Journal of Universal Computer Science, vol. 15, no. 13 (2009), 2547-2565
submitted: 31/10/08, accepted: 13/6/09, appeared: 1/7/09 © J.UCS

A Hammerstein-Wiener Recurrent Neural Network with Frequency-Domain Eigensystem Realization Algorithm for Unknown System Identification

Yi-Chung Chen and Jeen-Shing Wang
(School of Electrical and Computer Engineering
National Cheng Kung University
Tainan 701, Taiwan, R.O.C.
jeenshin@mail.ncku.edu.tw)

Abstract: This paper presents a Hammerstein-Wiener recurrent neural network (HWRNN) with a systematic identification algorithm for identifying unknown dynamic nonlinear systems. The proposed HWRNN resembles the conventional Hammerstein-Wiener model that consists of a linear dynamic subsystem that is sandwiched in between two nonlinear static subsystems. The static nonlinear parts are constituted by feedforward neural networks with nonlinear functions and the dynamic linear part is approximated by a recurrent network with linear activation functions. The novelties of our network include: 1) the structure of the proposed recurrent neural network can be mapped into a state-space equation; and 2) the state-space equation can be used to analyze the characteristics of the identified network. To efficiently identify an unknown system from its input-output measurements, we have developed a systematic identification algorithm that consists of parameter initialization and online learning procedures. Computer simulations and comparisons with some existing models have been conducted to demonstrate the effectiveness of the proposed network and its identification algorithm.

Keywords: Hammerstein-Wiener model, recurrent neural networks, parameter initialization/ optimization
Category: I.2.6, I.2.8, F.1.1

1 Introduction

In the past decade, block-oriented (BO) models, such as Hammerstein, Weiner or Hammerstein-Wiener models that consist of an interconnection of linear dynamic and nonlinear static subsystems, have been widely used in the problems of system identification. For example, Bai and Li [Bai and Li 04] introduced the convergence property of iterative Hammerstein systems in the system identification problem, and Jia *et al.* [Jia et al. 05] proposed a non-iterative identification procedure for a neuro-fuzzy-based Hammerstein model to overcome the problems of initialization and convergence of the model parameters. Chen *et al.* [Chen et al. 97] utilized a Wiener model to identify chaotic systems. The dynamic element is represented by a simple linear plant, and the static nonlinear element is represented by a feedforward neural network. Arto *et al.* [Arto et al. 01] proposed a MIMO Wiener model for the chromatographic separation process. The dynamic linear component was replaced by Laguerre filters and the static nonlinear component was described as a feedforward neural network. Westwick and Kearney [Westwick and Kearney 01] used a

Hammerstein model to identify a stretch reflex dynamic system. Kalafatis *et al.* [Kalafatis et al. 05] successfully applied a Wiener model to the PH processes. Also, a Hammerstein-Wiener model was utilized to analyze the submarine detection by Abrahamsson *et al.* [Abrahamsson et al. 07]. There are three main advantages of these block oriented models. First, the feature of dynamics is involved essentially in the linear system, while the complexity of nonlinearity is contained only in the static nonlinear subsystems. Thus we can use less computation time and memory in system identification problem. Second, the well-developed dynamic linear or static nonlinear theory can be applied in the modeling procedure directly instead of using a complex dynamic nonlinear theory. Finally, control of these models is easy since a divide-and-conquer strategy can be applied to these models. Among these BO models, the Hammerstein-Wiener model is expected to have better performance because it contains the advantages of both Hammerstein and Wiener models [Zhu 02].

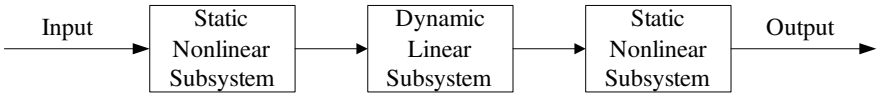

Figure 1: The *block diagram of the Hammerstein-Wiener model.*

In this paper, we proposed a Hammerstein-Wiener recurrent neural network for dynamic system identification. The static nonlinear parts of our proposed Hammerstein-Wiener model is constituted by feedforward neural networks with nonlinear functions, and the dynamic linear part is approximated by a recurrent network with linear activation functions. To identify an unknown system efficiently, our research effort has been directed to investigate the following subjects: 1) model selection, 2) model construction and initialization, and 3) model parameterization. In the model selection, we realized a conventional Hammerstein-Wiener model by a simple recurrent network whose structure can be mapped into a state-space equation. For the model construction and initialization, we developed a hybrid Hammerstein-Wiener initialization algorithm (HHWIA) which includes an active region boundary initialization algorithm for the first static nonlinear subsystem; a frequency domain eigensystem realization algorithm (FDERA) for the dynamic linear subsystem; and the least-squares method for the second static nonlinear subsystem to reduce the model error. The HHWIA guarantees that the initial network can be operated within the range that is close to a local minimum in the error space. Thus, the learning convergence of the identification process can be enhanced. Finally, for the model parameterization, we derived a recursive recurrent learning algorithm based on the concept of the ordered derivatives to adapt the network to emulate the dynamic behavior of the unknown system. In this study, our objective is to develop a powerful system identification algorithm for the proposed structure which can accomplish system identification automatically and effectively.

The organization of this paper is as follows. The structure of the proposed Hammerstein-Wiener recurrent neural network is presented in Section 2. In Section 3, we introduce the system identification algorithm that contains a model construction and initialization algorithm and a parameter learning algorithm in detail. Computer

simulations and comparisons with some existing approaches on benchmark examples are provided in Section 4. Finally, conclusion is given in Section 5.

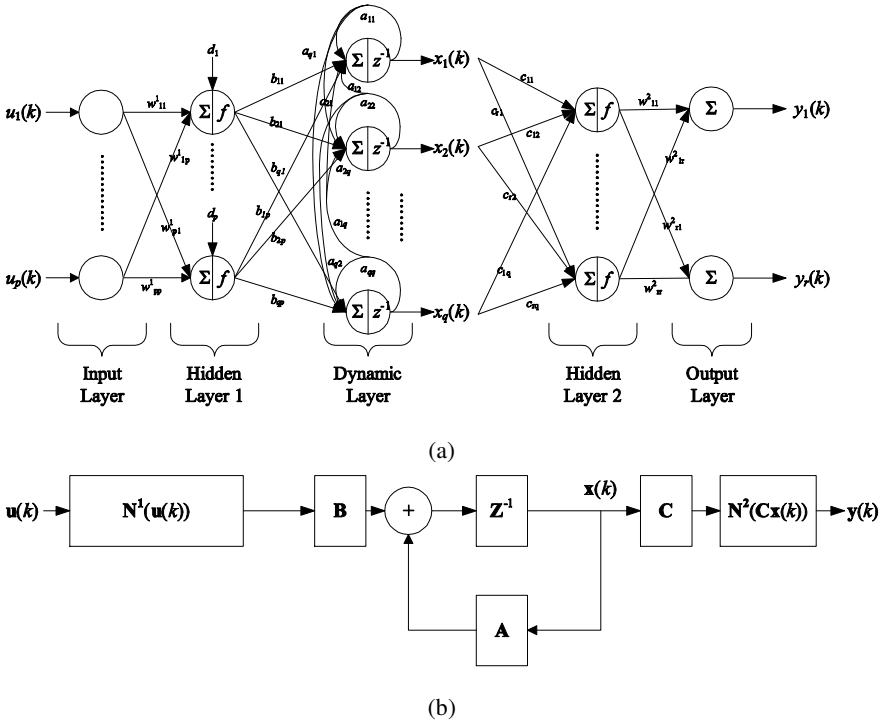

Figure 2: (a) The topology of the proposed Hammerstein-Wiener recurrent neural network. (b) The block diagram of the proposed network.

2 Structure of Hammerstein-Wiener Recurrent Neural Network

In this paper, we developed a novel recurrent neural network structure that realizes a Hammerstein-Wiener model. Figure 1 shows the block diagram of the proposed Hammerstein-Wiener model. The network consists of a linear dynamic subsystem that is sandwiched in two nonlinear static subsystems. The proposed network is a four-layered recurrent neural network with three subsystems. The first nonlinear static subsystem is implemented by a simple feedforward neural network composed of an input layer and a hidden layer. The input layer is only responsible to transmit the input values into the network while the hidden layer provides a nonlinear transformation for mapping the input values into a state-space. The second subsystem is a linear dynamic model that contains a dynamic layer. This layer consists of a set of neurons with feedback connections embedded with time-delay elements. The dynamic layer integrates the information of the transformed input data from the hidden layer

and the state history from the memories of the dynamic layer to form the current state of the network. Finally, the third subsystem, a nonlinear static subsystem consisting of summation functions and nonlinear activation functions, is the output layer of the whole network. In the output layer, the state variables acquired from the dynamic layer are linearly combined by different weights and are then sent to a nonlinear transformation to obtain the network output. In our network, hyperbolic tangent sigmoid functions are chosen as the activation functions in both nonlinear static subsystems because the functions can provide dual polarity signals. Moreover, the invertible property of the functions can help us to obtain the desired outputs of the linear dynamic subsystem conveniently for estimating the initial parameters of the linear dynamic subsystem. According to the block diagram shown in Fig. 2(b), the structure in Fig. 2(a) can be expressed by the following state-space equations:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{N}^1\left(\mathbf{u}(k)\right), \\ \mathbf{y}(k) &= \mathbf{N}^2(\mathbf{C}\mathbf{x}(k)), \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{q\times q}$, $\mathbf{B} \in \mathbb{R}^{q\times p}$, $\mathbf{C} \in \mathbb{R}^{r\times q}$, $\mathbf{N}^1 \in \mathbb{R}^{p}$, $\mathbf{N}^2 \in \mathbb{R}^{r}$, $\mathbf{u} = [u_1, \ldots, u_p]^T$ is the input vector, $\mathbf{y} = [y_1, \ldots, y_r]^T$ is the output vector, and $\mathbf{x} = [x_1, \ldots, x_q]^T$ is the state vector. In addition, p and r are the dimensions of the input and output layers, respectively. q is the total number of the states which is equal to the number of neurons in the dynamic layer. The components of matrix $\mathbf{A}$ represent the degrees of the inter-correlations among the state variables. Matrices $\mathbf{B}$ and $\mathbf{C}$ stand for the weights of the inputs in the dynamic layer and the weights of state variable to output layer. Finally, $\mathbf{N}^1 = [n_1{}^1, \ldots, n_p{}^1]^T$ and $\mathbf{N}^2 = [n_1{}^2, \ldots, n_r{}^2]^T$ are nonlinear function vectors in the first and second nonlinear static subsystem, respectively. We now summarize the equations of the proposed network as follows:

$$m_j(k) = \sum_{i=1}^{p} (w^1{}_{ji} u_i) + d_j, \tag{2}$$

$$n_j(k) = f(m_j) = \frac{\exp(m_j) - \exp(-m_j)}{\exp(m_j) + \exp(-m_j)}, \tag{3}$$

$$x_j(k) = \sum_{i=1}^{q} a_{ji} x_i(k-1) + \sum_{h=1}^{p} b_{jh} n_h(k-1), \tag{4}$$

$$s_j = = \sum_{i=1}^{q} c_{ji} x_i(k), \tag{5}$$

$$z_j = \frac{\exp(s_j) - \exp(-s_j)}{\exp(s_j) + \exp(-s_j)}, \tag{6}$$

$$y_j(k) = \sum_{i=1}^{r} w^2{}_{ji} z_i, \tag{7}$$

where $w^1{}_{ji}$ is the weight between the *i*th input neuron and the *j*th neuron in the hidden layer 1, $w^2{}_{ji}$ is the weight between the *i*th neuron of the hidden layer 2 and the *j*th output neuron, and d_j is the bias of the *j*th hidden neuron.

Since our network can be represented by a state-space equation as shown in (1), we can analyze the characteristics, such as the stability, controllability and observability of the proposed network without much effort. For instance, it is well known that the stability of a dynamic system is sensitive to time delay feedback term [Cao et al. 06]; i.e., the system matrix **A** in (1). If all of the eigenvalues of matrix **A** are located in the unit circle, then we can ensure the stability of the proposed network. Based on this advantage, we have developed a system identification algorithm for the proposed Hammerstein-Wiener recurrent neural network in the next section.

3 System Identification Algorithm

In this section, we will introduce a system identification algorithm for identifying an unknown system automatically by the proposed Hammerstein-Wiener recurrent neural network. This algorithm can automatically perform the identification task using the input-output measurements of the nonlinear system. For most of the studies on system identification using Hammerstein-Wiener models, researchers usually used multi-stage approaches to establish their models; i.e., take each subsystem as an individual problem which can be realized separately. Here, we adopted the concept of multi-stage approaches to develop an identification algorithm for our proposed Hammerstein-Wiener model. The proposed hybrid Hammerstein-Wiener initialization algorithm (HHWIA) consists of three parts: 1) an active region boundary initialization algorithm for first static nonlinear subsystem, 2) a frequency domain eigensystem realization algorithm (FDERA) for the dynamic linear subsystem, and 3) a least-squares method for the second static nonlinear subsystem. Finally, we introduce a recursive parameter learning algorithm to optimize the overall system performance.

3.1 Active Region Boundary Initialization Algorithm

The underlying idea of the active region boundary initialization algorithm is to select the parameters so that all activation functions in the first static nonlinear subsystem can always be operated in an active region. With this objective, we need to find a better parameter set for the Hammerstein-Wiener recurrent neural network, because the change of the derivative in an active region is more significant than that in a saturation region. Furthermore, the active region boundary initialization algorithm can avoid the network getting stuck in the beginning of the recursive learning algorithm, and thus the network is expected to reach a desirable local minimum using the parameter optimization algorithm with fewer training epochs.

This algorithm is used to search the parameters $w^1{}_{ij}$ and d_j in (2) to ensure that the outputs of the neurons can be operated in an active region. Here, we follow the suggestion of Yam and Chow [Yam and Chow 01]: the active region is defined in the

region where the derivative is greater than one-twentieth of the maximum derivative. Also, we assume that the weights w^1_{ij} are *independent and identically distributed* (*i.i.d.*) uniform random variables within the range $[-w^1_{max}, w^1_{max}]$. Then we can find the active region of hyperbolic tangent sigmoid functions should be:

$$\left|m_j(k)\right| \le \cosh^{-1}\left(\frac{1}{20}\right)^{-\frac{1}{2}} = 2.178. \quad (\mathbf{8})$$

where $m_j(k)$ is defined in (2). If $m_j(k)$ is located out of the region, the training process will be stuck in the saturation region and thus the training speed will be slow. Now, in order to guarantee the neurons operate within the boundaries for all input data, the maximum possible Euclidean distance D^{max} among input data points must be within the boundaries. With this property, we can derive the maximum value of w^1_{ji} and d_j in the following equations.

$$w^1_{max} = \frac{4.356}{D^{max}}\sqrt{\frac{3}{p}}, \quad (\mathbf{9})$$

$$d_j = -\sum_{i=1}^{P} c_i^{bound} w^1_{ji}, \quad (\mathbf{10})$$

where

$$\mathbf{C}^{bound} = 0.5\left(\mathbf{u}^{max} - \mathbf{u}^{min}\right) = \left[c_1^{bound}, \ldots, c_p^{bound}\right]^T, \quad (\mathbf{11})$$

and $\mathbf{u}^{max}$ and $\mathbf{u}^{min}$ are the upper and lower bounds of the input data. Next, we shall construct the dynamic linear subsystem of the Hammerstein-Wiener recurrent neural network according to this static nonlinear subsystem.

3.2 Frequency Domain Eigensystem Realization Algorithm

After the first static nonlinear subsystem is initialized, we shall focus on the initialization of the dynamic linear subsystem. In this study, the frequency domain eigensystem realization algorithm (FDERA) will be applied to determine a reasonable system size for network and realize the corresponding state-space equation for the dynamic linear subsystem. This algorithm is an extension of the eigensystem realization algorithm (ERA) [Juang and Pappa 85] and is expected to improve the disadvantages of the ERA. In the early 1960s, because a lot of control approaches were developed based on state-space models, a large amount of algorithms were developed to solve the state-space realization problems. To name a few, Ho and Kalman [Ho and Kalman 65] introduced an important principle of minimum realization theory, which realizes a state-space model by the Hankel matrix constructed by a sequence of Markov parameters of the system. Next, the well-known ERA, proposed by Juang and Pappa [Juang and Pappa 85], was developed to realize the unknown system with noisy input-output measurements, and the algorithm provided accurate parameter estimation and system order determination for multivariable linear state-space models. But this algorithm suffers some

disadvantages. This is due to the ERA takes the pulse response of the system as its Markov parameters. For a complex system, it is very difficult to excite all the modes by a single pulse input. Even if we can excite the system by a single pulse input, the Markov parameters and Hankel matrices will become too large to compute its singular value decompositions [Medina et al. 94, Quan 94]. Moreover, for some systems, such as natural phenomena, we may not obtain their system pulse responses, and thus we cannot apply the ERA to these systems [Quan 94]. To overcome these deficiencies, Juang and Suzuki [Juang and Suzuki 88] combined the frequency response function with the concept of the ERA. The algorithm derives the Markov parameters based on the data of the frequency response function first, and then realizes the state-space model according to the constructed Hankel matrix. The proposed algorithm not only conquers the drawbacks of the original ERA but also inherits the properties of the ERA. The algorithm achieves better performance in realizing multivariable linear state-space models than the original ERA. Some advantages of the ERA in frequency domain are summarized as follows. First, the ERA in frequency domain can be applied to the systems which cannot obtain the pulse response by experiments. Second, the computation time for identification can be substantially reduced. A copious time sequence for the ERA can be substituted by finite data in frequency domain, thus the proposed Hankel matrix for the ERA in the frequency domain will be smaller than that of the original ERA. Finally, the system size can be determined by cutting off the relatively insignificant singular values of the Hankel matrix, and thus we can avoid using a trial-and-error approach to find a suitable size for the proposed model.

This FDERA starts with the transformation of the original input-output patterns [$\mathbf{u}(k)$, $\mathbf{y}(k)$] to another input-output patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] for dynamic linear subsystem, where $\hat{\mathbf{u}}(k)$ is the output of the first static nonlinear subsystem. Note that the acquired input-output patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] can be regarded as the data with minimum nonlinearity, because the first static nonlinear subsystem is expected to eliminate the nonlinearity of the proposed unknown system. Thus we can use the patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] to obtain a linear state-space model for the dynamic linear subsystem. Subsequently, the patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] are mapped to the frequency domain to evaluate the spectral density, frequency response function and Markov parameters. The Fourier transformation will be applied to map the data from the time domain into the frequency domain. The transformations between the time domain input-output patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] and the frequency domain input-output patterns [$\mathbf{U}(\omega)$, $\mathbf{Y}(\omega)$] are:

$$\mathbf{U}(\omega) = \frac{1}{l}\sum_{k=0}^{l-1} \hat{\mathbf{u}}(k) e^{-j(\frac{2\pi\omega}{l})k}, \tag{12}$$

$$\mathbf{Y}(\omega) = \frac{1}{l}\sum_{k=0}^{l-1} \mathbf{y}(k) e^{-j(\frac{2\pi\omega}{l})k}, \tag{13}$$

where l is the length of the time domain pattern. With the frequency domain input-output patterns, we can evaluate the corresponding spectral density for the dynamic linear system, where the spectral density between the ith input and the jth input is defined as:

$$S_{u^{(i)}u^{(j)}}(\omega) = U^{(i)}(\omega)U^{*(j)}(\omega), \tag{14}$$

where $U^{*(j)}(\omega)$ is the conjugate of $U^{(j)}(\omega)$. Similarly, we can define the spectral density between the *i*th output and the *j*th input as:

$$S_{y^{(i)}u^{(j)}}(\omega) = Y^{(i)}(\omega)U^{*(j)}(\omega). \tag{15}$$

Thus, for a MIMO system with m inputs and n outputs, we can calculate its frequency response function $\mathbf{G}(\omega)$ with the spectral density:

$$\begin{aligned}\mathbf{G}(\omega) &= \mathbf{S}_{yu}(\omega)\mathbf{S}_{uu}^{-1}(\omega) \\ &= \begin{bmatrix} S_{y^{(1)}u^{(1)}}(\omega) & S_{y^{(1)}u^{(2)}}(\omega) & \cdots & S_{y^{(1)}u^{(m)}}(\omega) \\ S_{y^{(2)}u^{(1)}}(\omega) & S_{y^{(2)}u^{(2)}}(\omega) & \cdots & S_{y^{(2)}u^{(m)}}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ S_{y^{(n)}u^{(1)}}(\omega) & S_{y^{(n)}u^{(j)}}(\omega) & \cdots & S_{y^{(n)}u^{(m)}}(\omega) \end{bmatrix} \\ &\quad \times \begin{bmatrix} S_{u^{(1)}u^{(1)}}(\omega) & & & 0 \\ & S_{u^{(2)}u^{(2)}}(\omega) & & \\ & & \ddots & \\ 0 & & & S_{u^{(m)}u^{(m)}}(\omega) \end{bmatrix}^{-1},\end{aligned} \tag{16}$$

where $\mathbf{S}_{yu}(\omega)$ is an $n \times m$ matrix and $\mathbf{S}_{uu}(\omega)$ is an $m \times m$ diagonal matrix. Once we get the frequency response function of the system, we can compute the Markov parameters of the system simultaneously, where the Markov parameters $\mathbf{M}$ is the inverse Fourier transform of the frequency response function $\mathbf{G}$:

$$\mathbf{M} = \sum_{\omega=0}^{\infty} \mathbf{G}(\omega) e^{j(\frac{2\pi\omega}{l})k}, \tag{17}$$

Now, with these Markov parameters, we can form the Hankel matrix and perform the ERA to realize the proposed dynamic linear system. Before performing the ERA, we shall explain the meaning of Markov parameters in the proposed algorithm. The target representation of the dynamic linear subsystem can be written as:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\hat{\mathbf{u}}(k), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k),\end{aligned} \tag{18}$$

where we assume that the initial conditions, $\mathbf{x}(0) = 0$. The relationship between Markov parameters and the system is:

$$\mathbf{M} = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{\tau-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_\tau \end{bmatrix}, \tag{19}$$

where τ is the length of Markov parameters. Next, to apply the ERA, we form a generalized Hankel matrix $\mathbf{H}(0)$ and a shifted Hankel matrix $\mathbf{H}(1)$ by the Markov parameters as follows:

$$\mathbf{H}(0)=\begin{bmatrix}\mathbf{Y}_1 & \mathbf{Y}_2 & \cdots & \mathbf{Y}_{\beta}\\ \mathbf{Y}_2 & \mathbf{Y}_3 & \cdots & \mathbf{Y}_{\beta+1}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{Y}_{\alpha} & \mathbf{Y}_{\alpha+1} & \cdots & \mathbf{Y}_{\alpha+\beta-1}\end{bmatrix}_{\alpha p\times\beta r}=\begin{bmatrix}\mathbf{C}\\ \mathbf{CA}\\ \vdots\\ \mathbf{CA}^{\alpha-1}\end{bmatrix}\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{\beta-1}\mathbf{B}\end{bmatrix}, \tag{20}$$

$$\mathbf{H}(1)=\begin{bmatrix}\mathbf{Y}_2 & \mathbf{Y}_3 & \cdots & \mathbf{Y}_{\beta+1}\\ \mathbf{Y}_3 & \mathbf{Y}_4 & \cdots & \mathbf{Y}_{\beta+2}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{Y}_{\alpha+1} & \mathbf{Y}_{\alpha+2} & \cdots & \mathbf{Y}_{\alpha+\beta}\end{bmatrix}_{\alpha p\times\beta r}=\begin{bmatrix}\mathbf{C}\\ \mathbf{CA}\\ \vdots\\ \mathbf{CA}^{\alpha-1}\end{bmatrix}\mathbf{A}\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{\beta-1}\mathbf{B}\end{bmatrix}, \tag{21}$$

where p and r are the number of system input and output and α and β are self-defined integers. Subsequently, the Hankel matrix $\mathbf{H}(0)$ is decomposed by the singular value decomposition:

$$\mathbf{H}(0)=\mathbf{P}\Sigma\mathbf{Q}^T, \tag{22}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are orthogonal matrices and Σ is a diagonal matrix with its singular values $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_{\varphi}>0$, which φ is the index in the diagonal matrix. Thus, we can determine the order of the linear system by examining the singular values of the Hankel matrix $\mathbf{H}(0)$. The first relatively large q singular values will be selected to compute the system matrix $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, and q is the order of the network. The decomposition of $\mathbf{H}(0)$ then becomes

$$\mathbf{H}(0)=[\mathbf{P}_q\quad\mathbf{P}_0]\begin{bmatrix}\Sigma_q & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_q^T\\ \mathbf{Q}_0^T\end{bmatrix}=\mathbf{P}_q\Sigma_q\mathbf{Q}_q^T=\left[\mathbf{P}_q\Sigma_q^{1/2}\right]\left[\Sigma_q^{1/2}\mathbf{Q}_q^T\right], \tag{23}$$

where $\mathbf{P}_q$ and $\mathbf{Q}_q$ are the matrices formed by the first q columns of $\mathbf{P}$ and $\mathbf{Q}$, and $\Sigma_q = diag[\sigma_1, \sigma_2, \ldots, \sigma_q]$. Next, according to (19) and (22), we can obtain the following equalities:

$$\begin{bmatrix}\mathbf{C}\\ \mathbf{CA}\\ \vdots\\ \mathbf{CA}^{\alpha-1}\end{bmatrix}=\left[\mathbf{P}_q\Sigma_q^{1/2}\right], \tag{24}$$

and

$$\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{\beta-1}\mathbf{B}\end{bmatrix}=\left[\Sigma_q^{1/2}\mathbf{Q}_q^T\right]. \tag{25}$$

Therefore,

$$\mathbf{C}=\text{first } r \text{ rows of } \mathbf{P}_q\Sigma_q^{1/2}, \tag{26}$$

and

$$\mathbf{B} = \text{first } p \text{ rows of } \mathbf{\Sigma}_q^{1/2}\mathbf{Q}_q^T. \tag{27}$$

Based on (21), (24) and (25), the shifted Hankel matrix can be defined as

$$\mathbf{H}(1) = \left[\mathbf{P}_q\mathbf{\Sigma}_q^{1/2}\right]\mathbf{A}\left[\mathbf{\Sigma}_q^{1/2}\mathbf{Q}_q^T\right]. \tag{28}$$

Consequently, we can obtain the realization of **A** by the following equation:

$$\mathbf{A} = \left[\mathbf{\Sigma}_q^{-1/2}\mathbf{P}_q^T\right]\mathbf{H}(1)\left[\mathbf{Q}_q\mathbf{\Sigma}_q^{-1/2}\right]. \tag{29}$$

Finally, the system matrix **A**, **B**, and **C** in (26), (27), and (29) will be assigned to the network as the initial parameters of the Hammerstein-Wiener model. Next, with the constructed first static nonlinear subsystem and the dynamic linear subsystem, we will find a suitable parameter set for the second static nonlinear subsystem.

3.3 Least-Squares Method

We shall use a least-squares method to reduce the error of the proposed initialized network. Although the former two algorithms can help us closely mimic the property of the unknown dynamic system, there remain some errors between the desired output and the output of the former two subsystems. Thus, the objective of the second static nonlinear subsystem is to use a least-squares method to eliminate these errors. We first computed the output of the former two subsystems, **x**, and the patterns [**x**, **y**] are used to evaluate the parameters in the second nonlinear subsystem by:

$$\mathbf{w}^2 = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}, \tag{30}$$

where the acquired parameters $\mathbf{w}^2$ is an $r \times r$ matrix and are assigned to the weights of the second static nonlinear subsystem.

After we use the least-squares method to search the parameters of the second static nonlinear subsystem, the HHWIA is completed and the network is established simultaneously. We now summarize the HHWIA as the following steps.

Step 1. Construct the first static nonlinear subsystem by the active region boundary initialization algorithm.

Step 2. Obtain the training input-output patterns [$\hat{\mathbf{u}}(k)$, $\mathbf{y}(k)$] for the linear dynamic subsystem through the output of first static nonlinear subsystem.

Step 3. Evaluate the frequency response function of the dynamic linear subsystem.

Step 4. Estimate the Markov parameters from the frequency response function in Step 3.

Step 5. Use the estimated Markov parameters to construct a generalized Hankel matrix **H**(0) and a shifted Hankel matrix **H**(1).

Step 6. Find the decomposition of the generalized Hankel matrix **H**(0) by employing the singular value decomposition technique.

Step 7. Estimate the system order according to the singular values in the diagonal matrix **Σ**.

Step 8. Realize the matrices **A**, **B** and **C** by the matrix **H**(0) and **H**(1) via (20)-(29).

Step 9. Construct the second static nonlinear subsystem by the least-squares method according to the input-output patterns [**x**, **y**] of the second static nonlinear subsystem.

Upon the completion of the HHWIA, we can establish our network to identify the unknown dynamic system with good performance. To closely identify the unknown system, we have developed a recursive learning algorithm to fine-tune the parameters of the network.

3.4 Recursive Recurrent Learning Algorithm

In this learning phase, we derived a recursive recurrent learning algorithm for the Hammerstein-Wiener recurrent neural network based on the concept of ordered derivatives [Werbos 74]. The proposed learning algorithm is expected to tune the whole network parameters and thus improve the overall network performance. To ease our discussion, the optimization target is characterized to minimize the following error function with respect to adjustable parameters (**v**) of a MISO network.

$$E(\mathbf{v},k) = \tfrac{1}{2}(y_d(k) - y(k))^2 = \tfrac{1}{2}e(k)^2, \tag{31}$$

where $e(k) = y_d(k) - y(k)$, and $y_d(k)$ and $y(k)$ are the desired output of the unknown system and the actual output of the network. The parameter **v** represents all the adjustable parameters in the proposed network. The general update rule for all parameters is shown as follows:

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \xi(-\frac{\partial^+ E}{\partial \mathbf{v}}). \tag{32}$$

where ξ is the learning rate and $\partial^+ E/\partial v$ is the ordered derivative that considers the direct and indirect effects of changing a structure parameter. The adjustable parameter **v** contains six parameters in (2) to (7), a_{ji}, b_{jh}, c_{ij}, w^1_{ji}, w^2_{ji} and d_j. Now we derive the update rule of a_{ji} as an example. According to (32), the update rule of a_{ji} is derived as:

$$a_{ji}(k) = a_{ji}(k-1) + \xi_{ac}(-\frac{\partial^+ E(k)}{\partial a_{ji}}), \tag{33}$$

where $\frac{\partial^+ E(k)}{\partial a_{ji}} = \frac{\partial E(k)}{\partial x_j(k)} \frac{\partial^+ x_j(k)}{\partial a_{ji}}$. Also, according to (4) to (7), and (31),we can obtain

$$\frac{\partial E(k)}{\partial x_j} = -c_j(k-1)\frac{4}{(\exp(s)+\exp(-s))^2}e(k), \tag{34}$$

and

$$\frac{\partial^+ x_j(k)}{\partial a_{ji}} = \frac{\partial x_j(k)}{\partial a_{ji}} + \frac{\partial x_j(k)}{\partial x_j(k-1)}\frac{\partial^+ x_j(k-1)}{\partial a_{ji}}, \tag{35}$$

where $\partial x_j(k)/\partial a_{ji} = x_i(k-1)$, $\partial x_j(k)/\partial x_j(k-1) = a_{jj}(k-1)$ and $\partial x_j(1)/\partial a_{ji} = x_i(0)$ when k = 1. Also, the value of $\partial^+ x_j/\partial a_{ji}$ is set to zero initially. This value will be accumulated recursively as the error signal generated in each training time step.

Similarly, the update rules for b_{jh}, c_{ij}, w^1_{ji}, w^2_{ji} and d_j can be obtained in the same process. Also, We shall set different the learning rates (that is ξ and ξ_{ac}), for the parameters, of w^1_{ji}, w^2_{ji} and d_j , and the parameters, a_{ji}, b_{jh}, and c_{ij}, because the system is more sensitive to the parameters in the dynamic layer than the parameters in the hidden layer [Cao et al. 06].

4 Simulation Results

To validate the performance of the Hammerstein-Wiener recurrent neural network with its identification algorithm, we have conducted extensive computer simulations on benchmark examples. Also, to demonstrate the merit of our network and identification algorithm, we have made some comparisons with other existing notable approaches.

Example 1: Identification of nonlinear dynamic system. In this example, we first constructed a nonlinear plant with multiple time delays [Juang and Lin 99, Narendra and Parthasarathy 90].

$$y(k+1) = \alpha \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2)-\beta)+u(k)}{1+y(k-1)^2+y(k-2)^2}, \qquad (\mathbf{36})$$

where the above plant will become unstable when $\beta > 1$. From the above equation, it is obvious that the proposed system output is affected by three previous outputs and two previous inputs. In [Narendra and Parthasarathy 90], all of these five variables are fed into the feedforward neural network to determine the next output $y(k+1)$. Here, to provide a fair comparison with other existing models, we follow the method proposed by Juang and Lin [Juang and Lin 99]; that is, the nonlinear plant parameters (α, β) are assigned to (1, 1) and only $u(k)$ and $y(k)$ are used as the network inputs. Also, for the training data, an *i.i.d.* uniform sequence within [-2, 2] was generated as the input signal $u(k)$ for the half of the training time steps, and a sinusoid signal, $1.05\sin(\pi k/45)$, was subsequently given for the remaining training time steps. The training data are first used to initialize the proposed network and then used for training the network. Finally, the testing data is defined as

$$u(k) = \begin{cases} \sin(\pi k/25) & 0< k <250, \\ 1.0 & 250 \le k <500, \\ -1.0 & 500 \le k <750, \\ 0.3\sin(\pi k/25)+0.1\sin(\pi k/32) & \\ \quad +0.6\sin(\pi k/10) & 750 \le k <1000. \end{cases} \qquad (\mathbf{37})$$

To initialize the Hammerstein-Wiener recurrent neural network for identifying nonlinear dynamic system (36), we first constructed the first static nonlinear subsystem by the active region boundary initialization. The input series $u(k)$ and $y(k)$

are transformed to the dynamic linear subsystem through the first static nonlinear subsystem. Subsequently, the FDERA is applied to determine the state-space equation in the dynamic linear subsystem according to the transformed input series $\hat{u}(k)$ and the output $y(k)$. Moreover, according to the singular values evaluated from FDERA, only two neurons are constructed in the dynamic layer; thus, there are totally 17 parameters in the proposed network. Finally, the process of HHWIA focuses on determining the initial parameter of the second static nonlinear subsystem, where the least-squares method is used to reduce the overall error of the proposed initial network. After the HHWIA, the initialized network will only be trained for one epoch, i.e., 9000 time steps, with the learning rate $\xi_{ac} = 0.006$ and $\xi = 0.06$. After the training, we can obtain the state-space model of the trained network as follows:

Model	Before-Training MSE	After-Training MSE	Testing MSE
Random Initialization	0.2081	0.0215	0.0134
HHWIA	0.0906	0.0057	0.0026

Table 1: Performance Comparisons of the Proposed Network with HHWIA and Random Initialization

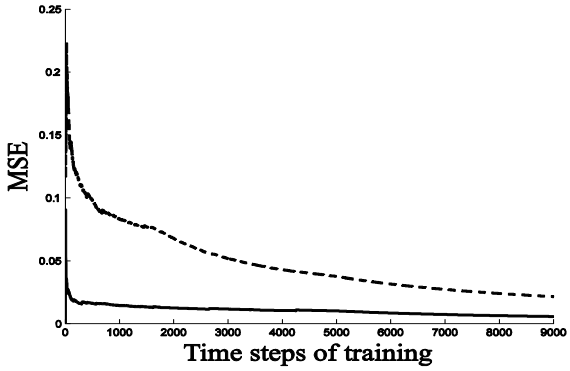

Figure 3: The learning curves of the proposed network with HHWIA (solid curve) and the random initialization (dotted curve).

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -0.0136 & 0.1637 \\ 0.2891 & -0.4424 \end{bmatrix} x(k) + \begin{bmatrix} -0.4405 & 0.3553 \\ -0.0878 & -0.1183 \end{bmatrix} N^1\left(u(k)\right), \\ y(k) &= N^2\left(\begin{bmatrix} -0.5860 & 0.0974 \end{bmatrix} x(k)\right), \end{aligned} \tag{38}$$

where the eigenvalues of the system matrix **A** are (0.0775, -0.5334), which indicates our network is stable after the training. Also, to demonstrate the effectiveness of the HHWIA, we have compared the HHWIA with the random initialization method. In the random initialization method, all of the weights and biases are generated randomly.

Figure 3 shows the learning curves of the Hammerstein-Wiener recurrent neural network with HHWIA and the random initialization. From this figure, it is obvious that the HHWIA can provide better initial condition and faster learning convergence than those of the random initialization. Also, the comparisons between the HHWIA and the random initialization are made in Table 1. Table 1 shows the average MSEs of the networks initialized by the random initialization method and our proposed HHWIA. Comparing the before-training and after-training MSEs, we can see that our proposed HHWIA outperforms the random initialization method very well. This validates the effectiveness of the proposed HHWIA. Finally, Fig. 4 shows the overall desired nonlinear dynamic system output and the output of the proposed network whose testing means square error (MSE) is 0.0026. The performance comparison with some existing notable models is summarized in Table 2, where the models are represented by the name of authors and the published year. In this table, we can see that the proposed network with only 17 parameters achieves a better performance.

Model	No. of Parameters	Training Time Steps	Testing MSE
Sastry *et al.*, 1994 [Sastry et al. 94]	81	62000	0.0752
Juang and Lin, 1999 [Juang and Lin 99]	30	9000	0.0441
Proposed Network	17	9000	0.0026

Table 2: Performance Comparisons of Hammerstein-Wiener Recurrent Neural Network with Other Existing Notable Models for Example 1

Plant Variations Caused by α and β	(1,1)	(1.03, 0.9)	(1.07, 0.8)	(1.11, 0.7)	(1.15, 0.6)
MSE for Testing Data (38)	0.0026	0.0027	0.0033	0.0041	0.0050

Table 3: Robustness Tests for Plant Variations

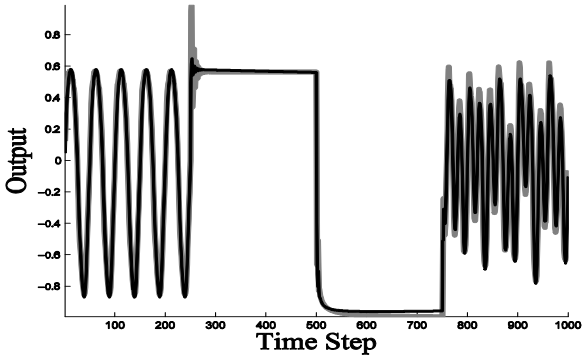

Figure 4: The desired output of nonlinear dynamic system (gray curve) and the output of proposed network (black curve) for Example 1.

To demonstrate the robustness of our identified network expressed in (38) to plant variations, we vary the plant parameters α from 1 to 1.15 and β from 1 to 0.6 in (36). Table 3 shows the results of the robustness tests. From this table, we can see that our network and its identification algorithm can tolerate the plant variations by a little sacrifice in system performance.

Example 2: Box and Jenkins time series. The data of Box and Jenkins time series was the recordings of a gas combustion process and have been widely used for evaluating the identification capability of networks [Kukolj and Levi 04, Wang and Rong 99]. The dataset contains 296 input-output samples, where the input variable is the input flow $u(k)$, and the output variable is the concentration of CO_2, $y(k)$, which is influenced by ten variables $y(t\text{-}1),\ldots, y(t\text{-}4), u(t\text{-}1), \ldots, u(t\text{-}6)$ [Kukolj and Levi 04]. Here, to conduct fair comparisons with other existing models, we followed the method of Kukolj and Levi [Kukolj and Levi 04]; that is, only $u(k\text{-}4)$ and $y(k\text{-}1)$ are used as the network inputs. The data are separated into two sets, the first 100 samples are used for the initialization and optimization of the proposed network, while the remaining 196 samples are used for testing.

The HHWIA is used to initialize the network and suggests that the system order of the dynamic subsystem is 2. Thus, after the initialization, we can obtain a model with a total 17 parameters. After we train the network for 100 epochs, we can acquire the state-space equation of the proposed network:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.2979 & 0.6124 \\ 0.7500 & 0.0393 \end{bmatrix} x(k) + \begin{bmatrix} 0.6692 & 0.3225 \\ -0.6008 & -0.1341 \end{bmatrix} N^1\left(u(k)\right), \\ y(k) &= N^2\left(\begin{bmatrix} -0.7881 & 0.8629 \end{bmatrix} x(k)\right), \end{aligned} \tag{39}$$

where the eigenvalues of the system matrix **A** are (0.8585, -0.5214), which means the system is stable after the training. Figure 5 depicts the desired output and the output of the trained proposed network. The MSE of the testing set is 0.1055. Also, the comparisons between the proposed network and some existing models are summarized in Table 4. From this table, we can confirm the effectiveness of our network and its identification algorithm.

Model	No. of Inputs	No. of Parameters	Testing 1 NMSE	Testing 2 NMSE	Training Epochs
Chellapilla and Rao, 1998 [Chellapilla and Rao 98]	-	16	0.113	0.1895	-
Jeong and Lee, 2000 [Jeong and Lee 00]	12	104	0.061	0.205	10000
Soltani, 2002 [Soltani 02]	10	-	0.076	0.23	-
Oh *et al.*, 2005 [Oh et al. 05]	3	-	0.099	0.168	-
Proposed Network	3	25	0.0799	0.167	10

Table 5: Performance Comparisons of Hammerstein-Wiener recurrent neural network with Other Existing Notable Models for Example 3

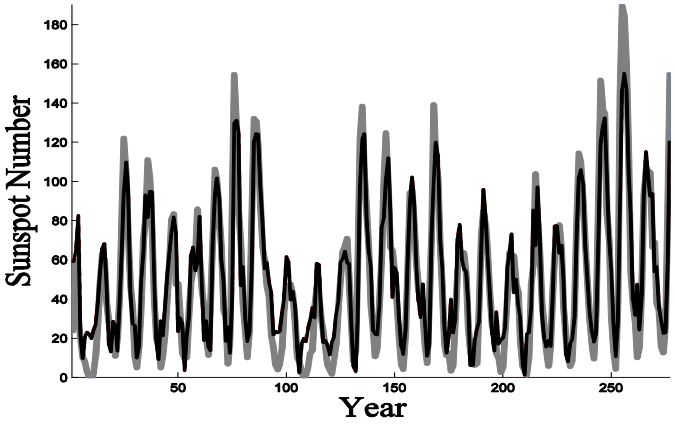

Figure 6: The desired Sunspot time series (gray curve) and the output of Hammerstein-Wiener recurrent neural network (black curve) for Example 3.

Example 3: Sunspot Time Series. The sunspot time series contains the annual average number of sunspots during the period from 1700 to 1979. The series is used to do the one-step-ahead prediction and the series is divided into one training set (years 1700-1920), and two testing sets (years 1921-1955 and years 1956-1979) [Chellapilla and Rao 98, Jeong and Lee 00, Oh et al. 05, Soltani 02]. To show the effectiveness of the proposed network and its identification capability, we followed the method in [Oh et al. 05] and only used three past inputs $u(k\text{-}1)$, $u(k\text{-}2)$, and $u(k\text{-}3)$ to predict the next output $y(k)$, instead of using 10 past inputs to predict the next output [Soltani 02]. According to the input-output patterns, the HHWIA constructs a Hammerstein-Wiener recurrent neural network with two neurons in its dynamic layer and 25 parameters in the network. Similar to Example 1, the proposed network was

trained for 10 epochs with the learning rate ξ_{ac} = 0.006 and ξ = 0.06 and the corresponding state-space equation for the network is

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.0475 & 0.5277 \\ 0.4238 & 0.4608 \end{bmatrix} x(k) \\ &\quad + \begin{bmatrix} -0.6754 & -0.2229 & -0.4589 \\ 0.1898 & 0.0368 & -0.1627 \end{bmatrix} N^1\left(u(k)\right), \\ y(k) &= N^2\left(\begin{bmatrix} -0.6584 & 0.3720 \end{bmatrix} x(k)\right). \end{aligned} \tag{40}$$

Figure 6 shows the prediction result of the sunspot time series. The normalized mean square error (NMSE) of the first testing set is 0.0799 and the second testing set is 0.1673. The performance comparisons with some existing models are summarized in Table 5. From this Table, we can confirm that our network is capable of making good predictions with fewer inputs, parameters and training epochs. From the above simulations, we can see that the proposed approach performs very well; however, our approach may fail to identify the system orders for some chaotic systems or systems with high nonlinearity, and thus may not perform as well as the above examples.

5 Conclusion

A novel Hammerstein-Wiener recurrent neural network with a systematic identification algorithm has been proposed for nonlinear dynamic system identification problems. The advantages of our approach include: 1) the integration of three subsystems into a simple connectionist neural network whose output can be expressed by a nonlinear transformation of a linear state-space equation, and 2) the systematic identification algorithm, including the HHWIA and the recursive recurrent learning algorithm, is capable of constructing the proposed network within a compact structure for a dynamic nonlinear system. Computer simulations on benchmark examples have validated the effectiveness and superiority of our proposed network and its identification algorithm. This study has reached a limited success; however, there still have some questions regarding how to ensure the efficiency and simplicity of stable learning algorithm, how to use the stable learning algorithm in a general block-oriented network, and how to balance between stability and performance, etc. These research problems deserve further investigations.

References

[Abonyi et al. 99] Abonyi, J., Nagy, L., Szeifert, F.: "Adaptive Fuzzy Inference System and Its Application in Modelling and Model Based Control"; Chemical Engineering Research and Design, 77 (1999), 281–290.

[Abrahamsson et al. 07] Abrahamsson, R., Kay, S. M., Stoica, P.: "Estimation of the Parameters of a Bilinear Model with Applications to Submarine Detection and System Identification"; Digital Signal Processing, 17 (2007), 756-773.

[Arto et al. 01] Arto, V., Hannu, P., Halm, A.: "Modeling of Chromatographic Separation Process with Wiener-MLP Representation"; Journal of Process Control, 11 (2001), 443-458.

[Bai and Li 04] Bai, E. W., Li, D.: "Convergence of the Iterative Hammerstein System Identification Algorithm"; IEEE Trans. Automatic Control, 49 (2004), 1929-1940.

[Cao et al. 06] Cao, J., Yuan, K., Li, H. X.: "Global Asymptotic Stability of Recurrent Neural Networks with Multiple Discrete Delays and Distributed Delays"; IEEE Trans. Neural Networks, 17 (2006), 1646-1651.

[Chen et al. 97] Chen, G., Chen, Y., Ogmen, H.: "Identifying Chaotic Systems via a Wiener-Type Cascade Model"; IEEE Control System Magazine, 17 (1997), 29-36.

[Chellapilla and Rao 98] Chellapilla, K., Rao, S. S.: "Optimization of Bilinear Time Series Models Using Fast Evolutionary Programming"; IEEE Trans. Signal Processing Letters, 5 (1998), 39-42.

[Ho and Kalman 65] Ho, B. L., Kalman, R. E.: "Effective Construction of Linear State-Variable Models from Input/Output Data"; Proc. of the 3rd Annual Allerton Conf. on Circuit and System Theory (1965), 449-459.

[Jeong and Lee 00] Jeong, S. Y., Lee, S. Y. : "Adaptive Learning Algorithms to Incorporate Additional Functional Constraints into Neural Networks"; Neurocomputing, 35 (2000), 73-90.

[Jia et al. 05] Jia, L., Chiu, M. S., Ge, S. S.: "A Noniterative Neuro-Fuzzy Based Identification Method for Hammerstein Processes"; Journal of Process Control, 15 (2005), 749-761.

[Juang and Lin 99] Juang, C. F., Lin, C. T.: "A Recurrent Self-Organizing Neural Fuzzy Inference Network"; IEEE Trans. Neural Network, 10 (1999), 828-845.

[Juang and Pappa 85] Juang, J. N., Pappa, R. S.: "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction"; J. Guidance, 8 (1985), 620-627.

[Juang and Suzuki 88] Juang, J. N., Suzuki, H.: "An Eigensystem Realization Algorithm in Frequency Domain for Modal Parameter Identification"; Journal of Vibration, Acoustics, Stress, and Reliability in Design, 110 (1988), 24-29.

[Kalafatis et al. 05] Kalafatis, A. D., Wang, L., Cluett, W. R.: "Linearizing Feedforward-Feedback Control of PH Processes Based on the Wiener Model"; Journal of Process Control, 15 (2005), 103-112.

[Kang et al. 00] Kang, S. J., Woo, C. H., Hwang, H. S., Woo, K. B.: "Evolutionary Design of Fuzzy Rule Base for Nonlinear System Modeling and Control"; IEEE Trans. Fuzzy Systems, 8 (2000), 37-45.

[Kukolj and Levi 04] Kukolj, D., Levi, E.: "Identification of Complex Systems Based on Neural and Takagi-Sugeno Fuzzy Model"; IEEE Trans. Systems, Man and Cybernetics, part B: Cybernetics, 34 (2004), 272-282.

[Kwak and Kim 05] Kwak, K. C., Kim, D. H.: "Adaptive Neuro-Fuzzy Networks with the Aid of Fuzzy Granulation"; IEICE Trans. Information and Systems, 88 (2005), 2189-2195.

[Medina et al. 94] Medina, E. A., Irwin, R. D., Mitchell, J. R., Bukley, A. P.: "MIMO System Identification Using Frequency Response Data"; The Journal of the Astronautical Sciences, 42 (1994), 113-129.

[Narendra and Parthasarathy 90] Narendra, K. S., Parthasarathy, K.: "Identification and Control of Dynamical Systems Using Neural Networks"; IEEE Trans. Neural Network, 1 (1990), 4-27.

[Oh et al. 05] Oh, S. K., Kim, M. S., Eom, T. D., Lee, J. J.: "Heterogeneous Local Model Networks for Time Series Prediction"; Applied Mathematics and Computation, 168 (2005), 164-177.

[Quan 94] Quan, R.: "System Identification Using Frequency Scanning and the Eigensystem Realization Algorithm"; Journal of Guidance, Control, and Dynamics, 17 (1994), 670-675.

[Sastry et al. 94] Sastry, P. S., Santharam, G., Unnikrishnan, K. P.: "Memory Neuron Networks for Identification and Control of Dynamical Systems"; IEEE Trans. Neural Network, 5 (1994), 306-319.

[Soltani 02] Soltani, S.: "On the Use of the Wavelet Decomposition for Time Series Prediction"; Neurocomputing, 48 (2002), 267-277.

[Wang and Rong 99] Wang, Y., Rong, G.: "A Self-Organizing Neural-Network-Based Fuzzy System"; Fuzzy Sets and Systems, 103 (1999), 1-11.

[Werbos 74] Werbos, P.: "Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences"; Ph.D. dissertation, Harvard Univ., Cambridge, MA (1974)

[Westwick and Kearney 01] Westwick, D. T., Kearney, R. E.: "Separable Least Squares Identification of Nonlinear Hammerstein Models Application to Stretch Reflex Dynamic"; Annals of Biomedical Engineering, 29 (2001), 707-718.

[Yam and Chow 01] Yam, Y. F., Chow, T. W. S.: "Feedforward Networks Training Speed Enhancement by Optimal Initialization of the Synaptic Coefficients"; IEEE Trans. Neural Networks, 12 (2001), 430-434.

[Zhu 02] Zhu, Y.: "Estimation of an N-L-N Hammerstein-Wiener Model"; Automatica, 38 (2002), 1607-1614.