A SUPPORT VECTOR REGRESSION METHOD FOR REDUCING THE HIGH-ORDER SYSTEMS TO FIRST-ORDER PLUS TIME-DELAY FORMS

Rana ORTAC-KABAOGLU[1]

[1]Istanbul University, Electrical-Electronics Engineering, Avcilar, Istanbul, Turkey
rana@istanbul.edu.tr

Abstract: *In this paper, a novel method for reducing the high-order systems to first-order plus time-delay forms is introduced. For this purpose Support Vector Machines, which became a popular learning algorithm, is employed. Three parameters of the first-order plus time-delay forms are estimated by three parallel support vector regression machines. Satisfactory performance is obtained at the simulations.*
Keywords: *Support vector regression machines, high-order systems, first-order plus time-delay systems, modeling.*

1. Introduction

Model order reduction for high order systems is always an attractive research area. There are many studies in the literature that meet several needs and requests for this purpose. A system can be reduced to its lower order equivalent or can be modeled in a special form. First-order plus delay-time (FOPDT) system model is one of these special forms and is popular in many applications [1]. There are many PID controller tuning techniques in literature [2,3,4]. The model order reduction method proposed by S.Skogestad presents an analytical approach for this problem [5]. In Skogestad's method, it is required to know the model and model order of the system being reduced and furthermore, the system has to be in s-domain. This method determining the effective time delay via 'half-rule' also contains controller design procedures followed by the rules for model order reduction.

At this work, an SVR algorithm is proposed to solve the simplification problem of high order systems to a first order time delay system. The model of the system is not required in the proposed SVR method. Instead, only some data applied to the input and their corresponding outputs is needed.

The paper is organized as follows: The first order plus time delay model and its parameters are presented in section 2. Support vector regression mechanisms are described in section 3. The proposed method is given in section 4. Finaly, in section 5, simulations related to this method are carried out for four different high-order systems. Discussions and conclusions are presented in section 6.

Received on: 13.01.2011
Accepted on: 20.09.2011

2. First Order Plus Time Delay Models

A large number of industrial plants can be modeled approximately with a first order plus time delay (FOPTD) transfer function as follows [4]:

$$\frac{K}{\tau s+1}e^{-Tds} \qquad (1)$$

where K is the plant gain, τ is the time constant and T_d is the time delay (dead time). Our method is based on computing these three parameters via support vector regression machines. Before the introduction of detailed structure of the method, a brief explanation for SVR machine is given.

3. Support Vector Machines

Support vector machine (SVM) has become a very popular intelligent learning algorithm and a good alternative for neural networks. This method is developed by Vapnik [6], still gaining popularity due to many of its attractive features [7,8,9,10]. The short summaries of support vector regression machines are given below.

3.1. Support Vector Regression (SVR)

The SVM can be used for the regression problems successfully [11,12,13]. Let us assume that the training

data group to be used for learning f(x) function having input-output relation is given as in the following set:

$$D=\{(x_1,y_1),(x_2,y_2).....(x_l,y_l)\},\quad x\in R^n, y\in R \tag{2}$$

where x is the input and y is the output of the regression problem. Support vector machines find the function to be estimated as f(x). Input-output relation can be modelled with a linear regression model in feature space as the following:

$$f(x,w)=w^T x+b \tag{3}$$

In the regression, differing from classification, approach error is used instead of margin between optimal distinctive hyper plane and support vectors. In practice, various loss functions exist. The loss function to be used here is ε-tolerance loss function defined as follows:

$$Y_\varepsilon=\begin{cases}0 & if \quad |y-f(x,w)|<\varepsilon\\ |y-f(x,w)|-\varepsilon & other\end{cases} \tag{4}$$

The philosophy of regression by SVM is that a tube or a band, which has radius ε, is defined around estimating function f(x,w). If the value f is inside the tube, this means there is no loss. In other words, the loss is zero if the difference between estimated f value and measured y value is less than ε. The loss for all other estimating points excluded the tube equals to the absolute value of the difference between that estimating point and the radius of ε. Vapnik's loss function for ε=0 is equal to absolute loss function.

The goal is to minimize the experimental and observational risk expressing total error in the formulation of SVM algorithm and $\|w\|^2$ simultaneously. Consequently, a linear $f(x,w)=w^T x+b$ regression hyper-plane is found (with ξ and ξ^* slack variables) by the following expression:

$$\begin{aligned}&\text{Minimize:}\quad R(w,\xi,\xi^*)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{1}(\xi+\xi^*)\\ &\text{subject to:}\quad y_i-w^T x_i-b\le\varepsilon+\xi \quad i=1,..,l\\ &\qquad\qquad w^T x_i+b-y_i\le\varepsilon+\xi^* \quad i=1,..,l\\ &\qquad\qquad \xi\ge 0 \quad i=1,..,l\\ &\qquad\qquad \xi^*\ge 0 \quad i=1,..,l\end{aligned} \tag{5}$$

where C is the tradeoff parameter and ε value is another tradeoff parameter between the sparseness of the representation and closeness to the data. These parameters are chosen by the user by trial-and-error. The constrained optimization problem stated above is solved by establishing the primary Lagrangian.

$$\begin{aligned}L:=\frac{1}{2}\|w\|^2+C\left(\sum_{i=1}^{n}\xi_i+\sum_{i=1}^{n}\xi_i^*\right)-\sum_{i=1}^{n}\alpha_i\left(y_i-w^T x_i-b+\varepsilon+\xi_i\right)\\ -\sum_{i=1}^{n}\alpha_i^*\left(y_i-w^T x_i-b+\varepsilon+\xi_i^*\right)-\sum_{i=1}^{n}\left(\beta_i^*\xi_i^*+\beta_i\xi_i\right)\end{aligned} \tag{6}$$

where α_i, β_i are the Lagrangian multipliers. This is a saddle point problem which is difficult to solve. The Lagrangian function L has to be minimized with respect to w, b, ξ and ξ^*. Thus, differentiating L with respect to w, b, ξ and ξ^*, and setting the results equal to zero, the equivalent optimization problem becomes,

$$\begin{aligned}\max_{\alpha,\alpha^*}\quad W(\alpha,\alpha^*)=&-\varepsilon\sum_{i=1}^{n}(\alpha_i^*+\alpha_i)+\sum_{i=1}^{n}(\alpha_i^*-\alpha_i)y_i\\ &-\frac{1}{2}\sum_{i=1,j=1}^{n}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)x_i^T x_j\end{aligned} \tag{7}$$

$$\text{subject to}\quad \begin{aligned}&\sum_{i=1}^{n}\alpha_i^*=\sum_{i=1}^{n}\alpha_i\\ &0\le\alpha_i^*\le C \quad i=1,..,n\\ &0\le\alpha_i\le C \quad i=1,..,n\end{aligned}$$

This secondary Lagrangian, W, is denoted by Lagrangian multipliers **α** ve **α*** and maximized applying Karush-Kuhn-Tucker (KKT) optimality condition. After Lagrangian multipliers **α** and **α*** are obtained by a quadratic problem (QP) solver, optimal hyper-plane weight vector w can be determined as,

$$w_0=\sum_{i=1}^{n}(\alpha_i^*-\alpha_i)x_i \tag{8}$$

w vector is a linear combination of the input data. Among the training data, x vectors are called support vectors, whose $(\alpha_i - \alpha_i^*)$ coefficient is different from zero. Bias term of the regression hyper-plane is

$$b_0=\frac{1}{n}\sum_{i=1}^{n}(y_i-x_i^T w_0) \tag{9}$$

The optimum regression hyper-plane can then be written as follows:

$$f(x,w)=w^T x+b=\sum_{\text{supportvectors}}(\alpha_i^*-\alpha_i)<x_i x>+b \tag{10}$$

where α and α^* are Lagrange multipliers. Kernel functions have been introduced to decrease all of these inner product difficulties in the feature space [8]. Using the Kernel function the optimum regression hyper plane becomes,

$$f(\mathrm{x}, \mathrm{w}) = \sum_{i=1}^{SVs} (\alpha_i^* - \alpha_i) K(x_i x) + b \quad (11)$$

Kernel functions need to satisfy the Mercer condition [6] and two of most commonly used Kernel functions are;

Polynomial kernel function: $K(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^p$

Radial basis function: $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\left\| \mathbf{x} - \mathbf{x}' \right\|^2 / 2\sigma^2 \right)$

4. SVR-Based Approximation Method to FOPTD System

In this study, representation of high-order systems in the form of $\frac{K}{\tau s + 1} e^{-Tds}$ of first-order plus time-delay (FOPTD) system is realized using SVR learning algorithms (Fig. 1). SVR is applied as follows:

- Values for K, τ and Td are chosen from various intervals. Time delay system models for the combinations of chosen parameter values are formed.
- First, it is necessary to form training data pairs. The responses of the systems formed in the preceding step are employed as inputs of SVR.
- The number of combination is (step number of K) x (step number of τ) x (step number of Td). Combinations are formed by three loops which lie one inside the other. In order to simultaneously estimate those three parameters simultaneously, K, τ and Td three SVRs are trained as parallel.
- In training phase, K, τ and Td values corresponding to the each input data are employed as outputs of SVR
- After completing the training phase, step responses of the reduced systems are applied to the inputs of the SVRs and then K, τ and Td values are obtained at the outputs of each SVRs. Training and test phases are considered consequently.
- RBF and extended RBF kernel functions are used in both training and test phases. C and ε are chosen as 200 and 0.01 respectively. Modeling errors of both methods are evaluated for 500 sample points.

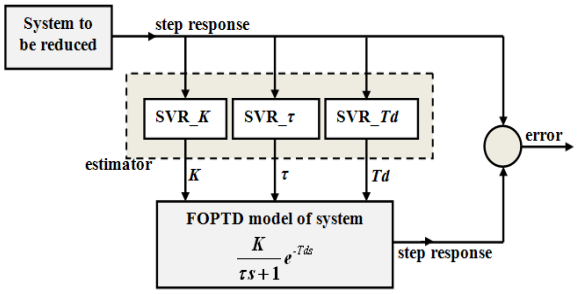

Figure 1. High order system reduction via 3 parallel SVR.

5. Simulation Examples

For four different systems, some of which are time-delay systems and all of which can be reduced to their first order representations, the SVR_foptd models, their Skogestad_foptd models and comparisons of their unit step responses are given below.

System 1: $G_1(s) = \frac{1}{(s+1)(0.2s+1)}$

SVR_foptd model and Skogestad_foptd model of the second-order system above are follows:

$$G_{1(SVR)} = \frac{0.9953}{1.0084s + 1} e^{-0.1396s} \quad , \quad \text{SVR\_error} = 0.0081$$

$$G_{1(Sko)} = \frac{1}{1.1s + 1} e^{-0.1s} \quad , \quad \text{Sko\_error} = 0.0087$$

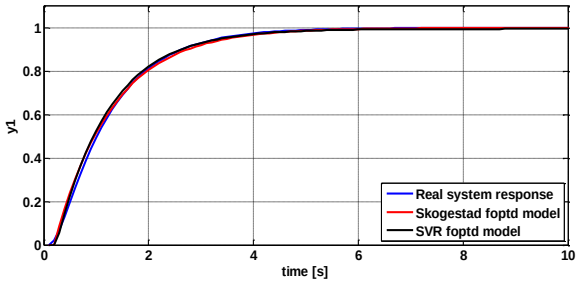

Figure 2: Step responses of the first system, Skogestad_foptd model and SVR_foptd model

System 2: $G_2(s) = \frac{2(15s+1)}{(20s+1)(s+1)(0.1s+1)^2}$

SVR_foptd model and Skogestad_foptd model of the system above are follows:

$$G_{2(SVR)} = \frac{1.9455}{1.4624s+1}e^{-0.1978s}, \text{ SVR\_error = 0.1072}$$

$$G_{2(Sko)} = \frac{1.5}{1.05s+1}e^{-0.15s}, \quad \text{Sko\_error = 0.3061}$$

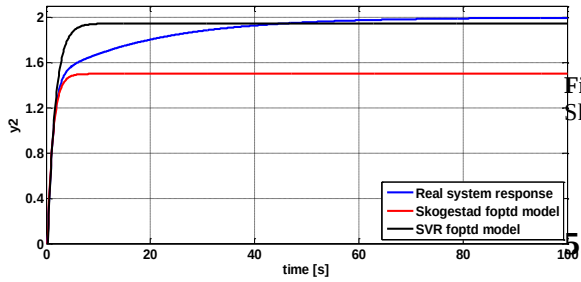

Figure 3: Step responses of the second system, Skogestad_foptd model and SVR_foptd model

System 3: $G_3(s) = \frac{1}{(s+1)^4}$

SVR_foptd model and Skogestad_foptd model of the fourth-order system above are follows:

$$G_{3(SVR)} = \frac{0.9991}{1.4649s+1}e^{-2.5084s}, \text{ SVR\_error = 0.0102}$$

$$G_{3(Sko)} = \frac{1}{1.5s+1}e^{-2.5s}, \quad \text{Sko\_error = 0.0102}$$

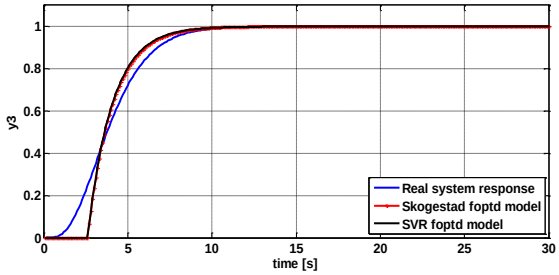

Figure 4: Step responses of the third system, Skogestad_foptd model and SVR_foptd model.

System 4: $G_4(s) = \frac{2s+1}{(10s+1)(0.5s+1)}e^{-s}$

SVR_foptd model and Skogestad_foptd model of the dead time system above are follows:

$$G_{4(SVR)} = \frac{1.0007}{8.5185s+1}e^{-0.9775s}, \text{ SVR\_error = 0.0185}$$

$$G_{4(Sko)} = \frac{0.625}{4.5s+1}e^{-1.25s}, \quad \text{Sko\_error = 0.2580}$$

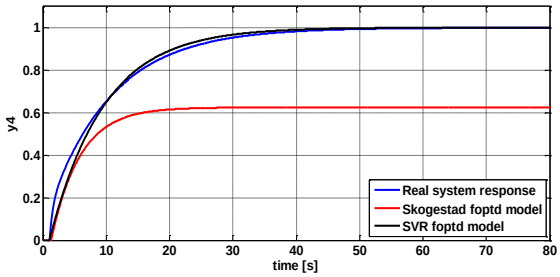

Figure 5: Step responses of the fourth system, Skogestad_foptd model and SVR_foptd model

5. Conclusions

An SVR algorithm is proposed to represent high order systems with their first order time delay models. The results obtained are compared with the method proposed by Skogestad as well as with the outputs of the real systems. The method proposed by Skogestad performs the model-order- reduction process by using a sequence of predefined rules. This method continues with some analytical rules following the reduction rules to design a controller. If obtaining FOPTD form of a high order system is the only thing demanded, Skogestad's method won't be suitable. Such as in the case of systems having zeros, serious differences and huge errors may appear while obtaining the system gain *K*. An SVR machine trained in a proper interval can overcome these problems. Moreover, the proposed SVR-based model-order reduction method has a great advantage that there is no need to know the model of the system being reduced since it only uses input data of and their corresponding outputs. The proposed SVR-based method outperforms the method of Skogestad in the sense of reduction to FOPTD form.

SVR comprises solution of quadratic programming and it works in a principle of minimization the squared error. Hence, SVR may not be able to calculate several important parameters for control systems, such as dominant poles, rise times as accurate as some control algorithms. Nevertheless, it is possible to obtain high performance in model-order-reduction using the results obtained from trials if proper search intervals for *K*, τ, and T_d and proper kernel function are chosen. If the number of parameter-intervals and the samples of system response are increased, then algorithm slackens due to its loop-based structure.

Table 1. FOPTD models and modeling errors.

System	SVR_foptd model	Skogestad_foptd model	SVR_error	Sko_error
$G_1(s)$	$G_{1(SVR)} = \frac{0.9953}{1.0084s+1}e^{-0.1396s}$	$G_{1(Sko)} = \frac{1}{1.1s+1}e^{-0.1s}$	0.0081	0.0087
$G_2(s)$	$G_{2(SVR)} = \frac{1.9455}{1.4624s+1}e^{-0.1978s}$	$G_{2(Sko)} = \frac{1.5}{1.05s+1}e^{-0.15s}$	0.1072	0.3061
$G_3(s)$	$G_{3(SVR)} = \frac{0.9991}{1.4649s+1}e^{-2.5084s}$	$G_{3(Sko)} = \frac{1}{1.5s+1}e^{-2.5s}$	0.0102	0.0102
$G_4(s)$	$G_{4(SVR)} = \frac{1.0007}{8.5185s+1}e^{-0.9775s}$	$G_{4(Sko)} = \frac{0.625}{4.5s+1}e^{-1.25s}$	0.0185	0.2580

6. References

[1] J.P., Richard, "Time Delay Systems: An Overview of Some Recent Advances and Open Problems", *Automatica*, Vol: 39, Issue: 10, pp. 1667-1694, 2003.

[2] S.V., Drakunov, W., Perruquetti, J.P., Richard, "Delay Identification in Time-Delay Systems Using Variable Structure Observers", *Annual Reviews in Control*, Vol: 30, Issue: 2, pp. 143-158, 2006.

[3] V., Venkatashankar, M., Chidambaram, "Design of P and PI Controllers for Unstable 1st Order Plus Time-Delay Systems", *International Journal of Control*, Vol: 60, No:1, pp.137-144, 1994.

[4] S., Tavakoli, M., Tavakoli, "Optimal Tuning of PID Controllers for First Order Plus Time Delay Models Using Dimensional Analysis", *The fourth International Conference on Control and Automation (ICCA'03)*, Montreal, Canada, 2003.

[5] S., Skogestad, "Simple Analytic Rules for Model Reduction and PID Controller Tuning", *Journal of Process Control*, Vol: 13, pp.291-309, 2003.

[6] V.N., Vapnik "The Nature of Statistical Learning Theory", Springer-Verlag, New York. 1995.

[7] V.N., Vapnik "Statistical Learning Theory", John Wiley and Sons, New York. 1998.

[8] N., Cristianini, J., Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", Cambridge University Press, Cambridge, 2000.

[9] L, Zhao, J, Sun, KR. Butts, "Linear programming support vector regression with wavelet kernel: A new approach to nonlinear dynamical systems identification", *Mathematics and Computers in Simulation*, 2009;79:2051–2063.

[10] S., Iplikci, "A support vector machine based control application to the experimental three-tank system", *ISA Transactions* 2010;49:376-386.

[11] A.J., Smola, B., Schoelkopf, "A tutorial on support vector regression", NeuroCOLT2 Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK. 1998.

[12] J., Ma, J., Theiler, S., Perkins, "Accurate on-line support vector regression", *Neural Computation*, vol.15, p: 2683-2703, 2003.

[13] H., Zhang, X., Wang, C., Zhang, X., Xu, "Modeling Nonlinear Dynamical Systems Using Support Vector Machine", *Proceeding of 4. International Conf. On Machine Learning and Cybernetics*, p:3204-3209, 2005.

Rana Ortac-Kabaoglu was born and grown in Istanbul. After she completed Istanbul University Depeartment of Electrical-Electronics Engineering, she had BSc degree from the same department. She got her PhD. degree from Istanbul Technical University, Control Engineering Department. She has been studying on control theory and its applications. Rana Ortac-Kabaoglu is also working in Istanbul University Electrical-Electronics Engineering as a research assistant.