Genetic algorithm based wavelet neural network speed controller for spindle motor of DVD-ROM

Chin-Hsing Cheng *

Department of Electrical Engineering
Feng Chia University
No. 100, Wen Hwa Rd.
Taichung 407
Taiwan
R.O.C.

Po-Jen Cheng

Department of Electrical Engineering
Nan Jeon Institute of Technology
No. 178, Chau-Chin Rd.
Yen-Shui, Tainan Hsien 737
Taiwan
R.O.C.

Abstract

This paper presents Wavelet Neural Network (WNN) constructed of general neural network employing the wavelet function as the activation function and genetic algorithm based instance and feature selection for setting the initial values of network's parameters to design spindle motor speed controller for DVD-ROM. With the advantages of global search abilities of genetic algorithm and the ability of multiresolution analysis of wavelet theory, the wavelet neural network with genetic algorithm has much factor convergence speed and can be used for controlling the high speed motor. A wavelet neural network controller with genetic algorithm used for controlling a three-phase, nine-slot, twelve-pole spindle motor of DVD-ROM drive has been designed. The experimental results demonstrated the feasibility and effectiveness of the proposed scheme.

Keywords: Genetic algorithm, wavelet neural network, DVD-ROM.

*E-mail: chchang@fcu.edu.tw

Journal of Information & Optimization Sciences
Vol. 27 (2006), No. 3, pp. 551–563
© Taru Publications 0252-2667/06 $2.00 + 0.25
I. Introduction

In real life, still, conventional controllers are widespread. Especially, PID and PID like controllers are very popular. Whereas, these-kind controllers are always in need of tuning process manually due to their being inadaptable. Besides, if the PID controller is unable to deal with the complex process, no matter how tuned it, it will not work properly. Therefore, intelligent-like controllers are compulsory and worth of introducing and taking their turns. Among them, self-tuning and other well-known adaptive controllers are already used in robotics, factories and other complex systems. In recent years, Artificial Neural Network (ANN) and fuzzy controllers have jumped into the real field and proved their dexterity of manu respects.[23]

Neural Networks (NNs) are a promising new generation of information processing systems that demonstrate the ability to learn from training data [1]. The models known as Artificial Neural Networks (ANN) inspired from the morphology of biological neural system and organization of brain structures and the attempt to emulate the human-like performance [2]. One issue of great importance in the field of ANN is that of input data pre-processing. From simple operations, some degree of preprocessing usually speeds the training process and improves the network performance [3]. Among many ANN’s models, the Multilayer Perceptron (MLP) is the widely used. Backpropagation (BP) algorithm is used to train MLP. However, he BP networks with sigmoidal nonlinearity have many open drawbacks, such as the unreliable choice of network dimension or architecture, local minimum. This is mostly because the translation and dilation of sigmoid function is not orthogonal, the correlation redundancy of these functions is large and the energy of sigmoid function is limitless, which inevitably cause the slow convergence speed.

In recent years, wavelets have become a very active subject in many scientific and engineering research areas. Especially, Wavelet Neural Networks (WNN) inspired by both the feedforward neural networks and wavelet decomposition has received considerable attention [4]-[7]. The main characteristic of WNN is that some kinds of wavelet functions are used as the nonlinear transformation function in the hidden layer, instead of the usual sigmoid function. WNN incorporate the good learning ability and generalization of neural networks and the good property of localiza-
tion of wavelet transform, which have shown powerful approximation capability and superiority to MLP in many applications.

In this paper, the WNN is trained with an algorithm of back propagation type to adjust the connection weight parameters. However, the network weights are difficult to learn because of its structural nonlinearity. Usually the local minima for neural network learning with gradient decent algorithm is inevitable in searching for optimal weights [8]. To obtain the optimal solution, a Genetic Algorithm (GA) combined with a steepest descent technique and least squares technique for optimal selection of the basis of the networks have been used [9]-[13].

With rapid progress in computer technologies in general and especially in microcomputers, the need for mass data storage devices has rapidly multiplied in recent years [14]-[16]. Recording densities for disk drive have gone up more than 100 times and data access performance has gone up by at least 15 times over the last two decades alone. What made this remarkable progress possible is a combination of developments in some key technologies in areas of materials for heads and media, electronics for data communications and especially head-positioning systems and innovative designs for spindle motors. The spindle motor is one of the most important elements in a disk drive and in many ways determines the drive capacity and performance through its configuration, operating speed and mechanical and electrical performance, like run-out, resonance, starting torque, run-current etc. The motivation of this paper is to develop a wavelet-neural-network controller for spindle motor of DVD-ROM. With the proposed design method, favorable performances of spindle motor are achieved. The analysis, design and experiment of the proposed method are described in details.

II. Wavelet neural network (WNN)

Wavelet transform has the natural characteristic of extension from high frequency to low frequency and it has good localization in both time and frequency space, so the wavelet coefficient defined as the inner product of signal and wavelets will response adaptively the variation of signal even signal is suddenly changing and unstable [17]-[19]. Therefore the neural networks using wavelet basis function are especially suitable for approximating arbitrary nonlinear functions. The structure of our wavelet neural network is shown in Figure 1.
As shown above, this is a three-layers wavelet neural network, including input layer, hidden layer and output layer. Where:

- $w_{ki}$: the weight between input layer and hidden layer
- $w_k$: the weight between hidden layer and output layer
- $a_k$: the dilation parameter of wavelet function in hidden neuron
- $b_k$: the translation parameter of wavelet function in hidden neuron
- $n$: the number of iteration
- $Y(n)$: represents actual output vector at the $n$th iteration
- $X, T$: represents input vector and target vector, respectively

The network structures applied for representation are determined by wavelet analysis. The output errors are minimized using an adaptation technique based on the LMS algorithms and the network gradually recruits hidden neurons to efficiently and sufficiently cover the time-frequency region by a given target. The parameters of the wavelet neural network are updated using steepest gradient descent method to minimize the output errors. Each hidden neuron has a square window in the time-frequency plane. The optimization rule is applied to the hidden neurons where the selected point falls into their window. Therefore, the learning cost can be reduced.

The following steps are for implementation of wavelet neural network.
Step 1. Initialize the parameters of WNN to some random value.

Step 2. Present an input pattern and corresponding target vector.

Step 3. Compute the output of WNN

\[ Y(n) = \sum_k w_k \Psi \left( \frac{\sum_i w_{ki} x_i(n) - b_k}{a_k} \right) \]  

the wavelet employed here is second differential of Gaussian function

\[ \Psi_{a,b}(x) = (1 - x^2) \cdot \exp \left( -\frac{1}{2} x^2 \right). \]  

Step 4. The WNN parameters are optimized using steepest gradient descent method to minimize a cost function or the energy function, \( E \), which is written as (3). The parameters \( w_{ki}, w_k, a_k \) and \( b_k \) are updated with \( \Delta w_{ki}, \Delta w_k, \Delta a_k \) and \( \Delta b_k \) respectively as follows:

\[ E(n) = \frac{1}{2} \sum (T(n) - Y(n))^2, \]  

\[ \Delta w_{ki}(n) = -\eta \frac{\partial E}{\partial w_{ki}(n)} + \alpha \Delta w_{ki}(n-1), \]  

\[ \Delta w_k(n) = -\eta \frac{\partial E}{\partial w_k(n)} + \alpha \Delta w_k(n-1), \]  

\[ \Delta a_k(n) = -\eta \frac{\partial E}{\partial a_k(n)} + \alpha \Delta a_k(n-1), \]  

\[ \Delta b_k(n) = -\eta \frac{\partial E}{\partial b_k(n)} + \alpha \Delta b_k(n-1). \]  

Where

\( \eta \): the learning rate, it specifies the magnitude of the update step for parameters in the negative gradient direction. It is reasonable that during the learning process should be large at the beginning of training and gradually decrease at the network converges.

\( \alpha \): the momentum, it ameliorate oscillation at the network converges and improve the convergence speed.

Step 5. If convergence is achieved, stop, else set \( n \leftarrow n + 1 \), the go back to Step 3.
III. Genetic algorithm

Genetic algorithms were introduced by Holland in early seventies as a special technique for function approximation. They are quite different from other more conventional optimization methods that they are mainly stochastic in nature. The basic process of genetic algorithm is as follows. First, a population of chromosomes is created. Second, the chromosomes are evaluated by a defined fitness function. Third, some of the chromosomes are selected for performing genetic operations. Fourth, genetic operations of crossover and mutation are performed. The produced good offspring replace their parents in the initial population. This process repeats until a user-defined criterion is reached [20]-[21].

They are in the following aspects:

1. Genetic algorithm works with a coding of the parameter set, and not the parameters themselves.
2. Genetic algorithm searches from a population of points, and not from a single point like conventional algorithms.
3. Genetic algorithm uses fitness function information, and not derivative or other auxiliary data.
4. Genetic algorithm use probabilistic transition rules by stochastic operands, and not deterministic rules.

Genetic algorithm is proposed in this paper to find the initial value of parameters of WNN, the algorithm is implemented as follows:

Step 1. Randomly generate initial population of binary chromosomes for the parameters of WNN, \( w_{ki}, w_k, a_k \) and \( b_k \).

Step 2. Evaluate fitness function of each chromosome in the population. The better chromosomes will return higher values in this process. The fitness function to evaluate a chromosome in the population can be written as the inverse of formula (3).

Step 3. Stop and output the optimum solutions if suitable solutions have been found or if the available computing time has expired; else proceeds.

Step 4. Generate some new chromosome (offspring) from their parents through genetic operations, which is given as follows:
a. Use roulette wheel selection to pick up the better chromosomes to be parent generation in the mating pool. The chromosomes with a higher fitness value have a higher probability of contributing one or more offspring in the next generation.

b. Produce new offspring by crossing from their parent generation. The offspring are expected to be more fit than the parents.

c. Randomly choose some chromosome from new offspring for mutation operation. This operator can create new genetic material in the population to maintain the population’s diversity.

Step 5. Reserve the best chromosome of each generation, because it is not guaranteed that the superior chromosome will reside in the offspring after crossover and mutation operation. As a result, reversing the elite of each generation can improve convergence speed.

Step 6. Proceed to Step 2.

IV. Design of GAWNN speed controller

The object of GAWNN speed controller is to drive spindle motor to reach the speed command. Figure 2 shows the relationship between input variables and output variables. The speed error $e$ and its derivative $de$ are defined as input variables, at a sampling time $t$, they are defined as:

$$e(t) = \omega^*(t) - \omega(t),$$  \hspace{1cm} (8)

$$de(t) = e(t) - e(t - 1),$$  \hspace{1cm} (9)

where $\omega^*(t)$ is speed command and $\omega(t)$ is actual feedback speed.

The output variable of the controller is the magnitude of voltage change $\Delta v$ for controlling the spindle motor.

![Figure 2](image_url)

The GAWNN speed control system
The training pattern of GAWNN speed controller is obtained from conventional PID controller for the speed range 3000rpm to 5000rpm in our previous work with poor data eliminated and better data added [22]. We expect the GAWNN may emulate the inherent characteristic of training pattern and expect to substitute for the conventional controller.

Figure 3 shows the output error after training, where the initial parameters of network are set to random value. The WNN procedure takes 5000 iterations and use 400 sets training pattern. The network is trained with 2 nodes as the input, 10 neurons for the hidden layer and 1 neuron as the output. The learning rate and momentum are 0.5 and 0.2, respectively.

To improve the learning performance of WNN, genetic algorithm is used to find the initial value of WNN parameters and steepest gradient descent method is used to train the network to find the optimum solutions. The genetic algorithm procedure takes 1000 iterations and the population size is 30. Each parameter is encoded into a 40-bit chromosome. The probability of crossover and mutation are 0.9 and 0.02, respectively. The setup of WNN is same as we mention above, the difference are that we set the number of iteration to be 2000 and 5 neurons is used in hidden layer. The output error of GAWNN after training is shown in Figure 4. As we see, it implies the GAWNN can improve the convergence speed and its architecture is simpler.
V. Experimental results

This section, we practically use the GAWNN speed controller to control spindle motor.

A. GAWNN controller for no-loading experiments

Figure 5 is the block diagram of closed-loop GAWNN control system. The laboratory setup comprise several subsystem: the spindle motor, a personal computer, the motor driving ICs and the A/D, D/A, F/V converter. The controller algorithm is housed inside the computer using Turbo-C language. The procedures are as follows: Input speed command is generated in the software environment and it is converted to an analog signal voltage, subsequently injected into the driving ICs. The output of driving ICs is used to drive the spindle motor. The A/D converter gets the feedback speed and conversion to digital signal, subsequently injected into the computer for GAWNN control. The sampling rate of the system is 4kHz.

The spindle motor speed response of no-loading conditions for 3000rpm and 5000rpm are shown in Figure 6 and 7, respectively. They imply spindle motor reach the speed command about on 2 to 3 seconds.
Figure 5
GAWNN control system

Figure 6
Speed response for 3000 rpm under no-loading conditions

Figure 7
Speed response for 5000 rpm under no-loading conditions
B. GAWNN controller for loading experiment

The laboratory setup of loading experiment is like the no-loading one except that the disk is put over the spindle motor. The spindle motor speed response of loading conditions for 3000rpm and 5000rpm are shown in Figure 8 and 9, respectively. They imply spindle motor reach the speed command about on 10 seconds.

VI. Conclusion

A GAWNN method has been introduced for designing spindle motor speed controller of DVD-ROM. The training data of GAWNN is obtained
from the conventional PID controller with poor data eliminated and better data added. Experimental result show that the GAWNN provide the better learning performance and some undesired peaks are eliminated and the response of the spindle motor is improved. Since the network is off-line trained by bounded input-output data of conventional controller, the response of the motor will be bounded. From the result of this paper, it can show that control of the spindle motor using GAWNN can reach its stable revolution speed more rapidly and has smaller stead-state error.

Acknowledgements. This work was supported by National Science Council of Taiwan, Republic of China, under Grant number NSC 89-2213-E-035-078.

References


WAVELET NEURAL NETWORK 563


Received December, 2005