Nonlinear Model Structure Identification
Based on Kernel Visualization

L. Keviczky and Cs. Bányaśz
Computer and Automation Research Institute of the Hungarian Academy of Sciences
Control Engineering Research Group of the HAS at the
Dept. of Automation and Applied Informatics
Budapest University of Technology and Economics
H-1111 Budapest, Kende u 13-17, HUNGARY

Abstract - The success of the nonlinear dynamic system identification strongly depends on the applied model structure. Nonlinear systems have almost infinite varieties of structures. This paper shows a simple structure identification technique based on image processing recognition method to distinguish between Hammerstein and Wiener models.

Keywords: nonlinear model, structure identification

1 Introduction

Nonlinear dynamic systems have infinite variety so it can not be expected that a unique optimal control solution exists for all of these complex processes. However, the history and development of nonlinear control systems show that one can expect relatively simple methods, which can be close or even reach the effectiveness of linear methodology. A very wide class of approaches are based on the usual Jacobean linearization [5] of a nonlinear system (NS). A large class of nonlinear systems can be made to have linear input-output behavior through a choice of nonlinear state feedback control law [10]. Other approaches assume special topology, when the structure of the NS makes the linearization possible. Such NS classes are e.g. the bilinear and the block-oriented factorable (cascade) systems [4].

The most well known factorable models are the simple Hammerstein and Wiener models.

\[ N_W = N_{W,stat} \cdot Y_{W,dyn} \]  \[ N_W = N^W \cdot Y^W \]  \[ N_H = Y^{H,dyn} \cdot N_{H,stat} \]  \[ N_H = Y^H \cdot N^H \]

The simple Hammerstein model \( N_H \) (shown in Fig. 1b) is a cascade structure of a nonlinear static \( N_{H,stat} \) and a linear dynamic \( Y^{H,dyn} \) terms connected in series, i.e. \( N_H = Y^{H,dyn} \cdot N_{H,stat} \) or simply \( N_H = Y^H \cdot N^H \).

Note that the order of the nonlinear operators in the formulas is opposite to the order of the blocks shown in the figures. (This is a usual source of mistakes calculating transfer characteristics of open- and closed-loop nonlinear schemes.)

There exists many control methods published, including methods from the authors (e.g. [1], [7]), which can be applied for the Wiener and Hammerstein nonlinear dynamic model classes. These methods assume a proper process model identification procedure to obtain a good approximate model. A natural question always arises, whether the true process falls into these model structures.

Many nonlinear dynamic systems with input signal \( u(t) \) and output signal \( y(t) \) can be approximated in the vicinity of the working point at least by the so-called Volterra integral or the Volterra weighting function model [4].

\[ y(t) = g_0 + \int_{\tau_1=0}^t g_1(\tau_1)u(t-\tau_1)d\tau_1 + \int_{\tau_2=0}^t \int_{\tau_1=0}^t g_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1d\tau_2 + \ldots \]

Similarly to the discrete time description of the linear systems, a multi-dimensional convolution sum describes
the relation between the sampled input $u[k]$ and output signals $y[k]$

$$y[k] = h_0 + \sum_{k_1=0}^{k} h_1[k_1]u[k - k_1] +$$
$$+ \sum_{k_1=0}^{k} \sum_{k_2=0}^{k} h_2[k_1, k_2]u[k - k_1]u[k - k_2] + \ldots$$

$$= \sum_{i=0}^{n} \ldots \sum_{k_j=0}^{k} h_i[k_1, \ldots, k_j] \prod_{j=1}^{i} u[k - k_j]$$

We call (3) the Volterra weighting function series. Another name for it is the Gabor-Kolmogorov series. The process is characterized by its Volterra kernels:

$$g_n(t_1, \ldots, t_n) \quad n = 0, 1, 2, \ldots \quad \text{in the continuous time case}$$

$$h_n[k_1, \ldots, k_n] \quad n = 0, 1, 2, \ldots \quad \text{in the discrete time case.}$$

(There are several methods to compute the relation between the continuous and discrete time kernels of a system.)

In the engineering practice only second order Volterra kernels are determined and used. The coefficients of a such a form are not difficult to estimate relatively simply model identification procedure, contrary to the above cascade models. Therefore it is very useful to perform a structure identification to determine which cascade model is the best for the measured process data.

While the Volterra kernels can easily be derived from the block oriented models, the structure and the parameters of the block oriented models cannot be computed in a trivial way from the estimated Volterra kernels.

The constant term $g_0$ is equal to $h_0$, the first degree Volterra kernel $g_1(t_1)$ is equal to the weighting function of the linear channel. The quadratic channel can be described as

$$y_2(t) = \int_{t_1=0}^{\infty} \int_{t_2=0}^{\infty} g_2(\tau_1)g_2(\tau_1 - \tau_2)g_1(\tau_2 - \tau)$$

$$u(t - \tau_1)u(t - \tau_2)dt \, d\tau_1 \, d\tau_2$$

thus the quadratic kernel can be calculated by the following convolution integral

$$g_2(\tau_1, \tau_2) = \int_{\tau=0}^{\infty} g_4(\tau)g_2(\tau_1 - \tau)g_3(\tau_2 - \tau)d\tau$$

It seems there exists an unequivocal relation between the parameters of the block oriented model and the Volterra kernels. The transformation is unequivocal only in one direction because the weighting functions $g_i(t_j)$ of the block oriented models cannot be reconstructed from the identified Volterra kernels.

On the basis of the above considerations a so-called parametric Volterra model can be obtained for the second degree discrete-time Volterra kernel

$$y[k + d] = h_0 + \sum_{k_1=0}^{k} h_1[k_1]u[k - k_1] +$$

$$+ \sum_{j=0}^{\infty} \sum_{k_1=0}^{k} h_{2j}[k_1]u[k - k_1]u[k - k_1 - j]$$

which is linear in the parameters. Here a usual time delay $d$ was also introduced. If the parameters in this second order kernel are identified they can give information on the possible cascade structure for a second, more accurate identification method. So the first estimations of the kernel parameters $h_0, h_1, h_{2j}$ and their variances can be easily obtained from a linear regression.

Fig. 2 Plots of the Volterra kernels of a Hammerstein model with first-order lag term
2 Model Structures in the Parameter Space of a Volterra Kernel

The kernel parameter distributions of the two basic cascade models shown in Fig. 1 can be easily determined by computer simulation. The Figs. 2 and 3 illustrate the Volterra kernels for the Hammerstein and for the Wiener models [4]. The Volterra kernels of Figs. 2 suggest that for Hammerstein model the quadratic kernels differ from zero only at the main skew (or secondary) diagonal. On the Fig. 3 the sections \( h_2[k_1, k_2] \) for \( k = \text{const} \) are proportional to each other, and the level lines are straight (subdiagonals), because \( h_2[k_1, k_2] = h_1^2[k_1] \).

It is possible to compute further kernel patterns for other cascade block-oriented nonlinear model classes.

3 Analytical Structure Indices

The quadratic kernels of the Hammerstein models differ from zero only at the main diagonal. This feature can be seen from the plot of \( h_2[k_1, k_2] \) and from building the following index:

\[
\alpha_1 = \frac{2 \sum_{k_1=0}^{m} \sum_{k_2=k_1+1}^{m} h_2^2[k_1, k_2]}{m(m+1) \sum_{k_1=0}^{m} h_2^2[k_1, k_2]} \tag{8}
\]

In (8) the mean square values of the off-diagonal elements are divided by that of the main diagonal. Here and further on any characteristic measure of the size of the element can be used instead of the square value (e.g., absolute value). The main skew diagonal is the straight line for which \( k_1 = k_2 \) and \( m \) is the memory of the kernels of the discretized model. It is easy to see that \( \alpha_1 \) becomes zero only for the Hammerstein model.

The quadratic Volterra kernels along the main skew diagonal are proportional to the squares of the linear kernels in the simple Wiener models. Form the ratio of them as a discrete time function

\[
\beta_1[k_1] = \frac{h_2[k_1, k_1]}{h_1[k_1]^2} \tag{9}
\]

and its average value

\[
\bar{\beta}_4 = \frac{1}{m+1} \sum_{k_1=0}^{m} \beta_4[k_1] \tag{10}
\]

Then the normalized deviation of (9)

\[
\alpha_4 = \frac{1}{m+1} \sum_{k_1=0}^{m} \left[ \beta_4[k_1] - \bar{\beta}_4 \right]^2 \tag{11}
\]

becomes zero only for the Wiener model.

4 Pattern Recognition of the Kernel Shapes

The above analytical indices are exact measures for finding Wiener and Hammerstein model structures. However, it is very difficult to compute a statistical probe, how close they are to zero in case of noisy measurements. In the practice these measures are more or less heuristical norms for the model structure determination.

In our practice it was found that relatively simple pattern (shape) recognition algorithms work more robust than the analytical indices.
Mathematicians typically define shape as an equivalence class under a group of transformations. This definition is incomplete in the context of visual analysis. This only tells us when two shapes are exactly the same. We need more than that for a theory of shape similarity or shape distance. The statistician’s definition of shape, e.g. [3] or [6], addresses the problem of shape distance, but assumes that correspondences are known. Other statistical approaches to shape comparison do not require correspondences – e.g. one could compare feature vectors containing descriptors such as area or moments – but such techniques often discard detailed shape information in the process. Shape similarity has also been studied in the psychology literature.

There are several extensive surveys of shape matching in computer vision literature. There are basically two approaches: feature-based, which involve the use of special arrangements of extracted features such as edge elements or junctions; and brightness-based, which make more direct use of pixel brightness [9].

Assume that the variances of the estimated kernel parameters are denoted by \( \sigma_0 \), \( \sigma_1[K_1] \) and \( \sigma_2[K_1,j] \) and available from the linear regression performed using the model (7) linear in the parameters. Compute the relative significance factors:

\[
\gamma_0 = \frac{\hat{h}_0}{\sigma_0}, \quad \gamma_1[K_1] = \frac{\hat{h}_1[K_1]}{\sigma_1[K_1]}, \quad \gamma_2[K_1,j] = \frac{\hat{h}_2[j,K_1,j]}{\sigma_2[j,K_1,j]}, \quad \gamma_2 = \frac{\hat{h}_2[j,K_1,j]}{\sigma_2[j,K_1,j]}, \quad (12)
\]

For the second degree kernel transform the estimated parameters to a dot matrix \( \mathbf{K} \) putting 1 to those elements where the \( \gamma_2[K_1,j] \geq \delta_2 \) and 0 elsewhere. Here \( \delta_2 \) can be given heuristically or can be selected similarly than in the well-known Student’s \( t \)-probe for linear model identification. This simple method transforms the first model into the second brightness-based case.

For the Hammerstein model the special pattern is the skew diagonal of the kernel matrix. So it is possible to use a classical character recognition package, which has to find a “slash” (i.e. \( \div \)) character for the transformed matrix \( \mathbf{K} \).

For the Wiener model it is required to recognize special subdiagonals, which form a Toeplitz matrix structure. So it is possible to use a classical character recognition package, which has to find a “backslash” (i.e. \( \\backslash \)) character for all Toeplitz lower sub matrices in \( \mathbf{K} \).

To the application of the feature-based method the feature extraction rule set is quite difficult to construct for these simplest cascade models. (This work, however, can not be avoided for higher order dynamics.) Instead, it is not difficult to construct another algorithm, which does not transform the estimated kernel to \( \mathbf{K} \), but uses special masks, shown in Fig. 4 to ease the feature extraction.

![Fig. 4 Filter masks for heuristic shape recognition](image)

Using these masks it is not so difficult to form the feature extraction rule set and these rules can be applied in the combination with the analytical indices.

![Fig. 5 Plots of the Volterra kernels of a Wiener model with second-order lag term](image)
5 Generalization Possibilities

The above presented methods for Hammerstein models can be used without modification for the so-called generalized Hammerstein model, too, because the second degree kernels also concentrate on the main skew diagonal.

The generalized Wiener model with a first order lag can be also handled in the same way as shown above. Unfortunately more complex Wiener models (e.g. second order dynamics) need different feature set to be applied, because the kernel contour level shapes are considerably different (see Fig. 5 for the obtained kernel behaviors). The only practical solution is to collect a proper shape database for the different structure primitives. This is not difficult, because the database can be computed by computer simulation, however, very time consuming.

6 Conclusions

The purpose of the paper was to present the special feature sets of nonlinear dynamic cascade Wiener and Hammerstein models using the second degree Volterra kernel parameters.

The special shapes on the two-dimensional plots of the elements of the kernel matrix invokes the application possibilities of different pattern recognition algorithms for nonlinear structure identification purposes. The major contribution is that the presented methods are very good to distinguish the Wiener and Hammerstein model classes. Further structure determination, especially within the Wiener models, require more complex shape recognition algorithms and/or shape features extraction data bases.

7 References