Support Vector Machine Identification of Subspace Hammerstein Models

Mujahed Al Dhaifallah and K. S. Nisar

***Abstract*—In this work a new method for identifying subspace Hammerstein systems based on Support vector machine regression is presented. It has been developed by modifying a least-square support vector machine based approach presented earlier. The new algorithm exploits the properties of generic SVM which LS-SVM based algorithm lacks. These properties are robustness in the presence of outliers and sparseness of solution. The proposed algorithm is reduced to include the least number of quadratic programming problems needed to estimate the system matrices and nonlinearity which in turn will reduce the computation complexity of the algorithm.**

***Index Terms*—Hammerstein models, subspace identification, support vector machines.**

I. Introduction

For researchers and practitioners, modeling is an essential instrument to realize and improve system dynamics [1]. In physical modeling, similarity is used to simulate the real system. For example, an analog computer may be used to build something that behaves almost like the original system. However, to model a complicated engineering system, one must use different method, such as pure mathematical representation, as an analog computer representation would be huge and complicated [2]. Such mathematical representations can be found from fundamental principles, for instance: Newton's laws, Kirchhoff's laws, conservation laws, etc. However, this approach is inappropriate for complicated systems [3]. System identification, which is the science of deriving mathematical procedures that form a suitable mathematical model of a system from available input and output data, is good modeling candidate for complex systems. It has caught the attention of researchers and practitioners for many years [4], [5]. In the last two decades, subspace identification theory [6], [7] has attracted researchers' interest because of its efficiency in 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 yield exact estimates of a systems behavior, especially, if it is limited to operate within a short region. Nevertheless, If the model is needed to represent a wider operating region, then a nonlinear model may be needed. One further step toward accuracy is to consider block structured models, sequence of static nonlinearities and dynamic linear systems. Common structures include the Hammerstein (nonlinearity, N, followed by a linear subsystem, L), the Wiener (LN), the Wiener-Hammerstein (LNL), and the Hammerstein-Wiener (NLN) systems. These structures will give a substantial progress compared to the linear estimate if the real system structure similar to this class. Some of the subspace identification methods have been extended to some block structured nonlinear models identifications. 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. The main drawback of this approach is that the nonlinearity was parameterized with a polynomial because it is easily calculated by computing a linear regression problem. Yet, this regression problem can be ill-conditioned, particularly with high order polynomials, yielding biased coefficient estimates. Another drawback is the doubtful extrapolation of polynomials approximation of hard nonlinearities (dead zone, saturation, rectification) specifically outside the boundaries, and even close to the boundaries of the training data [10]. Some of these drawbacks can be solved if the nonlinearity is estimated by spline function. Nevertheless, spline function is composed of a series of knot points which must either be selected beforehand, or considered as model parameters and involved in the (non-convex) optimization. Neural networks are another method to estimate nonlinear functions. Their excellent estimation makes them attractive. However, the necessity to state the neural network topology in terms of the number of nodes and layers, and the necessity to compute non-convex optimization make its implementation difficult. Lately, support vector machines (SVMs) and least squares support vector machines (LS-SVMs) have revealed excellent abilities in estimating

Manuscript received February 24, 2014; revised April 21, 2014. The authors would like to acknowledge the support provided by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project No. FT131015.

Mujahed Al Dhaifallah is with the Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Kingdom of Saudi Arabia (e-mail: mujahed@kfupm.edu.sa).

K. S. Nisar is with the Department of Mathematics, Salman bin Abdulaziz University, Wadi Al Dawaser, Kingdom of Saudi Arabia (e-mail: ksnisar1@gmail.com).

linear and nonlinear functions ([11], [12]).Goethals *et al*. [13] 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 model the nonlinear part of the Hammerstein system. Recently, Bako *et al*. [14] extended Goethals work to time-varying systems by using LS-SVM to recursively estimate the non-linear part of the system and ordinary least squares for recovering the linear part in state space form. The LS-SVM solution proposed in [13] lacks sparseness. Also, because the LS-SVM regression uses the least squares loss function, the existence of non-Gaussian noise or outliers may decrease the accuracy of its approximation. To solve these issues, an identification algorithm based on SVM regression was presented in [15]. Instead of the least squares cost function optimized in a LS-SVM, as used in [13], we will employ the Vapnik [11] ε−insensitive cost function, which allocates zero cost to a tube of radius ε about the solution, to improve sparseness. Due to the ε -tube, only the support vectors effect the cost function and result in non-zero Lagrange multipliers. Moreover, this permits the usage of the L-1, which is robust to non- Gaussian noise and outliers. The main drawback of the approach given in [15] is that six successive SVM problems must be computed to identify the model: the first four estimate the oblique projections which are used to compute the extended observability matrices Γ_i and Γ_{i-1} and estimates of the state, while the fifth and sixth optimization problems are used to estimate the system matrices *A*, *B*, *C* and *D*. This approach is computationally heavy. The goal of this contribution is to reduce the number of optimization problem to three which in turn will reduce the computation load by half.

II. Problem Definition

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

$$\begin{cases} x_{k+1} = Ax_k + Bf(u_k) + v_k \\ y_k = Cx_k + Df(u_k) + u_k \end{cases} \tag{1}$$

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

$$E\left\{\begin{bmatrix} v_p \\ u_p \end{bmatrix}\begin{bmatrix} v_q^{\ T} & u_q^{\ T} \end{bmatrix}\right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \quad \text{if } p = q$$

$$= 0 \quad \text{if } p \neq q \tag{2}$$

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

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

$$= \begin{bmatrix} U_{0|i-1} \\ U_{i|2i-1} \end{bmatrix} = \begin{bmatrix} U_p \\ U_f \end{bmatrix}$$

$$= \begin{bmatrix} U_{0|i} \\ U_{i+1|2i-1} \end{bmatrix} = \begin{bmatrix} U_p^{\ +} \\ U_f^{\ -} \end{bmatrix}$$

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

$$= \begin{bmatrix} Y_{0|i-1} \\ Y_{i|2i-1} \end{bmatrix} = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}$$

$$= \begin{bmatrix} Y_{0|i} \\ Y_{i+1|2i-1} \end{bmatrix} = \begin{bmatrix} Y_p^{\ +} \\ Y_f^{\ -} \end{bmatrix}$$

With *i* and *j* user defined indices such that.

III. The N4SID Algorithm for Subspace Identification of Hammerstein Systems

In this section, the subspace algorithm developed by Ivan Goethals *et al* [13] will be extended to the case where a 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 $\varepsilon$is not necessarily restricted to be zero which results in sparse solution. We will follow the development in Goethals *et al*. [13], 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

$$O_i = L_{u(:.1,i)}\Phi_f(U_p) + L_y Y_p \tag{3}$$

$$O_{i+1} = L^-_{u(:.1,i+1)}\Phi_f(U_p^+) + L_y^- Y_p^+ \tag{4}$$

where f is a nonlinear function defined on R^m and ϕ_f is defined as an operator on a block Hankel matrix such that

$$\Phi_f \begin{bmatrix} Z_1 & Z_2 & \cdots & Z_p \\ Z_2 & Z_3 & \cdots & Z_{p+1} \\ \vdots & \vdots & & \vdots \\ Z_q & Z_{q+1} & \cdots & Z_{p+q-1} \end{bmatrix} = \begin{bmatrix} f(Z_1) & f(Z_2) & \cdots & f(Z_p) \\ f(Z_2) & f(Z_3) & \cdots & f(Z_{p+1}) \\ \vdots & \vdots & & \vdots \\ f(Z_q) & f(Z_{q+1}) & & f(Z_{p+q-1}) \end{bmatrix} \in R^{q\times p} \quad (5)$$

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

$$\left(\hat{L}_u, \hat{L}_y, \hat{f}\right) = arg \min_{L_u, L_y, f} \|E\|_F^2 ,$$
$$\text{s.t. } E = \begin{bmatrix} L_u & L_y \end{bmatrix} \begin{bmatrix} \Phi_{\hat{f}}\left(\hat{U}_{0|2i-1}\right) \\ Y_p \end{bmatrix} - Y_f \quad (6)$$

$$\left(\hat{L}_u^-, \hat{L}_y^-, \hat{f}\right) = arg \min_{L_u^-, L_y^-, f} \left\|E^-\right\|_F^2$$
$$\text{s.t } E^- = \begin{bmatrix} L_u^- & L_y^- \end{bmatrix} \begin{bmatrix} \Phi_{\hat{f}}\left(\hat{U}_{0|2i-1}\right) \\ Y_p^+ \end{bmatrix} - Y_{\hat{f}} \quad (7)$$

From (6) we have

$$Y_f(s,t) = L_y(s,t)Y_p(:,t) + \sum_{h=1}^{2i} L_u(s,h) f(u_{h+t-2}) + E(s,t) \quad (8)$$

for $s = 1,2,\ldots.,il$ and $t = 1,2,\ldots..,j$. It is clear from (6), (7), (8) that *Lu* and *fk* appear cross multiplied which makes the optimization problem non convex. To overcome this problem, we apply over parameterization.

Let

$$g_{s,h} = L_u(s,h) f \quad (9)$$

hence

$$Y_f(s,t) = L_y(s,t)Y_p(:,t) + \sum_{h=1}^{2i} g_{s,h} f(u_{h+t-2}) + E(s,t) \quad (10)$$

Now, to formulate the SVM regression algorithm, let

$$g_{s,h} = \omega_{s,h}^T \varphi \quad (11)$$

Substituting (11) in (10) gives

$$Y_f(s,t) = L_y(s,t)Y_p(:,t) + \sum_{h=1}^{2i} \omega_{s,h}^T \varphi(u_{h+t-2}) + E(s,t) \quad (12)$$

As shown in [13], one should put in mind that expanding a nonlinear function as the sum of a set of nonlinear functions is not unique, for example

$$\omega_1^T \varphi_1(u) + \omega_2^T \varphi_2(u) = \left(\omega_1^T \varphi_1(u) + \delta\right) + \left(\omega_2^T \varphi_2(u) - \delta\right) \quad (13)$$

for all $\delta \in R$. Such problem can be prevented by including Centering constraints of the form

$$\sum_{t=0}^{N-1} f(u_t) = \sum_{t=0}^{N-1} \omega_{s,h}^T \varphi(u_t) = 0 \quad (14)$$

However, to apply centering constraints (14) new parameter δ_y should be added to (12) to get [13]

$$Y_f(s,t) + [1_i \otimes \delta_y](s) = L_y(s,:)(Y_p(:,t) + 1_i \otimes \delta_y) + \sum_{h=1}^{2i} \omega_{s,h}^T \varphi(u_{h+t-2}) + E(s,t) \quad (15)$$

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

$$\min_{\omega,\xi}(\omega,\xi) = \frac{1}{2}\sum_{s=1}^{i}\sum_{h=1}^{2i} \omega_{s,h}^T \omega_{s,h} + \frac{1}{2}\sum_{s=1}^{i} L_y(s,:) L_y^T(s,:) + \gamma \sum_{s=1}^{i}\sum_{t=1}^{j} \left(\xi(s,t)\right) + \xi^*(s,t) \quad (16)$$

Subject to

$$Y_f(s,t) - L_y(s,:)Y_p(:,t) - \sum_{h=1}^{2i} \omega_{s,h}^T \varphi(u_{h+t-2}) + d(s) \le \varepsilon + \xi(s,t)$$

$$L_y(s,:) - Y_p(:,t) - \sum_{h=1}^{2i} \omega_{s,h}^T \varphi(u_{h+t-2}) - Y_f(s,t) - d(s) \le \varepsilon + \xi^*(s,t)$$

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

$$\xi^*(s,t), \xi(s,t) \ge 0,$$

$$s = 1,\ldots., i,\ h = 1,\ldots., 2i \text{ and } t = 1,\ldots..,j \quad (17)$$

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

The optimization just described is the primal problem for regression. To formulate the corresponding dual problem, we have to write the Lagrangian function *L*. Then, minimize *L* with respect to the weight vector ω, *Ly*, and slack variables ξ and $\xi*$ and maximize with respect to the Lagrange multipliers. By carrying out this optimization we can write ω and *Ly* in terms of the Lagrange multipliers. Finally, we can substitute the value of *w*, *Ly* and use the so-called "kernel trick" [11], to replace the inner products with the kernel function, and simplify to get the following dual problem

$$
\begin{aligned}
&\max_{\alpha,\alpha^*,\rho} L\left(\alpha,\alpha^*,\rho\right)\\
&=\frac{1}{2}\sum_{s=1}^{i}\sum_{h=1}^{2i}\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}\rho_{s,h}\rho_{s,h}K\left(u_{tt},u_t\right)\\
&-\frac{1}{2}\sum_{s=}^{i}\sum_{h=1}^{2i}\sum_{t=1}^{j}\left(\alpha_{s,t_1}-\alpha^*_{s,t_1}\right)\left(\alpha_{s,t_1}-\alpha^*_{s,t_1}\right)\\
&\times\sum_{h=1}^{2i}\left(K\left(u_{h+t_1-2},u_{h+t\ -2}\right)\right)+\sum_{s_1=1}^{i}Y_p\left(s_1,t_1\right)Y_p\left(s_1,t\right)\\
&-\sum_{s=1}^{i}\sum_{t=1}^{j}\left(\alpha_{s,t}+\alpha^*_{s,t}\right)\varepsilon
\end{aligned}
\tag{18}
$$

subject to

$$
\begin{aligned}
&\sum_{tt=1}^{j}\left(\alpha_{s,tt}-\alpha^*_{s,tt}\right)\sum_{t=0}^{N-1}K\left(u_{h+tt-2,}u_t\right)\\
&\quad+\sum_{t=0}^{N-1}\sum_{t_2=0}^{N-1}\rho_{s,h}K\left(u_{t_2},u_t\right)=0\\
&\sum_{t=1}^{j}\left(\alpha_{s,t}-\alpha^*_{s,t}\right)\left(Y_p\left(:,t\right)\right)=0\\
&\sum_{t=1}^{j}\left(\alpha_{s,t}-\alpha^*_{s,t}\right)=0\\
&0\le\alpha_{s,t},\alpha^*_{s,t}\le\gamma\\
&s=1,\ldots.,j,h=1,\ldots..,2i,t=1,\ldots..,j
\end{aligned}
\tag{19}
$$

Then L_y is given by

$$
L_y\left(s,s_1\right)=\sum_{t=1}^{j}\left(\alpha_{s,t}-\alpha^*_{s,t}\right)Y_p\left(s_1,t\right) \tag{20}
$$

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

$$
\bar{d}=\text{mean}\left(\eta-\left(YA\right)^T L_y\right) \tag{21}
$$

$$
\delta_y=\left(M^TM\right)^{-1}M^T\bar{d}^T \tag{22}
$$

$$
M=\left(1_i\otimes I_1-L_y\left(1_i\otimes I_1\right)\right) \tag{23}
$$

$$
\begin{aligned}
&\eta\left(\mathrm{s},t\right)=\mathrm{Y_f}\left(s,t\right)-\sum_{h=1}^{2i}\left(\sum_{tt=1}^{j}\left(\alpha_{s,tt}-\alpha^*_{s,tt}\right)K_0\left(tt,h\right)\right.\\
&\left.+\rho_{s,h}\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}K\left(u_{tt},u_t\right)\right)
\end{aligned}
\tag{24}
$$

$$
\left(YA\right)=\begin{bmatrix} y_{r-1} & y_r & \cdots & y_{N-1}\\ y_{r-2} & y_{r-1} & \cdots & y_{N-2}\\ \vdots & \vdots & & \vdots\\ y_{r-n} & y_{r-n+1} & \cdots & y_{N\text{-}n}\end{bmatrix} \tag{25}
$$

Recalling (4)

$$
O_i=L_u\left(:,1:i\right)\Phi_f\left(U_p\right)+L_y\left(Y_p\left(:,t\right)-\left(1_i1_j^T\right)\otimes\delta_y\right) \tag{26}
$$

$$
\begin{aligned}
&O_i\left(s,t\right)=\sum_{t_1=1}^{j}\left(\alpha_{s,t_1}-\alpha^*_{s,t_1}\right)\left(\mathrm{KA}\right)_p\left(t_1,t\right)\\
&+\sum_{h=1}^{2i}\rho_{s,\mathrm{h}}\left(\mathrm{SA}\right)_p\left(h,t\right)+\mathrm{L_y}\left(\mathrm{Y}_p\left(:,t\right)-\left(1_i1_j^T\right)\otimes\delta_y\right)
\end{aligned}
\tag{27}
$$

where

$$
\left(KA\right)_p\left(t_1,t\right)=\sum_{h=1}^{i}K\left(u_{h+t_1-2},u_{h+t-2}\right)
$$

$$
\left(SA\right)_p\left(h,t\right)=\sum_{t_2=1}^{N-1}K\left(u_{t_{21}},u_{h+t-2}\right)
$$

The same approach can be followed to calculate O_{i+1}

$$
\begin{aligned}
&\max_{\alpha^-,\alpha^{*-},\rho^-} L\left(\alpha^-,\alpha^{*-},\rho^-\right)\\
&=\frac{1}{2}\sum_{s=1}^{i-1}\sum_{h=1}^{2i}\sum_{t=0}^{N-1}\sum_{tt=0}^{N-1}\rho^-_{s,h}\rho^-_{s,h}K\left(u_{tt},u_t\right)\\
&-\sum_{s=1}^{i-1}\sum_{t_1}^{j}\sum_{t=1}^{j}\left(\alpha^-_{s,t_1}-\alpha^{*-}_{s,t_1}\right)\left(\alpha^-_{s,t}-\alpha^{*-}_{s,t}\right)\\
&\times\left(\sum_{h=1}^{2i}K\left(u_{h+t_1-2},u_{h+t-2}\right)+\sum_{s_1}^{i+1}Y^+_p\left(s_1,t\right)Y^+_p\left(s_1,t_1\right)\right)\\
&+\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*-}_{s,t}\right)Y_{f^-}\left(s,t\right)\\
&-\sum_{s=1}^{i-1}\sum_{t=1}^{j}\left(\alpha^-_{s,t}+\alpha^{*-}_{s,t}\right)\varepsilon
\end{aligned}
\tag{28}
$$

Subject to

$$
\begin{aligned}
&\sum_{tt=1}^{j}\left(\alpha^-_{s,tt}-\alpha^{*-}_{s,tt}\right)\sum_{t=0}^{N-1}K\left(u_{h+tt-2},u_t\right)\\
&+\sum_{t=0}^{N-1}\sum_{t_2=0}^{N-1}\rho^-_{s,h}K\left(u_{t_2},u_t\right)=0\\
&\sum_{t=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*-}_{s,t}\right)\left(Y^+_p\left(:,t\right)\right)=0_i\\
&\left(\alpha^-_{s,t}-\alpha^{*-}_{s,t}\right)=0\\
&\sum_{t=1}^{j}0\le\alpha^-_{s,t},\alpha^{*-}_{s,t}\le\gamma\\
&s=1,\ldots.,j-1,h=1,\ldots.,2i,t=1,\ldots.,j
\end{aligned}
\tag{29}
$$

Then L_y^- is given by

$$
L^-_y\left(s,s_1\right)=\sum_{t=1}^{j}\left(\alpha^-_{Ls,t}-\alpha^{*-}_{Ls,t}\right)Y^+_p\left(s_1,t\right) \tag{30}
$$

Recalling (4)

$$
O_{i+1}=L^-_u\left(:,1:i+1\right)\Phi_f\left(U^+_p\right)+L^-_y\left(Y^+_p\left(:,t\right)-\left(1_i1_j^T\right)\otimes\delta_y\right) \tag{31}
$$

$$
\begin{aligned}
&O_{i+1}\left(s,t\right)=\sum_{t_1=1}^{j}\left(\alpha^-_{s,t}-\alpha^{*-}_{s,t_1}\right)\left(KA\right)^+_p\left(t_1,t\right)+\sum_{h=1}^{2i}\rho^-_{s,h}\left(SA\right)^-_p\left(h,t\right)\\
&\qquad+L^-_y\left(Y^+_p\left(:,t\right)-\left(1_i1_j^T\right)\otimes\delta_y\right)
\end{aligned}
\tag{32}
$$

where

$$
\left(KA\right)^+_p\left(t_1,t\right)=\sum_{h=1}^{i+1}K\left(u_{h+t_1-2},u_{h+t-2}\right)
$$

$$
\left(SA\right)^-_p\left(h,t\right)=\sum_{t_2=0}^{N-1}K\left(u_{t_{21}},u_{h+t-2}\right)
$$

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

$$\Gamma_i = U_1 S_1^{1/2}, \Gamma_{i-1} = \Gamma_i(1:i-1,:) \tag{33}$$

where U_1 and $S1$ are obtained by partitioning the SVD of O_i as follows

$$O_i = USV^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{34}$$

Recall that the system order can be determined by inspecting the singular values of O_i. Now, estimates for the state sequences can be computed from

$$\begin{cases} \bar{X}_i = \Gamma_i^+ O_i \\ \bar{X}_{i+1} = \Gamma_{i-1}^+ O_{i+1} \end{cases} \tag{35}$$

IV. Extraction of the System Matrices and the Nonlinearity

Extraction of the System Matrices and the Nonlinearity

$$\begin{bmatrix} \bar{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{X}_i \\ f(U_{i|i}) \end{bmatrix} + \begin{bmatrix} \rho_v \\ \rho_u \end{bmatrix} \tag{36}$$

In what follows, a SVM regression problem will be formulated to identify (38). Representing

$$(XA)_{i+1} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix},\ \Theta_{AC} = \begin{bmatrix} A \\ C \end{bmatrix},\ \Theta_{BD} = \begin{bmatrix} B \\ D \end{bmatrix}$$

results in

$$(XA)_{i+1} = \Theta_{AC}\ \mathrm{X_i} + \Theta_{BD}\ \Phi_f(U_{i|i}) + \mathrm{E} \tag{37}$$

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

$$(XA)_{i+1} = \Theta_{AC}\ X_i + \begin{bmatrix} \omega_1^T \\ \omega_2^T \\ \vdots \\ \omega_{n+1}^T \end{bmatrix} \Phi_{\varphi(U_{i|i})} + E \tag{38}$$

Now, The SVM primal problem is

$$\min_{\omega,\xi,\xi^*} \frac{1}{2}\sum_{s=1}^{n+1} \omega_s^T \omega_s \frac{1}{2}\sum_{s_1=1}^{n+1}\sum_{s_2}^{n} \Theta^2{}_{\mathrm{AC}}(s_1,s_2) + \gamma \sum_{s=1}^{n+1}\sum_{t=1}^{j}\left(\xi(s,t)+\xi^*(s,t)\right) \tag{39}$$

subject to

$$\begin{gathered} (XA)_{i+1}(s,t) - \Theta_{AC}(s,:)X_i(:,t) + \omega_s^T \varphi(u_{i+t-2}) \le \varepsilon + \xi(s,t) \\ \Theta_{AC}(s,:)X_i(:,t) + \omega_s^T \varphi(u_{i+t-2}) - \mathrm{X}_{i+1}(s,t) \le \\ \varepsilon + \xi^*(s,t), \xi(s,t) \ge 0 \\ s = 1,\ldots\ldots, n+1 \text{ and } t = 1,\ldots.,j \end{gathered} \tag{40}$$

By defining the Lagrangian, eliminating the primal variables $\omega_s, \Theta_{AC}(s_1,s_2), \xi(s,t)$ and $\xi^*(s,t)$ using the kernel trick and, simplification, the dual problem can be shown to be

$$\begin{aligned} &\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{s=1}^{n+1}\sum_{t=1}^{j}\sum_{t_1=0}^{j}\left(\alpha_{s,t}-\alpha_{s,t}^*\right)\left(\alpha_{s,t_1}-\alpha_{s,t_1}^*\right) \\ &\times\left\{K\left(u_{h+t-2},u_{h+t_1-2}\right)+\sum_{s_1=1}^{n} X_i(s_2,t)X_i(s_2,t_2)\right\} \\ &+\sum_{s=}^{n}\sum_{t=1}^{j}\left(\alpha_{s,t}-\alpha_{s,t}^*\right)(XA)_{i+1}(s,t) \\ &+\sum_{s=}^{n}\sum_{t=1}^{j}\left(\alpha_{s,t}+\alpha_{s,t}^*\right)\varepsilon \end{aligned} \tag{41}$$

subject to

$$\begin{aligned} &\sum_{t=1}^{j}\left(\alpha_{s,t}-\alpha_{s,t}^*\right)X_i(:,t) \\ &0 \le \alpha_{s,t}, \alpha_{s,t}^* \le \gamma \\ &s = 1,\ldots.,n+1, t = 1,\ldots.,j \end{aligned} \tag{42}$$

then Θ_{AC} is given by

$$\begin{gathered} \Theta_{AC}(s_1,s_2) = \sum_{t=1}^{j}\left(\alpha\Theta_{AC\ s_1,t} - \alpha^*\Theta_{AC\ s_1,t}\right)\bar{X}(s_2,t) \\ s_1 = 1,\ldots.,n+1, s_2 = 1,\ldots.,n \end{gathered} \tag{43}$$

To extract B and D in Θ_{BD} and the nonlinearity f, we use the solution presented in [13], which involves using the SVD of a m by n matrix. Then, using the training input data $[u_1 \ \cdots \ u_N]$ and the estimated values of the nonlinearity responses, a SVM can be trained to approximate the nonlinear part of the Hammerstein system.

V. Algorithm

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

1) Obtain estimates for α, $\alpha*$, ρ by solving (18), (19).
2) Compute $\boldsymbol{L}y$, $\bar{d}$ and δy using (20) and (24).
3) Find estimates for the oblique projection Oi from (27).
4) Obtain estimates for $\alpha-$, $\alpha*-$, $\rho-$ by solving (28), (29).
5) Compute, $\bar{L}_y$ using (30).
6) Find estimates for the oblique projection $Oi+1$ from (32).
7) Calculate the SVD of Oi, and determine the order by inspecting the singular values and partition the SVD accordingly to obtain U_1 and S_1.
8) Compute the extended observability matrices Γi and Γ_{i-1} from (33).
9) Find estimates for the state using (35).

10) Obtain estimates for α, $\alpha*$ by solving (41), (42).
11) Obtain estimates for Θ_{AC} using (43).
12) Obtain estimates for B and D and the nonlinearity f from a rank-m approximation presented in [13].
13) Use the input sequence $[u_1, u_2, \cdots u_{n-1}]$ and the estimates of response of the nonlinearity to this input $[f(u_1), f(u_2), \cdots f(u_{n-1})]$ to train a SVM to approximate the nonlinear function f.

VI. Illustrative Example

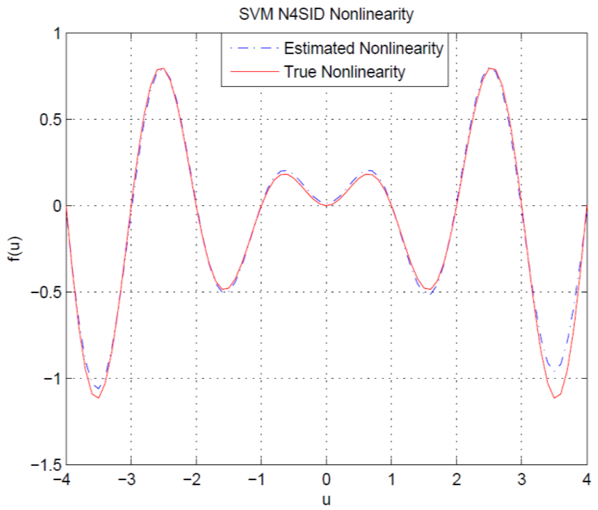

Fig. 1. True nonlinearity, and mean of the SVM, with ε=0.001 estimate with statistics estimated from a hundred trial Monte-Carlo simulation.

TABLE I: Comparison between Computation Speed of Algorithm with 5 Quadratic Programs and Algorithm with 3 Quadratic Programs

Algorithm	Computation time
5 quadratic programs	107.8
3 quadratic programs	43.8

To compare performance of the proposed algorithm to the algorithm presented in [15], the following system which belongs to the Hammerstein class of models is considered:

$$A(Z)(y+v) = B(z) f(u) \tag{44}$$

where

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

such that be the static nonlinearity. 1000 points white Gaussian noise with zero mean and variance 2 was generated and fed into this system. The output noise sequence $\{v_t\}_{t=0}^{N-1}$ was a zero mean Gaussian white noise with signal to noise ratio equals to 10. The hyper-parameters were chosen to be $\sigma = 1, \gamma = 1000, \gamma_{BD} = 10$. It is obvious from Fig. 1 that the nonlinearity was estimated very well. Table I shows that the suggested algorithm is faster than the algorithm presented in [15] by almost 60% which reflects the effect of reducing the number of quadratic programs needed to identify the system from 5 to 3.

VII. Conclusion

In this paper, the subspace algorithm developed in previous paper has been improved. It is clear from the simulation that the computation time has been reduced by reducing the number of quadratic programming problems used to identify the system. For future work one might apply the algorithm to real data.

References

[1] L. Ljung, *System Identification: Theory for the User*, Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[2] T. Pelt and D. Bernstein, "Non-linear system identification using Hammerstein and non-linear feedback models with piecewise linear static maps," *Int. J. Control*, vol. 74, pp.1807-1823, August 2001.

[3] H. Unbehauen, "System identification methods using parameter estimation/a survey," *Annual Review in Automatic Programming*, vol. 12, pp. 69-81, January 1985.

[4] N. K. Sinha, "System identification: from frequency response to soft computing," in *Proc. IEEE International Conference on Industrial Technology*, 2000, vol. 1, pp. 76-80.

[5] K. J. Astrom and P. Eykhoff, "System identification-A survey," *Automatica*, vol. 7, issue 2, pp. 123-62, 1971.

[6] M. Verhaegen and P. Dewilde, "Subspace model identification part 2. Analysis of the elementary output- error state-space model identification algorithm," *Int. J. Control*, vol. 56, issue 5, pp. 1211-1241, 1992.

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

[8] W. Larimore, "Canonical variate analysis in identification, filtering, and adaptive control," in *Proc. the IEEE Conference on Decision and Control*, 1990, vol. 2, pp. 596-604.

[9] M. Verhaegen and D. T. Westwick, "Identifying MIMO hammerstein systems in the context of subspace model identification methods," *Int. J. Control*, vol. 63, issue 2, pp. 331-349, 1996.

[10] D. Westwick and R. Kearney, *Identification of Nonlinear Physiological Systems*, Piscataway, NJ: John Wiley & Sons, August 2003.

[11] V. N. Vapnik, *Statistical Learning Theory*, NY: John Wiley & Sons, Inc., September 1998.

[12] J. Suykens, T. De Braanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, Singapore: World Scientific Pub. Co., 2002.

[13] I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. De Moor, "Subspace identification of Hammerstein systems using least squares supports vector machines," *IEEE Transactions on Automatic Control*, vol. 50, issue 10, pp. 1509-1519, October 2005.

[14] L. Bako, G. Mercère, S. Lecoeuche, and M. Lovera, "Recursive subspace identification of Hammerstein models based on least squares support vector machines," *IET Control Theory and Applications*, vol. 3, issue 9, pp. 1209-1216, January 2009.

[15] M. Al Dhaifallah, "Subspace identification of Hammerstein models using support vector machines," in *Proc. 8th International Multi-Conference on Systems, Signals & Devices (SSD11)*, Tunisia, 2011.

Mujahed Al-Dahifallah received the B.A.Sc. and M.Sc.E. degrees in systems engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and the Ph.D. degree in electrical and computer engineering from the University of Calgary, Calgary, AB, Canada.

He has been an assistant professor of systems engineering with the King Fahd University of Petroleum and Minerals, Saudi Arabia since 2009. One of his recently published papers is Mujahed Al-Dhaifllah and David T. Westwick, "Identification of Auto-Regressive Exogenous Hammerstein Models Based on Support Vector Machine regression", IEEE Transactions

On Control Systems Technology, Vol. 21, No. 6, November 2013. His current research interests include nonlinear systems identification.

K. S. Nisar was born in Kerala, India on May 20, 1982. He received M.Sc. degree in mathematics in 2005, M.Phil. degree in applied mathematics in 2008 and Ph.D. degree for his research with the Department of Applied Mathematics, Faculty of Engineering, Aligarh Muslim University in 2011. His main research in the field of special functions, umbral calculus and statistical convergence.

He has been working as an assistant professor in Salman bin Abdulaziz University, Wadi Al Dawaser, Saudi Arabia. Some of his recently published papers are: 1) M. Mursaleen, Asif Khan, H. M Srivastava and K. S. Nisar: Operators constructed by means of q-Lagrange polynomials and A-statistical approximation, Applied Mathematics and Computation, Vol. 219, Issue 12, 2013. 2) Hassan Yousif Ahmed and K. S. Nisar, Diagonal Eigenvalue Unity (DEU) code for spectral amplitude coding-optical code division multiple access, Optical Fiber Technology, vol.19, pp: 335-347, 2013. 3). His current research interests are: system identification and SAC-OCDMA code networks.