2_{id,\max}$.

Then the convergence is guaranteed if η_{id} is chosen to be $0 < \eta_{id} < 2/G^2_{id,\max}$.

The convergence is guaranteed as long as Equation (A5) is satisfied, i.e., $\eta_{id}(2 - \eta_1) > 0$ or $\eta_1(2 - \eta_1)/G^2_{id,\max} > 0$.

This implies that any η_1, $0 < \eta_1 < 2$, guarantees the convergence. However, from $\eta_1(2 - \eta_1)$, let the maximum learning rate η^*_{id} be that which guarantees the most rapid or optimal convergence corresponding to $\eta_1 = 1$, i.e.,

$$\eta^*_{id} = 1/G^2_{id,\max}, \tag{A6}$$

which is half of the upper limit in Equation (A2).

This shows an interesting result, namely that any other learning rate larger than η^*_{id} does not guarantee faster convergence.

The general convergence theorem can be applied to find the specific convergence criterion for each type of weight.

Theorem 2. *Let η_{id}^O and η_{id}^I be the learning rates for the neuroidentifier weights W_{id}^O and W_{id}^I, respectively. The learning rates are chosen to be*

$$0 < \eta_{id}^O < 2/h_{id}, \tag{A7}$$

$$0 < \eta_{id}^I < 2/(n_{id}.h_{id})[1/W_{id,\max}^O.P_{id,\max}]^2 \tag{A8}$$

where h_{id} is the number of neurons in the hidden layer, n_{id} is the number of inputs to the neuroidentifier, $W_{id,max}^O = \max_k||W_{id}^O(k)||$, $P_{id,max} = \max_k||P_{id}(k)||$, and $||.||$ is the sup-norm, i.e., $||W_{id}^O(k)|| = \max_j|W_{(id)j}^O(k)|$, $||P_{id}(k)|| = \max_i|P_{(id)i}(k)|$.

(1) **Proof of Equation (A7):** let

$$G_{id} = \partial O_{id}(k)/\partial W_{id}^O = X_{(id)j}(k), \tag{A9}$$

where $X_{id} = [X_{(id)1}, X_{(id)2}, \ldots X_{(id)h}]^T$, $X_{(id)j}$ is the output value of the jth neuron in the hidden layer, and h_{id} is the number of neurons in the neuroidentifier hidden layer.

Since $-1 < X_{(id)j}(k) < 1$, $j = 1, 2, \ldots, h_{id}$, by definition of the usual Euclidean norm, $||G_{id}(k)|| = (X_{(id)1}^2 + X_{(id)2}^2 + \ldots X_{(id)h}^2)^{1/2} < (h_{id})^{1/2}$ and $G_{id,\max}^2(k) = h_{id}$. Then from Theorem 1 Equation (A2) follows.

$$0 < \eta_{id}^O < 2/G_{id,\max}^2 = 2/h_{id}. \tag{A10}$$

Therefore, the proof of Equation (A7) is completed and $(\eta_{id}^O)^* = 1/h_{id}$.

(2) **Proof of Equation (A8):** from Equation (20),

$$\begin{aligned}\partial X_{(id)j}(k)/\partial W_{(id)ij}^I(k) &= \partial X_{(id)j}(k)/\partial N_{(id)j}(k).\partial N_{(id)j}(k)/\partial W_{(id)ij}^I(k)\\ &= g_j'(N_{(id)j}(k))P_{id}(k).\end{aligned} \tag{A11}$$

Since $0 < g_j'(N_{(id)j}(k)) \le 1$, from Equation (21), $g_j'(N_{(id)j}(k)) = 1 - (X_{(id)j}(k))^2$, $-1 < X_{(id)j}(k) < 1$ for $j = 1, 2, \ldots, h_{id}$, we can obtain

$$|\partial X_{(id)j}(k)/\partial W_{(id)ij}^I(k)| \le P_{id}(k). \tag{A12}$$

Let $||P_{id}(k)|| = \max_j |P_{id}(k)|$, $P_{id,\max} = \max_k ||P_{id}(k)||$, thus we obtain

$$||\partial X_{id}(k)/\partial W_{id}^I|| \le (n_{id}.h_{id})^{1/2} P_{id,\max}, \tag{A13}$$

where h_{id} is the number of neurons in the hidden layer, and n_{id} is the number of inputs to the neuroidentifier. From Equation (20),

$$\partial O_{id}(k)/\partial W_{(id)ij}^I(k) = W_{(id)j}^O(k).\partial X_{(id)j}(k)/\partial W_{(id)ij}^I(k),$$

thus we obtain

$$||\partial O_{id}(k)/\partial W_{id}^I|| \le (n_{id}.h_{id})^{1/2} P_{id,\max}. \max_k||W_{id}^O(k)|| = (n_{id}.h_{id})^{1/2} P_{id,\max} W_{id,\max}^O,$$

where

$$||W_{id}^O(k)|| = \max_j|W_{(id)j}^O(k)|, \quad W_{id,\max}^O = \max_k||W_{id}^O(k)||.$$

Let $G_{id} = \partial O_{id}(k)/\partial W_{id}^I$, $||G_{id}|| \le (n_{id}.h_{id})^{1/2} P_{id,\max}\ W_{id,\max}^O$ and $G_{id,\max}^2 = n_{id}.h_{id}\ P_{id,\max}^2 (W_{id,\max}^O)^2$, then from Equation (A2),

$$0 < \eta_{id}^I < 2/G_{id,\max}^2 = 2/(n_{id}.h_{id})[1/W_{id,\max}^O.P_{id,\max}]^2. \tag{A14}$$

Therefore, the proof of Equation (A8) is completed and $(\eta_{id}^I)^* = 1/(n_{id}.h_{id})[1/W_{id,\max}^O.P_{id,\max}]^2$.

Convergence of the neurocontroller. From the updating rule for Equations (26) and (27),

$$\Delta W_c = -\eta_c \partial E_c(k)/\partial W_c = \eta_c e_c(k).\partial y_p(k)/\partial u(k).\partial O_c(k)/\partial W_c, \tag{A15}$$

where $\partial y_p(k)/\partial u(k) = y_u(k)$, and W_c and η_c respectively represent an arbitrary weight and the corresponding learning rate in the neurocontroller, and $O_c(k)$ is the output of the neurocontroller. Then we have the following general convergence theorem:

Theorem 3. *Let η_c be the learning rate for the weights of the neurocontroller, and $G_{c,\max}$ be defined as $G_{c,\max} : \max_k ||G_c(k)||$, where $G_c(k) = \partial O_c(k)/\partial W_c$, and $S_{\max} = h_{id}\ W^O_{id,\max} W^I_{id,1\max}$, and $||.||$ is the usual Euclidean norm. The convergence is guaranteed if η_c is chosen as follows:*

$$0 < \eta_c < 2/(S^2_{\max} G^2_{c,\max}). \tag{A16}$$

Proof: From Equations (38), (39) and (A15), $\Delta V(k)$ can be represented as

$$\begin{aligned}\Delta V(k) &= 1/2[e_c(k) + \Delta e^2_c(k) - e^2_c(k)]\\ &= \Delta e_c(k)[e_c(k) + 1/2\Delta e_c(k)]\\ &= [\partial e_c(k)/\partial W_c]^T \eta_c e_c(k).\partial y_p(k)/\partial u(k).\partial O_c(k)/\partial W_c\{e_c(k) + 1/2[\partial e_c(k)/\partial W_c]^T\\ &\quad \eta_c e_c(k).\partial y_p(k)/\partial u(k).\partial O_c(k)/\partial W_c\}.\end{aligned} \tag{A17}$$

Since for the neurocontroller

$$\partial e_c(k)/\partial W_c = -\partial y_p(k)/\partial u(k).\partial O_c(k)/\partial W_c = -y_u.\partial O_c(k)/\partial W_c,$$

we obtain

$$\begin{aligned}\Delta V(k) &= -\eta_c e^2_c(k) y^2_u ||\partial O_c(k)/\partial W_c||^2 + 1/2\eta^2_c e^2_c(k) y^4_u ||\partial O_c(k)/\partial W_c||^4\\ &= -\lambda_c e^2_c(k),\end{aligned} \tag{A18}$$

where

$$\lambda_c = \eta_c y^2_u ||\partial O_c(k)/\partial W_c||^2 - 1/2\eta^2_c y^4_u ||\partial O_c(k)/\partial W_c||^4. \tag{A19}$$

Comparing Equation (A18) with Equation (A4), it can be seen that both conditions are similar, except that $y_u(k)$ needs to be incorporated in the neurocontroller. Therefore, the limit on $y_u(k)$ or $y^2_u(k)$ remains to be found.

Since from Equation (31),

$$y_u(k) = \sum_{j=1}^{n} [W^O_{(id)j}(k) g'_j(N_{(id)j}(k)) W^I_{(id)1j}(k)], \tag{A20}$$

where $0 < g'_j(N_{(id)j}(k)) \leq 1$. If $W^I_{(id)1\max}$ is defined as $W^I_{(id)1\max} = \max_k ||W^I_{(id)1}(k)||$ and $||W^I_{(id)1}(k)|| = \max_j |W^I_{(id)j}(k)|$, then

$$|y_u(k)| \leq h_{id} ||W^O_{(id)j}(k)|| \, || g'(N_j(k)) W^I_{(id)1j}(k)|| \leq h_{id} W^O_{(id)\max} W^I_{(id)1\max}(k) \equiv S_{\max}, \tag{A21}$$

where $S_{\max}$ is the limit on sensitivity.

Let $G_c(k) = \partial O_c(k)/\partial W_c$, $G_{c,\max} : \max_k ||G_c(k)||$, and $\eta_1 = \eta_c G^2_{c,\max}$. Then, Equation (A19)

$$\begin{aligned}\lambda_c &= \eta_c S^2_{\max} ||G_c(k)||^2 - 1/2\eta^2_c S^4_{\max} ||G_c(k)||^4\\ &= 1/2||G_c(k)||^2 \eta_c S^2_{\max}(2 - \eta_1 S^2_{\max} ||G_c(k)||^2 / G^2_{c,\max})\\ &\geq 1/2||G_c(k)||^2 \eta_c S^2_{\max}(2 - \eta_1 S^2_{\max}) > 0.\end{aligned} \tag{A22}$$

From Equation (A22), $\eta_c(2 - \eta_1 S^2_{\max})$ >0 and we obtain $0 < \eta_1 S^2_{\max} < 2 => 0 < \eta_c S^2_{\max} G^2_{c,\max} < 2 => 0 < \eta_c < 2/S^2_{\max} G^2_{c,\max}$.

Then, the convergence is guaranteed if η_c is chosen to be $0 < \eta_c < 2/S^2_{\max}G^2_{c,\max}$.

As in the case of the neurocontroller, the optimal convergence rate is

$$\eta^*_c = 1/S^2_{\max}G^2_{c,\max},$$

which is half of the upper limit in Equation (A16). Again, any other learning rate larger than η^*_c does not guarantee faster convergence.

Thus, in a way similar to the neuroidentifier, the specific convergence criteria can be found as follows:

Theorem 4. Let η^O_c and η^I_c be the learning rates for the neurocontroller weights W^O_c and W^I_c, respectively. The learning rates are chosen to be

$$0 < \eta^O_c < 2/h_c S^2_{\max}, \tag{A23}$$

$$0 < \eta^I_c < 2/(n_c.h_c)S^2_{\max}[1/W^O_{c,\max}.P_{c,\max}]^2, \tag{A24}$$

where h_c is the number of neurons in the hidden layer, n_c is the number of inputs to the neurocontroller, $W^O_{c,\max} = \max_k ||W^O_c(k)||$, $P_{c,\max}= \max_k||P_c(k)||$, and $||.||$ is the sup-norm, i.e., $||W^O_c(k)|| = \max_j|W^O_{(c)j}(k)|$, $||P_c(k)|| = \max_i|P_{(c)i}(k)|$ and $P_c = \{u_r, u(k-1), y_p(k-1)\}$.

(1) **Proof of Equation (A23):** let $G_c = \partial O_c(k)/\partial W^O_c = X_{(c)j}(k)$, where $X_c = [X_{(c)1}, X_{(c)2}, \ldots X_{(c)h}]^T$, $X_{(c)j}$ is the output value of the jth neuron in the hidden layer, and h_c is the number of neurons in the neurocontroller's hidden layer.

Since $-1 < X_{(c)j}(k) < 1$, $j = 1, 2, \ldots, h_c$, by definition of the usual Euclidean norm, $||G_c(k)|| = (X^2_{(c)1} + X^2_{(c)2} + \ldots X^2_{(c)h})^{1/2} < (h_c)^{1/2}$ and $G^2_{c,\max}(k) = h_c$. Then, from Theorem 4 Equation (A23) follows. $0 < \eta^O_c < 2/S^2_{\max}G^2_{c,\max} = 2/S^2_{\max}h_c$.

Therefore, the proof of Equation (A23) is completed and $(\eta^O_c)^* = 1/S^2_{\max}h_c$.

(2) **Proof of Equation (A24):** from Equation (34)

$$\partial O_{(c)}(k)/\partial W^I_{(c)ij} = W^O_{(c)j}(k)\partial X_{(c)j}(k)/\partial W^I_{(c)ij} = W^O_{(c)j}(k)g'_{(c)j}(N_{(c)j}(k))P_{(c)i}(k). \tag{A25}$$

Since $0 < g'_{(c)j}(N_{(c)j}(k)) \leq 1$, from $g'_{(c)j}(N_{(c)j}(k)) = 1 - (X_{(c)j}(k))^2$, $-1 < X_{(c)j}(k) < 1$ for $j = 1, 2, \ldots, h_c$, $||P_c(k)|| = \max_j |P_c(k)|$, $P_{c,\max} = \max_k||P_c(k)||$, $||W^O_c(k)|| = \max_j |W^O_{(c)j}(k)|$, $W^O_{c,\max} = \max_k ||W^O_c(k)||$.

Thus, we obtain

$$||\partial O_c(k)/\partial W^I_c|| \leq (n_c.h_c)^{1/2}\max_k||P_c(k)||\max_k||W^O_c(k)|| \leq (n_c.h_c)^{1/2}P_{c,\max}W^O_{c,\max}, \tag{A26}$$

where h_c is the number of neurons in the hidden layer, and n_c is the number of inputs to the neurocontroller.

Let

$$G_c = \partial O_c(k)/\partial W^I_c,\ ||G_c|| \leq (n_c.h_c)^{1/2}P_{c,\max}W^O_{c,\max} \text{ and } G^2_{c,\max} = n_c.h_cP^2_{c,\max}(W^O_{c,\max})^2,$$

then from Equation (A16),

$$0 < \eta^I_c < 2/S^2_{\max}G^2_{c,\max} = 2/(n_c.h_cS^2_{\max})[1/W^O_{c,\max}.P_{c,\max}]^2. \tag{A27}$$

Therefore, the proof of Equation (A24) is completed and

$$(\eta^I_c)^* = 1/(n_c.h_cS^2_{\max})[1/W^O_{c,\max}.P_{c,\max}]^2. \tag{A28}$$