

An Analysis of the Effects of SVM Parameters on the Dead-Time System Modeling

Nihat Kabaoğlu¹ , Rana Ortaç Kabaoğlu² 

¹Department of Electrical and Electronics Engineering, Istanbul Medeniyet University School of Engineering and Natural Science, Istanbul, Turkey
²Department of Electrical and Electronics Engineering, Istanbul University School of Engineering, Istanbul, Turkey

Cite this article as: N. Kabaoğlu, R. Ortaç Kabaoğlu. "An Analysis of the Effects of SVM Parameters on the Dead-Time System Modeling". *Electrica*, vol. 18, no. 1, pp. 1-5, 2018.

ABSTRACT

Modeling a dead-time system is a common issue in engineering applications. To address this issue, existing research has employed neural networks and fuzzy logic-based intelligent systems. Herein, a dead-time system modeled with the aid of support vector machine regression, which has a good generalization feature, was investigated. The performance of the method proposed herein was examined with different parameters in linear and nonlinear dead-time systems.

Keywords: Modeling, support vector machine, regression

Introduction

Optimal generalization success can be achieved for a learning task with a finite number of training cases, if the machine capacity is correctly balanced, which means that the result is a definitive conclusion on a given training group and the learning ability every single training set excellently. The classic neural network attitude has faced several challenges, especially in applications with few data, in generalizing and generating data-compatible models. The Support Vector Machines (SVM) presented by Vapnik in the 1970s is a combination of statistics, machine learning, Neural Networks (NN) [1, 2]. The formulation includes the Structural Risk Minimization rule. SVM can also carry out a well generalising of classifying and regressing even on little sized data using the kernel function mapping technique [3, 4]. In addition, SVM are becoming increasingly popular due to their many attractive features in theoretical and engineering applications. Unlike NN, SVM can overcome the problem of classifying and regressing via being transformed to the problem of quadratic programming without being deterred by any regional minimum. One other adverse side of NN is not to know exactly to choose the hidden cells numbers. SVM that can be utilized at the nonlinear system modelling, is utilized at many fields of model identification, identification of hidden handwriting, identification of objects, determination of sound, search and characterization of a face, regression, estimation of a function and optimal control areas [5, 6].

The purpose of this work is to present the regression method according to SVM by applying it in order to model of the system with dead-time. The proposed article is arranged as below:

A general information about SVM is given in Chapter 2. The proposed support vector machine regression method and its structure are expressed in Chapter 3, the examples are in Chapter 4. They are followed by conclusion and references sections.

Support Vector Regression

Vapnik found in 1997 that support vector techniques were successfully applied to solve regression problems [1]. Contrary to pattern recognition problems, there are real-value func-

Address for Correspondence:

Nihat Kabaoğlu

E-mail:

nihat.kabaoglu@medeniyet.edu.tr

Received: 12.07.2017

Accepted: 04.01.2018

© Copyright 2018 by Electrica

Available online at

<http://dergipark.gov.tr/ijueee>

DOI: 10.5152/ijueee.2018.1801

tions. Support vector machine is an effective way to model linear and nonlinear systems.

The training data set to be used to learn the function $f(x)$ with input-output relation is as follows:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \quad x \in R^N, \quad y \in R \quad (1)$$

Here, input data, X , stand for N -size vectors and generally system responds, Y , stand for scalar values. The function to be estimated in the support vector machines is processed as follows. The input/output connection may be modeled as follows at the feature space by the linear regressing model:

$$f(x, w) = w^T x + b \quad (2)$$

Several loss functions are used in regression operations. The ϵ -tolerance loss function used here is given as follows:

$$Y_\epsilon = \begin{cases} 0 & \text{if } |y - f(x, w)| < \epsilon \\ |y - f(x, w)| - \epsilon & \text{otherwise} \end{cases} \quad (3)$$

It is the objective to minimize the observational and also the experimental risk that symbolizes the whole of errors at SVM algorithm formula and $\|w\|^2$ at once. As a result, a regressing hyper-plane $f(x, w)$ can be detected in x and x^* slack variables via minimization as follow:

$$R(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi + \xi^*) \quad (4)$$

The trade-off parameter between error and $\|w\|$ weight vector form is C and it is chosen by the user. ϵ value, another parameter, which will be chosen by the user, like C , and defines the size of the ϵ -tube. This optimization problem with restriction is resolved via installing a secondary Lagrangian as follows:

$$\max_{\alpha, \alpha^*} W(\alpha, \alpha^*) = -\epsilon \cdot \sum_{i=1}^n (\alpha_i^* + \alpha_i) + \sum_{i=1}^n (\alpha_i^* - \alpha_i) y_i - \frac{1}{2} \sum_{i=1, j=1}^n (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) x_i^T x_j \quad (5)$$

The restrictions are as the following.

$$\begin{aligned} \sum_{i=1}^n \alpha_i^* &= \sum_{i=1}^n \alpha_i \\ 0 \leq \alpha_i^* &\leq C \quad i = 1, \dots, n \\ 0 \leq \alpha_i &\leq C \quad i = 1, \dots, n \end{aligned} \quad (6)$$

By combining the results of optimal regression hyper-plane, it is rewritten as:

$$z = f(x, w) = w^T x + b = \sum_{i=1}^n (\alpha_i^* - \alpha_i) \langle x_i, x \rangle + b \quad (7)$$

In non-linear SVM regression, it is passed through from nonlinear input space to a linear features space using "kernel functions", which meet Mercer conditions and are symmetric final positive. Any kernel function, which corresponds to any inter-

nal multiplication to be done, in this space having high-degree size has been defined in the present space [7].

$$K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \quad (8)$$

Using kernel function, need for making multiplications in (7) is eliminated. In the feature space, which is linear, the algorithm for linear regression by SVM is applied. Thus, $f(x, w)$ hyperplane of regression, which is linear in the features space, will create nonlinear regressing hyper-plane in the specific input space. The formulation for non-linear regression is rewritten as follows:

$$f(x, w) = \sum_{i=1}^n (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (9)$$

The most popular kernel functions, the polynom kernel function, Radial Base kernel Function (RBF) have been used and tested in this study.

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

$$\text{RBF: } K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Modeling with SVM-Regression

The proposed form of the modelling of the dynamic system using the SVM-Regression (SVM-R) is shown in Figure 1.

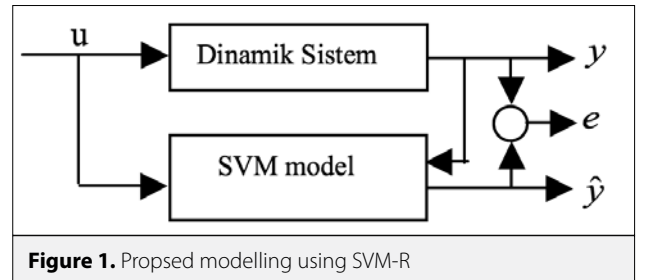


Figure 1. Proposed modelling using SVM-R

where u and y 's refer system input and outputs. y is the actual one belonging to system given. \hat{y} is the one of the SVM-R modeling, e refers the error inbetween outputs of the actual and the modelled systems. This is a modeling error and also depends on the approximate ability.

It is presumed that the structure of the actual system or its mathematical model is not known. SVM modeling data set is planned as to input - output values batch of the system. Usually, the outputs of each systems at the $k+1$ time, can be stated as a function with former output specimens, and-with former input specimens.

$$y(k+1) = f(y(k), \dots, y(k-n+1), u(k-d), \dots, u(k-d-m+1), \eta(k), \dots, \eta(k-l)) = f(x(k)) \quad (10)$$

where, $\eta(k)$ is uncertainty (including some noises, unmodeled dynamics... etc), d is the time lag being in samples (when T_s is the sampling period, the system dead time is $d.T_s$) n, m, l

is the maximum delay of the output, input and noise respectively, and $f(\cdot)$ is a nonlinear function. To enter the SVM, the vector is as follows:

$$x(k)=[y(k),\dots,y(k-n+1),u(k-d),\dots,u(k-d-m+1),\eta(k),\dots,\eta(k-l)]^T \quad (11)$$

where $x \in R^{n+m+l}$, $k = d+1, \dots, d+N$. $y(k+1)$ is output. $\hat{y}(k+1)$ refers predicted output value at time $(k+1)$. As a result, this nonlinear system model is rewritten as follows:

$$\hat{y}(k+1)=\sum_{i=1}^{SVs}(\alpha_i^*-\alpha_i).K(x_i,x(k))+b \quad (12)$$

Modeling algorithm with SVM-R

Stage 1: Create training data set $\{(\bar{x}_i, y_i), i=1, \dots, N\}$. The number of training data pairs in the group is N.

Stage 2: Parameter and kernel function selections depending on user are done. (C and ϵ)

Stage 3: SVM is trained with training data.

Stage 4: Various test and input data are used for testing of the performance of the SVM based model.

Examples

Linear system modeling example

A first order dead system is seen as below [8].

$$y(k+1)=0.9512.y(k)+0.07316.u(k-d) \quad (13)$$

- Sample period is 0.1s., delay step is $d=5$. The design parameters are selected and applied as follows.
- $u(t)=1+\sin(10.t+5)+\sin(2.t+1)$ is the selected sign as input. The system is run for a period of time to collect input - output training data.
- $X(k)=[y(k.) y(k-1) u(k.) u(k-1)]^T$, $Y(k)=y(k+1)$ are the selected input and output of the SVM.
- User defined parameters are selected like C: 200, also ϵ : 0.01. Every training set involves 150 components.
- This method is applied for both RBF and also polynomial kernel function. RBF is tested for $\sigma=0.5, 2, 3, 4$ and polynomial kernel parameter p is selected as $p=1, 2, 3, 4$. In addition, 3 different test inputs are applied for comparison (Table 1). The outputs for different inputs are shown in Figure 2, 3.

Nonlinear system modeling example

The method was applied to a nonlinear dead time system after it was applied to a linear system. Here a nonlinear second order dead time system is considered.

The system considered with 3 varied dead times $T=0.25\text{sec.}, 1\text{sec.}, 2\text{sec.}$, are given like follows [9].

$$\ddot{y}(t)+\dot{y}(t.)+y(t.)+y^3(t)=u(t-T) \quad (14)$$

The design parameters are selected and applied as follows.

- $u(t)=1+\sin(10.t+5)+\sin(2.t+1)$ is the selected sign as input. the system is run for a period of time to collect input - output training data.
- $X(k)=[y(k.) y(k-1) u(k.) u(k-1)]^T$, $Y(k)=y(k+1)$ are the selected input and output of the SVM.
- User defined parameters are selected like C: 200, ϵ : 0.001. Every training set involves 200 components.
- This method is applied for both RBF and also polynomial kernel function. RBF is tested for $\sigma=9, 10$ and polynomial kernel parameter p is selected as $p=1, 2$. In addition, 3 different test inputs are applied for comparison (Table 2). The outputs for different inputs are shown in Figure 4, 5.

Conclusion

In this work, a method based on modeling of dead time systems using SVM regression algorithm is presented. In the proposed method, any mathematical or theoretical forms of the system do not need to be known. The system is modeled by the SVM-R method constituted via a loss function. The SVM regression has high generalization potency and gives very good results even with a small number of training data. As it guarantees a global minimum it is also a very dependable approach. The method has been applied on linear time and

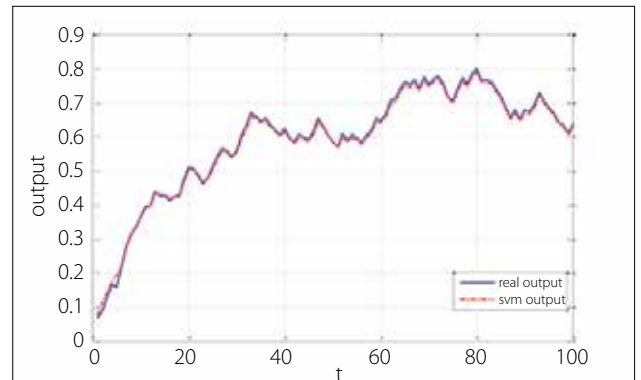


Figure 2. Polinomial kernel ($p=3$), $u=\text{random}$

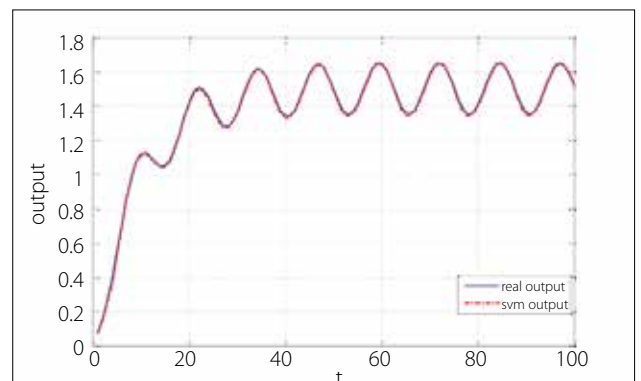


Figure 3. RBF kernel ($\sigma=0.5$), $u=\exp(-t)+1+\sin(5t+5)$

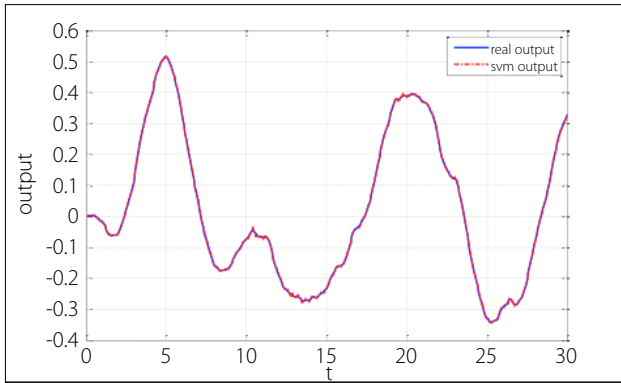


Figure 4. Polinomial kernel ($p=3$), u =random

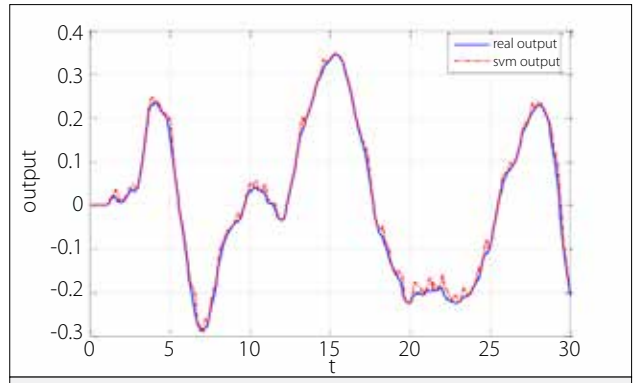


Figure 5. RBF kernel ($\sigma=9$), u =random

Table 1. Training and testing errors for different kernel parameters and different inputs (for linear system)

Kernel function	Kernel parameter	Training error	Testing error	Step error	Error of $u=\exp(-t)+1+\sin(5t+5)$	Error of random signal
Polinomial	$p=1$	0.0047	0.0057	0.0047	0.0052	0.0035
	$p=2$	0.0070	0.0065	0.0073	0.0064	0.0053
	$p=3$	0.0059	0.0062	0.0065	0.0059	0.0084
	$p=4$	0.0057	0.0053	0.0087	0.0080	0.0089
RBF	$\sigma=0.5$	0.0065	0.0069	0.0109	0.0087	0.0136
	$\sigma=2$	0.0058	0.0060	0.0081	0.0084	0.0081
	$\sigma=3$	0.0059	0.0064	0.0083	0.0082	0.0062
	$\sigma=4$	0.0052	0.0057	0.0088	0.0073	0.0072

Table 2. Training and testing errors for different kernel parameters and different inputs (for nonlinear system)

Kernel function	Kernel parameter	Dead time (second)	Training error	Testing error	Step error	Error of $u=\exp(-t)+1+\sin(5t+5)$	Error of 1. random signal	Error of 2. random signal	Error of 3. random signal	Error of 4. random signal
Polinomial	$p=1$	0.25	0.0047	0.0046	0.0018	0.0028	0.0091	0.0098	0.0085	0.0095
		1	0.0049	0.0046	0.0022	0.0031	0.0091	0.0097	0.0080	0.0095
		2	0.0051	0.0048	0.0027	0.0036	0.0094	0.0093	0.0081	0.0099
	$p=2$	0.25	0.0037	0.0035	0.0025	0.0037	0.0093	0.0102	0.0097	0.0099
		1	0.0049	0.0045	0.0028	0.0043	0.0099	0.0108	0.0100	0.0096
		2	0.0052	0.0048	0.0033	0.0038	0.0101	0.0109	0.0101	0.0099
RBF	$\sigma=9$	0.25	0.0062	0.0061	0.0025	0.0035	0.0145	0.0166	0.0122	0.0162
		1	0.0064	0.0062	0.0028	0.0038	0.0140	0.0157	0.0112	0.0155
		2	0.0065	0.0064	0.0032	0.0043	0.0152	0.0145	0.0112	0.0160
	$\sigma=10$	0.25	0.0063	0.0062	0.0027	0.0036	0.0128	0.0149	0.0112	0.0145
		1	0.0066	0.0065	0.0029	0.0040	0.0130	0.0144	0.0106	0.0143
		2	0.0067	0.0066	0.0033	0.0045	0.0140	0.0135	0.0106	0.0148

nonlinear dead time systems, and a very high modeling and generalization success has been observed even at various inputs. These results are evident from the small size of the modeling errors. Also, it has been found that as the dead time increased, there was a small increase in modeling errors and the modeling performance was not adversely effected by the change of the system's dead time. This is very important in terms of modeling performance. As Table 2 shows, the error grows as the polynomial level grows for the polynomial kernel, and the error decreases as the variance for the rbf kernel grows. On the other hand, selecting kernel and its parameters is a very important question of the SVM. This can be seen on the tables. the choice of parameters depends on the user, there are still studies focusing on making the right choice [9]. increasing the training data provides a better performance, but this can lead time problems in some complex systems. Various studies have also been done on this topic [10-12].

References

1. V. N. Vapnik. "Statistical Learning Theory", John Wiley and Sons, New York, 1998.
2. V. N. Vapnik. "The Nature of Statistical Learning Theory", Springer-Verlag. New York, 1995. [CrossRef]
3. C. Junli, J. Licheng, "Classification Mechanism of Support Vector Machines", *IEEE Proceedings of ICSP* p. 1556-1559, 2000.
4. A. Smola, B. Scholkopf, "A Tutorial on Support Vector Regression", *NeuroCOLT2 Technical Report NC-TR* pp. 98-30, 1998.
5. L. Guohai, Z. Dawei, X. Haixia, M. Congli "Soft sensor modeling using SVM in fermentation process", *Chinese Journal of Scientific Instrument*, vol. 6, 2009.
6. V. Kecman, "Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models", Cambridge, Mass.: MIT Press, 2001.
7. C. Campbell, "Kernel Methods: A Survey of Current Techniques", *Neurocomputing* vol. 48, pp. 63-84, 2002. [CrossRef]
8. Z. Hongdong, S. Huihe, "A Novel Method of Process Dead-Time Identification: Support Vector Machine Approach", *Proceeding of American Control Conference*, pp. 880- 884, 2004.
9. G. Yan, C. Li, G. Ma, "Parameter selection for support vector machines based on hybrid genetic algorithms", *Journal of Harbin Institute of Technology*, vol.5, 2008.
10. 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*, pp. 3204-3209, 2005.
11. J. Ma, J. Theiler, S. Perkins, "Accurate on-line support vector regression", *Neural Computation*, vol.15, pp. 2683-2703, 2003. [CrossRef]
12. <http://www.sciencedirect.com/science/article/pii/S1226086X13003742> - I.A. M. Ghaedi, M. Hossainpour, A. Ansari, M.H. Habibi, A.R. Asghari "Least square-support vector (LS-SVM) method for modeling of methylene blue dye adsorption using copper oxide loaded on activated carbon" *Journal of Industrial and Engineering Chemistry*, Volume 20, Issue 4, 25 July 2014, pp. 1641-1649.



Nihat Kabaoğlu received the B.S. and M.S. degrees in electronics engineering from İstanbul University, and the Ph.D. degree in electronics and telecommunication engineering from İstanbul Technical University, İstanbul, Turkey, in 1997, 2000, 2005 respectively. After working in different positions at İstanbul University, The Scientific and Technological Research Council of Turkey, Kadir Has University and Maltepe University, he joined to İstanbul Medeniyet University as an associate professor in 2015. His general research interests cover estimation theory, statistical signal processing, communication theory, wireless sensor networks and array processing. His current research areas are focused on wireless communication concepts with specific attention to channel estimation for spread-spectrum and multicarrier systems, on beamforming design for next generation systems and on optimal power allocation in wireless sensor networks.



Rana Ortaç Kabaoğlu received the B.S. and M.S. degrees in electrical and electronics engineering from İstanbul University, and the Ph.D. degree in control engineering from İstanbul Technical University, İstanbul, Turkey, in 1997, 2001, 2010 respectively. At 1998, she joined to İstanbul University, electrical and electronics engineering department as a reseach assistant. She has been working as an associated professor at the same department since 2015. Her research areas are control systems, modelling, intelligent systems, fault dedection and control.