A NEURO-FUZZY SYSTEM DESIGN METHODOLOGY FOR VIBRATION CONTROL

S. M. Yang, Y. J. Tung, and Y. C. Liu

ABSTRACT

Fuzzy system has been known to provide a framework for handling uncertainties and imprecision by taking linguistic information from human experts. However, difficulties arise in determining effectively the fuzzy system configuration, i.e., the number of rules, input and output membership functions. A neuro-fuzzy system design methodology by combining neural network and fuzzy logic is developed in this paper to adaptively adjust the fuzzy membership functions and dynamically optimize the linguistic-fuzzy rules. The structure of a five-layer feedforward network is shown to determine systematically the correct fuzzy logic rules, tune optimally (in the sense of local region) the parameters of the membership functions, and perform accurately the fuzzy inference. It is shown both numerically and experimentally that engineering applications of the neuro-fuzzy system to vibration control have been very successful.

KeyWords: Neural network, fuzzy logic, vibration control.

I. INTRODUCTION

Since Zadeh proposed the fuzzy set in 1965, applications of fuzzy logic system have had increasing attention. A fuzzy system that contains human thinking and reasoning is able to deal with inexact information. Most applications are aimed at fuzzy modeling and fuzzy logic control [1-2]. The former uses linguistic description to establish a logic-based system model with fuzzy predicates, while the latter is a knowledge-based system operating in linguistic, rule-based structure. One of the advantages of fuzzy logic controller is that it does not require mathematical model. However, its performance strongly depends on the selection of input and output membership functions and the fuzzy rules, which, conventionally, are determined by experts' knowledge or experiences. For systems with practical complexity and/or uncertainty, it is often difficult to extract the “inside” knowledge so as to determine the adequate fuzzy structure, membership functions, and logic rules.


In this paper, a neuro-fuzzy controller with self-organized, optimal fuzzy rules and membership functions is developed. The structure of a five-layer feedforward
network is to determine systematically the fuzzy logic rules, tune optimally (local optimal) the parameters of the membership functions, and perform accurately the fuzzy inference. Both numerical simulation and experimental verification are conducted to illustrate the effectiveness of the neuro-fuzzy system on vibration control.

II. AN INTEGRATED NEURO-FUZZY DESIGN

A typical fuzzy system is comprised of four principal components: fuzzifier, fuzzy rule base, inference mechanism, and defuzzifier. The fuzzifier performs the fuzzification that converts input data into suitable linguistic value, which may be viewed as labels of fuzzy sets. The fuzzy rule base is comprised by a set of fuzzy if-then rules with the antecedents and consequents in linguistic variables. This collection of fuzzy rules characterizes the simple input-output relation of the system. In practice, however, it is difficult to establish effectively and efficiently such a fuzzy model. A practical way is to use neural network of highly interconnected processing elements and information-encoded connection weights to map the numerical input-output function.

A five-layer neuro-fuzzy system by integrating the fuzzy logic and neural network as shown in Fig. 1 is developed to determine the fuzzy logic rules and optimize the membership functions. Layer 1 defines the input nodes and layer 5 the output nodes. Layer 2 and layer 4 are the nodes of membership functions representing the linguistic terms. Among many types of membership function, the Gaussian function is adopted. Layer 3 defines the nodes representing the fuzzy rules,

Rule $i$: If $x_1$ is $A_{i1}$ and $x_2$ is $A_{i2}$ and $x_n$ is $A_{in}$ then $y$ is $B_i$. (1)

where $x_j$, $j = 1, 2, \ldots, n$, is the input, $y$ is the output of the fuzzy rule $i$, and $A_{i1}$, $A_{i2}$, ..., $A_{in}$ and $B_i$ are the fuzzy membership functions. The links of layer 3 define the precondition of the rule nodes while those of layer 4 define the consequence.

The functions of the nodes in each of the five layers are described as follows:

Layer 1. Input Layer

Each node in this layer is an input variable. The node directly transmits the input signal to the next layer,

$$O_{i1} = x_i$$ (2)

where $O_{i1}$ is the output of the $i$th (the first subscript) node in layer 1 (the second subscript), and $x_i$ is the $i$th input variable.

Layer 2. Fuzzification Layer

Fuzzification is done in this layer with each node corresponding to one linguistic term of the input variables in layer 1. The value of the membership function of a fuzzy set is calculated by the Gaussian function

$$net_{j2} = O_{i1}$$ (3)

$$O_{j2} = \exp \left( -\frac{net_{j2} - m_{j2}}{\sigma_{j2}} \right)^2$$ (4)

where $net_{j2}$ is the input value of the $j$th node in layer 2, $m_{j2}$ and $\sigma_{j2}$ are the center (mean) and width (variance) of the Gaussian membership function of the $j$th node, respectively. All links between layer 1 and 2 are set to unity.

Layer 3. Fuzzy Rule Layer

Each node represents one fuzzy rule conducting precondition matching. The weight of the links are set to unity. The output of node $j$ is determined by fuzzy AND operation,

$$net_{j3} = \min_{i \in I_j}(O_{i2})$$ (5)

$$O_{j3} = net_{j3}$$ (6)

where $I_j$ is the set of indices of the nodes in layer 2 that are connected to node $j$ in layer 3.

Layer 4. Consequent Layer

The nodes of layer 3 and 4 are fully connected before network learning, and each node in this layer represents the consequent part of a fuzzy rule in OR operation,

$$net_{j4} = \max_{j \in I_l}(O_{j3}(w_{lj})^2)$$ (7)

$$O_{l4} = net_{l4}$$ (8)
where \( I_i \) is the set of indices of the nodes in layer 3 connected to node \( i \) in layer 4. The connection weight \( w_{ij} \) indicates the dependence of the \( j \)-th rule on the \( i \)-th output linguistic variable. The square of the weight \( w_{ij} \) is used because the strength of the IF- and THEN-part of the rule represented by the nodes in layer 3 and 4 are always positive.

**Layer 5. Output Layer**

The node in this layer calculates the output value of the neuro-fuzzy model by using the center of area defuzzification scheme. The defuzzified value of a node is

\[
O_{5} = \sum_{j \in I_i} \left( m_{j} \sigma_{j} O_{j} \right) / \sum_{j \in I_i} \left( \sigma_{j} O_{j} \right)
\]

where \( I_i \) is the set of indices of the nodes in layer 4 connected to node \( i \) in layer 5. \( m_{j} \) and \( \sigma_{j} \) are the center (mean) and width (variance) of the membership function of the \( j \)-th node in layer 4, respectively. The weights of the links between the layer 4 and 5 are set to unity.

The five-layer network in Mamdani fuzzy model is to determine fuzzy logic rules and optimize the membership functions. By modifying adaptively the connection weights (links) and the parameters of the Gaussian membership function, the integrated neural network and fuzzy system facilitates a systematic neuro-fuzzy design.

**III. THE LEARNING ALGORITHM**

The neuro-fuzzy system design methodology is a three-phase learning process. A self-organized learning algorithm is employed to locate the initial membership functions in phase one, the error backpropagation learning algorithm is then used to find the fuzzy rules in phase two, and a supervised learning scheme is applied to adjust optimally the membership functions of the input and output variables in phase three. The links between the rule nodes in layer 3 and the consequent nodes in layer 4 are fully connected before network training, for the consequence of the rule nodes are not yet decided. Only one suitable term in each output variable will be selected after learning process. The three-phase learning is described as follows,

**3.1 Phase-One: The initial membership function**

In this phase, the center (or mean) and the width (or variance) of the membership function are determined by self-organized learning from the given training data. This serves to allocate the network resources efficiently by placing the domains of membership function covering only the input-output space where data are present. Kohonen’s feature-map algorithm is adopted to find the center of the membership function,

\[
\|x(k) - m_i(k)\| = \min_{i \in I_i} \|x(k) - m_i(k)\| \quad \text{(10)}
\]

\[
m_i(k + 1) = m_i(k) + \alpha (x(k) - m_i(k)) \quad \text{for } m_i \neq m_c \quad \text{(11)}
\]

where \( x(k) \) and \( m_i(k) \) are the input data and the center of existed membership function, respectively. The subscript \( c \) defines the associated closest value and \( \alpha \) is a monotonically decreasing scalar learning rate. This adaptive formulation runs independently for each input and output linguistic variable.

Once the center \( m_i(k) \) of each membership function is calculated, its width \( \sigma_i(k) \) can be determined either by minimizing the objective function of the N-nearest-neighbor heuristic

\[
J = \frac{1}{2} \sum_{k} \left[ \sum_{j \in N_{nearest}} \left( \frac{m_i - m_j}{\sigma_i} \right)^2 - r \right]^2 \quad \text{(13)}
\]

where \( r \) is an overlap parameter, or by the first-nearest-neighbor heuristic as

\[
\sigma_i = \frac{|m_i - m_{closest}|}{r} \quad \text{(14)}
\]

Note that the third learning phase will optimally adjust the center and width of each membership function.

**3.2 Phase-Two: The fuzzy logic rules**

After the parameters \( (m_i \) and \( \sigma_i \)) of each membership function have been calculated, the output of layer 2 can be transmitted to layer 3 to find the firing strength of each rule node. Based on the firing strength and the node output in layer 4, the correct consequence-link of each rule node can be determined by minimizing the error function

\[
E = \frac{1}{2} (d(k) - y(k))^2 \quad \text{(15)}
\]

through the backpropagation learning algorithm, where \( d(k) \) is the desired output and \( y(k) \) is the current output. The weight of the links from layer 3 to 4 are adjusted by using the gradient decent as

\[
w_{ij}(k + 1) = w_{ij}(k) - \eta \left( \frac{\partial E}{\partial w_{ij}} \right) \quad \text{(16)}
\]

where \( \eta \) is the learning rate. By using Eqs. (5)-(9) and (15), the update rule in Eq. (16) can be written as

\[
w_{ij}(k + 1) = w_{ij}(k)
\]
After adjusting the weight, the correct consequent link of each rule node can be determined. For every antecedent clause, the centroid of all the possible consequent is calculated. Consider the subnet as shown in Fig. 2 as an illustration. The output is

$$y_{ij} = \frac{\sum m_{ij} \sigma_{ij} O_{ij}}{\sum \sigma_{ij} O_{ij}}$$  \quad (18)

where $y_{ij}$ can be viewed as the centroid of $c_{ij}$ of all links from the rule node $i$ in layer 3 to the consequent node in layer 4. This centroid can be used for rule reduction, and then only the dominant rule whose consequent has the highest membership value is selected.

### 3.3 Phase-Three: The membership function

After the fuzzy rules have been deduced, the neuro-fuzzy model structure is fully established. Supervised learning by using the backpropagation algorithm is applied to optimize the parameters of each membership function. Starting at the output node, a backward pass calculates the gradient of the error function for all the hidden nodes. In layer 5, the center and width of each Gaussian membership function are the adjustable parameters. The error propagated to the preceding layer is

$$\delta_5 = -\frac{\partial E}{\partial O_5} = d(k) - y(k)$$  \quad (19)

By using Eq. (9) and the gradient of the center $m_{ij}$, the center parameter is updated by

$$m_{ij}(k+1) = m_{ij}(k) + \eta (d(k) - y(k)) \frac{\sigma_{ij} O_{ij}}{\sum \sigma_{ij} O_{ij}}$$  \quad (20)

Similarly, the width parameter is updated by

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) + (d(k) - y(k)) \frac{m_{ij} O_{ij} (\sum \sigma_{ij} O_{ij}) - (\sum m_{ij} \sigma_{ij} O_{ij}) O_{ij}}{(\sum \sigma_{ij} O_{ij})^2}$$  \quad (21)

In layer 4, however, there is no parameter to be adjusted, and the error signal $\delta_{ij}$ is

$$\delta_{ij} = (d(k) - y(k)) \frac{m_{ij} \sigma_{ij} (\sum \sigma_{ij} O_{ij}) - (\sum m_{ij} \sigma_{ij} O_{ij}) \sigma_{ij}}{(\sum \sigma_{ij} O_{ij})^2}$$  \quad (22)

Similarly in layer 3, only the error signal $\delta_3$ is needed and it is identical to $\delta_{ij}$. In layer 2, the center and width parameters are updated by

$$m_{ij}(k+1) = m_{ij}(k) - \eta \frac{\partial E}{\partial O_{ij}} \left[ \exp\left( -\frac{\text{net}_{ij} - m_{ij}}{\sigma_{ij}} \right) \right]^2 \frac{2(\text{net}_{ij} - m_{ij})}{(\sigma_{ij})^3}$$  \quad (23)

The above learning algorithm highlights the computation procedures in the neuro-fuzzy system design methodology.

### IV. CONTROLLER DESIGN FOR VIBRATION SUPPRESSION

Many neuro-fuzzy controller designs were previously presented. One is the inverse control method to identify the inverse dynamics of an unknown plant and then apply the identification model in the controller design such that the overall transfer function of the model and plant becomes unity. This method seems straightforward and only one learning task is needed to find the inverse model of the plant. However, the assumption of the existence of the inverse plant is questionable. Moreover, minimization of the error during identification of the inverse model does not guarantee the closed-loop system stability. Thus, the learning from a reference model to minimize the system error as shown in Fig. 3(a) is used to design the neuro-fuzzy controller. A reference model can implicitly specify the desired performance of the closed-loop system,
and the neuro-fuzzy controller for vibration suppression can be self-organized via the output error between the plant and reference model by using the three-phase learning algorithm. The appropriate fuzzy rules and membership functions can thus be determined to meet the design requirement. The controller stability is guaranteed by the selection of an asymptotically stable reference model. In addition, the feedback of displacement and velocity of a measurement point in the system further increases the stability by higher damping and stiffness [15].

An example is employed to illustrate the effectiveness of the neuro-fuzzy controller. Consider the vibration control of the hard spring system. A linear reference model

\[ \ddot{x} + 5 \dot{x} + 1000x = 1000u(t) \]  

(25)

is selected to define the system performance,

\[ \ddot{x} + 2 \dot{x} + 400x + 200x^3 = 500u(t) \]  

(26)

The schematic diagram of the training process is shown in Fig. 3(b). The objective is to determine a bounded control input \( u(t) \) such that \( \lim_{t \to \infty} e_i(t) = 0 \) where \( e_i(t) = x_i(t) - \dot{x}(t) \). The neuro-fuzzy controller has three input nodes for receiving the command input, displacement and velocity, and one output node of the control input signal. The input of displacement and velocity are partitioned into seven fuzzy linguistic spaces, while the input of command has just one fuzzy cluster; the output is partitioned into eleven fuzzy spaces. A total of five hundred training sets randomly distributed in the range of [5, −5] is generated and collected by the sampling frequency of 100 Hz for training the controller. The sum-squared error in phase-two and phase-three learning are shown in Figs. 4(a) and 4(b), respectively. After sufficient learning, the performance of the neuro-fuzzy controller is validated by comparing its response to that of the system without control. Figure 5 shows the open-loop response and the closed-loop response of the hard spring under an initial displacement \( x(0) = 4, \dot{x}(0) = 0 \). It is shown that the vibration is suppressed efficiently by the neuro-fuzzy controller.

The controller effectiveness is validated by the vibration suppression experiment of a smart composite beam with embedded piezoelectric sensor and actuator. The \([90/90/90/90/0]_2\) glass fiber composite laminated beam of \(265 \times 40.5 \times 1\) mm is made of S-glass/epoxy uni-directional pre-prag tapes (Fiberite Hy-E9134B) with two embedded \(120 \times 30 \times 0.375\) mm piezoelectric actuators. Detail description of the smart composite can be found in Yang and Lee [16].

In neuro-fuzzy controller design, the displacement, velocity and command input are partitioned into 5, 3, and 1 fuzzy set. The controller has 3 input nodes, \( 5 + 3 + 1 = 9 \) fuzzification nodes, \( 5 \times 3 \times 1 = 15 \) rule nodes, 5
defuzzification nodes, and 1 output node of displacement, and the design is completed by the three-phase learning. Once the number of linguistic term for each input and output variables are selected, the structure of the neuro-fuzzy network can be automatically constructed. The data for training the controller is selected in the estimated vibration range, and the desired output is then calculated from a second-order reference model.

\[ \ddot{x}_m + 3\dot{x}_m + 3948x_m = 3948u(t) \]  

(27)

A total of 500 training sets uniformly distributed in the possible vibration range is collected for training the neuro-fuzzy controller by updating the connection weights between layer 3 and layer 4 for the fuzzy rules in phase-two and by optimizing the parameters of membership functions in phase-three.

Figure 5 shows the simulation of time response of the hard spring system under the initial condition \( x(0) = 4, \dot{x}(0) = 0 \).

Fig. 5. Simulation of time response of the hard spring system under the initial condition \( x(0) = 4, \dot{x}(0) = 0 \).

Fig. 6. Schematic diagram of the vibration control experiment.

Fig. 7. Experimental result of the composite smart structure with the neuro-fuzzy controller.

V. CONCLUSION

1. A systematic design methodology of neuro-fuzzy system is developed. The model of a five-layer network includes not only the fuzzy logic rules and the membership functions but also the fuzzy inference. By utilizing the neural network learning, the system can automatically and simultaneously identify the fuzzy logic rules and tune the membership functions. The controller stability is based on an asymptotically stable
(desired) reference model as indicated in Fig. 3. It should be noted that both stability theory and convergence study remain to be investigated.

2. A three-phase learning algorithm is developed to determine the optimal parameters of the neuro-fuzzy network. The self-organized learning algorithm locates the initial membership functions in phase-one, the error backpropagation learning algorithm deduces the fuzzy rules in phase-two, and a supervised learning scheme optimally adjusts the membership functions of input and output variables in phase-three. Note that the membership functions are optimal in the sense of locally region(s), for there is no guarantee in engineering implementation that they are globally optimal. Nevertheless, such “optimal” membership functions are shown effective to controller design in numerical studies and experimental verification. The fuzzy controller can be automatically constructed, with or without experts’ knowledge.

3. The neuro-fuzzy controller is successfully applied to the vibration suppression of a hard spring system and a composite smart structure with embedded piezoelectric sensor/actuator. No expert knowledge is needed in constructing the neuro-fuzzy model. Simulation and experimental results show that the neuro-fuzzy system is effective to vibration control.

ACKNOWLEDGEMENTS

The authors are grateful to the support in part by the National Science Council, Taiwan, R.O.C., under NSC92-2212-E006-012 and the reviewers’ comments in enhancing the clarity of the paper.

REFERENCES


Shih-Ming Yang has been with the Institute of Aeronautics and
Yu-Ju Tung received B.S. and M.S. in Aeronautics and Astronautics from the National Cheng Kung University, Taiwan in 1999 and 2001. Her research interests have been in Digital Signal Processing of computer peripherals. She is currently a Firmware Engineer on wireless applications in Syntech Information Co. in Taiwan.

Yu-Chuan Liu is an Assistant Professor in the Department of Information Management at Tainan Women’s University. Dr. Liu received his Ph.D. in Aeronautics and Astronautics from the National Cheng Kung University of Taiwan in 1993 and served as the R&D manager and vice-president in mechanical and optoelectronics industries. Dr. Liu joined the faculty in 2002 and his research interests are in process control and management information system, computer-integrated manufacturing (CIM), and e-Business. Dr. Liu has published more than 25 technical articles in this area.