Identification for Polynomial Wiener Model Using Improved Back Propagation Method

Xiaobing Shao
School of Information Science and Technology, Jiujiang University, Jiujiang, Jiangxi province, 332005, China
E-mail: wxh_jj_mail@163.com

Abstract

A new method is introduced for the identification of nonlinear dynamic system described by Wiener model, which consists of a linear dynamic block followed by a static non-linearity. Firstly, it is assumed that static non-linear part is invertible and its inverse characteristics can be expressed or approximated by a polynomial of known orders. Secondly, based on these assumptions, a novel neural network structure is designed, the weights in which are corresponding with the parameters of polynomial Wiener model. Finally, to solve the problem of non-convergence of conventional back propagation iterative, the improved one is derived, through which the non-linear and linear dynamic part can be optimized and the coefficients of the polynomial Wiener model are gotten with a higher convergence rate. A numerical example is included to show the effectiveness and the practical feasibility of the presented approach.

Index Terms—Wiener model, neural network, back propagation, identification

Introduction

Wiener model is the most known and the most widely implemented member of this class with the linear dynamic block (element) preceding the nonlinear static one [1].

Models of nonlinear static elements can be realized in different forms such as polynomials [2], splines [3], basis functions [4], wavelets [5], neural networks [6], look-up tables [7], and fuzzy models [8]. Impulse response models, pulse transfer models, and state space models are common representations of linear dynamic systems [9].

Polynomial models are widely used as models of nonlinear elements. A great advantage of these is effective parameter optimization, which can be performed off-line with the least squares method or on-line with its recursive version [10].

The rest of the paper is organized as follows: In section 2, the architecture of polynomial Wiener model is discussed. Section 3 describes the basic principle of the sequential pattern learning. Calculations of the gradient in series-parallel Wiener models with the improved back propagation algorithm are carried out in
Section 4. Simulation experiments for Wiener model identification and corresponding analysis are presented in Section 4. Finally, Section 5 concludes the paper.

**Problem formulation**

Nonlinear models of a given internal structure composed of sub-models, referred to also block-oriented models, are members of the class of gray box models. Wiener model, shown in Figure 1, is well-known examples of such single input single output (SISO) models, composed of sub-models.

![Model of Wiener system](image)

The model contains a linear dynamic system and a nonlinear static element in a cascade. While in the Wiener system the nonlinear element follows the linear dynamic system. The output $y(t)$ of the Wiener system at the time $t$ is

$$y(t) = f(s(t)) + e(t),$$  \hspace{1cm} (1)

where $f(\cdot)$ denotes the nonlinear function describing the nonlinear element, $e(t)$ is the additive system output disturbance, and $s(t)$ is the output of the linear dynamic system and input of the nonlinear static element

$$s(t) = \frac{B(q^{-1})}{A(q^{-1})} u(t)$$  \hspace{1cm} (2)

with

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n},$$  \hspace{1cm} (3)
$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_m q^{-m},$$  \hspace{1cm} (4)

and $B(q^{-1})/A(q^{-1})$ is the pulse transfer function of the linear dynamic system, $q^{-1}$ is the backward shift operator, with the properties that $q^{-m} y(t) = y(t - m)$ and $q^{-m} f(s(t)) = f(s(t - m))$, $a_1, \ldots, a_n, b_1, \ldots, b_m$ are the unknown parameters of the linear dynamic system.

Let us assume that, the orders $m$ and $n$ of the polynomials $A(q^{-1})$ and $B(q^{-1})$ are known, the linear dynamic system is casual and asymptotically stable, and the non-linear function $f(\cdot)$ is continuous. And assume also that the polynomials $A(q^{-1})$ and $B(q^{-1})$ are coprime with the input $u(t)$ has finite moments and is independent of $e(t)$ for all time $t$.

The steady state characteristic of the system can be approximated by a polynomial $f(\cdot)$ of the order $p$ as

$$\hat{f}(s(t)) = \sum_{i=0}^{p} \hat{c}_i s^i(t) = \hat{c}_0 + \hat{c}_1 s(t) + \hat{c}_2 s^2(t) + \ldots + \hat{c}_p s^p(t),$$  \hspace{1cm} (5)
where \( \hat{s}(t) \) is the output of the linear dynamic system model

\[
\hat{s}(t) = \frac{\hat{B}(q^{-1})}{\hat{A}(q^{-1})} \hat{u}(t),
\]

Therefore, the parameter vector \( \hat{\theta} \) of the SISO Wiener model defined by (5) and (6) is

\[
\hat{\theta} = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_n, \hat{b}_1, \hat{b}_2, ..., \hat{b}_m, \hat{c}_0, \hat{c}_1, ..., \hat{c}_p]^T.
\]

**Prediction error method**

The identification problem can be formulated as follows: Given the sequence of the system input and output measurements \( \{u(t), y(t)\}_{t=1}^N \) estimate the parameters of the linear dynamic system and the characteristic of the nonlinear element minimizing the following global cost function

\[
J(t) = \frac{1}{2} \sum_{t=1}^N (y(t) - \hat{y}(t))^2,
\]

where \( \hat{y}(t) \) is the output of the SISO Wiener model.

In this paper, we use the sequential pattern learning to find the minimum of the global cost function. The method uses pattern-by-pattern updating of model parameters, changing their values by an amount proportional to the negative gradient of the local cost function. This results in the following rule for the adaptation of model parameters

\[
\theta(t) = \theta(t-1) - \eta \frac{\partial J(t)}{\partial \theta(t-1)},
\]

\[
\frac{\partial J(t)}{\partial \theta(t-1)} = (y(t) - \hat{y}(t)) \frac{\partial \hat{y}(t)}{\partial \theta(t-1)},
\]

where \( \theta(t) \) is the weight vector containing all model parameters at the time \( t \), and \( \eta > 0 \) is the learning rate. It has been shown that such a learning procedure minimizes the global cost function \( J(t) \) provided that the learning rate \( \eta \) is sufficiently small.

The output of the linear dynamic model is given by the following difference equation

\[
\hat{s}(t) = -\sum_{i=1}^n \hat{a}_i s(t-i) + \sum_{i=1}^m \hat{b}_i u(t-i).
\]

Note that the characterization of the SISO Wiener system is not unique as the linear dynamic system and the nonlinear element are connected in series. In other words, Wiener systems described by \( \frac{B(q^{-1})}{kA(q^{-1})} \) and \( f(k \cdot) \) reveal the same input-output behavior for any \( k \neq 0 \).

**Gradient calculation**
A. Back Propagation Algorithm

The calculation of the gradient in Wiener models can be carried out with the well-known back propagation algorithm. The parallel Wiener models are dynamic systems and the corresponding neural networks are recurrent ones with one linear recurrent node that models the linear dynamic system followed by the polynomial model of the nonlinear element. The architecture of the parallel model is shown in Figure 2.

![Figure 2. Architecture of polynomial SISO Wiener model](image)

In the architecture, Wiener model does not contain any inverse element and is even simpler in comparison with that of the series-parallel one. The parallel model is of the recurrent type as it contains a single feedback connection and \( u(t) \) is the only input to the model. The output \( \hat{y}(t) \) of the parallel Wiener model is given by \( \hat{y}(t) = f(\hat{s}(t)) \).

In parallel Wiener models, the architecture complicates the calculation of the gradient considerably. A crude approximation of the gradient can be obtained with the back propagation method, which does not take into account the dynamic nature of the model. This simplifies the training of the model considerably.

In the back propagation method, the dependence of the past linear dynamic model outputs \( \hat{s}(t-i), i=1...n \), on the parameters \( a_i \) and \( b_i \) is neglected.

Hence, from (5) and (11), it follows that

\[
\frac{\partial \hat{y}(t)}{\partial a_i} = \frac{\partial \hat{y}(t)}{\partial \hat{s}(t)} \frac{\partial \hat{s}(t)}{\partial a_i} = -\hat{s}(t-i) \frac{\partial \hat{y}(t)}{\partial \hat{s}(t)} , \quad i = 1,...,n , \quad (12)
\]

\[
\frac{\partial \hat{y}(t)}{\partial b_i} = \frac{\partial \hat{y}(t)}{\partial \hat{s}(t)} \frac{\partial \hat{s}(t)}{\partial b_i} = u(t-i) \frac{\partial \hat{y}(t)}{\partial \hat{s}(t)} , \quad i = 1,...,m , \quad (13)
\]

\[
\frac{\partial \hat{y}(t)}{\partial \hat{c}_i} = \hat{s}(t) , \quad i = 1,...,p . \quad (14)
\]

The partial derivative of the parallel Wiener model output with respect to the output of the linear dynamic model is

\[
\frac{\partial \hat{y}(t)}{\partial \hat{s}(t)} = \sum_{i=1}^{p} i \hat{c}_i \hat{s}^{i-1}(t) = \hat{c}_1 + 2 \hat{c}_2 \hat{s}(t) + ... + p \hat{c}_p \hat{s}^{p-1}(t) . \quad (15)
\]
B. Improved Back Propagation Methods

The improved back propagation method differs from the standard one markedly in the calculation of partial derivatives of the output of the linear dynamic model with respect to its parameters \( \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n, \hat{b}_1, \hat{b}_2, \ldots, \hat{b}_m \). In general, as the improved back propagation method uses a more accurate evaluation of the gradient than the standard one, a higher convergence rate can be expected.

Assuming that the parameters \( \hat{a}_i \) and \( \hat{b}_i \) do not change and differentiating, we have

\[
\frac{\partial \hat{y}(t)}{\partial \hat{a}_i} = -\hat{s}(t-i) - \sum_{j=1, j \neq i}^{n} \hat{a}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{a}_i}, \quad i = 1, \ldots, n, \quad (16)
\]

\[
\frac{\partial \hat{y}(t)}{\partial \hat{b}_i} = u(t-i) - \sum_{j=1, j \neq i}^{m} \hat{b}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{b}_i}, \quad i = 1, \ldots, m. \quad (17)
\]

The partial derivatives (18) and (19) can be computed on-line by simulation, usually with zero initial conditions.

The substitution of (18) and (19) into (14) and (15) gives

\[
\frac{\partial \hat{y}(t)}{\partial \hat{a}_i} = \left( -\hat{s}(t-i) - \sum_{j=1}^{n} \hat{a}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{a}_i} \right) \sum_{i=1}^{p} i \hat{c}_i \hat{s}^{i-1}(t), \quad i = 1, \ldots, n \quad (18)
\]

\[
\frac{\partial \hat{y}(t)}{\partial \hat{b}_i} = \left( u(t-i) - \sum_{j=1}^{m} \hat{b}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{b}_i} \right) \sum_{i=1}^{p} i \hat{c}_i \hat{s}^{i-1}(t), \quad i = 1, \ldots, m \quad (19)
\]

In comparison with the back propagation method, the improved one is only a little more computationally intensive. The increase in computational burden comes from the simulation of \( m + n \) sensitivity models, whereas dynamic models of a low order are used commonly.

Note that to obtain the exact value of the gradient, the parameters \( \hat{a}_i \) and \( \hat{b}_i \) should be kept constant. That is the case only in the batch mode, in which the parameters are updated after the presentation of all learning patterns. In the sequential mode (pattern by pattern learning), the parameters \( \hat{a}_i \) and \( \hat{b}_i \) are updated after each learning pattern and an approximate value of the gradient is obtained. Therefore, to achieve good approximation accuracy, the learning rate should be sufficiently small to keep changes in the parameters negligible.

C. Iterative Convergence Process

Followed by the global cost function \( J(t) \) defined in (8), the iterative learning of the SISO Wiener parameter as shown in figure 2 can be estimate with one certain input and output measurements \( \{u(t), y(t)\}_{t=1}^{N} \) of the system.

Error expression:

\[
e(t) = y(t) - \hat{y}(t). \quad (20)
\]

Parameter update expression:
\[
\Delta \hat{a}_i = \eta(t) \frac{\partial \hat{y}(t)}{\partial \hat{a}_i} = \eta(t) \left( - \hat{s}(t-1) \sum_{j=1}^{n} \hat{a}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{a}_i} \right) \sum_{i=1}^{p} i \hat{c}_i \hat{s}^{i-1}(t), \quad (21)
\]

\[
\Delta \hat{b}_i = \eta(t) \frac{\partial \hat{y}(t)}{\partial \hat{b}_i} = \eta(t) \left( u(t-1) - \sum_{j=1}^{m} \hat{b}_j \frac{\partial \hat{s}(t-j)}{\partial \hat{b}_i} \right) \sum_{i=1}^{p} i \hat{c}_i \hat{s}^{i-1}(t), \quad (22)
\]

\[
\Delta \hat{c}_i = \eta(t) \frac{\partial \hat{y}(t)}{\partial \hat{c}_i} = \eta(t) \hat{s}^{i}(t), \quad i = 1, \ldots, p, \quad (23)
\]

where \(\Delta \hat{a}_i, \Delta \hat{b}_i, \Delta \hat{c}_i\) are the update of the SISO Wiener parameters \(\hat{a}_i, \hat{b}_i, \hat{c}_i\) after each learning pattern.

It is well known that parallel models can lose their stability during the learning process even if the identified system is stable. A common way to avoid this is to keep the learning rate \(\eta\) small. To preserve the stability of Wiener models, some other heuristic rules can be applied easily. Another simple approach, which can be applied in the sequential mode, is based on the idea of a trial update of parameters, testing the stability of the model, and performing an actual update only if the obtained model is asymptotically stable; otherwise the parameters are not updated and the next training pattern is processed.

The initial parameters \(\hat{a}_i, \hat{b}_i, \hat{c}_i\) can be all set zero. At the beginning of each batch of iterations, the initial parameters are \(\hat{y}(0) = 0, \frac{\partial \hat{y}(t)}{\partial \hat{a}_i} = 0, \frac{\partial \hat{y}(t)}{\partial \hat{b}_i} = 0, \frac{\partial \hat{y}(t)}{\partial \hat{c}_i} = 0\) and \(s(t) = 0\).

After hundreds of iterations, the parameters do not reach convergence until the global cost function \(J(\cdot)\) is less than a preset mini-parameter \(\varepsilon\).

**Simulation experiment**

In the simulation experiment, the second order Wiener system composed of a continuous linear dynamic system given by the transfer function was converted to discrete time, assuming a zero order hold on the input and the sampling interval 1ms, leading to the following difference equation

\[
s(t) = 0.25s(t-1) + 0.5s(t-2) + u(t) + 0.2u(t-1) + 0.7u(t-2).
\]

The system contained also a nonlinear element given by

\[
y(t) = f(s(t)) = -0.5s(t) + 0.2s^2(t) + s^3(t)
\]

The system was driven by a sequence of 500 numbers with the following function

\[
u(t) = \frac{1}{20} \sin \left( \frac{\pi t}{25} \right) + \frac{1}{20} \sin \left( \frac{\pi t}{120} \right).
\]

The obtained responses of the SISO Wiener for sine excitation signal are illustrated in Figure 3.
The SISO Wiener model was trained recursively using the steepest descent method, with the learning rate $\eta = 0.1$, and calculating the gradient with the improved back propagation algorithms. To observe convergence rates of the algorithms, a mean square error (MSE) is defined as

$$MSE = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2.$$  

The simulation results for a Wiener system are disturbed by the additive output Gaussian noise $N(0,0.01)$. In the learning process, the terminal condition is $10^{-8}$. After iterative training of 125 times, the SISO Wiener model is convergence.

Take the excitation signal $u(t)$ into the identified SISO Wiener model, and the practical outputs of the model are shown in Figure 4. It is not difficult to find that the simulation output $y(t)$ is in good agreement with the identified ones $\hat{y}(t)$, which means the identification of Wiener is successful.

**Conclusion**
In this chapter, a steepest descent-learning algorithm for SISO Wiener model has been derived and analyzed. An important advantage of the improved back propagation algorithm is that its training requires almost half the computational burden for the training of the series-parallel model with the standard one. Applying the improved back propagation algorithms, one can expect a higher convergence rate as a more accurate gradient approximation is used.

References