

Research Article

Twin Support Vector Machine Method for Identification of Wiener Models

Mujahed Al-Dhaifallah

King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Correspondence should be addressed to Mujahed Al-Dhaifallah; mujahed@kfupm.edu.sa

Received 7 January 2015; Revised 11 April 2015; Accepted 16 April 2015

Academic Editor: Shiliang Sun

Copyright © 2015 Mujahed Al-Dhaifallah. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Twin support vector regression is applied to identify nonlinear Wiener system, consisting of a linear dynamic block in series with static nonlinearity. The linear block is expanded in terms of basis functions, such as Laguerre or Kautz filters, and the static nonlinear block is determined using twin support vector machine regression. Simulation of a control valve model and pH neutralization process have been presented to show the features of the proposed algorithm over support vector machine based algorithm.

1. Introduction

The goal of system identification is to find a model, within a selected class of models, that produces the best predictions of a system's output. In general, one forms a cost function that depends on some norm of the prediction errors and finds the model that minimizes this cost function. Since the model is an approximation to the true system, there is a trade-off between the complexity of the structure of the model and the accuracy of its predictions. In many cases, linear models can be used to produce accurate predictions of a system behavior, particularly, if it is restricted to operating within a narrow region. However, if the model is required to cover a broader operating region, then a nonlinear model may be required [1]. Block structured models, cascades of static nonlinearities and dynamic linear systems, are often a good trade-off as they can represent some dynamic nonlinear systems very accurately but are nonetheless quite simple. Common nonlinear models of this type are the Wiener and Hammerstein models [2]. Many algorithms have been proposed to identify Wiener models [3–11]. As one might notice from these researches, the extensive knowledge about linear time invariant (LTI) system representations was applied to the dynamic linear blocks. On the other hand, finding an effective representation for the nonlinearity is an active area of research. Traditionally, the nonlinearity is represented by a polynomial because it is simple and easy to estimate. However, the problem with

polynomial approximation is that it cannot deal with many common nonlinearities (saturation, threshold, dead zone, etc.). Better approximation of these nonlinearities can be achieved by spline functions. However, spline functions are defined by a series of knot points which must either be chosen a priori or treated as model parameters and included in the (nonconvex) optimization. Neural networks are another tool to approximate nonlinear functions. Their powerful approximation abilities make them attractive. However, the need to specify the neural network topology in terms of the number of nodes and layers and the need to solve nonconvex optimization complicate their implementation. Recently, support vector machines (SVMs) and least squares support vector machines (LS-SVMs) have demonstrated powerful abilities in approximating linear and nonlinear functions [12, 13]. In contrast with other approximation methods, SVMs do not require a priori structural information. Furthermore, there are well established methods with guaranteed convergence (ordinary least squares, quadratic programming) for fitting LS-SVMs and SVMs [14]. One of the common drawbacks of SVM based algorithms is that they are computationally heavy. Recently, a twin support vector machine (TSVM) regression algorithm has been proposed [15]. The formulation of TSVM is very close to classical SVM except that it aims to enclose data points between two parallel planes such that each plane is closer to one class and is as far as possible from the other. Such formulation reduces the computation complexity which

makes the TSVM one of the common methods in machine learning. Lately, many extensions of TSVM classifier have been proposed to improve its performance in certain aspects. A TSVM classifier has been extended to put the TSVM to multitask learning [16]. A generalized framework of TSVM for learning from labeled and unlabeled data was investigated by [17, 18].

Recently, Tötterman and Toivonen [19] have developed a new algorithm to identify Wiener models based on SVM regression, where the linear part is described by a basis filter expansion while the nonlinear part is represented by SVM. In this work, TSVM regression is used to formulate an identification algorithm for Wiener models. Simulation examples are presented to show the virtues of the proposed algorithm over Tötterman's algorithm. The outline of this paper is as follows: TSVM theory is reviewed in Section 2. In Section 3, an algorithm for the identification of Wiener models based on twin support vector machine is proposed. Section 4 presents two illustrative examples to test the proposed algorithm. In Section 5, concluding remarks are given.

2. Twin Support Vector Machines Regression

Twin support vector regression (TSVR) is obtained by solving the following pair of quadratic programming problems (QPPs):

$$\begin{aligned} \min \quad & \frac{1}{2} \left(Y - e\epsilon_1 - K(A, A^T) w_1 - eb_1 \right)^T \\ & \times \left(Y - e\epsilon_1 - K(A, A^T) w_1 - eb_1 \right) + C_1 \epsilon_1 e^T \xi \quad (1) \\ \text{s.t.} \quad & Y - K(A, A^T) w_1 - eb_1 \geq e\epsilon_1 - \xi, \quad \xi \geq 0, \end{aligned}$$

$$\begin{aligned} \min \quad & \frac{1}{2} \left(Y - e\epsilon_2 - K(A, A^T) w_2 - eb_2 \right)^T \\ & \times \left(Y - e\epsilon_2 - K(A, A^T) w_2 - eb_2 \right) + C_2 \epsilon_2 e^T \eta \quad (2) \\ \text{s.t.} \quad & K(A, A^T) w_2 + eb_2 - Y \geq e\epsilon_2 - \eta, \quad \eta \geq 0, \end{aligned}$$

where $C_1, C_2 > 0$, $\epsilon_1, \epsilon_2 \geq 0$ are parameters, ξ, η are slack variables, e is vector of ones, and $K(\cdot, \cdot)$ is a nonlinear kernel function. Given training data points (A, Y) where A and Y represent input and output vectors, respectively, the TSVR algorithm finds two functions $\phi_1(x) = w_1^T K(A, x) + b_1$, which determines the ϵ_1 -insensitive down bound regressor, and $\phi_2(x) = w_2^T K(A, x) + b_2$, which 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 Figure 1. The objective function of (1) or (2) is the sum of the squared distances from the shifted functions $y = w_1^T K(A, x) + b_1 + \epsilon_1$ or $y = w_2^T K(A, x) + b_2 - \epsilon_2$ to the training points. Therefore, minimizing it leads to the function $\phi_1(x)$ or $\phi_2(x)$. The constraints require the estimated function $\phi_1(x)$ or $\phi_2(x)$ 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 $\phi_1(x)$ at least ϵ_1 , while they should be smaller than the function $\phi_2(x)$ at least ϵ_2 . The slack variables ξ and η are introduced to measure the error whenever the distance is

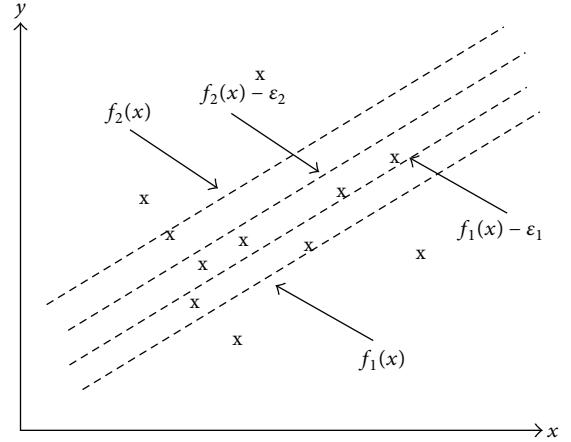


FIGURE 1: The geometric interpretation of TSVR.

closer than ϵ_1 or ϵ_2 . The second term of the objective function minimizes the sum error variables, thus attempting to overfit the training points.

To find the solution of (1) and (2), the dual problems are needed to be derived. The optimization problems (1) and (2) just described are the primal problems for regression. To formulate the corresponding dual problems, the Lagrangian function L is written. Then, L is minimized with respect to the weight w and slack variables ξ, η and maximized with respect to the Lagrange multipliers α_1 and α_2 . By carrying out this optimization, w can be written in terms of Lagrange multipliers α_1 and α_2 . Finally, substituting the value of w and simplifying with the help of Karush-Kuhn-Tucker (KKT) the following dual problem is obtained:

$$\begin{aligned} \max \quad & -\frac{1}{2} \alpha_1^T H (H^T H)^{-1} H^T \alpha_1 \\ & + \psi_1^T H (H^T H)^{-1} H^T \alpha_1 - \psi_1^T \alpha_1 \quad (3) \\ \text{s.t.} \quad & 0 \leq \alpha_1 \leq C_1 e, \end{aligned}$$

where $H = [K(A, A^T) \ e]$, $\psi_1 = Y - e\epsilon_1$. Then, problem (3) leads to

$$-H^T \psi_1 + H^T H u_1 + H^T \alpha_1 = 0. \quad (4)$$

That is,

$$u_1 = (H^T H)^{-1} H^T (\psi_1 - \alpha_1). \quad (5)$$

The solution of (5) requires the inverse of $H^T H$. Sometimes, matrix may be ill-conditioned. This situation may be avoided by adding a regularization term, $\sigma > 0$, to $H^T H$. Here, “I” is an identity matrix of suitable dimension. Therefore, (5) is reformulated as

$$u_1 = [w_1^T \ b_1]^T = (H^T H + \sigma I)^{-1} H^T (\psi_1 - \alpha_1). \quad (6)$$

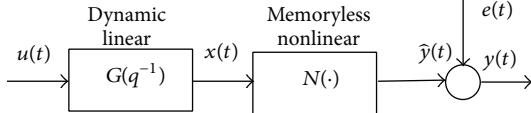


FIGURE 2: Block diagram of Wiener model. The investigator is assumed to have access to the input, $u(t)$, and the output, $y(t)$, but not the intermediate signals, $x(t)$ and $\hat{y}(t)$, or the innovation, $e(t)$.

Similarly, problem (2) is considered and its dual is obtained as

$$\begin{aligned} \max \quad & -\frac{1}{2}\alpha_2^T H(H^T H)^{-1} H^T \alpha_2 \\ & + \psi_2^T H(H^T H)^{-1} H^T \alpha_2 - \psi_2^T \alpha_2 \\ \text{s.t.} \quad & 0 \leq \alpha_2 \leq C_2 e, \end{aligned} \quad (7)$$

where $\psi_2 = Y + e\varepsilon_2$. Hence, problem (7) leads to

$$u_2 = [w_2^T \ b_2]^T = (H^T H + \sigma I)^{-1} H^T (\psi_2 + \alpha_2). \quad (8)$$

Note that, in the duals (3) and (7), the inversion of matrix $H^T H$ of size $(n+1) \times (n+1)$ should be computed [15]. Once the vectors u_1 and u_2 are known from (6) and (8) the two up and down bound functions are obtained. Then, the estimated regressor is constructed as follows:

$$\begin{aligned} \phi(x) &= \frac{1}{2}(\phi_1(x) + \phi_2(x)) \\ &= \frac{1}{2}(w_1 + w_2)^T K(A, x) + \frac{1}{2}(b_1 + b_2) \\ &= w_m^T K(A, x) + b_m. \end{aligned} \quad (9)$$

3. Identification of Wiener Models

The Wiener cascade, a linear filter followed by a static nonlinearity as shown in Figure 2, is often used to represent certain higher-order nonlinear systems.

In this section, the development in [19] is followed, up to the point where the SVM optimization is introduced (where TSVM is used). The output error Wiener model can be described as

$$\begin{aligned} x[n] &= G(q^{-1})u[n], \\ y[n] &= \mathcal{N}(x[n]) + e[n], \end{aligned} \quad (10)$$

where $u[n], x[n], y[n] \in \mathbb{R}$ are the input and output of the linear block and the measured output signals, respectively, for $n = 1, \dots, N$. The innovation $e[n]$ is assumed to be white. Identifying output error models is computationally heavier than autoregressive exogenous (ARX) models even for linear systems. Several techniques have been suggested to deal with linear output error (OE) models such as instrumental variables (IV), subspace methods, algorithms based on the Steiglitz-McBride, and using orthonormal filter expansions

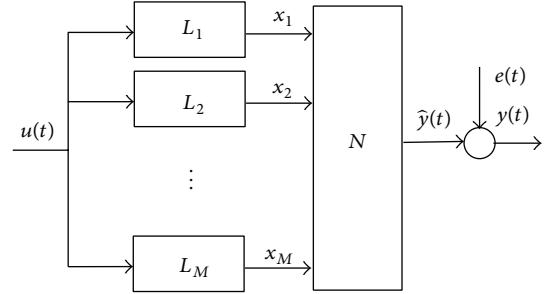


FIGURE 3: Wiener model with basis function expansion of linear block.

like Laguerre or Kautz functions [20]. In this paper the linear part will be represented by Laguerre filters. The truncated discrete Laguerre expansion of the transfer function $G(q^{-1})$ is given by

$$G(q^{-1}) = \sum_{k=1}^M g_k \frac{q^{-1} \sqrt{1-\alpha^2}}{1-\alpha q^{-1}} \left(\frac{q^{-1}-\alpha}{1-\alpha q^{-1}} \right)^{k-1}, \quad (11)$$

for all $\alpha < 1$, where the function $G(q^{-1})$ is assumed to be strictly proper, analytic in $|q| > 1$, and continuous in $|q| \geq 1$.

Remark 1. Notice that the finite impulse response FIR model can be obtained from (11) when $\alpha = 0$.

Substituting (11) into the first equation of (10) results in

$$x[n] = \sum_{k=1}^M g_k \frac{q^{-1} \sqrt{1-\alpha^2}}{1-\alpha q^{-1}} \left(\frac{q^{-1}-\alpha}{1-\alpha q^{-1}} \right)^{k-1} u[n], \quad (12)$$

which can be written as

$$x[n] = \sum_{k=1}^M g_k L_k u[n], \quad (13)$$

where $L_k = (q^{-1} \sqrt{1-\alpha^2} / (1-\alpha q^{-1}))((q^{-1}-\alpha) / (1-\alpha q^{-1}))^{k-1}$. Applying such expansion to the linear part of Wiener model enables us to rewrite the Wiener model (10) as follows:

$$y[n] = \mathcal{N} \left(\sum_{k=1}^M g_k L_k u[n] \right) + e[n]. \quad (14)$$

Remark 2. If the nonlinearity was modeled as a polynomial of order Q

$$N(x[n]) = \sum_{q=1}^Q c_q (x[n])^q, \quad (15)$$

then model (14) can be seen as bank of parallel linear filters L_k 's followed by multi-input-single-output (MISO) nonlinear block as shown in Figure 3.

Remark 3. Our contribution is to replace the polynomial used in (14) by TSVM which gives

$$\begin{aligned} y[n] &= \frac{1}{2}(w_1 + w_2)^T K(A, x[n]) + \frac{1}{2}(b_1 + b_2) \\ &+ e[n] = w_m^T K(A, x[n]) + b_m + e[n], \end{aligned} \quad (16)$$

TABLE 1: TSVR: root mean square errors between true and estimated output in Example 1 for test data set using various filter poles and filter orders. The other hyperparameters were fixed as $\varepsilon_1 = \varepsilon_2 = 0.08$, $C_1 = C_2 = 10$, and $\sigma = 1$.

| α | M | RMSE | CPU time |
|----------|-----|--------|----------|
| 0.1 | 1 | 0.1142 | 19.8875 |
| | 2 | 0.0807 | 19.7093 |
| | 3 | 0.0657 | 18.1007 |
| | 4 | 0.0638 | 22.0200 |
| | 5 | 0.0625 | 23.6666 |
| | 6 | 0.0574 | 18.2303 |
| | 7 | 0.0540 | 18.6477 |
| | 8 | 0.0542 | 22.4743 |
| | 9 | 0.0560 | 21.5188 |
| | 10 | 0.0588 | 25.2510 |
| 0.2 | | 0.0548 | 18.6072 |
| 0.3 | | 0.0569 | 24.6136 |
| 0.4 | 7 | 0.0615 | 19.4013 |
| 0.5 | | 0.0833 | 20.5413 |
| 0.6 | | 0.1011 | 18.9904 |

where $\mathbf{x}[n] = [x_1[n], \dots, x_M[n]]^T$ is the output of the bank of parallel linear filters L_k 's whose components are computed using (13).

Remark 4. Model (16) might be identified by solving (3)–(8). It is clear that, solving (3)–(8), which consists of two simple quadratic programming problems without equality constraint, is easier than solving the quadratic program developed in [19].

4. Example 1

To show the features of the proposed algorithm, the simulation example presented in [19] is identified. Consider a valve for fluid flow control described by

$$\begin{aligned} x(n) &= 1.4138x(n-1) + 0.6065x(n-2) \\ &= 0.1044u(n-1) + 0.0883u(n-2), \\ y_0(n) &= \frac{x(n)}{\sqrt{0.10 + 0.9x(n)^2}}, \end{aligned} \quad (17)$$

where $u(n)$ is a pneumatic control signal applied to the valve, $x(n)$ is the position of the valve plug, and $y_0(n)$ is the fluid flow. The fluid flow measurement $y(n)$ is given by

$$y(n) = y_0(n) + e(n), \quad (18)$$

where $e(n)$ consists of independent sequence of normally distributed random numbers with variance 0.0025. Two input-output sequences consisting of 1000 data points were generated as follows. A pseudorandom binary signal fluctuating between -1 and +1 with a basic clock period of seven sampling intervals was produced. Then, in each time interval, the input was multiplied by a uniformly distributed random

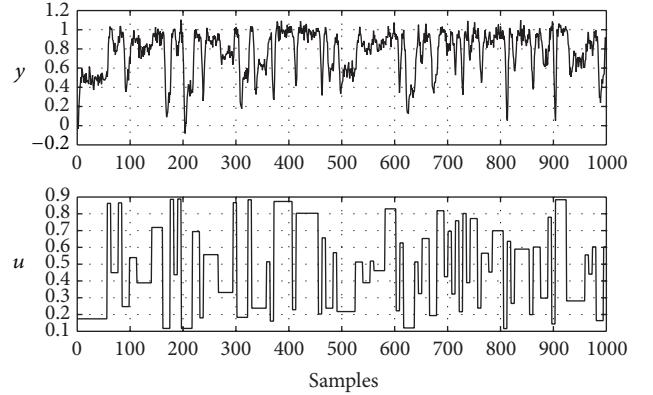


FIGURE 4: Training data of Example 1.

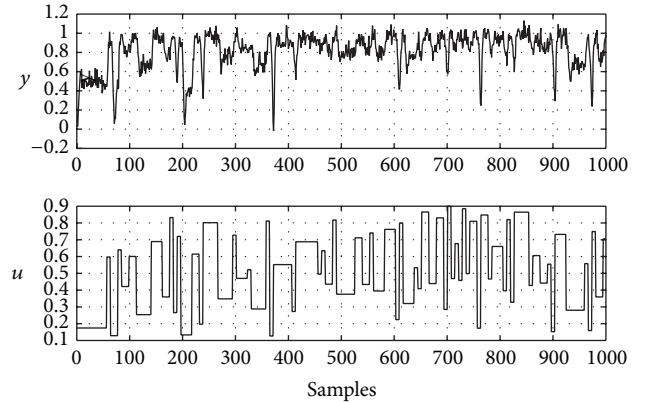


FIGURE 5: Testing data of Example 1.

factor between 0 and 0.4. Finally, a bias 0.5 was added on it to result in a signal having an amplitude between 0.1 and 0.9. This procedure was described and used in [19]. The training and testing data sets are shown in Figures 4 and 5, respectively. The hyperparameters (M , α , ε_1 , ε_2 , C_1 , C_2 , and σ) were selected based on cross-validation method where one parameter was varied and the others were kept fixed. The value of the varying parameter that gives the least root mean square error (RMSE) value is chosen as shown in Tables 1 and 2. The cross-validation process resulted in $\varepsilon_1 = 0.04$, $\varepsilon_2 = 0.04$, $C_1 = 100$, $C_2 = 100$, and $\sigma = 1$. The Laguerre filter pole and order were chosen to be $\alpha = 0.1$ and $M = 7$, respectively. Figure 6 shows TSVR based algorithm estimates together with measured output for the last 100 samples of the test data. It is clear from the figure that the algorithm produced accurate result.

5. Example 2

Consider pH neutralization process consisting of a continuous stirred tank reactor (CSTR) where a strong base (NaOH) reacts with the feed stream, a strong acid (HCl). The process is shown in Figure 7. The process input is the flow rate of the strong base and the process output is the pH of the effluent solution. The acid flow rate as well as the volume of the

TABLE 2: TSVR: root mean square errors between true and estimated output in Example 1 for test data set using various values of hyperparameters $C_1 = C_2$, $\varepsilon_1 = \varepsilon_2$, and σ . The filter pole and filter order were fixed as $\alpha = 0.1$, $M = 7$, respectively.

| ε | C | σ | RMSE | CPU time |
|---------------|------|----------|--------|----------|
| 0.08 | 10 | | 0.0540 | 17.9305 |
| | 100 | | 0.0540 | 18.2326 |
| | 200 | 1 | 0.0543 | 18.6875 |
| | 1000 | | 0.0539 | 17.9402 |
| | 2000 | | 0.0564 | 18.5952 |
| 0.08 | 100 | 0.1 | 0.1019 | 18.3051 |
| | | 0.2 | 0.0676 | 18.8480 |
| 0.04 | 100 | 1 | 0.0540 | 18.3080 |

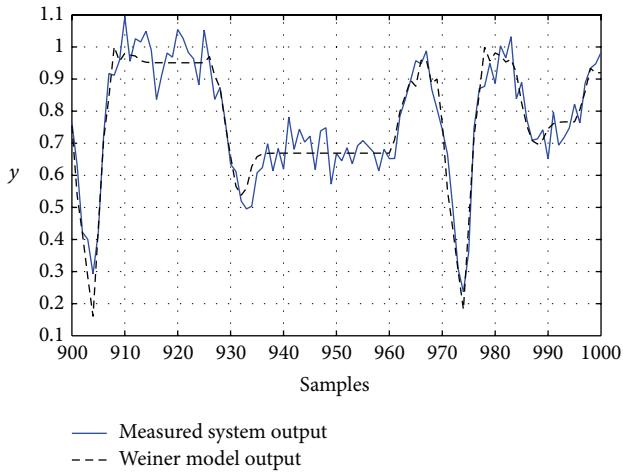


FIGURE 6: Measured system output (solid line), output of identified TSVM based model (dashed line).

TABLE 3: Nominal value.

| Nominal value | Process variable |
|---------------|--------------------------|
| 19.4 mL/s | Base flow rate (q_1) |
| 37 mL/s | Acid flow rate (q_2) |
| 0.000286 M | Base concentration |
| 0.00015 M | Acid concentration |

tank is assumed to be constant. The identification algorithm described in Section 3 is used to estimate a Wiener model of the simulation system presented in [21]. The nominal operating conditions of the system are given in Table 3. The model is highly nonlinear due to the implicit output equation, known as the titration curve. The system was excited with band limited white noise, with zero mean and 0.01 variance, around the nominal value of the base flow rate. The output of the system was corrupted with additive Gaussian white noise with zero mean and standard deviation $\sigma = 0.001$.

The training and testing data sets are shown in Figures 8 and 9, respectively. The TSVM and SVM hyperparameters and the Laguerre filter pole and order were chosen based on cross-validation method. For TSVM, the cross-validation resulted in $\varepsilon_1 = 0.08$, $\varepsilon_2 = 0.08$, $C_1 = 100$, $C_2 = 100$, and

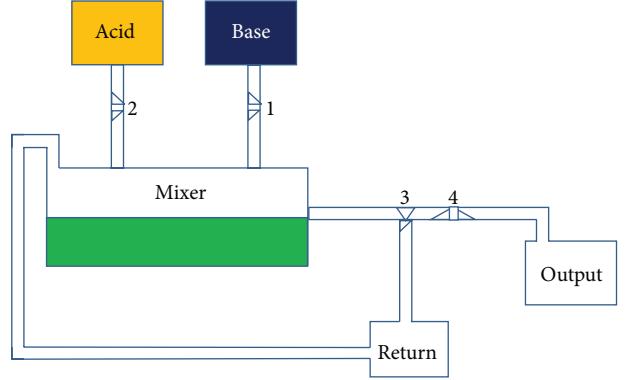


FIGURE 7: pH neutralization process.

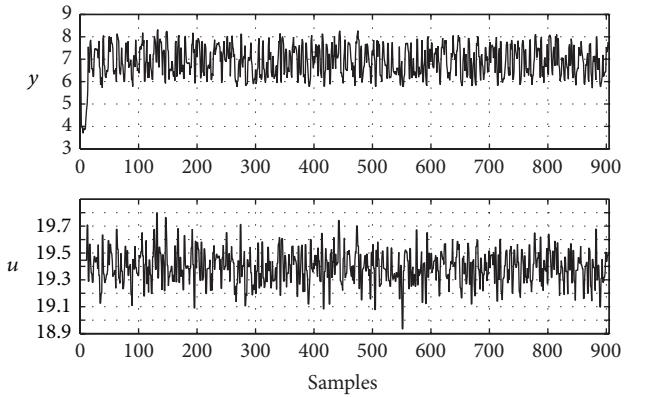


FIGURE 8: Training data of Example 2.

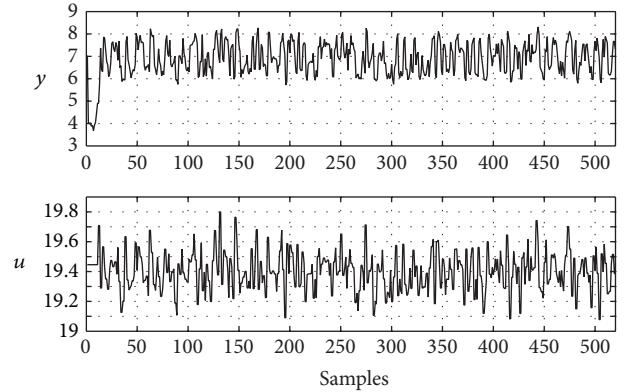


FIGURE 9: Testing data of Example 2.

$\sigma = 1$ (see Table 5). In Table 4, the Laguerre filter pole and order were chosen to be $\alpha = 0.1$ and $M = 9$, respectively. The SVM hyperparameters were chosen to be $\varepsilon = 0.08$, $C = 1000$, and $\sigma = 1$ as shown in Table 7 and the Laguerre filter pole and order were selected as $\alpha = 0.5$ and $M = 16$ (see Table 6).

By comparing the least RMSE (0.1248) and CPU time (21.36 sec.) values of the TSVR algorithm (Table 5) with the RMSE (0.1311) and CPU time (72.58 sec.) values of the SVR algorithm (Table 7), it is clear that TSVM algorithm

TABLE 4: TSVR: root mean square errors between true and estimated output in Example 2 for test data set using various filter poles and filter orders. The other hyperparameters were fixed as $\varepsilon_1 = \varepsilon_2 = 0.08$, $C_1 = C_2 = 100$, and $\sigma = 1$.

| α | σ | C | M | RMSE | CPU time |
|----------|----------|-----|-----|--------|----------|
| 0.1 | 1 | 100 | 1 | 0.4466 | 16.3546 |
| | | | 2 | 0.4125 | 15.5290 |
| | | | 3 | 0.3831 | 15.8178 |
| | | | 4 | 0.3465 | 19.4221 |
| | | | 5 | 0.2821 | 17.2951 |
| | | | 6 | 0.2439 | 18.0242 |
| | | | 7 | 0.1953 | 18.2451 |
| | | | 8 | 0.1462 | 17.7310 |
| | | | 9 | 0.1248 | 16.7690 |
| | | | 10 | 0.1272 | 19.6912 |
| | | | 11 | 0.1311 | 20.2375 |
| | | | 12 | 0.1313 | 19.3700 |
| 0.2 | | | | 0.1275 | 19.6035 |
| 0.3 | | | | 0.1280 | 20.6774 |
| 0.4 | | | | 0.1287 | 22.1147 |
| 0.5 | | 9 | | 0.1283 | 22.0277 |
| 0.6 | | | | 0.1303 | 16.1832 |
| 0.7 | | | | 0.1303 | 21.4789 |
| 0.8 | | | | 0.1334 | 16.3681 |
| 0.9 | | | | 0.1473 | 18.8526 |

TABLE 5: TSVR: root mean square errors between true and estimated output in Example 2 for test data set using various values of hyperparameters $C_1 = C_2$, $\varepsilon_1 = \varepsilon_2$, and σ . The filter pole and filter order were fixed as $\alpha = 0.1$, $M = 9$, respectively.

| ε | C | σ | RMSE | CPU time |
|---------------|-----|----------|--------|----------|
| 0.08 | 200 | 1 | 1 | 0.1250 |
| | | | 10 | 0.1248 |
| | | | 100 | 0.1248 |
| | | | 150 | 0.1248 |
| | | | 200 | 0.1249 |
| | | | 500 | 0.1250 |
| | | | 1000 | 0.1248 |
| | | | 1500 | 0.1264 |
| | | | 2000 | 0.1255 |
| | | | 0.1 | 0.1372 |
| | | | 0.2 | 0.1353 |
| | | | 0.3 | 0.1321 |
| 0.08 | 100 | 1 | 0.1248 | 16.0475 |
| | | 2 | 0.1298 | 23.4172 |
| | | 3 | 0.1368 | 21.0749 |
| | | 4 | 0.1453 | 19.6277 |
| | | | | |

outperforms the SVM algorithm in terms of accuracy and speed of computation.

6. Conclusion

In this paper a new algorithm for identification of Wiener systems using TSVR has been derived and used to identify

TABLE 6: SVR: root mean square errors between true and estimated output in Example 2 for test data set using various filter poles and filter orders. The other hyperparameters were fixed as $\varepsilon = 0.08$, $C = 100$, and $\sigma = 1$.

| α | σ | C | M | RMSE | CPU time |
|----------|----------|-----|-----|--------|----------|
| 0.1 | 1 | 100 | 1 | 0.4533 | 33.3899 |
| | | | 2 | 0.4291 | 27.5467 |
| | | | 3 | 0.3944 | 29.6294 |
| | | | 4 | 0.3655 | 27.5746 |
| | | | 5 | 0.3074 | 29.2262 |
| | | | 6 | 0.2651 | 32.6605 |
| | | | 7 | 0.2318 | 29.8089 |
| | | | 8 | 0.1926 | 31.2993 |
| | | | 9 | 0.1498 | 27.3892 |
| | | | 10 | 0.1483 | 26.7030 |
| | | | 11 | 0.1472 | 28.4650 |
| | | | 12 | 0.1466 | 26.9067 |
| 0.2 | 16 | 16 | 13 | 0.1455 | 22.8941 |
| | | | 14 | 0.1444 | 22.9346 |
| | | | 15 | 0.1439 | 22.8492 |
| | | | 16 | 0.1433 | 22.1835 |
| | | | 17 | 0.1435 | 23.0951 |
| | | | | | |
| 0.2 | | | | 0.1429 | 28.7001 |
| 0.3 | | | | 0.1403 | 28.4638 |
| 0.4 | | 16 | | 0.1371 | 27.4459 |
| 0.5 | | | | 0.1345 | 28.8956 |
| 0.6 | | | | 0.1364 | 24.4976 |

TABLE 7: SVR: root mean square errors between true and estimated output in Example 2 for test data set using various values of hyperparameters C , ε , and σ . The filter pole and filter order were fixed as $\alpha = 0.5$, $M = 16$, respectively.

| ε | C | σ | RMSE | CPU time |
|---------------|------|----------|--------|----------|
| 0.08 | 100 | 1 | 1 | 0.4803 |
| | | | 10 | 0.1894 |
| | | | 100 | 0.1433 |
| | | | 150 | 0.1393 |
| | | | 200 | 0.1370 |
| | | | 500 | 0.1323 |
| | | | 1000 | 0.1311 |
| | | | 2000 | 0.1313 |
| | | | 0.1 | 0.1369 |
| | | | 0.2 | 0.1371 |
| | | | 0.3 | 0.1371 |
| | | | | |
| 0.08 | 1000 | 1 | 0.1369 | 73.95 |
| | | 2 | 0.1371 | 73.48 |
| | | 3 | 0.1371 | 73.19 |
| | | 4 | 0.1369 | 73.23 |
| | | 5 | 0.1364 | 73.92 |
| | | 1 | 0.1311 | 72.58 |
| | | 2 | 0.1643 | 72.46 |

simulation examples. The algorithm was able to model the nonlinearity, without requiring any a priori assumptions regarding its structure. The linear model was represented by

Laguerre filter. The main advantage of the proposed algorithm over the method presented in [19] is that the proposed algorithm results in two smaller quadratic programs which is easier to solve than the quadratic program developed in [19]. It was shown in Example 2 that TSVM algorithm is computationally lighter than SVM algorithm.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Mujahed Al-Dhaifallah would like to acknowledge the support provided by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through Project no. FT131015.

References

- [1] J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain, "Identification of linear systems with nonlinear distortions," *Automatica*, vol. 41, no. 3, pp. 491–504, 2005.
- [2] D. Westwick and R. Kearney, *Identification of Nonlinear Physiological Systems*, John Wiley & Sons, Piscataway, NJ, USA, 2003.
- [3] D. Westwick and M. Verhaegen, "Identifying MIMO Wiener systems using subspace model identification methods," *Signal Processing*, vol. 52, no. 2, pp. 235–258, 1996.
- [4] D. Q. Wang and F. Ding, "Hierarchical least squares estimation algorithm for hammerstein-wiener systems," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 825–828, 2012.
- [5] L. Zhou, X. Li, and F. Pan, "Least-squares-based iterative identification algorithm for Wiener nonlinear systems," *Journal of Applied Mathematics*, vol. 2013, Article ID 565841, 6 pages, 2013.
- [6] F. Ding, J. Ma, and Y. Xiao, "Newton iterative identification for a class of output nonlinear systems with moving average noises," *Nonlinear Dynamics*, vol. 74, no. 1–2, pp. 21–30, 2013.
- [7] Y. Hu, B. Liu, Q. Zhou, and C. Yang, "Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises," *Circuits, Systems, and Signal Processing*, vol. 33, no. 2, pp. 655–664, 2014.
- [8] M. Schoukens, G. Vandersteen, Y. Rolain, and F. Ferranti, "Fast identification of Wiener-Hammerstein systems using discrete optimisation," *Electronics Letters*, vol. 50, no. 25, pp. 1942–1944, 2014.
- [9] M. Schoukens, A. Marconato, R. Pintelon, G. Vandersteen, and Y. Rolain, "Parametric identification of parallel Wiener-Hammerstein systems," *Automatica*, vol. 51, pp. 111–122, 2015.
- [10] S. L. Lacy and D. S. Bernstein, "Identification of FIR Wiener systems with unknown, noninvertible, polynomial nonlinearities," in *Proceedings of the American Control Conference*, pp. 893–898, May 2002.
- [11] M. Pawlak, Z. Hasiewicz, and P. Wachel, "On nonparametric identification of Wiener systems," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 482–492, 2007.
- [12] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [13] 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.
- [14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [15] P. Xinjun, "TSVR: an efficient twin support vector machine for regression," *Neural Networks*, vol. 23, no. 3, pp. 365–372, 2010.
- [16] Y. Tian and Z. Qi, "Review on: twin support vector machines," *Annals of Data Science*, vol. 1, no. 2, pp. 253–277, 2014.
- [17] Z. Qi, Y. Tian, and Y. Shi, "Laplacian twin support vector machine for semi-supervised classification," *Neural Networks*, vol. 35, pp. 46–53, 2012.
- [18] Z. X. Yang, "Nonparallel hyperplanes proximal classifiers based on manifold regularization for labeled and unlabeled examples," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 5, Article ID 1350015, pp. 1–19, 2013.
- [19] S. Tötterman and H. T. Toivonen, "Support vector method for identification of Wiener models," *Journal of Process Control*, vol. 19, no. 7, pp. 1174–1181, 2009.
- [20] L. Ljung, *System Identification: Theory for the User*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [21] S. A. AlSabbah, M. A. Al-Khedher, M. K. Abu Zalata, and T. M. Younes, "Evaluation of multiregional fuzzy cascade control for pH neutralization process," *International Journal of Research Reviews in Applied Sciences*, vol. 10, no. 2, pp. 193–199, 2012.

