Subspace Identification of Hammerstein-Wiener systems using Kernel Canonical Correlation Analysis

Ivan Goethals, Luc Hoegaerts, Vincent Verdult,

Johan A.K. Suykens, Bart De Moor

K.U. Leuven, ESAT-SCD-SISTA

Kasteelpark Arenberg 10

B-3001 Leuven (Heverlee), Belgium

Tel: 32/16/32 86 58 - Fax: 32/16/32 19 70

Email: ivan.goethals@esat.kuleuven.ac.be

March 2004

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, instead of the more classical CCA approach. The linear model and static nonlinearities on input and output are readily obtained once the state is known using Least Squares Support Vector Machines (LS-SVM) -regression.

Keywords: Subspace identification, Hammerstein-Wiener models, Least Squares Support Vector Machines.

1 Introduction

Subspace identification algorithms are widely known and appreciated for their quick and reliable estimation of linear models based on available input/output measurements. Much of the reliability of subspace identification algorithms is attributed to the fact that a model is obtained, solely by using numerically reliable matrix and vector manipulations such as projections and singular value decompositions. This in contrast to the more classical predictor error methods which often involve minimization of a non-convex cost-function, with no guarantee that the obtained local minimum yields a good model.

Although subspace identification algorithms are fast and robust, a major drawback is that their use is mostly restricted to the class of linear systems. Some attempts to extend the use of subspace identification algorithms to nonlinear systems have been made in the past for general nonlinear systems [16, 27], or more restricted model structures such as bilinear models [11, 17, 28], Wiener models [30], and Hammerstein models [31, 3].

In this paper, we will consider the extension of the classical subspace intersection algorithm [18] to Hammerstein-Wiener systems in state-space form:

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

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 mappings 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 nonlinear 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 Least Squares - Support Vector Machines (LS-SVMs) regression problem, which was earlier used in an extension of the N4SID [13] identification algorithm towards Hammerstein systems.

To our knowledge, few results have been reported on the identification of Hammerstein-

Wiener models. In [3], a scheme for the identification of SISO Hammerstein-Wiener systems is developed based on the idea of overparametrization [6]. 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 [8, 7] but imposes certain restrictions on the inputs and is iterative in nature. Other contributions such as [10, 32] 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 such as white noise or periodicity, 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.

The outline of this paper is as follows: In Section 2 the basic ingredients of the subspace intersection algorithm for linear systems are reviewed briefly. Section 3 extends the linear intersection algorithm towards a nonlinear setting using a variation on the theme of LS-SVMs and kernel CCA. Section 4, finally, presents an illustrative example. A brief overview of LS-SVM regression techniques and regularized kernel CCA is provided in Appendices A and B for readers that are unfamiliar with the kernel framework.

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 $A(:,i)$ denotes 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.

2 The subspace intersection algorithm

The subspace algorithm considered in this paper was originally proposed in [9, 18] and is largely based on the idea that the state of a linear or nonlinear model can be considered as the minimal intersection between past and future measurement data [16].

The subspace intersection algorithm identifies models of the form

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

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 $\{u_t, y_t\}, t = 0, \ldots, N-1$, 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

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \ldots & u_{j-1} \\ u_1 & u_2 & u_3 & \ldots & u_j \\ \vdots & \vdots & \vdots & & \vdots \\ u_{i-1} & u_i & u_{i+1} & \ldots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \ldots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \ldots & u_{i+j} \\ \vdots & \vdots & \vdots & & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \ldots & u_{2i+j-2} \end{array}\right] \triangleq \left[\begin{array}{c} U_p \\ \hline U_f \end{array}\right] \in \mathbb{R}^{(2i)m\times j},$$

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

$$W_p \triangleq \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad W_f \triangleq \begin{bmatrix} U_f \\ Y_f \end{bmatrix}.$$

Finally,

$$\begin{aligned} X_p &\triangleq \begin{bmatrix} x_0 & x_1 & \dots & x_{j-1} \end{bmatrix}, \\ X_f &\triangleq \begin{bmatrix} x_i & x_{i+1} & \dots & x_{i+j-1} \end{bmatrix}, \end{aligned}$$

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:

$$\mathrm{row}(\mathrm{X_f}) = \mathrm{row}(\mathrm{W_p}) \cap \mathrm{row}(\mathrm{W_f}).$$

Hence, the order of the system and a realization of 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 correlations 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 [9, 18], 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:

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

It was proven in [18] that when both the inputs and outputs are corrupted by additive spatially and temporary white noise sequences of equal covariance, a consistent estimate $\widehat{X}_f$ for the state sequence X_f is still obtained when using the algorithm described above. When this assumption is violated, it is possible to alter the algorithm by introducing weights based on the knowledge of the noise correlation [19]. For instructive purposes, we will however only consider the unweighted version in this paper.

3 Hammerstein-Wiener subspace intersection

3.1 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:

$$\begin{cases} x_{t+1} = Ax_t + Bf(u_t), \\ g^{-1}(y_t) = Cx_t + Df(u_t). \end{cases} \tag{4}$$

As mentioned in Section 2, 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 regression (Subsections 3.4 and 3.5).

3.2 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 [15, 2]. In kernel methods [23] 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

$\varphi^u(u_\cdot)$ and $\varphi^y(y_\cdot)$, one constructs a feature matrix

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

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 reproducing 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 [26, 20].

3.3 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 [15, 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:

$$\begin{aligned} K_p K_f \; \mathcal{V}_f &= K_p K_p \; \mathcal{V}_p \; \Lambda, \\ K_f K_p \; \mathcal{V}_p &= K_f K_f \; \mathcal{V}_f \; \Lambda. \end{aligned} \tag{8}$$

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 [15, 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 [23] is formulated using a support vector machine approach [26] 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 [23] and Appendix B), we state here the final 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 so-called mean 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 $K_f^c = M_c K_f M_c$ with $M_c = (I_j - (1/j) 1 1_j^T)$ [22].

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 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:

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

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

3.4 Estimation of A, B and the nonlinear function f

Once a state estimate has been obtained, estimates for the system matrices A and B and the nonlinear function f can be calculated 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

$$\begin{aligned} \widehat{X}_i &\triangleq \widehat{X}_f(:, 1:j-1), \\ \widehat{X}_{i+1} &\triangleq \widehat{X}_f(:, 2:j), \\ \mathcal{U}_f &\triangleq \begin{bmatrix} f(u_i) & f(u_{i+1}) & \dots & f(u_{i+j-2}) \end{bmatrix}, \end{aligned} \tag{12}$$

and $\widehat{X}_f$ estimated from (10). 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

$$Bf = \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \varphi^u, \tag{13}$$

with φ^u the feature-map introduced in Section 3. 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

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

The resulting LS-SVM primal problem is as follows:

$$\begin{aligned} \min_{w,E,A} \mathcal{J}(w, E) &= \tfrac{1}{2} \textstyle\sum_{s=1}^n w_{f,s}^T w_{f,s} + \tfrac{\gamma_u}{2} \sum_{s=1}^n \sum_{t=1}^{j-1} E(s,t)^2, \\ \text{subject to} \quad & \widehat{X}_{i+1}(s,t) = A\widehat{X}_i(:,t) + w_{f,s}^T \mathcal{U}_\varphi(:,t) + E(s,t), \\ & \forall s = 1, \dots, n, \ t = 1, \dots, j-1, \end{aligned} \tag{14}$$

Lemma 3.1. Primal-dual derivation: *Given the least squares problem (11) and related primal problem (14), LS-SVM estimates for B and the transformed inputs $\mathcal{U}_f$ can be*

obtained from a rank m approximation of $\mathcal{A}^T\mathcal{K}^u$. $\mathcal{A}$ *and* A *are obtained from the following set of linear equations:*

$$\left[\begin{array}{c|c} 0 & \widehat{X}_i \\ \hline \widehat{X}_i^T & \mathcal{K}^u + \gamma_u^{-1} I_{j-1} \end{array}\right] \left[\begin{array}{c} A^T \\ \hline \mathcal{A} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \widehat{X}_{i+1}^T \end{array}\right], \tag{15}$$

with

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \dots & \alpha_{n,1} \\ \alpha_{1,2} & \alpha_{2,2} & \dots & \alpha_{n,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j-1} & \alpha_{2,j-1} & \dots & \alpha_{n,j-1} \end{bmatrix}, \\ w_{f,s} &= \textstyle\sum_{t=1}^{j-1} \alpha_{s,t} \varphi^u(u_{i+t-1}),\ s = 1, \dots, n, \\ \mathcal{K}^u(p,q) &= K^u(u_{i+p-1}, u_{i+q-1}),\ p, q = 1, \dots, j-1. \end{aligned}$$

Proof. This directly follows from the Lagrangian:

$$\begin{aligned} \mathcal{L}(w, A, E; \alpha) &= \mathcal{J}(w, E) \\ &- \sum_{s=1}^{n} \sum_{t=1}^{j-1} \alpha_{s,t} \left(\widehat{X}_{i+1}(s,t) - A\widehat{X}_i(:,t) - w_{f,s}^T \mathcal{U}_\varphi(:,t) - E(s,t) \right), \end{aligned}$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w_{f,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial A} = 0$, $\frac{\partial \mathcal{L}}{\partial E(s,t)} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$ and by observing that:

$$B\mathcal{U}_f = \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \mathcal{U}_\varphi = \mathcal{A}^T \mathcal{K}^u. \tag{16}$$

□

Note that the estimates obtained in Lemma 3.1 will in general not be uniquely defined, especially if $n \leq m$. This is an intrinsic property of Hammerstein-Wiener models and the choice of the actual representation is left to the user. From $\mathcal{U}_f$ and the inputs $u_t,\ t = i, \dots, i+j-2$, obtaining an estimate for f is a straightforward matter.

3.5 Estimation of C, D and the nonlinear function g

Once an estimate $\widehat{\mathcal{U}}_f$ for $\mathcal{U}_f$ has been found, estimates for the system matrices C and D and the nonlinearity g^{-1} are obtained from:

$$(\widehat{C}, \widehat{D}, \hat{g}^{-1}) = \underset{C,D,g^{-1}}{\arg\min} \left\| \mathcal{Y}_g - \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} \widehat{X}_i \\ \widehat{\mathcal{U}}_f \end{bmatrix} \right\|_F^2, \tag{17}$$

with

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

Note that C, D and g^{-1} in (17) are only defined up to a constant scaling factor. To avoid conditioning problems, we therefore fix $C(:,1) = 1_l$, which is a feasible assumption for SISO systems given that the first component of the state is generally observable in subspace models [24, 25]. For MIMO systems, it is possible that certain outputs are unexcited by the first state component, in which case a more complicated constraint (such as $\sum_{k=1}^{n} C(:,k) = 1_l$) might be more appropriate. Although such constraints can easily be incorporated into the LS-SVM framework [23, 5], we will further assume that $C(:,1) = 1_l$ for simplicity. Derivations involving other constraints can easily be derived along the lines of the calculations below. With $C(:,1) = 1_l$, the resulting LS-SVM primal problem is as follows:

$$\begin{aligned} \min_{w,E,C,D} \mathcal{J}(w,E) &= \tfrac{1}{2}\sum_{s=1}^{l} w_{g,s}^T w_{g,s} + \tfrac{\gamma_y}{2}\sum_{s=1}^{n}\sum_{t=1}^{j-1} E(s,t)^2, \\ \text{subject to} \quad & X_i(1,t) = w_{g,s}^T \mathcal{Y}_\varphi(:,t) - C(s,2:n)X_i(2:n,t) - D(s,:)\mathcal{U}_f(:,t) \\ & \qquad -E(s,t), \ \forall s = 1,\dots,l, \ t = 1,\dots,j-1. \end{aligned} \tag{18}$$

with

$$\mathcal{Y}_\varphi = \begin{bmatrix} \varphi^y(y_i) & \varphi^y(y_{i+1}) & \dots & \varphi^y(y_{i+j-2}) \end{bmatrix},$$

whereby φ^y is as in Section 3.

Lemma 3.2. Primal-dual derivation: *Given the least squares problem (17), and the related primal problem (18), LS-SVM estimates for the transformed outputs $\mathcal{Y}_g$ are obtained as $\mathcal{A}^T\mathcal{K}^y$. $\mathcal{A}$, C and D are obtained from the following set of linear equations:*

$$\left[\begin{array}{c|c|c} 0 & 0 & X_i(2:n,:) \\ \hline 0 & 0 & \widehat{\mathcal{U}}_f \\ \hline X_i(2:n,:)^T & \widehat{\mathcal{U}}_f^T & \mathcal{K}^y + \gamma_y^{-1} I_{j-1} \end{array}\right] \left[\begin{array}{c} -C(:,2:n)^T \\ \hline -D^T \\ \hline \mathcal{A} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline 1_l^T \otimes X_i(1,:)^T \end{array}\right], \qquad (19)$$

whereby

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \dots & \alpha_{l,1} \\ \alpha_{1,2} & \alpha_{2,2} & \dots & \alpha_{l,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j-1} & \alpha_{2,j-1} & \dots & \alpha_{l,j-1} \end{bmatrix}, \\ w_{g,s} &= \textstyle\sum_{t=1}^{j-1} \alpha_{s,t} \varphi^y(y_{i+t-1}),\ s = 1, \dots, l, \\ \mathcal{K}^y(p,q) &= K^y(y_{i+p-1}, y_{i+q-1}),\ p,q = 1, \dots, j-1. \end{aligned}$$

Proof. This directly follows from the Lagrangian:

$$\begin{aligned} \mathcal{L}(w, E, C, D; \alpha) &= \mathcal{J}(w, E) - \sum_{s=1}^{l} \sum_{t=1}^{j-1} \alpha_{s,t} \left(w_{g,s}^T \mathcal{Y}_\varphi(:,t) - X_i(1,t) \right. \\ &- \left. C(s, 2:n) X_i(2:n,t) - D(s,:) \mathcal{U}_f(:,t) - E(s,t) \right). \end{aligned}$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w_{g,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial C} = 0$, $\frac{\partial \mathcal{L}}{\partial D} = 0$, $\frac{\partial \mathcal{L}}{\partial E(s,t)} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$ and by observing that:

$$\mathcal{Y}_g = \begin{bmatrix} w_{g,1}^T \\ w_{g,2}^T \\ \vdots \\ w_{g,n}^T \end{bmatrix} \mathcal{Y}_\varphi = \mathcal{A}^T \mathcal{K}^y. \qquad (20)$$

□

Again, from $\mathcal{Y}_g$ in Lemma 3.2 and the outputs y_t, $t = i, \dots, i+j-2$, obtaining an estimate for g is a trivial matter.

3.6 Practical implementation

Following the discussion in the former sections, the final algorithm for the estimation of Hammerstein-Wiener systems can be summarized as follows:

1. Find estimates for the state sequences $\widehat{X}_i$ and $\widehat{X}_{i+1}$ from (10) and (12).

2. Obtain estimates for A, B and f following the procedure outlined in Subsection 3.4.

3. Obtain estimates for C, D and g following the procedure outlined in Subsection 3.5.

Some practical issues remain, regarding the tuning of the hyper-parameters, the need for persistently exciting inputs, and the behavior of the presented algorithm under the presence of process and/or measurement-noise.

3.6.1 Tuning of the hyper-parameters

Many tunable parameters, the so-called hyper-parameters, are present in the proposed algorithm such as the system order n, the number of block rows i in the block Hankel matrices, the regularization parameters γ, γ_u and γ_y in the KCCA- and LS-SVM estimation steps, and other potential kernel parameters such as the bandwidths σ_u and/or σ_y when RBF-kernels are used. In principle, these parameters could be tuned by validating the performance of the obtained Hammerstein-Wiener model on an independent validation dataset. However, as this would constitute a highly non-convex high-dimensional search, the resulting identification algorithm would computationally be too expensive for most modern computers.

Fortunately, it can be shown that the tuning problem can be split up in several sub-problems. From the resulting $\mathcal{V}_p$ and $\mathcal{V}_f$ from the KCCA step in 3, for instance, a validation state based on past data can be calculated as follows:

$$\widehat{X}_f^{\text{val}} = \widehat{\mathcal{V}}_p(1:n,:)^T K_p^{\text{val}},$$

with

$$K_p^{\text{val}} = (\Phi_p - 1_j^T \otimes \mu_p)^T (\Phi_p^{\text{val}} - 1_j^T \otimes \mu_p),$$

and the superscript $\cdot^{\text{val}}$ denoting that inputs and outputs from the validation dataset, rather than the training-dataset are considered. If the KCCA-step is well tuned, the following relation should hold:

$$\text{Row}\left(\widehat{X}_f^{\text{val}}\right) = \text{Row}\left(\widehat{\mathcal{V}}_f(1:n,:)^T K_f^{\text{val}}\right). \tag{21}$$

with

$$K_f^{\text{val}} = (\Phi_f - 1_j^T \otimes \mu_f)^T (\Phi_f^{\text{val}} - 1_j^T \otimes \mu_f),$$

the equivalent validation state based on future data. The extend to which relation (21) holds can easily be checked by calculating the largest canonical angle between both row-spaces. Hence, the hyper-parameters necessary for the KCCA step can be obtained without having to estimate the full Hammerstein-Wiener model. A similar reasoning can be used for the other steps in the proposed algorithm such as the estimation of A, B and f in Subsection 3.4.

3.6.2 Persistency of excitation

Given the fact that regularization is inherently present in the proposed identification algorithm, lack of persistency of excitation, a mathematical concept to denote that the linear system was sufficiently excited by the input to allow identification, will not lead to any numerical problems. However, to ensure that all aspects of the linear system are properly identified, persistency of excitation of the inputs $f(u)$ to the linear system of at least order $2im$ is necessary. For some nonlinear functions f, persistency of excitation of $f(u)$ can be guaranteed if u is persistently exciting (see [30] for a discussion on this issue).

3.6.3 Influence of process and measurement-noise

As noted in the introduction to the linear subspace intersection algorithm in Section 2, the intersection algorithm is in principal limited to noiseless systems. Hence, in the presence of process and/or measurement-noise, biased results are to be expected. Nevertheless, in practical cases, and if the amount of noise is moderate, the intersection algorithm is still known to perform adequately. To further illustrate this point, a modest amount of output-noise will be added to the simulation in Section 4.

3.6.4 Re-estimation of the linear model

Following the discussion on the behavior of the intersection algorithm in the presence of process and/or measurement noise, it is in principle possible to use the estimates for f and g^{-1} in the proposed intersection algorithm to generate the inputs $f(u)$ and outputs $g^{-1}(y)$ to the linear system. Based on these data-sequences a more noise-robust subspace identification such as the PO-MOESP algorithm [29] can be used in a second step to replace the original model obtained using the intersection algorithm. It will be shown in the example in Section 4 that such a second step can indeed improve on the accuracy of the obtained linear model, certainly in noisy conditions.

4 Illustrative example

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

$$y = g\left(\frac{B(z)}{A(z)}f(u)\right), \tag{22}$$

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 $f : \mathbb{R} \to \mathbb{R} : f(u) = \text{sinc}(u)$ and

$$g : \mathbb{R} \to \mathbb{R} : g(y) = \begin{cases} y/12, & y \leq 0, \\ \tanh(\text{y}/4), & \text{y} > 0. \end{cases} \tag{23}$$

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 principle only designed for deterministic systems, 5% of zero mean white Gaussian noise was added to the outputs in both datasets. The first dataset so obtained was used to train the model, the second one was used to tune the model. The number of block-rows in the Hankel matrices was fixed beforehand to 10, a common choice in subspace algorithms.

For K^u and K^y, RBF kernels were chosen with $\sigma_u = 1$ and $\sigma_y = 0.5$ respectively. These kernel bandwidth parameters, together with the hyper-parameter $\gamma = 1$ and the obtained $\mathcal{V}_p$ and $\mathcal{V}_f$ from the KCCA estimation step outlined in Subsection 3.3, were validated on the validation dataset according to the procedure explained in Subsection 3.6. In a second step, estimates for A ,B and the nonlinear function f_u were obtained using the procedure presented in Subsection 3.4 where $\gamma_u = 1$ was chosen after validation of the relation (15) on the validation dataset. Finally, estimates for C, D and the nonlinear function f_y were obtained following the procedure presented in 3.5 with $\gamma_y = 315$, again chosen by validation on the validation dataset.

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. Note that the first two resonances are nicely captured by the model, whereas the less energetic third resonance is not found. In itself not a bad result considering that the used system has already been shown [13] to be a difficult one to estimate with classical techniques such as ARX, and that the intersection algorithm is in essence a deterministic algorithm. Nevertheless, as mentioned in Subsection 3.6 a further improvement is possible by rerunning a linear subspace algorithm such as the PO-MOESP

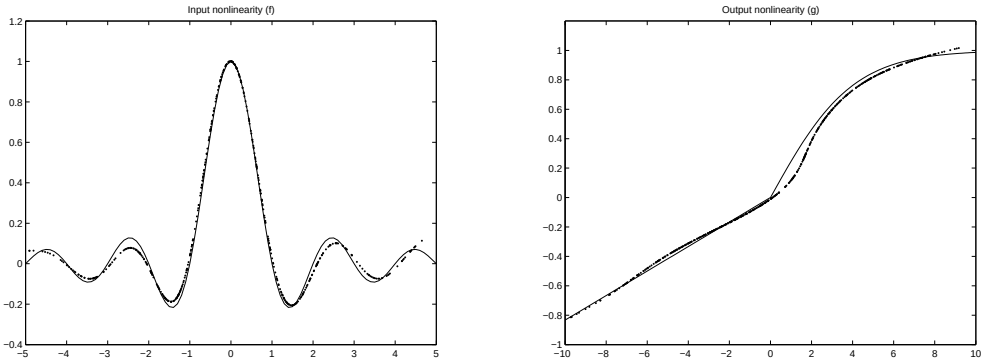


Figure 1: Estimated input nonlinearity f and output nonlinearity g evaluated on the validation inputs and outputs (dots) compared with the true nonlinearities (solid line) for the SISO example described in Section4.

[29] on the estimated inputs $\hat{f}(u)$ and outputs $\hat{g}^{-1}(y)$ of the linear system. That such a second step indeed improves the accuracy of the estimate of the linear model is shown in Figure 3.

As mentioned before, the algorithms proposed in this paper can also conveniently be applied to MIMO system. An example involving MIMO systems is discussed in [12].

5 Conclusions

In this paper, a method for the identification of Hammerstein-Wiener systems was presented making use of 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. The algorithm was seen to work well on a simulated example.

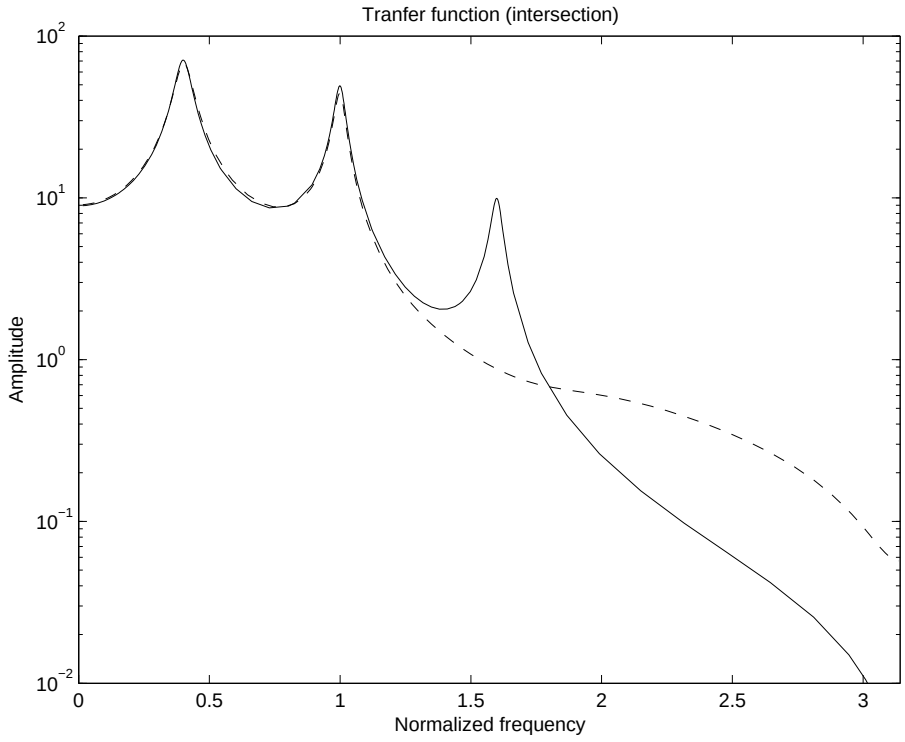


Figure 2: Estimated Transfer functions (dashed) for the SISO example described in Section 4 using the intersection-algorithm. The true Transfer function is displayed in solid.

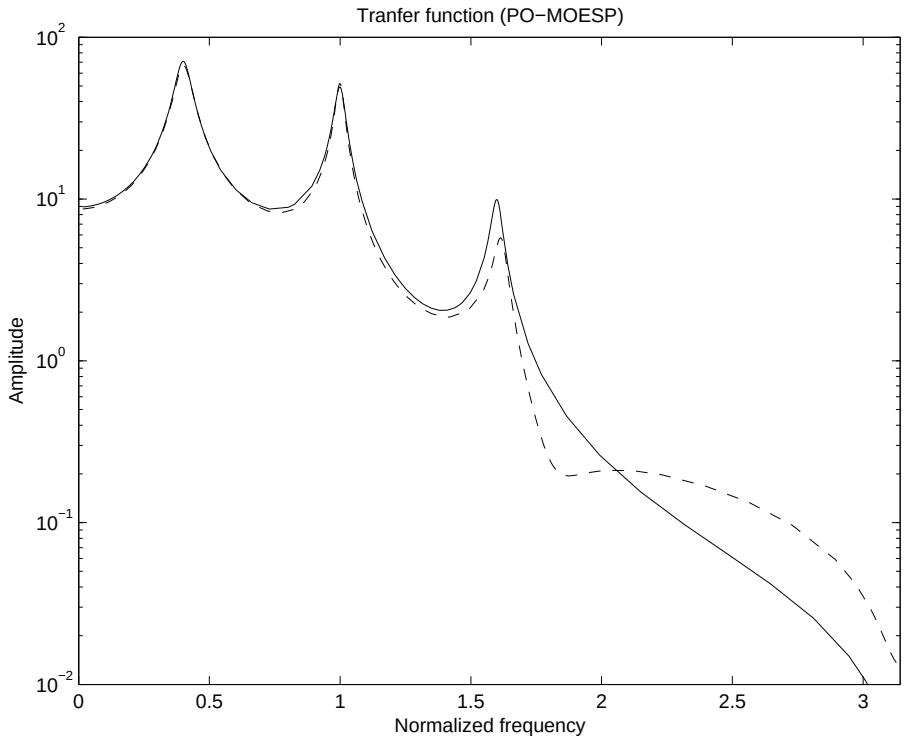


Figure 3: Estimated Transfer functions (dashed) for the SISO example described in 4 using a PO-MOESP after estimation of the functions f and g. The true Transfer function is displayed in solid.

Acknowledgments

Acknowledgments. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. It is supported by grants from several funding agencies and sources: Research Council KU Leuven: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientific Research Flanders (several PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0256.97 (subspace), G.0115.01 (bio-i and microarrays), G.0240.99 (multilinear algebra), G.0197.02 (power islands), G.0499.04 (robust statistics), G.0211.05 (nonlinear identification), G.0080.01 (collective behaviour), research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s (softsensors), STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), Eureka-Impact (MPC-control), Eureka-FLiTE (flutter modeling), 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 (CP/40: Sustainibility effects of Traffic Management Systems); Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS. JS is an associate professor and BDM is a full professor at K.U.Leuven Belgium, respectively.

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] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] 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.

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

[8] 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.

[9] 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.

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

[11] W. Favoreel, B. De Moor, and P. Van Overschee. Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, 44(6):1157–1165, 1999.

[12] I. Goethals, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B. De Moor. Subspace intersection identification of Hammerstein-Wiener systems. Technical Report 05-46, ESAT-SISTA, K.U.Leuven (Leuven Belgium), 2005 *Submitted for publication to joint CDC/ECC 2005, available online at* `ftp.esat.kuleuven.ac.be/pub/SISTA/goethals/goethals_hammer_wiener.pdf`.

[13] 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.

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

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

[16] 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.

[17] R. R. Mohler. *Nonlinear systems, Volume II: Applications to Bilinear Control*. Englewood-Cliffs, New Jersey: Prentice-Hall, 1991.

[18] 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.

[19] M. Moonen and J. Vandewalle. A QSVD approach to on- and off-line state space identification. *International Journal of Control*, 51(5):1133–1146, 1990.

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

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

[22] B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[23] 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.

[24] P. Van Overschee and B. De Moor. Choice of state-space basis in combined deterministic-stochastic subspace identification. *Automatica,* Special Issue on Trends in System Identification, 31(12):1877–1883, 1995.

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

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

[27] V. Verdult, J.A.K. Suykens, J. Boets, I. Goethals, and B. De Moor. Least squares support vector machines for kernel CCA in nonlinear state-space identification. In *Proceedings of the 16th international symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium*, July 2004.

[28] V. Verdult and M. Verhaegen. Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *International Journal of Control*, 74(18):1824–1836, 2001.

[29] M. Verhaegen and P. Dewilde. Subspace model identification, part II: analysis of the elementary output-error state space model identification algorithm. *International Journal of Control*, 56:1211–1241, 1992.

[30] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63:331–349, 1996.

[31] D. Westwick and M. Verhaegen. Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, 52(2):235–258, 1996.

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

A LS-SVM function regression

Let $\{x_t, y_t\}_{t=0}^{N-1} \subset \mathbb{R}^d \times \mathbb{R}$ be a set of input/output training data with input x_t and output y_t. Consider the regression model $y_t = f(x_t) + e_t$ where $x_0, \ldots, x_{N-1}$ are deterministic points, $f : \mathbb{R}^d \to \mathbb{R}$ is an unknown real-valued smooth function and $e_0, \ldots, e_{N-1}$ are uncorrelated random errors with $E[e_t] = 0$, $E[e_t^2] = \sigma_e^2 < \infty$. 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 $\varphi(x) : \mathbb{R}^d \to \mathbb{R}^{n_H}$ denotes a potentially infinite ($n_H = \infty$) dimensional feature map. The regularized cost function of the Least Squares SVM (LS-SVM) is given as

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,e) &= \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{t=1}^{n} e_t^2, \\ \text{subject to : } y_t &= w^T \varphi(x_t) + b + e_t, \;\; t = 0, \ldots, N-1. \end{aligned}$$

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 [14] in feature space. In order to solve the constrained optimization problem, the Lagrangian $\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{t=0}^{N-1} \alpha_t \{w^T \varphi(x_t) + b + e_t - y_t\}$ is constructed, with α_t the Lagrange multipliers. After application of the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w} = 0$, $\frac{\partial \mathcal{L}}{\partial b} = 0$, $\frac{\partial \mathcal{L}}{\partial e_t} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0$, the following set of linear equations is obtained:

$$\left[\begin{array}{c|c} 0 & 1_N{}^T \\ \hline 1_N & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{24}$$

where $y = \begin{bmatrix} y_0 & \dots & y_{N-1} \end{bmatrix}^T$, $1_N = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1 & \dots & \alpha_N \end{bmatrix}^T$, $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j),\ \forall i, j = 0, \dots, N-1$, with K the positive definite kernel function. Note that in order to solve the set of equations (24), 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 [26, 21]. For the choice of the kernel $K(\cdot, \cdot)$, see e.g. [21]. Typical examples are the use of a polynomial kernel $K(x_i, x_j) = (\tau + x_i^T x_j)^d$ 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

$$\hat{f}(x_*) = \sum_{t=0}^{N-1} \alpha_t K(x_*, x_t) + b,$$

where (b, α) is the solution to (24).

B Regularized KCCA

The regularized version of KCCA as derived in [23] 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 mean centers. and solving the following mini-max 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 $s = 1, \dots, j$. This leads to the following primal cost-function:

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

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

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

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 [23].