Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Subspace intersection identification of Hammerstein-Wiener systems

Ivan Goethals, Kristiaan Pelckmans, Luc Hoegaerts, Johan Suykens and Bart De Moor
KULeuven - ESAT - SCD/SISTA
Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium
Ivan.Goethals@esat.kuleuven.ac.be

Abstract— **In this paper, a method for the identification of Hammerstein-Wiener systems is presented. The method extends the linear subspace intersection algorithm, mainly by introducing a kernel canonical correlation analysis (KCCA) to calculate the state as the intersection of past and future. The linear model and static nonlinearities are obtained from a regression problem using componentwise Least Squares Support Vector Machines (LS-SVMs).**

I. Introduction

In this paper, we will consider the extension of the classical subspace intersection algorithm [15] to Hammerstein-Wiener systems in state-space form:

$$\begin{cases} x_{t+1} &= Ax_t + Bf(u_t), \\ g^{-1}(y_t) &= Cx_t + Du_t, \qquad \forall t \end{cases} \tag{1}$$

where $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^l$ are the input and output at time t and $x_t \in \mathbb{R}^n$ denotes the state. $f : \mathbb{R}^m \to \mathbb{R}^m$ and $g : \mathbb{R}^l \to \mathbb{R}^l$ are static nonlinear maps with g such that g^{-1} exists for all possible outputs of the system. The extension is obtained by replacing the linear CCA-step, used for the estimation of the state by a kernel CCA (KCCA) approximator. In a second step, the system matrices A, B, C and D and the nonlinearities f and g are obtained from the solution of a componentwise Least Squares - Support Vector Machine (LS-SVM) regression problem, which was earlier used in an extension of the N4SID [11] identification algorithm towards Hammerstein systems.

In [3], a scheme for the identification of SISO Hammerstein-Wiener systems is developed based on the idea of overparametrization [5]. However, in this scheme a very specific model structure is assumed, limiting its practical applicability. Based on [3], a more general blind approach for the identification of SISO systems was proposed in [4]. An identification method for Hammerstein-Wiener MIMO systems was proposed in [7], [6] but imposes certain restrictions on the inputs and is iterative in nature. Other contributions such as [9], [21] are limited to SISO systems and/or iterative in nature.

In contrast to these methods, a clear advantage of the proposed technique in this paper is that it does not rely on restrictive assumptions on the inputs, except for the well known persistency of excitation [19], that it is non-iterative in nature, and that it can conveniently be applied to MIMO systems. Furthermore, other than the invertibility of g and a certain degree of smoothness, no specific restrictions are imposed on the nonlinear maps f and g.

Due to space limitations this paper mainly focuses on the estimation step of the state. The subsequent estimation of the system matrices and the nonlinearities f and g is only very briefly commented upon and the reader is kindly refered to a companion paper [10] for further reading on this subject. The outline of this paper is as follows: in Section II the basic ingredients of the subspace intersection algorithm for linear systems are reviewed briefly. Section III extends the linear intersection algorithm towards a nonlinear setting using a variation on the theme of LS-SVMs and kernel CCA. Section IV, finally, presents some illustrative examples.

As a general rule in this paper, lowercase symbols will be used to denote column vectors. Uppercase symbols are used for matrices. Elements of matrices and vectors are selected using Matlab standards, e.g. $A(i,j)$ denotes the ij^{th} entry of a matrix A, and $A(:,i)$ symbolizes the i^{th} column of the same matrix. Estimates for a parameter x will be denoted by $\hat{x}$. The symbol $\triangleq$ is used for definitions.

II. The subspace intersection algorithm

The subspace algorithm considered in this paper was originally proposed in [8], [15] and is largely based on the idea that the state of a linear or nonlinear model can be considered as the intersection between past and future measurement data [14].

The subspace intersection algorithm identifies deterministic models of the form

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t, \qquad \forall t \end{cases} \tag{2}$$

for all $t = 0, \ldots, N-1$ and with $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^l$ the input and output at time t. $x_t \in \mathbb{R}^n$ denotes the state. Based on a finite set of training data $\{(u_t, y_t)\}_{t=0}^{N-1}$, intersection algorithms are concerned with finding an estimate for the model order n of the system (2), and estimates for the system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$ and $D \in \mathbb{R}^{l\times m}$ up to a similarity transformation. Block Hankel matrices play an important role in these algorithms. The input block Hankel matrices are defined as

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccc} u_0 & \cdots & u_{j-1} \\ \vdots & & \vdots \\ u_{i-1} & \cdots & u_{i+j-2} \\ \hline u_i & \cdots & u_{i+j-1} \\ \vdots & & \vdots \\ u_{2i-1} & \cdots & u_{2i+j-2} \end{array}\right] \triangleq \left[\begin{array}{c} U_p \\ \hline U_f \end{array}\right] \in \mathbb{R}^{2im\times j},$$

0-7803-9568-9/05/$20.00 ©2005 IEEE

with i and j user defined indices such that $2i + j - 1 = N$. The output block Hankel matrices $Y_p, Y_f \in \mathbb{R}^{il \times j}$ are defined in a similar way. The joint past and future W_p and W_f are defined as $W_p \triangleq \begin{bmatrix} U_p^T & Y_p^T \end{bmatrix}^T$, $W_f \triangleq \begin{bmatrix} U_f^T & Y_f^T \end{bmatrix}^T$. Finally, $X_p \triangleq \begin{bmatrix} x_0 & x_1 & \dots & x_{j-1} \end{bmatrix}$ and $X_f \triangleq \begin{bmatrix} x_i & x_{i+1} & \dots & x_{i+j-1} \end{bmatrix}$, are introduced as finite state sequences of length j. The main reasoning behind subspace intersection algorithms follows from the fact that under the assumptions that:

1) the input u_t is persistently exciting of order $2i$, i.e. the input block Hankel matrix $U_{0|2i-1}$ is of full rank,
2) The intersection of the row space of U_f (the future inputs) and the row space of X_p (the past states) is empty,

the following relation holds: $\mathrm{row(X_f)} = \mathrm{row(W_p)} \cap \mathrm{row(W_f)}$. Hence, the order of the system and a realization for the state can be obtained from the intersection of past and future. Mathematically, this step is typically performed using a CCA algorithm, and retaining the canonical variates corresponding to canonical angles equal to 1. Once the state is known, extraction of A, B, C and D is straightforward.

Without going into further theoretical details of the subspace intersection algorithm (interested readers are referred to [8], [15], we summarize here a practical implementation that will be used towards the Hammerstein-Wiener model extension:

1) Perform canonical correlation analysis on W_p and W_f:

$$\begin{aligned} W_p W_f^T V_f &= W_p W_p^T V_p \Lambda, \\ W_f W_p^T V_p &= W_f W_f^T V_f \Lambda, \end{aligned} \tag{3}$$

with Λ a diagonal matrix containing the canonical correlations.

2) Determine the order n from the number of canonical correlations equal to one. Retain X_f as the n corresponding canonical variates in W_p.

$$X_f = V_p(:, 1:n)^T W_p.$$

3) Extract A, B, C and D from:

$$\begin{bmatrix} X_f(:, 2:j) \\ Y_{i|i}(:, 1:j-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_f(:, 1:j-1) \\ U_{i|i}(:, 1:j-1) \end{bmatrix}.$$

The intersection algorithm is a so-called deterministic subspace identification algorithm, in that no process and measurement noise are assumed. However, under certain conditions outlined in [15] the method is known to be consistent, even in the presence of process and measurement noise. We refer the reader to [15] for further reading on this subject.

III. Hammerstein-Wiener subspace intersection

A. Introducing the static nonlinearities

Equation (2) is transformed into a Hammerstein-Wiener system by introducing two static nonlinearities $f : \mathbb{R}^m \to \mathbb{R}^m$ and $g : \mathbb{R}^l \to \mathbb{R}^l$. With this definition for the nonlinearities, and assuming that $g : \mathbb{R}^l \to \mathbb{R}^l$ is such that g^{-1} exists for all possible outputs of the system, equation (2) is rewritten as:

$$\left\{ \begin{aligned} x_{t+1} &= A x_t + B f(u_t), \\ g^{-1}(y_t) &= C x_t + D f(u_t), \qquad \forall t. \end{aligned} \right. \tag{4}$$

As mentioned in Section II, a CCA algorithm could be used to extract the state x if $f(u)$ and $g^{-1}(y)$ were known. The state is then obtained from

$$\begin{aligned} \mathcal{F}(W_p)\mathcal{F}(W_f)^T V_f &= \mathcal{F}(W_p)\mathcal{F}(W_p)^T V_p \Lambda, \\ \mathcal{F}(W_f)\mathcal{F}(W_p)^T V_p &= \mathcal{F}(W_f)\mathcal{F}(W_f)^T V_f \Lambda, \end{aligned}$$

where $\mathcal{F}(W_p)$ is defined as follows

$$\mathcal{F}(W_p) \triangleq \begin{bmatrix} f(u_0) & f(u_1) & \dots & f(u_{j-1}) \\ \vdots & \vdots & & \vdots \\ f(u_{i-1}) & f(u_i) & & f(u_{i+j-2}) \\ g^{-1}(y_0) & g^{-1}(y_1) & \dots & g^{-1}(y_{j-1}) \\ \vdots & \vdots & & \vdots \\ g^{-1}(y_{i-1}) & g^{-1}(y_i) & & g^{-1}(y_{i+j-2}) \end{bmatrix}$$

with an equivalent definition for $\mathcal{F}(W_f)$. However, because $f(u)$ and $g^{-1}(y)$ are unknown, another approach is required to extract the state. A well-suited technique to fulfill this task is kernel CCA, a nonlinear extension of CCA, which will be treated in the following subsection. Once the state is known, $f(u)$ and $g^{-1}(y)$ will be estimated in a second step using LS-SVM regerssion.

B. Introducing the kernel

To extract a state of a *nonlinear* dynamical system, a nonlinear extension of CCA is employed, known as kernel CCA or KCCA [13], [2]. In kernel methods [18] the available data are mapped into a high-dimensional feature space of dimension n_H, where classical CCA is applied. The nonlinearity is condensed in the transformation, which is represented by feature maps $\varphi^u : \mathbb{R}^m \to \mathbb{R}^{n_H}$ and $\varphi^y : \mathbb{R}^l \to \mathbb{R}^{n_H}$. Using the mapped past data points $\varphi^u(u)$ and $\varphi^y(y)$, one constructs a feature matrix

$$\Phi_p \triangleq \Phi(W_p) \triangleq \begin{bmatrix} \varphi^u(u_0) & \varphi^u(u_1) & \dots & \varphi^u(u_{j-1}) \\ \vdots & \vdots & & \vdots \\ \varphi^u(u_{i-1}) & \varphi^u(u_i) & & \varphi^u(u_{i+j-2}) \\ \varphi^y(y_0) & \varphi^y(y_1) & \dots & \varphi^y(y_{j-1}) \\ \vdots & \vdots & & \vdots \\ \varphi^y(y_{i-1}) & \varphi^y(y_i) & & \varphi^y(y_{i+j-2}) \end{bmatrix}, \tag{5}$$

with a similar definition for $\Phi_f \triangleq \Phi(W_f)$.

In the kernel method context the feature maps are assumed to be associated with kernels $K^u : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R} : (u_s, u_t) \mapsto K^u(u_s, u_t)$ and $K^y : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R} : (y_s, y_t) \mapsto K^y(y_s, y_t)$. These kernels are bilinear functions that serve as similarity measure between data points ($s, t = 0, \dots, N-1$). If the kernels are symmetric and positive definite then they are referred to as Mercer kernels [1] and it can be shown that feature maps are implicitly defined by the kernels such that scalar products between mapped points equal kernel evaluations:

$$\begin{aligned}\varphi^u(u_s)^T\varphi^u(u_t) &= K^u(u_s,u_t),\\ \varphi^y(y_s)^T\varphi^y(y_t) &= K^y(y_s,y_t).\end{aligned}$$

This property allows in practice to circumvent working with the high-dimensional vectors (as long as only scalar products appear), and instead perform computations with (elements of) the $j\times j$ kernel Gram matrices $K_p \triangleq \Phi_p^T\Phi_p$ and $K_f \triangleq \Phi_f^T\Phi_f$. With the feature matrix introduced in Eq. (5) we adopted a so-called ANOVA kernel [20], [16].

C. From CCA to KCCA; the state estimate

For reasons of clarity of presentation we adopt here a formal introduction into the KCCA algorithm as it was initially presented in [13], [2]. For a more rigorous description of the main concepts behind KCCA, the reader is kindly referred to the latter references.

By mapping the elements of W_p and W_f the CCA problem in feature space becomes:

$$\begin{aligned}\Phi_p\Phi_f^T\ V_f &= \Phi_p\Phi_p^T\ V_p\ \Lambda,\\ \Phi_f\Phi_p^T\ V_p &= \Phi_f\Phi_f^T\ V_f\ \Lambda,\end{aligned}\tag{6}$$

Remark that the coefficient matrices V_p, V_f are elements of $\mathbb{R}^{2i(m+l)n_H\times 2i(m+l)n_H}$ where n_H can be potentially infinite-dimensional, which is not practical. However, if these matrices are restricted to the subspace spanned by the mapped data by redefining:

$$\mathcal{V}_p = \Phi_p V_p,\ \ \mathcal{V}_f = \Phi_f V_f,\tag{7}$$

and the first and second equation of (6) are left multiplied by Φ_p^T and Φ_f^T, respectively, we obtain:

$$\begin{aligned}K_pK_f\ \mathcal{V}_f &= K_pK_p\ \mathcal{V}_p\ \Lambda,\\ K_fK_p\ \mathcal{V}_p &= K_fK_f\ \mathcal{V}_f\ \Lambda.\end{aligned}\tag{8}$$

Assuming that K_p and K_f are invertible, which is hypothetically the case for RBF kernels (but not necessarily for linear kernels), this can further be reduced to

$$\begin{aligned}K_f\ \mathcal{V}_f &= K_p\ \mathcal{V}_p\ \Lambda\\ K_p\ \mathcal{V}_p &= K_f\ \mathcal{V}_f\ \Lambda,\end{aligned}\tag{9}$$

which is the classical form of the KCCA algorithm as presented in [13], [2]. A disadvantage of this KCCA version is the fact that the used kernel derivations do not contain regularization leaving the possibility of a severe over-fitting of the nonlinearities involved.

The KCCA version proposed in [18] is formulated using a support vector machine approach [20] with primal and dual characterizations for the optimization problems and an additional centering of the data-points in feature space. Regularization is thereby incorporated within the primal formulation in a well-established manner leading to numerically better conditioned solutions. Without going into the details of this algorithm (the interested reader is kindly referred to [18] and Appendix B), we state here the obtained generalized eigenvalue problem:

$$\begin{aligned}K_f^c\ \mathcal{V}_f &= \left(K_p^c + \frac{1}{\gamma}I_j\right)\ \mathcal{V}_p\ \Lambda,\\ K_p^c\ \mathcal{V}_p &= \left(K_f^c + \frac{1}{\gamma}I_j\right)\ \mathcal{V}_f\ \Lambda,\end{aligned}$$

where $\mu_p = (1/j)\sum_{s=1}^{j}\Phi_p(:,s)$ and $\mu_f = (1/j)\sum_{s=1}^{j}\Phi_f(:,s)$ are the expected centers of the mapped past and future, and with

$$\begin{aligned}K_p^c &= (\Phi_p - 1_j^T\otimes\mu_p)^T(\Phi_p - 1_j^T\otimes\mu_p),\\ K_f^c &= (\Phi_f - 1_j^T\otimes\mu_f)^T(\Phi_f - 1_j^T\otimes\mu_f),\end{aligned}$$

the centered kernels. The symbol $\otimes$ denotes the matrix Kronecker product. The tuning parameter γ controls the amount of regularization. Comparison with the derived result without centering yields that $K_f^c = M_cK_fM_c$ with $M_c = (I_j - (1/j)11_j^T)$ [17].

Thus by solving a generalized eigenvalue problem in the dual space, one can find the correlations and the nonlinear canonical variates, gathered resp. in the KCCA estimates $\widehat{\Lambda}$, $\widehat{V}_p$ and $\widehat{V}_f$. From the number of canonical correlations which are equal to one, we determine the order n. The estimated state is obtained as the n corresponding linear combinations of the centered variates in Φ_p. Hence, the final state is obtained as:

$$\widehat{X}_f = \widehat{\mathcal{V}}_p(1:n,:)^TK_p^c.\tag{10}$$

In the following subsection we will show how the state (10) can be used to obtain estimates for A, B and f.

D. Estimation of A, B and the nonlinear function f

Once state estimates are obtained, estimates for f and the system matrices A and B can be obtained in a second step as follows:

$$(\widehat{A},\widehat{B},\hat{f}) = \underset{A,B,f}{\arg\min}\left\|\widehat{X}_{i+1} - \begin{bmatrix}A & B\end{bmatrix}\begin{bmatrix}\widehat{X}_i\\ \mathcal{U}_f\end{bmatrix}\right\|_F^2,\tag{11}$$

with

$$\begin{aligned}\widehat{X}_i &\triangleq \widehat{X}_f(:,1:j-1),\\ \widehat{X}_{i+1} &\triangleq \widehat{X}_f(:,2:j),\\ \mathcal{U}_f &\triangleq \begin{bmatrix}f(u_i) & f(u_{i+1}) & \dots & f(u_{i+j-2})\end{bmatrix},\end{aligned}$$

It will be shown below that this least-squares problem can be written as a classical LS-SVM regression problem (see Appendix A for a brief introduction into LS-SVM regression). The first step towards such a regression problem is to make the replacement

$$Bf = \begin{bmatrix}w_{f,1}^T\\ w_{f,2}^T\\ \vdots\\ w_{f,n}^T\end{bmatrix}\varphi^u,\tag{12}$$

with φ^u the feature-map introduced in Section III. With this replacement, equation (11) is rewritten as

$$(\widehat{A},\widehat{B},\hat{f}) = \underset{A,B,f}{\arg\min} \left\| \widehat{X}_{i+1} - A\widehat{X}_i - \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \mathcal{U}_\varphi \right\|_F^2 ,$$

with

$$\mathcal{U}_\varphi \triangleq \begin{bmatrix} \varphi^u(u_i) & \varphi^u(u_{i+1}) & \ldots & \varphi^u(u_{i+j-2}) \end{bmatrix}.$$

In a companion paper [10], it is shown that this problem can readily be solved for A, B and f using the concept of componentwise LS-SVM regression. We refer the reader to [10] for the full details.

E. Estimation of C, D and the nonlinear function g

From $\hat{f}$, an estimate $\widehat{\mathcal{U}}_f$ for $\mathcal{U}_f$ can be obtained. With $\widehat{\mathcal{U}}_f$ estimates for the system matrices C and D and the nonlinearity g^{-1} are obtained from:

$$(\widehat{C},\widehat{D},\hat{g}^{-1}) = \underset{C,D,g^{-1}}{\arg\min} \left\| \mathcal{Y}_g - \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} \widehat{X}_i \\ \widehat{\mathcal{U}}_f \end{bmatrix} \right\|_F^2 , \quad (13)$$

with

$$\mathcal{Y}_g \triangleq \begin{bmatrix} g^{-1}(y_i) & g^{-1}(y_{i+1}) & \ldots & g^{-1}(y_{i+j-2}) \end{bmatrix}.$$

Again this problem can readily be solved for C, D and g^{-1} using the concept of componentwise LS-SVM regression (see the companion paper [10]).

IV. Illustrative examples

A. A SISO system

Consider the following artificial linear system which belongs to the class of Hammerstein-Wiener models:

$$y = g\left(\frac{B(z)}{A(z)} f(u)\right), \quad (14)$$

with A and B polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$ and $A(z) = (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$, the input- and output-nonlinearities are given by $f : \mathbb{R} \to \mathbb{R} : f(u) = \mathrm{sinc}(u)$ and

$$g : \mathbb{R} \to \mathbb{R} : g(y) = \begin{cases} y/12, & y \leq 0, \\ \mathrm{tanh(y/4)}, & \mathrm{y > 0}. \end{cases} \quad (15)$$

Two datasets were generated from this system with the inputs $u_k \sim \mathcal{N}(0,2)$ white Gaussian noise sequences for $t = 0,\ldots,N-1$ with $N = 500$. Although the intersection algorithm is in principal only designed for deterministic systems, 5% of zero mean white Gaussian noise was added to the outputs in both datasets to illustrate the relative robustness of the proposed technique to moderate amounts of noise. The first dataset obtained using the procedure described above was used to train the model, the second one was used to tune the model. Only the less critical number of block-rows in the Hankel matrices was fixed beforehand at 10, a common choice in subspace algorithms.

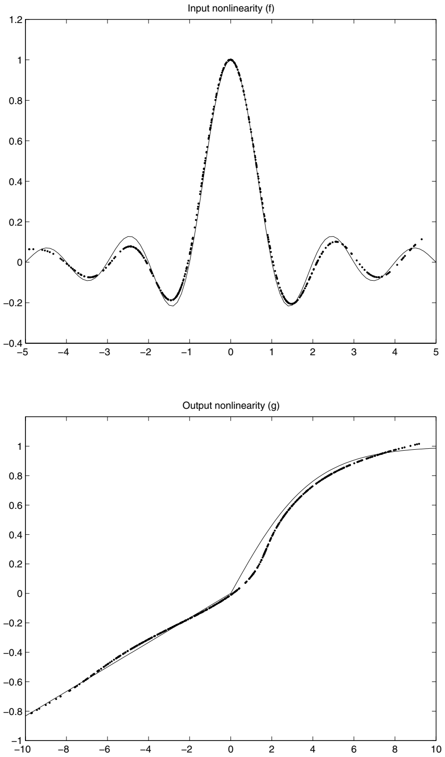

Fig. 1. Estimated input nonlinearity f and output nonlinearity g evaluated on the validation inputs and outputs (dots) compared with the true nonlinearities (solid line) for the SISO example described in IV.

For K^u and K^y, RBF kernels were chosen with $\sigma_u = 1$ and $\sigma_y = 0.5$ respectively. The hyper-parameter γ was chosen as $\gamma = 1$. The obtained nonlinear functions $\hat{f}$ and $\hat{g}$ evaluated on the validation inputs and outputs, are compared with the true functions f and g in Figure 1. As can be seen in the figure, the obtained estimates are quite reliable. The obtained linear system is compared with the true system in Figure 2.

B. A MIMO system

To illustrate the freedom that one gets by plugin of an appropriate kernel, in a second example, the proposed identification method was applied to a 2×2 purely deterministic MIMO Hammerstein system with a static input-nonlinearity involving saturation and a saddle point. The system is given as:

$$y = \begin{bmatrix} \frac{b_1(z)}{a_1(z)} & \frac{b_2(z)}{a_1(z)} \\ \frac{b_1(z)}{a_2(z)} & \frac{b_2(z)}{a_2(z)} \end{bmatrix} f(u) + \begin{bmatrix} \frac{1}{a_1(z)} \\ \frac{1}{a_2(z)} \end{bmatrix} e \quad (16)$$

with

$$\begin{aligned} a_1(z) &= (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i}), \\ a_2(z) &= (z - 0.97e^{\pm 0.7i})(z - 0.98e^{\pm 1.4i})(z - 0.97e^{\pm 2.3i}), \\ b_1(z) &= z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3, \\ b_2(z) &= z^6 + 0.9z^5 + 0.7z^4 + 0.2z^3, \\ f(u) &= \begin{bmatrix} -\arctan(u(1))\arctan(u(2)) \\ \arctan(u(1)) - \arctan(u(2)) \end{bmatrix}. \end{aligned}$$

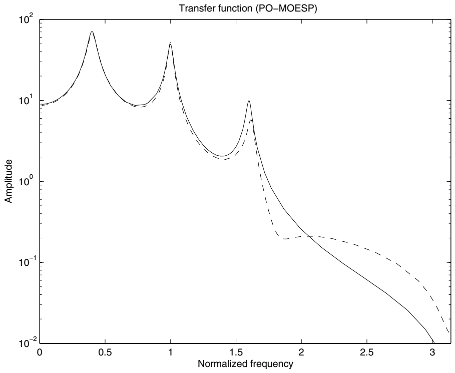

Fig. 2. Estimated transfer functions (dashed) for the SISO example described in IV using a PO-MOESP after estimation of the functions f and g. The true transfer function is displayed in solid.

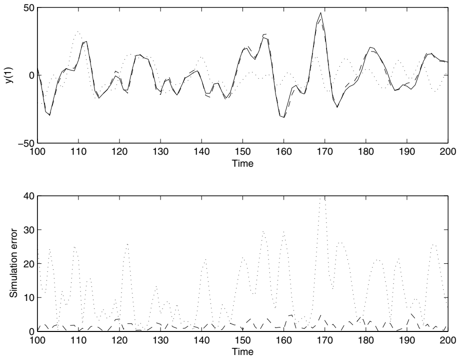

Fig. 3. Simulation on an independent test-set of the first output $y(1)$ of a twelfth order MIMO Hammerstein model described in IV-B using an LS-SVM Hammerstein-Wiener estimator (dashed line) and a linear PO-MOESP subspace estimator (dotted line). The true output is depicted with a solid line. All simulations are initialized with $x_0 = 0$. The error between the estimated and the true output is also shown in the Figure.

A two-component zero mean white Gaussian input sequence u with length 500 and standard deviation 1 was generated and fed into the system (16). Based on u and the obtained output y, estimates for the linear system and f are obtained using the Hammerstein-Wiener identification algorithm proposed in this paper, whereby an RBF kernel was chosen for K^u and a linear kernel for K^y to effectively limiting the Hammerstein-Wiener algorithm to the identification of Hammerstein systems.

As in the SISO example, the number of block-rows in the Hankel matrices was chosen equal to 10. The hyper-parameters were again obtained by evaluation on a validation set and chosen as $\sigma_u = 1$, $\gamma = 0.1$ and $\gamma_u = \gamma_y = 1$. The order was easily found to be 12 from an inspection of the canonical correlations in the kernel CCA step. As an indication of the performance, the results of a simulation on an independent test-set using the obtained model are shown in Figure 3 for the first component of the output. Also available in the figure is the result of a classical linear PO-MOESP subspace estimator which is clearly inferior to that obtained using the Hammerstein-Wiener approach.

V. Conclusions

In this paper, a method for the identification of Hammerstein-Wiener systems was presented based on the theory of kernel canonical correlation analysis and Least Squares Support Vector Machines. The proposed algorithm is applicable to SISO and MIMO systems and does not impose restrictive assumptions on the input sequence in contrast to most existing Hammerstein-Wiener approaches. Furthermore, the algorithm was seen to work well on a set of examples.

Acknowledgements. Ivan Goethals is a research assistant with the FWO flanders, Kristiaan Pelckmans is a researh assistent with the KULeuven, Luc Hoegaerts is a reserch assistent with the IWT flanders, Johan Suykens is an assistent professor with the KULeuven, Bart De Moor is a full professor with the KULeuven. Our research is supported by: Research Council KU Leuven: GOA-Mefisto 666, ID, several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientific Research Flanders (several PhD/postdoc grants, projects G.0407.02, G.0256.97, G.0115.01, G.0240.99, G.0197.02, G.0499.04, G.0211.05, G.0080.01, research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s, STWW-Genprom), GBOU-McKnow, Eureka-Impact, Eureka-FLiTE, several PhD grants); Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-10-29 (2002-2006) (2002-2006): Dynamical Systems and Control: Computation, Identification & Modelling), Program Sustainable Development PODO-II; Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 686:337–404, 1950.

[2] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[3] E.W. Bai. An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica*, 4(3):333–338, 1998.

[4] E.W. Bai. A blind approach to the Hammerstein-Wiener model identification. *Automatica*, 38:967–979, 2002.

[5] F.H.I. Chang and R. Luus. A noniterative method for identification using the Hammerstein model. *IEEE Transactions on Automatic Control*, 16:464–468, 1971.

[6] P. Crama. *Identification of block-oriented nonlinear models*. PhD thesis, Vrije Universiteit Brussel, Dept. ELEC, June 2004.

[7] P. Crama and J. Schoukens. Hammerstein-Wiener system estimator initialization. In *Proc. of the International Conference on Noise and Vibration Engineering (ISMA2002), Leuven*, pages 1169–1176, 16-18 September 2002.

[8] B. De Moor. *Mathematical concepts and techniques for modeling of static and dynamic systems*. PhD thesis, Katholieke universiteit Leuven, K.U.Leuven (Leuven, Belgium), 1988.

[9] A.H. Falkner. Iterative technique in the identification of a non-linear system. *International Journal of Control*, 1:385–396, 48.

[10] I. Goethals, L. Hoegaerts, J.A.K. Suykens, V. Verdult, and B. De Moor. Hammerstein-Wiener subspace identification using kernel Canonical Correlation Analysis. Technical Report 05-30, ESAT-SISTA, K.U.Leuven (Leuven Belgium), 2005 *available online at* ftp.esat.kuleuven.ac.be/pub/SISTA/goethals/goethals_hammer_wiener.ps .

[11] I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Subspace identification of Hammerstein systems using least squares support vector machines. Technical Report 04-114, ESAT-SISTA, K.U.Leuven (Leuven Belgium), *Submitted for publication*, 2004.

[12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1989.

[13] P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.

[14] W.E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 26th Conference on Decision and Control (CDC90), Hawaii, US*, pages 594–604, 1990.

[15] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state-space models. *International Journal of Control*, 49:219–232, Jan. 1989.

[16] K. Pelckmans, I. Goethals, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Componentwise least squares support vector machines. *Support Vector Machines: Theory and Applications,* L. Wang (Ed.), Springer, 2005, *In press*.

[17] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[18] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[19] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.

[20] V.N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.

[21] Y. Zhu. Estimation of an N-L-N Hammerstein-Wiener model. *Automatica*, 38:1607–1614, 2002.

APPENDIX

A. LS-SVM function regression

Let $\{x_i, y_i\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ be a set of input/output training data with input x_i and output y_i. Consider the regression model $y_i = f(x_i) + e_i$ where $x_1, \ldots, x_N$ are deterministic points, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an unknown real-valued smooth function and $e_1, \ldots, e_N$ are uncorrelated random errors with $E\left[e_i\right] = 0$, $E\left[e_i^2\right] = \sigma_e^2 < \infty$. In recent years, Support Vector Machines (SVMs) have been used for the purpose of estimating the nonlinear f. The following model is assumed:

$$f(x) = w^T\varphi(x) + b,$$

where $\varphi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_H}$ denotes a potentially infinite ($n_H = \infty$) dimensional feature map. The regularized cost function of the Least Squares SVM (LS-SVM) is given as

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,e) &= \frac{1}{2}w^Tw + \frac{\gamma}{2}\sum_{i=1}^N e_i^2, \\ \text{subject to} : y_i &= w^T\varphi(x_i) + b + e_i, \;\; i = 1, \ldots, N. \end{aligned}$$

The relative importance between the smoothness of the solution and the data fitting is governed by the scalar $\gamma \in \mathbb{R}_0^+$ referred to as the regularization constant. The optimization performed corresponds to ridge regression [12] in feature space. In order to solve the constrained optimization problem, the Lagrangian $\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,e) - \sum_{i=1}^N \alpha_i\{w^T\varphi(x_i) + b + e_i - y_i\}$ is constructed, with α_i the Lagrange multipliers. After applocation of the conditions for optimality: $\frac{\partial\mathcal{L}}{\partial w} = 0$, $\frac{\partial\mathcal{L}}{\partial b} = 0$, $\frac{\partial\mathcal{L}}{\partial e_i} = 0$, $\frac{\partial\mathcal{L}}{\partial \alpha_i} = 0$, the following set of linear equations is obtained:

$$\left[\begin{array}{c|c} 0 & 1_N{}^T \\ \hline 1_N & \Omega + \gamma^{-1}I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \qquad (17)$$

where $y = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}^T$, $1_N = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1 & \ldots & \alpha_N \end{bmatrix}^T$, $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$, $\forall i,j = 1, \ldots, N$, with K the positive definite kernel function. Note that in order to solve the set of equations (17), the feature map φ does never have to be defined explicitly. Only its inner product, a positive definite kernel, is needed. This is called the kernel trick [20], [17]. For the choice of the kernel $K(\cdot,\cdot)$, see e.g. [17]. Typical examples are the use of a polynomial kernel $K(x_i, x_j) = (\tau + x_i^Tx_j)^d$ of degree d or the RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ where σ denotes the bandwidth of the kernel. The resulting LS-SVM model for function estimation can be evaluated at a new point x_* as

$$\hat{f}(x_*) = \sum_{i=1}^N \alpha_i K(x_*, x_i) + b,$$

where (b, α) is the solution to (17).

B. Regularized KCCA

With the notations of III-C, the regularized version of KCCA as derived in [18] follows by defining $\mu_p = (1/j)\sum_{s=1}^j \Phi_p(:,s)$ and $\mu_f = (1/j)\sum_{s=1}^j \Phi_f(:,s)$ as the expected centers. and solving the following minimax problem:

$$\begin{cases} \max \sum_{s=1}^j e_s r_s, & \text{(Maximize correlations)} \\ \min v_p^T v_p + v_f^T v_f, & \text{(Minimize norms)} \\ \min \sum_{s=1}^j (e_s^2 + r_s^2), & \text{(Regularization)} \end{cases}$$

subject to $e_s = v_p{}^T(\Phi_p(:,s) - \mu_p)$ and $r_s = v_f{}^T(\Phi_f(:,s) - \mu_f)$ for $s = 1, \ldots, j$. This leads to the following primal cost-function:

$$\max_{v_p, v_f} \sum_{s=1}^j \left[\frac{1}{\lambda}e_s r_s - \frac{1}{2\gamma}e_s^2 - \frac{1}{2\gamma}r_s^2\right] - \frac{1}{2}v_p{}^Tv_p - \frac{1}{2}v_f{}^Tv_f$$

with hyperparameters $\lambda, \gamma \in \mathbb{R}_0^+$. Introducing α_s, β_s as Lagrange multiplier parameters, the Lagrangian is written as

$$\begin{aligned} L(v_p, v_f, e, r; a, b) = \sum_{s=1}^j \left[\frac{1}{\lambda}e_s r_s - \frac{1}{2\gamma}e_s^2 - \frac{1}{2\gamma}r_s^2\right] \\ - \frac{1}{2}v_p{}^Tv_p - \frac{1}{2}v_f{}^Tv_f - \sum_{s=1}^j \alpha_s(e_s - v_p{}^T(\Phi_p(:,s) - \mu_p)) \\ - \sum_{s=1}^j \beta_s(r_s - v_f{}^T(\Phi_f(:,s) - \mu_f)). \end{aligned}$$

After application of the conditions for optimality: $\frac{\partial L}{\partial v} = 0$, $\frac{\partial L}{\partial w} = 0$, $\frac{\partial L}{\partial e_s} = 0$, $\frac{\partial L}{\partial r_s} = 0$ $\frac{\partial L}{\partial \alpha_s} = 0$, $\frac{\partial L}{\partial \beta_s} = 0$, and elimination of the primal variables w_p, w_f, e and r, the dual problem that is finally obtained is given by the following regularized system of equations

$$\begin{aligned} K_f^c\; \mathcal{V}_f &= \left(K_p^c + \frac{1}{\gamma}I_j\right)\; \mathcal{V}_p\; \Lambda, \\ K_p^c\; \mathcal{V}_p &= \left(K_f^c + \frac{1}{\gamma}I_j\right)\; \mathcal{V}_f\; \Lambda, \end{aligned}$$

which is the standard form of the regularized kernel CCA algorithm as presented in [18].