An Improved Adaptive PID Controller Based on Online LSSVR with Multi RBF Kernel Tuning

Kemal Ucak and Gulay Oke

Department of Control Engineering
Istanbul Technical University
Maslak,TR-34469, Istanbul, Turkey
{kemal.ucak,gulay.oke}@itu.edu.tr

Abstract. In this paper, the effects of using multi RBF kernel for an online LSSVR on modeling and control performance are investigated. The Jacobian information of the system is estimated via online LSSVR model. Kernel parameter determines how the measured input is mapped to the feature space and a better plant model can be achieved by discarding redundant features. Therefore, introducing flexibility in kernel function helps to determine the optimal kernel. In order to interfuse more flexibility to the kernel, linear combinations of RBF kernels have been utilized. The purpose of this paper is to improve the modeling performance of the LSSVR and also control performance obtained by adaptive PID by tuning bandwidths of the RBF kernels. The proposed method has been evaluated by simulations carried out on a continuously stirred tank reactor (CSTR), and the results show that there is an improvement both in modeling and control performances.

Keywords: Adaptive PID, Kernel Polarization, Multi Kernel, Online LSSVR.

1 Introduction

PID Controllers have been the most popular controller structure owing to their simplicity and robustness, in spite of further developments in control engineering. The strength of the PID control lies in the simplicity, lucid meaning and clear effect [1]. The PID's long life force comes from its clear meaning and effectiveness in practice [1]. In adaptive control, tuning of the PID parameters rapidly and optimally is significant to get a good tracking performance in nonlinear systems [2]. Parameters of the model based PID controller can be tuned employing Support Vector Machine (SVM) or Neural Network model of the system.

Support Vector Machine theory has recently been utilized for tuning controller parameters [3,4,5,6], instead of Neural Network approach since it ensures global minima and it has powerful generalization ability. Since parameters of the controller are tuned according to the model of the plant, the control performance of PID based on Support Vector Regression (SVR) is directly related to the performance of the model.

The main design component in SVM is the kernel which is a nonlinear mapping function from the input to the feature space [7]. The main function of the kernel is to convert a linearly non separable classification problem in low dimension to a separable

A. Bouchachia (Ed.): ICAIS 2011, LNAI 6943, pp. 40–51, 2011.

one in high dimension and hence plays a crucial role in the modeling and control performance [2]. Kernel functions are generally parametric [2] and the numerical values of these parameters have significant effect on both modeling and control performance. Depending on the initial values of kernel parameters some features significant for the model may be discarded or redundant or irrelevant features may also be mapped to the feature space and, better performance may be achieved by discarding some features [8]. Owing to such factors, the selection of optimal kernel parameters is crucial in terms of the solution of the SVR problem. In order to seek optimal kernel parameters for regression, particle swarm optimization, pattern search and grid-diamond search have been utilized in [9],[10] and [11]. The common aim in all these approaches is to compute the kernel parameters offline. In [2], gradient descent optimization method has been utilized to tune the bandwidth of the single RBF kernel during online operation. The purpose of this paper is to improve the proposed method given in [2] introducing more flexibility to the kernel function using multi RBF.

In this paper, taking into consideration that more flexibility in kernel provides better SVM model, it's been our aim to estimate optimal values for parameters of multi RBF kernels and controller parameters using gradient information. In section 2, a brief overview of LSSVR and Online LSSVR are given. Tuning of multi RBF kernel parameters using online LSSVR is presented in Section 3. Adaptive PID controller based on online LSSVR is detailed in section 4. Simulation results, performance analysis of the online multi RBF kernel and adaptive PID, are given in section 5. The paper ends with a brief conclusion in Section 6.

2 Online Least Square Support Vector Regression

Given a training data set:

$$(y_1, \underline{x}_1).....(y_k, \underline{x}_k)\ ,\ \ x \in R^n, y \in R \quad k = 1,2,,,,N\ . \tag{1}$$

where N is the size of training data and n is the dimension of the input matrix, the optimization problem for LSSVR is defined as follows to maximize the geometric margin and minimize the training error:

$$\min_{(w,b,e)} \frac{1}{2} w^T w + \frac{1}{2} C \sum_{k=1}^{N} e_k^2 \tag{2}$$

subject to

$$y_k - w^T \varphi(x_k) - b - e_k = 0\ ,\ \ k = 1,2,....,N \tag{3}$$

The minimization problem presented in equation (2, 3) is called the primal objective function [12]. By utilizing the primal objective function and its constraints, Lagrangian function can be derived as follows:

$$L(w,b,e,a) = \frac{1}{2} w^T w + \frac{1}{2} C \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} \alpha_k (w^T \varphi(x_k) + b + e_k - y_k) \tag{4}$$

In (4), L is the Lagrangian and, α_k are Lagrange multipliers [13-15]. For optimality primal variables have to vanish at the saddle point:

$$\frac{\partial L}{\partial b} = \sum_{k=1}^{N} \alpha_k = 0 \tag{5}$$

$$\frac{\partial L}{\partial w} = \underline{w} - \sum_{k=1}^{N} \alpha_k \varphi(x_k) = 0 \longrightarrow \underline{w} = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \tag{6}$$

$$\frac{\partial L}{\partial e_k} = C \sum_{k=1}^{N} e_k - \sum_{k=1}^{N} \alpha_k = 0 \longrightarrow \alpha_k = Ce_k \tag{7}$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \longrightarrow y_k = w^T \varphi(x_k) + b + e_k \quad k = 1, 2, \ldots, N \tag{8}$$

The solution of the problem is as follows via (5-8):

$$\begin{bmatrix} \underline{0} & \underline{1} \\ \underline{1}^T & \Omega_{km} + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ \underline{a}^T \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{y}^T \end{bmatrix} \tag{9}$$

with

$\underline{y} = [y_1, y_2, .., y_N]$, $\underline{a} = [a_1, a_2, .., a_N]$, $\underline{1} = [1,1,..,1], \Omega_{km} = K(x_k, x_m)$, $k, m = 1, 2, \ldots, N$

Further information about LSSVR is available in [13], [14] and [15].

The dynamics of a non-linear system, can be represented by the Nonlinear AutoRegressive with eXogenous inputs (NARX) model ,

$$y(n+1) = f(u(n),..,u(n-n_u), y(n-1),.., y(n-n_y)) \tag{10}$$

where $u(n)$ is the control input applied to the plant at time n , $y(n+1)$ is the output of the plant , and n_u and n_y stand for the number of past control inputs and the number of past plant outputs involved in the model respectively [8].

The state vector of the system at time index n is represented as follows:

$$\underline{x}(n) = [u(n),..,u(n-n_u), y(n-1),.., y(n-n_y)] \tag{11}$$

The output of the model can be written as:

$$\hat{y}(n+1) = \sum_{i=n-L}^{n-1} a_i(n) K(\underline{x}(n), \underline{x}(i)) + b(n) \tag{12}$$

using equations (8),(10) and (11).

A training data set $(\underline{X}(n), \underline{Y}(n))$ can be established using inputs $\underline{X}(n) = [\underline{x}(n-1), \ldots. \underline{x}(n-L)]$, the corresponding outputs of the system $\underline{Y}(n) = [y(n),.., y(n-L+1)]$, and L, the length of the sliding window. $\alpha(n)$ and $b(n)$ are obtained as follows assuming $\underline{U}(n) = [\underline{\Omega}(n) + \frac{I}{C}]^{-1}$ and using (9):

$$\underline{\alpha}^T(n) = \underline{U}(n)[\underline{Y}(n) - \underline{1}^T b(n)] \tag{13}$$

$$b(n) = \frac{\underline{1}\underline{U}(n)\underline{Y}(n)}{\underline{1}\underline{U}(n)\underline{1}^T} \tag{14}$$

At time index *n,* we have:

$$\underline{X}(n) = [\underline{x}(n-1),....\underline{x}(n-L)]\ ,\ \underline{Y}(n) = [y(n),.., y(n-L+1)] \tag{15}$$

At time index *n+1,* $\underline{X}(n+1)$ can be expressed as:

$$\underline{X}(n+1) = [\underline{x}(n), \underline{x}(n-1),....\underline{x}(n-L+1)]\ ,\ \underline{Y}(n+1) = [y(n+1), y(n),..,(n-L+2)] \tag{16}$$

New data pair ($\underline{x}(n), y(n+1)$) is added to the training data set and, old data pair ($\underline{x}(n-L), y(n-L+1)$) is discarded from the training data set at time index *n+1.*

References [2],[3] and [16] can be referred for detailed information on the online identification procedure of nonlinear systems via online least square support vector regression.

3 Tuning of Multi-RBF Kernel

In this work, an online tuning procedure for the bandwidths and scaling coefficients of a multi RBF kernel has been proposed to improve modeling and control performance. The multi RBF kernel is given as follows:

$$Ker(\underline{x}_c(n), x_{svi}(n)) = \sum_{j=1}^{m} \frac{k_j \exp(-\frac{d_{c,svi}(n)}{2\sigma_j^{\,2}(n)})}{\sum_{z=1}^{m} k_z} \tag{17}$$

where $\sigma_j(n)$ is the bandwidth of the kernel function, $\underline{x}_c(n)$ is the current state vector of the plant and $d_{c,svi}(n)$ is the Euclidean distance between current data and the i^{th} support vector.

$$d_{c,svi}(n) = (\underline{x}_c(n) - x_{svi}(n))^T (\underline{x}_c(n) - x_{svi}(n)) \tag{18}$$

In order to reveal the behavior of the multi RBF kernel function with fixed bandwidth, the behavior of the multi RBF kernel has been analyzed by synchronizing it with a single kernel. Assume that the multi kernel is the linear combination of 2 RBF kernels, the new kernel is as follows:

$$Ker(\underline{x}_c(n), x_{svi}(n)) = \frac{k_1(n)}{k_1(n)+k_2(n)} K_1(n) + \frac{k_2(n)}{k_1(n)+k_2(n)} K_2(n) \tag{19}$$

where

$$K_j(n) = K(\underline{x}_c(n), x_{svi}(n), \sigma_j) = \exp(-\frac{d_{c,svi}(n)}{2\sigma_j^{\,2}(n)}) \tag{20}$$

The following equation is written to examine whether this multi kernel can be represented using a single kernel or not.

$$\exp(-\frac{d_{c,svi}(n)}{2\sigma_s^{\,2}(n)}) = \frac{k_1(n)}{k_1(n)+k_2(n)}K_1(n) + \frac{k_2(n)}{k_1(n)+k_2(n)}K_2(n) \tag{21}$$

$$\sigma_s^{\,2}(n) = -\frac{d_{c,svi}(n)}{2\ln\left(\frac{k_1(n)}{k_1(n)+k_2(n)}K_1(n) + \frac{k_2(n)}{k_1(n)+k_2(n)}K_2(n)\right)} \tag{22}$$

As can be seen from equation (22), multi kernel with fixed bandwidths equals to a single kernel with varying bandwidth depending on scaling coefficients and Euclidean distance between features. That is, multi RBF kernel with fixed bandwidths behaves like a single kernel with time varying bandwidth. In this paper, it is proposed to tune multi kernel to improve modeling performance of the system using the flexibility of multi kernel. The regression function with multi RBF kernel can be rewritten as in (23)

$$\hat{y}(n) = \sum_{i=n-L}^{n-1} \alpha_i(n) Ker((\underline{x}_c(n), x_{svi}(n))) + b(n) \tag{23}$$

Partial derivatives of LSSVR model with respect to weights and bandwidths of the kernels are obtained as follows:

$$\frac{\partial \hat{y}}{\partial k_j(n)} = \sum_{i=n-L}^{n-1} \alpha_i(n)\Big[\sum_{z=1}^{m} \frac{k_z(n)(K_j(n)-K_z(n))}{[\sum_{z=1}^{m} k_z(n)]^2}\Big] \tag{24}$$

$$\frac{\partial \hat{y}(n)}{\partial \sigma_j(n)} = \sum_{i=n-L}^{n-1} \alpha_i(n) \frac{k_j}{\sum_{z=1}^{m} k_m} K_j \frac{d_{c,svi}^{\;2}(n)}{\sigma_j^{\,3}(n)} \tag{25}$$

The objective function to be minimized to improve model performance is selected as:

$$J_{\mathrm{m}}(n) = \frac{1}{2}[\,y(n) - \hat{y}(n)\,]^2 = \frac{1}{2}[\hat{e}_m(n)^2] \tag{26}$$

where $\hat{e}_m$ is model error. Since $\hat{y}(n)$ is a function of the multi kernel functions, kernel bandwidths and scaling coefficients can be tuned by applying gradient descent method to minimize the objective function given in (26) as follows [18]:

$$\Delta\sigma_j(n) = -\eta(n)\frac{\partial J_m(n)}{\partial \hat{e}_m(n)}\frac{\partial \hat{e}_m(n)}{\partial \hat{y}(n)}\frac{\partial \hat{y}(n)}{\partial \sigma_j(n)} = \eta(n)\hat{e}_m(n)\frac{\partial \hat{y}(n)}{\partial \sigma_j(n)} \tag{27}$$

$$\Delta k_j(n) = -\eta(n)\frac{\partial J_m(n)}{\partial \hat{e}_m(n)}\frac{\partial \hat{e}_m(n)}{\partial \hat{y}(n)}\frac{\partial \hat{y}(n)}{\partial k_j(n)} = \eta(n)\hat{e}_m(n)\frac{\partial \hat{y}(n)}{\partial k_j(n)} \tag{28}$$

where $\eta(n)$ is the learning rate ($0<\eta(n)<1$), which can be obtained utilizing any line search algorithm. Thus, the kernel parameters can be tuned using (27, 28) as follows:

$$\begin{aligned}\sigma_j(n+1) &= \sigma_j(n)+\Delta\sigma_j(n)\\ k_j(n+1) &= k_j(n)+\Delta k_j(n)\end{aligned} \tag{29}$$

4 Adaptive PID Controller with Multi RBF Kernel

An adaptive PID controller has been utilized to control a third order continuously stirred tank reactor (CSTR). Online LSSVR has been employed to model the dynamics of the plant. The system Jacobian information has been approximated from the model and this information is used to tune the PID controller coefficients.

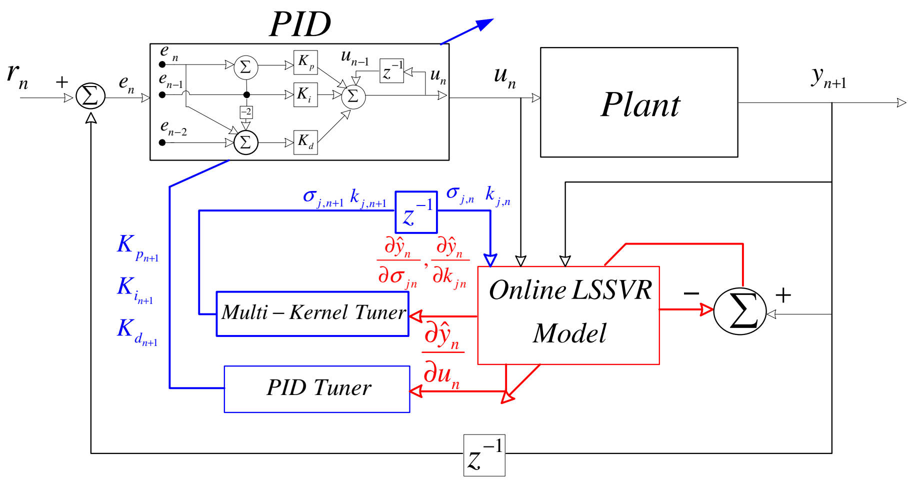

Fig. 1. Adaptive PID controller and kernel tuner

In fig. 1, the adaptive PID controller and the proposed multi-kernel tuner are illustrated, where $\hat{y}(n)$ is the output of the model, $y(n)$ is the output of the system, $\sigma_j(n)$ is the bandwidth of the kernels, $r(n)$ is the reference trajectory for plant, and $e(n), \hat{e}(n)$ are the tracking and model error respectively at time index n.

$$e(n) = r(n) - y(n) \quad , \quad \hat{e}(n) = y(n) - \hat{y}(n) \tag{30}$$

The adaptive PID controller consists of 4 parts: PID controller, online LSSVR, PID and kernel parameter tuners. The classical incremental PID controller produces a control signal as follows [3,6,17]:

$$\begin{aligned}u(n) &= u(n-1) + K_p(n)(e(n)-e(n-1)) + K_i(n)(e(n))\\ &\quad + K_d(n)(e(n)-2e(n-1)+e(n-2))\end{aligned} \tag{31}$$

where K_p, K_i and K_d are the controller parameters to be tuned. At the beginning of the control, the controller and kernel parameters are not optimal [6], and hence, they need to be tuned using optimization techniques [6,18]. The tracking error results from non optimal controller parameters. That is, tracking error is the function of controller parameters.

$$e(n) = f(K_i(n), K_d(n), K_p(n)) \tag{32}$$

The model error is the function of model parameters.

$$\hat{e}(n) = f([\sigma_1(n),.....\sigma_m(n)],[k_1(n),.....k_m(n)], C, \alpha(n), b(n)) \tag{33}$$

Since the Lagrange multipliers and bias of the regression model are obtained using LSSVR, and C is taken as a fixed value, the modeling error can be rewritten as follows:

$$\hat{e}(n) = f([\sigma_1(n),.....\sigma_m(n)],[k_1(n),.....k_m(n)]) \tag{34}$$

In order to make the controller and kernel parameters converge to their optimal values, gradient descent algorithm has been employed. The inputs of the PID controller can be defined as follows:

$$xc(1) = e(n) - e(n-1),\ xc(2) = e(n),\ xc(3) = e(n) - 2e(n-1) + e(n-2) \tag{35}$$

Since the controller parameters are tuned using model based method, gradient descent is utilized to minimize not only the tracking but also the modeling error by approximating the one step ahead future behavior of the plant with online LSSVR. That is, the following objective function is tried to be minimized:

$$J(n) = \frac{[r(n) - y(n)]^2}{2} + J_m(n) = \frac{1}{2}[e^2(n) + \hat{e}_m^2(n)] \tag{36}$$

The PID parameters are tuned as follows:

$$\Delta K_p(n) = -\eta(n)\frac{\partial J(n)}{\partial K_p(n)},\ \Delta K_i(n) = -\eta(n)\frac{\partial J(n)}{\partial K_i(n)},\ \Delta K_d(n) = -\eta(n)\frac{\partial J(n)}{\partial K_d(n)} \tag{37}$$

where $\eta(n)$ is the learning rate, determined using line search algorithm. Golden section has been used to compute $\eta(n)$. In this paper, multi kernel adaptation proposed in (27)-(29) has been used to improve the modeling performance and consequently the control performance. In the calculation of the control parameter updates above, Jacobian information, relating the input and output of the controlled system has been provided by the online LSSVR. Thus, parameters are tuned as follows:

$$\frac{\partial \hat{y}(n)}{\partial u(n)} = -\sum_{i=n-L}^{n-1} \alpha_i(n) \frac{\sum_{z=1}^{m} [\frac{k_z(n) K_z(\underline{x}_c(n), x_{svi}(n))}{\sigma_z^2(n)} (u(n) - x_{svi}(1, n-i))]}{\sum_{z=1}^{m} k_z(n)} \tag{38}$$

$$\begin{aligned}
\Delta K_p(n) &= -\eta(n) e(n) \frac{\partial y(n)}{\partial u(n)} xc(1) \ , \ K_p(n+1) = K_p(n) + \Delta K_p(n) \\
\Delta K_i(n) &= -\eta(n) e(n) \frac{\partial y(n)}{\partial u(n)} xc(2) \ , \ K_i(n+1) = K_i(n) + \Delta K_i(n) \\
\Delta K_d(n) &= -\eta(n) e(n) \frac{\partial y(n)}{\partial u(n)} xc(3) \ , \ K_d(n+1) = K_d(n) + \Delta K_d(n)
\end{aligned} \tag{39}$$

As depicted by the simulation results given in section 5 these update rules improve the controller and modeling performance and the parameters are expected to converge to their optimal values in the long run.

5 Simulation Results

The performance of the proposed kernel parameter adaptation method in modeling and control has been evaluated on a third order continuously stirred tank reactor (CSTR). The dynamics of CSTR is given by the following set of differential equations:

$$\begin{aligned}
\dot{x}_1(t) &= 1 - x_1(t) - Da_1 x_1(t) + Da_2 x_2^{\ 2}(t) \\
\dot{x}_2(t) &= -x_2(t) + Da_1 x_1(t) - Da_2 x_2^{\ 2}(t) - Da_3 d_2 x_2^{\ 2}(t) + u(t) \\
\dot{x}_3(t) &= -x_3(t) + Da_3 d_2 x_2^{\ 2}(t)
\end{aligned} \tag{40}$$

where $Da_1 = 3$, $Da_2 = 0.5$, $Da_3 = 1$, $d_2 = 1$, $u(t)$ is the control signal and $x_3(t)$ is the output of the process [6,19,20]. The magnitude of the control signal has been restricted between 0 and 1. The Jacobian information obtained as the output of the online LSSVR block in fig.1 is used to tune adaptive PID controller parameters. The initial values of all controller parameters are set to zero. In order to compare the adaptive single and adaptive multi kernel performance, initial values of the kernel bandwidth parameter are set to 1.

Fig. 2 illustrates that PID controllers, obtained using both adaptive single kernel and adaptive multi kernel methods attain good tracking performance, furthermore the control signal produced by controller with multi RBF kernel is more moderate than the other one. As can be seen from fig.2, introducing flexibility using multi RBF accelerates the response of the system. The adaptations of controller parameters are depicted in fig. 3 for both cases and the variation of kernel bandwidth is illustrated in fig. 4 and 5. The kernel parameters converge to their optimal values to obtain better plant model depending on the alternation of reference signal.

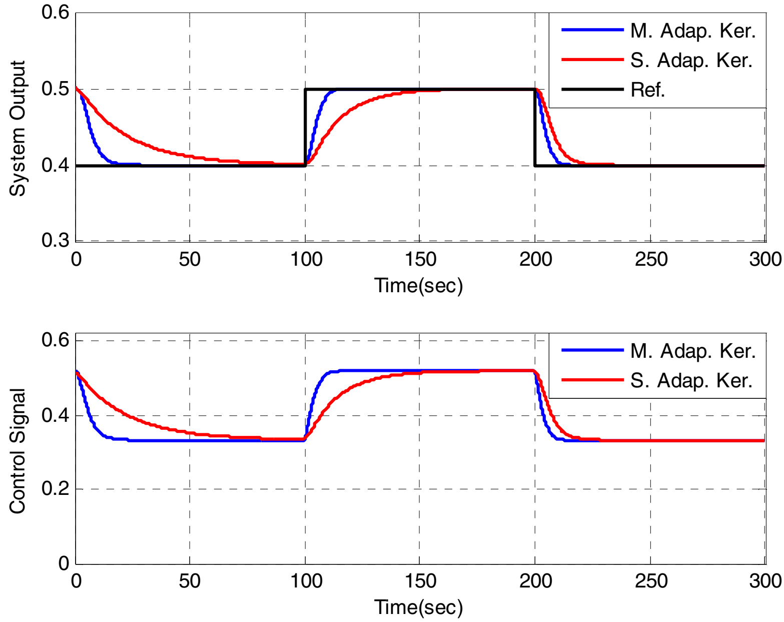

Fig. 2. System outputs for adaptive single(S.) kernel and adaptive multi(M.) kernel

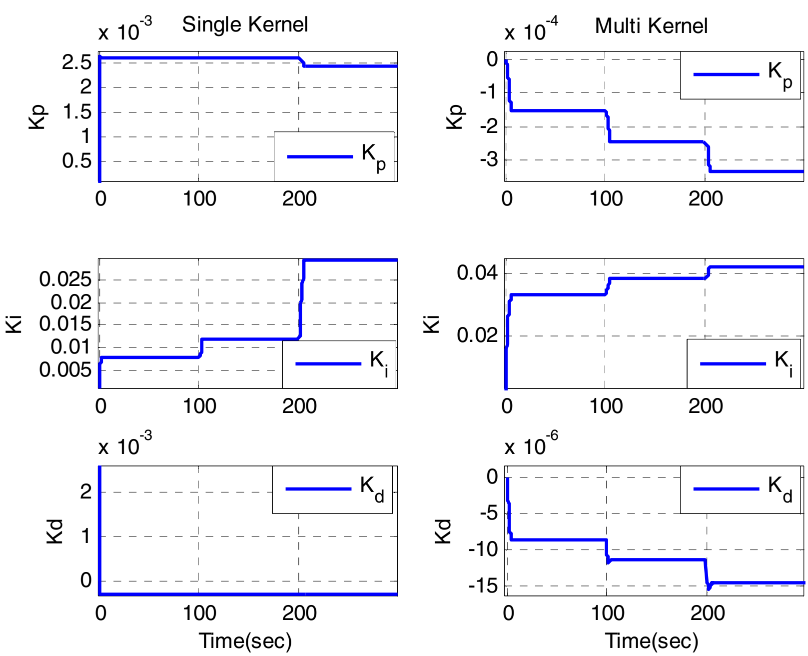

Fig. 3. Controller parameters for single and multi kernel case

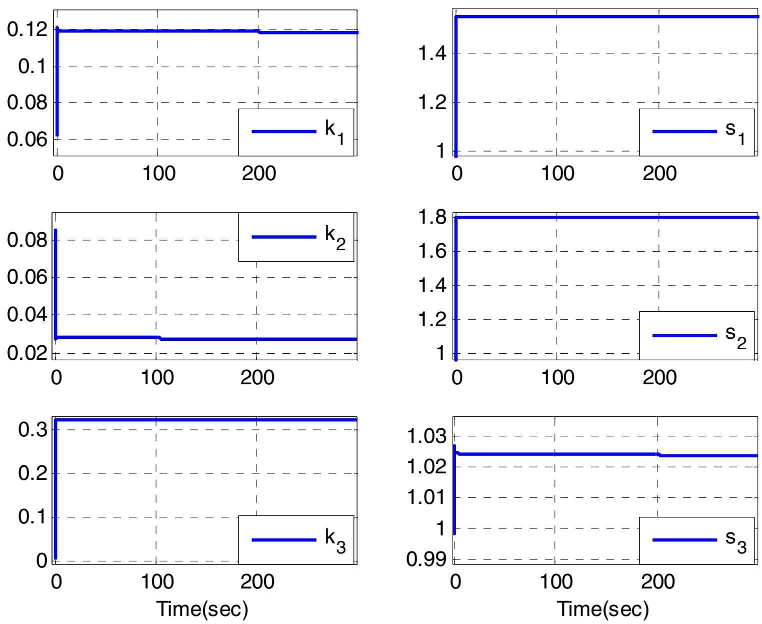

Fig. 4. Kernel parameters for adaptive multi kernel

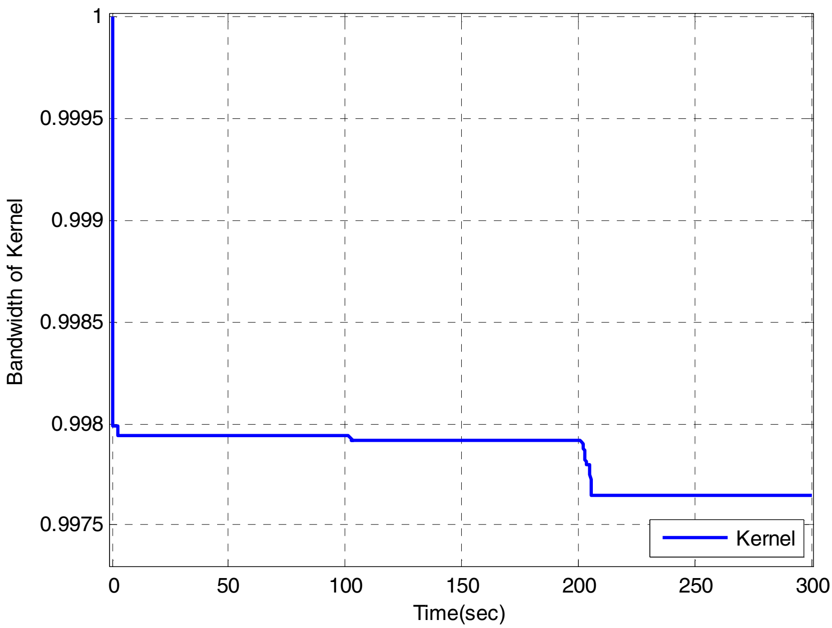

Fig. 5. Bandwidth of the adaptive single kernel

The modeling and tracking performances for single and multi adaptive bandwidth kernels are compared in Table 1. The following equation is used to compare the performance of the controllers:

$$J_{comp} = \sum_{n=1}^{N} (r(n) - y(n))^2 + (u(n) - u(n-1))^2 \tag{41}$$

The table indicates that utilizing multi kernel provides more flexibility in kernel machine and improves the performance of the controller.

As can be seen from third column of Table 1, the performance of the overall system obtained for the case with adaptive multi kernel bandwidth is better than that obtained with adaptive single kernel bandwidth, in terms of tracking-model error and performance index given in (41).

Table 1. Model and Control Performance

Error(MAE)	Single Adaptive Kernel	Multi Adaptive Kernel	Improvement (%)
Tracking Error	0.0160	0.0058	63.6374
Modeling Error	5.4017e-004	3.1902e-004	40.9401
J_{comp}	2.7436	1.1964	56.3928

6 Conclusions

In this paper, taking into consideration that flexibility in kernel function reduces the modeling and control error resulting from kernels, it has been aimed to tune the parameters of a multi kernel online LSSVR using gradient information. Controller parameters are also tuned using gradient descent. Employing online identification to model the system reduces the time consuming calculations of SVR. The results show that the proposed tuning method improves the performance of the controller in terms of tracking and modeling error. By combining the powerful features of fuzzy logic and support vector machines, more sophisticated and successful modeling and control techniques can be employed in future works.

References

1. Choi, Y., Chung, W.K.: PID Trajectory Tracking Control for Mechanical Systems. LNCIS. Springer, Berlin (2004)
2. Ucak, K., Oke, G.: Adaptive PID Controller Based on Online LSSVR with Kernel Tuning. In: International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey (2011)
3. Wanfeng, S., Shengdun, Z., Yajing, S.: Adaptive PID Controller Based on Online LSSVM Identification. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol. 1-3, pp. 694–698 (2008)
4. Zhao, J., Li, P., Wang, X.S.: Intelligent PID Controller Design with Adaptive Criterion Adjustment via Least Squares Support Vector Machine. In: 21st Chinese Control and Decision Conference (2009)

5. Takao, K., Yamamoto, T., Hinamoto, T.: A design of PID controllers with a switching structure by a support vector machine. In: International Joint Conference on Neural Network, Vancouver, BC, Canada (2006)
6. Iplikci, S.: A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems. International Journal of Robust and Nonlinear Control 20, 1483–1501 (2010)
7. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13(5) (2006)
8. Iplikci, S.: Controlling the Experimental Three-Tank System via Support Vector Machines. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) Adaptive and Natural Computing Algorithms. LNCS, vol. 5495, pp. 391–400. Springer, Heidelberg (2009)
9. Guo, Y.C.: An Integrated PSO for Parameter Determination and feature selection of SVR and its application in STLF. In: International Conference on Machine Learning and Cybernetics, Baoding, PR China, pp. 359–364 (2009)
10. Momma, M., Bennett, K.P.: A pattern search method for model selection of support vector regression. In: International Conference on Data Mining, Arlington, VA, pp. 261–274 (2002)
11. Hou, L.K., Yang, Q.X.: Study on parameters selection of LSSVR based on Grid-Diamond search method. In: International Conference on Machine Learning and Cybernetics, Baoding, PR China, vol. 1-6, pp. 1219–1224 (2009)
12. Farag, A., Mohamed, R.M.: Regression using support vector machines: basic foundations, Technical Report (2004)
13. Suykens, J.A.K.: Nonlinear modeling and support vector machines. In: IEEE Instrumentation and Measurement Technology Conference Budapest, Hungary (2001)
14. Gunn, S.: Support Vector Machines for Classification and Regression, ISIS Technical Report (1998)
15. Smola, A.J., Scholkopf, B.: A tutorial on support vector regression. Statistics and Computing 14, 199–222 (2004)
16. Zhu, Y.F., Mao, Z.Y.: Online Optimal Modeling of LS-SVM based on Time Window. In: IEEE International Conference on Industrial Technology, Hammamet, Tunisia, vol. 1-3, pp. 1325–1330 (2004)
17. Bobal, V., Böhm, J., Fessl, J., Macháček, L.: Digital-Self Tuning Controllers, Advanced Textbooks in Control and Signal Processing. Springer, London (2005)
18. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming, 3rd edn. Springer Science + Business Media, LLC (2008)
19. Wu, W., Chou, Y.S.: Adaptive feedforward and feedback control of non-linear time-varying uncertain systems. International Journal of Control 72, 1127–1138 (1999)
20. Ungar, L.H.: A bioreactor benchmark for adaptive-network based process control. In: Neural Network for Control. MIT Press, Cambridge (1990)