Fixed-Size LS-SVM Applied to the Wiener-Hammerstein Benchmark

K. De Brabanter [*] **, Ph. Dreesen** [*] **, P. Karsmakers** [*,***]
K. Pelckmans [*] **, J. De Brabanter** [*,**] **, J.A.K. Suykens** [*]
B. De Moor [*]

[*] *Department of Electrical Engineering ESAT-SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10 B-3001 Leuven, Belgium*
{kris.debrabanter,johan.suykens}@esat.kuleuven.be
[**] *Hogeschool KaHo Sint-Lieven (Associatie K.U.Leuven), Departement Industrieel Ingenieur, B-9000 Gent*
[***] *K.H.Kempen (Associatie K.U.Leuven), Dep. IBW, B-2440 Geel*

Abstract: This paper reports on the application of Fixed-Size Least Squares Support Vector Machines (FS-LSSVM) for the identification of the SYSID 2009 Wiener-Hammerstein benchmark data set. The FS-LSSVM is a modification of the standard Support Vector Machine and Least Squares Support Vector Machine (LS-SVM) designed to handle very large data sets. This approach is taken to estimate a nonlinear black-box (NARX) model from given input/output measurements. We indicate how to tune this approach to the specific case study. We obtain a best root mean squared error of 4.7×10^{-3} on simulation of the predefined test set.

Keywords: Black-box; Identification; Nonlinear Models; NARX; Kernel Methods

1. INTRODUCTION

System identification in general is concerned with the identification of an appropriate model from observations of input/output signals [Söderström and Stoica, 1989]. Black-box modeling strategies try to do so when little a priori knowledge is available. In particular, black-box modeling strategies are concerned with modeling nonlinear relationships using different approaches including artificial neural networks, orthogonal basis functions [Sjöberg et al., 1995, Ljung, 1999] or smoothing techniques [Greblicki and Pawlak, 2008]. It is a classical result that the identification of a nonlinear ARX model can be written as a static regression problem by a suitable definition of the regression covariates as lagged observations of the input/output signals.

Recently, a novel technique for nonlinear modeling became increasingly popular in different research fields, namely the Support Vector Machine (SVM), both for classification as regression [Vapnik, 1999, Suykens et al., 2002]. The key ingredients of this technique are (1) the use of convex optimization theory (with primal-dual interpretation) and the integration of the model training in a convex program. Specifically, SVMs boil down to solving a convex Quadratic Program (QP), which can be solved efficiently; (2) The extension of linear techniques to nonlinear modeling using Mercer kernels. Least Squares Support Vector Machines (LS-SVMs) are a modification of the original SVM where a least squares norm replaces the role of the loss of SVMs [Suykens et al., 2002] and the inequality constraints are replaced by equality constraints. It is empirically found that this method yields a similar performance compared to the classical SVM on classification problems, and often outperforms SVM in the case of regression. Further primal-dual formulations have been given, in the context of LS-SVMs, for kernel Fisher Discriminant Analysis (KFDA), kernel Partial Least Squares (KPLS), kernel Principal Component Analysis (KPCA) and kernel Canonical Correlation Analysis (KCCA) [Suykens et al., 2002]. Both kernel techniques (SVMs and LS-SVMs) map the input data into a high dimensional feature space (possibly infinite dimensional). A linear model is then built in this high dimensional feature space. By using Mercer's theorem, a positive definite kernel and the primal-dual interpretation of convex programs, no explicit computation of the feature map is needed. The LS-SVM learning problem boils down to solving a linear system in the dual space. It is in general easier to solve a linear system than a QP, but a drawback is the loss of sparseness of the representation of the estimate. In the context of LS-SVMs, such sparseness can be introduced by sequentially pruning the support vector (SV) spectrum [Suykens et al., 2002]. Kernel based methods require the determination of tuning parameters, e.g. a regularization constant and a kernel bandwidth. A widely used technique to obtain these parameters is cross-validation (CV).

Although the LS-SVM is mostly solved in its dual form, the problem can also be solved in the primal space by estimating a finite dimensional feature map. It is possible to compute a sparse approximation by using only a subsample of selected support vectors from the entire data set. The advantage of this approach is that one can easily handle nonlinear estimation problems up to 1.000.000 samples [Suykens et al., 2002]. Successful applications of Fixed-Size Least Squares Support Vector Machines (FS-LSSVM) to the field of system identification can be found in [Espinoza et al., 2004] and [Espinoza et al., 2007]. In this paper we

illustrate the application of the optimized extension of FS-LSSVM as presented in [De Brabanter et al., 2008].

In this paper we apply FS-LSSVM to the SYSID 2009 Wiener-Hammerstein benchmark data set and compare this performance with linear ARX and nonlinear ARX using Multi-layer Perceptrons (MLPs). These last two model structures are obtained by using the Matlab System Identification Toolbox. This paper is organized as follows. The basic description of LS-SVM is presented in Section 2. In Section 3 the explicit computation of the feature map is explained and the methodology to estimate in primal space is described. Section 4 presents the problem and describes the overall setting for the working procedure. The results are reported in Section 5.

2. KERNEL BASED FUNCTION ESTIMATION IN THE DUAL SPACE

The standard framework for LS-SVM is based on a primal-dual formulation. Given a training data set $\mathcal{D}_n = \{(X_k, Y_k) : X_k \in \mathbb{R}^d, Y_k \in \mathbb{R}; k = 1, \dots n\}$ of size n drawn i.i.d. from an unknown distribution F_{XY} according to

$$Y_k = m(X_k) + e_k, \qquad k = 1, \dots, n, \tag{1}$$

where $e_k \in \mathbb{R}$ are assumed to be i.i.d. random errors with $\mathrm{E}[e_k|X = X_k] = 0$, $\mathrm{Var}[e_k] = \sigma^2 < \infty$, $g \in C^z(\mathbb{R})$ with $z \geq 2$, is an unknown real-valued smooth function and $\mathrm{E}[Y_k|X = X_k] = m(X_k)$. The optimization problem of finding the vector w and $b \in \mathbb{R}$ for regression can be formulated as follows [Suykens et al., 2002]

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w, e) &= \tfrac{1}{2} w^T w + \tfrac{\gamma}{2} \sum_{k=1}^{n} e_k^2 \\ \text{s.t.} \quad Y_k &= w^T \varphi(X_k) + b + e_k, \quad k = 1, \dots, n, \end{aligned} \tag{2}$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is the feature map to the high dimensional feature space [Vapnik, 1999] and unknowns $w, b \in \mathbb{R}$. However, we do not need to evaluate w and $\varphi(\cdot)$ explicitly. By using Lagrange multipliers, the solution of (2) can be obtained by considering the Karush-Kuhn-Tucker (KKT) conditions for optimality and solve the dual problem. The result is given by the following linear system [Suykens et al., 2002]

$$\left(\begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega + \frac{1}{\gamma} I_n \end{array}\right) \left(\begin{array}{c} b \\ \hline \alpha \end{array}\right) = \left(\begin{array}{c} 0 \\ \hline Y \end{array}\right), \tag{3}$$

with $Y = (Y_1, \dots, Y_n)^T$, $1_n = (1, \dots, 1)^T$, $\alpha = (\alpha_1, \dots, \alpha_n)^T$ and $\Omega_{kl} = \varphi(X_k)^T \varphi(X_l) = K(X_k, X_l)$, with $K(X_k, X_l)$ a positive definite kernel $(k, l = 1, \dots, n)$. According to Mercer's theorem, the resulting LS-SVM model for function estimation becomes

$$\hat{m}(x) = \sum_{k=1}^{n} \hat{\alpha}_k K(x, X_k) + \hat{b}, \tag{4}$$

For $K(X_k, X_l)$ there are usually the following choices:

- linear kernel: $K(X_k, X_l) = X_k^T X_l$,
- polynomial kernel of degree k with $c \geq 0$: $K(X_k, X_l) = (X_k^T X_l + c)^k$,
- radial basis function with bandwidth σ: $K(X_k, X_l) = \exp\left(-\|X_k - X_l\|_2^2/\sigma^2\right)$.

The training of the LS-SVM model involves an optimal selection of these tuning parameters e.g. σ and γ, which can be done using cross-validation [Burman, 1989], bootstrap methods [Davison and Hinkley, 2003], plug-ins [Härdle, 1989] or complexity criteria (e.g. Akaike's Information Criterion [Akaike, 1973], Vapnik-Chervonenkis dimension [Vapnik, 1999] and others).

3. ESTIMATION IN PRIMAL SPACE

When working in the primal space we need an explicit approximation for the feature map φ. Only then we can use techniques such as ridge regression for parametric estimation in the primal space.

3.1 Approximation of the Feature Map

Approximation to the feature map φ can be obtained by means of an eigenvalue decomposition of the kernel matrix Ω with entries $K(X_k, X_l)$. The Fredholm integral equation of the first kind defines the eigenvalues and eigenfunctions of the kernel function, i.e.

$$\int_C K(x, x_j)\phi_i(x) dF_X(x) = \lambda_i \phi_i(x_j), \tag{5}$$

where F_X is the unknown marginal distribution, λ_i and ϕ_i are the eigenvalues and eigenfunctions respectively. Given a data set $\{(X_k, Y_k) : X_k \in \mathbb{R}^d, Y_k \in \mathbb{R}; k = 1, \dots n\}$, the Nyström method [Nyström, 1930] approximates the integral by means of the sample average and determines an approximation of ϕ_i. This leads to the following eigenvalue problem [Williams and Seeger, 2001]

$$\frac{1}{n} \sum_{k=1}^{n} K(X_k, X_j) u_i(X_k) = \lambda_i^{(s)} u_i(X_j), \tag{6}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem can be approximated by the sample eigenvalue $\lambda_i^{(s)}$ and eigenvectors u_i as

$$\hat{\lambda}_i = \frac{1}{n} \lambda_i^{(s)} \qquad \text{and} \qquad \hat{\phi}_i = \sqrt{n} u_i. \tag{7}$$

Based on this approximation, an explicit expression for the ith entry of the approximated feature map $\hat{\varphi}(X) = (\hat{\varphi}_1(X), \hat{\varphi}_2(X), \dots, \hat{\varphi}_m(X))^T$ with $\hat{\varphi}_i : \mathbb{R}^d \to \mathbb{R}$ is given by

$$\begin{aligned} \hat{\varphi}_i(x) &= \sqrt{\lambda_i} \hat{\phi}_i(X) \\ &= \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{n} u_{ki} K(X_k, x), \end{aligned} \tag{8}$$

where u_{ki} represents the k-th element of the i-th eigenvector. This finite dimensional approximation $\hat{\varphi}(x)$ can be used in the primal problem (2) to finally estimate $w \in \mathbb{R}^{n_h}$ and $b \in \mathbb{R}$.

3.2 Imposing Sparseness and Subset Selection

It is important to emphasize that the use of the entire training sample of size n to compute the approximation of φ will yield at most n components. Each of these components can be calculated by means of (8). Considering large scale problems, it has been motivated by Suykens et al. [2002] to choose a working set of fixed size m ($m \leq n$

and typically $m \ll n$) where the value m is related to the Nyström subsample and an entropy based subset selection. In order to make a more suitable selection of the subsample instead of a random selection, an entropy based criterion can be used. Indeed, we select the m support vectors which maximize Rényi's entropy H_{Rq} of order q defined as

$$H_{Rq} = \frac{1}{1-q} \log \int p(x)^q \, dx, \tag{9}$$

with $q > 1$, $q \neq 0$ and p the density of the selected support vectors. Further in this paper we set $q = 2$, which is also called quadratic Rényi entropy H_{R2}. The quadratic Rényi entropy can be approximated [Girolami, 2002] by using

$$\int \hat{p}(x)^2 \, dx = \frac{1}{m^2} 1_m^T \Omega 1_m, \tag{10}$$

where $1_m = (1, \ldots, 1)^T$ and $\Omega_{kl} = K(X_k, X_l)$. By using such a criterion we can be sure that the selected subsample is well spread over the entire data region. Hence, the subsample will not be concentrated in a certain area of the entire data set. Note that the positive definiteness property is not required here for Ω, since entropy estimation is related to density estimation.

3.3 Estimation Technique

Once the feature map is estimated by using (8) (either using the full sample or using a sparse approximation based on a subsample), the model can be estimated in primal space. Consider the m-dimensional approximation, based on the subsample, of the feature map φ given by

$$\hat{\varphi}(X) = (\hat{\varphi}_1(X), \hat{\varphi}_2(X), \ldots, \hat{\varphi}_m(X))^T. \tag{11}$$

The solution to (2), using (11), can be written as the following linear system

$$\left(\hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma} \right) \begin{pmatrix} w \\ b \end{pmatrix} = \hat{\Phi}_e^T Y, \tag{12}$$

where $\hat{\Phi}_e$ is the $n \times (m+1)$ extended feature matrix

$$\hat{\Phi}_e = \begin{pmatrix} \hat{\varphi}_1(X_1) & \cdots & \hat{\varphi}_m(X_1) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \hat{\varphi}_1(X_n) & \cdots & \hat{\varphi}_m(X_n) & 1 \end{pmatrix} \tag{13}$$

and I_{m+1} the $(m+1) \times (m+1)$ identity matrix. The complexity of calculating the feature map and solving the linear system (12) is of order $\mathcal{O}(m^3 + m^2 n)$.

4. IMPLEMENTATION FOR THE WIENER-HAMMERSTEIN BENCHMARK

The definition of the training and validation using the data set and the predefined accuracy measurements are described in this section.

4.1 Model Structure

We use FS-LSSVM in the primal space. For this method different number of support vectors are selected. All subsamples are selected by maximizing the quadratic Rényi entropy criterion.

The general model structure is a NARX of the form

$$y_t = f(y_{t-1}, \ldots, y_{t-p}; u_{t-1}, \ldots, u_{t-p}) + e_t, \tag{14}$$

where p denotes the order of the NARX model (number of lags). The number of lags are determined via 10-fold cross-validation (CV).

4.2 Data Description and Training Procedure

The data consists of samples for the input u_i and the outputs y_i, with $i = 1, \ldots, 188.000$. A plot of the inputs and outputs is given in Figure 1. Next, the strategy for the using the data in terms of training and testing will be outlined. This goes as follows:

- Training + validation sample: from data point 1 to data point 100.000. Using 10-fold CV, models are repeatedly (10 times) estimated using 90% of the data and validated on the remaining 10%. Two approaches will be used here i.e CV on a one-step-ahead-basis (CV-RMSE$_1$) and CV based on simulating the estimated model (CV-RMSE$_{\text{sim}}$). The mean squared error (MSE) for a one-step-ahead prediction/simulation can be computed using this validation sample. The number of lags p are determined by the lowest value of the MSE of the cross-validation function.

- Test sample: from data points 100.001 to data points 188.000. After defining the optimal lags p and optimal tuning parameters γ and σ (in case of RBF kernel), the prediction on the test set can be done. In this paper, an iterative prediction is computed for the entire test set. This is done by using each time the past predictions as inputs while using the estimated model in simulation mode.

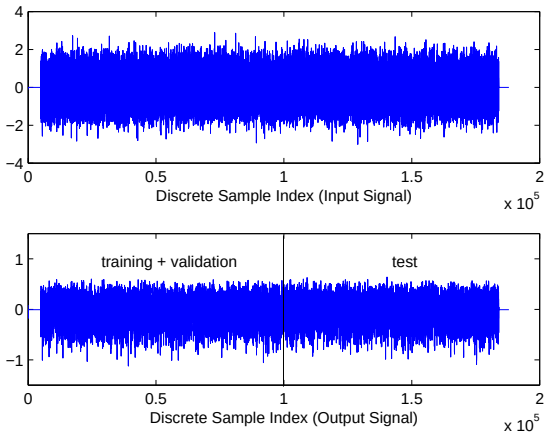

Fig. 1. Available data for the Wiener-Hammerstein identification problem. The zones for training + validation (estimation set) and test are indicated in the output series.

4.3 Finding the Optimal Tuning Parameters

The FS-LSSVM requires the evaluation of a kernel matrix $\Omega_{ij} = K(X_i, X_j)$, where K is a positive definite kernel. The tuning parameters of the kernels together with the regularization constant γ are tuned via 10-fold CV. In order to find the optimal tuning parameters in the non-smooth CV surface we used Coupled Simulated Annealing with variance control [Xavier de Souza et al., 2006] combined with a gridsearch [De Brabanter et al., 2008]. Using such a procedure results in more accurate tuning parameters and hence better performance.

5. RESULTS

In this Section the main results for the iterative prediction obtained with FS-LSSVM, ARX and MLP-NARX (nonlinear ARX) are given together with some intermediate results. In addition, the best models are combined with an ensemble method.

5.1 Estimation and Model Selection

Using the above described training + validation scheme (10-fold CV), we start checking different lag orders and tuning parameters. Each time the model is repeatedly estimated using the training set (90% of the training data) and validated using the remaining 10%. This is done on a one-step-ahead basis and simulation basis. The best model is selected based on the lowest MSE on cross-validation (CV-MSE$_1$ or CV-MSE$_{\text{sim}}$).

Consider a linear ARX with varying input and output lags. The model order is determined by 10-fold CV (CV-MSE$_1$). Figure 2 shows the CV-MSE$_1$ obtained for lags varying from 1 to 40. Table 1 shows the best results in RMSE on

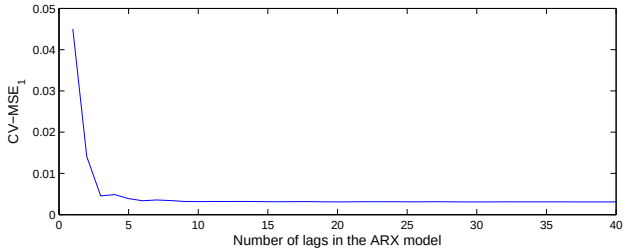

Fig. 2. The error on cross-validation (CV-MSE$_1$) using a linear ARX model with increasing number of lags.

cross-validation (one-step-ahead based (CV-RMSE$_1$) and simulation based CV-RMSE$_{\text{sim}}$) obtained for each of the techniques. NARX is a nonlinear ARX model obtained with the Matlab System Identification Toolbox. The lags for MLP-NARX are found by validation on a single set. The nonlinearity was modeled with an MLP and a sigmoid activation function with 10 hidden neurons. Due to the use of a single validation set the lags for MLP-NARX differ from the rest. For the FS-LSSVM three kernel types are reported i.e. RBF, polynomial and linear. All three techniques make use of the complete training data set of 100.000 data points. All RMSE figures are expressed in the original units of the data.

From the results in Table 1, it is clear that the FS-LSSVM using the RBF kernel outperforms the others. The linear ARX is unable to capture the nonlinearity in the data resulting in a lower performance on cross-validation RMSE (up to 2 orders of magnitude). Although two nonlinear techniques (NARX and FS-LSSVM) are used, their performances are quite different.

The effect of varying numbers of selected support vectors m on both cross-validation (RMSE) techniques are reported in Table 2 for the FS-LSSVM with RBF kernel. The best performance is bold faced in Table 2. The total number of support vectors is set to $m = 5000$.

The position of the selected support vectors (quadratic Rényi entropy criterion) is shown according to the

Table 1. Best models based on cross-validation RMSE. MLP-NARX is a nonlinear ARX model obtained with the Matlab SYSID Toolbox. Two types of CV are displayed: CV based on one-step-ahead (CV-RMSE$_1$) and simulation (CV-RMSE$_{\text{sim}}$)

Method	Kernel	lags	CV-RMSE$_1$	CV-RMSE$_{\text{sim}}$
ARX	-	10	5.67×10^{-2}	5.66×10^{-2}
MLP-NARX	-	11	7.62×10^{-4}	2.15×10^{-2}
FS-LSSVM	Lin	10	8.64×10^{-4}	4.51×10^{-2}
	Poly	10	5.63×10^{-4}	5.87×10^{-3}
	RBF	10	4.77×10^{-4}	4.81×10^{-3}

Table 2. Effect of different numbers of support vectors m on the performance (CV-RMSE$_1$ and CV-RMSE$_{\text{sim}}$) of the FS-LSSVM estimator with RBF kernel. The chosen number of support vectors is bold faced.

m	CV-RMSE$_1$	CV-RMSE$_{\text{sim}}$
100	5.82×10^{-4}	2.14×10^{-2}
400	5.36×10^{-4}	7.85×10^{-3}
600	5.13×10^{-4}	6.76×10^{-3}
800	5.05×10^{-4}	5.87×10^{-3}
1200	4.95×10^{-4}	5.52×10^{-3}
1500	4.93×10^{-4}	5.05×10^{-3}
1750	4.91×10^{-4}	5.01×10^{-3}
2000	4.89×10^{-4}	4.98×10^{-3}
2500	4.88×10^{-4}	4.97×10^{-3}
5000	$\mathbf{4.77\times10^{-4}}$	$\mathbf{4.81\times10^{-3}}$

corresponding position of the input data. Figure 3 shows the training input data together with the position of the selected support vectors, represented by dark bars. It can be shown [De Brabanter et al., 2008] that the subsample selected by this entropy criterion has a uniform distribution. An advantage of using a uniform subset is that the subsample is well spread over the full data set leaving no large gaps in between.

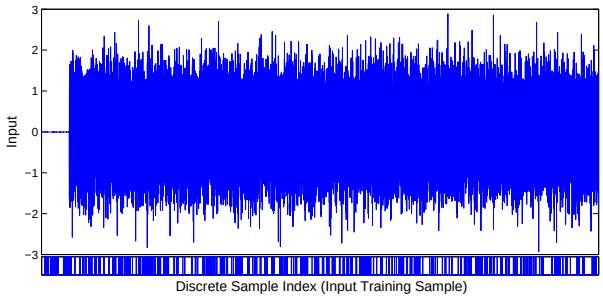

Fig. 3. (top) The input training sample; (bottom) the position, as time index, of the 2000 selected support vectors by quadratic Rényi entropy is represented by the dark bars.

Finally, the effect of different lags was tested for lags varying from 2 to 35. Figure 4 shows the evolution of the lags on cross-validation MSE (one-step-ahead based and simulation based). In these experiments the number of input lags and output lags was kept equal to each other (setting different input and outputs lags did not result in better performance on cross-validation MSE). For this example it did not matter whether the lags were selected by CV-MSE$_1$ or CV-MSE$_{\text{sim}}$. The CV line only moves up and does not show any significant shifts to left or right. Thus, the selection of the number of lags seems independent of the chosen CV criterion. Selecting the number of lags is based on the least complex model that

falls within one standard error (represented by the error bar at lag 23) of the best model [Hastie et al., 2001]. In this case the number of lags have chosen to be 10 (vertical dashed line in Figure 4).

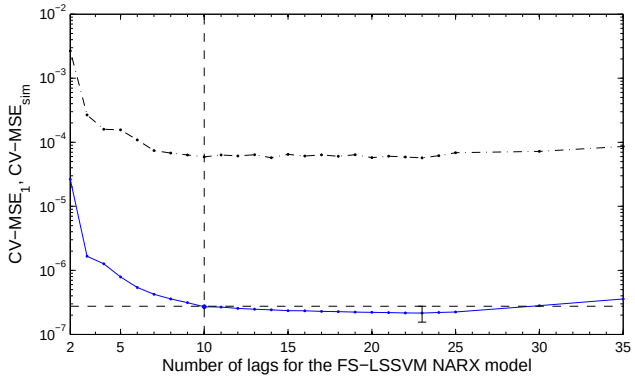

Fig. 4. MSE on cross-validation (CV-MSE$_1$ full line, CV-MSE$_{\text{sim}}$ dash-dotted line) for FS-LSSVM NARX model for different number of lags. The number of lags are chosen so that the least complex model falls within one standard error (error bar at lag 23) of the best, i.e. number of lags equal to 10 .

5.2 Results on Test Data

After selecting the model order and the involved parameters each of the models is used to make an iterative prediction, i.e. using only past predictions and input information, for data points starting at sample 100.001 to sample 188.000. Since this is unseen data for the model, the following source of error can be expected: due to the iterative nature of the simulation, past errors can propagate to the next predictions. From the difference between the iterated prediction and the true values, the Root Mean Squared Error (RMSE) on the test set (RMSE$_{\text{test}}$) is computed. In all results on test data the initial conditions for simulation were set to the real output values (first lag samples). Setting the initial conditions to zero did not result into lower/higher performance on RMSE since the first 1001 samples from the prediction are omitted from consideration to eliminate the influence of transient errors at the beginning of the simulation.

We also tried 2 combined models consisting of (1) a combination of FS-LSSVM (RBF) + FS-LSSVM (poly) and (2) a combination of FS-LSSVM (RBF) + FS-LSSVM (poly) + FS-LSSVM (lin). These different submodels, each consisting of 2000 support vectors, can then be combined via a linear combination (CM$_{1,\text{lin}}$ and CM$_{2,\text{lin}}$) of the submodels or via an MLP. The linear combination results in the following model form

$$f(x) = \sum_{i=1}^{q} \beta_i f_i(x), \tag{15}$$

with q the number of submodels and f_i the i-th individual FS-LSSVM. The optimal weights β are determined using a committee network approach [Suykens et al., 2002]. An extension to this is to consider a nonlinear combination (CM$_{1,\text{MLP}}$ and CM$_{2,\text{MLP}}$) of the submodels. Taking an MLP in the second layer, the model is described by

$$\hat{g}(z) = w_{\text{MLP}}^T \tanh(V\hat{z} + d) \tag{16}$$

with

$$\hat{z}_i(x) = w_i^T \varphi_i(x) + b_i, \quad i = 1, \ldots, q \tag{17}$$

where q denotes the number of individual FS-LSSVM models whose outputs z_i are the input to an MLP with output weight vector $w_{\text{MLP}} \in \mathbb{R}^{n_h}$, hidden layer matrix $V \in \mathbb{R}^{n_h \times q}$, bias vector $d \in \mathbb{R}^{n_h}$ and n_h are the number of hidden neurons. In both models the regression vector $x = [y_{t-1}, \ldots, y_{t-p}; u_{t-1}, \ldots, u_{t-p}]$ is used. The MLPs used in the ensemble model are trained with respect to w_{MLP}, V and d using backpropagation incorporating a Bayesian regularization scheme. All FS-LSSVM submodels in the ensemble use 2000 support vectors. Only the model FS-LSSVM (RBF) consist of 5000 support vectors.

Table 3 shows the performances (RMSE$_{\text{test}}$) of the iterative prediction on test data for all types of model structures. The FS-LSSVM, using an RBF kernel, outperforms the ARX and NARX by a factor 10 and 4 respectively on RMSE on test data. The last column is Table 3 gives a fit percentage, i.e. the percentage of the output variation that is explained by the model and is defined by the following equation, fit $= 100\left(1 - \frac{\|y-\hat{y}\|}{\|y-\bar{y}\|}\right)$, where y is the measured output, $\hat{y}$ the simulated output and $\bar{y}$ the mean of y. Table 4 shows for the three different models the following results: the mean value of the iterative prediction on test data $\mu_t = (1/8700)\sum_{t=101001}^{188000} e(t)$, the standard deviation of the error (on test data) $s_t = (1/8700)\sum_{t=101001}^{188000}(e(t) - \mu_t)^2)^{1/2}$, the RMSE value of the training error (RMSE$_{\text{tr}}$) and $e(t)$ is the simulation error. Figure 5 shows the result of the final iterative prediction and the corresponding errors in the time and frequency domain.

Table 3. RMSE and fit percentage with the final iterative prediction using the model in simulation mode on the predefined test set. n_h denotes the number of hidden neurons in the MLP. n_h and lags for MLP-NARX are found by validation on a single set.

Method	lags/n_h	RMSE$_{\text{test}}$	fit (%)
Linear ARX	10/-	5.6×10^{-2}	76.47
MLP-NARX	11/15	2.3×10^{-2}	86.06
FS-LSSVM (Lin)	10/-	4.3×10^{-2}	81.93
FS-LSSVM (Poly)	10/-	6.0×10^{-3}	96.86
FS-LSSVM (RBF)	10/-	4.7×10^{-3}	97.98
CM$_{1,\text{lin}}$	10/-	5.8×10^{-3}	96.54
CM$_{2,\text{lin}}$	10/-	6.2×10^{-3}	95.25
CM$_{1,\text{MLP}}$	10/10	5.3×10^{-3}	96.98
CM$_{2,\text{MLP}}$	10/10	4.8×10^{-3}	97.88

Table 4. RMSE on training (RMSE$_{\text{tr}}$) with iterative prediction on the training data using the model in simulation mode. The mean value of the simulation error μ_t and the standard deviation of the error s_t are also reported (on test data).

Method	μ_t	s_t	RMSE$_{\text{tr}}$
Linear ARX	-3.6×10^{-2}	4.3×10^{-2}	5.5×10^{-2}
MLP-NARX	-2.4×10^{-3}	2.7×10^{-2}	2.2×10^{-2}
FS-LSSVM (Lin)	-1.4×10^{-4}	4.3×10^{-2}	4.2×10^{-2}
FS-LSSVM (Poly)	1.2×10^{-4}	6.1×10^{-3}	5.9×10^{-3}
FS-LSSVM (RBF)	6.3×10^{-5}	4.8×10^{-3}	4.5×10^{-3}
CM$_{1,\text{lin}}$	1.2×10^{-4}	5.8×10^{-3}	5.6×10^{-3}
CM$_{2,\text{lin}}$	1.8×10^{-4}	6.3×10^{-3}	6.1×10^{-3}
CM$_{1,\text{MLP}}$	1.1×10^{-4}	5.4×10^{-3}	5.2×10^{-3}
CM$_{2,\text{MLP}}$	9.8×10^{-5}	4.8×10^{-3}	4.7×10^{-3}

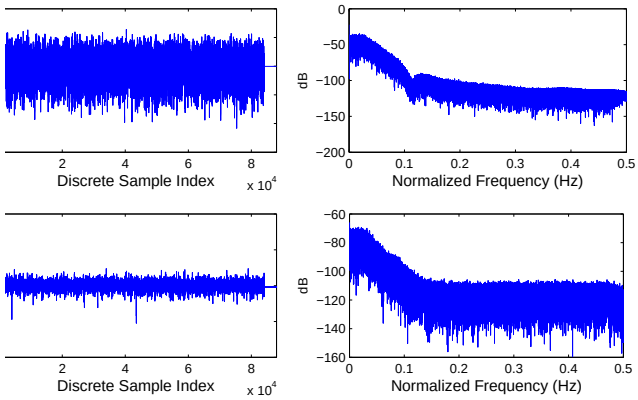

Fig. 5. (top left) Iterative prediction (simulation mode) of the test data; (top right) Normalized frequency plot of the simulated test data; (bottom left) Errors of the iterative prediction (simulation mode) in the test set; (bottom left) Normalized frequency plot of the errors of the iterative prediction.

6. CONCLUSION

This paper reports the application of a black-box NARX approach on the SYSID 2009 Wiener-Hammerstein Benchmark. Since for large data sets the LS-SVM methodology is computationally too expensive to take the information of the entire training set into account we used a variant of LS-SVM called FS-LSSVM. This algorithm is designed to estimate in the primal where an explicit expression for the approximated feature map is needed. The results show that FS-LSSVM obtains the best result on the iterative prediction for the Wiener-Hammerstein benchmark data. The best performance yields an RMSE on the predefined test set of 4.7×10^{-3}. These results can be approximated by using an ensemble strategy of three FS-LSSVM models with different kernels each consisting of 2000 support vectors. We report an optimal RMSE on the predefined test set and of this procedure of 4.7×10^{-3}.

ACKNOWLEDGEMENTS

(BDM/JS) is a professor at the Katholieke Universiteit Leuven, Belgium. Research supported by: Research Council KUL: GOA AM-BioRICS, CoE EF/05/006 Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite+; Helmholtz: viCERP. Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; Contract Research: AMINAL.

REFERENCES

H. Akaike. Statistical predictor identification. *Ann. Inst. Statist. Math.*, 22:203–217, 1973.

P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.

A.C. Davison and D.V. Hinkley. *Bootstrap Methods and their Application, reprinted with corrections*. Cambridge University Press, 2003.

K. De Brabanter, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Optimized fixed-size least squares support vector machines for large data sets. Technical Report 08-193, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), October 2008.

M. Espinoza, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B. De Moor. A comparative study of LS-SVMs applied to the silver box identification problem. *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2004)*, September 2004.

M. Espinoza, J.A.K. Suykens, R. Belmans, and B. De Moor. Electric load forecasting - using kernel based modeling for nonlinear system identification. *IEEE Control Systems Magazine, Special Issue on Applications of System Identification*, 27(5):43–57, October 2007.

M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14: 669–688, 2002.

W. Greblicki and M. Pawlak. *Nonparametric System Identification*. Cambridge University Press, 2008.

W. Härdle. Resampling for inference from curves. *In Proceedings of the 47th Session of International Statistical Institute*, pages 59–69, 1989.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

L. Ljung. *System Identification: Theory for the User*. Prentice Hall, second edition, 1999.

E.J. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54:185–204, 1930.

J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear bloack-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.

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.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1999.

C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *In Advances in Neural Information Processing Systems*, 2001.

S. Xavier de Souza, J.A.K. Suykens, J. Vandewalle, and D. Bollé. Cooperative behavior in coupled simulated annealing processes with variance control. *Proc. of the International Symposium on Nonlinear Theory and its Applications (NOLTA2006)*, pages 114–119, 2006.