2011 8th International Multi-Conference on Systems, Signals & Devices

SUBSPACE IDENTIFICATION OF HAMMERSTEIN MODELS USING SUPPORT VECTOR MACHINES

Mujahed Al-Dhaifallah

Systems Engineering Department
KFUPM Box 5067, Dhahran 31261, Saudi Arabia
email: mujahed@kfupm.edu.sa

ABSTRACT

System identification is the art of finding mathematical tools and algorithms that build an appropriate mathematical model of a system from measured input and output data. Hammerstein model, consisting of a memoryless nonlinearity followed by a dynamic linear element, is often a good trade-off as it can represent some dynamic nonlinear systems very accurately, but is nonetheless quite simple. Moreover, the extensive knowledge about LTI system representations can be applied to the dynamic linear block. On the other hand, finding an effective representation for the nonlinearity is an active area of research. Recently, support vector machines (SVMs) and least squares support vector machines (LS-SVMs) have demonstrated powerful abilities in approximating linear and nonlinear functions. In contrast with other approximation methods, SVMs do not require a-priori structural information. Furthermore, there are well established methods with guaranteed convergence (ordinary least squares, quadratic programming) for fitting LS-SVMs and SVMs. The general objective of this research is to develop new subspace algorithms for Hammerstein systems based on SVM regression.

Index Terms— Subspace Identification, Hammerstein Models, Support Vector Machines

1. INTRODUCTION

For engineers and scientists, modeling is an important tool to understand, improve, and control system dynamics [1]. In physical modeling, analogy is used to build something that behaves almost like the original system. For example, an analog computer may be used to simulate the behavior of a system. However, to represent a complex engineering process, one must use different approach, an abstract mathematical model, since an analog computer model would be impractically large and complex [2]. Sometimes mathematical models can be obtained from first principles, for example: Newton's laws of motion, Kirchhoff's voltage and current laws, conservation of momentum, energy, or mass, etc. However, this modeling approach is unsuitable for complex systems [3]. System identification is the art of finding mathematical tools and algorithms that build an appropriate mathematical model of a system from measured input and output data. It has attracted the interest of scientists and engineers for many decades for many reasons, e.g. process engineers can use more accurate models of their plants to design improved controllers, aerospace engineers need accurate models to evaluate high performance aerospace vehicles, mechatronic engineers need models to design robots, physiologists use models to understand the functions of biological systems, etc. [4], [5]. In the last two decades, subspace identification theory [6], [7] has attracted researchers' interest because of its efficient way of identifying state-space models for high order, multiple input, multiple output, linear time-invariant systems. CVA (Canonical Variate Analysis, [8]), MOESP (Multivariable Output Error State space, [6]), and N4SID (Numerical Subspace State-Space System IDentification, [7]) are the most significant methods. The main theme in these methods is to find an estimate of the state variables or the extended observability matrix using the available record of input and output data.

All these algorithms are designed for linear models which may produce accurate predictions of a systems behavior, particularly, if it is restricted to operating within a narrow region. However, If the model is required to cover a broader operating region, then a nonlinear model may be required. One further step toward accuracy is to consider block structured models, cascades of static nonlinearities and dynamic linear systems. Common cascades include the Wiener (linear subsystem, L, followed by a nonlinearity, N), the Hammerstein (NL), the Wiener-Hammerstein (LNL), and the Hammerstein-Wiener (NLN) systems. These models represent natural extensions of the linear model. These nonlinear models will lead to a significant improvement compared to the linear approximation if the actual system fits within this class (e.g. there is an appropriately positioned nonlinearity) and if the nonlinearity has very fast dynamics (i.e. much faster than the linear part of the system) where it can be treated as if it were a static nonlinearity. Some of the subspace identification methods have been extended to some block structured nonlin-

Thanks to King Fahd University of Petroleum and Minerals for funding the research reported in this paper under the grant of JF100012.

ear models identifications. For Hammerstein systems, Verhaegen and Westwick [9] considered the extension of the MOESP family of subspace model identification schemes to the Hammerstein-type of nonlinear system where they assumed polynomial representation of the static nonlinearity. Goethals et al. [10] considered the extension of the N4SID family of subspace model identification schemes to the Hammerstein-type of nonlinear system. However, they used a least square support vector machine to represent the nonlinear part of the Hammerstein system. Recently, Bako et al. [11] extended Goethals work to time-varying systems by using LS-SVM to recursively estimate the nonlinear part of the system and ordinary least squares for recovering the linear part in state-space form. Wiener models have been considered less because the identification of a nonlinear map is more tractable when the measurement of the input to the map is available. Westwick and Verhaegen [12] extended the MOESP class of subspace model identification to identify Wiener systems. The main drawback of these methods is that the nonlinearity was parameterized with a polynomial because it is easy to fit by solving a linear regression problem. However, this regression problem could become ill-conditioned, especially with high order polynomials, resulting in bad coefficient estimates. Another limitation is the unreliable extrapolation of polynomials when used to approximate hard nonlinearities (deadzone, saturation, rectification) especially beyond the limits, and even near the limits of the training data [13]. Some of these limitations can be solved if the nonlinearity is approximated by spline function. However, spline functions are defined by a series of knot points which must either be chosen apriori, or treated as model parameters and included in the (non-convex) optimization. Neural networks are another tool to approximate nonlinear functions. Their powerful approximation abilities make them attractive. However, the need to specify the neural network topology in terms of the number of nodes and layers, and the need to solve non-convex optimization complicate its implementation. Recently, support vector machines (SVMs) and least squares support vector machines (LS-SVMs) have shown powerful abilities in approximating linear and nonlinear functions ([14], [15]). The general objective of this paper is to develop new subspace algorithm for Hammerstein systems based on SVM regression.

2. PROBLEM DEFINITION

The state space version of the Hammerstein model can be written as

$$\left\{ \begin{array}{l} x_{k+1} = Ax_k + Bf\left(u_k\right) + \nu_k \\ y_k = Cx_k + Df\left(u_k\right) + \upsilon_k \end{array} \right. \tag{1}$$

with $u_k \in \mathbb{R}$, $y_k \in \mathbb{R}$, $x_k \in \mathbb{R}^n$, $k \in \mathbb{Z}^+$ and $\{(u_k, y_k)\}$ is a set of input and output measurements. The process noise $\nu_k \in \mathbb{R}^n$ and the measurement noise $\upsilon_k \in \mathbb{R}$ are zero-mean white Gaussian noise vector sequences, statistically independent of the input u_k with covariance matrix:

$$\left\{ \begin{array}{ll} E\left\{ \begin{bmatrix} \nu_p \\ \upsilon_p \end{bmatrix} \begin{bmatrix} \nu_q^T & \upsilon_q^T \end{bmatrix} \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} & \text{if } p = q \\ \qquad\qquad = 0 & \text{if } p \neq q \end{array} \right. \tag{2}$$

Before proceeding, we need to define input and output block Hankel matrices which are

$$\begin{aligned} U_{0|2i-1} &\triangleq \begin{bmatrix} u_0 & u_1 & \cdots & u_{j-1} \\ u_1 & u_2 & \cdots & u_j \\ \vdots & \vdots & & \vdots \\ u_{i-1} & u_i & \cdots & u_{i+j-2} \\ u_i & u_{i+1} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \cdots & u_{i+j} \\ \vdots & \vdots & & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{2i+j-2} \end{bmatrix} \in \mathbb{R}^{(2i)m\times j} \\ &\triangleq \begin{bmatrix} U_{0|i-1} \\ U_{i|2i-1} \end{bmatrix} \triangleq \begin{bmatrix} U_p \\ U_f \end{bmatrix} \\ &\triangleq \begin{bmatrix} U_{0|i} \\ U_{i+1|2i-1} \end{bmatrix} \triangleq \begin{bmatrix} U_p^+ \\ U_f^- \end{bmatrix} \end{aligned}$$

$$\begin{aligned} Y_{0|2i-1} &\triangleq \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_{j-1} \\ y_1 & y_2 & y_3 & \cdots & y_j \\ \vdots & \vdots & \vdots & & \vdots \\ y_{i-1} & y_i & y_{i+1} & \cdots & y_{i+j-2} \\ y_i & y_{i+1} & y_{i+2} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & y_{i+3} & \cdots & y_{i+j} \\ \vdots & \vdots & \vdots & & \vdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \cdots & y_{2i+j-2} \end{bmatrix} \in \mathbb{R}^{(2i)l\times j} \\ &\triangleq \begin{bmatrix} Y_{0|i-1} \\ Y_{i|2i-1} \end{bmatrix} \triangleq \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} \\ &\triangleq \begin{bmatrix} Y_{0|i} \\ Y_{i+1|2i-1} \end{bmatrix} \triangleq \begin{bmatrix} Y_p^+ \\ Y_f^- \end{bmatrix} \end{aligned}$$

with i and j user defined indices such that $2i + j - 1 = N$.

3. THE N4SID ALGORITHM FOR SUBSPACE IDENTIFICATION OF HAMMERSTEIN SYSTEMS

In this section, the subspace algorithm developed by Ivan Goethals et al [10] will be extended to the case where an ε-insensitive loss function is used as cost function. This cost function is a L-1 cost function, rather than L-2, which in consequence improves the robustness in the presence of outliers and missing data. Moreover, the value of ε is not necessarily restricted to be zero which results in sparse solution. we will follow the development in Goethals et al. [10], up until the point where the LS- SVM optimization is introduced (where we use a SVM).

The first step in any N4SID algorithm is to calculate the oblique projections (Projection of the future outputs onto the past inputs and outputs along the future inputs). These projections can be calculated as

$$\left\{ \begin{array}{r} \mathcal{O}_i = L_u\left(:, 1:i\right)\Phi_f\left(U_p\right) + L_y Y_p \\ \mathcal{O}_{i+1} = L_u^-\left(:, 1:(i+1)\right)\Phi_f\left(U_p^+\right) + L_y^- Y_p^+ \end{array} \right. \tag{3}$$

where f is a nonlinear function defined on $\mathbb{R}^m$ and Φ_f is defined as an operator on a block Hankel matrix such that

$$\Phi_f\left(\begin{bmatrix} Z_1 & Z_2 & \dots & Z_p \\ Z_2 & Z_3 & \dots & Z_{p+1} \\ \vdots & \vdots & & \vdots \\ Z_q & Z_{q+1} & \dots & Z_{p+q-1} \end{bmatrix}\right) = \begin{bmatrix} f(Z_1) & f(Z_2) & \dots & f(Z_p) \\ f(Z_2) & f(Z_3) & \dots & f(Z_{p+1}) \\ \vdots & \vdots & & \vdots \\ f(Z_q) & f(Z_{q+1}) & \dots & f(Z_{p+q-1}) \end{bmatrix} \in \mathbb{R}^{q\times p} \tag{4}$$

One can find estimates for the matrices $L_u \in \mathbb{R}^{i\times 2i}$, $L_u^- \in \mathbb{R}^{(i-1)\times 2i}$, $L_y \in \mathbb{R}^{i\times i}$, $L_y^- \in \mathbb{R}^{(i-1)\times(i+1)}$, and the static nonlinearity $f:\mathbb{R}\rightarrow\mathbb{R}$ in the least square sense from

$$\begin{gathered}\left(\hat{L}_u, \hat{L}_y, \hat{f}\right) = \underset{L_u, L_y, f}{\arg\min} \|E\|_F^2, \\ \text{s.t. } E = \begin{bmatrix} L_u & L_y \end{bmatrix} \begin{bmatrix} \Phi_{\hat{f}}\left(U_{0|2i-1}\right) \\ Y_p \end{bmatrix} - Y_f \end{gathered} \tag{5}$$

$$\begin{gathered}\left(\hat{L}_u^-, \hat{L}_y^-, \hat{f}\right) = \underset{L_u^-, L_y^-, f}{\arg\min} \|E^-\|_F^2, \\ \text{s.t. } E^- = \begin{bmatrix} L_u^- & L_y^- \end{bmatrix} \begin{bmatrix} \Phi_{\hat{f}}\left(U_{0|2i-1}\right) \\ Y_p^+ \end{bmatrix} - Y_f^- \end{gathered} \tag{6}$$

From (5) we have

$$\begin{gathered}Y_f(s,t) = L_y(s,:)\,Y_p(:,t) + \textstyle\sum_{h=1}^{2i} L_u(s,h)\, f(u_{h+t-2}) \\ +E(s,t)\end{gathered} \tag{7}$$

for $s = 1,\dots,il$ and $t = 1,\dots,j$. It is clear from (5), (6), (7) that L_u and f_k appear cross multiplied which makes the optimization problem nonconvex. To overcome this problem, we apply overparameterization.

Let

$$g_{s,h} = L_u(s,h)\, f \tag{8}$$

hence

$$Y_f(s,t) = L_y(s,:)\,Y_p(:,t) + \sum_{h=1}^{2i} g_{s,h}(u_{h+t-2}) + E(s,t) \tag{9}$$

Now, to formulate the SVM regression algorithm, let

$$g_{s,h} = w_{s,h}^T \varphi \tag{10}$$

Substituting (10) in (9) gives

$$Y_f(s,t) = L_y(s,:)\,Y_p(:,t) + \sum_{h=1}^{2i} w_{s,h}^T \varphi(u_{h+t-2}) + E(s,t) \tag{11}$$

As shown in [10], one should put in mind that expanding a nonlinear function as the sum of a set of nonlinear functions is not unique, for example

$$w_1^T\varphi_1(u) + w_2^T\varphi_2(u) = \left(w_1^T\varphi_1(u) + \delta\right) + \left(w_2^T\varphi_2(u) - \delta\right)$$

for all $\delta \in \mathbb{R}$. Such problem can be prevented by including centering constraints of the form

$$\sum_{t=0}^{N-1} f(u_t) = \sum_{t=0}^{N-1} w_{s,h}^T \varphi(u_t) = 0 \tag{12}$$

However, to apply centering constraints (12), new parameter δ_y should be added to (11) to get [10]

$$\begin{gathered}Y_f(s,t) + [1_i \otimes \delta_y](s) = L_y(s,:)\left(Y_p(:,t) + 1_i \otimes \delta_y\right) \\ + \textstyle\sum_{h=1}^{2i} w_{s,h}^T \varphi(u_{h+t-2}) + E(s,t)\end{gathered}$$

Where $\otimes$ denotes the matrix Kronecker product. The SVM primal problem will be

$$\begin{gathered}\min_{w,\xi} \mathcal{J}(w,\xi) = \tfrac{1}{2}\textstyle\sum_{s=1}^{i}\sum_{h=1}^{2i} w_{s,h}^T w_{s,h} \\ +\gamma \textstyle\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\xi(s,t) + \xi^*(s,t)\right)\end{gathered} \tag{13}$$

subject to

$$\begin{gathered}Y_f(s,t) - L_y(s,:)\,Y_p(:,t) - \textstyle\sum_{h=1}^{2i} w_{s,h}^T\varphi(u_{h+t-2}) \\ +\bar{d}(s) \le \varepsilon + \xi(s,t) \\ L_y(s,:)\,Y_p(:,t) + \textstyle\sum_{h=1}^{2i} w_{s,h}^T\varphi(u_{h+t-2}) - Y_f(s,t) \\ -\bar{d}(s) \le \varepsilon + \xi^*(s,t) \\ \textstyle\sum_{t=0}^{N-1} w_{s,h}^T\varphi(u_t) = 0 \\ \xi^*(s,t), \xi(s,t) \ge 0, \\ s = 1,\dots,i,\ h = 1,\dots,2i, \text{ and } t = 1,\dots,j\end{gathered} \tag{14}$$

where $\bar{d}(s) = [1_i \otimes \delta_y](s) - L_y(s,:)\left(1_i \otimes \delta_y\right)$

The optimization just described is the primal problem for regression. To formulate the corresponding dual problem, we have to write the Lagrangian function $\mathcal{L}$. Then, minimize $\mathcal{L}$ with respect to the weight vector w and slack variables ξ and ξ^* and maximize with respect to the Lagrange multipliers. By carrying out this optimization we can write w in terms of the Lagrange multipliers. Finally, we can substitute the value of w, use the so-called "kernel trick" [14], to replace the inner products with the kernel function, and simplify to get the following dual problem

$$\begin{gathered}\max_{\alpha,\alpha^*,\rho} \mathcal{L}(\alpha,\alpha^*,\rho) \\ = \tfrac{1}{2}\textstyle\sum_{s=1}^{i}\sum_{h=1}^{2i}\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1} \rho_{s,h}\rho_{s,h} K(u_{tt}, u_t) \\ -\tfrac{1}{2}\textstyle\sum_{s=1}^{i}\sum_{t_1=1}^{j}\sum_{t=1}^{j}\left(\alpha_{s,t_1} - \alpha_{s,t_1}^*\right)\left(\alpha_{s,t} - \alpha_{s,t}^*\right) \\ \times \textstyle\sum_{h=1}^{2i} K(u_{h+t_1-2}, u_{h+t-2}) \\ + \textstyle\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right)\left(Y_f(s,t)\right) \\ - \textstyle\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\alpha_{s,t} + \alpha_{s,t}^*\right)\varepsilon\end{gathered} \tag{15}$$

subject to

$$\begin{gathered}\textstyle\sum_{tt=1}^{j}\left(\alpha_{s,tt} - \alpha_{s,tt}^*\right)\sum_{t=0}^{N-1} K(u_{h+tt-2}, u_t) \\ + \textstyle\sum_{t=0}^{N-1}\sum_{t_2=0}^{N-1} \rho_{s,h} K(u_{t_2}, u_t) = 0 \\ \textstyle\sum_{t=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right)\left(Y_p(:,t)\right) = \mathbf{0}_i \\ \textstyle\sum_{t=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right) = 0 \\ 0 \le \alpha_{s,t}, \alpha_{s,t}^* \le \gamma \\ s = 1,\dots,i \\ h = 1,\dots,2i \\ t = 1,\dots,j\end{gathered} \tag{16}$$

Now to compute $\mathbf{L}_y$, one has to solve the following optimization problem

$$\begin{aligned}
\max_{\alpha,\alpha^*} & -\tfrac{1}{2}\textstyle\sum_{s=1}^{i}\sum_{t_1=1}^{j}\sum_{t=1}^{j}\left(\alpha_{L_{s,t_1}}-\alpha^*_{L_{s,t_1}}\right) \\
& \times\left(\alpha_{L_{s,t}}-\alpha^*_{L_{s,t}}\right)\textstyle\sum_{s_1=1}^{i}Y_p(s_1,t)Y_p(s_1,t_1) \\
& +\textstyle\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\alpha_{L_{s,t}}-\alpha^*_{L_{s,t}}\right)\Big(Y_f\left(s,t\right) \\
& -\textstyle\sum_{h=1}^{2i}\Big(\sum_{tt=1}^{j}\left(\alpha_{s,tt}-\alpha^*_{s,tt}\right)K_0\left(tt,h\right) \\
& +\rho_{s,h}\textstyle\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}K\left(u_{tt},u_t\right)\Big)\Big) \\
& -\textstyle\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\alpha_{L_{s,t}}-\alpha^*_{L_{s,t}}\right)\varepsilon
\end{aligned} \tag{17}$$

subject to

$$\begin{gathered}
\textstyle\sum_{t=1}^{j}\left(\alpha_{L_{s,t}}-\alpha^*_{L_{s,t}}\right)=0 \\
0\le\alpha_{L_{s,t}},\alpha^*_{L_{s,t}}\le\gamma \\
s=1,\ldots,i \\
h=1,\ldots,2i \\
t=1,\ldots,j
\end{gathered} \tag{18}$$

then L_y is given by

$$\begin{gathered}
L_y\left(s,s_1\right)=\textstyle\sum_{t=1}^{j}\left(\alpha_{Ls,t}-\alpha^*_{Ls,t}\right)Y_p(s_1,t) \\
s_1=1,\ldots,i
\end{gathered} \tag{19}$$

Finally, $\overline{\mathbf{d}}$ and δ_y can be calculated as

$$\begin{gathered}
\overline{\mathbf{d}}=\text{mean}\left(\mathfrak{Y}-\mathcal{Y}_P^T\cdot\boldsymbol{L}_y\right) \\
\delta_y=\left(\mathcal{M}^T\mathcal{M}\right)^{-1}\mathcal{M}^T\overline{\mathbf{d}}^T \\
\mathcal{M}=\left(1_i\otimes I_l-L_y\left(1_i\otimes I_l\right)\right) \\
\mathfrak{Y}\left(s,t\right)=Y_f\left(s,t\right) \\
-\textstyle\sum_{h=1}^{2i}\Big(\sum_{tt=1}^{j}\left(\alpha_{s,tt}-\alpha^*_{s,tt}\right)K_0\left(tt,h\right) \\
+\rho_{s,h}\textstyle\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}K\left(u_{tt},u_t\right)\Big)
\end{gathered} \tag{20}$$

Recalling (3)

$$\begin{gathered}
\mathcal{O}_i=L_u\left(:,1:i\right)\Phi_f\left(U_p\right)+L_y\left(Y_p(:,t)-\left(1_i1_j^T\right)\otimes\delta_y\right) \\
\mathcal{O}_i\left(s,t\right)=\textstyle\sum_{t_1=1}^{j}\left(\alpha_{s,t_1}-\alpha^*_{s,t_1}\right)\mathcal{K}_p\left(t_1,t\right) \\
+\textstyle\sum_{h=1}^{2i}\rho_{s,h}\mathcal{S}_p\left(h,t\right)+L_y\left(Y_p(:,t)-\left(1_i1_j^T\right)\otimes\delta_y\right)
\end{gathered} \tag{21}$$

where

$$\begin{aligned}
\mathcal{K}_p\left(t_1,t\right) &= \sum_{h=1}^{i}K\left(u_{h+t_1-2},u_{h+t-2}\right) \\
\mathcal{S}_p\left(h,t\right) &= \sum_{t_2=0}^{N-1}K\left(u_{t_2},u_{h+t-2}\right)
\end{aligned}$$

The same approach can be followed to calculate $\mathcal{O}_{i+1}$

$$\begin{gathered}
\max_{\alpha^-,\alpha^{*^-},\rho^-}\mathcal{L}\left(\alpha^-,\alpha^{*^-},\rho^-\right) \\
=\tfrac{1}{2}\textstyle\sum_{s=1}^{i-1}\sum_{h=1}^{2i}\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}\rho^-_{s,h}\rho^-_{s,h}K\left(u_{tt},u_t\right) \\
-\tfrac{1}{2}\textstyle\sum_{s=1}^{i-1}\sum_{t_1=1}^{j}\sum_{t=1}^{j}\left(\alpha^-_{s,t_1}-\alpha^{*^-}_{s,t_1}\right)\left(\alpha^-_{s,t}-\alpha^{*^-}_{s,t}\right) \\
\times\textstyle\sum_{h=1}^{2i}K\left(u_{h+t_1-2},u_{h+t-2}\right) \\
+\textstyle\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*^-}_{s,t}\right)\left(Y_f^-\left(s,t\right)\right) \\
-\textstyle\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{s,t}+\alpha^{*^-}_{s,t}\right)\varepsilon
\end{gathered} \tag{22}$$

subject to

$$\begin{gathered}
\textstyle\sum_{tt=1}^{j}\left(\alpha^-_{s,tt}-\alpha^{*^-}_{s,tt}\right)\sum_{t=0}^{N-1}K\left(u_{h+tt-2},u_t\right) \\
+\textstyle\sum_{t=0}^{N-1}\sum_{t_2=0}^{N-1}\rho^-_{s,h}K\left(u_{t_2},u_t\right)=0 \\
\textstyle\sum_{t=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*^-}_{s,t}\right)\left(Y_p^+(:,t)\right)=\mathbf{0}_i \\
\textstyle\sum_{t=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*^-}_{s,t}\right)=0 \\
0\le\alpha^-_{s,t},\alpha^{*^-}_{s,t}\le\gamma \\
s=1,\ldots,i-1 \\
h=1,\ldots,2i \\
t=1,\ldots,j
\end{gathered} \tag{23}$$

Now to compute $\mathbf{L}_y^-$, one has to solve the following optimization problem

$$\begin{aligned}
\max_{\alpha_L^-,\alpha_L^{*^-}} & -\tfrac{1}{2}\textstyle\sum_{s=1}^{i-1}\sum_{t_1=1}^{j}\sum_{t=1}^{j}\left(\alpha^-_{L_{s,t_1}}-\alpha^{*^-}_{L_{s,t_1}}\right) \\
& \times\left(\alpha^-_{L_{s,t}}-\alpha^{*^-}_{L_{s,t}}\right)\textstyle\sum_{s_1=1}^{i+1}Y_p^+(s_1,t)Y_p^+(s_1,t_1) \\
& +\textstyle\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{L_{s,t}}-\alpha^{*^-}_{L_{s,t}}\right)\Big(Y_f^-\left(s,t\right) \\
& -\textstyle\sum_{h=1}^{2i}\Big(\sum_{tt=1}^{j}\left(\alpha^-_{s,tt}-\alpha^{*^-}_{s,tt}\right)K_0\left(tt,h\right) \\
& +\rho^-_{s,h}\textstyle\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}K\left(u_{tt},u_t\right)\Big)\Big) \\
& -\textstyle\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{L_{s,t}}-\alpha^{*^-}_{L_{s,t}}\right)\varepsilon
\end{aligned} \tag{24}$$

subject to

$$\begin{gathered}
\textstyle\sum_{t=1}^{j}\left(\alpha^-_{L_{s,t}}-\alpha^{*^-}_{L_{s,t}}\right)=0 \\
0\le\alpha^-_{L_{s,t}},\alpha^{*^-}_{L_{s,t}}\le\gamma \\
s=1,\ldots,i-1 \\
h=1,\ldots,2i \\
t=1,\ldots,j
\end{gathered} \tag{25}$$

then L_y^- is given by

$$L_y^-\left(s,s_1\right)=\sum_{t=1}^{j}\left(\alpha^-_{Ls,t}-\alpha^{-*}_{Ls,t}\right)Y_p^+(s_1,t) \tag{26}$$

Recalling (3)

$$\begin{gathered}
\mathcal{O}_{i+1}=L_u^-\left(:,1:i+1\right)\Phi_f\left(U_p^+\right) \\
+L_y^-\left(Y_p^+(:,t)-\left(1_{i+1}1_j^T\right)\otimes\delta_y\right) \\
\mathcal{O}_{i+1}\left(s,t\right)=\textstyle\sum_{t_1=1}^{j}\left(\alpha^-_{s,t_1}-\alpha^{*^-}_{s,t_1}\right)\mathcal{K}_p^+\left(t_1,t\right) \\
+\textstyle\sum_{h=1}^{2i}\rho^-_{s,h}\mathcal{S}_p^-\left(h,t\right)+L_y^-\left(Y_p^+(:,t)-\left(1_{i+1}1_j^T\right)\otimes\delta_y\right)
\end{gathered} \tag{27}$$

where

$$\begin{aligned}
\mathcal{K}_p^+\left(t_1,t\right) &= \sum_{h=1}^{i+1}K\left(u_{h+t_1-2},u_{h+t-2}\right) \\
\mathcal{S}_p^-\left(h,t\right) &= \sum_{t_2=0}^{N-1}K\left(u_{t_2},u_{h+t-2}\right).
\end{aligned}$$

Similar to the linear N4SID algorithm, one should determine the extended observability matrices Γ_i and Γ_{i-1} to estimate the state sequences

$$\Gamma_i=U_1S_1^{1/2},\Gamma_{i-1}=\Gamma_i\left(1:i-1,:\right) \tag{28}$$

where U_1 and S_1 are obtained by partitioning the SVD of $\mathcal{O}_i$ as follows

$$\mathcal{O}_i = USV^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \quad (29)$$

Recall that the system order can be determined by inspecting the singular values of $\mathcal{O}_i$. Now, estimates for the state sequences can be computed from

$$\begin{cases} \tilde{X}_i = \Gamma_i^\dagger \mathcal{O}_i \\ \tilde{X}_{i+1} = \Gamma_{i-1}^\dagger \mathcal{O}_{i+1} \end{cases} \quad (30)$$

4. EXTRACTION OF THE SYSTEM MATRICES AND THE NONLINEARITY

Extraction of the System Matrices and the Nonlinearity

$$\begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{X}_i \\ f\left(U_{i|i}\right) \end{bmatrix} + \begin{bmatrix} \rho_\nu \\ \rho_\upsilon \end{bmatrix} \quad (31)$$

In what follows, a SVM regression problem will be formulated to identify (29). Denoting

$$\chi_{i+1} = \begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix}, \ \Theta_{AC} = \begin{bmatrix} A \\ C \end{bmatrix}, \ \Theta_{BD} = \begin{bmatrix} B \\ D \end{bmatrix}$$

results in

$$\chi_{i+1} = \Theta_{AC}\tilde{X}_i + \Theta_{BD}\Phi_f\left(U_{i|i}\right) + E \quad (32)$$

Replacing the product of scalars and nonlinear function '$\Theta_{BD}(s)\,f$' by a linear combination of nonlinear functions $w_s^T\varphi$, gives

$$\chi_{i+1} = \Theta_{AC}\tilde{X}_i + \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_{n+1}^T \end{bmatrix} \Phi_\varphi\left(U_{i|i}\right) + E \quad (33)$$

Now, The SVM primal problem is

$$\min_{w,\xi,\xi^*} \frac{1}{2}\sum_{s=1}^{n+1} w_s^T w_s + \gamma \sum_{s=1}^{n+1}\sum_{t=1}^{j}\left(\xi(s,t) + \xi^*(s,t)\right) \quad (34)$$

subject to

$$\begin{array}{c} \chi_{i+1}(s,t) - \Theta_{AC}(s,:)\,\tilde{X}_i(:,t) \\ +w_s^T\varphi\left(u_{i+t-2}\right) \le \varepsilon + \xi(s,t) \\ \Theta_{AC}(s,:)\,\tilde{X}_i(:,t) + w_s^T\varphi\left(u_{i+t-2}\right) \\ -\chi_{i+1}(s,t) \le \varepsilon + \xi^*(s,t) \\ \xi^*(s,t), \xi(s,t) \ge 0, \\ s = 1,\ldots,n+1 \text{ and } t = 1,\ldots,j \end{array} \quad (35)$$

By defining the Lagrangian, eliminating the primal variables w_s,$\xi(s,t)$, and $\xi^*(s,t)$, using the kernel trick and, simplification,the dual problem can be shown to be

$$\begin{array}{l} \max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{s=1}^{n+1}\sum_{t=1}^{j}\sum_{t_1=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right) \\ \times\left(\alpha_{s,t_1} - \alpha_{s,t_1}^*\right) K\left(u_{i+t-2}, u_{i+t_1-2}\right) \\ +\sum_{s=1}^{n}\sum_{t=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right)\chi_{i+1}(s,t) \\ -\sum_{s=1}^{n}\sum_{t=1}^{j}\left(\alpha_{s,t} + \alpha_{s,t}^*\right)\varepsilon \end{array} \quad (36)$$

subject to

$$\begin{array}{c} \sum_{t=1}^{j}\left(\alpha_{s,t} - \alpha_{s,t}^*\right)\tilde{X}_i(:,t) = 0 \\ 0 \le \alpha_{s,t}, \alpha_{s,t}^* \le \gamma \\ s = 1,\ldots,\ n+1 \\ t = 1,\ldots,j \end{array} \quad (37)$$

Now to estimate Θ_{AC}, we have to solve

$$\begin{array}{l} \min_{\Theta_{ACy}\xi} \mathcal{J}\left(\Theta_{AC},\xi,\xi^*\right) = \frac{1}{2}\sum_{s_1=1}^{n+1}\sum_{s_2=1}^{n}\Theta_{AC}^2(s_1,s_2) \\ +\gamma\sum_{s_1=1}^{n+1}\sum_{t=1}^{j}\left(\xi(s_1,t) + \xi^*(s_1,t)\right) \end{array} \quad (38)$$

subject to

$$\begin{array}{c} \mathfrak{Y}_{\Theta_{AC}}(s_1,t) - \Theta_{AC}(s_1,:)\,\tilde{X}_i(:,t) \le \varepsilon + \xi(s_1,t) \\ \Theta_{AC}(s_1,:)\,\tilde{X}_i(:,t) - \mathfrak{Y}_{\Theta_{AC}}(s_1,t) \le \varepsilon + \xi^*(s_1,t) \\ \xi^*(s_1,t), \xi(s_1,t) \ge 0,\ s_1 = 1,\ldots,n+1 \\ \text{and } t = 1,\ldots,j \end{array} \quad (39)$$

$$\begin{array}{c} \mathfrak{Y}_{\Theta_{AC}}(s_1,t) = \chi_{i+1}(s_1,t) \\ -\sum_{t_1=1}^{j}\left(\alpha_{s_1,t_1} - \alpha_{s_1,t_1}^*\right) K\left(u_{i+t-2}, u_{i+t_1-2}\right) \end{array} \quad 40$$

Whose dual counterpart is

$$\begin{array}{l} \min_{\alpha_{\Theta_{AC}}\alpha_{\Theta_{AC}}^*} \frac{1}{2}\sum_{s_1=1}^{n+1}\sum_{t_1=1}^{j}\sum_{t_2=1}^{j}\left(\alpha_{\Theta_{AC}s_1,t_1} - \alpha_{\Theta_{AC}s_1,t_1}^*\right) \\ \times\left(\alpha_{\Theta_{AC}s_1,t_2} - \alpha_{\Theta_{AC}s_1,t_2}^*\right)\sum_{s_2=1}^{n}\tilde{X}_i(s_2,t_1)\ \tilde{X}_i(s_2,t_2) \\ -\sum_{s_1=1}^{n+1}\sum_{t_1=1}^{j}\left(\alpha_{\Theta_{AC}s_1,t_1} - \alpha_{\Theta_{AC}s_1,t_1}^*\right)\mathfrak{Y}_{\Theta_{AC}}(s_1,t) \\ +\sum_{s_1=1}^{n+1}\sum_{t_1=1}^{j}\left(\alpha_{\Theta_{AC}s_1,t_1} + \alpha_{\Theta_{AC}s_1,t_1}^*\right)\varepsilon \end{array} \quad (41)$$

subject to

$$0 \le \alpha_{\Theta_{AC}s,t}, \alpha_{\Theta_{AC}s,t}^* \le \gamma \quad (42)$$

where

$$\begin{array}{c} \mathfrak{Y}_{\Theta_{AC}}(s_1,t) = \chi_{i+1}(s_1,t) \\ -\sum_{t_1=1}^{j}\left(\alpha_{s_1,t_1} - \alpha_{s_1,t_1}^*\right) K\left(u_{i+t-2}, u_{i+t_1-2}\right) \\ t = 1,\ldots,j \\ t_1 = 1,\ldots,j \\ s_1 = 1,\ldots,n+1 \end{array} \quad (43)$$

then Θ_{AC} is given by

$$\begin{array}{c} \Theta_{AC}(s_1,s_2) = \sum_{t=1}^{j}\left(\alpha_{\Theta_{AC}s_1,t} - \alpha_{\Theta_{AC}s_1,t}^*\right)\tilde{X}_i(s_2,t) \\ s_1 = 1,\ldots,n+1,\ s_2 = 1,\ldots,n \end{array} \quad (44)$$

To extract B and D in Θ_{BD} and the nonlinearity f, we use the solution presented in [10] , which involves using the SVD of a m by N matrix. Then, using the training input sequence $[u_1,\ldots,u_N]$ and the extracted sequence of the nonlinearity responses, we can train a SVM to represent the nonlinear part of the Hammerstein system.

4.1. Algorithm

the algorithm for Hammerstein N4SID subspace identification can be summarized as follows.

1) Obtain estimates for α, α^*, ρ by solving (15),(16).

2) Obtain estimates for α_L, α_L^* by solving (17),(18).

3) Compute $\mathbf{L}_y$, $\bar{\mathbf{d}}$, and δ_y using (19) and (20).

4) Find estimates for the oblique projection $\mathcal{O}_i$ from (21).

5) Obtain estimates for α^-, α^{*^-}, ρ^- by solving (22),(23).

6) Obtain estimates for α_L^-, $\alpha_L^{*^-}$ by solving (24),(25).

7) Compute $\mathbf{L}_y^-$, using (26).

8) Find estimates for the oblique projection $\mathcal{O}_{i+1}$ from (27).

9) Calculate the SVD of $\mathcal{O}_i$, and determine the order by inspecting the singular values and partition the SVD accordingly to obtain U_1 and S_1.

10) Compute the extended observability matrices Γ_i and Γ_{i-1} from (28).

11) Find estimates for the state using (30).

12) Obtain estimates for α, α^* by solving (36),(37).

13) Obtain estimates for $\alpha_{\Theta_{AC}}$, $\alpha^*_{\Theta_{AC}}$ by solving (41),(42).

14) Obtain estimates for Θ_{AC} using (44).

15) Obtain estimates for B and D and the nonlinearity f from a rank-m approximation presented in [10].

16) Use the input sequence $[u_1, u_2, \ldots, u_{n-1}]$ and the estimates of response of the nonlinearity to this input $[f(u_1), f(u_2), \ldots, f(u_{n-1})]$, to train a SVM to approximate the nonlinear function f.

5. ILLUSTRATIVE EXAMPLE

Consider the following system which belongs to the Hammerstein class of models:

$$A(z)(y+\nu) = B(z) f(u)$$

where

$$\begin{aligned} B(z) &= z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3 \\ A(z) &= \left(z - 0.98e^{\pm i}\right)\left(z - 0.98e^{\pm 1.6i}\right)\left(z - 0.97e^{\pm 0.4i}\right) \end{aligned}$$

such that $f(u) = \sin c(u)\, u^2$ be the static nonlinearity. A white Gaussian input sequence u with length 1000, zero mean and variance 2 was generated and fed into this system. The output noise sequence $\{v_t\}_{t=0}^{N-1}$ was chosen to be zero mean Gaussian white noise such that a signal to noise ratio of 10 was obtained at the output signal.

The number of block-rows i in the block Hankel matrices was set arbitrarily to 10. The hyper-parameters were chosen to be $\sigma = 1, \gamma = 1000, \gamma_{BD} = 10$.

From figure 1, it is clear that the algorithm estimated the nonlinearity very well.

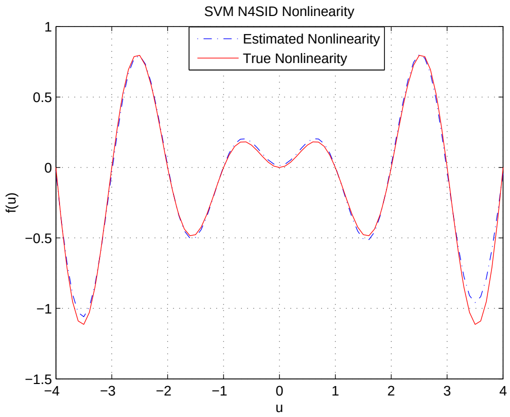

Figure 1. True nonlinearity, and mean of the SVM, with $\varepsilon = 0.001$, estimate with statistics estimated from a hundred trial Monte-Carlo simulation.

6. CONCLUSION

In this paper, the subspace algorithm developed by Ivan Goethals et al [10] has been extended to the case where an ε-insensitive loss function is used as cost function. The developed algorithm outperforms Goethal's algorithm in the following aspects. First, This algorithm uses ε-insensitive loss function as cost function. This cost function is a L-1 cost function, rather than L-2, which in consequence improves the robustness in the presence of outliers and missing data. Second, using this approach, one can adjust the compromise between model accuracy and parsimony as is clear from the example.

7. REFERENCES

[1] Ljung, L., System identification : theory for the user 1999, Upper Saddle River, NJ Prentice Hall PTR, c1999.

[2] Pelt, T. and Bernstein, D. Non-linear system identification using Hammerstein and non-linear feedback models with piecewise linear static maps. Int. J. Control, 74(18):1807–1823, 2001.

[3] Unbehauen, H., System Identification Methods using Parameter Estimation/A Survey. Annual Review in Automatic Programming. 12: p. 69-81, 1985.

[4] Sinha, N.K., System identification: from frequency response to soft computing. Proceedings of IEEE International Conference on Industrial Technology 2000. vol.1: p. 76-80, 2000.

[5] Astrom, K.J. and P. Eykhoff, System identification-a survey. Automatica. 7(2): p. 123-62, 1971.

[6] Verhaegen, M. and Dewilde P. Subspace Model Identification Part 2. Analysis of the Elementary Output-Error State-Space Model Identification Algorithm, Int. J. Control, 56(5): 1211-1241, 1992.

[7] Van Overschee, P. and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, 30,75-93, 1994.

[8] Larimore, W. Canonical variate analysis in identification, filtering, and adaptive control. Proceedings of the IEEE Conference on Decision and Control, 2, 596-604, 1990.

[9] M. Verhaegen and D. T. Westwick. Identifying MIMO Hammerstein Systems in the Context of Subspace Model Identification Methods. Int. J. Control, 63(2): 331-49, 1996.

[10] Goethals, I., et al. Subspace identification of Hammerstein systems using least squares support vector machines. IEEE Transactions on Automatic Control 50(10): 1509-19, 2005.

[11] Bako, L., et al.Recursive subspace identification of Hammerstein models based on least squares support vector machines. IET Control Theory and Applications 3(9): 1209-1216,2009.

[12] Westwick, D. and M. Verhaegen, Identifying MIMO Wiener systems using subspace model identification methods. Signal Processing. 52(2): p. 235-58, 1996.

[13] Westwick, D. and Kearney, R. Identi…cation of Nonlinear Physiological Systems 2003,Piscataway, NJ John Wiley & Sons, 2003.

[14] Vapnik, V.N., Statistical Learning Theory 1998, New York: New York John Wiley & Sons, Inc. (US), 1998.

[15] Suykens, J., De Brabanter, T., De Moor, B., Vandewalle, J., Least Squares Support Vector Machines 2002, Singapore: World Scientific Pub. Co., 2002.