Journal of Process Control 19 (2009) 1174–1181

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Support vector method for identification of Wiener models

Stefan Tötterman, Hannu T. Toivonen *

Faculty of Technology, Åbo Akademi University, Biskopsgatan 8, FIN–20500 Åbo, Finland

ARTICLE INFO

Article history:
Received 16 July 2008
Received in revised form 3 March 2009
Accepted 3 March 2009

Keywords:
Support vector machines
System identification
Wiener models

ABSTRACT

Support vector regression is applied to identify nonlinear systems represented by Wiener models, consisting of a linear dynamic system in series with a static nonlinear block. The linear block is expanded in terms of basis functions, such as Laguerre or Kautz filters, and the static nonlinear block is determined using support vector machine regression.

1. Introduction

In empirical black-box modelling and identification of nonlinear dynamical systems the choice of model structure plays an important role. In model structure selection one needs to take into account the kind of system behaviours which should be described, the effort needed to identify the model parameters from empirical data, and how well the model is suited for its intended use, such as prediction or control [22,23,26]. A useful class of nonlinear models consists of block-oriented representations [23], where various system characteristics, such as nonlinearities and dynamical response, are represented by separate blocks connected in series. Standard nonlinear models of this type are the Wiener and Hammerstein models [2,11], where the nonlinearities are captured in a static block and the dynamics are represented by a linear dynamical component.

The Wiener class of models is based on Volterra series representations of nonlinear dynamic systems, which are obtained by a generalization of Taylor series expansions to dynamic systems. A drawback of the original Volterra model is that it typically requires inclusion of a large number of terms for good accuracy. In order to obtain a more parsimonious representation, Wiener [32] introduced an expansion of the Volterra series consisting of a linear dynamic component followed by a nonlinear static block. Wiener introduced an expansion of the linear dynamic part in terms of orthonormal Laguerre functions, while the nonlinearity was captured using Hermite polynomials. However, the Wiener class of models is not restricted to the Laguerre–Volterra representation, but other representations of the linear dynamic and the static nonlinear blocks can be used as well [20]. The linear dynamic block has been represented by transfer function models [33] and expansions in terms of Kautz functions [6,12] or other orthonormal filter expansions [9]. In order to represent the static nonlinear block, multilayer perceptrons [29,34] and radial basis functions [9] have been proposed as alternatives to polynomial expansions.

Although Wiener models can approximate any nonlinear system satisfying some mild continuity conditions with any degree of accuracy [3], they are particularly well suited for modelling processes whose dynamics can naturally be decomposed in a linear dynamical component followed by a static nonlinearity. Typical examples are processes with sensor nonlinearities and pH processes [9,13,21]. Wiener models have also been applied to model fluid flow systems [33], separation processes [29], fluid catalytic cracker units [35] as well as various biological systems [11]. Other types of systems are better described using models with alternative structures. An important class of models are Hammerstein models, where the order of the blocks are reversed, i.e., they consist of a static nonlinear block followed by a linear dynamic component. For a discussion of the dynamical differences between Wiener and Hammerstein models, see [2].

A number of methods have been proposed to identify Wiener models from experimental data [9,10,13,23,25,29,33,35], which are based on the standard least-squares method in which the sum of squared prediction errors is minimized. If both the linear dynamic component and the static nonlinearity are identified, the model output is a nonlinear function of the parameters, resulting in a nonlinear least-squares problem which may have local optima. This is avoided by using an appropriate basis filter expansion of the dynamic block and by expanding the nonlinearity using polynomials or basis function, but in return the number of terms needed in the expansion may be high. When using functional

* Corresponding author. Tel.: +358 2 2154676; fax: +358 2 2154808.
E-mail address: Hannu.Toivonen@abo.fi (H.T. Toivonen).

0959-1524/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2009.03.003

expansions to represent the model components, the number of parameters may also be increased by the fact that a fairly high number of basis functions may be required for good accuracy. The high number of model parameters makes the identification problem sensitive to possible overtraining. In order to avoid this, regularization terms have been used [25].

Support vector machines have been introduced as a powerful robust approach to classification and regression problems [28]. Support vector regression (SVR) is based on Vapnik's ϵ-insensitive loss function and structural risk minimization, which aims to bound the mean approximation error when the model is determined from a finite data set. In support vector regression, a nonlinear model is represented by an expansion in terms of nonlinear mappings of the model input. The nonlinear mappings define a feature space, which may have infinite dimension. The solution can, however, be expressed in terms of an associated kernel which has the same dimension as the data set, and it is therefore not necessary to compute the nonlinear mappings explicitly. Support vector regression has several appealing properties for black-box identification. The solution is defined in terms of a convex quadratic minimization problem, for which convergence to the global solution can be guaranteed. A further convenient feature of the approach is that the optimal model complexity is obtained as part of the solution, and does not have to be separately determined. Moreover, structural risk minimization introduces robust performance with respect to new data. In system identification, support vector methods have been applied to linear ARX models [1,24] and nonlinear dynamical models [16,17], including partially linear models [7], state-dependent parameter models [27] and Hammerstein models [8].

In this contribution support vector regression is applied to identify Wiener type models, where the linear dynamic block is described by a basis filter expansion. The model structure is similar to the one considered in [9,29,34], but the nonlinear static block is here represented by a nonlinear kernel and support vector regression is used for identification.

The paper is organized as follows. In Section 2 the identification problem is stated, and a support vector procedure for Wiener model identification is presented in Section 3. A numerical example to illustrate the properties of the proposed method is presented in Section 4.

2. Problem formulation

In this paper we study identification of discrete-time nonlinear systems. We consider a single-input single-output discrete-time nonlinear dynamical system

$$\begin{aligned} \mathbf{x}(n+1) &= f(\mathbf{x}(n), u(n)) \\ y(n) &= h(\mathbf{x}(n)) + e(n) \end{aligned} \tag{1}$$

where u is the scalar input signal, $\mathbf{x}$ is the state vector, y is the scalar measured output signal, e is additive white noise, and $f(\cdot,\cdot)$ and $h(\cdot)$ are nonlinear mappings. In black-box identification, various function approximators, such as multilayer perceptrons, radial basis functions and fuzzy models, can be used to model the system output in terms of past inputs and outputs [26]. Such black-box models do not provide much insight into the system dynamics, however. In order to capture the system properties more explicitly block-oriented representations have been introduced, where various system characteristics are captured by separate blocks. Two important model classes of this form are the Hammerstein and the Wiener models [2,9,11,23], where the nonlinearities are represented by a static block and the dynamic component is described by a linear block.

In this paper we focus on the Wiener class of models depicted in Fig. 1. The model consists of a dynamic linear block G followed by a static nonlinearity $\mathcal{N}$, and can be described as

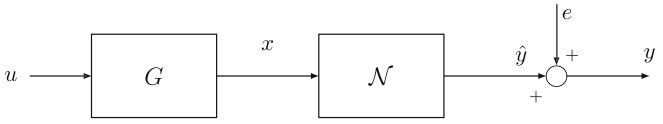

Fig. 1. Wiener model.

$$\begin{aligned} \mathbf{x}(n) &= G(q^{-1})u(n) \\ \hat{y}(n) &= \mathcal{N}(\mathbf{x}(n)) \\ y(n = \hat{y}(n) + e(n) \end{aligned} \tag{2}$$

where $\hat{y}(n)$ denotes the model output, which can be taken as a prediction of the system output, and $G(q^{-1})$ is a transfer function in the backward shift operator q^{-1} ($q^{-1}u(n) = u(n-1)$). It can be seen that the Wiener model is a special case of (1) where the dynamic component $f(\cdot,\cdot)$ is linear and $\mathcal{N}(\cdot)$ corresponds to the output nonlinearity $h(\cdot)$.

The identification problem studied in this paper can be stated as follows. We assume that a sequence of observed experimental input–output training data $\{u_t(k), y_t(k)\}$ has been collected from system (1). For clarity, subscript t is used to distinguish the inputs and outputs of the training sequence from general input and output sequences. The problem then is to find a Wiener model representation which explains the given data as well as possible. In the literature, a number of approaches addressing this problem have been proposed [9,10,13,23,25,29,33,35]. In identification of dynamical systems, it is important to distinguish between one-step ahead prediction error methods and output error, or multistep ahead prediction error methods [15], since the former approach requires restrictive assumptions on the model.

In one-step ahead prediction error methods the system parameters are determined to minimize a norm (usually the sum of squares) of the one-step ahead prediction errors

$$\hat{e}(k|k-1) = y_t(k) - \hat{y}(k|k-1) \tag{3}$$

where $\hat{y}(k|k-1)$ denotes the predicted model output based on observed data up to time instant $k-1$, i.e., $\{u_t(m), y_t(m), m = 1,2,\ldots,k-1\}$. Hence the model output $\hat{y}(k \mid k-1)$ is a function of measured outputs y_t as well as inputs u_t. However, for a static nonlinearity $\mathcal{N}(\cdot)$ the model output is a function of input u_t only. For the one-step ahead prediction error method to be meaningful, it should be possible to reconstruct the internal state $\mathbf{x}$ from the measured outputs y_t. This is possible in the special case where the static nonlinearity $\mathcal{N}(\cdot)$ is assumed invertible [9,13,23,33] (which requires that the state variable x is a scalar), so that the internal state associated with the measured outputs can be reconstructed as $x(k) = \mathcal{N}^{-1}(y_t(k))$.

In output error identification the system parameters are determined to minimize the output errors

$$\hat{e}(k) = y_t(k) - \hat{y}(k) \tag{4}$$

where $\hat{y}(k)$ denotes the model output which can be calculated using the input sequence $\{u_t(k)\}$ only (cf. (2)), whereas the observed measurements $\{y_t(k)\}$ are not used to compute the model output. Output error identification of Wiener models has been considered in [10,25,29,33,35].

Output error identification is in general computationally much more complex than one-step ahead prediction error identification, as even for linear transfer function models the output is a nonlinear function of the model parameters. It is in general also much harder to find a model which achieves small output errors, and selection of correct model structure is particularly important in output error identification of nonlinear systems. Notably, accurate one-step ahead predictions do not guarantee that the output errors are small as well [19]. In many applications where the model is

used for control or prediction purposes it is, however, important that the model gives accurate predictions of the system output several steps ahead.

In output error identification of linear systems, it has been found useful to introduce orthonormal filter expansions, using for example Laguerre or Kautz functions [18]. Stable causal systems can be approximated to any degree of accuracy (in $\mathscr{H}_2$ norm) by such basis filters. Moreover, by parameterizing a linear system model in terms of the weights of a filter expansion, the multistep ahead prediction of the output is a linear function of the parameters, simplifying the output error identification problem substantially.

The idea of representing the linear dynamic block using a basis filter expansion can be applied to Wiener models as well. The internal state variable is then vector valued, consisting of the outputs x_i of the individual filters used in the filter expansion, see Fig. 2. In this model, the linear dynamic part is described by the transfer functions G_i,

$$x_i(n) = G_i(q^{-1})u(n), \quad i = 1, 2, \ldots, M \tag{5}$$

The static nonlinearity is given by

$$\hat{y}(n) = \mathscr{N}(\mathbf{x}(n)) \tag{6}$$

where

$$\mathbf{x}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_M(n) \end{bmatrix} \tag{7}$$

In the original Laguerre–Volterra models [32], the linear systems G_i are taken as Laguerre filters and the nonlinearity is represented by a polynomial. More generally, the nonlinear block can be represented by general function approximators, such as artificial neural networks [34,29] or basis function expansions [9,25], while the dynamic part can be represented by appropriately selected generalized orthonormal filters [18], which are characterized by their pole locations. Laguerre filters [30], which have a single real-valued pole, are well suited for modelling well damped systems, while systems which show oscillatory behaviour are appropriately represented by Kautz filters [31], which are characterized by a pair of complex-valued poles. The pole locations are important design variables, which determine the filter order needed to achieve a given accuracy. The optimal choice of pole locations has been studied for Laguerre–Volterra models by [4] and for Kautz–Volterra models by [6].

The model structure in Fig. 2 has been used for output error identification for example by [25,29]. In [29] the nonlinearity was represented by a multilayer perceptron artificial neural network function approximator, resulting in a nonlinear least-squares estimation problem, whereas [25] used a basis function expansion whose parameters can be determined using linear least-squares estimation. In this paper, robust support vector regression is applied to identify the nonlinear block in Fig. 2.

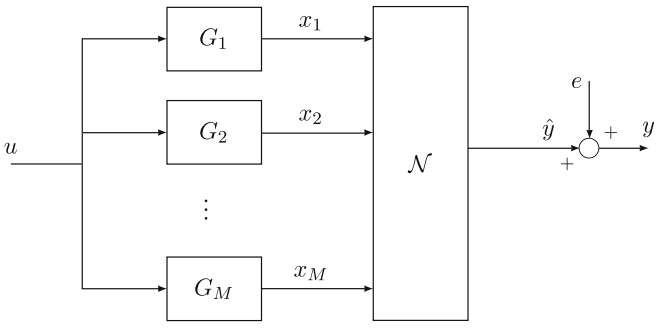

Fig. 2. Wiener model with basis function expansion of linear block.

3. Support vector identification method

In this section we will use a support machine method to identify the static block in the Wiener model (5)–(7). In this approach the nonlinear block $\mathscr{N}(\cdot)$ is expanded in terms of nonlinear mappings $g_i(\cdot)$ as

$$\mathscr{N}(\mathbf{x}(n)) = \sum_i w_i g_i(\mathbf{x}(n)) = \mathbf{w}^T \mathbf{g}(\mathbf{x}(n)) \tag{8}$$

where

$$\mathbf{w}^T = [w_1, w_2, \ldots] \tag{9}$$

and

$$\mathbf{g}(\mathbf{x}(n)) = \begin{bmatrix} g_1(\mathbf{x}(n)) \\ g_2(\mathbf{x}(n)) \\ \vdots \end{bmatrix} \tag{10}$$

A continuous nonlinear block $\mathscr{N}(\cdot)$ can be approximated to arbitrary accuracy by properly selected basis functions $\{g_i(\cdot)\}$. An important feature of support vector machine regression is that the nonlinear mappings $g_i(\cdot)$ need not be explicitly known. Instead, it turns out that they can be defined implicitly in terms of an associated kernel function [28], cf. below.

We consider the problem of identifying the weights w_i of the expansion (8) based on experimental data $\{u_t(k-1), y_t(k), k = 1, 2, \ldots, N\}$. Notice that as the filters G_i are specified, the internal state vector $\mathbf{x}_t(k), k = 1, 2, \ldots, N$ associated with the input sequence can be computed according to (5). Hence the experimental data can be represented in terms of the data pairs $\{\mathbf{x}_t(k), y_t(k), k = 1, 2, \ldots, N\}$. By (8), the model output error for the experimental data can be described as

$$\hat{e}_t(k) = y_t(k) - \mathbf{w}^T \mathbf{g}(\mathbf{x}_t(k)), \quad k = 1, 2, \ldots, N \tag{11}$$

Vapnik's ϵ-insensitive loss function [28] is defined as

$$L_\epsilon(\hat{e}_t(k)) = \begin{cases} |\hat{e}_t(k)| - \epsilon, & \text{if } |\hat{e}_t(k)| \geqslant \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Support vector regression consists of minimizing the structural risk function [28]

$$R_\epsilon(\mathbf{w}, \{\hat{e}_t(k)\}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{k=1}^{N} L_\epsilon(\hat{e}_t(k)) \tag{13}$$

Here the first term is a regularization term which bounds the approximation error by constraining the magnitudes of the weights in the expansion (8), while the second term is an empirical risk which describes the model accuracy for the experimental data. The tradeoff between the two terms is defined by the positive number C.

In order to minimize the structural risk function (13) it is convenient to rewrite the problem as the minimization of

$$R(\mathbf{w}, \xi, \xi^*) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{k=1}^{N}(\xi_k + \xi_k^*) \tag{14}$$

subject to the constraints

$$\begin{aligned} &y_t(k) - \mathbf{w}^T\mathbf{g}(\mathbf{x}_t(k)) \leqslant \epsilon + \xi_k \\ &-y_t(k) + \mathbf{w}^T\mathbf{g}(\mathbf{x}_t(k)) \leqslant \epsilon + \xi_k^*, \quad k = 1, 2, \ldots, N \end{aligned} \tag{15}$$

where $\xi_k, \xi_k^* \geqslant 0$ are slack variables. The solution is obtained as the saddle-point of the associated Lagrange function

$$L(\mathbf{w},\xi,\xi^*,\alpha,\alpha^*) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{k=1}^{N}(\xi_k+\xi_k^*) + \sum_{k=1}^{N}\alpha_k(y_t(k)-\mathbf{w}^T\mathbf{g}(\mathbf{x}_t(k))-\epsilon-\xi_k) + \sum_{k=1}^{N}\alpha_k^*(-y_t(k)+\mathbf{w}^T\mathbf{g}(\mathbf{x}_t(k))-\epsilon-\xi_k^*) \quad (16)$$

which should be minimized with respect to $\mathbf{w}$, ξ_k and ξ_k^*, and maximized with respect to the Lagrange multipliers $\alpha_k \geqslant 0, \alpha_k^* \geqslant 0$.

The saddle-point condition $\partial L/\partial \mathbf{w} = 0$ implies that the optimal weights can be expressed in terms of Lagrange multipliers as

$$\mathbf{w} = \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)\mathbf{g}(\mathbf{x}_t(k)) \quad (17)$$

Introducing the saddle-point condition (17) into (16) and observing that the terms involving ξ_k, ξ_k^* can be replaced by the constraints $\alpha_k, \alpha_k^* \leqslant C$, gives the dual cost

$$W(\alpha,\alpha^*) = -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)K(\mathbf{x}_t(k),\mathbf{x}_t(l)) + \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)y_t(k) - \epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*) \quad (18)$$

where $K(\cdot,\cdot)$ are kernel functions defined as

$$K(\mathbf{x}(n),\mathbf{x}(m)) = \mathbf{g}(\mathbf{x}(n))^T\mathbf{g}(\mathbf{x}(m)) \quad (19)$$

The solution is obtained by maximizing the dual cost (18) with respect to Lagrange multipliers $0 \leqslant \alpha_k, \alpha_k^* \leqslant C$.

Introducing the optimal weight according to (17) into (8) and (6) gives the SVR model

$$\hat{y}(n) = \mathcal{N}(\mathbf{x}(n)) = \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)K(\mathbf{x}(n),\mathbf{x}_t(k)) \quad (20)$$

where $\mathbf{x}(n)$ is defined by (5) and (7). A characteristic feature of the model (20) is that it includes only the support vectors $\mathbf{x}_t(k)$, such that one of the constraints (15) is active, i.e., α_k and α_k^* are not both zero.

The dual cost (18) and the support vector model (20) are defined in terms of the kernel function (19), and therefore there is no need to define the nonlinear mappings $g_i(\cdot)$ in (8) explicitly. For the problem to be well defined, the kernels should satisfy Mercer's conditions [28]. Commonly used kernels which satisfy Mercer's conditions are gaussian radial basis functions, polynomial kernels, multilayer perceptrons and splines.

Maximization of the dual cost (18) subject to the constraints $0 \leqslant \alpha_k, \alpha_k^* \leqslant C$ is a box-constrained quadratic programming problem in $2N$ variables. The global optimum of a quadratic programming problem can be found in a finite number of iterations, but for large data sets the optimization becomes computationally demanding due to the large number of constrained variables involved. However, special-purpose methods which exploit the particular structure of the cost (18) have been developed for efficient solution of support vector regression problems [5].

Application of the support vector method requires selection of design parameters ϵ in the loss function (12) and C in the structural risk function (14). Guidelines to find robust parameter values have been discussed in the literature, cf. for example [16] for a discussion in the context of dynamic system identification. The threshold value ϵ affects the model accuracy and the number of support vectors, which contribute to the sum in (20). As the support vectors consist of those components of the training data set for which the approximation error exceeds ϵ, it follows that the model complexity increases with decreasing ϵ. A standard robust choice is to select a value of ϵ for which approximately 30–50% of the training data points are support vectors. The parameter C in (13) determines the tradeoff between the empirical risk $L_\epsilon(\cdot)$ and the complexity penalty in the structural risk function. For a very small C, the complexity term will dominate, resulting in larger prediction errors. Increasing the value of C results in better approximation, but for excessively large C the relative importance of the prediction errors grows with respect to the complexity penalty term, which may result in large model weights $\mathbf{w}_i$ associated with overtraining and poor generalization properties.

As the parameter values are problem specific, it is in practice usually preferable to tune the parameters to the specific data at hand. This can be achieved by cross-validation techniques, in which selection of design parameters is based on the model performance on independent test data which are not used for identification. In practice the first half of the observed input–output sequence can be taken as a training sequence and used for identification, whereas the latter half is used for cross-validation. This method will be used in the numerical studies in Section 4.

Remark 3.1. Using the support vector Wiener model to compute an estimate $\hat{y}(n)$ of the system output requires evaluation of (20), where $\mathbf{x}(n)$ is calculated from the system inputs according to (5) and (7), and the support vectors $\mathbf{x}_t(k)$ are determined a priori in a similar way from the training data. The support vectors correspond to the active constraints of the quadratic programming problem defined by (18). Hence the number of support vector may be rather large, implying a complex model. Therefore, methods to reduce model complexity in support vector regression is a very important topic of research [14].

Remark 3.2. For the sake of clarity, the identification method has been formulated for single-input single-output system. At the cost of increased complexity, it is, however, straightforward to apply the procedure to multivariable systems as well. In this case, basis filters are introduced in analogy with (5) for each individual input, and separate support vector models of the form (20) are then identified for each output.

4. Numerical example

The purpose of the following simulated example is to illustrate the properties of the identification method described in Section 3. In [33,10] a valve for fluid flow control was modelled by the equations

$$x(n) - 1.4138x(n-1) + 0.6065x(n-2) = 0.1044u(n-1) + 0.0883u(n-2) \quad (21)$$

$$y_0(n) = \frac{x(n)}{\sqrt{0.10 + 0.90x(n)^2}} \quad (22)$$

where $u(n)$ is a pneumatic control signal applied to the valve, $x(n)$ is the position of the valve plug and $y_0(n)$ is the fluid flow. The fluid flow measurement $y(n)$ is given by

$$y(n) = y_0(n) + e(n) \quad (23)$$

where $e(n)$ is measurement noise. The model (21) and (22) is a Wiener model, with a second-order linear dynamic block having the complex conjugate poles $0.7069 \pm 0.3268j$.

In this example the noise $e(n)$ consists of independent normally distributed random numbers with standard deviation 0.05. The input u was generated using a pseudo-random binary signal alternating between -1 and $+1$ and having a basic clock period of seven sampling intervals. In each time interval of constant signal level, the input was multiplied by a random factor, uniformly distributed

number between 0 and 0.4 added on a bias of 0.5. This gives the signal an amplitude between 0.1 and 0.9. The same procedure of generating the input signal was used in [10,33]. Two input–output sequences consisting of 1000 samples each were generated as described above. The training data sequence shown in Fig. 3 was used for model identification, while the test data sequence in Fig. 4 was used for model validation.

Although the system is described by a Wiener model, no knowledge is assumed about the dynamics or the structure of the static nonlinearity. Here the linear part (5) of the Wiener model was expanded using Laguerre filters [30], i.e.,

$$G_i(q^{-1}) = \frac{q^{-1}\sqrt{1-\alpha^2}}{1-\alpha q^{-1}}\left(\frac{q^{-1}-\alpha}{1-\alpha q^{-1}}\right)^{i-1}, \quad i = 1, 2, \ldots, M \tag{24}$$

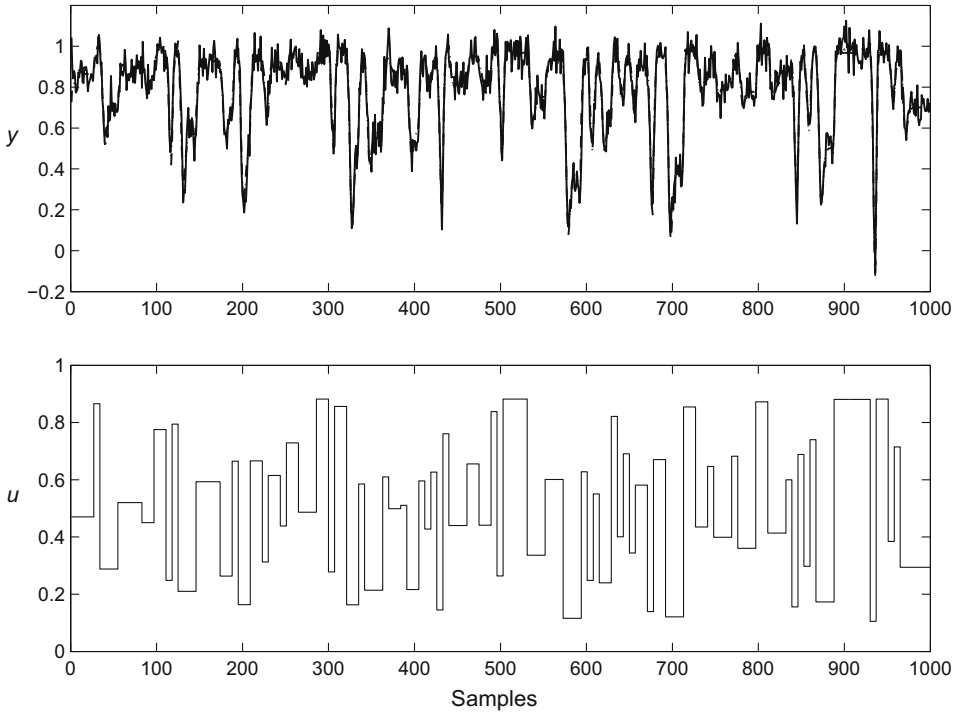

Fig. 3. Training data.

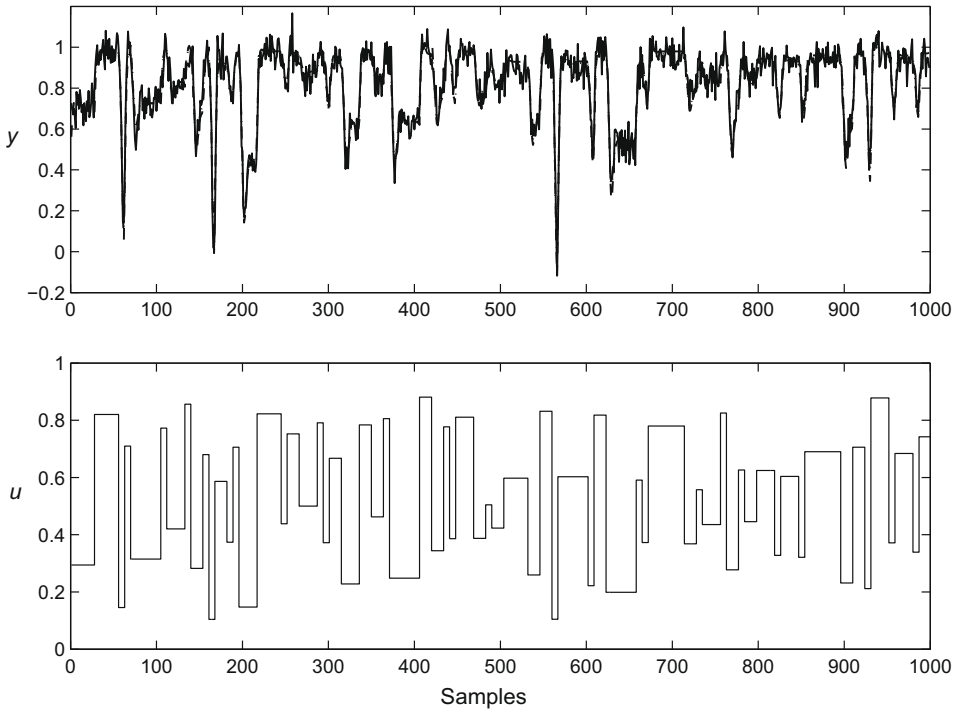

Fig. 4. Test data.

The static nonlinear block of the support vector Wiener model (cf. Eq. (20)) was represented by gaussian kernels

$$K(\mathbf{x}(n), \mathbf{x}(m)) = e^{-\gamma||\mathbf{x}(n)-\mathbf{x}(m)||^2} \quad (25)$$

The Laguerre filter pole α, filter order M and kernel parameter $\gamma > 0$ are design parameters to be selected.

Support vector Wiener models based on the training sequence in Fig. 3 were computed using the support vector machine program package LIBSVM [5]. The SVR design parameters were selected by cross-validation to minimize the root mean square error (RMSE) for the test data sequence. Tables 1 and 2 show how the identification results depend on the choice of design parameters. Table 1 shows the RMSEs achieved with various filter poles and orders. It is seen that filter order $M = 5$ minimizes the RMSE on test data and results in the smallest number of support vectors. For higher filter orders, overtraining results in larger mean square errors. As expected, increasing filter order results in smaller mean square errors for training data. It is also seen that the value of Laguerre filter pole is not very critical as long as extreme values (close to 0 or 1) are avoided. Table 2 shows how the identification results depend on the choice of support vector design parameters. It is seen that a small ϵ (high prediction accuracy) results in smaller RMSEs at the expense of increased model complexity (support vectors).

The cross-validation approach gives SVR model accuracy $\epsilon = 0.08$, weight parameter $C = 2000$, and kernel parameter $\gamma = 0.1$. The Laguerre filter pole is $\alpha = 0.4$ and filter order is $M = 5$. The RMSEs between the model output $\hat{y}(n)$ and the measured system output $y(n)$ are 0.0541 for the training data and 0.0556 for the test data, and between the model output and the noise-free system output $y_0(n)$ the values are 0.0218 and 0.0191, respectively. Fig. 5 shows the model output together with measured and noiseless system outputs during the last 100 samples for the test data. The output errors $\hat{y}(n) - y(n)$ and $\hat{y}(n) - y_0(n)$ are shown in Fig. 6. The results show that the identified model describes training data

Table 1
Root mean square errors between system output y and output $\hat{y}$ of identified models in example using various Laguerre filter poles α and filter orders M. The design parameters were taken as $\epsilon = 0.08$, $C = 2000$ and $\gamma = 0.1$. $\hat{y}$ denotes the output of the identified Wiener model and y is the system output (23). The number of support vectors (NSV) is also given.

α	M	RMSE($y - \hat{y}$)		NSV
		Training	Test	
0.4	1	0.1070	0.0967	337
	2	0.0761	0.0729	224
	3	0.0697	0.0680	212
	4	0.0564	0.0577	165
	5	0.0541	0.0556	146
	6	0.0540	0.0559	161
	8	0.0525	0.0556	168
0.1	5	0.0697	0.0694	210
0.2		0.0623	0.0634	194
0.3		0.0561	0.0575	160
0.5		0.0541	0.0564	152
0.6		0.0523	0.0553	153
0.7		0.0528	0.0623	176

Table 2
Root mean square errors between system output y and output $\hat{y}$ of identified models in example using various values for design parameters ϵ, C and γ. A Laguerre-filter expansion with pole $\alpha = 0.4$ and order $M = 5$ was used. $\hat{y}$ denotes the output of the identified Wiener model and y is the system output (23). The number of support vectors (NSV) is also given.

ϵ	C	γ	RMSE($y - \hat{y}$		NSV
			Training	Test	
0.08	200	0.1	0.0571	0.0567	146
	1000		0.0548	0.0560	148
	2000		0.0541	0.0556	146
	4000		0.0539	0.0560	150
	20,000		0.0538	0.0571	156
0.08	2000	0.01	0.0630	0.0611	167
		0.2	0.0537	0.0567	150
		1	0.0536	0.0783	187
0.04	2000	0.1	0.0531	0.0542	478
0.12			0.0573	0.0589	41

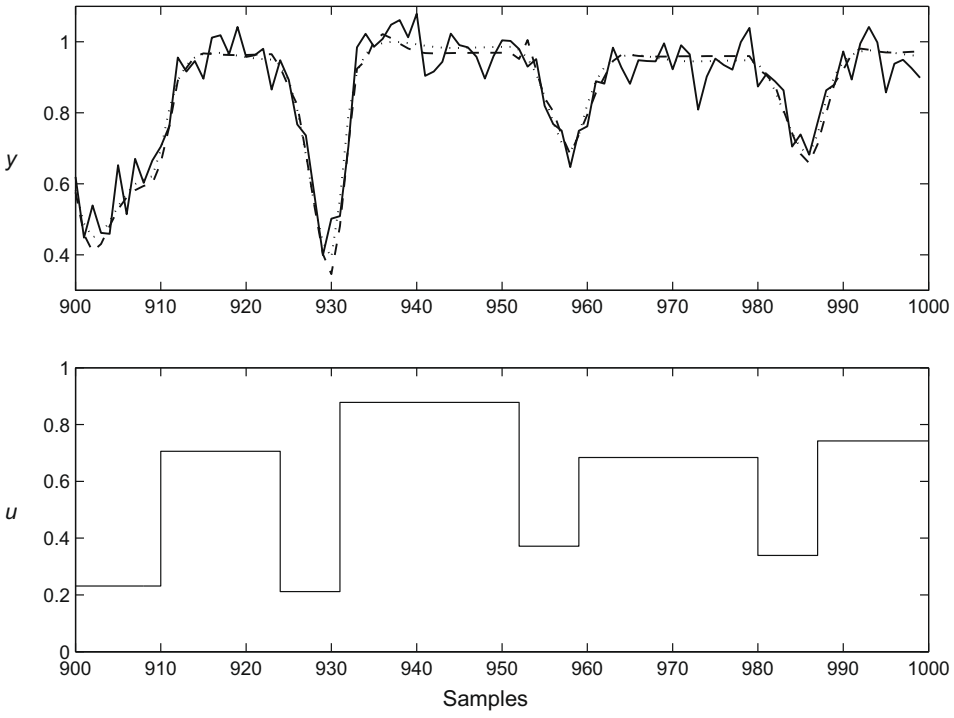

Fig. 5. Test data (last 100 samples). Measured system output $y(n)$ (solid line), noiseless output $y_0(n)$ (dashed line) and output of identified model $\hat{y}(n)$ (dotted line).

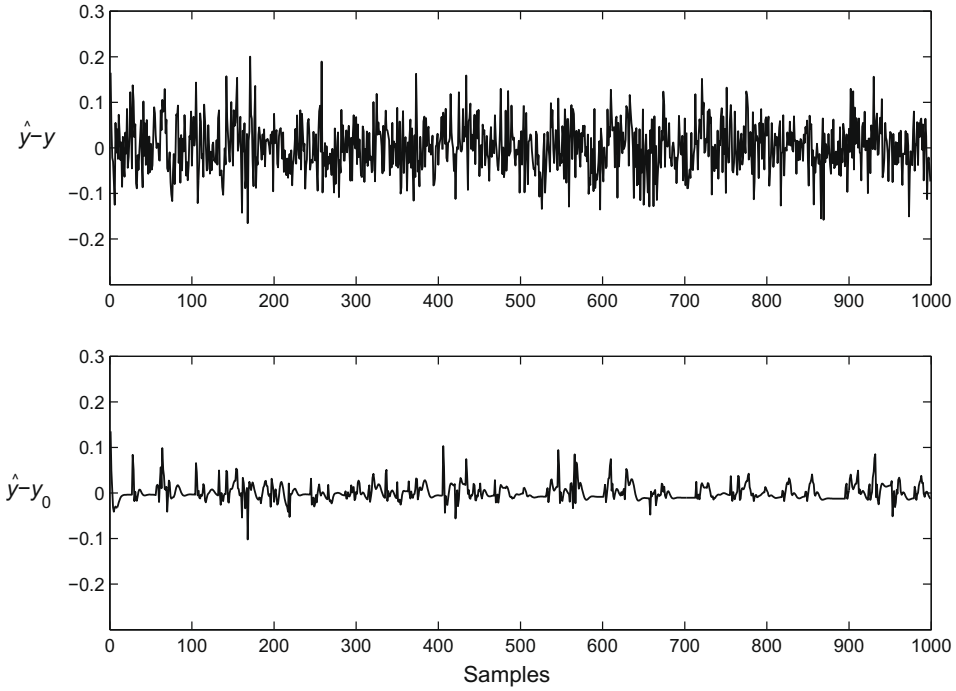

Fig. 6. Output error for test data between model output $\hat{y}(n)$ and measured output $y(n)$ (upper diagram) and noise-free output $y_0(n)$ (lower diagram).

and independent test data with equal accuracy, and hence it can be concluded that the identified model incorporates the relevant dynamic behaviour of the system.

The accuracy achieved with the support vector regression method can be compared to results obtained using a least-squares prediction error method, in which the mean square error on training data is minimized. For the same system and noise characteristics, Hagenblad [10] reports the minimum RMSE 0.049 (mean square error 0.0024) on training data. Hence the support vector method achieves comparable accuracy in terms of RMSE, although the mean square error is not explicitly minimized in this method.

It should be noted that system (21), (22) studied in this example has a model structure which is compatible with (2), and Wiener model identification can therefore be expected to perform well on this system. In general, it is important that the selected model structure is compatible with the dynamic behaviour of the system [2,22,23], and for other systems some other model class, such as Hammerstein models, Wiener Hammerstein cascade models or general nonlinear ARX models, may be a better choice.

5. Conclusion

Support vector regression has been applied to identify nonlinear systems described by Wiener models, where the linear component of the model is represented by a basis filter expansion, and the static nonlinear block is represented by kernel functions. In this model structure, the model output is determined as a function of the system input only, and it does not depend on the measured system outputs. Similar Wiener model structures have been used by [9,34,29]. The support vector method is well suited to this kind of system identification problem, as the model can be determined by solving a convex quadratic minimization problem, for which convergence to the global optimum is obtained. In contrast, methods where the nonlinearity is described by a multilayer perceptron [29,34] require the solution of nonlinear minimization problem which may have local minima. Moreover, support vector regression is based on structural risk minimization, which introduces robust performance with respect to new data.

Numerical simulations show that the procedure gives accurate models of systems which have a Wiener model structure, and for a Wiener model benchmark problem an output error comparable to the error obtained using a least-squares prediction error method reported in the literature was achieved.

Acknowledgement

Stefan Tötterman was supported by the Finnish Graduate School in Chemical Engineering (GSCE) and Åbo Akademi Foundation (Stiftelsen för Åbo Akademi).

References

[1] S. Adachi, T. Ogawa, A new system identification method based on support vector machines, in: IFAC Workshop Adaptation and Learning in Control and Signal Processing, Italy, 2001.
[2] L.A. Aguirre, M.C.S. Coelho, M.V. Corrêa, On the interpretation and practice of dynamical differences between Hammerstein and Wiener models, IEE Proceedings-Control Theory and Applications 152 (5) (2005) 349–356.
[3] S. Boyd, L.O. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series, IEEE Transactions on Circuits and Systems 32 (11) (1985) 1150–1161.
[4] R.J.G.B. Campello, G. Favier, W.C. do Amaral, Optimal expansions of discrete-time Volterra models using Laguerre functions, Automatica 40 (5) (2004) 815–822.
[5] C.-C. Chang, C.-J. Lin, LIBSVM: A Libray for Support Vector Machines, 2001, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
[6] A. da Rosa, R.J.G.B. Campello, W.C. Amaral, Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions, Automatica 43 (12) (2007) 1084–1091.
[7] M. Espinoza, J.A.K. Suykens, B. De Moor, Kernel based partially linear models and nonlinear identification, IEEE Transactions on Automatic Control 50 (10) (2005) 1602–1606.
[8] I. Goethals, K. Pelckmans, J.A.K. Suykens, B. De Moor, Identification of MIMO Hammerstein models using least-squares support vector machines, Automatica 41 (7) (2005) 1263–1272.
[9] J.C. Gómez, E. Baeyens, Identification of block-oriented nonlinear systems using orthonormal bases, Journal of Process Control 14 (6) (2004) 685–697.
[10] A. Hagenblad, Aspects of the Identification of Wiener Models, Linköping Studies in Science and Technology, Thesis No. 793, 1999.
[11] I.W. Hunter, M.J. Korenberg, The identification of nonlinear biological systems: Wiener and Hammerstein cascade models, Biological Cybernetics 55 (1986) 134–144.

[12] M. Isaksson, D. Rönnow, A Kautz–Volterra behavioral model for RF power amplifiers, in: IEEE Mtt-S International Microwave Symposium Digest, 2006, pp. 485–488.
[13] A. Kalafatis, N. Arifin, L. Wang, W. Cluett, A new approach to the identification of pH processes based on Wiener model, Chemical Engineering Science 50 (23) (1995) 3693–3701.
[14] K.-M. Lin, C.J. Lin, A study on reduced support vector machines, IEEE Transactions on Beural Networks 14 (6) (2003) 1449–1459.
[15] L. Ljung, System Identification: Theory for the User, Prentice-Hall, Englewood Cliffs, 1987.
[16] D. Mattera, S. Haykin, Support vector machines for dynamic reconstruction of a chaotic system, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 211–241.
[17] K.-R. Müller, A.J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 243–254.
[18] B. Ninness, H. Hjalmarsson, F. Gustafsson, The fundamental role of generalized orthonormal basis in system identification, IEEE Transactions on Automatic Control 44 (7) (1999) 1384–1406.
[19] M. Nørgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, London, 2000.
[20] T. Ogunfunmi, Adaptive Nonlinear System Identification, The Volterra and Wiener Model Approaches, Springer, New York, 2007.
[21] G.A. Pajunen, Adaptive control of Wiener type nonlinear systems, Automatica 28 (1992) 781–785.
[22] R.K. Pearson, Nonlinear input/output modelling, Journal of Process Control 5 (4) (1995) 197–211.
[23] R.K. Pearson, M. Pottmann, Gray-box identification of block-oriented models, Journal of Process Control 10 (4) (2000) 301–315.
[24] J.L. Rojo-Álvarez, M. Martínez-Ramón, M. de Prado-Cumplido, A. Artés-Rodríguez, A.R. Figueiras-Vidal, Support vector method for robust ARMA system identification, IEEE Transactions on Signal Processing 52 (1) (2004) 155–164.
[25] D. Sbárbaro, T.A. Johansen, Multiple local Laguerre models for modelling nonlinear dynamic systems of the Wiener class, IEE Proceedings-Control Theory and Applications 144 (5) (1997) 375–380.
[26] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, B. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (12) (1995) 1691–1724.
[27] H.T. Toivonen, S. Tötterman, B. Åkesson, Identification of state-dependent parameter models with support vector regression, International Journal of Control 80 (2007) 1454–1470.
[28] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[29] A. Visala, H. Pitkänen, A. Halme, Modeling of chromatographic separation process with Wiener-MLP representation, Journal of Process Control 11 (5) (2001) 443–458.
[30] B. Wahlberg, System identification using Laguerre models, IEEE Transactions on Automatic Control 36 (5) (1991) 551–562.
[31] B. Wahlberg, System identification using Kautz models, IEEE Transactions on Automatic Control 39 (6) (1994) 1276–1282.
[32] N. Wiener, Nonlinear Problems in Random Theory, Technology Press, M.I.T. and Wiley, New York, 1958.
[33] T. Wigren, Recursive prediction error identification using the nonlinear Wiener model, Automatica 29 (4) (1993) 1011–1025.
[34] J. Wray, G.G.R. Green, Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network, Biological Cybernetics 71 (1994) 187–195.
[35] Q. Zheng, E. Zafiriou, Volterra–Laguerre models for nonlinear process identification with application to a fluid cracker unit, Industrial and Engineering Chemistry Research 43 (2) (2004) 340–348.