Design of an Adaptive Support Vector Regressor Controller for a Spherical Tank System

Kemal Uçak(✉) and Gülay Öke Günel

Faculty of Electrical-Electronics, Department of Control and Automation Engineering, Istanbul Technical University, Ayazaga Campus, 34469 Istanbul, Turkey
{kemal.ucak,gulay.oke}@itu.edu.tr
http://www.kontrol.itu.edu.tr

Abstract. In this study, an adaptive support vector regressor (SVR) controller which has previously been proposed [1] is applied to control the liquid level in a spherical tank system. The variations in the cross sectional area of the tank depending on the liquid level is the main cause of nonlinearity in system. The parameters of the controller are optimized depending on the future behaviour of the system which is approximated via a seperate online SVR model of the system. In order to adjust controller parameters, the "closed-loop margin" which is calculated using the tracking error has been optimized. The performance of the proposed method has been examined by simulations carried out on a nonlinear spherical tank system, and the results reveal that the SVR controller together with SVR model leads to good tracking performance with small modeling, transient state and steady state errors.

Keywords: Model based adaptive control · Online support vector regression · Spherical tank system · SVR controller · SVR model identification

1 Introduction

Changing of living organisms' characteristics physically or behaviorally to enhance their resistance against alternating environmental aspects is called "adaptation" [2]. Inspired by this feature of living organisms, adaptation capability can be interfused to conventional controllers which are especially essential for nonlinear systems that are hard to control using only fixed parameter controllers. Adaptation of controller parameters to fluxional dynamics of closed-loop system is required to obtain acceptable control performance. For this purpose, intelligent systems such as ANN (Artificial Neural Networks), ANFIS (Adaptive Neuro-Fuzzy Inference Systems) and SVR (Support Vector Regression) can be utilized to design adjustable controllers for nonlinear systems.

Adaptive controller structures based on SVR have proved to be effective controller design methods among other intelligent methods such as ANN, ANFIS because of their superior generalization capabilities, in the last decade. The major

S. Arik et al. (Eds.): ICONIP 2015, Part III, LNCS 9491, pp. 1–8, 2015.
DOI: 10.1007/978-3-319-26555-1_1

strength of SVR is that it ensures global minimum owing to its convex objective function and linear constraints, which avoids getting stuck at local minima.

In technical literature, various controllers based on SVR have been proposed for nonlinear systems such as adaptive PID controller, inverse controller and model predictive control(MPC). Iplikci [3], Shang et al. [4] and Zhao et al. [5] have utilized SVR model of the system to update the parameters of PID controllers. Yuan et al. [6] have proposed a control law based on SVR which is derived via Taylor expansion of system model. Liu et al. [7], Wang et al. [8] and Yuan et al. [9] have deployed SVR as an inverse controller to identify inverse dynamics of the controlled system. In MPC, first and second order derivatives of system output with respect to control input are required. In order to increase accuracy of the required information about system, Iplikci [10,11], Du and Wang [12] and Shin et al. [13] have proposed to utilize SVR in MPC framework.

In this study, an adaptive online SVR controller previously proposed in [1] is used to control the liquid level of a spherical tank system. Two separate SVRs are employed in the control architecture, one for estimating the system model and the other for calculating the control input. The paper is organized as follows: Sect. 2 describes the working prensiples of adaptive SVR controller. In Sect. 3, optimization problem for SVR controller is constructed. In Sect. 4, the effectiveness of the proposed controller has been examined on nonlinear spherical tank system and performance analysis of the controller is given. The paper ends with a brief conclusion in Sect. 5.

2 Adaptive Online SVR Controller

The tuning mechanism of the adaptive SVR controller based on estimated system model is depicted in Fig. 1. The proposed mechanism has two SVR structures; $\text{SVR}_{\text{controller}}$ generates the control input to be applied to the system and $\text{SVR}_{\text{model}}$ is utilized to approximate system behaviour. The control signal produced by online $\text{SVR}_{\text{controller}}$ is computed as:

$$u_n = \sum_{k \in SV_{\text{controller}}} \alpha_k K_{\text{controller}}(\mathbf{\Pi_c}, \mathbf{\Pi_k}) + b_{\text{controller}}. \tag{1}$$

where $\mathbf{\Pi_c}$ is input vector, $K_{\text{controller}}(,.,)$ is the kernel, α_k, $\mathbf{\Pi_k}$ and $b_{\text{controller}}$ are the parameters of the controller to be tuned at time index n. The future behaviour of the controlled system is estimated via $\text{SVR}_{\text{model}}$ as

$$\hat{y}_{n+1} = \sum_{j \in SV_{\text{model}}} \lambda_j K_{\text{model}}(\mathbf{M_c}, \mathbf{M_j}) + b_{\text{model}} \tag{2}$$

where K_{model} is the kernel matrix of the system model, $\mathbf{M_c}$ is current input, and λ_j, $\mathbf{M_j}$ and b_{model} are the parameters of the system model to be adjusted.

The estimation of system model and computation of control input are carried out in two consecutive phases at each sampling interval, namely the training and application phases. In training phase of the controller, $\text{SVR}_{\text{model}}$ is employed to

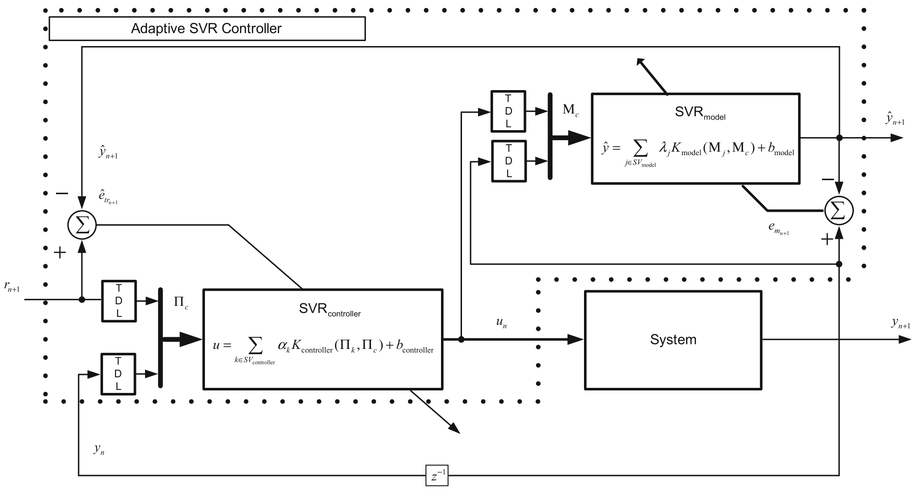

Fig. 1. Adaptive $\text{SVR}_{\text{controller}}$ mechanism.

observe the impact of the tuned controller parameters on closed-loop system performance, and $\text{SVR}_{\text{controller}}$ can be optimized depending on approximated tracking error $(\hat{e}_{tr_{n+1}} = r_{n+1} - \hat{y}_{n+1})$. $\hat{y}_{n+1}$ is the system output estimate calculated by $\text{SVR}_{\text{model}}$. Therefore, while $\text{SVR}_{\text{controller}}$ is in training phase, $\text{SVR}_{\text{model}}$ is in application phase. After the training phase of $\text{SVR}_{\text{controller}}$ is completed, the control signal(u_n) is computed and applied to the system(y_{n+1}) in the application phase of the controller. Thus, the training data pair for $\text{SVR}_{\text{model}}$ $(\mathbf{M_c}, y_{n+1})$ is obtained for training phase of the system model. The parameters of $\text{SVR}_{\text{model}}$ are adjusted via modelling error $e_{m_{n+1}} = y_{n+1} - \hat{y}_{n+1}$. The training algorithm for the overall architecture is explained in [1,14].

The regressor margins of $\text{SVR}_{\text{controller}}$ and $\text{SVR}_{\text{model}}$ of the closed-loop system are illustrated in Fig. 2 where $f_{\text{controller}}$ and f_{model} denote the regression

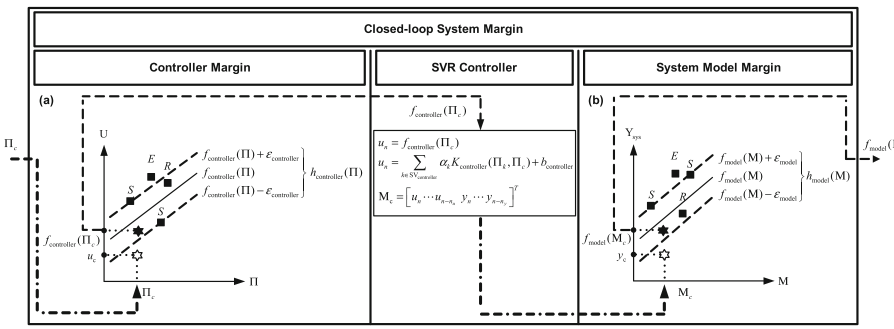

Fig. 2. Margins of $\text{SVR}_{\text{controller}}$(**a**) and $\text{SVR}_{\text{model}}$(**b**).

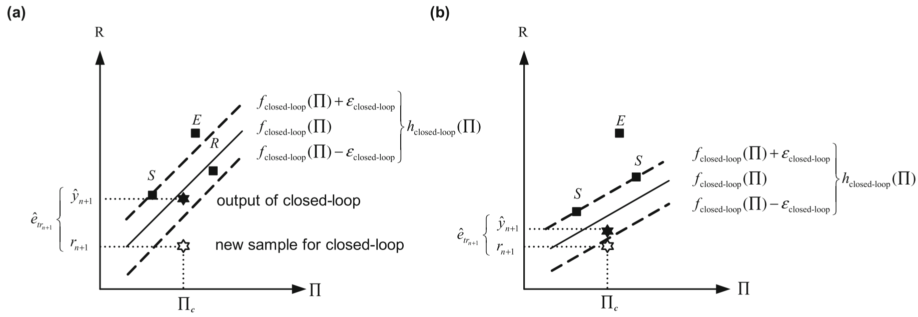

Fig. 3. Projected closed loop margin before(**a**) and after(**b**) training.

functions of controller and system model, respectively. Actually, in training phase of the $\text{SVR}_{\text{model}}$, it can be thought that the output of the system model($\hat{y}_{n+1}$) is forced to track system output (y_{n+1}) by optimizing model parameter and using ($\mathbf{M_c}$,y_{n+1}) as the input-output training data pair. Therefore, the axes for $\text{SVR}_{\text{model}}$ regresion surface are given as $\mathbf{M}$ and Y_{sys} in Fig. 2 (b). Since the control signal that minimize tracking error is unknown, the parameters of $\text{SVR}_{\text{controller}}$ can not be obtained directly. For this reason, closed-loop margin notion which is emerged by combining controller and system model margins has been proposed to optimize controller parameters in [1]. If the margins of the controller and system model are fused, the combined closed-loop margin is projected onto R-$\mathbf{\Pi}$ axes as in Fig. 3. Since the aim in controller design is to force closed-loop system output (y_{n+1}) to track reference signal(r_{n+1}), ($\mathbf{\Pi_c}$,r_{n+1}) data pair has been utilized to optimize closed-loop margin. For this reason, the input-output axes for closed-loop system are defined as $\mathbf{\Pi}$ and R with respect to input-output data pair of closed-loop system as in Fig. 3. That is, the axis R which denotes the reference signal is used in place of Y_{sys} for closed-loop system as in Fig. 3. For more detailed information, it can be consulted to [1].

3 Online ε-SVR for Controller Design

Consider a training data set for the closed-loop system as:

$$\mathbf{T} = \{\mathbf{\Pi_i}, r_{i+1}\}_{i=1}^{N} \quad \mathbf{\Pi_i} \in \mathbf{\Pi} \subseteq R^n, r_{i+1} \in R \tag{3}$$

where N is the size of the training data, n is the dimension of the input, $\mathbf{\Pi_i}$ is input feature vector of controller and r_{i+1} is the reference signal that system is forced to track. The closed-loop error margin function for the i^{th} sample $\mathbf{\Pi_i}$ is described as:

$$h_{\text{closed-loop}}(\mathbf{\Pi_i}) = \hat{y}_{i+1} - r_{i+1} = f_{\text{model}}(\mathbf{M_i}) - r_{i+1} \tag{4}$$

where

$$
\begin{aligned}
\hat{y}_{i+1} &= f_{\text{model}}(\mathbf{M_i}) = \sum_{j \in SV_{\text{model}}} \lambda_j K_{\text{model}}(\mathbf{M_j}, \mathbf{M_i}) + b_{\text{model}} \\
\mathbf{M_i} &= [u_i \cdots u_{i-n_u}, y_i \cdots y_{i-n_y}] \\
u_i &= f_{\text{controller}}(\mathbf{\Pi_i}) = \sum_{k \in SV_{\text{controller}}} \alpha_k K_{\text{controller}}(\mathbf{\Pi_k}, \mathbf{\Pi_i}) + b_{\text{controller}} \\
\mathbf{\Pi_i} &= [r_i \cdots r_{i-n_r}, y_i \cdots y_{i-n_y}, u_{i-1} \cdots u_{i-n_u}]
\end{aligned}
$$

and $\hat{e}_{tr_{i+1}}$ is approximated tracking error. As mentioned before, $\text{SVR}_{\text{model}}$ is utilized to approximate system behaviour, the system model is fixed and system model parameters are known in training phase of the controller. Therefore, the closed loop margin in (4) can be rewritten as

$$
h_{\text{closed-loop}}(\mathbf{\Pi_i}) = \hat{y}_{i+1} - r_{i+1} = f_{\text{closed-loop}}(\mathbf{\Pi_i}) - r_{i+1} = -\hat{e}_{tr_{i+1}} \tag{5}
$$

with respect to an input-output data pair of closed-loop system ($\mathbf{\Pi_i}$,r_{i+1}) where $f_{\text{controller}}$ is the approximated output of the closed-loop system. The main aim is to adjust the unknown parameters of $\text{SVR}_{\text{controller}}$ (α_k,$b_{\text{controller}}$) for the given training samples ($\mathbf{\Pi_i}$,r_{i+1}). Using ($\mathbf{\Pi_i}$,r_{i+1}) data pair and closed-loop error margin defined in (4), online learning rules for the parameters of $\text{SVR}_{\text{controller}}$ (α_k,$b_{\text{controller}}$) can be acquired. The basic idea is to change the coefficient α_c corresponding to the new sample $\mathbf{\Pi_c}$ in a finite number of discrete steps until it meets the KKT conditions while ensuring that the existing samples in $\mathbf{T}$ continue to satify the KKT conditions at each step [14]. The derivation of update rules for controller design are described in detail in [1].

4 Simulation Results

The performance of the controller has been examined on the spherical tank system which is pictured in Fig. 4. Dynamics of the spherical tank system are defined with the following set of differential equation:

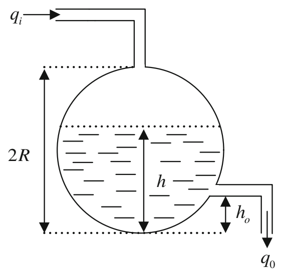

Fig. 4. Spherical tank system.

$$\frac{dh(t)}{dt} = \frac{q_i(t-d) - q_o(t)}{\pi R^2(1 - \frac{R^2 - h(t)}{R^2})}, \ q_0(t) = \sqrt{2g(h(t) - h_0)} \tag{6}$$

where R is the radius of spherical tank, $q_i(t)$ is the input flow rate and control signal, $h(t)$ is the level of the liquid system and controlled output of the system, $q_o(t)$ is the outlet flow rate and d indicates the delay in system. In simulations, the dynamics of the system are defined via fourth order Runge-Kutta method with 0.1 s sampling period, system parameters are chosen as $d = 0\,\text{s}$, $R = 1\,\text{m}$, $h_0 =$

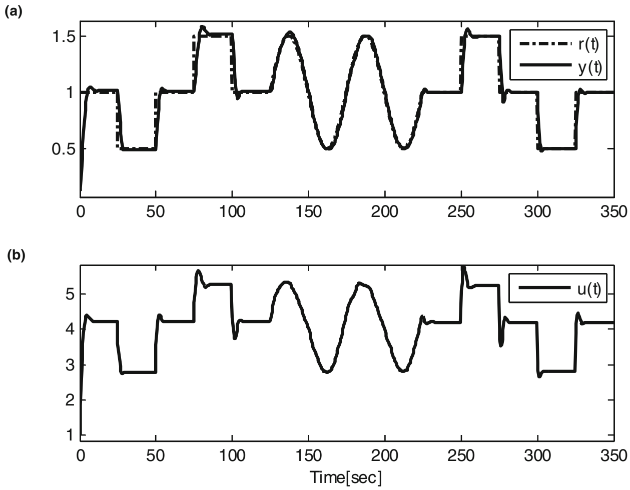

Fig. 5. System (**a**) and controller output (**b**) with no measurement noise.

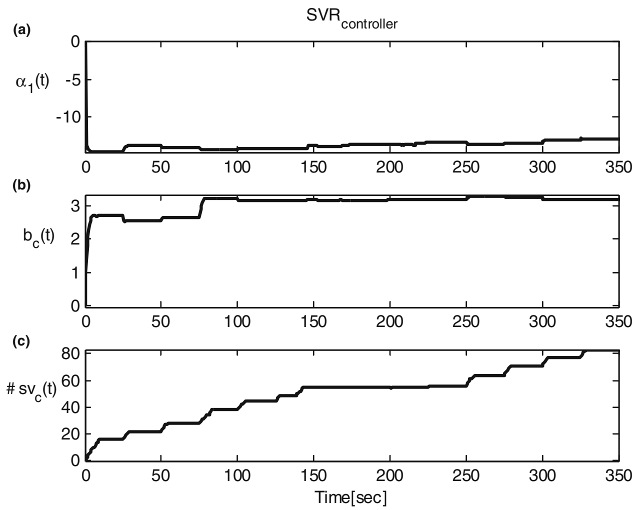

Fig. 6. Adaptation of $\text{SVR}_{\text{controller}}$ parameters.

0.1 m, magnitude of the control signal is allowed to vary between $u_{min} = 0$ and $u_{max} = 6$; and its duration is kept constant at $\tau_{min} = \tau_{max} = 0.5$ s. The input feature vector for $\mathrm{SVR}_{\mathrm{controller}}$ is selected as: $\mathbf{\Pi_c} = [r_n, P_n, I_n, D_n, y_n, u_{n-1}]^T$ where $P_n = e_n - e_{n-1}$, $I_n = e_n$, $D_n = e_n - 2e_{n-1} + e_{n-2}$ and $e_n = r_n - y_n$. In order to identify the dynamics of the controlled system, $\mathrm{SVR}_{\mathrm{model}}$ with $\mathbf{M_c} = [u_n \cdots u_{n-n_u}, y_n \cdots y_{n-n_y}]^T$ as the input feature vector where $n_u = n_y = 1$ is utilized. The closed-loop tracking performance of the controller and the control signal are illustrated in Fig. 5 (a) and (b) respectively. It can be deduced that the closed-loop system has very small transient-state and steady state errors. The first Lagrange multiplier and bias of $\mathrm{SVR}_{\mathrm{controller}}$ are depicted in Fig. 6 (a) and (b) respectively to exemplify the adaptation of the $\mathrm{SVR}_{\mathrm{controller}}$ in order to capture new dynamics [1]. In Fig. 6 (c), the number of the support vectors are illustrated to demonstrate the evolution of $\mathrm{SVR}_{\mathrm{controller}}$.

5 Conclusion

In this paper, liquid level of a spherical tank system has been controlled by an adaptive architecture based on SVR. The control mechanism is composed of two seperate SVR structures where $\mathrm{SVR}_{\mathrm{controller}}$ and $\mathrm{SVR}_{\mathrm{model}}$ are concurrently utilized to compute the control input signal and estimate the system model. The proposed mechanism adjusts $\mathrm{SVR}_{\mathrm{controller}}$ parameters without an explicit knowledge of the control signal applied to the system. The results indicate that the closed-loop system can be successfully forced to track reference signal with small transient and steady-state errors. In future works, new SVR type adaptive controllers can be developed for nonlinear liquid level systems.

References

1. Uçak, K., Günel, G.Ö.: An adaptive support vector regressor controller for nonlinear systems. Soft Comput. (2015). (Article in Press)
2. Aström, K.J., Wittenmark, B.: Adaptive Control. Dover Publications, Mineola (2008)
3. Iplikci, S.: A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems. Int. J. Robust Nonlinear Control **20**(13), 1483–1501 (2010). doi:10.1002/rnc.1524
4. Shang, W.F., Zhao, S.D., Shen, Y.J.: Adaptive PID controller based on online LSSVM identification. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2008), pp. 694–698. IEEE Press, Xian (2008). doi:10.1109/AIM.2008.4601744
5. 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 (CCDC 2009), pp. 7–12. IEEE Press, Guilin (2009)
6. Yuan, X.F., Wang, Y.N., Wu, L.H.: Composite feedforward-feedback controller for generator excitation system. Nonlinear Dyn. **54**(4), 355–364 (2008). doi:10.1007/s11071-008-9334-6

7. Liu, X., Yi, J., Zhao, D.: Adaptive inverse control system based on least squares support vector machines. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3498, pp. 48–53. Springer, Heidelberg (2005)
8. Wang, H., Pi, D.Y., Sun, Y.X.: Online SVM regression algorithm-based adaptive inverse control. Neurocomputing **70**(4–6), 952–959 (2007). doi:10.1016/j.neucom.2006.10.021
9. Yuan, X.F., Wang, Y.N., Wu, L.H.: Adaptive inverse control of excitation system with actuator uncertainty. Neural Process. Lett. **27**(2), 125–136 (2008). doi:10.1007/s11063-007-9064-7
10. Iplikci, S.: Online trained support vector machines-based generalized predictive control of non-linear systems. Int. J. Adapt. Control Signal Process. **20**(10), 599–621 (2006). doi:10.1002/acs.919
11. Iplikci, S.: Support vector machines-based generalized predictive control. Int. J. Robust Nonlinear Control **16**(17), 843–862 (2006). doi:10.1002/rnc.1094
12. Du, Z.Y., Wang, X.F.: Nonlinear generalized predictive control based on online SVR. In: 2nd International Symposium on Intelligent Information Technology Application, pp. 1105–1109. IEEE Press, Shanghai (2008)
13. Shin, J., Kim, H.J., Park, S., Kim, Y.: Model predictive flight control using adaptive support vector regression. Neurocomputing **73**(4–6), 1031–1037 (2010). doi:10.1016/j.neucom.2009.10.002
14. Ma, J., Theiler, J., Perkins, S.: Accurate online support vector regression. Neural Comput. **15**(11), 2683–2703 (2003). doi:10.1162/089976603322385117