Neural Process Lett (2018) 48:419–440
https://doi.org/10.1007/s11063-017-9738-8

Safety-Critical Support Vector Regressor Controller for Nonlinear Systems

Kemal Uçak[1] · **İlker Üstoğlu**[2] · **Gülay Öke Günel**[3]

Published online: 10 November 2017

Abstract In this study, a novel safety-critical online support vector regressor (SVR) controller based on the system model estimated by a separate online SVR is proposed. The parameters of the controller are optimized using closed-loop margin notion proposed in Uçak and Günel (Soft Comput 20(7):2531–2556, 2016). The stability analysis of the closed-loop system has been actualised to design an architecture where operation is interrupted and safety is assured in case of instability. The SVR controller proposed in Uçak and Günel (2016) has been improved to a safety-critical structure by the addition of a failure diagnosis block which carries out Lyapunov stability analysis and detects failures when the overall system becomes unstable. The performance of the proposed method has been evaluated by simulations carried out on a process control system. The results show that the proposed safety-critical SVR controller attains good modelling and control performances and failures arising from instability can be successfully detected.

Keywords Model based adaptive control · Online support vector regression · Safety-critical SVR controller · SVR controller · SVR model identification · Stability analysis

✉ Kemal Uçak
ucak@mu.edu.tr

İlker Üstoğlu
ustoglu@yildiz.edu.tr

Gülay Öke Günel
gulay.oke@itu.edu.tr

1 Department of Electrical and Electronics Engineering, Faculty of Engineering, Muğla Sıtkı Koçman University, Kötekli, 48000 Muğla, Turkey

2 Department of Control and Automation Engineering, Faculty of Electrical-Electronics Engineering, Yildiz Technical University, Esenler, 34220 Istanbul, Turkey

3 Department of Control and Automation Engineering, Faculty of Electrical- Electronics Engineering, Istanbul Technical University, Maslak, 34469 Istanbul, Turkey

1 Introduction

The early detection of failures and faults is crucial to maintain reliability and safety of modern controlled industrial systems. Thus, system shut-down, breakdown and even catastrophes involving human fatalities and material damages can be averted [2]. In case a failure is detected, the system can be provided to complete the operation safely by taking the necessary actions [3].

Modern control systems can handle a variety of constraints arising from nonlinearities and saturations. However, the implementation of control architectures in real time faces significant certification problems, such as guarantee of convergence, time to converge, stability, robust stability and robust performance.

The primary objective of every control system design is safety. Safety means that even in the case of an occuring failure the system should not go into a critical state. Compared to control systems, safety critical systems have additional requirements concerning safety related aspects, e.g., failure correction or safety integrity. It is clear that the design of a safety-critical system is based on detecting and controlling the hazardous actions. A way to control danger is to use blocking and protection devices, which enable hazardous activities only when it is safe and ensure safety up to the unavoidable residual risk. Besides, the safety requirements must clearly specify the hazards that may result from the system and define suitable blocking and protection devices.

A special case is when the safety-critical system itself is the cause of danger and is not controlled by any external device. Safety of these systems can be guaranteed by continuous monitoring of their correct functioning. The software implementation of control laws can be analyzed by simulation, model checking [4], abstract interpretation [5] and by using some theorem proving techniques [6]. These tools have frequently been used in the verification phase of safety critical systems [7].

Model-based control methodologies can be utilized to take precautions in fault diagnosis. However, the main drawback of this class of methods is that they are influenced by the modeling errors. Therefore, the quality of fault diagnosis directly depends on the quality of the model. Owing to their high nonlinear approximation competency, the dynamics of nonlinear systems have frequently been identified via intelligent methods such as Adaptive Neural Networks (ANN), Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and Support Vector Regressors (SVR) to approximate their future behaviour accurately.

In modelling, SVR generally performs superior than ANN and ANFIS since it ensures global minimum while the latter may get stuck at local minima and the model can be obtained only locally [8–14]. Therefore, due to their good prediction ability and generalization performance, SVR based identification and control methods have frequently been applied in recent years to obtain highly accurate models and enhanced controller performance [1].

In technical literature, there exist various controller structures based on SVR modelling. These structures can be examined under two main groups; in the first group, SVR is utilized to optimize conventional controllers and in the second, SVR is employed directly to derive the control law. For instance, Wanfeng et al. [15], Zhao et al. [16] and Iplikci [9] proposed to adjust the parameters of PID controllers via adaptation mechanisms based on system model obtained by SVR where SVR has been utilized to estimate system Jacobian. To exemplify the studies in the second group, inverse controller based on SVR proposed by Liu et al. [17], Wang et al. [18] and Yuan et al. [19], SVR based model predictive controller (MPC) proposed by Iplikci [20,21], Zhiyong and Xianfang [22] and Shin et al. [23] and SVR controller in which SVR is utilized directly as a controller block [1] can be cited.

In this paper, a novel safety-critical SVR controller is proposed to control a nonlinear dynamical system. The controller proposed in [1] has been improved to detect failures resulting from instability. For this purpose, a failure diagnosis block has been integrated to the controller structure.The controller consists of three main parts: SVR controller, SVR model of the system and a failure diagnosis block to detect transition of the system from a stable state to instability. SVR controller parameters are optimized by utilizing the margin between reference input and system output. A second online SVR is used to estimate the model of the system to be controlled; the estimated system output is used to tune the controller parameters. A failure diagnosis block, which is the main contribution of this paper, has been constituted to analyze the stability of the closed-loop system and to maintain the safety of the overall architecture. The failure diagnosis block carries out Lyapunov stability derivations and concludes whether the system is in the stable range or it is transiting towards the unstable range. It signals a stability indicator to provide information about the stability of the system. In case the overall system becomes unstable, the value of the stability indicator changes and the system halts to avoid any hazardous result. SVR model of the system is deployed to observe the possible behaviour of the system in response to controller parameter adjustment as well as to approximate system Jacobian in stability analysis.

The integration of Lyapunov stability analysis to create a safety-critical architecture was initially proposed in [7,24–26]. Those publications concentrated mainly on code generation, and only linear systems were given as examples. States of the system were fed into the Lyapunov analysis block. The main novelty in our work is that we extend this idea to the stability analysis of general nonlinear systems, and more importantly stability analysis of the nonlinear system can be achieved without requiring and observing each state of the system. State information is not needed in Lyapunov calculations, input and output of the system are adequate to conclude about stability. Furthermore, the adaptive structure of the Support Vector Regressor Controller helps to tolerate the instability of the closed loop system to some extent.

The performance of the proposed safety-critical SVR controller has been evaluated by simulations carried out on a process control system. Robustness of the proposed controller has been examined by adding parametric uncertainty to the system. The results indicate that the proposed failure diagnosis block for online SVR controller attains good performance in detecting failures resulting from transition from stable to unstable operation range.

The paper is organized as follows: Sect. 2 describes the basic principles of online ε-SVR. In Sect. 3, the proposed control architecture and failure diagnosis block are explained in detail. Also, the stability analysis of the closed loop system is presented. In Sect. 4, simulation results for the controller are given for a process control system. The paper ends with a brief conclusion in Sect. 5.

2 Online ε-Support Vector Regression

Modeling inaccuracies are the main factor that influence the performance of the adaptive controller structures which are tuned via model based adaptation methodologies. SVR, first asserted by Vapnik et al. [27–29], has been the leading identification method for nonlinear control systems among machine-learning algorithms in recent years since it achieves global minimum and has effective non-linear prediction and generalization competency.

In SVR, first,a non-convex optimization problem in primal form is formulated, then using Lagrange multipliers method, it is converted to a convex objective function with linear constraints, known also as the dual form. Gradient effects which are common in NN and ANFIS are not observed in SVR, due to its convex objective function and global extremum is obtained

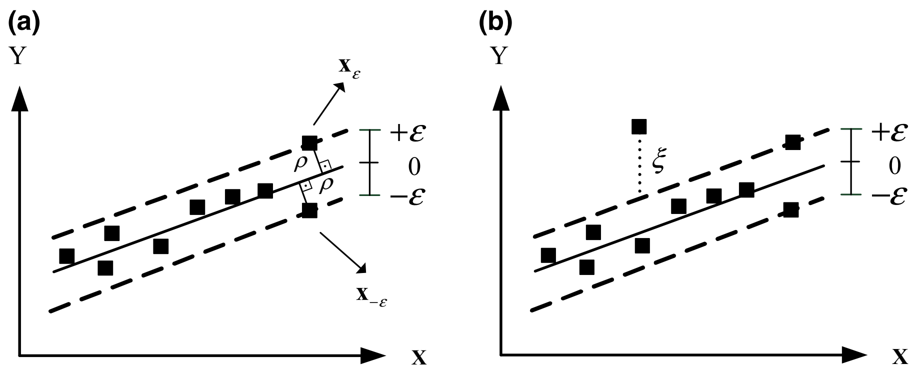

Fig. 1 ε-Support Vector Regression (**a**, **b**), geometric margin (**a**) and slack variables (**b**)

[21]. This leads to identification of system models without errors when SVR is used in model estimation [12].

In this section, a concise description of SVR is given. In Sect. 2.1, the basics of ε-SVR has been presented. The adjustment rules for online ε-SVR are derived in Sect. 2.2.

2.1 An Overview of ε-Support Vector Regression

This subsection briefly reviews the basic principles of support vector regression. Consider a training data set:

$$\mathbf{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{N} \quad \mathbf{x}_i \in \mathbf{X} \subseteq R^n, y_i \in R \tag{1}$$

where N denotes the size of the training data and n is the dimension of the input samples. The data in $\mathbf{T}$ can be represented using a linear regression surface as given in (2) and depictured in Fig. 1a.

$$y_i = \mathbf{w}^T \mathbf{x}_i + b, \quad i = 1, 2, \ldots N \tag{2}$$

where "$\mathbf{w}$" represents the weights of the network, "$\mathbf{x}_i$" is the input data, "b" typifies the bias of regressor and $<, >$ is the inner product [20]. In ε-SVR, ε-error tube representing the deviations of the all training samples from regression surface underpins the construction of the optimization problem. Since ε which can be defined as the maximum tolerable error is initialized to a fixed value at the beginning of the training phase, it can be interpreted as the maximum training error which the SVR network is allowed to have when learning the data. In ε-SVR, the optimization problem is based on the maximization of the geometric margin between the data and regression surface, thus the optimal regression surface is obtained. Using the approximation of the regressor for frontier points $\mathbf{x}_\varepsilon$ and $\mathbf{x}_{-\varepsilon}$, the geometric margin among the outliers ($2\rho = \|\mathbf{x}_\varepsilon - \mathbf{x}_{-\varepsilon}\|$) of the ε-tube can be defined as follows:

$$\begin{aligned}
\hat{y}(\mathbf{x}_\varepsilon) &= \mathbf{w}^T \mathbf{x}_\varepsilon + b = y_r - \varepsilon \\
\hat{y}(\mathbf{x}_{-\varepsilon}) &= \mathbf{w}^T \mathbf{x}_{-\varepsilon} + b = y_r + \varepsilon \\
\hat{y}(\mathbf{x}_\varepsilon) - \hat{y}(\mathbf{x}_{-\varepsilon}) &= \mathbf{w}^T (\mathbf{x}_\varepsilon - \mathbf{x}_{-\varepsilon}) = -2\varepsilon \\
2\rho = \|\mathbf{x}_\varepsilon - \mathbf{x}_{-\varepsilon}\| &= \frac{2\varepsilon}{\|\mathbf{w}\|}
\end{aligned} \tag{3}$$

The aim in ε-SVR is to maximize the geometric margin to obtain the optimal surface. In order to ease the derivation of the primal and dual forms of the problem, the term $\left(\frac{\rho\sqrt{2}}{\varepsilon}\right)^2 = \frac{2\varepsilon}{\|\mathbf{w}\|^2}$ is maximized instead of $\rho = \|\mathbf{x}_\varepsilon - \mathbf{x}_{-\varepsilon}\| = \frac{\varepsilon}{\|\mathbf{w}\|}$. Therefore, primitive form of the primal optimization problem is given as follows:

$$\min_{(\mathbf{w},b)} \quad J_{Pr} = \frac{1}{2}\|\mathbf{w}\|^2 \tag{4}$$

with the following constraints constituted via ε-insensitive loss function

$$\begin{aligned} & y_i - \mathbf{w}^T\mathbf{x}_i - b \le \varepsilon \\ & \mathbf{w}^T\mathbf{x}_i + b - y_i \le \varepsilon \\ & i = 1, 2, \ldots N \end{aligned} \tag{5}$$

Depending on the ε value, as can be seen from Fig. 1b, some samples may stay out of the ε-error tube. These samples can be represented using slack variables ($\xi_i, \xi_i^\star$) and can be integrated to the primal form of the optimization problem as follows:

$$\min_{(\mathbf{w},b,\xi,\xi^\star)} \quad J_{Pr} = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^\star) \tag{6}$$

with the following constraints constituted via ε-insensitive loss function

$$\begin{aligned} & y_i - \mathbf{w}^T\mathbf{x}_i - b \le \varepsilon + \xi_i \\ & \mathbf{w}^T\mathbf{x}_i + b - y_i \le \varepsilon + \xi_i^\star \\ & \xi_i, \xi_i^\star \ge 0\ , \ i = 1, 2, \ldots N \end{aligned} \tag{7}$$

where J_{Pr} indicates primal objective function, ε is the upper value of tolerable error, ξ's and $\xi^\star$'s denote the deviation from ε tube and called as slack variables, and C is a penalty term to optimize slack variables [8,20]. Occassionally, the samples in input space may be nonlinearly distributed. These samples are mapped to a higher dimensional feature space where linear regression can be successfully performed using kernel functions ($\Phi(\mathbf{x_i})$).The objective function of the primal form is non-convex with respect to primal variables ($\mathbf{w}$, b, ξ_i, $\xi_i^\star$).Therefore, in order to obtain a convex representation of the problem in dual form, Lagrangian function is constructed using the primal objective function and corresponding constraints by introducing a dual set of variables as follows:

$$\begin{aligned} L_{Pr} = & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^\star) - \sum_{i=1}^{N}\beta_i(\varepsilon + \xi_i - y_i + \mathbf{w}^T\Phi(\mathbf{x_i}) + b) \\ & - \sum_{i=1}^{N}\beta_i^\star(\varepsilon + \xi_i^\star + y_i - \mathbf{w}^T\Phi(\mathbf{x_i}) - b) - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^\star\xi_i^\star) \end{aligned} \tag{8}$$

A saddle point occurs at the solution with respect to the primal and dual variables. Therefore, first order optimality conditions for L_{Pr} can be acquired as in (7–11) [20,30]:

$$\frac{\partial L_{Pr}}{\partial \mathbf{w}} = 0 \longrightarrow \ \mathbf{w} - \sum_{i=1}^{N}(\beta_i - \beta_i^\star)\mathbf{w}^T\Phi(\mathbf{x_i}) = 0 \tag{9}$$

$$\frac{\partial L_{Pr}}{\partial b} = 0 \longrightarrow \ \sum_{i=1}^{N}(\beta_i - \beta_i^\star) = 0 \tag{10}$$

$$\frac{\partial L_{Pr}}{\partial \xi_i} = 0 \longrightarrow \ C - \beta_i - \eta_i = 0\ , \ i = 1, 2, \ldots N \tag{11}$$

$$\frac{\partial L_{Pr}}{\partial \xi_i^\star} = 0 \longrightarrow \; C - \beta_i^\star - \eta_i^\star = 0 \; , \quad i = 1, 2, \ldots N \tag{12}$$

and the corresponding Karush–Kuhn–Tucker (KKT) complementary conditions are given as [20]

$$\beta_i (y_i - \mathbf{w}^T \Phi(\mathbf{x_i}) - b - \varepsilon - \xi_i) = 0 \; , \quad i = 1, 2, \ldots N \tag{13}$$

$$\beta_i^\star (\mathbf{w}^T \Phi(\mathbf{x_i}) + b - y_i - \varepsilon - \xi_i^\star) = 0 \; , \quad i = 1, 2, \ldots N \tag{14}$$

$$\xi_i \xi_i^\star = 0 \;\; \beta_i \beta_i^\star = 0 \; , \quad i = 1, 2, \ldots N \tag{15}$$

$$\frac{\partial L_{Pr}}{\partial \xi_i^\star} = 0 \longrightarrow \; C - \beta_i^\star - \eta_i^\star = 0 \; , \quad i = 1, 2, \ldots N \tag{16}$$

By substituting the optimality conditions to Lagrangian function, the dual representation and constraints of the problem can be attained as in (17, 18). The optimal parameters of the regressor are obtained by finding the minimum of the following QP problem using training samples in (1).

$$\min_{(\beta,\beta^\star)} J_D = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\beta_i - \beta_i^\star)(\beta_j - \beta_j^\star) K_{ij} + \varepsilon \sum_{i=1}^{N} (\beta_i + \beta_i^\star) - \sum_{i=1}^{N} y_i (\beta_i - \beta_i^\star) \tag{17}$$

with the following constraints

$$\begin{gathered} 0 \le \beta_i \le C \\ 0 \le \beta_i^\star \le C \\ \sum_{i=1}^{N} (\beta_i - \beta_i^\star) = 0 \; , \quad i = 1, 2, \ldots N \end{gathered} \tag{18}$$

where $K_{ij} = \Phi(\mathbf{x_i})^T \Phi(\mathbf{x_j})$ and ε is the upper value of tolerable error [8,20]. As can be seen in optimization problem in (17, 18), the problem has a convex objective function with corresponding linear constraints, so a global solution is ensured. SVR regression model which can represent the data in (1) can be obtained as in (19) by inserting (9) into (2).

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i K(\mathbf{x}_i, \mathbf{x}) + b \; , \;\; \lambda_i = \beta_i - \beta_i^\star \tag{19}$$

where λ are Lagrange multipliers of the regressor, $K(\mathbf{x}_i, \mathbf{x})$ is the kernel function which stores the similarities of the input samples in feature space and b is the bias of the regressor. The training samples $(\mathbf{x}_i)$ with the corresponding Lagrange multiplier $\lambda_i \neq 0$ are called as support vectors [9,20,31].

2.2 Basic Principles of Online ε-Support Vector Regression

In order to derive online learning rules, the Lagrange function constructed via (17, 18) must be solved. Using Lagrange multipliers method, a dual Lagrange function can be formed as in (20).

$$\begin{aligned} L_D = & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\beta_i - \beta_i^\star)(\beta_j - \beta_j^\star) K_{ij} + \varepsilon \sum_{i=1}^{N} (\beta_i + \beta_i^\star) - \sum_{i=1}^{N} y_i (\beta_i - \beta_i^\star) \\ & - \sum_{i=1}^{N} (\delta_i \beta_i + \delta_i^\star \beta_i^\star) - \sum_{i=1}^{N} u_i (C - \beta_i) + u_i^\star (C - \beta_i^\star) + z \sum_{i=1}^{N} (\beta_i - \beta_i^\star) \end{aligned} \tag{20}$$

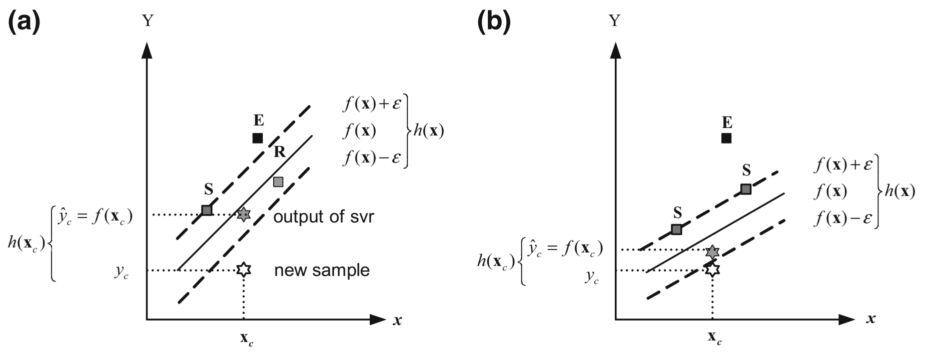

Fig. 2 **E**,**S** and **R** subsets before (**a**) and after (**b**) training

KKT optimality conditions which require that the first order derivatives of Lagrange function with respect to dual variables equal to zero are derived in (21):

$$\begin{aligned}
\frac{\partial L_D}{\partial \beta_i} &= \sum_{j=1}^{N}(\beta_j - \beta_j^{\star})K_{ij} + \varepsilon - y_i - \delta_i + u_i + z = 0 \\
\frac{\partial L_D}{\partial \beta_i^{\star}} &= -\sum_{j=1}^{N}(\beta_j - \beta_j^{\star})K_{ij} + \varepsilon + y_i - \delta_i^{\star} + u_i^{\star} - z = 0 \\
\delta_i^{(\star)} &\geq 0, u_i^{(\star)} \geq 0, \delta_i^{(\star)}\beta_i^{(\star)} = 0, u_i^{(\star)}(C - \beta_i^{(\star)}) = 0
\end{aligned} \tag{21}$$

According to KKT conditions in (21), at most one of the β_i and $\beta_i^{\star}$ should be nonzero and both are nonnegative [30]. The error margin function for the ith sample $\mathbf{x_i}$ to be minimized can be defined as:

$$h(\mathbf{x_i}) = f(\mathbf{x_i}) - y_i = \sum_{j=1}^{N} \lambda_j K_{ij} + b - y_i \tag{22}$$

The training samples in (1) are seperated into three subsets depending on corresponding Lagrange multipliers and margin values [30,32]. These subsets are called as:
Set **E**: Error Support Vectors $\mathbf{E} = \{i|\ |\lambda_i| = C, |h(\mathbf{x_i})| \geq \varepsilon\}$
Set **S**: Margin Support Vectors $\mathbf{S} = \{i|\ 0 < |\lambda_i| < C, |h(\mathbf{x_i})| = 0\}$
Set **R**: Remaining Samples $\mathbf{R} = \{i|\ |\lambda_i| = 0, |h(\mathbf{x_i})| \leq \varepsilon\}$

They are depicted in Fig. 2.When a new sample $\mathbf{x}_c$ is to be learned by the regressor, the Lagrange multipliers of the previously learned samples must be updated and as a result of the adaptation some samples in **E**,**S** and **R** may change their subsets. The aim is to classify $\mathbf{x}_c$ into one of the three sets, while KKT conditions are still satisfied automatically [32]. Firstly, the Lagrange multiplier of the new added data is set to $\lambda_c = 0$, then the value of λ_c is gradually updated so that all other samples satisfy KKT conditions. When $\lambda_c = 0$, the margin for this new data is obtained as

$$h(\mathbf{x_c}) = f(\mathbf{x_c}) - y_c = \sum_{j=1}^{N} \lambda_j K(\mathbf{x}_j, \mathbf{x}_c) + b - y_c \tag{23}$$

The variation in Lagrange multipliers of previously learned samples($\Delta\lambda_j$) , changes in the bias of the regressor Δb and margin values ($\Delta h(\mathbf{x}_i)$) are related as in (24) for the obtained λ_c [30,33].

$$\Delta h(\mathbf{x_i}) = K_{ic}\Delta\lambda_c + \sum_{j=1}^{N} K_{ij}\Delta\lambda_j + \Delta b \tag{24}$$

Since adaptation must prove the constraint in (18), the Lagrange value of the new sample must satisfy (25)

$$\lambda_c + \sum_{j=1}^{N}\lambda_j = 0 \tag{25}$$

If any vector related to previous or new data is an element of the subset **E** or **R**, the corresponding value of the Lagrange multiplier (λ_c) equals to "0" or "C" [1]. If the previously learned sample in **S** remains in subset **S** again, then $\Delta h(\mathbf{x_i}) = 0, i \in \mathbf{S}$ [33]. However, it is required to update the Lagrange values of the samples in subset **S**. If $\Delta h(\mathbf{x_i}) = 0, i \in \mathbf{S}$ in (24),the variations of Lagrange multipliers for the data in the support vector set can be easily computed for the obtained $\Delta\lambda_c$ as follows:

$$\begin{aligned} \sum_{j=1}^{N} K_{ij}\Delta\lambda_j + \Delta b &= -K_{ic}\Delta\lambda_c \\ \sum_{j\in SV} \Delta\lambda_j &= -\Delta\lambda_c \end{aligned} \tag{26}$$

(26) can be expressed in matrix form as

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1s_1} & \cdots & K_{s_1s_k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_ks_1} & \cdots & K_{s_ks_k} \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta\lambda_{s_1} \\ \vdots \\ \Delta\lambda_{s_k} \end{bmatrix} = - \begin{bmatrix} 1 \\ K_{s_1c} \\ \vdots \\ K_{s_kc} \end{bmatrix} \Delta\lambda_c \tag{27}$$

where s_k's indicate the indices of the kth sample in **S**. Thus,

$$\Delta\boldsymbol{\lambda} = \begin{bmatrix} \Delta b \\ \Delta\lambda_{s_1} \\ \vdots \\ \Delta\lambda_{s_k} \end{bmatrix} = \boldsymbol{\beta}\Delta\lambda_c \tag{28}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_k} \end{bmatrix} = -\boldsymbol{\Theta} \begin{bmatrix} 1 \\ K_{s_1c} \\ \vdots \\ K_{s_kc} \end{bmatrix}, \ \boldsymbol{\Theta} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1s_1} & \cdots & K_{s_1s_k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_ks_1} & \cdots & K_{s_ks_k} \end{bmatrix}^{-1} \tag{29}$$

as given in [30]. Thus, the relation between the model parameters of the samples in the support set (**S**) and a given $\Delta\lambda_c$ can be defined via (28–29). As mentioned before, Lagrange multipliers (λ_i) of the samples in subsets **E** or **R** equal to "0" or "C"; therefore, the margin values ($\Delta h(\mathbf{x}_{z_m})$, $z_m \in \mathbf{E}\, or\, \mathbf{R}$) of non-support samples can be calculated as follows:

$$\begin{bmatrix} \Delta h(\mathbf{x}_{z_1}) \\ \Delta h(\mathbf{x}_{z_2}) \\ \vdots \\ \Delta h(\mathbf{x}_{z_m}) \end{bmatrix} = \boldsymbol{\gamma} \Delta\lambda_c \ , \ \boldsymbol{\gamma} = \begin{bmatrix} K_{z_1c} \\ K_{z_2c} \\ \vdots \\ K_{z_mc} \end{bmatrix} + \begin{bmatrix} 1 & K_{z_1s_1} & \cdots & K_{z_1s_l} \\ 1 & K_{z_2s_1} & \cdots & K_{z_2s_l} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{z_ms_1} & \cdots & K_{z_ms_l} \end{bmatrix} \boldsymbol{\beta} \tag{30}$$

where $z_1, z_2, \ldots, z_m$ are the indices of non-support samples, $\boldsymbol{\gamma}$ are margin sensitivities and $\boldsymbol{\gamma} = 0$ for samples in **S**. Up to this point, it is assumed that $\Delta\lambda_c$ is known. $\Delta\lambda_c$ is calculated in two steps. As in all tuning rules, firstly the direction of the update is obtained, then the length of the update is calculated. The first step is to determine whether the change $\Delta\lambda_c$ should be positive or negative as follows [30]:

$$q = sign(\Delta\lambda_c) = sign(y_c - f(\mathbf{x}_c)) = sign(-h(\mathbf{x}_c)) \tag{31}$$

After the sign of the $\Delta\lambda_c$ is specified, in the second step, the bound on $\Delta\lambda_c$ imposed by each sample in the training set is computed [30]. $\Delta\lambda_c$ is calculated as the minimum absolute value among all possible $\Delta\lambda_c$. Thus increment of the current data is

$$\Delta\lambda_c = q\ min(|L_{c_1}|, |L_{c_2}|, |\mathbf{L}^S|, |\mathbf{L}^E|, |\mathbf{L}^R|) \tag{32}$$

where $q = sign(-h(\mathbf{x}_c))$ and L_{c_1} , L_{c_2} are variations of the current sample and $\mathbf{L^S}$, $\mathbf{L^E}$, $\mathbf{L^R}$ are the variations of the $\mathbf{x}_i$ data in sets **S**, **E**, **R** respectively. In order to calculate the largest possible $\Delta\lambda_c$, a bookkeeping procedure which includes five possible cases and takes into account all possible immigrations among subset **S**, **E** and **R** that new added data result in, is performed. The bookkeeping procedure is detailed in [1].

3 Safety-Critical Adaptive SVR Controller

In this paper, the Adaptive Support Vector Regressor Controller proposed in [1] is augmented with a failure diagnosis block and converted to a safety-critical architecture. The failure diagnosis block basically computes the Lyapunov function of the system and its derivative at each iteration, checks for stability conditions and outputs a stability indicator. In case the system becomes unstable, it is stopped to prevent any hazardous events. Section 3.1 briefly reviews the Adaptive Support Vector Regressor Controller which was introduced in [1], and Sect. 3.2 summarizes the Online Support Vector Regression algorithm derived to train the controller. The architecture proposed in this paper, namely the safety-critical online adaptive SVR controller is described in detail in Sect. 3.3.

3.1 An Overview of the Adaptive Support Vector Regressor Controller

The adaptation mechanism of Adaptive Support Vector Regressor Controller proposed in [1] is depicted in Fig. 3. The mechanism is composed of two SVR structures: SVR$_{\text{controller}}$ generates the control input to be applied to the system and SVR$_{\text{model}}$ is utilized to observe the impacts resulting from tuned controller parameters on system behaviour. The parameters of SVR$_{\text{controller}}$ are optimized via approximated tracking error ($\hat{e}_{tr_{n+1}}$) as given in Fig. 3 where SVR$_{\text{model}}$ is used to approximate corresponding system output $\hat{y}_{n+1}$ and SVR$_{\text{model}}$ parameters are adjusted via modelling error $e_{m_{n+1}}$.

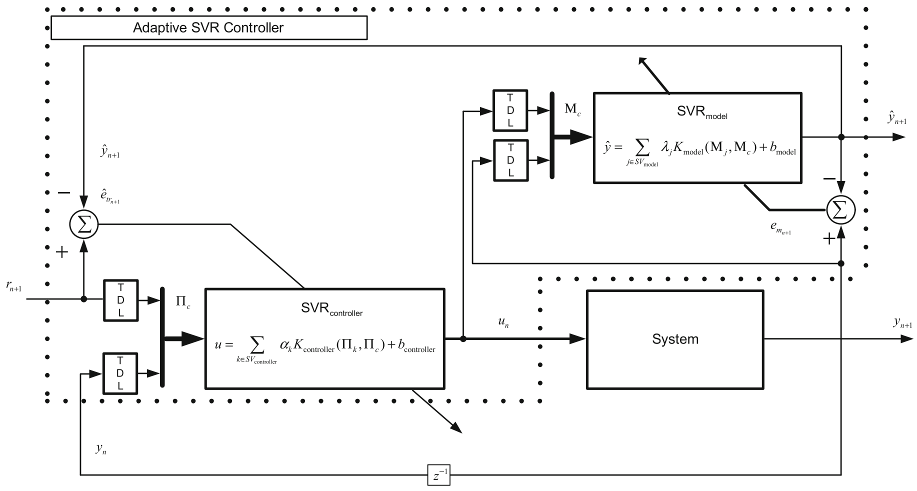
Adaptive SVR Controller
$\hat{y}_{n+1}$
$\hat{e}_{tr_{n+1}}$
−
+
r_{n+1}
TDL
Π_c
$\mathrm{SVR}_{\mathrm{controller}}$
$u = \sum_{k \in SV_{\mathrm{controller}}} \alpha_k K_{\mathrm{controller}}(\Pi_k, \Pi_c) + b_{\mathrm{controller}}$
u_n
M_c
$\mathrm{SVR}_{\mathrm{model}}$
$\hat{y} = \sum_{j \in SV_{\mathrm{model}}} \lambda_j K_{\mathrm{model}}(\mathrm{M}_j, \mathrm{M}_c) + b_{\mathrm{model}}$
$\hat{y}_{n+1}$
$e_{m_{n+1}}$
System
y_{n+1}
y_n
z^{-1}

Fig. 3 The adaptation mechanism of $\mathrm{SVR}_{\mathrm{controller}}$ and $\mathrm{SVR}_{\mathrm{model}}$

The output of $\mathrm{SVR}_{\mathrm{controller}}$ is computed as:

$$u_n = \sum_{k \in SV_{controller}} \alpha_k K_{controller}(\mathbf{\Pi_c}, \mathbf{\Pi_k}) + b_{controller} \tag{33}$$

where $\mathbf{\Pi_c}$ is input vector, $K_{controller}(, .,)$ is the kernel , α_k, $\mathbf{\Pi_k}$ and $b_{controller}$ are the parameters of the controller to be tuned at time index n. The output of $\mathrm{SVR}_{\mathrm{model}}$ is given as

$$\hat{y}_{n+1} = \sum_{j \in SV_{model}} \lambda_j K_{model}(\mathbf{M_c}, \mathbf{M_j}) + b_{model} \tag{34}$$

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. Learning, prediction and control phases are consecutively carried out online both in $\mathrm{SVR}_{\mathrm{controller}}$ and $\mathrm{SVR}_{\mathrm{model}}$. When the parameters of $\mathrm{SVR}_{\mathrm{controller}}$ are optimized, in order to calculate and observe the effect of the computed control signal(u_n) on system behaviour and train $\mathrm{SVR}_{\mathrm{controller}}$ precisely, the computed control signal is applied to $\mathrm{SVR}_{\mathrm{model}}$ at every step of training phase of the controller to predict behaviour of the system (y_{n+1}). It is expected that $\hat{y}_{n+1}$ will eventually converge to y_{n+1} during the course of online working [1]. After the training phase for $\mathrm{SVR}_{\mathrm{controller}}$ is accomplished, the control signal is applied to the system and the input of system model $\mathbf{M_c}$ and output y_{n+1} which are training samples for $\mathrm{SVR}_{\mathrm{model}}$ can be operationalized for training phase of system model.

The input and output of $\mathrm{SVR}_{\mathrm{model}}$, namely the training data pair ($\mathbf{M_c}$, y_{n+1}) is available during online operation, therefore the training process can be performed straightforwardly as explained in Sect. 2.2. However, the training of $\mathrm{SVR}_{\mathrm{controller}}$ is not apparent since the input of $\mathrm{SVR}_{\mathrm{controller}}$ ($\mathbf{\Pi_c}$) is known, but the desired output of the controller, namely the control signal (u_n) is not available to the designer in advance. Therefore, the parameters of the $\mathrm{SVR}_{\mathrm{controller}}$ must be optimized without explicit information of control input (u_n). This situation causes a significant dilemma to train SVR structures without the explicit information of desired output data [34].

Uçak and Öke Günel proposed "closed-loop system margin" notion to solve this situation in order to overcome this dilemma in [1]. The closed-loop margin emerges from the combined

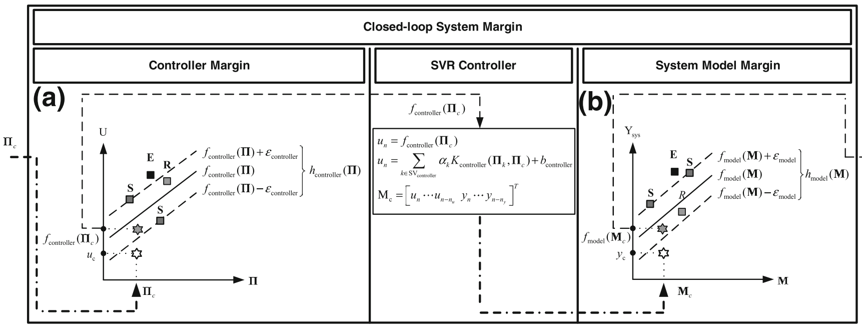
Closed-loop System Margin
Controller Margin
SVR Controller
System Model Margin
(a)
(b)
U
E
R
S
$f_{controller}(\Pi)+\varepsilon_{controller}$
$f_{controller}(\Pi)$
$f_{controller}(\Pi)-\varepsilon_{controller}$
$h_{controller}(\Pi)$
$f_{controller}(\Pi_c)$
u_c
Π
Π_c
$u_n = f_{controller}(\Pi_c)$
$u_n = \sum_{k\in SV_{controller}} \alpha_k K_{controller}(\Pi_k,\Pi_c)+b_{controller}$
$M_c = [u_n \cdots u_{n-n_u}\ y_n \cdots y_{n-n_y}]^T$
Y_{sys}
$f_{model}(M)+\varepsilon_{model}$
$f_{model}(M)$
$f_{model}(M)-\varepsilon_{model}$
$h_{model}(M)$
$f_{model}(M_c)$
y_c
M
M_c

Fig. 4 Margins of $\text{SVR}_{\text{controller}}$ (**a**) and $\text{SVR}_{\text{model}}$ (**b**)

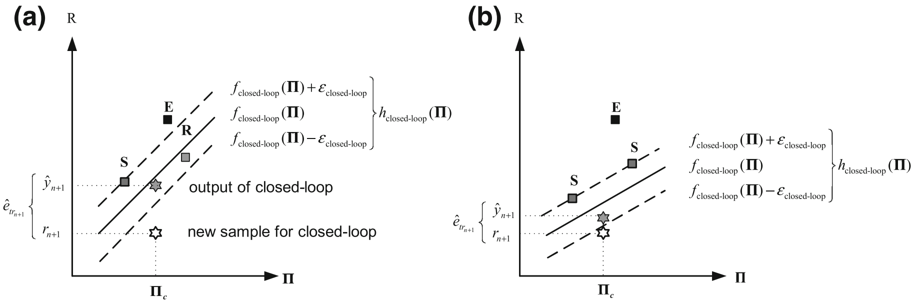
(a)
R
E
R
S
$f_{closed\text{-}loop}(\Pi)+\varepsilon_{closed\text{-}loop}$
$f_{closed\text{-}loop}(\Pi)$
$f_{closed\text{-}loop}(\Pi)-\varepsilon_{closed\text{-}loop}$
$h_{closed\text{-}loop}(\Pi)$
output of closed-loop
new sample for closed-loop
$\hat{y}_{n+1}$
r_{n+1}
$\hat{e}_{tr_{n+1}}$
Π_c
Π
(b)
R
E
S
S

Fig. 5 Projected closed loop margin before (**a**) and after (**b**) training

effects of the margin of the controller and margin of the system model. Considering the adaptation mechanism in Fig. 3, the margin of each subsystem can be depicted as in Fig. 4a, b where $f_{\text{controller}}$ and f_{model} denote the regression functions of controller and system model, respectively. As system model margin is projected onto $\mathbf{M} - Y_{sys}$ in terms of input–output data pair for system model in Fig. 4b, the closed-loop margin is projected onto $\mathbf{\Pi}$ and R axes as in Fig. 5. As explained elaborately in [1], the training data pair $(\mathbf{\Pi}_c, r_{n+1})$ is utilized to force the closed-loop system track the reference input.

3.2 Online Support Vector Regression for Controller Design

Assume that the training data set for closed-loop system is given as:

$$\mathbf{T} = \{\mathbf{\Pi_i}, r_{i+1}\}_{i=1}^{N} \quad \mathbf{\Pi_i} \in \mathbf{\Pi} \subseteq R^n, r_{i+1} \in R \tag{35}$$

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 input–output relationship for closed-loop system can be predicted as:

$$\begin{aligned}
\hat{y}_{i+1} &= f_{model}(\mathbf{M_i}) = \sum_{j\in SV_{model}} \lambda_j K_{model}(\mathbf{M_j}, \mathbf{M_i}) + b_{model}, \ \ \lambda_j = \beta_j - \beta_j^\star \\
\mathbf{M_i} &= [u_i \cdots u_{i-n_u}, y_i \cdots y_{i-n_y}] \\
u_i &= f_{controller}(\mathbf{\Pi_i}) = \sum_{k\in SV_{controller}} \alpha_k K_{controller}(\mathbf{\Pi_k}, \mathbf{\Pi_i}) + b_{controller}
\end{aligned} \tag{36}$$

$$\alpha_k = \theta_k - \theta_k^{\star}$$
$$\mathbf{\Pi_i} = [r_i \cdots r_{i-n_r}, y_i \cdots y_{i-n_y}, u_{i-1} \cdots u_{i-n_u}]$$

where $\mathbf{\Pi_i}$ is the input of the controller and $\hat{y}_{n+1}$ is the approximated system output by SVR$_{\text{controller}}$. The closed-loop error margin function of the system for the ith sample $\mathbf{\Pi_i}$ can be defined as follows in order to optimize controller parameters properly.

$$h_{closed-loop}(\mathbf{\Pi_i}) = \hat{y}_{i+1} - r_{i+1} = f_{model}(\mathbf{M_i}) - r_{i+1} \tag{37}$$

As mentioned before, the parameters of SVR$_{\text{controller}}$ and SVR$_{\text{model}}$ are optimized consecutively. Therefore, in the learning stage of the controller, the system model parameters are known and fixed, so the closed loop margin can be rewritten as

$$h_{closed-loop}(\mathbf{\Pi_i}) = \hat{y}_{i+1} - r_{i+1} = f_{closed-loop}(\mathbf{\Pi_i}) - r_{i+1} = -\,\hat{e}_{tr_{i+1}} \tag{38}$$

with respect to an input–output data pair of closed-loop system ($\mathbf{\Pi_i}$,r_{i+1}) where $f_{closed-loop}$ is the approximated output of the closed-loop system [1]. In training phase of the controller, 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 satisfy the KKT conditions at each step [30]. Since the Lagrange multiplier (α_c) value for training sample which is the element of subset $\mathbf{E}$ or $\mathbf{R}$ equals to "0" or "C", the Lagrange multipliers of the samples in subset $\mathbf{S}$ are optimized. The adjustment rule for samples in subset $\mathbf{S}$ depending on the Lagrange multiplier of the current sample ($\Delta\alpha_c$) is given as

$$\Delta\boldsymbol{\alpha} = \begin{bmatrix} \Delta b_{controller} \\ \Delta\alpha_{s_1} \\ \vdots \\ \Delta\alpha_{s_k} \end{bmatrix} = \boldsymbol{\beta}\Delta\alpha_c \tag{39}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_k} \end{bmatrix} = -\mathbf{\Theta} \begin{bmatrix} 1 \\ K_{controller_{s_1 c}} \\ \vdots \\ K_{controller_{s_k c}} \end{bmatrix}, \ \mathbf{\Theta} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{controller_{s_1 s_1}} & \cdots & K_{controller_{s_1 s_k}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{controller_{s_k s_1}} & \cdots & K_{controller_{s_k s_k}} \end{bmatrix}^{-1} \tag{40}$$

and s_k is the indices of the kth sample in support vector set $\mathbf{S}$. The increment for current data ($\Delta\alpha_c$) is defined as the one with minimum absolute value among all possible $\Delta\alpha_c$ as follows [30]:

$$\Delta\alpha_c = q\ min(|L_{c_1}|, |L_{c_2}|, |\mathbf{L}^S|, |\mathbf{L}^E|, |\mathbf{L}^R|) \tag{41}$$

where $q = sign(-h_{closed-loop}(\mathbf{\Pi_c})) = sign(e_{tr_{n+1}})$ and L_{c_1} , L_{c_2} are variations of the current sample and $\mathbf{L^S}$, $\mathbf{L^E}$, $\mathbf{L^R}$ are the variations of the $\mathbf{\Pi_i}$ data in sets $\mathbf{S}$, $\mathbf{E}$, $\mathbf{R}$ respectively. Since the Lagrange mutipliers of the samples in non-support samples (subset $\mathbf{E}$ or $\mathbf{R}$) are equal to "0" or "C", only the margin values of the non-support samples are influenced by

$\Delta\alpha_c$ and the alternation in margin for non-support samples can be updated via (42)

$$\begin{aligned}
&\begin{bmatrix} \Delta h_{closed-loop}(\mathbf{\Pi}_{n_1}) \\ \Delta h_{closed-loop}(\mathbf{\Pi}_{n_2}) \\ \vdots \\ \Delta h_{closed-loop}(\mathbf{\Pi}_{n_z}) \end{bmatrix} = \boldsymbol{\gamma}\Delta\lambda_c \\
&\boldsymbol{\gamma} = \begin{bmatrix} K_{controller_{n_{1c}}} \\ K_{controller_{n_{2c}}} \\ \vdots \\ K_{controller_{n_{zc}}} \end{bmatrix} + \begin{bmatrix} 1 & K_{controller_{n_1 s_1}} & \cdots & K_{controller_{n_1 s_l}} \\ 1 & K_{controller_{n_2 s_1}} & \cdots & K_{controller_{n_2 s_l}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{controller_{n_z s_1}} & \cdots & K_{controller_{n_z s_l}} \end{bmatrix} \boldsymbol{\beta}
\end{aligned} \tag{42}$$

where $n_1, n_2, \ldots, n_z$ are the indices of non-support samples, $\boldsymbol{\gamma}$ are margin sensitivities and $\boldsymbol{\gamma} = 0$ for samples in $\mathbf{S}$. The recursive algorithm is detailed in [30,32] and [33] for training and forgetting phases.

3.3 Safety-Critical Adaptive Online SVR Controller

The architecture of the proposed safety-critical $\text{SVR}_{\text{controller}}$ is depicted in Fig. 6. In addition to the $\text{SVR}_{\text{controller}}$ and $\text{SVR}_{\text{model}}$ which were explained in detail in Sect. 3.1 and illustrated in Fig. 3, a failure diagnosis block is added to the overall system to observe the problems resulting from instability, which is the main contribution in this paper. The failure diagnosis block takes the inputs and outputs of $\text{SVR}_{\text{controller}}$ and $\text{SVR}_{\text{model}}$ blocks, namely, $\mathbf{\Pi_c}$, u_n, $\mathbf{M_c}$ and $\hat{y}_{n+1}$, and produces the signal δ which is an indicator of the stability of the overall system. The main function of the failure diagnosis block is to carry out the Lyapunov stability analysis of the system as presented in detail in [1], and compute δ accordingly. δ is an indicator of stability, depending on its value the system will continue its functioning or stop in case the system enters a hazardous range. Hence, the safety of the nonlinear control system will be assured. The main advantage of the proposed method is that the stability of the system is analysed without requiring the states of the system since the input–output relationship of the system represented by $\text{SVR}_{\text{model}}$ is adequate for determining stability and designing the controller. In Fig. 6, the failure diagnosis block is utilized to carry out the stability analysis and determine whether the overall system is in stable operation region or it has become unstable. A comprehensive stability analysis of closed-loop system has been derived in [1]. In this sequel, the Lyapunov function is chosen as

$$V(e_{tr_{n+1}}) = \frac{e_{tr_{n+1}}^T \, \mathbf{P} \, e_{tr_{n+1}}}{2} \tag{43}$$

The derivative of the Lyapunov function in (43) is derived as

$$\frac{\partial V(e_{tr_{n+1}})}{\partial t} = -e_{tr_{n+1}}^T \, (\mathbf{G} + \mathbf{Z}) \, e_{tr_{n+1}} \tag{44}$$

with

$$\begin{aligned}
\mathbf{G} &= \mathbf{P}\frac{\partial y_{n+1}}{\partial u_n}\left[\frac{\partial f_{controller}(\boldsymbol{\alpha}, \mathbf{\Pi_c})}{\partial \boldsymbol{\alpha}}\right]^T \boldsymbol{\beta}\,\mu(e_{tr_{n+1}}, \alpha_i, C) \\
\mathbf{Z} &= \mathbf{P}\frac{\partial y_{n+1}}{\partial u_n}\left[\frac{\partial f_{controller}(\boldsymbol{\alpha}, \mathbf{\Pi_c})}{\partial \mathbf{\Pi_c}}\right]^T
\end{aligned} \tag{45}$$

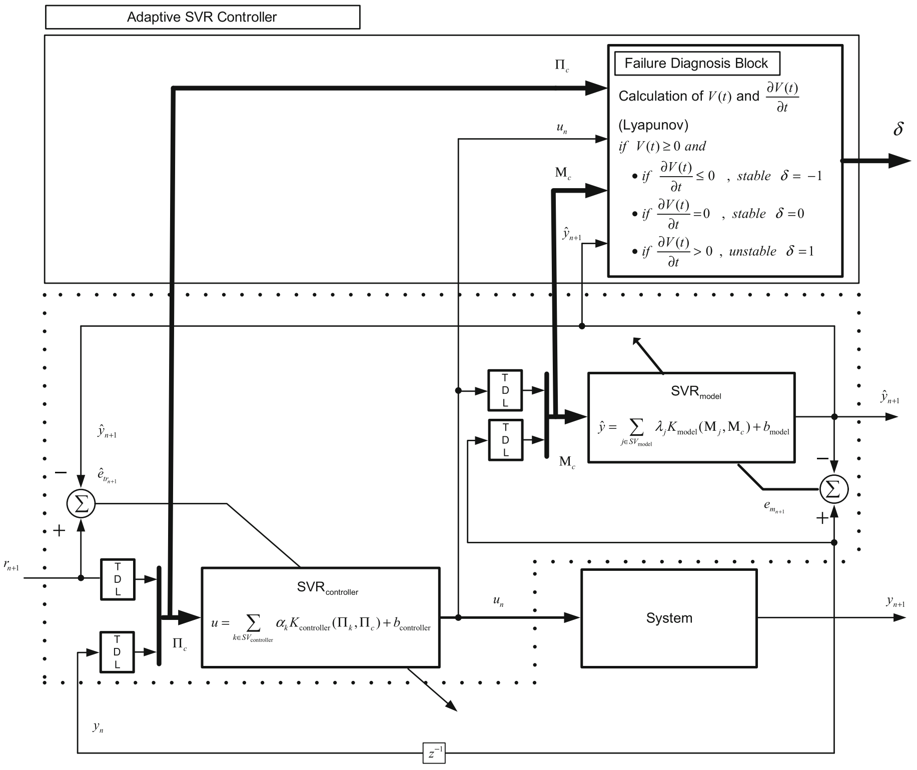
Adaptive SVR Controller
Failure Diagnosis Block
Calculation of $V(t)$ and $\frac{\partial V(t)}{\partial t}$ (Lyapunov)
if $V(t) \geq 0$ and
• if $\frac{\partial V(t)}{\partial t} \leq 0$, stable $\delta = -1$
• if $\frac{\partial V(t)}{\partial t} = 0$, stable $\delta = 0$
• if $\frac{\partial V(t)}{\partial t} > 0$, unstable $\delta = 1$
δ
Π_c
u_n
M_c
$\hat{y}_{n+1}$
SVR$_{model}$
$\hat{y} = \sum_{j \in SV_{model}} \lambda_j K_{model}(M_j, M_c) + b_{model}$
$e_{m_{n+1}}$
$\hat{e}_{tr_{n+1}}$
r_{n+1}
TDL
SVR$_{controller}$
$u = \sum_{k \in SV_{controller}} \alpha_k K_{controller}(\Pi_k, \Pi_c) + b_{controller}$
System
y_{n+1}
y_n
z^{-1}

Fig. 6 Safety-critical adaptive SVR$_{\text{controller}}$

where $\frac{\partial y_{n+1}}{\partial u_n}$ is the system Jacobian, approximated via system model (f_{model}), computed in (45), and $\mu(e_{tr_{n+1}}, \alpha_i, C) = \frac{min(|L_{c_1}|, |L_{c_2}|, |\mathbf{L}^S|, |\mathbf{L}^E|, |\mathbf{L}^R|)}{|e_{tr_{n+1}}|} \geq 0$. Both the stability of the system and the convergence of the controller are guaranteed when $\frac{\partial V}{\partial t} \leq 0$ [35]. Thus, the stability conditions for closed-loop system can be summarized as follows:

- **Condition 1:** If $\mathbf{G} \geq 0$ and $\mathbf{Z} \geq 0$, the closed-loop system is stable
- **Condition 2:** If $\mathbf{G} \geq 0$ and $\mathbf{Z} \leq 0$ and $\| \mathbf{G} \| \geq \| \mathbf{Z} \|$, the closed-loop system is stable
- **Condition 3:** If $\mathbf{G} \leq 0$ and $\mathbf{Z} \geq 0$ and $\| \mathbf{Z} \| \geq \| \mathbf{G} \|$, the closed-loop system is stable

As can be seen from Fig. 6, the failure diagnosis block performs the above stability analysis and outputs δ as a stability indicator parameter. Depending on $\frac{\partial V}{\partial t}$ the value of δ is set as -1, 0 or 1. If $\frac{\partial V}{\partial t} < 0$, δ is assigned as $\delta = -1$. In the case that $\frac{\partial V}{\partial t} = 0$, δ is set to $\delta = 0$. Therefore, the closed-loop system is stable for $\delta = 0$ or $\delta = -1$. In the case that $\frac{\partial V}{\partial t} > 0$, in other words the closed loop system is unstable, δ is set to $\delta = 1$. When $\delta = 1$ the control operation is interrupted and safety is foregrounded.

4 Simulation Results

The performance of the online safety-critical SVR$_{\text{controller}}$ based on system model estimated by SVR$_{\text{model}}$ is examined on a process control system. We have selected the bioreactor benckmark system to evaluate the performance of the proposed method since bioreactor involves highly nonlinear unstable dynamics. Therefore, it is required to control the system actively in order to hinder divergent behaviour since the system involves severe nonlinearity with a tendency to instability [36]. Bioreactor is a challenging benchmark problem used to test the performance of the controller designs in technical literature. Simulations are performed to show how the failure detection block is utilized to determine the stability of the system and how the proposed controller can be used to prevent any hazardous events resulting from instability.

4.1 Bioreactor System

A bioreactor is a tank containing water and cells (e.g., yeast or bacteria) which consume nutrients (substrate) and produce product (both desired and undesired) and more cells [37]. In nonlinear control theory, the performances of the developed control methodologies can be examined and compared on the bioreactor benchmark system since it has highly nonlinear dynamics and exhibits limit cycles [9,20,37,38]. The dynamics of the system can be expressed via the following differential equations

$$\begin{aligned}\dot{c}_1(t) &= -c_1(t)u(t) + c_1(t)(1-c_2(t))e^{\frac{c_2(t)}{\gamma(t)}} \\ \dot{c}_2(t) &= -c_2(t)u(t) + c_1(t)(1-c_2(t))e^{\frac{c_2(t)}{\gamma(t)}}\frac{1+\beta(t)}{1+\beta(t)-c_2(t)}\end{aligned} \tag{46}$$

where $c_1(t)$ is the cell concentration which is the controlled output of the system ($y(t) = c_1(t)$), $c_2(t)$ is the amount of nutrients per unit volume, $u(t)$ is the flow rate used as the control signal, $\gamma(t)$ is nutrient inhibition parameter, $\beta(t)$ is grow rate parameter [9,20,37,38]. The control signal is limited to the range $u_{min} = 0$ and $u_{max} = 2$; and its duration is kept constant at $\tau_{min} = \tau_{max} = 0.5$ s. Simulations have been performed for two separate cases. 1) Nominal case with no parametric uncertainty 2) Parametric uncertainty is introduced to the system to derive the system to the unstable region. For both cases, the input feature vectors for SVR$_{\text{model}}$ and SVR$_{\text{controller}}$ are designated as $\mathbf{M_c} = [u_n \ldots u_{n-n_u}, y_n \ldots y_{n-n_y}]^T$ where $n_y = n_u = 2$ and $\mathbf{\Pi_c} = [r_n \ldots r_{n-n_r}, y_n \ldots y_{n-n_y}, u_{n-1} \ldots u_{n-n_u}]^T$ where n_r, n_y and n_u express the number of the past features. The exponential kernel parameters of SVR$_{\text{model}}$ and SVR$_{\text{controller}}$ are chosen as $\sigma = 0.75$, error tolerance parameters are utilized as $\varepsilon_{closed-loop} = \varepsilon_{model} = 10^{-3}$ and C is fixed at 1000.

4.2 Nominal Case with No Parametric Uncertainty

The tracking performance of the controller for noiseless condition is depicted in Fig. 7. It is observed that the reference signal is tracked accurately. The parameters of SVR$_{\text{controller}}$ and SVR$_{\text{model}}$ are illustrated in Fig. 8. In Fig. 8, $\alpha_1(t)$, $b_c(t)$ and $\#sv_c$ stand for the first Lagrange multiplier, bias of the regression function and number of the support vectors for SVR$_{\text{controller}}$, respectively. Similarly, the first lagrange multiplier, bias of the regression function and number of the support vectors for SVR$_{\text{model}}$ are denoted as $\lambda_1(t)$, $b_m(t)$ and $\#sv_m$. In Fig. 9, $\frac{\partial V}{\partial t}$, the time derivative of the Lyapunov function is depicted. For stability, both $V(t) > 0$ and $\frac{\partial V}{\partial t} \leq 0$ must be satisfied simultaneously. Since in Fig. 9, it is observed that $\frac{\partial V}{\partial t} \leq 0$ and $\delta = -1$ or 0 during the course of control, we can conclude that the closed-loop system is stable.

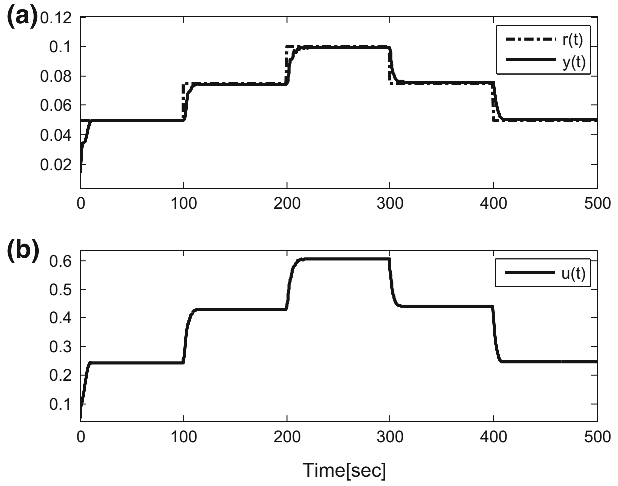

Fig. 7 System output, $y(t)$ (**a**), control signal, $u(t)$ (**b**) for variable step input for the nominal case without parametric uncertainty

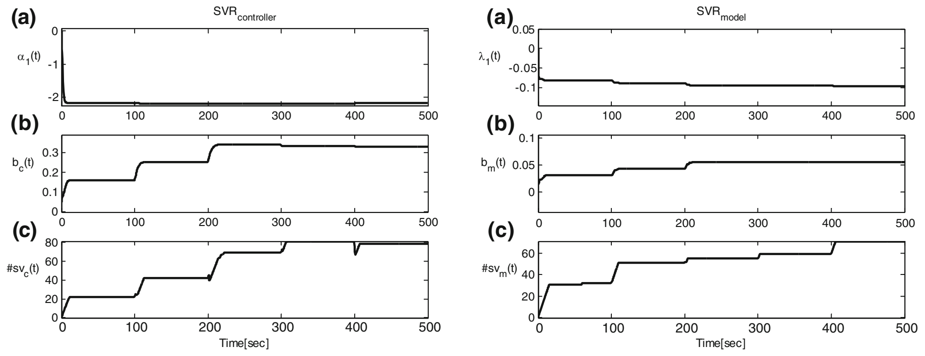

Fig. 8 The first Lagrange multiplier, $\alpha_1(t)$ (**a**), bias value, $b_c(t)$ (**b**) and number of support vectors, $\#sv_c(t)$(**c**) for SVR$_{\text{controller}}$ (*left*), the first Lagrange multiplier, $\lambda_1(t)$ (**a**), bias value, $b_m(t)$ (**b**) and number of support vectors, $\#sv_m(t)$ (**c**) for SVR$_{\text{model}}$ (*right*) for the nominal case without parametric uncertainty

4.3 Uncertainty in System Parameters

In order to observe the stability of the system for the case with parameter uncertainty, $\gamma(t)$ is considered as the time-varying parameter of the system, where its nominal value is $\gamma_{nom}(t) = 0.48$ and it is allowed to vary slowly in the purlieu of its nominal value as $\gamma(t) = 0.48 + 0.06\sin(0.004\pi t)$ as depicted in Fig. 10c. Fig. 10 illustrates the tracking performance of SVR$_{\text{controller}}$, the control signal applied to the system and the time-varying system parameter. Parameters of SVR$_{\text{controller}}$ and SVR$_{\text{model}}$ are depicted in Fig. 11. In Fig. 11, $\alpha_1(t)$, $b_c(t)$ and $\#sv_c$ stand for the first lagrange multiplier, bias of the regression function and number of the support vectors for SVR$_{\text{controller}}$, respectively. The first lagrange multiplier, bias of the regression function and number of the support vectors for SVR$_{\text{model}}$ are denoted as $\lambda_1(t)$, $b_m(t)$ and $\#sv_m$. The closed-loop system including uncertainty in system parameter is successfully controlled and maintained in the stable range as depicted in Fig. 12. When

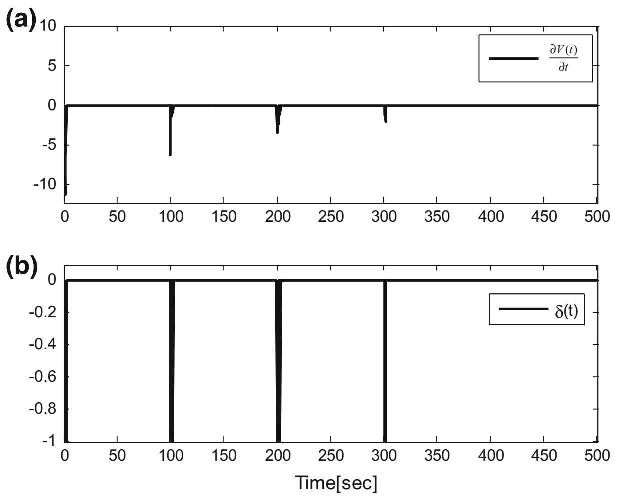

Fig. 9 Time derivative of Lyapunov function, $\frac{\partial V}{\partial t}$ (**a**) and stability indicator, $\delta(t)$ (**b**) for the nominal case without parametric uncertainty

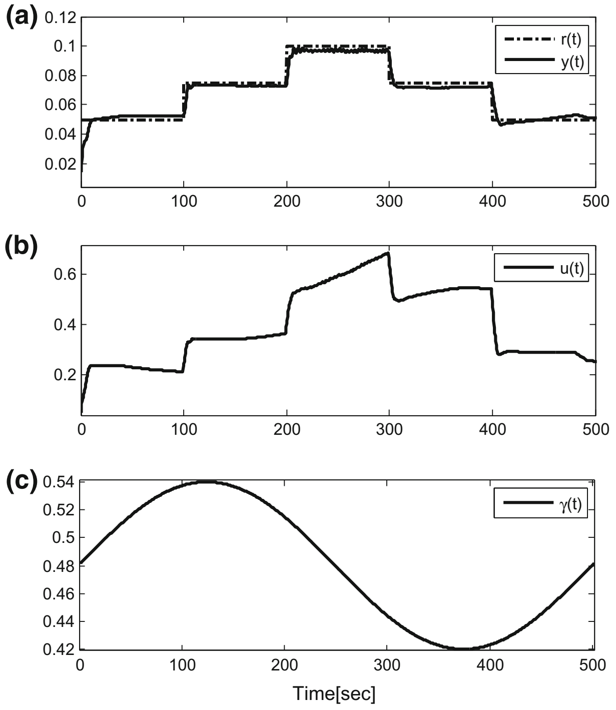

Fig. 10 System output, $y(t)$ (**a**), control signal, $u(t)$ (**b**), time varying parameter, $\gamma(t)$ (**c**) for the case with parameteric uncertainty

the control signal produced for nominal system parameters in Fig. 7 and for the time varying parameter situation in Fig. 10 are compared, it can be observed how the control signal in Fig. 10 tries to tolerate the uncertainty of the time varying system parameter.

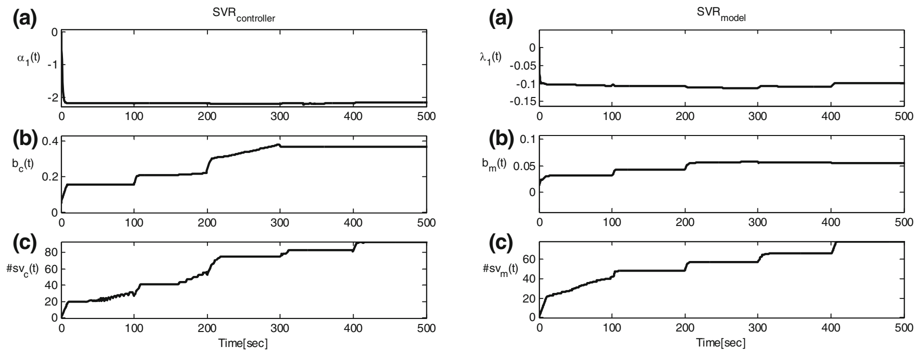

Fig. 11 The first Lagrange multiplier, $\alpha_1(t)$ (**a**), bias value, $b_c(t)$ (**b**) and number of support vectors, $\#sv_c(t)$ (**c**) for SVR$_{\text{controller}}$ (*left*), the first Lagrange multiplier, $\lambda_1(t)$ (**a**), bias value, $b_m(t)$ (**b**) and number of support vectors, $\#sv_m(t)$(**c**) for SVR$_{\text{model}}$ (*right*) for the case with parameteric uncertainty

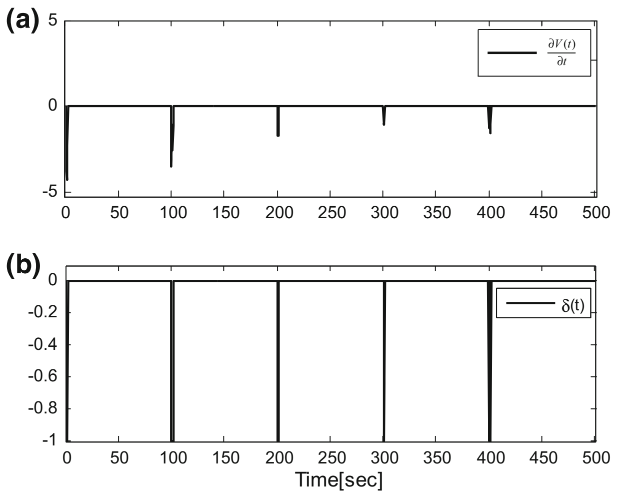

Fig. 12 Time derivative of Lyapunov function, $\frac{\partial V}{\partial t}$ (**a**) and stability indicator, $\delta(t)$ (**b**) for the case with parameteric uncertainty

4.4 Closed-loop Lyapunov Stability Analysis

In the simulations performed in this section, the parameters of system are deliberately alternated to force system towards the unstable region where the adaptive mechanism can not manage to control the system. The tracking performance of the system is illustrated in Fig. 13. The parameters of the system are changed in two regions: at 130th sec and 350th sec as shown in Fig. 13. The time derivative of the Lyapunov function and the stability indicator δ are depicted in Fig. 14. There are three cases to be examined in Fig. 14. These cases are given as follows:

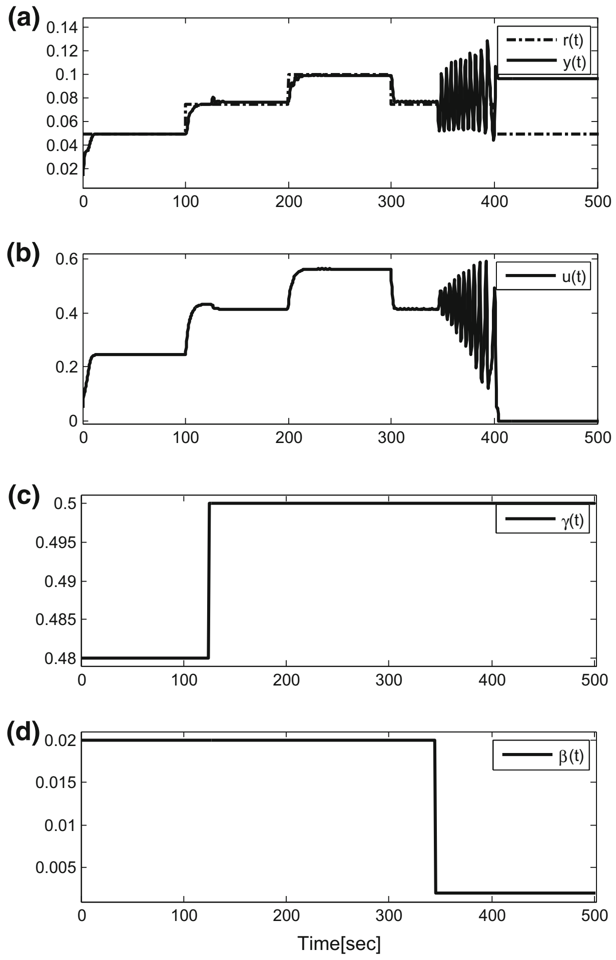

Fig. 13 System output, $y(t)$ (**a**), control signal, $u(t)$ (**b**), time varying parameter, $\gamma(t)$ (**c**), $\beta(t)$ (**d**)

- **Case 1:** The system parameter $\gamma(t) = 0.48$ is switched to $\gamma(t) = 0.5$ at 150th sec. The system becomes unstable and the adaptive structure of controller succesfully deals with unstability within 20 sec.
- **Case 2:** The system becomes unstable transiently at 200th and 210th secs resulting from alternation in reference signal, but the controller has overcome this situation.
- **Case 3:** The parameters of system are changed as $\gamma(t) = 0.5$, $\beta = 2 \times 10^{-3}$ at 350th sec. The controller endeavours to eliminate unstability, but it can not be achieved.

In a nutshell, as can be seen from Fig. 14b, the system becomes unstable at times as in case 1–2, but the adaptive mechanism of the controller can immediately manage to derive system to the region where closed-loop system is stable. However, for cases where the adaptive mechanism is inadequate to stabilize the system, as in case 3, the failure diagnosis block of the proposed mechanism given in Fig. 6 detects the situation, the control operation is interrupted and safety is foregrounded.

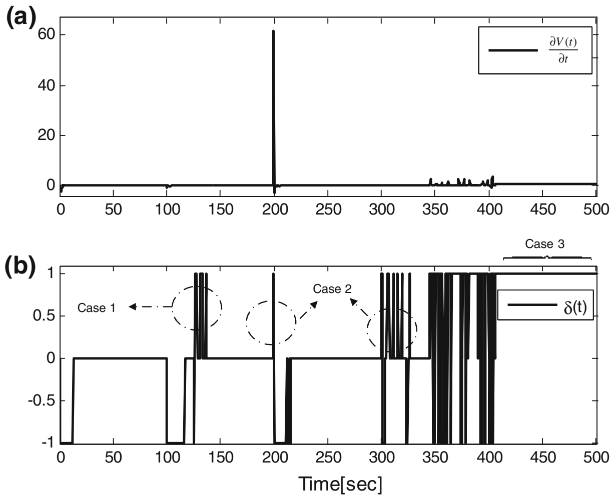

Fig. 14 Time derivative of Lyapunov function, $\frac{\partial V}{\partial t}$ (**a**) and stability indicator, $\delta(t)$ (**b**) for case 1, 2 and 3

5 Conclusion

In this paper, a novel safety-critical SVR$_{\text{controller}}$ has been proposed by combining a failure diagnosis block to the controller in [1]. Failures resulting from instability can be detected via Lyapunov stability analysis of the overall system. Owing to the adaptive structure of the controller, the proposed mechanism achieves to tolerate the instability of the closed loop system to some extent. To evaluate the performance of the failure diagnosis block, simulations have been performed where the stability of closed-loop system has intentionally been ruined and the proposed mechanism to detect the instability of the system is tested. In future works, it is planned to realize the proposed controller structure on ANSYS SCADE [39], one of the commercially available autocoding environments to obtain a more reliable and predictable code and to develop new SVR based fail-safe controllers for nonlinear systems.

References

1. Uçak K, Öke Günel G (2016) An adaptive support vector regressor controller for nonlinear systems. Soft Comput 20(7):2531–2556. https://doi.org/10.1007/s00500-015-1654-0
2. Patton RJ, Chen J, Siew TM (1994) Fault diagnosis in nonlinear dynamic systems via neural networks. In: International conference on control. Coventry, England
3. Ucak K, Caliskan F, Oke G (2013) Fault diagnosis in a nonlinear three-tank system via ANFIS. In: International conference on electrical and electronics engineering (ELECO 2013). Bursa, Turkey
4. Baier C, Katoen JP (2007) Principles of model checking. The MIT Press, London
5. Cousot P (200) Abstract interpretation based formal methods and future challenges. In: International conference on informatics—10 years back, 10 years ahead. Dagstuhl, Germany
6. Clavel M, Duran F, Hendrix J, Lucas S, Meseguer J, Olveczky P (2007) The Maude formal tool environment. In: International conference on algebra and coalgebra in computer science. Bergen, Norway
7. Feron E (2010) From control systems to control software. IEEE Control Syst 30(6):50–71. https://doi.org/10.1109/MCS.2010.938196
8. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88
9. Iplikci S (2010) 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

10. Lee M-C, To C (2010) Comparison of support vector machine and back propagation neural network in evaluating the enterprise financial distress. Int J Artif Intell Appl 1(3):31–43
11. Collazo RA, Pessôa LAM, Bahiense L, de Pereira B, Reis AF, Silva NS (2016) A comparative study between artificial neural network and support vector machine for acute coronary syndrome prognosis. Pesquisa Operacional 36(2):321–343
12. Shao Y, Lunetta RS (2012) Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. ISPRS J Photogramm Remote Sens 70:78–87
13. Sheta AF, Ahmed SEM, Faris H (2015) A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. Int J Adv Res Artif Intell 4(7):55–63
14. Pan S, Iplikci S, Warwick K, Aziz TZ (2012) Parkinson's disease tremor classification—a comparison between support vector machines and neural networks. Expert Syst Appl 39(12):10764–10771
15. Wanfeng S, Shengdun Z, Yajing S (2008) Adaptive PID controller based on online LSSVM identification. In: IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2008). Xian, China
16. Zhao J, Li P, Wang XS (2009) Intelligent PID controller design with adaptive criterion adjustment via least squares support vector machine. In: 21st Chinese control and decision conference (CCDC 2009). Guilin, China
17. Liu X, Yi J, Zhao D (2005) Adaptive inverse control system based on least squares support vector machines. In: 2nd International symposium on neural networks (ISNN 2005). Chongqing, China
18. Wang H, Pi DY, Sun YX (2007) Online SVM regression algorithm-based adaptive inverse control. Neurocomputing 70(4–6):952–959. https://doi.org/10.1016/j.neucom.2006.10.021
19. Yuan XF, Wang YN, Wu LH (2008b) Adaptive inverse control of excitation system with actuator uncertainty. Neural Process Lett 27(2):125–136. https://doi.org/10.1007/s11063-007-9064-7
20. Iplikci S (2006a) Online trained support vector machines-based generalized predictive control of non-linear systems. Int J Adapt Control Signal Process 20(10):599–621. https://doi.org/10.1002/acs.919
21. Iplikci S (2006b) Support vector machines-based generalized predictive control. Int J Robust Nonlinear Control 16(17):843–862. https://doi.org/10.1002/rnc.1094
22. Zhiying D, Xianfang W (2008) Nonlinear generalized predictive control based on online SVR. In: 2nd International symposium on intelligent information technology application. Shanghai, China
23. Shin J, Kim HJ, Park S, Kim Y (2010) Model predictive flight control using adaptive support vector regression. Neurocomputing 73(4–6):1031–1037. https://doi.org/10.1016/j.neucom.2009.10.002
24. Wang T, Jobredeaux R, Herencia H, Garoche PL, Dieumegard A, Feron E, Pantel M (2012) From design to implementation: an automated, credible autocoding chain for control systems. In: Workshop on advances in control system technology for aerospace applications. Atlanta, GA
25. Feron E, Jobredeaux R, Wang T (2011) Autocoding control software with proofs I: annotation translation. In: IEEE/AIAA 30th digital avionics systems conference (DASC) on closing the generation gap—increasing capability for flight operations among legacy, modern and uninhabited aircraft. Seattle, WA
26. Wang T, Jobredeaux R, Feron E (2011) A graphical environment to express the semantics of control systems. Comput Res Respository, http://arxiv.org/abs/1108.4048
27. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
28. Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V (1996) Support vector regression machines. In: 10th annual conference on neural information processing systems (NIPS). Denver, CO
29. Vapnik V, Golowich SE, Smola A (1997) Support vector method for function approximation, regression estimation, and signal processing. In: Annual conference on neural information processing systems (NIPS), Denver, CO
30. Ma J, Theiler J, Perkins S (2003) Accurate online support vector regression. Neural Comput 15(11):2683–2703
31. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other Kernel-based learning methods. Cambridge University Press, Cambridge
32. Wang X, Du Z, Chen J, Pan F (2009) Dynamic modeling of biotechnical process based on online support vector machine. J Comput 4(3):251–258. https://doi.org/10.4304/jcp.4.3.251-258
33. Mario M (2002) On-line support vector machine regression. In: 13th European conference on machine learning (ECML 2002). Helsinki, Finland
34. Uçak K, Günel GO (2016) Generalized self-tuning regulator based on online support vector regression. Neural Comput Appl. https://doi.org/10.1007/s00521-016-2387-4, (**Accepted**)
35. Saadia N, Amirat Y, Pontnau J, M'Sirdi NK (2001) Neural hybrid control of manipulators, stability analysis. Robotica 19:41–51. https://doi.org/10.1017/S0263574700002885
36. Puskorius GV, Feldkamp LA (1994) Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. IEEE Trans Neural Netw 5(2):279–297

37. Ungar LH (1990) Neural networks for control. In: Miller III WT, Sutton RS, Werbos PJ (eds) A bioreactor benchmark for adaptive network based process control. MIT Press, Cambridge, pp 387–402
38. Efe MO (2007) Discrete time fuzzy sliding mode control of a biochemical process. In: 9th WSEAS international conference on automatic control, modeling and simulation (ACMOS'07). Istanbul, Turkey
39. ANSYS® SCADE Suite ®17.0