Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Identification of NARX Hammerstein Models Based on Support Vector Machines

Mujahed Al Dhaifallah and David T. Westwick

[*] *Department of Electrical and Computer Engineering, Schulich School of Engineering, Calgary, Canada*

Abstract: This paper presents a new algorithm for identification of NARX Hammerstein systems using support vector machines (SVMs) to model the static nonlinear elements. The SVM is fitted by minimizing an ε-insensitive, L-1 cost function which is robust in the presence of outliers. Another advantage of this algorithm is that the value of the uncertainty level epsilon can be specified by the user which gives more control on the sparseness of the solution. The effect of this choice is demonstrated using simulations.

Keywords: Hammerstein, Identification, Support Vector Machines

1. INTRODUCTION

Having an accurate system model is important in many control systems applications. While *a priori* modeling approaches can be used to develop models of simple systems, this approach becomes impractical as the system complexity increases. System identification can be used to find a mathematical description from measured input/output data. In either case, the resulting model will only be an approximation to the true system.

The goal of a system identification, is to find the model, within a selected class of models, that produces the best predictions of the system's output. In general, one forms a cost function that depends on some norm of the prediction errors, and finds the model that minimizes this cost function explicitly (Prediction Error Methods) or analytically (Subspace Methods). Thus, system identification can be viewed as an optimization problem.

Since the model is an approximation to the true system, there is a trade-off between the complexity of the model, and the accuracy of its predictions. In many cases, linear models can be used to produce accurate predictions of a system's behavior, particularly, if it is restricted to operating within a narrow region. If the model is required to cover a broader operating region, then a nonlinear model may be required [Schoukens et al., 2005]. Block structured models, cascades of static nonlinearities and dynamic linear systems, can often be used to represent nonlinear systems. They retain much of the simplicity of linear models, but can nevertheless be used to approximate many nonlinear systems very accurately. The simplest of these is the Hammerstein cascade, consisting of a memoryless nonlinearity followed by a dynamic linear element.

Many algorithms have been proposed to identify Hammerstein systems. They differ in the representations used for the linear and nonlinear parts. If both of the parts are parametric, then the algorithm is considered to be parametric. If both are nonparametric (as in Greblicki and Pawlak [1989]), then the algorithm is considered to be non-parametric. The static nonlinearity has been represented in many ways. It has been represented as a sum of basis functions [Hachino et al., 2004], a neural network [Janczak, 2003], and finite number of cubic spline functions [Dempsey and Westwick, 2004].

By expanding the mathematical model of any Hammerstein system, one will end up with expression involving cross-products between the linear dynamical system parameters and the static nonlinearity parameters which in consequence apply to the final cost function. This results in a non-convex optimization problem. To overcome this difficulty, some authors use an overparameterization technique, where one replaces every cross product term by a new independent parameter, resulting in an overparameterized, but convex, problem. Then, a rank one approximation technique, like singular value decomposition, is used to project the overparameterized system onto the Hammerstein model class, as suggested by Bai [1998].

Recently, support vector machines (SVMs) and least squares support vector machines (LS-SVMs) have shown powerful ability in approximating linear and nonlinear functions [Vapnik, 1998, Suykens et al., 2002]. In San and Ge [2004] and Wang and Ye [2004], the authors employed conventional SVM and LS-SVM to model general nonlinear systems. Adaptive SVM and LS-SVM methods for nonlinear system identification were proposed in Resendiz-Trejo et al. [2006] and Wang et al. [2006]. In [Rojo-Álvarez et al., 2004], the authors proposed new approaches to linear ARX identification based on SVMs.

Moreover, some authors [Espinoza et al., 2004, Goethals et al., 2004, 2005a,b] used LS-SVM to represent the nonlinear part of block structured nonlinear models. In Goethals et al. [2005a], a method for the identification of Hammerstein models based on LS-SVMs was proposed. The main advantage of this algorithm is that it allows for the determination of the memoryless static nonlinearity as well as the estimation of the model parameters of the

dynamic ARX part. In addition, we can benefit from the flexibility of using SVM to represent nonlinearities.

In this paper, we propose a new algorithm for identification of NARX Hammerstein systems using support vector machines (SVM). The proposed algorithm differs from the algorithm proposed in Goethals et al. [2005a] in two aspects. First, 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. Second, the value of ε is not necessarily restricted to be zero which results in sparse solution. We will investigate the relationship between sparseness and accuracy, as a function of ε.

The outline of this paper is as follows: support vector machine theory will be reviewed in Section 2. In Section 3, an algorithm for the identification of Hammerstein systems is proposed. Section 4 presents an illustrative example to test the proposed algorithm.

2. SUPPORT VECTOR MACHINES FOR FUNCTION ESTIMATION

Basically, to construct a support vector machine for real-valued function estimation problems, the input data are mapped into a high-dimensional feature space where a linear function is constructed. A kernel function is used to avoid constructing this mapping explicitly.

2.1 STANDARD SVM REGRESSION

Consider the nonlinear regression model $y = f(\mathbf{x}) + v$ where $f : \mathbb{R}^d \to \mathbb{R}$ is an unknown scalar-valued function and v is an additive white noise term. $\mathbf{x}_i$ is a sample value of the input vector $\mathbf{x}$ and y_i is the corresponding value of the model output y. In the primal space, the following model is assumed for $f(\mathbf{x})$

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_H}$ denotes a mapping to high dimensional feature space which can be infinite dimensional, $\mathbf{w}$ is a vector of weights in this feature space, and b represents the bias term. Before formulating the optimization problem which is used to compute the parameters of $f(\mathbf{x})$, we need to introduce the so called $\varepsilon-$insensitive linear loss function described by

$$\xi = |y - f(x)|_\varepsilon = \begin{cases} 0, & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases}$$

Figure 1 depicts the situation graphically. It is clear from this figure that in SVM regression, a tube with radius ε is fitted to the data. Since the points outside the tube are the only ones that contribute to the ε-insensitive cost function, they are referred to as the support vectors. Now, to find an estimate of the dependence of y on $\mathbf{x}$ in the standard SVM sense, a cost function consisting of a weighted average of ε-insensitive cost function and the L-2 norm of the weight vector is minimized,

$$\min J(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{N}((\xi_i) + (\xi_i^*)) \qquad (1)$$

subject to

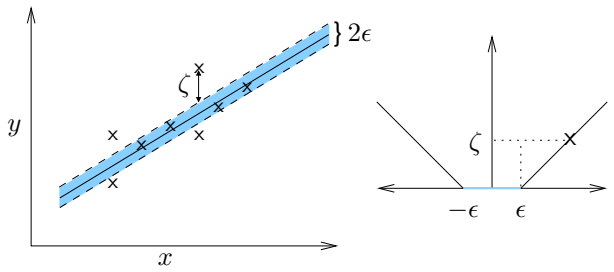

Fig. 1. (Left) Tube of $\varepsilon-$accuracy; (Right) Vapnik $\varepsilon-$insensitive linear loss function

$$\begin{aligned} y_i - \mathbf{w}^T\varphi(\mathbf{x}_i) - b &\le \varepsilon + \xi_i \\ \mathbf{w}^T\varphi(\mathbf{x}_i) + b - y_i &\le \varepsilon + \xi_i^* \\ \xi_i^*, \xi_i &\ge 0,\ i = 1, \ldots, l, \end{aligned} \qquad (2)$$

where ε is the accuracy level of the approximation, $c > 0$ is a constant that determines the relative weighting of the two terms, and ξ_i and ξ_i^* are the errors in the $\epsilon-$ insensitive cost function, shown in Figure 1, which are treated as slack variables in the optimization problem.

The optimization of $J(\mathbf{w}, \xi)$ just described is the primal problem for regression. To formulate the corresponding dual problem, we write the Lagrangian function L. Then, we minimize L with respect to the weight vector $\mathbf{w}$ and slack variables ξ and $\xi^{'}$ and maximize with respect to the Lagrange multipliers. By carrying out this optimization we can write $\mathbf{w}$ in terms of the Lagrange multipliers. Finally, we can substitute the value of $\mathbf{w}$ and simplify to get the following dual problem

$$\max W(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{N}\Big((\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)$$

$$\times K(\mathbf{x}_i, \mathbf{x}_j)\Big) + \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)y_i - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\varepsilon \qquad (3)$$

subject to

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0,$$

$$0 \le \alpha_i \le c,\ i = 1, \ldots, N$$
$$0 \le \alpha_i^* \le c,\ i = 1, \ldots, N$$

where α_i and $\alpha_i^{'}$ are the Lagrange multipliers. Finally, the nonlinear function model takes the form

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(\mathbf{x}, \mathbf{x}_i) + b \qquad (4)$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function used to represent the inner product in the feature space

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) \qquad (5)$$

Kernels $K(\mathbf{x}_i, \mathbf{x}_j)$ can be any symmetric function satisfying Mercer's condition [Vapnik, 1998]. Typical examples are the use of a polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \left(\tau + \mathbf{x}_i^T\mathbf{x}_j\right)^d$ of degree d or the RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2\right)$, where $\sigma \in \mathbb{R}^+$ denotes the bandwidth of the kernel.

3. IDENTIFICATION OF NONLINEAR ARX HAMMERSTEIN MODELS

The Hammerstein cascade, a static nonlinearity followed by a linear filter as shown in Fig. 2, is often used to represent certain higher-order nonlinear systems.

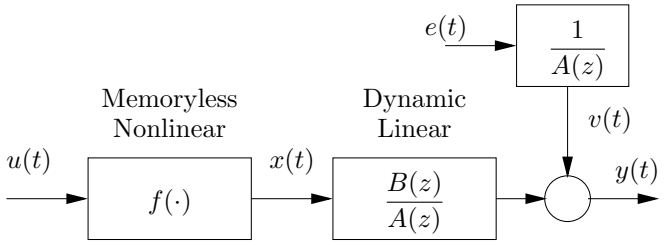

Fig. 2. Block diagram of an ARX-Hammerstein cascade. The investigator is assumed to have access to the input, $u(t)$, and the output, $y(t)$, but not the intermediate signal, $x(t)$ or the innovation, $e(t)$.

In this section, we are following the development in Goethals et al. [2005a], up until the point where the LS-SVM optimization is introduced (where we use a SVM). The output of the NARX model is given by:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f\left(u_{t-j}\right) + e_t \tag{6}$$

Where $u_t, y_t \in \mathbb{R}$, are the input and output measurements, respectively, for $t = 1 \ldots N$. The noise e_t is assumed to be white and m and n denote the order of the numerator and denominator in the transfer function of the linear model. The static nonlinearity is assumed to have the following form:

$$f\left(u\right) = \mathbf{w}^T \varphi\left(u\right) + d_0 \tag{7}$$

Hence (7) can be rewritten as follows

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j \left(\mathbf{w}^T \varphi\left(u_{t-j}\right) + d_0\right) + e_t \tag{8}$$

Since b_j and $\mathbf{w}$ never appear alone in (8), they cannot be uniquely identified. As an initial step, define $\mathbf{w}_j$ and d as

$$\mathbf{w}_j = b_j \mathbf{w};\; d = d_0 \sum_{j=0}^{m} b_j$$

Hence, (8) can be rewritten as

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \mathbf{w}_j^T \varphi\left(u_{t-j}\right) + d + e_t \tag{9}$$

Note that models of this form can be uniquely identified, but this model class is more general than the Hammerstein model, which it includes as a special case (when $\mathbf{w}_j = b_j \mathbf{w}$ for $j = 1..m$). The strategy will be to identify this model first, and then use a low-rank projection to force the estimated model to be a Hammerstein cascade.

So far, the development has followed the method presented in Goethals et al. [2005a] for LS-SVMs. To use a SVM instead, we must use the following optimization to identify the linear and nonlinear parts

$$\min_{\mathbf{w}_j, \xi, \xi^*} J\left(\mathbf{w}, \xi, \xi^*\right) = \frac{1}{2} \sum_{j=0}^{m} \mathbf{w}_j^T \mathbf{w}_j + c \sum_{t=r}^{N} \left(\xi_t + \xi_t^*\right) \tag{10}$$

subject to

$$y_t - \sum_{i=1}^{n} a_i y_{t-i} - \sum_{j=0}^{m} \mathbf{w}_j^T \varphi\left(u_{t-j}\right) - d \leq \varepsilon + \xi_t$$

$$\sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \mathbf{w}_j^T \varphi\left(u_{t-j}\right) + d - y_t \leq \varepsilon + \xi_t^* \tag{11}$$

$$\xi_t, \xi_t^* \geq 0,\; t = r, \ldots, N \tag{12}$$

$$\sum_{t_1=1}^{N} \mathbf{w}_j^T \varphi\left(u_{t_1}\right) = 0,\; j = 0, \ldots, m \tag{13}$$

Note that (10) is identical to the standard SVM objective, (1). The constraints in (11) are derived by modifying (2) to include the dynamics of the ARX model. Constraints (13) were added to center the nonlinear functions $\mathbf{w}_j^T \varphi\left(\cdot\right)$, $j = 0, \ldots, m$ around their average over the training set [Goethals et al., 2005a]. From these constraints, one can show that

$$d_0 = \frac{1}{N} \sum_{t_1=1}^{N} f\left(u_{t_1}\right)$$

Hence

$$d = \left(\frac{1}{N} \sum_{t_1=1}^{N} f\left(u_{t_1}\right)\right) \sum_{j=0}^{m} b_j \tag{14}$$

The Lagrangian is defined as

$$\begin{aligned} &L\left(\mathbf{w}_j, d, \xi, \xi^*, \boldsymbol{a}; \alpha, \alpha^*, \beta, \beta^*, \gamma\right) = \\ &\frac{1}{2} \sum_{j=0}^{m} \mathbf{w}_j^T \mathbf{w}_j + c \sum_{t=r}^{N} \left(\xi_t + \xi_t^*\right) \\ &- \sum_{j=0}^{m} \gamma_j \Big(\sum_{t_1=1}^{N} \mathbf{w}_j^T \varphi\left(u_{t_1}\right) \Big) - \sum_{t=r}^{N} \alpha_t \Big(\sum_{i=1}^{n} a_i y_{t-i} \\ &+ \sum_{j=0}^{m} \mathbf{w}_j^T \varphi\left(u_{t-j}\right) + d - y_t + \varepsilon + \xi_t \Big) - \sum_{t=r}^{N} \alpha_t^* \Big(y_t \\ &- \sum_{i=1}^{n} a_i y_{t-i} - \sum_{j=0}^{m} \mathbf{w}_j^T \varphi\left(u_{t-j}\right) - d + \varepsilon + \xi_t^* \Big) \\ &- \sum_{t=r}^{N} \left(\beta_t \xi_t + \beta_t^* \xi_t^*\right) \end{aligned} \tag{15}$$

where α_i, α_i^*, β_i, β_i^* are non-negative Lagrange multipliers and $\gamma_j \in \mathbb{R}$. Setting $\frac{\partial L}{\partial \mathbf{w}_j}$ to zero yields

$$\mathbf{w}_j = \gamma_j \sum_{t_1=1}^{N} \varphi\left(u_{t_1}\right) + \sum_{t=r}^{N} \left(\alpha_t - \alpha_t^*\right) \varphi\left(u_{t-j}\right) \tag{16}$$

Which leads to

$$\begin{aligned} \mathbf{w}_j^T \varphi\left(u_*\right) = &\gamma_j \sum_{t_1=1}^{N} \varphi\left(u_{t_1}\right)^T \varphi\left(u_*\right) \\ &+ \sum_{t=r}^{N} \left(\alpha_t - \alpha_t^*\right) \varphi\left(u_{t-j}\right)^T \varphi\left(u_*\right) \end{aligned}$$

$$= \gamma_j \sum_{t_1=1}^{N} K(u_{t_1}, u_*) + \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) K(u_{t-j}, u_*)$$

From the last expression and the centering constraints (13), one can show that

$$\gamma_j \sum_{t_2=1}^{N} \sum_{t_1=1}^{N} K(u_{t_2}, u_{t_1}) + \sum_{t=r}^{N} \sum_{t_1=1}^{N} (\alpha_t - \alpha_t^*) K(u_{t-j}, u_{t_1}) = 0, \ j = 0, \ldots, m \quad (17)$$

$$\frac{\partial L}{\partial d} = 0 \rightarrow \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) = 0 \quad (18)$$

$$\frac{\partial L}{\partial a_i} = 0 \rightarrow \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) y_{t-i} = 0, \ i = 1, \ldots, n \quad (19)$$

$$\frac{\partial L}{\partial \xi_t} = 0 \rightarrow \alpha_t + \beta_t = c, \ t = r, \ldots, N$$
$$\frac{\partial L}{\partial \xi_t^*} = 0 \rightarrow \alpha_t^* + \beta_t^* = c, \ t = r, \ldots, N \quad (20)$$

Substituting (16) and $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ into the constraints (11) leads to

$$y_t - \sum_{i=1}^{n} a_i y_{t-i} - \sum_{j=0}^{m} \gamma_j \sum_{t=1}^{N} K(u_t, u_{t-j}) - \sum_{j=0}^{m} \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) K(u_{t-j}, u_{t-j}) - d \leq \varepsilon + \xi_t$$

$$\sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \gamma_j \sum_{t=1}^{N} K(u_t, u_{t-j}) + \sum_{j=0}^{m} \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) K(u_{t-j}, u_{t-j}) + d - y_t \leq \varepsilon + \xi_t^* \ , t = r, \ldots, N \quad (21)$$

From (16), and using $K = \varphi^T \varphi$, the Lagrangian, in the minimization (15), can be written in matrix form

$$\min_{\alpha, \alpha^*, \gamma} L(\alpha, \alpha^*, \gamma) =$$

$$\frac{1}{2} \begin{bmatrix} \gamma^T & \alpha^T & \alpha^{*T} \end{bmatrix} \begin{bmatrix} -\mathcal{S} I_{m+1} & 0 & 0 \\ 0 & \mathcal{K} & -\mathcal{K} \\ 0 & -\mathcal{K} & \mathcal{K} \end{bmatrix} \begin{bmatrix} \gamma \\ \alpha \\ \alpha^* \end{bmatrix} + \begin{bmatrix} 0 & -y_{r:N}^T & y_{r:N}^T \end{bmatrix} \begin{bmatrix} \gamma \\ \alpha \\ \alpha^* \end{bmatrix} + \begin{bmatrix} 0 & \varepsilon \cdot \mathbf{1}_{N-r+1}^T & \varepsilon \cdot \mathbf{1}_{N-r+1}^T \end{bmatrix} \begin{bmatrix} \gamma \\ \alpha \\ \alpha^* \end{bmatrix} \quad (22)$$

subject to

$$\sum_{t=r}^{N} (\alpha_t - \alpha_t^*) = 0$$

$$\begin{bmatrix} \mathcal{Y}_P^T & -\mathcal{Y}_P^T \end{bmatrix} \begin{bmatrix} \alpha \\ \alpha^* \end{bmatrix} = 0,$$

$$\gamma_j \mathcal{S} + \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) K^0(t, j) = 0, \ j = 0, \ldots, m$$

$$0 \leq \alpha_t^* \leq c$$
$$0 \leq \alpha_t \leq c$$
$$t = r, \ldots, N$$

with

$$\mathcal{Y}_P = \begin{bmatrix} y_{r-1} & y_r & \cdots & y_{N-1} \\ y_{r-2} & y_{r-1} & \cdots & y_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{r-n} & y_{r-n+1} & \cdots & y_{N-n} \end{bmatrix}$$

$$\mathcal{K}(p, q) = \sum_{j=0}^{m} K(u_{p+r-j-1}, u_{q+r-j-1})$$

$$K^0(t, j) = \sum_{t_1=1}^{N} K(u_{t_1}, u_{t+r-j})$$

$$\mathcal{S} = \sum_{t_1=1}^{N} \sum_{t_2=1}^{N} K(u_{t_1}, u_{t_2})$$

Now to compute d and $\boldsymbol{a}$, one has to solve the following optimization problem which has been derived and proved in Rojo-Álvarez et al. [2004].

$$\min_{\alpha^L, \alpha^{L^*}} L\left(\alpha^L, \alpha^{L^*}\right) =$$

$$\frac{1}{2} \begin{bmatrix} \alpha^{L^T} & \alpha^{L^{*T}} \end{bmatrix} \begin{bmatrix} \mathcal{R}_y & -\mathcal{R}_y \\ -\mathcal{R}_y & \mathcal{R}_y \end{bmatrix} \begin{bmatrix} \alpha^L \\ \alpha^{L^*} \end{bmatrix} + \begin{bmatrix} \varepsilon \cdot \mathbf{1}_{N-r+1}^T - \mathfrak{Y}^T & \varepsilon \cdot \mathbf{1}_{N-r+1}^T + \mathfrak{Y}^T \end{bmatrix} \begin{bmatrix} \alpha^L \\ \alpha^{L^*} \end{bmatrix} \quad (23)$$

s.t.

$$\sum_{t=r}^{N} \left(\alpha_t^L - \alpha_t^{L^*}\right) = 0$$

$$0 \leq \alpha_t^{L^*} \leq c$$
$$0 \leq \alpha_t^L \leq c$$
$$t = r, \ldots, N$$

with

$$\mathcal{R}_y(p, q) = \sum_{i=1}^{n} y_{p+r-i-1} y_{q+r-i-1} \quad (24)$$

$$\mathfrak{Y}(t) = y_t - \sum_{j=0}^{m} \left(\gamma_j \sum_{t_1=1}^{N} K(u_{t_1}, u_{t-j}) + \sum_{t_2=r}^{N} \left(\alpha_{t_2} - \alpha_{t_2}^*\right) K(u_{t_2-j}, u_{t-j}) \right) \quad (25)$$

Then, a_i and d are given by

$$a_i = \sum_{t=r}^{N} \left(\alpha_t^L - \alpha_t^{L^*} \right) y_{t-i} \tag{26}$$

$$d = \text{mean}\left(\mathfrak{Y} - \mathcal{Y}_P^T \cdot \boldsymbol{a} \right) \tag{27}$$

3.1 Separating Numerator and Nonlinearity Parameters

In this section, we return to the development in Goethals et al. [2005a], since this material is independent of the type of SVM employed in the model. Recall

$$f(u) = \mathbf{w}^T \varphi(u) + d_0$$

Substituting $d_0 = \frac{1}{N} \sum_{t_1=1}^{N} f(u_{t_1})$ into the last equation results in

$$f(u) = \mathbf{w}^T \varphi(u) + \frac{1}{N} \sum_{t_1=1}^{N} f(u_{t_1})$$

Subtracting $\frac{1}{N} \sum_{t_1=1}^{N} f(u_{t_1})$ from both sides and replacing $f(u) - (1/N) \sum_{t=1}^{N} f(u_t)$ with $\underline{f}(u)$ gives

$$\underline{f}(u) = \mathbf{w}^T \varphi(u)$$

Multiplying both sides by b_j gives

$$b_j \underline{f}(u) = b_j \mathbf{w}^T \varphi(u)$$

Recalling $\mathbf{w}_j = b_j \mathbf{w}^T$, the last expression can be rewritten as

$$b_j \underline{f}(u) = \mathbf{w}_j^T \varphi(u)$$

Substituting (16) and (5) into the last equation results in

$$b_j \underline{f}(u) = \gamma_j \sum_{t_1=1}^{N} K(u_{t_1}, u) + \sum_{t=r}^{N} (\alpha_t - \alpha_t^*) K(u_{t-j}, u)$$

Based on the last expression, one can show that for the training input sequence $[\, u_1 \ \cdots \ u_N \,]$, the following equality holds

$$\begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} \widehat{\underline{f}}(u_1) \\ \vdots \\ \widehat{\underline{f}}(u_N) \end{bmatrix}^T$$

$$= \begin{bmatrix} \alpha_N - \alpha_N^* & \cdots & \alpha_r - \alpha_r^* & & 0 \\ & \alpha_N - \alpha_N^* & \cdots & \alpha_r - \alpha_r^* & \\ & & \ddots & & \ddots \\ 0 & & & \alpha_N - \alpha_N^* & \cdots \ \alpha_r - \alpha_r^* \end{bmatrix}$$

$$\times \begin{bmatrix} K(N,1) & K(N,2) & \cdots & K(N,N) \\ K(N-1,1) & K(N-1,2) & \cdots & K(N-1,N) \\ \vdots & \vdots & & \vdots \\ K(r-m,1) & K(r-m,2) & \cdots & K(r-m,N) \end{bmatrix}$$

$$+ \begin{bmatrix} \gamma_0 \\ \vdots \\ \gamma_m \end{bmatrix} \sum_{t=1}^{N} \begin{bmatrix} K(t,1) \\ \vdots \\ K(t,N) \end{bmatrix}^T \tag{28}$$

with $\widehat{\underline{f}}(u)$ an estimate for $\underline{f}(u) = f(u) - (1/N) \sum_{t=1}^{N} f(u_t)$. In consequence, estimates for b_j and the static nonlinearity f can be obtained from a rank 1 approximation of the right-hand side of (28), for example using singular value decomposition. After obtaining the estimates of b_j, estimate for $\sum_{t=1}^{N} f(u_t)$ can be obtained as

$$\sum_{t=1}^{N} f(u_t) = \frac{Nd}{\sum_{j=0}^{m} b_j}$$

hence

$$f(u) = \underline{f}(u) + (1/N) \sum_{t=1}^{N} f(u_t)$$

Now, using the training input sequence $[\, u_1 \ \cdots \ u_N \,]$ and the sequence of the nonlinearity responses to this input $[\, f(u_1) \ \cdots \ f(u_N) \,]$, we can train a support vector machine to represent the nonlinear part of the Hammerstein system.

4. ILLUSTRATIVE EXAMPLE

To test the proposed algorithm, the simulation example presented in Goethals et al. [2005a] is repeated. However, in this case the noise was uniformly distributed.

Based on validation test the regularization parameter was set to $c = 50$. An RBF-kernel with $\sigma = 1$ was used.

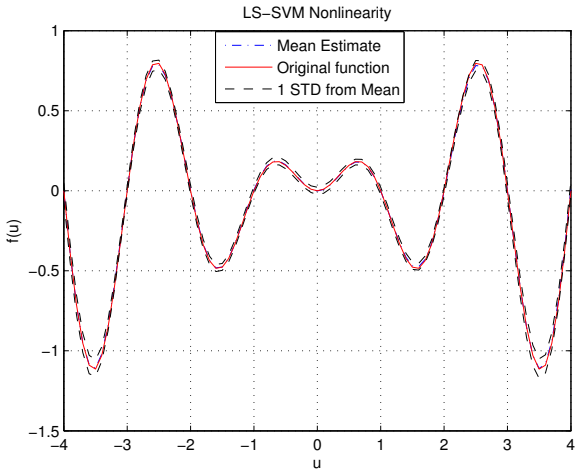

Fig. 3. True nonlinearity, and mean plus and minus one standard deviation of the LS-SVM estimate. Statistics are estimated from a twenty trial Monte-Carlo simulation

Table 1. Mean absolute error and mean square error between true and estimated nonlinearity and the average number of Support Vectors

Method	Mean absolute error	Mean Square error	AVG number of SV
LS-SVM	0.0037	2.6×10^{-5}	194
SVM with $\varepsilon = 0.01$	0.0048	4.5×10^{-5}	190
SVM with $\varepsilon = 0.1$	0.007	1.1×10^{-4}	115

Figure 3 shows the mean nonlinearity estimated using the algorithm proposed in Goethals et al. [2005a] while Figures 4 and 5 show the mean nonlinearity estimated using the proposed algorithm with $\varepsilon = 0.01$. and $\varepsilon = 0.1$ respectively. From Table 1, it is clear that the model

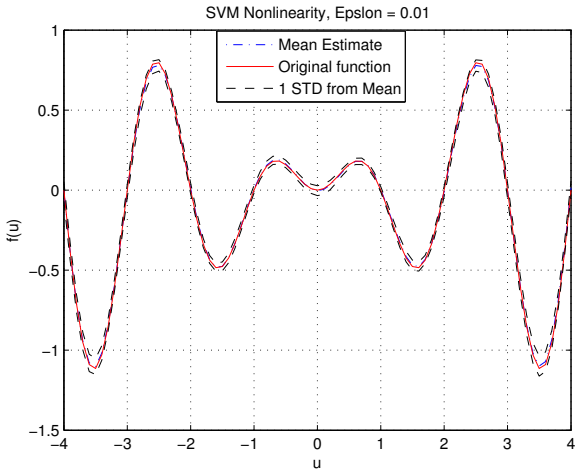

Fig. 4. True nonlinearity, and mean plus and minus one standard deviation of the SVM estimate. Statistics are estimated from a twenty trial Monte-Carlo simulation. The SVM was fitted with an ϵ-insensitive loss function with $\epsilon = 0.01$

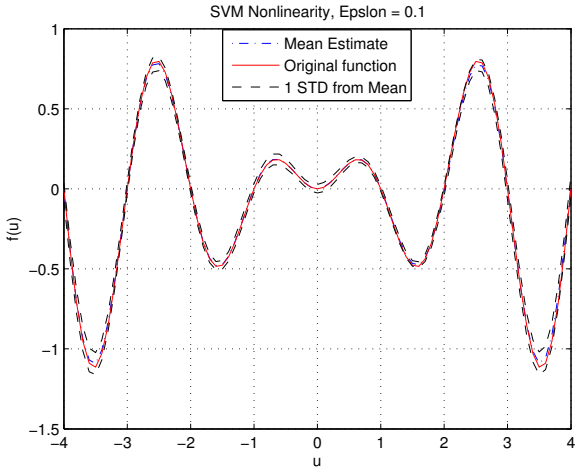

Fig. 5. True nonlinearity, and mean plus and minus one standard deviation of the SVM estimate. Statistics are estimated from a twenty trial Monte-Carlo simulation. The SVM was fitted with an ϵ-insensitive loss function with $\epsilon = 0.1$

produced by Goethals et al. [2005a] algorithm used more support vectors to fit the nonlinearity than the proposed algorithm. Furthermore, by increasing the uncertainty level from $\varepsilon = 0.01$ to $\varepsilon = 0.1$, the number of support vectors decreased by 40% but the mean absolute error and mean square error increased. So, one has to compromise between the number of support vectors and the estimation error.

5. CONCLUSION

In this paper a new algorithm for identification of NARX Hammerstein systems using support vector machines (SVM) has been derived. It was clear from the SISO example that increasing the uncertainty level ε decreases the number of support vectors needed to estimate the nonlinearity but the mean absolute error and mean square error increases. Using this approach, one can adjust the compromise between model accuracy and parsimony.

REFERENCES

E.-W. Bai. An optimal two stage identification algorithm for Hammerstein-Wiener nonlinear systems. In *Proc. ACC*, pages 2756–2760, June 1998.

E.J. Dempsey and D.T. Westwick. Identification of Hammerstein models with cubic spline nonlinearities. *IEEE Trans. Biomed. Eng.*, 51(2):237–245, February 2004.

M. Espinoza, J. Suykens, and B. De Moor. Partially linear models and least squares support vector machines. In *Proc. IEEE CDC*, volume 43, pages 3388–3393, December 2004.

I. Goethals, K. Pelckmans, J. Suykens, and B. De Moor. NARX identification of Hammerstein models using least squares support vector machines. In *NOLCOS 2004*, pages 507–512, 2004.

I. Goethals, K. Pelckmans, J. Suykens, and B. De Moor. Identification of MIMO Hammerstein models using least squares support vector machines. *Automatica*, 41(7): 1263–1272, jul 2005a.

I. Goethals, K. Pelckmans, J. Suykens, and B. De Moor. Subspace identification of Hammerstein systems using least squares support vector machines. *IEEE Trans. Auto. Cntrl.*, 50(10):1509–1519, oct 2005b.

W. Greblicki and M. Pawlak. Nonparametric identification of Hammerstein systems. *IEEE Trans. Info. Thy.*, 35(2): 409–418, 1989.

T. Hachino, K. Deguchi, and H. Takata. Identification of Hammerstein model using radial basis function. In *Proc. Asian Cont. Conf.*, pages 124–129, 2004.

A. Janczak. Neural network approach for identification of Hammerstein systems. *Int. J. Control*, 76(17):1749–1766, 2003.

J. Resendiz-Trejo, W. Yu, and X. Li. Support vector machine for nonlinear system on-line identification. In *3rd IEEE ICEEE*, pages 1–4, 2006.

J.L. Rojo-Álvarez, M. Martínez-Ramón, A.R. Figueiras-Vidal, de Prado-Cumplido, and A. Artés-Rodríguez. Support vector method for robust ARMA system identification. *IEEE Trans. Sig. Pro.*, 52(1):155–164, 2004.

L. San and M. Ge. An effective learning approach for nonlinear system modeling. In *Proc. IEEE Int. Symp. Intel. Control*, pages 73–77, 2004.

J. Schoukens, R. Pintelon, T. Dobrowieki, and Y. Rolain. Identification of linear systems with nonlinear distortions. *Automatica*, 41(3):491–504, March 2005.

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.

V. Vapnik. *Statisical Learning Theory*. John Wiley & Sons, New York, 1998.

X. Wang and M. Ye. Nonlinear dynamic system identification using least squares support vector machine regression. In *Proc. Int. Conf. Machine Learning & Cybern.*, pages 941–945, 2004.

X. Wang, W. Liang, X. Cai, G. Lv, C. Zhang, and H. Zhang. Application of adaptive least square support vector machines in nonlinear system identification. In *Proc. World Con. Intel. Cont. Autom.*, pages 1897–1900, 2006.