

Identification of Auto-Regressive Exogenous Models Based on Twin Support Vector Machine Regression

Mujahed Aldhaifallah¹, K. S. Nisar²

¹. Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Kingdom of Saudi Arabia

². Department of Mathematics, Salman bin Abdulaziz University, Wadi Al Dawaser, Kingdom of Saudi Arabia
mujahed@kfupm.edu.sa

Abstract: In this paper a new algorithm to identify Auto-Regressive Exogenous Models (ARX) based on Twin Support Vector Machine Regression (TSVR) has been developed. The model is determined by minimizing two ϵ insensitive loss functions. One of them determines the ϵ_1 -insensitive down bound regressor while the other determines the ϵ_2 -insensitive up-bound regressor. The algorithm is compared to Support Vector Machine (SVM) and Least Square Support Vector Machine (LSSVM) based algorithms using simulation and experimental data.

[Mujahed Aldhaifallah and K. S. Nisar. **Identification of Auto-Regressive Exogenous Models Based on Twin Support Vector Machine Regression.** *Life Sci J* 2013;10(4):3049-3054]. (ISSN:1097-8135).
<http://www.lifesciencesite.com>. 406

Keywords: Auto-Regressive Exogenous Models, Identification, Twin Support Vector Machines

1. Introduction

The system identification problem is as follows: Given observations of the inputs u and outputs y from a dynamic system

$$\varphi_t = [\{u, y_1\} \{u_2, y_2\} \cdots \{u_t, y_t\}]$$

Obtain a relationship between the past

observations φ_t and the current output(s) y_t :

$$y_t = g(\varphi_t, \theta) + v_t \quad (1)$$

where θ is a vector of unknown parameters. The term v_t represents additive "noise" in the output. This includes any information in the output sample y_t that cannot be predicted using only the past data. This noise may be the result of unmeasured inputs driving the system, or it could be measurement noise, or both. Let $\hat{y}(t)$ be an estimate of the output. A reasonable objective is to make the error $y_t - \hat{y}_t(\theta)$ "small" in some sense, so that the function $g(\varphi_{t-1}, \theta)$ could be considered a good predictor of y_t .

Generally an identification experiment is started by exciting the system using some sort of input signal u_t over a time interval. This signal and its response y_t are usually recorded and processed in a computer so the time t is assumed to be discrete. Next, a set of candidate models $M(\theta)$ is determined and a suitable criterion, a function of the difference between the actual output y_t and the predicted output \hat{y}_t , is chosen to assess the model fitness. A model is then selected from the candidate set, most often, by minimizing the chosen criterion. Finally, the model obtained is tested to see whether it is a valid representation of the

system. If this is not the case, the model structure must be revised and the identification procedure is repeated. The model could be either linear or nonlinear [Ljung, 1999].

A system is said to be linear if the net response caused by two or more combined excitations is the sum of the responses caused by each stimulus individually [Ljung, 1999]. In continuous time, linear systems are usually described using ordinary differential equations. In discrete-time, difference equations are used instead. In this paper, discussion will be restricted to discrete-time linear models.

In a linear difference equation model, the relationship between the input sequence $\{u_t\}$ and the output sequence $\{y_t\}$ is described by

$$y_t + a_1 y_{t-1} + \cdots + a_{na} y_{t-na} = b_1 u_{t-1} + b_2 u_{t-2} + \cdots + b_{nb} u_{t-nb} \quad (2)$$

which can be rewritten in more compact form

$$A(Z)y_t = B(Z)u_t \quad (3)$$

where $A(z) = 1 + a_1 z^{-1} + \cdots + a_{na} z^{-na}$ and z^{-1} backward shift operator. In (3), the relation between inputs and outputs is assumed to be deterministic [Soderstrom and Stoica, 1989].

This assumption is not always practical because there are usually modeled dynamics, v_t , which should be included the model

$$A(Z)y_t = B(Z)u_t + v_t \quad (4)$$

The way of representing v_t allows for many descriptions. If $v_t = e_t$, where e_t is a white noise, then one obtains the following model

$$A(Z)y_t = B(Z)u_t + e_t \tag{5}$$

Such model is known as the auto-regressive with exogenous input (ARX) model [Mujahed Aldhaifallah, 2009].

ARX model can be identified using least square algorithm which will give the maximum likelihood estimate if the residuals have Gaussian distribution with zero mean and σ^2 variance. However, it is not the case if the residuals distribution is different. In [Lindgren,2005], author applied least square to identify ARX model to fit Hair Dryer data. Support vector machine (SVM) [Vapnik, 1998] regression is another alternative. SVM outperform ordinary least square algorithm especially if the noise is non Gaussian. In [Rojo-Alvarez et al, 2004] an algorithm to identify ARMA model based on support vector machine has been developed. The SVM based algorithms are computationally heavy because they have at least two groups of constraints, each of which shows that more training samples locate in the given ϵ -insensitive regain.

In this paper, a new algorithm to identify Auto-Regressive Exogenous Models (ARX) based on Twin Support Vector Machine Regression (TSVR) was proposed. The outline of this paper is as follows: TSVM theory will be reviewed in Section 2. In Section 3, an algorithm for the identification of ARX models based on twin support vector machine is proposed. Section 4 presents illustrative examples to test the proposed algorithm. In Section 5, concluding remarks are given.

2. Standard Twin Support Vector Machine regression (TSVR):

Twin Support Vector Regression (TSVR) is obtained by solving the following pair of quadratic programming problems (QPPs) [Xinjun, 2010]

$$\begin{aligned} \min & \frac{1}{2}(Y - e\epsilon_1 - A\omega_1 - eb_1)^T \\ & (Y - e\epsilon_1 - A\omega_1 - eb_1) + C_1\epsilon_1 e^T \xi \end{aligned} \tag{6}$$

s.t. $Y - (A\omega_1 + eb_1) \geq e\omega_1 - \xi, \quad \xi \geq 0$

$$\begin{aligned} \min & \frac{1}{2}(Y - e\epsilon_2 - A\omega_2 - eb_2)^T \\ & (Y - e\epsilon_2 - A\omega_2 - eb_2) + C_2\epsilon_2 e^T \eta \end{aligned} \tag{7}$$

s.t. $Y - (A\omega_2 + eb_2) \geq e\omega_2 - \eta, \quad \eta \geq 0$
 where $C_1, C_2 > 0, \epsilon_1, \epsilon_2 \geq 0$ are parameters, and ξ, η are slack vectors. The TSVR algorithms find given the training data points (A, Y) , the functions $f_1(x) = w^1 x + b_1$ determines the ϵ_1 -insensitive down

bound regressor, and $f_2(x) = w^2 x + b_2$ determines the ϵ_2 -insensitive up-bound regressor. The end regressor is computed as the mean of these two functions. The geometric interpretation is given in Fig.1 the objective function of (6) or (7) is the sum of the squared distances from the shifted functions $y = w^1 x + b_1 + \epsilon_1$ or $y = w^2 x + b_2 - \epsilon_2$ to the training points. Therefore, minimizing it leads to the function $f_1(x)$ or $f_2(x)$. The constraints require the estimated function $f_1(x)$ or $f_2(x)$ to be at a distance of at least ϵ_1 or ϵ_2 from the training points. That is, the training points should be larger than the function $f_1(x)$ at least ϵ_1 , while they should be smaller than the function $f_2(x)$ at least ϵ_2 . The slack variables ξ and η are introduced to measure the error wherever the distance is closer than ϵ_1 or ϵ_2 . The second term of the objective function minimizes the sum error variables, thus attempting to over fit the training points.

To find the solution of (6) and (7), their dual problems are needed to be derived. The optimization of $J(w, \xi)$ just described is the primal problem for regression. To formulate the corresponding dual problems, the Lagrangian functions L is defined. Then, L is minimized with respect to the weight w and slack variable ξ and maximized with respect to the Lagrange multipliers. By carrying out this optimization w can be written in terms of Lagrange multipliers. Finally, substituting the value w and simplifying with the help of Karush-Kuhn-Tucker (KKT) [Boyd and Vandenberghe, 2004] the following dual problem is obtained.

$$\begin{aligned} \max & -\frac{1}{2} \alpha_1^T G (G^T G)^{-1} G^T \alpha_1 \\ & + f_1^T \alpha_1^T G (G^T G)^{-1} G^T \alpha_1 - f_1^T \alpha_1 \end{aligned} \tag{8}$$

s.t. $0 \leq \alpha \leq C_1 e$

where

$$G = [A \quad e], f_1 = Y - e\epsilon_1$$

similarly, one can consider the problem (7) and obtain its dual as

$$\begin{aligned} \max & -\frac{1}{2} \alpha_2^T G (G^T G)^{-1} G^T \alpha_2 \\ & + f_2^T \alpha_2^T G (G^T G)^{-1} G^T \alpha_2 - f_2^T \alpha_2 \end{aligned} \tag{9}$$

s.t. $0 \leq \alpha \leq C_2 e$

$$G = [A \quad e], f_2 = Y - e\epsilon_2 \text{ where}$$

then

$$-G_1^T f_1 + G^T G u_1 + G^T \alpha_1 = 0 \tag{10}$$

Solving for u_1 results in

$$u_1 = \begin{bmatrix} \omega_1 \\ b_1 \end{bmatrix} = (G^T G)^{-1} G^T (f_1 - \alpha_1) \tag{11}$$

Note that $G^T G$ is always positive semi definite. It is possible that it may not be well conditioned in some situations. To overcome this case, there is a regularization term σI , where σ is a very small positive number, such as $\sigma = 1e - 7$. Therefore (11) is modified to

$$u_1 = (G^T G + \sigma I)^{-1} G^T (f_1 - \alpha_1) \tag{12}$$

similarly,

$$u_2 = \begin{bmatrix} \omega_2 \\ b_2 \end{bmatrix} = (G^T G)^{-1} G^T (f_2 + \alpha_2) \tag{13}$$

Note that in the duals (8) and (9), the inversion of matrix $G^T G$ of size $(n + 1) \times (n + 1)$ should be computed [7]. Once the vectors u_1 and u_2 are known from (12) and (13) the two up and down bound functions are obtained. Then the estimated regressor is constructed as follows

$$f(x) = \frac{1}{2} (f_1(x) + f_2(x)) \tag{14}$$

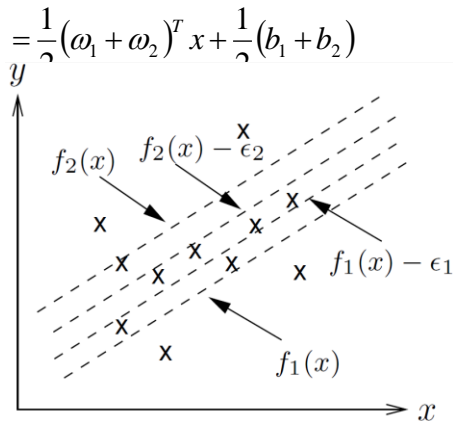


Figure 1: The geometric interpretation of TSVR.

2.1. TSVM Identification of ARX Models

Autoregressive with exogenous input model shown in Fig. 2 is used widely to represent linear systems. Assume that a system can be described as an ARX

$$y_t = \sum_{i=1}^n a_i y_{t-i} + \sum_{j=0}^m b_j u_{t-j} + e_t \tag{15}$$

where $u_t, y_t \in \mathbb{R}$ are positive input and output respectively for $t = r, \dots, N$, where $r = \max(m, n) + 1$. The noise e_t is assumed to be white m and n denote the order of the numerator and denominator in the

transfer function of the ARX model.

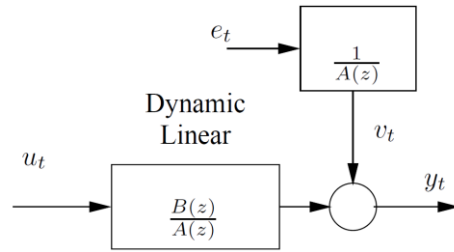


Figure 2: Block diagram of ARX model.

$$\min_{a_i, b_j, \xi_1, \xi_2, \eta_1} \frac{1}{2} \xi_1^T \xi_2 + C_1 e^T \eta_1 \tag{16}$$

s.t.

$$y_t - \sum_{i=1}^n a_{1i} y_{t-i} - \sum_{j=0}^m b_{1j} u_{t-j} - \epsilon_1 = \xi_{1t}$$

$$y_t - \sum_{i=1}^n a_{1i} y_{t-i} - \sum_{j=0}^m b_{1j} u_{t-j} - \epsilon_1 \geq -\eta_1, \eta_1 \geq 0$$

$$\min_{a_2, b_2, \xi_2, \eta_2} \frac{1}{2} \xi_2^T \xi_2 + C_2 e^T \eta_2 \tag{17}$$

s.t.

$$y_t - \sum_{i=1}^n a_{2i} y_{t-i} - \sum_{j=0}^m b_{2j} u_{t-j} - \epsilon_2 = \xi_{2t}$$

$$y_t - \sum_{i=1}^n a_{2i} y_{t-i} - \sum_{j=0}^m b_{2j} u_{t-j} - \epsilon_2 \geq -\eta_2, \eta_2 \geq 0$$

Notice that (16) and (17) are identical to the standard TSVR objectives, (6) and (7). The constraints in (16) are derived by modifying (6) to include the dynamics of the ARX model. The Lagrangian of (16) is defined as

$$\begin{aligned} L(\Theta_1, \xi_1, \eta_1, \alpha_1, \beta_1) = & \frac{1}{2} (Y - y a_1 - u b_1 - e \epsilon_1)^T \\ & (Y - y a_1 - u b_1 - e \epsilon_1) + C_1 e^T \eta_1 \\ & - \alpha_1^T (Y - y a_1 - u b_1 - e \epsilon_1 + \eta_1) - \beta_1^T \eta_1 \\ = & \frac{1}{2} (Y - \Theta_1 - e \epsilon_1)^T \\ & (Y - \Theta_1 - e \epsilon_1) + C_1 e^T \eta_1 \\ & - \alpha_1^T (Y - \Theta_1 - e \epsilon_1 + \eta_1) - \beta_1^T \eta_1 \end{aligned} \tag{18}$$

where α_1 and β_1 are Lagrange multiplier vectors with

$$y = \begin{bmatrix} y_{r-1} & y_{r-2} & \cdots & y_{r-n} \\ y_r & y_{r-1} & \cdots & y_{r-n+1} \\ \vdots & & & \\ y_{N-1} & y_{N-2} & \cdots & y_{N-n} \end{bmatrix}$$

$$u = \begin{bmatrix} u_r & u_{r-1} & \cdots & u_{r-m} \\ u_{r+1} & u_r & \cdots & u_{r-m+1} \\ \vdots & & & \\ u_N & u_{N-1} & \cdots & u_{N-m} \end{bmatrix}$$

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$G = \begin{bmatrix} y & u \end{bmatrix}$$

$$Y = \begin{bmatrix} y_r \\ y_{r+1} \\ \vdots \\ y_N \end{bmatrix} \quad \text{and} \quad \Theta = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$$

Now, given training data, the TSVR algorithm identifies the function

$$y_t = \sum_{i=1}^n a_{1i} y_{t-i} + \sum_{j=0}^m b_{1j} u_{t-j}$$

which determines the ϵ_1 -insensitive down bound regressor and

$$y_t = \sum_{i=1}^n a_{2i} y_{t-i} + \sum_{j=0}^m b_{2j} u_{t-j}$$

which determines the ϵ_2 -insensitive down bound regressor

$$\frac{\partial L}{\partial \Theta_1} = 0 \Rightarrow$$

$$\Theta_1 + G^T (\Theta_1 - G^T Y + G^T e \epsilon_1 + G^T \alpha_1) = 0 \quad (19)$$

$$\frac{\partial L}{\partial \eta_1} = 0 \Rightarrow C_1 e - \alpha_1 - \beta_1 = 0 \quad (20)$$

$$Y - \Theta_1 - e \epsilon_1 + \eta_1 \geq 0, \quad \eta_1 \geq 0 \quad (21)$$

$$\alpha_1^T (Y - \Theta_1 - e \epsilon_1 + \eta_1) = 0 \quad (22)$$

$$\beta_1^T \eta_1 = 0, \quad \beta_1 \geq 0, \quad \alpha_1 \geq 0 \quad (23)$$

Since $\beta \geq 0$

$$0 \leq \alpha_1 \leq C_2 e \quad (24)$$

Substituting (19) - (24) in to (18) Lagrangian can be written as

$$L(\Theta_1, \xi_1, \eta_1, \alpha_1, \beta_1) = \quad (25)$$

$$\frac{1}{2} \left\{ Y - (G^T G)^{-1} (G^T Y - G^T e \epsilon_1 - G^T \alpha_1) - e \epsilon_1 \right\}$$

$$+ C_2 e^T \eta_1 - \alpha_1^T Y + \alpha_1^T G (G^T G)^{-1}$$

$$(G^T Y - G^T e \epsilon_1 - G^T \alpha_1) + \alpha_1^T e \epsilon_1 - \alpha_1^T \eta_1$$

$$- C_1 e^T \eta_1 + \alpha_1^T \eta_1$$

Let

$$f_1 = Y - e \epsilon_1 \quad (26)$$

Simplifying (25) results in the following dual QPP

$$\min \frac{1}{2} G (G^T G)^{-1} G^T \alpha_1$$

$$- f_1^T G (G^T G)^{-1} G^T \alpha_1 + f_1^T \alpha_1 \quad (27)$$

Similarly problem (17) can be mapped to the dual space to get

$$\min \frac{1}{2} G (G^T G)^{-1} G^T \alpha_2$$

$$- f_2^T G (G^T G)^{-1} G^T \alpha_2 + f_2^T \alpha_2 \quad (28)$$

Now, Θ_2 can be computed as

$$\Theta_2 (G^T G)^{-1} (G^T Y - G^T e \epsilon_2 - G^T \alpha_2) \quad (29)$$

Once the vectors Θ_1 and Θ_2 are known from (19) and (29), the two up and down bound functions are obtained. Then, the regressor is obtained as follows:

$$y(t) = \frac{1}{2} \left(\sum_{i=1}^n a_{1i} y_{t-i} + \sum_{i=1}^n a_{2i} y_{t-j} \right.$$

$$\left. + \sum_{j=1}^m b_{1j} u_{t-j} + \sum_{i=1}^n b_{2j} u_{t-j} \right) \quad (30)$$

2.2. Algorithm

The algorithm for ARX identification using TSVM can be summarized as follows

(i) Obtain estimates for α_1 and α_2 by solving (27) and (28).

(ii) Use (19) and (29) to compute Θ_1 and Θ_2 .

(iii) Use (30) to find the estimated regressor.

3. Illustrative examples:

3.1. Example 1

Consider the example presented in [Rojo-Alvarez, et al., 2004] which used the following ARX model

$$y_t = 0.03y_{t-1} - 0.01y_{t-2} + 3x_t - 0.5x_{t-1} + 0.2x_{t-2} + e_t$$

The input x_t is white, Gaussian noise sequence with unit variance. Four types of equation noise $\{e_t\}$ =0 were used; namely Gaussian noise with zero mean and 0.1 variance, Gaussian noise with zero mean and 0.2 variance, uniformly distributed noise in [0,0.1], and uniformly distributed noise in [0,0.2]. The number of data points generated is 400. However identification was performed using the first 300 points of data while the remaining were used for testing and validation. The TSVM algorithm presented in Section 4, the SVM approach in [Rojo-Alvarez, et al., 2004], and LSSVM were employed to the training data. All regression methods are implemented in MATLAB 7.7.0 on windows 7 professional running on a PC with system configuration Intel (R) Core (TM)2 Duo. To compare the CPU time and the accuracy of the three algorithms MATLAB "qp.m" function was used to solve the proposed TSVR and SVR quadratic problems, and the "inv.m" function was used to solve the LSSVR problem. The accuracy criteria which are used to evaluate the algorithms performances are sum squared error (SSE) of testing samples, sum squared deviation (SST) of testing samples, sum squared deviation that can be explained by the estimator (SSR) ratio between sum squared error and sum squared deviation of testing samples (SSE/SST).

The hyper parameters governing the three algorithms were selected based on cross validation test, where the values that produced the best performance on the validation set data were chosen, by computing the sum of square error between the true system and the models outputs.

Table.1 presents the average results of TSVR,SVR, and LSSVR with 100 independent runs, in which four different types of noises are used. It's clear that our TSVR algorithm outperforms the other algorithm in terms of SSE, SSE/SST, and SSR/SST values when uniform distributed noise is present. This indicates that TSVR can fit the real system with fairly small regression errors .By comparing the training CPU time of these three methods, its clear that LSSVR is the fastest learning method among them with almost equal speed of TSVR in some cases. Moreover, TSVR and LSSVR is almost 20 times faster than the SVR. This is because TSVR is obtained by solving two smaller sized QPPs without any equality constraints compared with SVR.

3.2. Example 2:

To test performance of the proposed algorithm on experimental data, the Hair dryer system presented in [4] is considered. The data were downloaded from the DAISY data base for system identification, see [9]. The data were created as follows: air is fanned through a tube and heated at the inlet. A random binary sequence of voltage were generated and fed to the heating device (a mesh of resistor wires). The output is the outlet air temperature. Sample of 1000 data of input and output were generated at the rate of 1 Hz for 1000 s. The first 200 samples were reserved for validation and the rest for estimation.

The hyper parameters governing the three algorithms were selected based on cross validation test. An ARX model of order $n = 1, m=6$ were fitted for the data using the three algorithms; namely TSVR, SVR, and LSSVR. Fig. 3 shows the first 200 points of the real data together with the TSVR estimate, SVR estimate, and LSSVR estimate. It's clear from Table .2 that the TSVR gave the best results in terms of all accuracy indices in addition to the best fit criterion. Also, the computation time of TSVR is reasonable compared to the SVR algorithm which was almost 8 times slower than the LSSVR and TSVR algorithms for the same reason mentioned in the previous example.

Table 1: Comparisons of TSVR , SVR and LSSVR on the regression of simulation example with different types of noises.

Noise	Regressor	SSE	SSE/SST	SSR/SST	CPU Time
N(0,0.1)	TSVR	31.2±36.2	0.26±0.16	0.97±0.2	0.02
	SVR	31.6±36.6	0.25±0.16	0.96±0.2	0.5
	LSSVR	31.1±36.1	0.25±0.16	0.97±0.2	0.02
N(0,0.2)	TSVR	32.8±20.2	0.44±0.28	0.68±0.3	0.03
	SVR	32.8±20.3	0.44±0.28	0.67±0.3	0.57
	LSSVR	32.8±20.3	0.44±0.28	0.67±0.3	0.01
U[0,0.1]	TSVR	300.8±359	2.33±1.01	2.75±1.4	0.03
	SVR	387.8±448	3.17±1.67	3.50±1.9	0.48
	LSSR	396.1±459	3.20±1.66	3.53±1.9	0.02
U[0,0.2]	TSVR	580.9±725	3.90±1.74	4.61±2.1	0.03
	SVR	714.3±874	5.02±2.50	5.70±2.9	0.49
	LSSR	714.6±874	5.03±2.50	5.70±2.9	0.02

Table 2: Comparisons of TSVR , SVR and LSSVR on the regression of Hair Dryer example.

Dataset	Regressor	SSE	SSE/SST	SSR/SST	Best Fit	CPU Time
Hair Dryer	TSVR	3.71	0.0198	1.0415	86	0.235
	SVR	4.97	0.0265	1.0541	84	1.966
	LSSVR	4.99	0.0266	1.0544	84	0.116

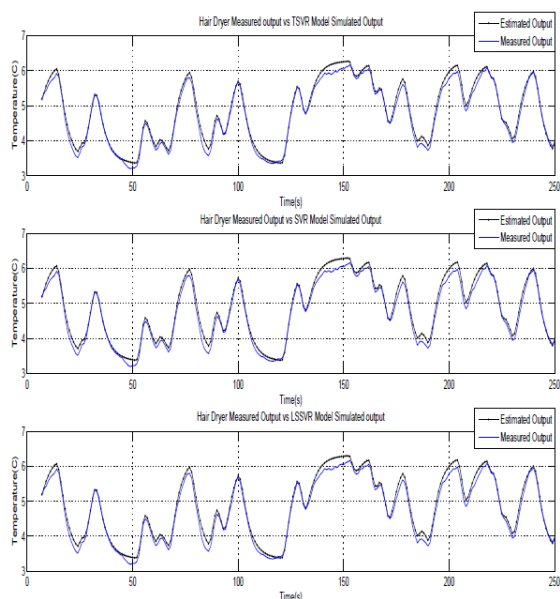


Figure 3: Predictions of TSVR, SVR, and LSSVR on the hair dryer dataset

4. Conclusion

In this paper a new algorithm to identify Auto-Regressive Exogenous Models (ARX) based on Twin Support Vector Machine Regression (TSVR) has been developed. The algorithm was compared to Support Vector Machine (SVM) and Least Square Support Vector Machine (LSSVM) based algorithms using simulation and experimental data. It is clear from examples that the proposed algorithm outperforms SVR and LSSVR in terms of accuracy. Moreover, the CPU time spent by the TSVR algorithm is much less than the time spent by SVR algorithm. That difference is due to the simplicity of TSVR formulation where it consists of solving two simple QPP without equality constraints compared with SVR. In conclusion, fast and accurate solution has been gained in this algorithm.

Acknowledgement

The authors would like to acknowledge the support provided by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project No. FT131015 and the

Deanship of Scientific Research at Salman bin Abdulaziz University.

Corresponding Author:

Dr. Mujahed Aldhaifallah
 Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Kingdom of Saudi Arabia
 E-mail: mujahed@kfupm.edu.sa

References

1. Ljung, L., System identification : theory for the user, Upper Saddle River, NJ Prentice Hall PTR 1999.
2. Soderstrom, T and Stoica, P. System Identification 1989, Upper Saddle River, NJ Prentice Hall PTR 1989.
3. Aldhaifallah, Mujahed. Nonlinear Systems Identification Using Support Vector Machines, PhD thesis, University of Calgary, Calgary, Canada 2009.
4. Lindgren, D. Projection Techniques for Classification and Identification, PhD thesis, Linkoping University, Linkoping, Sweden. Linkoping Studies in Science and Technology, Dissertations (2005); 915.
5. Vapnik, V.N., Statistical Learning Theory 1998, New York: New York John Wiley & Sons, Inc. (US) 1998.
6. Rojo-Alvarez, J.L., et al., Support vector method for robust ARMA system identification. IEEE Transactions on Signal Processing 2004;52(1):155-64.
7. Xinjun, P., TSVR: An efficient Twin Support Vector Machine for regression. Neural Networks, 2010;23:365-372.
8. Boyd, S. and Vandenberghe, L., Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.
9. De Moor, B. Daisy: Database for the identification of systems. Department of Electrical Engineering, ESAT/SISTA. K.U. Leuven, Belgium, <http://www.esat.kuleuven.ac.be/sista/daisy>. Data set name: Hair Dryer, Mechanical Systems 2004:96-006.