Nonlinear System Identification Using Hammerstein-Wiener Neural Network and subspace algorithms

Maryam Ashtari Mahini1, Mohammad Teshnehlab2 and Mojtaba Ahmadieh khanehsar3

Received (2015-01-12)
Accepted (2015-03-12)

Abstract — Neural networks are applicable in identification from input-output data. In this report, we analyze the Hammerstein-Wiener models and identify them. The Hammerstein-Wiener systems are the simplest type of block-oriented nonlinear systems where the linear dynamic block is sandwiched in between two static nonlinear blocks, which appear in many engineering applications; the aim of nonlinear system identification by Hammerstein-Wiener neural network is finding model order, state matrices and system matrices. We propose a robust approach for identifying the nonlinear system by neural network and subspace algorithms. The subspace algorithms are mathematically well-established and non-iterative identification process. The use of subspace algorithm makes it possible to directly obtain the state space model. Moreover the order of state space model is achieved using subspace algorithm. Consequently, by applying the proposed algorithm, the mean squared error decreases to 0.01 which is less than the results obtained using most approaches in the literature.

Index Terms — Neural Network, nonlinear system identification, Hammerstein-Wiener model, state space and subspace identification.

I. INTRODUCTION

Recently, identification of block-oriented models has received more attention. Because most of the physical systems are nonlinear so we use nonlinear models to describe a system. Hammerstein-Wiener recurrent neural network (H-W) models and their combinations are block-oriented models and commonly used in nonlinear models.

Hammerstein model consists of a static nonlinearity followed by a linear dynamic system [7], Wiener model consists of a linear dynamic system followed by a static nonlinearity [30]. Hammerstein-Wiener model (H-W) consists of a linear dynamic subsystem that is sandwiched in between two nonlinear static subsystems[31]. For these models, it is assumed that only input and output signals of the models are measurable.

In order to have an efficient model, the model has to be simple and easily presentable. In comparison to other models, block-oriented models are the most efficient models because they consist of linear and nonlinear blocks separately [22]. In the following, three kinds of block-oriented models are mentioned:

In Wiener model that is depicted in “Figure 1. “, a linear dynamic block (G) is placed before a static nonlinear block. Inputs and outputs are measurable, but (state variable) is not measurable [30].

In Hammerstein model which is depicted in "Figure 2. “, static subsystem receives as input and transforms to . All dynamics models by linear discrete transform function . is output[7]. In Hammerstein-Wiener model a linear dynamic block (LS) is sandwiched in between a nonlinear static block (N1) and another nonlinear static

1- Dept of Computer Engineering. Science and Research Branch Islamic Azad University, Tehran, Iran. (Sha_ashtari@yahoo.com).
2- Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran (teshnehlab@eetd.kntu.ac.ir).
3- Department of Control Engineering, Semnan University, Semnan, Iran (ahmadieh@profs.semnan.ac.ir).

I. INTRODUCTION

Recently, identification of block-oriented models has received more attention. Because most of the physical systems are nonlinear so we use nonlinear models to describe a system. Hammerstein-Wiener recurrent neural network (H-W) models and their combinations are block-oriented models and commonly used in nonlinear models.

Hammerstein model consists of a static nonlinearity followed by a linear dynamic system [7], Wiener model consists of a linear dynamic system followed by a static nonlinearity [30]. Hammerstein-Wiener model (H-W) consists of a linear dynamic subsystem that is sandwiched in between two nonlinear static subsystems[31]. For these models, it is assumed that only input and output signals of the models are measurable.

In order to have an efficient model, the model has to be simple and easily presentable. In comparison to other models, block-oriented models are the most efficient models because they consist of linear and nonlinear blocks separately [22]. In the following, three kinds of block-oriented models are mentioned:

In Wiener model that is depicted in “Figure 1. “, a linear dynamic block (G) is placed before a static nonlinear block. Inputs and outputs are measurable, but (state variable) is not measurable [30].

In Hammerstein model which is depicted in "Figure 2. “, static subsystem receives as input and transforms to . All dynamics models by linear discrete transform function . is output[7]. In Hammerstein-Wiener model a linear dynamic block (LS) is sandwiched in between a nonlinear static block (N1) and another nonlinear static

I. INTRODUCTION

Recently, identification of block-oriented models has received more attention. Because most of the physical systems are nonlinear so we use nonlinear models to describe a system. Hammerstein-Wiener recurrent neural network (H-W) models and their combinations are block-oriented models and commonly used in nonlinear models.

Hammerstein model consists of a static nonlinearity followed by a linear dynamic system [7], Wiener model consists of a linear dynamic system followed by a static nonlinearity [30]. Hammerstein-Wiener model (H-W) consists of a linear dynamic subsystem that is sandwiched in between two nonlinear static subsystems[31]. For these models, it is assumed that only input and output signals of the models are measurable.

In order to have an efficient model, the model has to be simple and easily presentable. In comparison to other models, block-oriented models are the most efficient models because they consist of linear and nonlinear blocks separately [22]. In the following, three kinds of block-oriented models are mentioned:

In Wiener model that is depicted in “Figure 1. “, a linear dynamic block (G) is placed before a static nonlinear block. Inputs and outputs are measurable, but (state variable) is not measurable [30].

In Hammerstein model which is depicted in "Figure 2. “, static subsystem receives as input and transforms to . All dynamics models by linear discrete transform function . is output[7]. In Hammerstein-Wiener model a linear dynamic block (LS) is sandwiched in between a nonlinear static block (N1) and another nonlinear static

I. INTRODUCTION

Recently, identification of block-oriented models has received more attention. Because most of the physical systems are nonlinear so we use nonlinear models to describe a system. Hammerstein-Wiener recurrent neural network (H-W) models and their combinations are block-oriented models and commonly used in nonlinear models.

Hammerstein model consists of a static nonlinearity followed by a linear dynamic system [7], Wiener model consists of a linear dynamic system followed by a static nonlinearity [30]. Hammerstein-Wiener model (H-W) consists of a linear dynamic subsystem that is sandwiched in between two nonlinear static subsystems[31]. For these models, it is assumed that only input and output signals of the models are measurable.

In order to have an efficient model, the model has to be simple and easily presentable. In comparison to other models, block-oriented models are the most efficient models because they consist of linear and nonlinear blocks separately [22]. In the following, three kinds of block-oriented models are mentioned:

In Wiener model that is depicted in “Figure 1. “, a linear dynamic block (G) is placed before a static nonlinear block. Inputs and outputs are measurable, but (state variable) is not measurable [30].

In Hammerstein model which is depicted in "Figure 2. “, static subsystem receives as input and transforms to . All dynamics models by linear discrete transform function . is output[7]. In Hammerstein-Wiener model a linear dynamic block (LS) is sandwiched in between a nonlinear static block (N1) and another nonlinear static
block (N2)” Figure 3. “[20], [2].

The structure of this network can be mapped into a state-space equation, so the determining the network structure is equivalent to finding system order[19], [12]. The subspace identification is represented in 1960s. It is designed especially for the time-invariant systems and MIMO system which has state space model form. Subspace algorithms estimate the state-space models dynamic and because of using numerically reliable matrices, linear algebra, projection, SVD and QR[18], so they are quick, stable, reliable, non-iterative, efficient, simple, easy to interpret and convenient for estimation, filtering, prediction and control [9],[34]. So, in this paper we use subspace algorithm for representation of the linear part of the system to speed up the computational rate and accuracy.

There exists a lot of works on nonlinear system identification which use block-oriented models. For example estimating the formal information of neurons[27], using ARMA model for dynamic linear block and a multilayer feed forward neural network to model the static nonlinear[13], least square and SVD for Hammerstein model[11],[8], recursive identification for Hammerstein system with state space model[33], eigensystem realization algorithm(ERA) for accurate parameter estimation and the system order determination[23], over parameterization and iterative methods[22], iterative approaches[20][29], frequency-domain method[24], subspace method[17],[18],[25],[26], stochastic algorithm[6], blind approaches[4], magnetosphere identification[3], constructing a model for ionospheric dynamics[16], using Genetic algorithm for H-W identification[14], initializing parameters and order determination by Lipchitz[1], fully automated recurrent neural network[31], using the spectral magnitude matching method[32], using a new maximum-likelihood based method[33].

H-W models have been used in biomedical application, heat exchanger, electrical drive, thermal microsystem, physiological system[9], sticky control valves, solid oxide fuel cells[21], submarine detection[28], RF power amplifier modeling[10] and signal processing application[34].

In this paper, in section II and III we introduce Hammerstein-wiener neural network and parameter initialization (first subsystem). In addition we identify the first subsystem parameters (bias and weight), second subsystem parameters (system order, system and state matrices), and third and fourth subsystems parameters (weights) in detail. In section IV, we present a proposed identifying algorithm based on Hammerstein-Wiener nonlinear recurrent neural network. In section V, computer simulation and comparisons with some approaches are provided. Finally, conclusions and future works are given in section VI.

II. SUBSPACE IDENTIFICATION ALGORITHMS

Subspace identification algorithms include system theory (realization theory), statistics, optimization and linear algebra (projection and singular value decompositions) are used to estimate the dynamics of state-space models. These quick and reliable algorithms are called “subspace algorithms”, because the dynamics can be estimated based on available input/output measurements, known matrices row space and column space [25],[26]. In recent years, numerical subspace algorithms for linear time invariant identification have been noticed a lot. All of these algorithms identify the system in the state-space form by input/output data projection. Subspace identification algorithms are divided into deterministic, stochastic and deterministic-stochastic algorithms. One of the deterministic-stochastic, non-iterative and convergent algorithms is “N4SID “. Moreover, this algorithm is stable, because it uses linear algebra methods such as SVD and QR[25],[26].

In classic identification algorithms, parameter initialization and having pre-knowledge about system model, controllable and observable indexes are necessary. But, in subspace algorithms, system order is the only parameter. In these algorithms, input and output data are available, then finding out system order, system matrices(A,B,C,D) are desirable [25],[26].

In 1996, Verhagen and Vest Wick were the first people who presented subspace algorithms for Hammerstein model. In 2005, Gomz and Byenz generalized these algorithms to subspace algorithms such as CVA, N4SID, MOESP[21]. Hankel matrices play an important role in these algorithms. We define the input block
Hankel matrix as \( U_{02^{-1}} = \begin{bmatrix} U_p & U_f \end{bmatrix} \). \( U_p, U_f \) are the past and future input Hankel matrices which subscript \( p \) stands for “past” and \( f \) for “future”(are resulted of splitting \( U_{02^{-1}} \) into two equal parts.) \( i \) is user-defined integer such that \( i \geq N \) and \( N \) is consecutive measurements of inputs \( U \) and outputs \( Y \). Also, \( U_p^+, U_f^- \) are defined by shifting the line between future and past one block row down. The output block Hankel matrices are defined as like as input Hankel matrices. \( W \) is defined as the weight block Hankel matrix containing the inputs and outputs.

In these algorithms \( X_p^d \) shows past state variables and \( X_f^d \) shows future state variables ( \( d \) stands for deterministic algorithm). The extended observability matrix \( \Gamma_i \) ( \( i \) is the number of block rows) is defined as in “(1)”:

\[
\Gamma_i = \begin{bmatrix} C & CA & \cdots & CA^{i-1} \end{bmatrix} \tag{1}
\]

We assume that \( \{A,C\} \) is observable and has rank \( N \). Also, we define two user defined weighting matrices \( W_1, W_2 \) so that \( W_1 \) is full rank and the rank of \( W_2 \) is equal to the rank of \( (W_p W_f^+ \). \( W_p \) is the block Hankel matrix containing past input/output data \( \begin{bmatrix} U_p^+ & U_f^- \end{bmatrix} \). In the following, you see the notation \( F / G \) that represents the orthogonal projection of the row space of \( F \) onto the row space of \( G \).

Here, \( O_i \) is defined as an oblique projection of the row space of \( Y_f^+ \) along the row space of \( U_f^- \) onto the row space of \( W_p^- \) in “(2)” [25],[26]:

\[
O_i = Y_f^+/U_f^- W_p^- \tag{2}
\]

We write the SVD of the weighted matrix \( W_i O W_2^- \) in “(3)”:

\[
W_i O W_2^- = U_i \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} V_i^T \tag{3}
\]

Where \( S_1 \) contains the dominant singular values. Now we calculate an oblique projection \( O_i \) in “(4)”:

\[
O_i = \Gamma_i X_f^d \tag{4}
\]

Then System order is equal to the number of singular values in (3) different from zero and we have the “(5), (6), (7)”:  

\[
\Gamma_i = W_1^{-1}U_i S_1^{1/2} T \tag{5}
\]

\[
X_f^d W_2 = T^{-1} S_1^{1/2} V_i^T \tag{6}
\]

\[
X_f^d = \Gamma_i^t O_i \tag{7}
\]

\( T \) is an arbitrary non-singular similarity transformation. To find system matrices, we define matrix \( O_{i+1} \) as in “(8)”:

\[
O_{i+1} = Y_f^+/U_f^- W_p^+ = \Gamma_{i+1} X_{i+1}^d \tag{8}
\]

If \( \Gamma_{i+1} = \Gamma_i \) ( \( \Gamma_i \) Is \( \Gamma \) Matrix, discarding the last \( l \) rows and \( l \) is output numbers), then we have “(9)”:

\[
X_{i+1}^d = \Gamma_{i+1}^t O_{i+1} \tag{9}
\]

After computing the state, the system matrices can be solved for as a least squares problem in “(10)”:

\[
\begin{bmatrix} X_{i+1}^d \\ Y_{i+l} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i^d \\ U_{i+l} \end{bmatrix} \tag{10}
\]
The N4SID and CVA algorithms (two of the numerical subspace algorithms) use the state to find the system matrices, while MOESP (another numerical subspace algorithm) is based on the extended observability matrix $\Gamma_i$.

Now we summarize N4SID algorithm in the following [25],[26]:

1. We obtain system order and extended observability matrix $\Gamma_i$ by oblique projection $O_i$ and SVD of the weighted matrix $W_i O_i W_i^T$ in “(11)”. System order is equal to the number of nonzero singular values in $s_i$.

\[
O_i = (U_i \quad 0 \quad V_i^T) \quad (11)
\]

2. Then we obtain extended observability matrix in “(12)”: $\Gamma_i = U_i (S_i)^{\frac{1}{2}}$ (12)

3. Now we find out state matrix by “(13)”: $\bar{X}_i = (S_i)^{\frac{1}{2}} (V_i)^T$ (13)

4. At the end, the system matrices (A, B, C, D) will be calculated by extended observability matrix and state matrices with “(10)”.

III. HAMMERSTEIN-WIENER RECURRENT NEURAL NETWORK STRUCTURE

In this paper, to identify H-W neural network system, approximating the static nonlinear subsystems is done with gradient descent algorithm and the dynamic recurrent linear subsystem with subspace algorithms is done. Before starting system identification, we use a parameter initialization algorithm, which guarantees that network initial values are very close to local optimums and leads algorithm to faster convergence [8]. Recurrent neural networks have less computation time and use less requested memory. Moreover, control of such network is easier.

When we use subspace algorithms, extracted rows/columns space matrices include information about model and have high calculation performance. Also, these algorithms work based on linear algebra, mathematics, SVD and QR methods. So, they are non-iterative and don’t have convergence problem. Moreover, by these methods we will calculate system order easily and represent system based on state space which is easily understandable[25],[26]. In this paper, parametric time domain method is utilized for identification. The first nonlinear static subsystem is simulated by a simple feed forward neural network. The second subsystem is a linear dynamic model. The third subsystem is a nonlinear static system which its sum and activation functions are nonlinear. Our aim is finding out state and weight matrices with gradient descent in neural network and numerical subspace algorithms with the minimum error.

This network consists of three subsystems. The first subsystem is nonlinear static, the second subsystem is recurrent linear dynamic which is placed between two nonlinear subsystems and the third subsystem is nonlinear static subsystem which produces network output. This network, as you can see in “Figure 4.” consists of one input layer, two hidden layers, one recurrent linear dynamic layer and one output layer.

In “Figure 4.” p denotes input dimension, r denotes output dimension and q denotes state variables dimension. $W^i$ is weights between the input layer and the first hidden layer, $W^2$ is weights between the first hidden layer and recurrent linear layer (matrix B), $W^3$ is weights between the recurrent linear layer and the second hidden layer (matrix C) and $W^4$ is weights between the output layer and the second hidden layer. $d$ is first layer’s biases. The matrix A plays the most important role in network stability if its Eigen values place in unit circle. The most important feature of a nonlinear dynamic network is stability. So we have to initialize matrix A with a stable matrix.

It is resumed that Out0 is output of input layer, Out1 is output of the first hidden layer (the first nonlinear static subsystem), Out2 is output of the dynamic linear recurrent layer, Out3 is output of the second hidden layer and Out4 is the network output.

In this neural network, p is network input’s number, r is network output layer neuron’ number and q is state vector neuron’s number. In order to place neurons’ outputs in active region we have
to initialize $W^i$ and $d^i$. Active region is a region
which its activation function deviation is greater
than the maximum deviation. We assume that $W^i$
is independent and uniform distributed in $[-W_{max} ,
W_{max}]$ domain. In this paper, hyperbolic
tangent sigmoid function is used because of
producing bipolar signal for output [30][2]. Based
on [2], we calculate the maximum Euclidean
distance between data ($D_{max}$) with “(14)”. “(14)”
Calculates the sum of second power of difference
between the maximum and minimum input data.

$$D_{max} = \sqrt{\sum_{j=1}^{p} (u_{j_{max}} - u_{j_{min}})^2}$$  \hspace{1cm} (14)

Then we calculate $W_{max}$ with “(15)”:

$$W_{max} = \frac{4.356}{D_{max}} \sqrt{\frac{3}{p}}$$  \hspace{1cm} (15)

“(17)” obtains the initial values of biases by
half of the difference between maximum input
data and minimum input data that is calculated in
“(16)”.

$$c^{bound} = \frac{1}{2} (u_{max} - u_{min}) = [c_1^{bound},...,c_p^{bound}]^T$$  \hspace{1cm} (16)

$$d_j = -\sum_{i=1}^{p} c_i^{bound} w_j$$  \hspace{1cm} (17)

In order to simplify work, we divide neural
network into following four substructures:

- The first substructure which its input
is network input and its output is Out'.
Initializing $d_j$ and $W^i$ is necessary. Out'
is resulted from this substructure which
is subspace algorithm inputs in the next
substructure.

- The second substructure which its input is
Out' and its output is Out". It is obvious
that its structure is as like as state space
models. Matrix A demonstrates system
dynamic and matrix B demonstrates $W^2$
weights.

- The third substructure which its input is $x_i(k)$ (it is equal to Out" that is extracted
of the previous substructure). Its function
is nonlinear activation function and its
output is Out'.

- The forth substructure which it’s input
is Out' that is extracted of previous
substructure. Its function is linear and
produces Out' as output.

Now by applying gradient descent method, we update $W^3, W^4$ and bias.

IV. LEARNING ALGORITHM BASED ON SUBSPACE ALGORITHMS IN
HAMMERSTEIN-WIENER RECURRENT
NEURAL NETWORK

In this step, the $W^2$, $W^3$ and A matrices will be updated. In “Figure 6.” dotted area shows the
subsystem which will be identified by subspace
identification algorithm. To start, we calculate
N4SID algorithm outputs which their number
is equal to total output numbers. We find out
N4SID outputs by inversing the second nonlinear
function (tansig). tansig function is defined as
“(18)”:

$$\text{tansig}(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$$  \hspace{1cm} (18)

Its inverse is calculated in “(19)”:

$$y = \frac{1}{2} \ln \left( \frac{x+1}{x-1} \right)$$  \hspace{1cm} (19)

To start, it is necessary to divide network
output by $W^4$. Then calculate its tansig and
inverse function by “(19)”. After that the N4SID
output will be resulted. At the end, because
the output and inputs of N4SID algorithm are
resulted from previous step, so system matrices,
state matrix and system order will be resulted by
N4SID algorithm.

The proposed algorithm is summarized in the
following:

1. Initializing the first and second nonlinear
subsystems randomly.
2. Calculating $W^1$ and bias for initializing
these parameters and faster convergence.
3. Obtaining N4SID inputs and outputs by
total desired output of neural network.
4. Calculating system matrix and state
matrix by subspace algorithms.
5. Updating nonlinear subsystems by
gradient descent and going back to step 3.

V. SIMULATION

At first, we gather input/output measured
data and select a suitable structure for the model.
Here we consider Hammerstein-Wiener recurrent
neural network. In training step, model and its
parameters are approximated. The resulted model
is evaluated to find that its error is less than other algorithm or not?

In this algorithm, we work with “(20)”:

\[
y(k+1) = \alpha \frac{y(k) y(k-1) (k-2) u(k-1) y(k-2) - \beta + u(k)}{1 + y(k-1) + y(k-2)}
\]

(20)

In “(20)”, it is necessary that \( \beta > 1 \). To start, we consider the values of \( \alpha, \beta \) equal to \((1, 1)\). The training input data are produced randomly in \([-2, 2] \) for the half of the training time and are produced by function \(1.05 \sin(\pi k / 45)\) for the remaining half of the training time. The test data are produced by “(21)”:

\[
u(k) = \begin{cases} 
\sin(\pi k / 25), & 0 < k < 250 \\
1.0, & 250 \leq k < 500 \\
-1.0, & 500 \leq k < 750 \\
0.3 \sin(\pi k / 25) + 0.1 \sin(\pi k / 32), & 750 \leq k < 1000 \\
0.6 \sin(\pi k / 10), & 1000 \leq k < 10000
\end{cases}
\]

(21)

In this algorithm, the elements of matrix \( A \) have to be initialized, to have stable network in the beginning. In addition, this matrix has to be stable. However, the first layer weight parameter and the first layer bias are resulted by initialization algorithm (it makes fast convergence); other parameters are initialized randomly. This algorithm is done in 100 epochs on 1000 data which its result is in TABLE I.

### VI. CONCLUSION

One of the most important issues which should be considered to increase the accuracy of identification is the initialization process of neural networks. Appropriate initialization can boost the convergence and may prevent local minima. Furthermore, in the design of the Hammerstein-Wiener the estimation of the parameters of the matrix \( A \) and its dimension is very important and challenging. Subspace methods make it possible to approximate this matrix and its dimension. The combination of subspace identification methods with GD makes it possible to have better convergence speed while obtaining better accuracy.

### TABLE I.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training step mse</th>
<th>Test step mse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural network</td>
<td>0.05</td>
<td>0.5</td>
</tr>
<tr>
<td>The proposed algorithm</td>
<td>0.0201</td>
<td>0.0108</td>
</tr>
</tbody>
</table>

### TABLE II.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Gradient descent</th>
<th>Gradient descent with momentum</th>
<th>Levenberg-Marquart</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mse</td>
<td>0.683</td>
<td>0.692</td>
<td>0.0108</td>
<td></td>
</tr>
<tr>
<td>Number of runs</td>
<td>1000</td>
<td>54</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>Run time</td>
<td>16 minutes</td>
<td>40 minutes</td>
<td>9 minutes</td>
<td>60 minutes</td>
</tr>
<tr>
<td>Validation error</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
REFERENCES
