Neural Comput & Applic (2015) 26:723–734
DOI 10.1007/s00521-014-1754-2

ORIGINAL ARTICLE

Black box modeling of PIDs implemented in PLCs without structural information: a support vector regression approach

Robert Salat · Michal Awtoniuk

Received: 6 December 2013 / Accepted: 16 October 2014 / Published online: 26 October 2014

Abstract In this report, the parameters identification of a proportional–integral–derivative (PID) algorithm implemented in a programmable logic controller (PLC) using support vector regression (SVR) is presented. This report focuses on a black box model of the PID with additional functions and modifications provided by the manufacturers and without information on the exact structure. The process of feature selection and its impact on the training and testing abilities are emphasized. The method was tested on a real PLC (Siemens and General Electric) with the implemented PID. The results show that the SVR maps the function of the PID algorithms and the modifications introduced by the manufacturer of the PLC with high accuracy. With this approach, the simulation results can be directly used to tune the PID algorithms in the PLC. The method is sufficiently universal in that it can be applied to any PI or PID algorithm implemented in the PLC with additional functions and modifications that were previously considered to be trade secrets. This method can also be an alternative for engineers who need to tune the PID and do not have any such information on the structure and cannot use the default settings for the known structures.

Keywords Support vector regression · Programmable logic controller · PID

R. Salat (✉) · M. Awtoniuk
Department of Production Engineering, Warsaw University of Life Sciences, Nowoursynowska 166, 02-787 Warsaw, Poland
e-mail: robert_salat@sggw.pl

M. Awtoniuk
e-mail: michal_awtoniuk@sggw.pl

1 Introduction

Proportional–integral–derivative (PID) algorithms are widely used for the control of industrial process loops. Due to their simplicity and ease of on-line re-tuning, approximately 90 % of control loops use the PID algorithm. Among the control loops that use the PID algorithm, 64 % are single loop and 36 % are multi-loop [1]. Approximately 85 % of control systems that use PID algorithms are the feedback type, up to 6 % are the feedforward type, and 9 % are connected in a cascade [1, 2]. PID algorithms implemented as a technical device are called PID controllers. Currently, the PID controller structures are different from the original analogue PID [3]. Presently, the implementation of the PID is based on a digital design. These digital PID controllers include several additional functions to improve their performance, such as anti-windup, set point filtering, auto-tuning, adaptive algorithms, fuzzy fine-tuning, genetic tuning, and so on [4]. The controllers come in several different forms, such as a standard single-loop controller, known as a dedicated process controller and a software component in the programmable logic controller, known as a programmable logic controller (PLC) [5], as well as in built-in controller machines, e.g., robots [6]. PID controllers are used in a wide range of application, such as process control, flight control, automotive control, motor drives, and so on. The PID algorithms found in industry may have different structures [3]. Currently, the three largest classes of PID algorithms implemented in the PLC are the ideal standard algorithm (ISA, non-interacting), parallel (non-interacting), and series (interacting) types. The general expressions of the PID algorithms are represented by (1, 2, 3) as follows:

$$u = K_C\left(1 + \frac{1}{T_I s} + T_D s\right) \quad (1)$$

$$u = K_P + \frac{K_I}{s} + K_D s \quad (2)$$

$$u = K'_C\left(\frac{1}{T'_I s} + 1\right)\left(T'_D s + 1\right) \quad (3)$$

where K_C is the proportional gain of the ISA; T_I is the reset time of the ISA; T_D is the derivative time of the ISA; K_P is the proportional gain of the parallel form; K_I is the integral gain of the parallel form; K_D is the derivative gain of the parallel form; K'_C is the proportional gain of the series form; T'_I is the reset time of the series form; and T'_D is the derivative time of the series form.

Previously, the parallel and ISA algorithms were less commonly observed in industrial processes compared to the series form. The reason for this phenomenon can be found in analog control, where pneumatic controllers were dominant. When pneumatic controllers were dominant, the PID algorithm was difficult to design due to its use of extremely expensive analog amplifiers [7]. Despite this drawback, Astrom and Hagglund [2] indicated that the ISA allows complex zeroes and is thus a more flexible structure than the series algorithm, which has real zeroes. Many controller manufacturers (ABB, Allen-Bradley, General Electric, Honeywell, Omron, Siemens, Toshiba, Yokogawa, etc.) offer a variety of modified versions of the above-mentioned forms, where a few of the modifications are improvements of the structures, a few of the modifications are derived from early pneumatic implementations, and a few of the modifications are more common in certain industries than others. According to [2], many useful features of PID control have not been widely disseminated because they are considered to be trade secrets. Typical examples include techniques for mode switches and anti-windup. However, the basic actions remain the same. The main issue is that the tuning behavior varies from one form to another. An understanding of the various forms of PID algorithms and the configuration options that are offered is necessary to properly design and apply process control strategies. For this type of modified structure, there are no readily available tuning rules [8, 9]. Rhinehart and Shinskey [3] reported that an operator who was accustomed to tuning a controller with a particular PID algorithm would be baffled when another controller did not respond as expected. An operator or software program that follows a tuning procedure to determine K_C, K_P, or K'_C; T_I, K_I, or T'_I; and T_D, K_D, or T'_D for the standard algorithm (1, 2, 3) could be surprised by the response when applying the procedure to a manufacturer-specific version.

Therefore, a PID controller's structure should be completely understood before it is tuned [10]. Furthermore, the structural difference becomes significant when one controller is replaced by another. A variety of structure identification methods are under development. A popular and frequently used method is the relay feedback for both the off-line and on-line automatic identification of the PID [7]; however, relay feedback has disadvantages because it is unacceptable for a few classes of processes, such as unstable and integrals of the second-order processes. A different approach to identify PI structures was shown in [11], where artificial intelligence was used and PI algorithms were treated as a black box [12].

In this report, we propose using the support vector regression (SVR) [e.g., the support vector machine (SVM) in regression mode] as a tool for PID-implemented modeling in a real PLC. According to [8], about fifty non-standard PID structures very often can be seen in the real PLC. Thus, this method can be an alternative for engineers who need to tune the PID and do not have any such information on the structure and cannot use the default settings for the known structures. Preliminary research for this method was presented in [11] and was limited to the PI algorithm.

In the first stage, we used the SVR for training PID algorithm structures based on input and output signals from the PLC (black box model). The goal was to train a SVR and obtain the response comparable to the response of the PID algorithm implemented in the PLC for any parameters K_P, T_I, and T_D. The advantage of this method is that the structures with modification can then be both known and unknown. The benefits of this method also indicate that after the training of the SVR, we can simulate the real output of the PID algorithm for a personal computer (PC) and apply, for example, an imperialist competitive algorithm (ICA) to tune the PID parameters and transfer them to the PLC.

We used the SVR because it is a good tool to estimate regression functions with generalization performances when using structural risk minimization [13]. One of the problems of using the SVR is that a large number of samples are gathered with the PLC in addition to a few of the features selected in mode training and testing. In this report, we propose the selection of an optimal feature vector and reduction samples to be trained. We focus on two industrial controllers: Siemens and General Electric (GE), the basic algorithms of which have already been described (1, 2). To the best of our knowledge, SVR has never been used in the context of PID algorithms implemented in the PLC.

The method is sufficiently universal and can be applied to any PI, PD, and PID algorithm implemented in the PLC, with additional functions and modifications that are considered to be trade secrets. The report is organized as follows. The structures of the PID algorithms implemented in the PLC are described in Sect. 2. The short studies on the SVR are described in Sect. 3. The proposed feature selection for the SVR is described in Sect. 4. Section 5 is devoted to describing the results. Finally, conclusions and future studies are provided.

2 Structures of PID algorithms that are most commonly implemented in PLC

Many controller manufacturers offer a variety of PID versions. It is important to remember that there is no standard terminology used among manufacturers. To reveal the controller's structure, the user should examine the mathematical expression included in the user manual rather than relying on the manufacturer's nomenclature.

PID algorithms that continuously or repetitively calculate the required position of the valve or other final actuator are called position algorithms. Conversely, algorithms that calculate the required change in position of the final actuator are called velocity or incremental algorithms. The position algorithm is the most popular algorithm in PLCs. Three basic PID structures are described below.

2.1 PID structures

The first controller structure, which is most often implemented, is called the ISA. This form is also labeled as the dependent or gain-dependent form. The controller output is calculated as presented in (2). The block diagram of the ISA PID is shown in Fig. 1, where y_{ref} is set point, y is measurement, e is control error, u is control variable.

The ISA is used in numerous controllers, e.g., GE VersaMax PLC ('PID ISA' function block), Siemens S7-300 ('FB41' function block), and Allen-Bradley PLC-5 ('PID' function block) [14, 15].

In this PID, K_{C} is dimensionless, the units of T_{I} are minutes per repeat, and the units of T_{D} are minutes. However, various manufacturers express T_{I} and T_{D} in seconds rather than minutes.

In certain cases, the proportional action can be expressed as a proportional band (PB) rather than a proportional gain, K_{C}. The integral mode can be described by the reset time, T_{I}, or reset rate, T_{R}. The units of T_{R} are repeats/minute.

The next PID structure, described by (2), is parallel, ideal parallel, independent, or gain independent. The proportional gain, K_{C}, is dimensionless; the integral gain, K_{I}, is expressed in units of time^{-1}; and the derivative gain, K_{D}, is expressed in units of time. Examples of this form are the 'PID IND' function block in the GE VersaMax PLC and 'PID' function block in the Allen-Bradley PLC-5 [16]. Figure 2 illustrates how the controller output is calculated.

Equation (3) and Fig. 3 refer to the series controller structure. This form should be considered as interacting

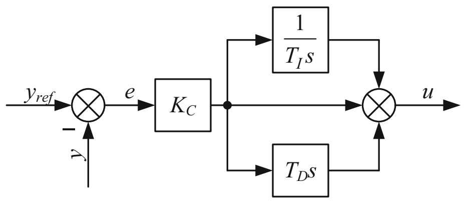

Fig. 1 Block diagram of an ISA PID controller

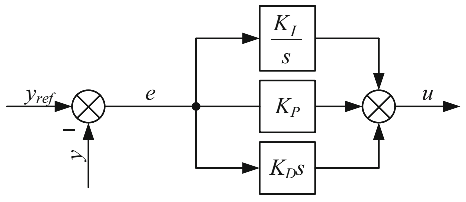

Fig. 2 Block diagram of a parallel PID controller

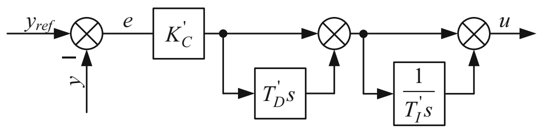

Fig. 3 Block diagram of a series PID controller

because the integral block is in series with the derivative block, and if one of the blocks change, the other block is affected. Using this nomenclature, the parallel and ISA form should be classified as non-interacting structures because the integral and derivative blocks have a parallel connection. It is worth noting that if the integral or derivative term is turned off, then the series and ISA forms are identical. The units of the controller's parameters, K'_{C}, T'_{I}, and T'_{D}, are the same as the ISA's parameters. An example of this form is the Foxbro controller [8].

2.2 Additional function and modification

In certain cases, a sudden set point change may cause a spiking reaction in the controller output. To prevent this phenomenon, numerous manufacturers propose an additional parameter that can weaken the proportional action and soften the response of the set point. Another approach to allow the controller output to be gentler is to add a derivative smoothing filter. Equation (4) describes the ISA controller with an embedded derivative filter. The coefficient N has a significant impact on the controller dynamics. There is no standard value of this coefficient among manufacturers, e.g., in the Siemens S7-300 and Toshiba T-series, the value of N is equal to 10, whereas in the Allen-Bradley PLC-5, the value of N is equal to 16 [16, 17]. Figure 4 shows the manner in which the N value can influence the controller dynamics. In the GE VersaMax PLC, the user can also enable derivative filtering by applying a first-order filter [15]. Unfortunately, manufacturers do not describe this filter in great detail.

$$u = K_C\left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D}{N}s}\right) \tag{4}$$

Furthermore, the PID structure can be modified, so that the derivative term acts only on the y signal. In most PLCs, this modification is a user option. This modification can be performed by switching the proper bit or register in the PLC's memory. This result contributes to the elimination of controller output bumps. The implementation of this option is shown in Fig. 5. Figure 6 shows a comparison of the derivative-on-measurement and derivative-on-error PID structures.

A popular addition is the deadband. This is the quantity that is compared to the error signal. If the error is within the deadband range, an update of the controller output does not occur.

Another problem strongly connected with PI and PID controllers is an integral or reset windup. This problematic situation occurs when the controller output signal remains at its maximum or minimum limit, even though the value of the error begins to decrease or increase. The integrator windup can be avoided by verifying that the integral is kept at a proper value when the controller's output saturates; thus, the controller is ready to resume action as soon as the error changes. Furthermore, there are several solutions for reset windup problems, but in practice, manufacturers of PLCs do not describe which solution they use.

3 SVR studies

Support vector machines (SVMs) are classification and regression methods, and the basis of these methods has been derived by Vapnik and Chervonenkis [18]. SVMs that address classification problems are called support vector classifications (SVCs) [19], and SVMs that address modeling and prediction are called SVRs [13]. The purpose of the SVR is to obtain a function with a maximum deviation of ε from the actual destination vectors for all given training data that is as flat as possible. SVR requires the setting of fewer user-defined parameters as well as the option of a kernel and its parameters. The advantage of SVR over conventional algorithms based on empirical risk minimization, such as an artificial neural network, is its optimization algorithm, which includes solving a linearly constrained quadratic programming function, leading to an optimal and global solution. SVR has been successfully used to solve problems in many fields, such as economics

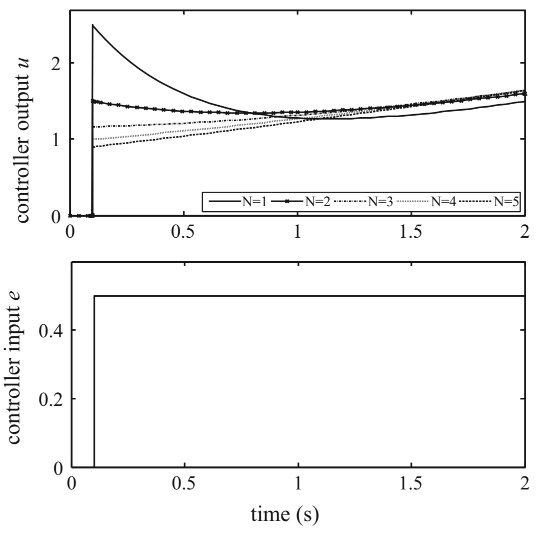

Fig. 4 Influence of the N value on controller dynamics; $K_C = 1$; $T_I = 1$; $T_D = 0.5$; ISA form

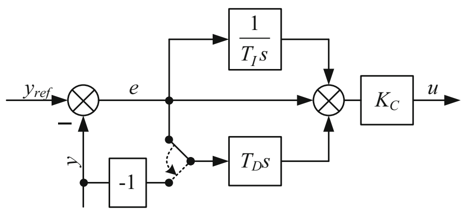

Fig. 5 ISA form with a switchable derivative action

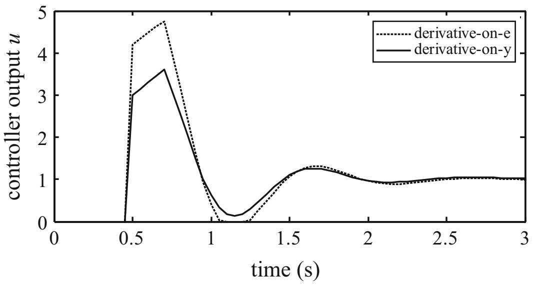

Fig. 6 Comparison of derivative on measurement and derivative on error

[20], medicine [21], electrical circuits [22], power systems [23], mechanics [24], and system identification [25, 26].

Let us assume that we have a data set of p training samples, $\{(x_1, d_1), (x_2, d_2), \ldots, (x_p, d_p)\}$, where $\mathbf{x}_i \in R^n$, $d_i \in R$. We can introduce a nonlinear mapping $\varphi(\cdot) : R^n \to H$, where H is a hypothetical feature space, and define ε—insensitive loss function—as follows:

$$L_\varepsilon = |d - y(\mathbf{x})|_\varepsilon = \max\{0, |d - y(\mathbf{x})| - \varepsilon\} \tag{5}$$

where $y(\mathbf{x})$ is the estimation of the function. The SVR formula can be expressed as follows:

$$y(\mathbf{x}) = \mathbf{w}^T \varphi(x) + b \quad \mathbf{w}, \mathbf{x} \in R^n \quad b \in R \tag{6}$$

where **w** is the weight vector and b is the offset.Then, $y(\mathbf{x})$ can be determined from the minimization problem as follows:

$$\min\ L_\varepsilon = \min \frac{1}{p} \sum_{i=1}^{p} \left(|d_i - \mathbf{w} \cdot \varphi(\mathbf{x}_i) - b| - \varepsilon \right) \tag{7}$$

By introducing slack variables ξ_i, ξ_i* into (7), an optimization problem can be formulated as follows:

$$\min_{w,b,\xi,\xi_i^*} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{p} \xi_i + C \sum_{i=1}^{p} \xi_i^* \tag{8}$$

which is subject to:

$$\begin{aligned} & d_i - \mathbf{w}^T \varphi(\mathbf{x_i}) - b \le \varepsilon + \xi_i \\ & \mathbf{w}^T \varphi(\mathbf{x_i}) + b - d_i \le \varepsilon + \xi_i^* \quad \xi_i, \xi_i^* \ge 0 \end{aligned} \tag{9}$$

The constant $C > 0$ determines the trade-off between the model flatness and the training error. The flatness in (6) indicates a small **w** value.

The solution to the optimization problem in (8) is given by the saddle point of the Lagrangian as follows:

$$\begin{aligned} J(w, \xi, \xi,^* \alpha, \alpha^*, \gamma, \gamma^*) = {} & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{p} \xi_i + C \sum_{i=1}^{p} \xi_i^* \\ & - \sum_{i=1}^{p} \alpha_i^* \left(d_i - \mathbf{w}^T \varphi(\mathbf{x_i}) - b + \varepsilon + \xi_i^* \right) \\ & - \sum_{i=1}^{p} \alpha_i \left(\mathbf{w}^T \varphi(\mathbf{x_i}) + b - d_i + \varepsilon + \xi_i \right) \\ & - \sum_{i=1}^{p} \left(\gamma_i \xi_i + \gamma_i^* \xi_i^* \right) \end{aligned} \tag{10}$$

It follows from the saddle point condition that the partial derivatives of J with respect to the primal variables ($\boldsymbol{w}$, ξ_i, ξ_i*) must be excluded for optimality. The variables α_i, α_i*, γ_i, γ_i* must satisfy the positivity constraints. The formulation of the dual problem involving the Lagrange multiplier α is equivalent to finding an expression as follows:

$$\begin{aligned} & \min_{\alpha,\alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \varepsilon \sum_{i=1}^{p} \left(\alpha_i + \alpha_i^* \right) \\ & + \sum_{i=1}^{p} d_i \left(\alpha_i - \alpha_i^* \right) \end{aligned} \tag{11}$$

which is subject to:

$$\sum_{i=1}^{p} \left(\alpha_i - \alpha_i^* \right) = 0 \quad 0 \le \alpha_i, \alpha_i^* \le C \tag{12}$$

where $Q_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$ is the kernel function in accordance with Mercer's condition [27]. The kernel function has been defined as a linear dot product of the nonlinear mapping.

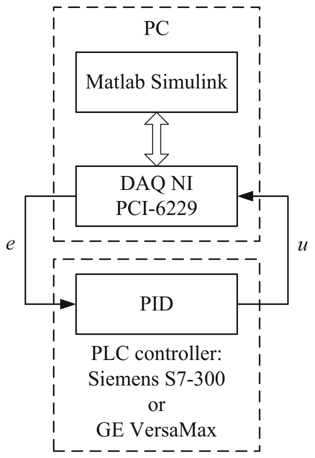

Fig. 7 Block diagram of data set collection

After solving the problem in (8), the regression function can be written as follows:

$$y(\mathbf{x}) = \sum_{j=1}^{K} (\alpha_i^* - \alpha_i) k(\mathbf{x}, \mathbf{x}_i) + b \tag{13}$$

where K is the number of so-called support vectors (SV). The vector $\mathbf{x}_i$ is associated with the coefficient α_i is called a support vector, and only those vectors have an effect on $y(\mathbf{x})$.

The selection of the coefficients ε and C is of utmost importance. The constant ε determines the margin within which the error is neglected. The smaller its value, the more support vectors will be determined by the algorithm. The constant C is the weight, which determines the trade-off between the complexity of the network, characterized by the weight vector and the error of approximation and is measured by the slack variables ($i = 1, 2, \ldots, p$).

4 Data collection

The preparation of training data and feature selection (i.e., a model selection) is usually the most important factors influencing the correct operation of the model and the ability to generalize the SVR. To generate a training and validation data set for the identification of the PID algorithm, an experiment is performed. In our study, a validation term is different than a test term and is explained in the next section.

The training and validation data sets collected by the system for our study are shown in Fig. 7.

However, there are difficulties with training the SVR on a real data set. As the number of training patterns increases, the generation of the SVR training takes significantly

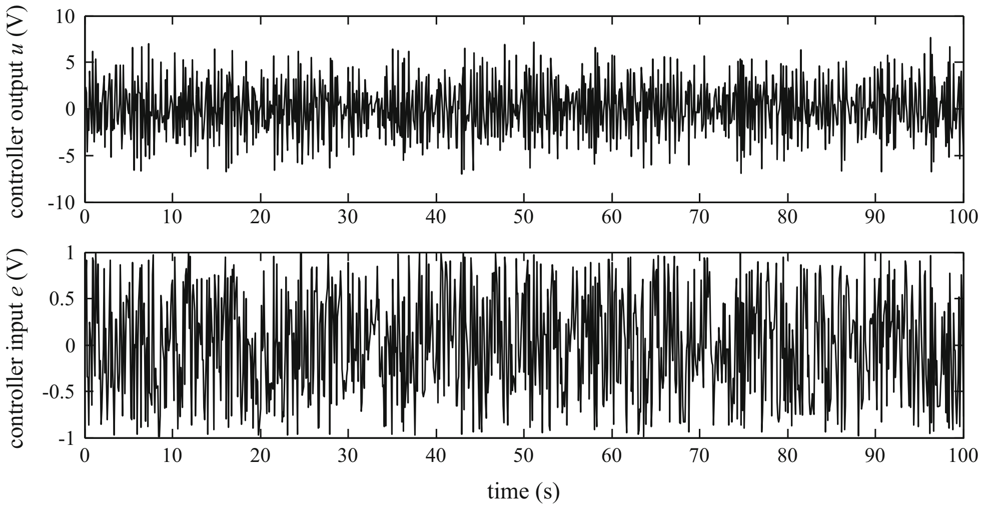

Fig. 8 Example of a signal input and output of a PID algorithm for $K_P = 0.47$, $K_I = 5.88$, $K_D = 0.08$ (PLC GE VersaMax 'PID IND' function block)

longer, with a time complexity of p^3, where p is the number of training patterns. Thus far, several algorithms, such as chunking, SMO, SVM light, and SOR, have been proposed to reduce the training time [28].

The control error, e, and control variable, u, of the PID algorithm have been administered and recorded using a PC with Matlab and Simulink 7.10.0 (R2010a) MathWorks. In the Matlab and Simulink toolbox, the real-time windows target was used. The connection between the PLC and PC was maintained through a National Instruments data acquisition board, NI PCI-6229. The sampling frequency of both the PLC and Matlab & Simulink was 10 Hz (recommended frequency by the manufacturer of the PLC).

Many attempts have shown that better SVR training results were obtained by artificial excitation signals of input rather than trying to collect data from a process that is under a closed-loop control. When the input–output patterns come from a PID under loop control, the variables are highly correlated with each other, and the information content is low. Most sampled states would reside in a narrow region around the operating point, giving minimal information on the interaction of the different input variables to produce the output [29]. Therefore, the control error, e, was administered in the form of uniformly distributed random signals of a different amplitude and frequency in the interval of -1 to 1 V. This type of artificial excitation should be selected so as to not turn on anti-windup because it will be tested at a later stage.

The example of a signal e and u collected for training and validation of the SVR is shown in Fig. 8.

As a result of performing simulations for different settings of the PID algorithm (PLC Siemens, GE) with different values of amplitude and frequency of the control error, e, administered on the input of the PID, hundreds of thousands of training samples were collected.

5 Model selection

Because the SVR has to emulate the work of the PID algorithm, the goal of this study is to construct a multiple-input and single-output (MISO) black box model for the output of the PID algorithms. The input structure of the SVR was used in accordance with the NARX (Nonlinear AutoRegressive with eXogenous inputs) model in a predictor form. To obtain a good NARX model, the selection of the amount of regressors is extremely important [30, 31]. The regressors are the inputs of the model. In our case, the output of the model depends on past inputs and outputs and can be described as follows:

$$\hat{u}(t) = f(\varphi(t), \boldsymbol{\theta}),$$
$$\varphi(t) = [e(t), e(t-1), \ldots e(t-n_e), u(t-1), \ldots u(t-n_u))] \tag{14}$$

where $\hat{u}(t)$ is the output of the model; $\boldsymbol{\varphi}(t)$ is the regression vector; $\boldsymbol{\theta}$ is the parameter vector; and n_e and n_u indicate the order of the NARX model (number of lags).

Finally, the set of SVR inputs (features input set) is extended with P, I, and D parameters, and the general form is as follows:

$$\mathbf{x} = [P\ I\ D\ e(t)\ e(t-1)\ldots e(t-n_e)\ u(t-1)\ldots u(t-n_u)] \tag{15}$$

where $P = K_C$ for the ISA; $P = K_P$ for the parallel form; $I = T_I$ for the ISA; $I = K_I$ for the parallel form; $D = T_D$ for the ISA; $D = K_D$ for the parallel form; e is the control error at time t, $t-1$, …, $t-n_e$; and u is the controller output at time $t-1$, …, $t-n_u$.

The output signal of the SVR was the control variable $\hat{u}(t)$.

The general block diagram of the full set feature vector covering both input and output is shown in Fig. 9.

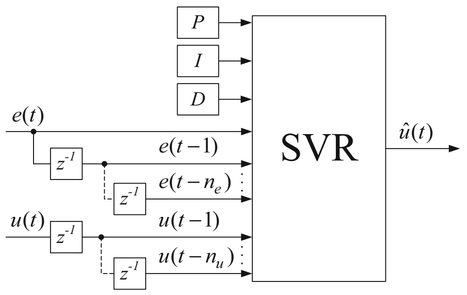

Fig. 9 General block diagram of the input vector and the output of the SVR

6 Results

The SVR was trained and tested for the PID algorithms, which were implemented in S7-300 Siemens and GE Versa Max controllers. The PID algorithms implemented in the controllers' PLC were ISA (Siemens, GE) and parallel (GE). The manufacturers provided the transfer function over a discrete time, described in (16) for the parallel form and (17) for the ISA, as follows:

$$u = K_P + K_I T_S \frac{1}{z-1} + K_D \frac{N}{1 + N T_S \frac{1}{z-1}} \tag{16}$$

$$u = K_C \left(1 + \frac{T_S}{T_I} \frac{1}{z-1} + T_D \frac{N}{1 + N T_S \frac{1}{z-1}} \right) \tag{17}$$

where T_S is the sampling time.

There is no information on the value of N or the modification of the structure. After many attempts, a value of N equal to 10 was chosen for all structures.

The results of the SVR with the neural network NARX model (NN) and transfer function (T_f) of the PID described by the manufacturers (16, 17) were compared. For NN structure, selection was made according to the procedure proposed for the SVR. Number of both inputs and outputs was the same as for the SVR. The initial number of neurons in the hidden layer was assumed as the root of the sum of inputs and outputs and then fine-tuned by trial and error method. The learning for NN was performed using Levenberg–Marquardt Method. The NN and Tf were implemented in Matlab & Simulink 7.10.0 (R2010a). More detailed description of the mechanism of the NN can be found in [32].

The data set was divided into the training set and validation set without averaging or filtering. The first 5 % of the data was used for estimation, and the last 95 % was used for validation. In the training and validation mode, a one-step-ahead prediction was performed, whereas in the test mode, the simulation (the measured inputs and estimated outputs are used to form the regressors) was performed. The differences between the validation and test mode are due to the lack of implementing the cross-validation function in the SVR toolbox.

The data set for training and validation for each of these structures was generated for dozens of different settings. The data set was normalized over the range of 0–1. The ranges for each setting were as follows: K_C: 0.5–0.9; T_I: 0.5–9.9; T_D: 0.03–0.48; K_P: 0.1–0.47; K_I: 0.1–0.91; and K_D: 0.01–0.5. A sample set for different settings is shown in Fig. 10.

Quality measures for the training, validation, and testing set are performed on the basis of the mean squared error (MSE) and fit measure (Fit) methods.

The mean squared error (MSE) is defined as follows:

$$\mathrm{MSE} = \frac{\| u(t) - \hat{u}(t) \|_2^2}{p} \tag{18}$$

The Fit measure is defined as follows:

$$\mathrm{Fit} = \left(1 - \frac{\| u(t) - \hat{u}(t) \|_2}{\| u(t) - \bar{u}(t) \|_2} \right) \times 100\ \% \tag{19}$$

where $\hat{u}(t)$ is the simulated output; $u(t)$ is the measured output; and $\bar{u}(t)$ is the mean of the measured output; p is the number of samples.

To define the optimal number of regressors in the training data set and the impact of the number of regressors on the model, a training approach of trial and error was performed. Because the order of the transfer function of the PID algorithm was low, the number of lags was determined over the range of 1–10. The minimum MSE values of the analysis of all possible combinations in relation to n_u: 1–10 and n_y: 1–10 for the three structures are plotted in Figs. 11,12, and 13.

The observation of the MSE values shows that there is no clear minimum in the above figures, which indicates the possibility that the model class is not entirely correct. Attempts to determine the minimum value or a point close to the minimum value for both the training data and validation were performed. We also tried to find a compromise between the MSE index and the size of the input vector. The effect of the selected values on the results of the test data was also observed.

Finally, based on the MSE method for the training and validation data set, optimal values of (3, 7), (3, 7), and (2, 7) were chosen, as depicted in Figs. 11, 12, and 13, respectively.

For those values, the feature vectors of the SVR were prepared as follows:

Parallel structure GE:

$$\mathbf{x} = [P\ I\ D\ e(t)\ e(t-1)\ldots e(t-3)\ u(t-1)\ldots u(t-7)] \tag{20}$$

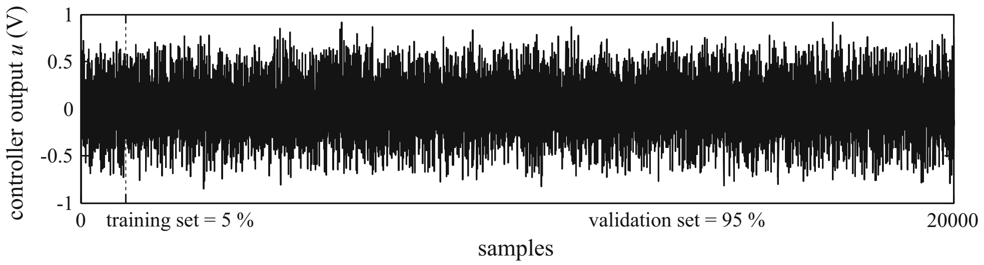

Fig. 10 Set of samples for different settings for both the training and validation mode

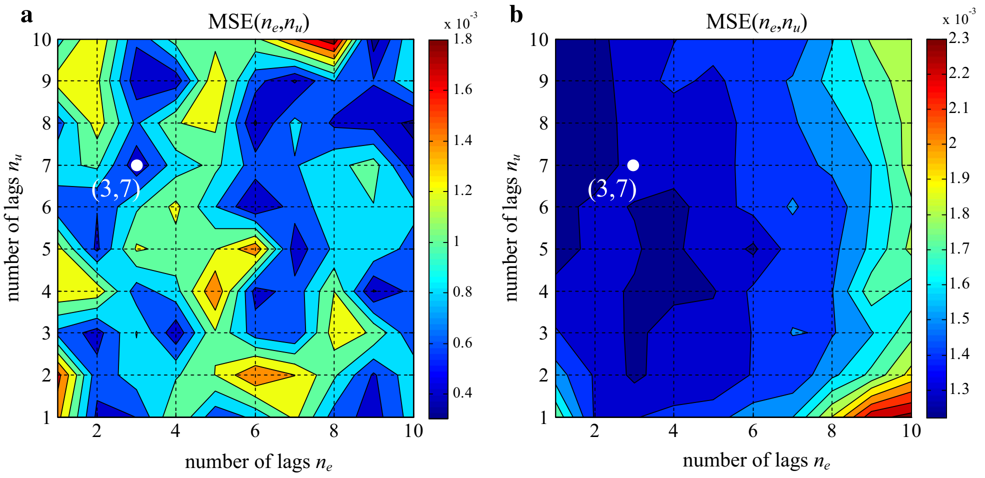

Fig. 11 MSE of the training data (**a**). MSE of the validation data (**b**); parallel form GE

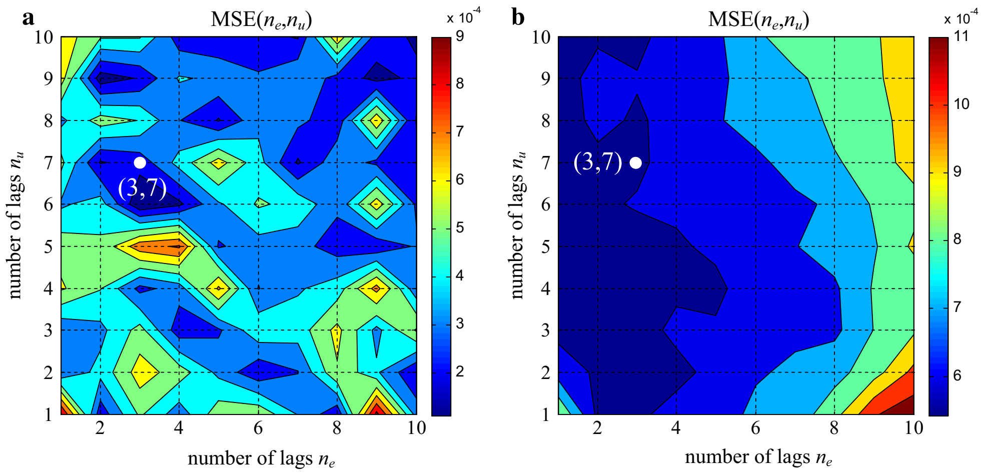

Fig. 12 MSE of the training data (**a**). MSE of the validation data (**b**); ISA form GE

ISA structure GE:

$$\mathbf{x} = [P\ I\ D\ e(t)\ e(t-1)\ldots e(t-3)\ u(t-1)\ldots u(t-7)] \tag{21}$$

ISA structure Siemens:

$$\mathbf{x} = [P\ I\ D\ e(t)\ e(t-1)\ e(t-2)\ u(t-1)\ldots u(t-7)] \tag{22}$$

After the feature vectors were selected, the training process of the SVR was performed using an SMO-type algorithm implemented in the toolbox LIBSVM [33]. We used the Gaussian radial basis function as the kernel function $k(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma\|\mathbf{x}-\mathbf{x}_i\|^2}$, where γ is the kernel parameter. The γ parameter was tuned by trial and error. The Gaussian radial basis function satisfied by the SVR kernel and is described as the relation:

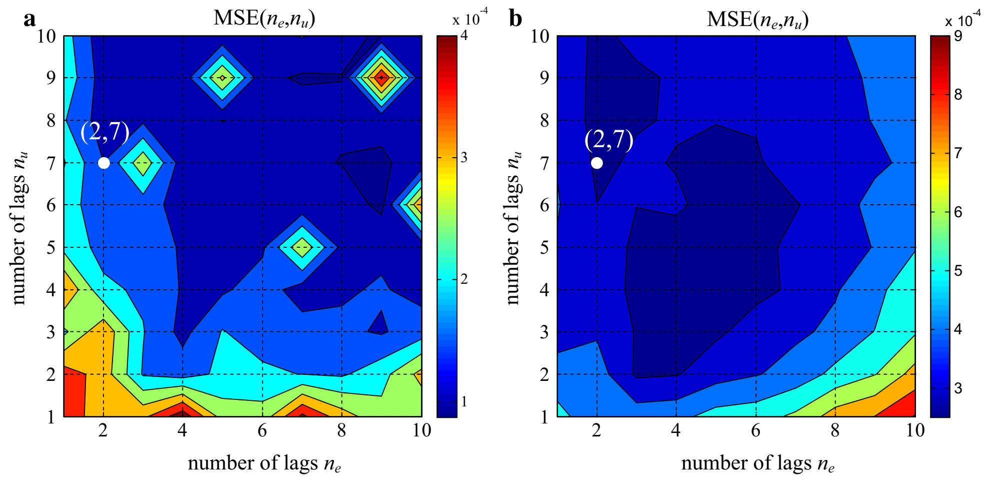

Fig. 13 MSE of training data (**a**). MSE of the validation data (**b**); ISA form Siemens

$k(\mathbf{x}, \mathbf{x}_i) = \varphi^T(\mathbf{x})\varphi(\mathbf{x}_i)$. The optimal value of the ε (i.e., insensitive loss function) parameter was determined after a series of experiments and was assumed to be $\varepsilon = 0.001$.

The test was performed under the simulation mode (i.e., estimated output is used to form the regressors). Both the sets of parameters and excitation signals of input were different from those in the training and validation mode. The procedure for testing the data collection was the same as for the training and validation mode. The number of SVR inputs (i.e., the feature vector) was the same as in the training and validation mode. The comparison of the PID algorithm's output and SVR's output was for the same sets of parameters. In both controllers, the anti-windup was turned on.

The diagram of the procedure for the test of the SVR and comparison of the accuracy Fit and MSE for any setting *P*, *I*, and *D* are shown in Fig. 14.

The signal *e* was varied over the range of −1 to 1 V. The output of the SVR and PID algorithm implemented in the PLC was compared at similar time points, and the MSE and Fit error were calculated. The test procedure was conducted in accordance with Fig. 14 and ten times repeated for different sets. The final results of the SVR, transfer function of the PID and NN test were averaged from several tests.

The examples of the SVR responses and the comparison with the response of the real output of the PID algorithm for Siemens and GE for all examined structures are shown in Figs. 15, 16, and 17.

As shown in Figs. 15, 16, and 17, the outputs of the SVR coincide with the response of the PID algorithms implemented in the PLCs. It should be noted that this option is a considerably more difficult task than the one-step-ahead prediction. In this case, there is a risk of cumulative error because the inputs and estimated output are used to form the regressors.

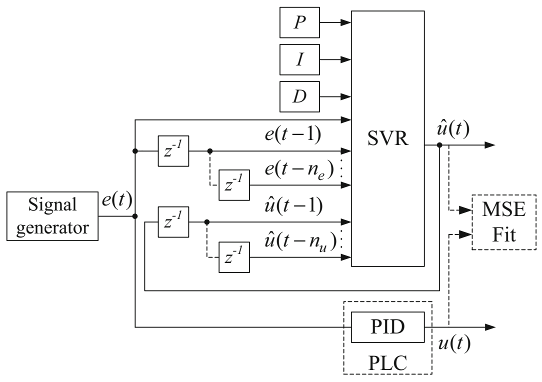

Fig. 14 Diagram of the test procedure all structures

The final results of the training and testing of the SVR combined with the results for the transfer function of PID algorithms implemented in the PLC and NNs for both Siemens and GE are shown in Tables 1, 2, and 3.

As shown in Tables 1, 2, and 3, the results of both the MSE and Fit methods for the training and testing of the SVR are better than the Tf and NNs for the PID algorithms implemented in both PLCs. The results of the SVR for the ISA (Fit—88.62, 89.67 %) are better compared to that of the parallel form (86.96 %). It can be assumed that the parallel form may be slightly modified, as confirmed by the worse result for the transfer function of the PID algorithm (83.52 %). The results for the ISA for both Siemens (89.67 %) and GE (88.62 %) are similar, most likely due to the similar implementation of the algorithms in the PLC. Generally, good results for the Tf (83.52, 88.18, 87.67 %) indicate for small modifications introduced by manufacturers. The worst results obtained for the NN (80.14, 81.07, 83.35 %) are clear because learning algorithms can stop at the local minimum. For the SVR,

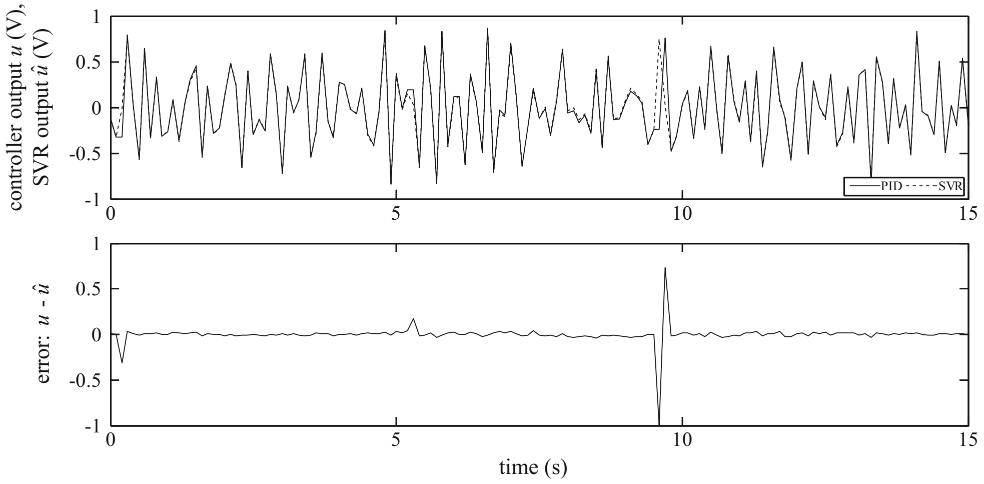

Fig. 15 Output of the PID algorithm (PLC GE VersaMax 'PID IND' function block) as well as the corresponding simulated outputs of the SVR for $K_P = 0.41$, $K_I = 0.21$, $K_D = 0.48$

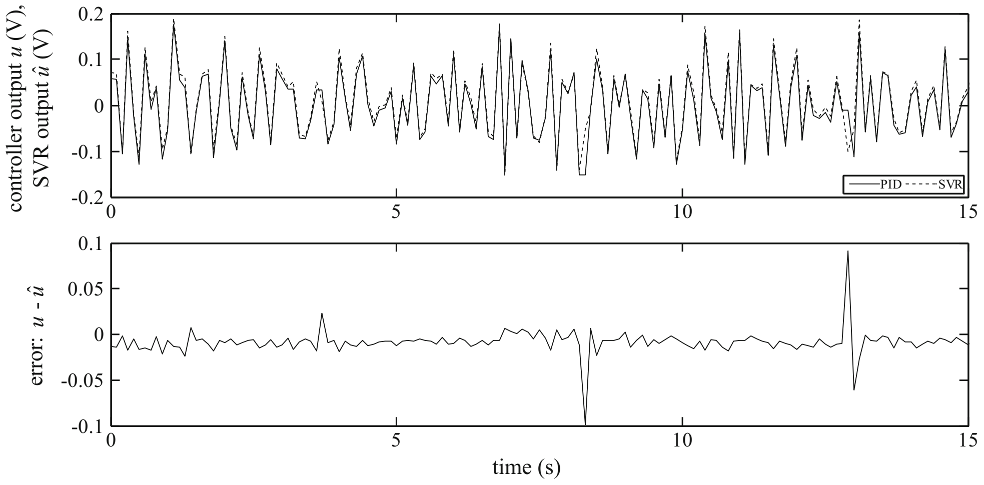

Fig. 16 Output of the PID algorithm (PLC GE VersaMax 'PID ISA' function block) as well as the corresponding simulated outputs of the SVR for $K_C = 0.66$, $T_I = 5.8$, $T_D = 0.1$

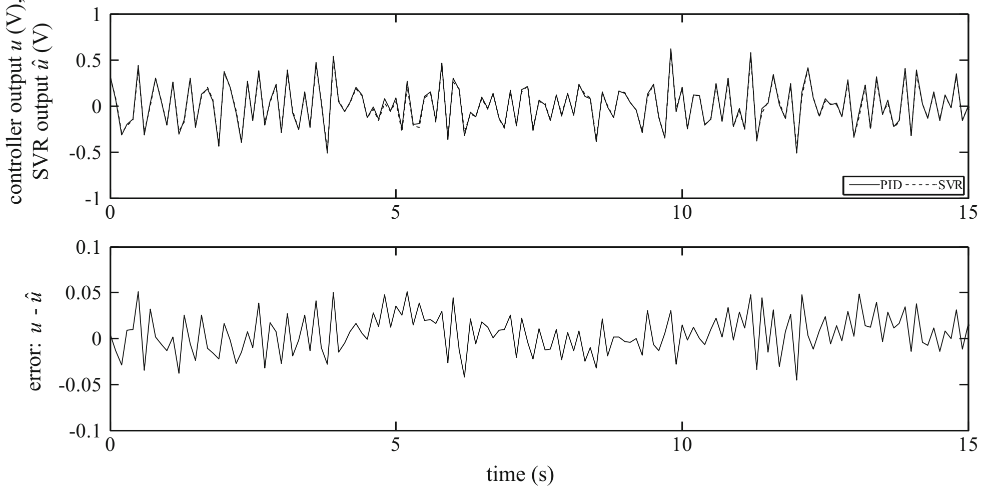

Fig. 17 Output of the PID algorithm (PLC Siemens 'FB41' function block) as well as the corresponding simulated outputs of the SVR for $K_C = 0.84$, $T_I = 6.5$, $T_D = 0.32$

the optimization process is convex and has a single optimum value. Furthermore, the minor difference between the results of the training and testing of the SVR (90.93–86.96 % for parallel form GE, 91.26–88.62 % for ISA GE, 92.00–89.67 % for ISA Siemens) demonstrates good generalization properties.

Table 1 Results for the SVR, Tf model and NNs, parallel form GE

	MSE (–)		Fit (%)	
	Training set	Test set	Training set	Test set
SVR: C = 100 $\gamma = 2.79 \times 10^{-2}$	4.6×10^{-4}	9.1×10^{-4}	90.93	86.96
Tf of PID	–	1.5×10^{-3}	–	83.52
NN[a] 14-5-1	1.2×10^{-3}	2.9×10^{-3}	84.89	80.14

[a] Where NN 14-5-1 means: 14 is the number of inputs; five is the number of hidden neurons; one is the number of outputs

Table 2 Results for the SVR, Tf model and NNs, ISA GE

	MSE (–)		Fit (%)	
	Training set	Test set	Training set	Test set
SVR: C = 150 $\gamma = 3.14 \times 10^{-2}$	2.4×10^{-4}	4.0×10^{-4}	91.26	88.62
Tf of PID	–	6.5×10^{-4}	–	88.18
NN 14-5-1	9.0×10^{-5}	1.9×10^{-3}	94.51	81.07

Table 3 Results for the SVR, Tf model and NNs, ISA Siemens

	MSE (–)		Fit (%)	
	Training set	Test set	Training set	Test set
SVR: C = 120 $\gamma = 3.04 \times 10^{-2}$	1.5×10^{-4}	2.8×10^{-4}	92.00	89.67
Tf of PID	–	4.6×10^{-4}	–	87.67
NN 13-7-1	2.8×10^{-4}	2.4×10^{-3}	90.18	83.35

7 Conclusions and future work

The goal of this study was to demonstrate a method that improves the accuracy of a PID algorithm simulation by considering the real dynamics of control algorithms. We have shown the practical advantages of the SVR and that it is better to simulate the PID algorithm using SVR than the ready transfer function of PID algorithms provided by the manufacturer. The manufacturer often provides only a general transfer function of PID algorithms and does not inform on additional functions or modifications. Furthermore, we show that the SVR maps the function of PID algorithms with the modifications introduced by the PLC manufacturer with high accuracy. In this report, only the most frequently implemented PID structures were studied. However, this approach can be extended to any structure due to the application of black box modeling, which does not require knowledge of the structure or the modifications provided by the manufacturer. With this approach, the SVR simulation results can be used to tune the PID algorithms in the PLC. In additional studies, we intend to focus on emulating additional functions, e.g., anti-windup, control zone, deadband, slew time, and weakening proportional action as well as their combinations. Furthermore, we intend to use ICA in connection with SVR, which will enable a quick and accurate tuning of different PID algorithms used in real PLCs from any manufacturer, and to compare the results with auto-tuning implemented in the PLC.

Acknowledgments This work was financially supported by a grant from the National Science Centre DEC—2012/05/B/NZ7/02705.

References

1. Li Y, Ang KH, Chong GCY (2006) Patents, software, and hardware for PID control. an overview and analysis of the current art. IEEE Control Syst Mag 26(1):42–54
2. Astrom KJ, Hagglund T (1995) PID controllers: theory, design, and tuning. Instrument Society of America, Research Triangle Park
3. Rhinehart RR, Shinskey FG, Wade HL (2006) Control modes—PID variations. In: Liptak BG (ed) Process control and optimization, 4th edn. Taylor and Francis, Boca Raton, pp 124–130
4. Astrom KJ, Murray RM (2008) Feedback systems an introduction for scientists and engineers. Princeton, Princeton
5. Bryan LA, Bryan EA (1997) Programmable controllers: theory and implementation. Industrial Text Company, Atlanta
6. Denisenko VV (2010) Modifications of PID regulators. Autom Remote Control 71(7):1465–1475
7. Tan KK, Lee TH, Jiang X (2001) On-line identification, assessment and tuning of PID controller. J Process Control 11(5):483–496
8. O'Dwyer A (2009) Handbook of PI and PID controller tuning rules. Imperial College Press, London
9. Tan W, Liu J, Chen T, Marquez HJ (2006) Comparison of some well-known PID tuning formulas. Comput Chem Eng 30(9):1416–1423
10. Shinskey FG (1990) How good are our controllers in absolute performance and robustness. Meas Control 23:114–121
11. Salat R, Awtoniuk M (2012) Dynamics mapping of the work of the PID controller implemented in the PLC by means of least squares-support vector machines. Przeglad Elektrotechniczny 88(5a):227–230
12. Sjöberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec P-Y, Hjalmarsson H, Juditsky A (1995) Non-linear black-box modeling in system identification: a unified overview. Automatica 31(12):1691–1724
13. Smola AB, Scholkopf B (1998) A tutorial on support vector regression. Neurocolt technical, rep. NV2-TR-1998-030, London

14. Pfeiffer B-M (2000) Towards 'plug and control': self-tuning temperature controller for PLC. Int J Adapt Control Signal Process 14(5):519–532
15. http://support.ge-ip.com/support/resources/sites/GE_FANUC_SUPPORT/content/live/DOCUMENT/0/DO54/en_US/gfk1503D.pdf
16. http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1785-rm001_-en-p.pdf
17. http://xtronics.com/toshiba/pid.pdf
18. Vapnik V, Chervonenkis A (1974) Theory of pattern recognition. Nauka, Mosco (in Russian)
19. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
20. Kim D, Lee H, Cho S (2008) Response modeling with support vector regression. Expert Syst Appl 34(2):1102–1108
21. Salat R, Salat K (2013) The application of support vector regression for prediction of the antiallodynic effect of drug combinations in the mouse model of streptozocin-induced diabetic neuropathy. Comput Methods Programs Biomed 111(2):330–337
22. Salat R, Osowski S (2011) Support vector machine for soft fault location in electrical circuits. J Intell Fuzzy Syst 2(1):21–31
23. Salat R, Osowski S (2004) Accurate fault location in the power transmission line using support vector machine approach. IEEE Trans Power Syst 19(2):979–986
24. Tijani IB, Rini Akmeliawati (2012) Support vector regression based friction modeling and compensation in motion control system. Eng Appl Artif Intell 25(5):1043–1052
25. Chevalier RF, Hoogenboom G, McClendon RW, Paz JA (2011) Support vector regression with reduced training sets for air temperature prediction: a comparison with artificial neural networks. Neural Comput Appl 20(1):151–159
26. Huyck B, De Brabanter K, Logist F, De Brabanter J, Van Impe J, De Moor B (2011) Identification of a pilot scale distillation column: a kernel based approach. In: 18th IFAC world congress (IFAC2011), Milano, Italy
27. Vapnik V (1998) Statistical learning theory. Wiley, New York
28. Platt JC (1998) Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, Burges CJC, Smola AJ (eds) Advanced in kernel methods—support vector machines. MIT Press, Cambridge, pp 185–208
29. Plutowski M (1994) Selecting training exemplars for neural networks learning. Dissertation, University of California
30. Lind I, Ljung L (2008) Regressor and structure selection in NARX models using a structured ANOVA approach. Automatica 44(2):383–395
31. Bonin M, Seghezza V, Piroddi L (2012) NARX model selection based on simulation error minimisation and LASSO. IET Control Theory Appl 4(7):1157–1168
32. Masri SF, Chassiakos AG, Caughey TK (1992) Structure-unknown non-linear dynamic systems: identification through neural networks. Smart Mater Struct 1(1):45–56
33. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27