UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SUPPORT VECTOR MACHINES FOR SYSTEM IDENTIFICATION

A. Marconato, M. Gubian, A. Boni, D. Petri

May 2007

Technical Report # DIT-07-032

Support Vector Machines for System Identification

A. Marconato, M. Gubian, A. Boni, D. Petri
Department of Information and Communication Technology
University of Trento,
Via Sommarive, 14 – 38050 Trento, Italy
Phone: +39 0461 883915, Fax: +39 0461 882093
email: anna.marconato@dit.unitn.it.

1 Introduction

In this document we propose the use of a widely known learning–from–examples paradigm, namely the Support Vector Machines (SVMs), for system identification problems.

Since they were first introduced by V. Vapnik in the mid-90s [1], SVMs have been successfully employed in a variety of applications, including both classification and regression problems. Here we will focus on the Support Vector Machines for Regression (SVRs), since they are expressly designed for situations in which a real function has to be estimated on the basis of a set of input/output measures characterizing the (typically unknown) system of interest.

In [2] we have proposed to exploit SVRs for the dynamic compensation of sensors based on inverse modeling. Although probably Neural Networks (NNs) are currently the most popular solution for sensor compensation, our choice was motivated by the fact that SVRs have shown interesting properties when compared to NNs: they do not suffer from the problem of encountering local minima during the optimization process because it consists in solving a constrained quadratic problem; moreover such kind of learning machines are based on a robust statistical theory, that is, Vapnik' Statistical Learning Theory [3].

Besides these evident theoretical advantages, SVR-based sensor compensation algorithms proved to be quite effective also when implemented on resource-constrained devices like simple 8-bits microcontrollers [2].

The encouraging results – both in terms of simulation and hardware implementation experiments – obtained applying SVRs to the case of sensor compensation have suggested the possibility to extend our approach to the more general class of system identification problems. In particular, we decided to start off with the identification of a simple linear system taken from [4], and to proceed with the non-linear case as a second step.

Before illustrating the proposed methodology, we need to provide a clear explaination of the theory of SVRs, in order to better understand the philosophy lying behind our approach, and a brief review on the state of the art.

2 Support Vector Machines for Regression

In this section we describe different formulations of the SVR which are considered in this work, namely standard ν-SVR [5], the Reduced SVR (RSVR) by Lee and Mangasarian [6], and a novel sub-optimal reduced-set method, Extended Reduced SVR (ERSVR).

2.1 Standard SVR algorithms

The broad class of regression problems refers to all those situations in which one has to reconstruct a real function $y = f(\boldsymbol{x})$, on the basis of a set of examples $Z = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, the training set. Here $y_i \in \Re$, and $\boldsymbol{x}_i \in \Re^d$ are vectors of d features $x_i(n), x_i(n-1), \ldots, x_i(n-d+1)$.

In particular, we are looking for an approximating function $\hat{f}(\cdot)$ that needs to be sufficiently smooth, and for which training errors are penalized only outside a so-called *insensitive zone*, whose width is indicated by ε [3]. This regression function is obtained, following the ν-SVR approach, by solving a constrained quadratic optimization problem, here expressed in the primal form [5]:

$$\min_{\hat{f},\boldsymbol{\xi},\boldsymbol{\xi}^*} \left[\frac{1}{2} \left\| \hat{f} \right\|_k^2 + C \cdot \left(\nu\varepsilon + \frac{1}{m} \sum_{i=1}^{m} (\xi_i + \xi_i^*) \right) \right] \tag{1}$$

$$\begin{aligned}
\text{subject to} \qquad & \hat{f}\left(\boldsymbol{x}_i, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) - y_i \leq \varepsilon + \xi_i && (2)\\
& y_i - \hat{f}\left(\boldsymbol{x}_i, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) \leq \varepsilon + \xi_i^* && (3)\\
& \xi_i \geq 0 && (4)\\
& \xi_i^* \geq 0 && (5)\\
& \varepsilon \geq 0. && (6)
\end{aligned}$$

Here $\left\| \hat{f} \right\|_k^2$ is a norm in the Hilbert space $\mathcal{H}$ defined by suitable kernel functions $k(\cdot,\cdot)$, C is the regularization factor, and $\boldsymbol{\xi}, \boldsymbol{\xi}^*$ are two vectors of *slack* variables, introduced in order to deal with small errors. More in details, $\boldsymbol{\xi}, \boldsymbol{\xi}^*$ represent the distances from the two edges of the ε-tube, the region within which errors are considered to be negligible. This formulation of the SVR problem was proposed in [5] by Schölkopf *et al.* in order to control the trade-off between the size of the ε-tube and model complexity. In fact, hyperparameter ν can be seen as an upper bound on the fraction of errors and a lower bound on the fraction of parameters needed to build the regression function. Moreover, the choice of ν permits to automatically compute the value of ε, which does not need to be specified beforehand [5].

Usually problem (1) is solved in its dual form, which is obtained exploiting the Lagrange multipliers approach [5]:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \left[\frac{1}{2} \sum_{i,j=1}^{m} \left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right) k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) - \sum_{i=1}^{m} y_i \left(\alpha_i^* - \alpha_i\right) \right] \tag{7}$$

$$\text{subject to} \qquad \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0 \tag{8}$$

$$0 \leq \alpha_i, \alpha_i^* \leq \frac{C}{m}, \forall i = 1, \ldots, m \tag{9}$$

$$\sum_{i=1}^{m} (\alpha_i + \alpha_i^*) \leq C \cdot \nu. \tag{10}$$

where α_i and α_i^* are the Lagrange multipliers associated with constraints (2) and (3).

As a result, the estimating function $\hat{y}$ can be expressed as follows [3]:

$$\hat{y} = \hat{f}\left(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, b\right) = \sum_{i \in SV} (\alpha_i^* - \alpha_i) k\left(\boldsymbol{x}_i, \boldsymbol{x}\right) + b \tag{11}$$

where SV is the set of indices of the only examples (called the Support Vectors) needed to build the regression function, and parameter b can be computed exploiting the Karush-Kuhn-Tucker conditions (see [5] for the details).

2.2 Sub-optimal reduced-set SVR algorithms

Although usually very accurate, in many cases standard SVR algorithms are not suitable for practical applications. Thus, alternative approaches should be investigated, in order to better control the number of parameters which characterize the approximating function. In recent years, a lot of work has been done in this direction, leading to the formulation of several algorithms with reduced complexity, but still guaranteeing a good level of accuracy [6], [7].

Among others, the Reduced Support Vector Machine (RSVR) proposed in 2001 by Lee and Mangasarian randomly selects a (small) portion of the training set as the set of "support vectors", to build a sparse approximating function [6].

We propose instead the use of a novel algorithm, the Extended RSVR (ERSVR), which is inspired by RSVR. However, while in the latter the support vectors are a randomly chosen subset of the original dataset, our approach considers this small set of samples only in the initial step. More specifically, we fix *a priori* a number N_{xv} of "support vectors" and write:

$$\boldsymbol{\varpi} = \sum_{i=1}^{N_{xv}} (\alpha_i - \alpha_i^*) \psi\left(\boldsymbol{x}_i\right) \tag{12}$$

where $\psi(\boldsymbol{x}_i)$ defines a kernel function $k(\cdot, \boldsymbol{x}_i)$.

Then a further optimization procedure is performed to generate new "support vectors", which can actually be quite different from the examples of the training set and will therefore be referred to as "expansion vectors" (and denoted with $\boldsymbol{z}_i$). This procedure somehow follows the idea presented in [8] for classification problems, but here it has been modified to fit the regression case. Moreover, we suggest to solve the primal form instead of the dual one, since it leads to several improvements, first of all in terms of computational complexity.

Thus, problem (1) becomes:

$$\min_{\boldsymbol{\varpi}} \left[\frac{1}{2} \|\boldsymbol{\varpi}\|^2 + C \cdot \frac{1}{m} \sum_{i=1}^{m} (\xi_i + \xi_i^*) \right] \tag{13}$$

obtained by adding the term $\frac{1}{2}b^2$ into the objective function to be minimized. Notice also that hyperparameter ν does not appear in this formulation of the ERSVR problem, since it has been developed on the basis of the standard ε-SVR approach [5]. To emphasize the role of the double optimization procedure, we can express the ERSVR problem in a compact notation as:

$$\min_{\boldsymbol{z}} \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \left[\frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \boldsymbol{Q} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + C \cdot \frac{1}{m} \sum_{i=1}^{m} (\xi_i + \xi_i^*) \right] \tag{14}$$

subject to the usual constraints (2)-(5). Here $Q_{ij}^z = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$ are the entries of the kernel matrix $\boldsymbol{Q}$. Problem (13) is in practice composed by two different levels of optimization. The inner one consists in solving a standard quadratic programming problem by Newton method, while the outer one finds, after several iterations, new vectors $\boldsymbol{z}$ by gradient-descent method.

As a solution to the above problem, the final estimating function will take the form:

$$\hat{y} = \hat{f}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \sum_{i=1}^{N_{xv}} (\alpha_i - \alpha_i^*) k(\boldsymbol{x}, \boldsymbol{z}_i) \tag{15}$$

Notice that the bias term b does not appear in the ERSVR function, since it has been incorporated in the minimization problem. Moreover, it is worth noting that the number N_{xv} of expansion vectors is typically chosen to be much smaller than the size m of the original training set.

Both RSVR and ERSVR are an approximation of the standard SVR problem, thus their solutions must be considered sub-optimal. However, they appear as interesting and promising approaches for practical applications, since they allow to reduce substantially the number of parameters of the regression function, thus lowering the complexity of the algorithm that needs to be implemented.

3 State of the Art on SVMs and System Identification

SVRs have been used for approximating linear and nonlinear functions, also in presence of noise. To select the best configuration of hyperparameters, some knowledge about noise distribution in the training data can be useful. For time series prediction and system identification problems, the goal is to find a set of parameters for a proposed model, on the basis of measured input/output values. One of the main advantages of using SVRs for estimating model parameters is that the number (and position) of the support vectors (that is, of the kernel functions needed to approximate the system) is found automatically and optimally. When dealing with noisy data, the choice of the ε-insensitivity can help in trading off model errors and complexity [9]. Moreover, in comparison with ANN-based methods, SVRs are not affected by local minima problems and guarantee faster convergence.

Traditional approaches for system identification try to obtain system model by a set of input/output data by minimizing an error cost. SVRs, instead, adopts structural risk minimizaton principle to guarantee good generalization ability on unseen samples. A good level of accuracy is reached, without specific knowledge about the system to be identified, or the model structure. Nonlinear identification problems appear as suitable candidate applications, thanks to the employment of kernel functions to express the nonlinear SVR relationship between input and output [10].

A different formulation of the SVR algorithm, namely the Least Squares SVM (LS-SVM), can be also used for dynamic system identification. However, the sparseness property which constitutes one of the nicest aspects of standard SVMs is lost for LS-SVMs [11].

A partially-linear version of the LS-SVM has been proposed in order to identify models for which there is a specific knowledge that nonlinearities apply only on a subset of the inputs. The goal is to increase the performance respect to considering a full linear model, but at the same time to reduce the complexity that would result from a full nonlinear technique [12].

SVRs have been also used to design a technique for ARMA modeling, where the parameters that characterize the model are included in the cost function to be minimized [13]. This formulation of the problem has been developed also in a reproducing kernel Hilbert space (RKHS) framework. Composite kernels can be considered, in order to emphasize the input/ouput cross information [14].

4 Proposed Methodology and Simulation Results

To verify the viability of a SVR-based approach for system identification we have decided to study first a simple linear case. In order to perform a comparison with traditional techniques, the “Simulation Example 1” illustrated in [4] in

Section 7.15.1 was chosen. The idea was to provide a performance plot as the one showed in Figure 7-8, obtained by evaluating the relative square error of the transfer function estimate similarly to Equation (7-126). (In this preliminar phase of the work, we did not consider different realizations of the data set, thus a proper (relative) *mean* square error could not be computed. However, we plan to complete this part as soon as possible).

The first step in order to be able to apply SVRs to the considered case study is to translate the problem from the frequency domain to the time domain, so that the data set can be expressed in the form $Z = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, where $y_i \in \Re$ and $\boldsymbol{x}_i = x_i(n), x_i(n-1), \dots, x_i(n-d+1)$. Starting from the coefficients of the transfer function that has to be estimated provided in Table 7-2, we have derived the following difference equation:

$$\sum_{i=0}^{5} a_i y(n-i) = \sum_{i=0}^{5} b_i x(n-i) \qquad (16)$$

with coefficients:

$$\begin{array}{ll}
a_0 = 1 & b_0 = 0.0188 \\
a_1 = -3.8503 & b_1 = 0.0192 \\
a_2 = 6.0347 & b_2 = -0.0364 \\
a_3 = -4.7971 & b_3 = -0.0364 \\
a_4 = 1.9298 & b_4 = 0.0192 \\
a_5 = -0.3138 & b_5 = 0.0188
\end{array}$$

The input signal $x(n)$ has been generated as a Gaussian-distributed random variable with zero mean and variance equal to 1. A Gaussian noise with zero mean and variance $2 \cdot 10^{-6}$ has been been added to both the input and output signals. Moreover, since the d features $x_i(n), x_i(n-1), \dots, x_i(n-d+1)$ represent past values of the input signal, information obtained from the analysis of the impulse response function can be exploited in order to have an idea of the number of past samples that need to be taken into consideration. In Figure 1 a plot of the impulse response function of system (16) is provided.

Different training sets Z_d has been generated, with an increasing number d of features (from 1 to 64). The idea is to add d to the set of SVR parameters that have to be tuned in the training phase, thus trying to reduce the overall complexity of the estimation function $\hat{y}$. Each training set Z_d was made of 400 input/output examples.

Optimal configurations of SVR parameters have been searched for by employing a popular Genetic Algorithm, namely NSGA-II [15], in the model selection phase, following a multi-objective optimization approach. Our objective functions to be minimized are (i) an error index, the MSE computed on a test set made of 100 unseen examples, and (ii) a complexity term that expresses the computational effort needed for a single evaluation of $\hat{y}$. The latter term is a suitably chosen function of the number of support vectors and the number of features. Resulting solutions of such a multi-objective approach can not be really considered "optimal" ones, but are rather trade-off solutions for which

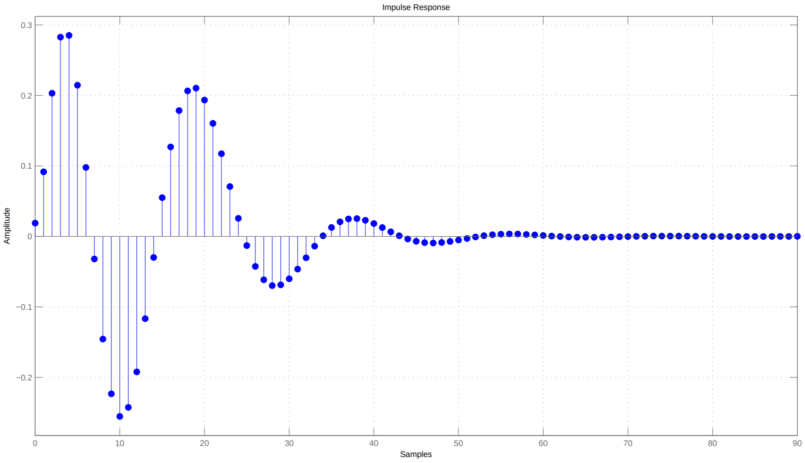

Figure 1: Impulse response function of the considered system.

one objective function can not be improved unless the other one is increased [16]. In the typical situation in which no *a priori* information is available for the choice of the final solution, the user has to select, among these trade-off points, the one that best fits to the requirements of the problem. In this work, since the implementation was not yet an issue, we have selected the solution with least MSE value (that is, with highest complexity). However, to test the validity of our methodology, we have considered also solutions whose complexity was reduced by one half (with an obvious increase of the MSE). In those cases, results are of course slightly worse in terms of the transfer function error, but can still be considered satisfactory when stringent constraints on the complexity have to be met.

After the training phase has been performed, and once the final SVR model is selected, one hundred validation sets representing sinusoids at different frequencies in the range $[0.05Hz, 5Hz]$ (with step $0.05Hz$) were generated. In this way it was possible to evaluate the SVR transfer function estimate in the band of interest, and to compute the relative square error $\left|(\widehat{G}_{SVR} - G_0)/G_0\right|^2$.

Figure 2 depicts the results obtained with the three different SVR algorithms. We can notice that, compared to the performance of the traditional estimators presented in [4] in Figure 7-8, the ν-SVR approach performs as well as (or even slightly better than) the GTLS method, but only in the first half of the considered band, then the error value tends to remain higher than the ones resulting from all traditional approaches. However, our goal here is not that of proposing the use of an SVR-based approach to solve linear identification problems, but instead these results should be considered as a first step towards the application of SVRs to the nonlinear case. Moreover, we have to stress the

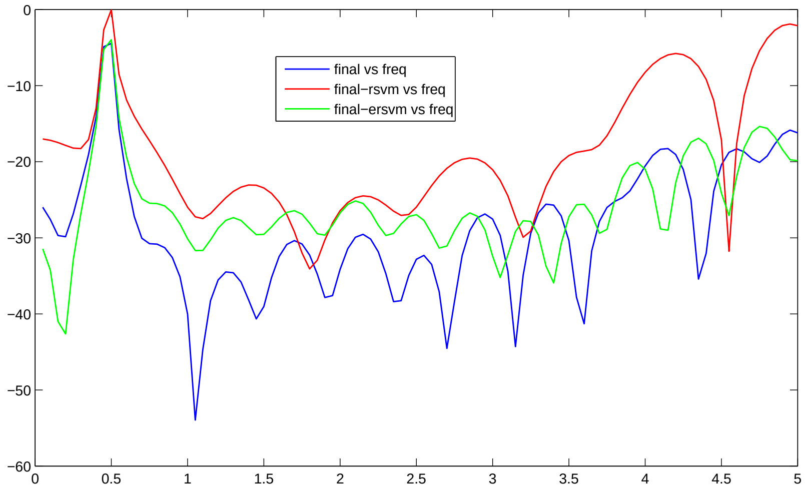

Figure 2: Relative square error (in dB) of the transfer function estimated with the ν-SVR (blue line), RSVR (red line) and ERSVR (green line).

fact that SVR algorithms try to approximate the behavior of the system only on the basis of a set of input/output examples, without specific knowledge about the structure of the system.

As far as RSVR and ERSVR algorithms are concerned we must observe an increase in the error values respect to the standard ν-SVR (due to the fact that they are sub-optimal approaches), although we can see that the ERSVR plot is still quite close to the ν-SVR curve.

As a second step, we have decided to change the power of the noise added to the input and output signals. In the previous examples the variance of the noise was fixed equal to $2 \cdot 10^{-6}$, now we will see the effects of increasing it to 10^{-4} and 10^{-2}. In particular, we would like to understand how robust to noise disturbance the SVR approach can be. In Figure 3 results in terms of the relative transfer function square error are shown, for different values of the noise power. We can see that increasing noise power to 10^{-4} does not really affect the performance, but when higher values are considered the ν-SVR approach does not seem to be particularly robust.

Finally, we have started to take a look at the nonlinear case, by introducing in the formulation of the difference equation (16) a nonlinear distorsion. The modified system equation is expressed as follows:

$$\sum_{i=0}^{5} a_i y(n-i) = b_0 (x(n))^3 + \sum_{i=1}^{5} b_i x(n-i) \tag{17}$$

with the same coefficients used previously. Since SVR algorithms are nonlinear

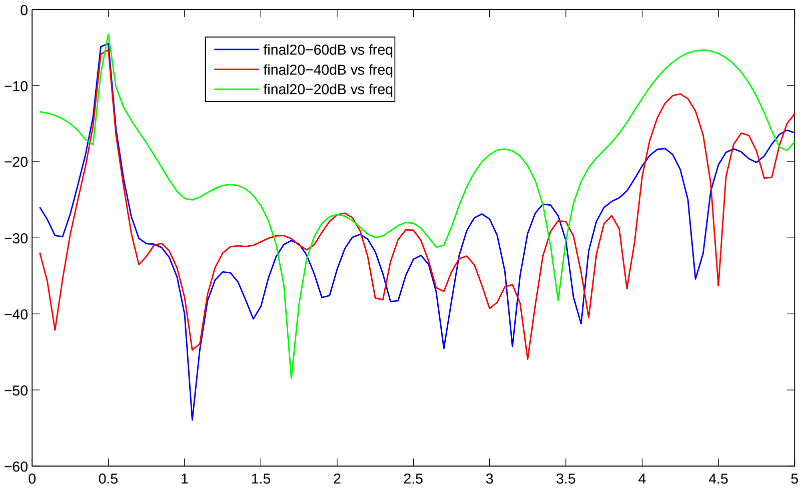

Figure 3: Relative square error (in dB) of the transfer function estimated with the ν-SVR, when noise power on input and output signals is increased from $2 \cdot 10^{-6}$ (blue line), to 10^{-4}(red line), and to 10^{-2} (green line).

approaches, they are expected to work well when the system to be identified is characterized by nonlinear distorsions. Clearly, in the nonlinear case we can not provide a result plot as for the examples above, but the SVR training MSE value can give an idea of the performance. Table 1 summarizes the MSE values obtained in the SVR training phase in all situations described in this work, together with indications about the complexity (in terms of number of features and number of support vectors). As a first comment, we can say that the results we have discussed so far essentially mirror the SVR MSE values. Moreover, we observe that for the analyzed problem the performance in the nonlinear case is slightly worse than the one obtained for the linear problem, but these results can still be considered quite satisfactory. As far as complexity is concerned, it is evident that RSVR and ERSVR allow to reduce significantly the number of support vectors.

5 Conclusions

This document briefly reports some preliminar results we have obtained applying state of the art learning–from–examples algorithms to the problem of system identification. The results for both the linear and the nonlinear case are acceptable, although a decrease in the performance is observed for the nonlinear problem. However, this work is to be intended as a first attempt to face these issues, many interesting aspects still need to be investigated more thoroughly.

algorithm	noise power	MSE	features	SVs
ν-SVR	$2 \cdot 10^{-6}$	$3.7 \cdot 10^{-4}$	51	220
RSVR	$2 \cdot 10^{-6}$	$1.5 \cdot 10^{-3}$	40	49
ERSVR	$2 \cdot 10^{-6}$	$2.3 \cdot 10^{-4}$	55	4
ν-SVR	10^{-4}	$5.8 \cdot 10^{-4}$	56	382
ν-SVR	10^{-2}	$2.5 \cdot 10^{-2}$	35	126
ν-SVR (nonlinear case)	$2 \cdot 10^{-6}$	$5.1 \cdot 10^{-3}$	16	23
RSVR (nonlinear case)	$2 \cdot 10^{-6}$	$3.4 \cdot 10^{-3}$	15	22
ERSVR (nonlinear case)	$2 \cdot 10^{-6}$	$4.3 \cdot 10^{-3}$	35	4

Table 1: MSE and complexity values resulting from the SVR training phase.

We are currently making an effort to complete the missing parts of this work, especially in order to study the possibility of employing SVR algorithms in the nonlinear case.

References

[1] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995.

[2] A. Marconato, M. Hu, C. Marzadro, A. Boni, and D. Petri, “A resource–constrained sensor dynamic compensation using a learning–from–examples approach,” in *Instrumentation and Measurement Technology Conference (To appear)*, Warsaw, Poland, 2007.

[3] V. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[4] R. Pintelon and J. Schoukens, *System Identification: a Frequency Domain Approach.* IEEE Press, 2001.

[5] B. Schölkopf and A. Smola, *Learning with Kernels.* The MIT Press, 2002.

[6] Y.-J. Lee and O. Mangasarian, “RSVM: Reduced support vector machines,” in *First SIAM International Conference on Data Mining*, Chicago, Apr. 2001.

[7] S. Keerthi, O. Chapelle, and D. DeCoste, “Building support vector machines with reduced classifier complexity,” *Journal of Machine Learning Research*, vol. 7, pp. 1493–1515, 2006.

[8] M. Wu, B. Schölkopf, and G. Bakir, “Building sparse large margin classifiers,” in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 1001–1008.

[9] P. Drezet and R. Harrison, “Support vector machines for system identification,” in *UKACC International Conference on Control ’98*, 1998.

[10] M.-G. Zhang, W.-W. Yan, and Z.-T. Yuan, "Study of nonlinear system identification based on support vector machine," in *Third International Conference on Machine Learning and Cybernetics*, 2004.

[11] X.-D. Wang and M.-Y. Ye, "Nonlinear dynamic system identification using least squares support vector machine regression," in *Third International Conference on Machine Learning and Cybernetics*, 2004.

[12] M. Espinoza, J. Suykens, and B. de Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans. on Automatic Control*, vol. 50, no. 10, pp. 1602–1606, 2005.

[13] J. L. Rojo-Álvarez, M. Martínez-Ramón, M. de Prado-Cumplido, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Support vector method for robust ARMA system identification," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 155–164, Jan. 2004.

[14] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, J. M. noz Marí, A. Navia-Vázquez, E. Soria-Olivas, and A. R. Figueiras-Vidal, "Support vector machines for nonlinear kernel ARMA system identification," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1617–1622, Nov. 2006.

[15] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," in *Proceedings of the Parallel Problem Solving from Nature VI Conference*. Paris, France: Springer. Lecture Notes in Computer Science No. 1917, 2000, pp. 849–858.

[16] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., New York, 2001.