Multi-Class SVMs for Automatic Performance Classification of Closed Loop Controllers

N. Pillay*, P. Govender**

Energy and Optimization Unit Laboratory,
Department of Electronic Engineering, Durban Univeristy of Technology,
Durban, Kwa Zulu Natal,
*Republic of South Africa,(email:*trevorpi@dut.ac.za, **poobieg@dut.ac.za)*

Abstract: The aim of this paper is to present a novel framework using Multi-Class Support Vector Machines (MC-SVMs) to classify the performance of closed loop single-input-single-output feedback controllers. A SVM is trained to recognize descriptive statistical patterns originating from an Autocorrelation Function (ACF) of process data vectors. ACF patterns emanating from different closed loop behaviors are used in the feature extraction procedure. Simulation study and application to real world industrial data sets show that the MC-SVM classification tool is capable of detecting problematic control loops with very good accuracy and efficiency.

Keywords: controller performance assessment, support vector machines, autocorrelation function, feature extraction, PI controllers.

1. INTRODUCTION

Automatic process control monitoring constitutes an integral part of performance improvement in modern industry. With the sheer number of process control loops functioning in a typical industrial facility, operators and maintenance personnel are usually inundated to manually detect and diagnose poorly performing control loops individually. Furthermore, many problematic control loops may not be easily detected from simple cursory inspection of recorded historian trends of plant data (Rengaswamy et al., 2001) and thus require continuous automatic evaluation. Such monitoring efforts are imperative to minimize product variability, improve production rates and reduce wastage. Various studies (Ender, 1993; Rinehart, 1997) conducted on process control loop performance indicate that as many as 60% of control loops often suffer from some kind of performance problem. Some of the major contributors to unsatisfactory loop performance are inappropriate controller tuning, presence of external or internal disturbances and hardware problems such as valve nonlinearities and sensor failure (Howard and Cooper, 2010). It is therefore an important task to detect unsatisfactory control loop behavior and suggest remedial action. Such a monitoring system must be integrated into the control system life span as plant changes and hardware issues become apparent.

It is well known in literature (Karra and Karim, 2009; Howard and Cooper, 2010; Jelali, 2013) that the autocorrelation function (ACF) of the process variable (PV) time series reveals important characteristics of closed loop behavior and is a valuable tool for first line controller performance assessment (CPA). However autonomous detection methods taking advantage of ACF properties have been limited. In the methodology presented by (Howard and Cooper, 2010), a second order continuous time model was fitted to the ACF curve. The damping coefficient of the second order model was then used to define a relative damping index (RDI) which provides a measure of loop performance. Performance measure falls into the categories of “sluggish” or “aggressive” closed loop behavior. Although the method is simple to understand and easy to implement it does depend heavily on the accuracy of the ACF curve fit.

In this work, a novel approach to data driven CPA framework using multi-class support vector machines (MC-SVMs) is presented. Emphasis is placed on proportional-integral (PI) controllers operating in a negative feedback scheme (see Fig.1) since they are common, robust and flexible enough to be applied to many control applications encountered in petrochemical, refining and chemical processing facilities. The proposed SVM framework will extend and complement existing CPA benchmark indices as it is developed for regulatory and set point tracking systems. Detection of poor performance caused by control valve nonlinearity is also investigated.

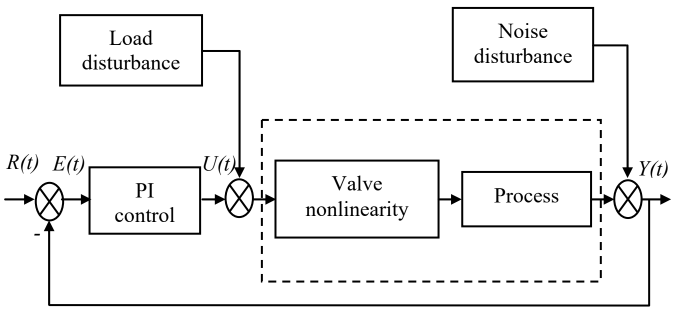

Fig. 1. Closed loop feedback system under consideration with valve nonlinearity and external disturbances.

Unlike the use of model based CPA methods, the proposed framework relies only on the salient statistical features of the input data ACF signal. Since MC-SVMs have not been used for CPA, their performance in this application is still an open issue.

The rest of the paper is organized as follows: Section 2 provides a brief review of MC-SVMs and introduces its use for CPA. Fifteen key ACF statistical attributes that distinguish different controller performance are presented. Feature extraction algorithms used in the development of the MC-SVM CPA tool are introduced in Section 3. Section 4 addresses application of the proposed classification monitoring tool. Several case studies are provided in order to validate and demonstrate the efficacy of the proposed framework in simulation and on real industrial data sets. Section 5 summarizes the work with a description of outstanding issues and limitations of the proposed methodology.

2. BACKGROUND ON MC-SVMs

Support vector machines are a powerful statistical learning theory approach to classification problems. This has been demonstrated in successful application of the algorithm in areas of image recognition, text detection and speech verification (Chapelle et al., 1999; Yang et al., 2005; Widodo and Yang, 2007; Kampouraki et al., 2009). Motivation for its use in this work is its superior accuracy and generalization capabilities in comparison to artificial neural networks, especially when a smaller number of samples are available in practice (Cortes and Vapnik, 1995). A review of SVMs in the research field of machine condition monitoring and fault diagnosis is given by (Widodo and Yang, 2007). Typically SVMs are used to recognise special patterns from an acquired signal which are classified according to specific fault occurrence in the machine. Following signal acquisition, statistical features are extracted from the data for the purpose of determining defining features of a specific fault. Such features are then considered suitable training patterns for recognition.

Consider a set of N training samples $\{(\mathbf{x}_i, y_i)\}_{i=1,2,\ldots,N}$, with $\mathbf{x_i} \in \Re$ representing the input vectors and $y_i \in \{-1,+1\}$ denoting the class labels for the binary classification problem. A feature space shown in Fig. 2 consists of data points belonging to two different classes.

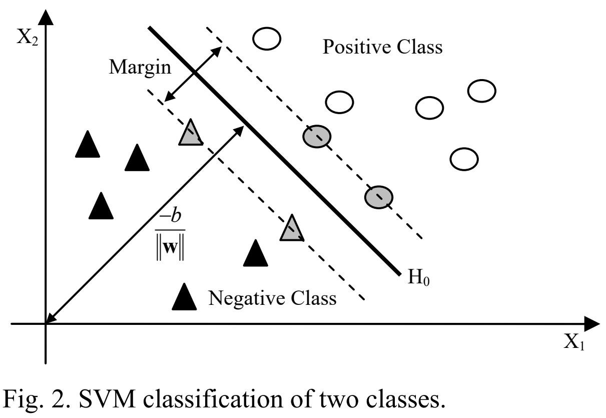

Fig. 2. SVM classification of two classes.

Black triangles correspond to the “Negative Class” and white circles represent the “Positive Class”. The primary objective of the SVM algorithm is to orientate a separating hyperplane (H_0) between the two distinct classes such that the "Margin" (m) between the dotted lines is maximised. The optimal separating hyperplane is positioned at the centre of the margin. Bordering sample points close to the separating hyperplane which define the margin are called support vectors as shown by the grey triangles and circles. Once the support vectors have been selected, the remaining data points in the feature space become redundant since the classification decision process is solely based on the information provided by the particular support vectors.

The points $\mathbf{x}$ which lie on the separating hyperplane satisfy the following condition:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{1}$$

where, $\mathbf{w}$ is a normal vector to the separating hyperplane and b denotes the bias term used to define the position of the separating hyperplane. $|b| / \|\mathbf{w}\|$ is the perpendicular distance from the origin to the separating hyperplane and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. For linearly separable data, the margin is defined by the following constraints:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \quad \text{for } y_i = +1 \tag{2}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{for } y_i = -1 \tag{3}$$

equivalent to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0\,, \qquad i = 1, \ldots, N \tag{4}$$

The binary SVM classifier is written as:

$$f(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x}_i + b) \tag{5}$$

The decision function is based on the sign of $f(\mathbf{x})$ to classify input data either as +1 or -1. Given that there are many possibilities of separating hyperplanes in the feature space, the SVM classifier locates the hyperplane that best maximizes the separating margins between the two classes. With regards to Fig. 2, $m = \frac{2}{\|\mathbf{w}\|}$ represents the maximum distance between the hyperplanes. We subject m to Eq. (2) and (3) and replace it with its equivalent minimisation of the cost function:

$$J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{6}$$

The optimisation problem given in Eq. (6) can be solved using quadratic programming but instead reformulated into its primal Lagrange (L_P) multiplier equivalent. This allows for efficient handling of constraints imposed by Eq. (4) and the training data only appears in the form of dot products between vectors. Conversion of Eq. (6) into its equivalent Lagrangian primal form with Karush-Kuhn-Tucker (KKT) conditions imposed yields:

$$L_P(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right] \tag{7}$$

where, $\alpha = (\alpha_1, ..., \alpha_i)$ represents non-negative Lagrange multipliers associated with constraints of Eq. (4).

Given Eq. (7), we find the derivative of L_P with respect to **w** and b, and simultaneously require that the derivatives of L_P with respect to all α vanish. For

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{8}$$

and

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{9}$$

we substitute Eq. (8) and (9) into Eq. (7) to get the dual Lagrangian (L_D) formulation:

$$L_D(\mathbf{w}, b, \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{10}$$

Given Eq. (10), the task is to solve for the Lagrangian multiplier (α) that maximizes the function:

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{11}$$

subject to, $\alpha_i \geq 0, \quad i = 1, \ldots, N, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$

Solving the dual optimization problem yields the Lagrangian multipliers necessary to express **w** in Eq. (5). This leads to the decision function given by:

$$f(\mathbf{x}) = sign\left(\sum_{i,j=1}^{N} y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_j) + b \right) \tag{12}$$

However in most real world applications, the sampled data may contain overlapping points which makes linear separation unattainable. Therefore a restricted number of misclassifications should be allowed around the margin. In this case, a set of slack variables; $\beta_i \geq 0, i = 1, ..., N$ are introduced. This represents the distance by which the linearity constraint is violated and is given by:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \beta_i, \; \beta_i \geq 0, \quad i = 1, \ldots, N \tag{13}$$

Hence, the modified cost function of Eq. (6) which accounts for the extent of the constraint violations becomes:

$$J(\mathbf{w}, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^{N} \beta_i \tag{14}$$

where 'c' represents a user defined positive regularization constant. In this instance $c = 1$ yielded the best cross validation results during trials using new data. The constant controls the stringency of the constrained violations and therefore defines the trade-off between a large margin and misclassification error. As before, the dual Lagrangian multipliers need to be formulated and solved in order to articulate the decision function.

If a linear boundary is unable to separate the two classes effectively, then the input data is mapped into a high-dimensional feature space through nonlinear mapping. Within this high-dimensional feature space, a separating hyperplane is constructed that linearly separates the class groups. This is achieved using nonlinear kernel vector functions, $\mathbf{\Phi(x)} = (\phi_1(\mathbf{x}), \ldots, \phi_q(\mathbf{x}))$ to map the m-dimensional input vector $\mathbf{x}$ onto the q-dimension feature space. In this work, linear, polynomial (Poly) and radial basis function (RBF) kernel functions are tested in the proposed CPA methodology and represented as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left[\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) \right] \tag{15}$$

Substituting Eq. (15) into Eq. (12), the SVM decision function becomes:

$$f(\mathbf{x}) = sign\left(\sum_{i,j=1}^{N} y_i \alpha_i \left[\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) \right] + b \right) \tag{16}$$

Although SVMs were originally developed for binary classification problems, it can be easily extended to multi-class problems. The basic approach is to construct and combine several binary classifiers. One-against-one (OAO) applies pair-wise classification between the different classes while one-against-all (OAA) compares a given class with all the other classes grouped together (Widodo and Yang, 2007). OAO method constructs *k(k-1)/2* classifiers, where each one is trained on data from two classes. For instance, if *k=5* then 10 binary SVM classifiers need to be constructed rather than *5* as in the case of the OAA approach. Although this requires a larger training time, the individual problems that need to be trained are significantly smaller. For this reason, OAO is utilized in this work. Formulation of the OAO SVM classifier requires training data from the *ith* and *jth* classes. $y_i \in \{1, ..., k\}$, where *k* denotes the number of classes and is *5* for this study. The following modification results in minimization of the binary classification given by:

$$J(\mathbf{w}, \beta) = \frac{1}{2} \left\| \mathbf{w}^{ij} \right\|^2 + c \sum_t \beta_t^{ij} \tag{17}$$

subject to the constraints

$$(\mathbf{w}^{ij})^T \phi(\mathbf{x}_t) + b^{ij} \geq 1 - \beta_t^{ij} \qquad \text{for} \qquad y_t = i,$$

$$(\mathbf{w}^{ij})^T \phi(\mathbf{x}_t) + b^{ij} \leq -1 + \beta_t^{ij} \qquad \text{for} \qquad y_t = j,$$

and

$\beta_t^{ij} \geq 0, \;$ for $j = 1, 2, \ldots, q.$

The training data $\mathbf{x}_t$ is mapped to a higher dimensional space by kernel function ϕ. The binary classification problem is therefore modified to include combinations of SVMs from different classes (OAO) as illustrated in Fig.3. If sign

$((\mathbf{w}^{ij})^T \phi(\mathbf{x}) + b^{ij})$ decision gives $\mathbf{x}$ in the ith class, then the vote for the ith class is incremented by one, otherwise the jth class is increased by one. New input vectors are predicted to belong to a certain class using the largest vote as the selection criteria. The next section proposes feature extraction for developing the MC-SVM for the CPA problem.

3. PROPOSED ACF FEATURE EXTRACTION AND AUTOMATED MC-SVM CPA

At first, ACF results for different closed loop performance commonly encountered in most processes are presented. Then, simple statistical features describing the different classes are used for feature extraction. Finally, these distinguishing features are used to train a MC-SVM classifier for controller performance detection.

3.1 ACF Analysis

It is well known that ACF pattern of the process output (Y) is a convenient way of determining a systems closed loop performance and is often used as a preliminary check (Howard and Cooper, 2010; Jelali, 2013).

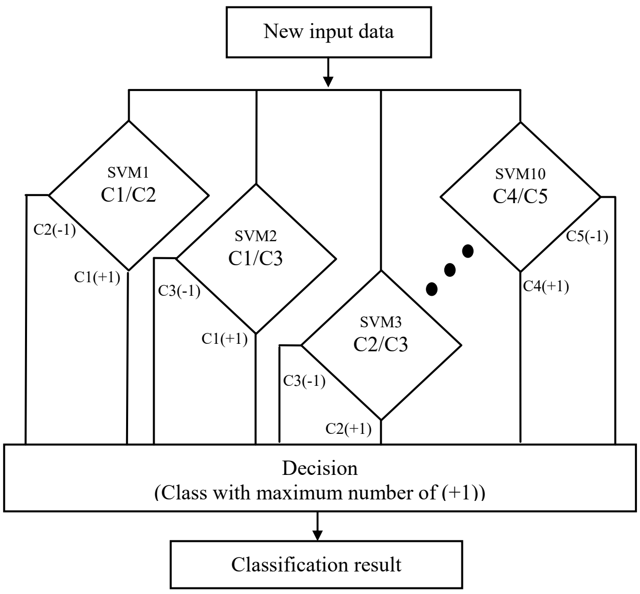

Fig. 3. MC-SVM based "one-against-one" strategy ('C' denotes the class).

For instance, oscillatory signals possess certain unique mathematical properties that can be characterised by its sample ACF (Karra and Karim, 2009). Further, Howard and Cooper (2010) showed that the disturbance rejection characteristic of a control system output is related to the ACF coefficients.

ACF lag ($\rho(l)$) coefficients are defined as:

$$\rho(l) = \frac{\sum_{t=1}^{n-l} (Y(t)-\bar{Y})(Y(t+l)-\bar{Y})}{\sum_{t=1}^{n} (Y(t)-\bar{Y})^2} \tag{18}$$

where $Y(t)$ is the measured value of the signal at time t and $\bar{Y}$ represents its mean; n *is* the number of samples. For control systems designed for setpoint (SP) tracking, $Y(t)$ is replaced with the error signal $E(t)=R(t)-Y(t)$ in Eq. (18). Consider a first-order-plus-dead-time (FOPDT) process model given by:

$$G_p(s) = \frac{K_p \exp^{-\theta_p s}}{T_p s + 1} \tag{19}$$

where, Kp and Tp represent the process gain and time constant respectively. θ_p is the plant dead time and s is the Laplace operator. Using the process model in Eq. (19), numerous simulations were conducted under different operating conditions with a PI controller. Varying conditions such as valve static friction nonlinearity (Choudhury *et al.*, 2005), stochastic sensor noise and load disturbance behaviours were used to replicate real world conditions. Within each subset of transfer functions, the values of Kp , Tp and θ_p were varied to simulate nonlinear time varying effects.

Fifteen key ACF patterns presented in Fig. 4 and Table 1 highlight the findings of the simulation study. The observed ACF features fall into five distinct classes with labels indicating the proposed controller performance classification criteria.

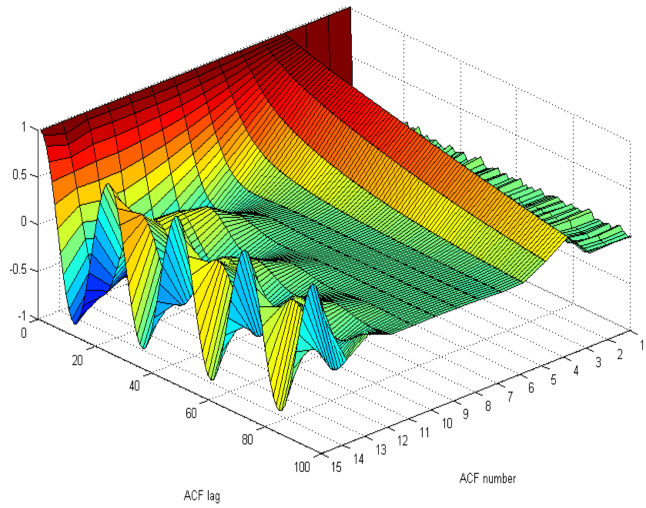

Fig. 4. Fifteen ACF signatures of different closed loop systems.

Table 1. ACF number with corresponding class and label.

ACF number	Class	Label
1-3	C1	Noise
4-6	C2	Sluggish
7-9	C3	Acceptable
10-12	C4	Aggressive
13-15	C5	Oscillatory

These classes were selected based on the following occurring system behaviour; high signal to noise ratio (C1), detuned systems which act slowly (C2), satisfactory performance with fast settling time and minimal overshoot (C3), high gain systems with large overshoots and longer settling time (C4), and systems which exhibit steady oscillatory behaviour (C5).

3.2 ACF Statistical Features

Simple box plots (see Fig. 5) of the ACF patterns (ACF1-ACF15) for each class (C1-C5) reveal several important characteristics. The distinguishing features of the ACF lag coefficients (ρ) are described below.

1) Mean,

$$\bar{\rho} = \frac{1}{l}\sum_{i=1}^{l}\rho(i) \tag{20}$$

$\bar{\rho}$ is the mean value of the ACF coefficients for each subset class. In this work, the number of lags is determined by number of samples (*l=n/5)*. The data sampling rate (T_s) is therefore an important consideration, whereby adequate sample time must provide a suitable length of data necessary to represent the whole picture of process activity. Howard and Cooper (2008) recommend sampling data at ten times the overall time constant of the process.

2) Variance,

$$\mathrm{var}(\rho) = \frac{1}{l}\sum_{i=1}^{l}(\rho(i)-\bar{\rho})^2 \tag{21}$$

provides a measure of dispersion of the set of ACF ρ-values for each class. Small variance indicates that the ACF lag coefficients are close to the mean, whereas a larger value will indicate data is spread out away from the mean.

3) Interquartile range,

$$iqr(\rho) = Q_3 - Q_1 \tag{22}$$

is the relative statistical probability distribution of the capabilities of the classification tool. ACF ρ-values between the upper 75% (Q_3) and lower 25% (Q_1) quartiles.

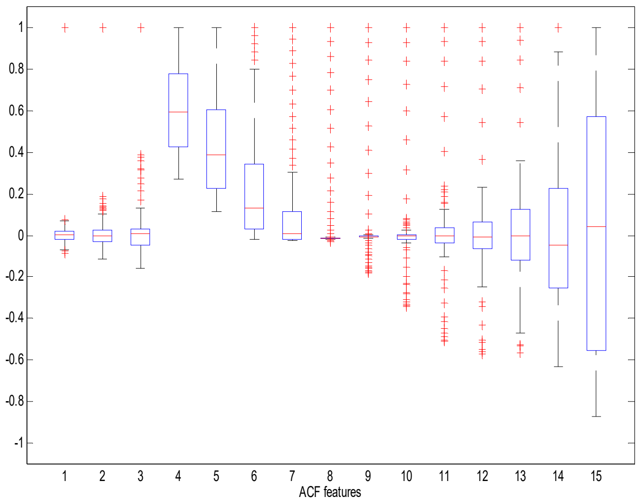

Fig. 5. Box plot representation for each ACF signature.

4) Skewness,

$$\gamma(\rho) = \frac{\frac{1}{l}\sum_{i=1}^{l}(\rho(i)-\bar{\rho})^3}{\left(\sqrt{\frac{1}{l}\sum_{i=1}^{l}(\rho(i)-\bar{\rho})^2}\right)^3} \tag{23}$$

describes the measure of symmetry of the probability distribution of ACF ρ-values. Normally distributed data that are symmetrical around its mean will have $\gamma(\rho)=0$.

By considering the descriptive statistical analysis on the ACF coefficients (ρ), a summary of the feature extraction for each ACF signature is presented in Fig.6.

3.3 Automated MC-SVM CPA Diagnostic Procedure

Given the template results illustrated in Fig 6, a MC-SVM is trained to recognise the statistical patterns. Three sets of data were chosen for each class to improve the generalisation capabilities of the classification tool. New input data can be categorised into one of the classes belonging to C1 to C5.

The MC-SVM CPA is summarised as follows:

Step 1: Collect plant data. *Y(t)* for disturbance rejection systems or *E(t)* for setpoint tracking. Ensure adequate data sampling and sufficient process activity.
Step 2: Compute ACF coefficients (ρ) using Eq.(18).
Step 3: Extract statistical features of ρ using Eq. (20-23).
Step 4: Apply new input features to the trained MC-SVM classification algorithm, Eq.(17).
Step 5: Present classification result (C1 to C5).
Step 6: Go to step 1 when new data set is available.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Preliminaries to the Experiments

The following experiments were conducted on simulated and real world data sets to test the accuracy and efficiency of the proposed CPA tool.

All simulation work was conducted in MATLAB™ SIMULINK™ using an Intel core i5 CPU running at 2.5 GHZ with 4 GB of RAM. A graphical user interface (GUI) was created using MATLAB™ *guide* command to handle multiple data sets from different loops and is illustrated in Fig.7. All data sets that were generated in simulation environment or captured online from the plant distributed control system (DCS) were stored on the ancillary computer's hardrive. Captured data sets were then analysed using the developed GUI tool to automatically determine the state of the controller performance for each loop.

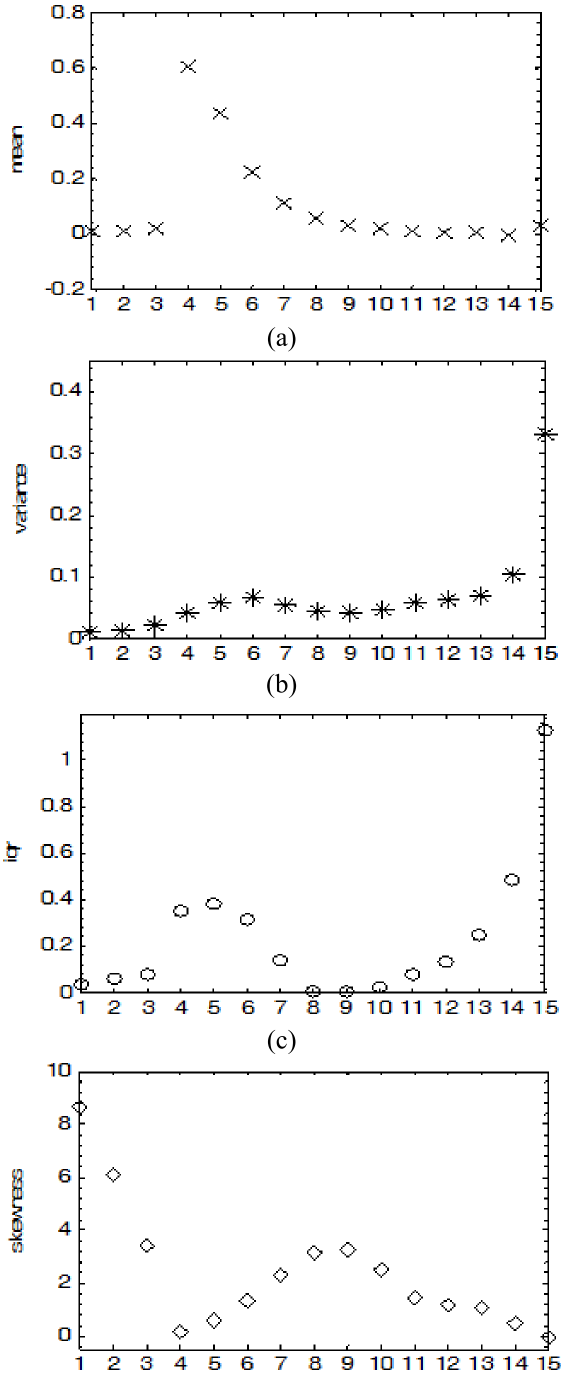

Fig 6. Features extracted from ρ-values of the data sets. (a)-mean, (b)-variance, (c)-interquartile range, (d)-skewness.

4.2 MC-SVM Kernel Selection

Performance of the MC-SVM is reliant on the choice of kernel function used to transfer input data to a higher dimensional space. The choice is data dependant and currently there are no well established guidelines to achieve satisfactory performance (Yang et al., 2005).

Table 2 highlights the results of different kernels used in the simulation study using generated data from the FOPDT model given by Eq. (19). d is the degree of the polynomial function (Poly) and the width of the Radial Basis Function (RBF) kernel is given by g.

Percentage misclassification is the main criteria for evaluating the performance of the SVM. Preliminary test results showed this criteria ranged from 4.8% to 32.1% with RBF ($g = 3$) outperforming linear and polynomial kernel types. Training time was fast and relatively similar for all kernel functions evaluated.

Table 2. Kernel selection test results using 875 simulated process data.

Kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$	Kernel type	No. of support vectors	Training time (ms)	% Mis-classified
$\mathbf{x}_i \cdot \mathbf{x}_j$	Linear	14	0.78	16.8 %
$(\mathbf{x}_i \cdot \mathbf{x}_j)^d$	Poly 1 (d=1)	14	0.48	15.2 %
	Poly 2 (d=2)	13	0.46	21.6 %
	Poly 3 (d=3)	13	0.46	32.1 %
$\exp\left\{\left(-\frac{1}{g}\right)\lVert\mathbf{x}_i - \mathbf{x}_j\rVert^2\right\}$	RBF (g=1)	15	0.49	16.3 %
	RBF (g=2)	15	0.51	10.4 %
	RBF (g=3)	15	0.49	4.8 %

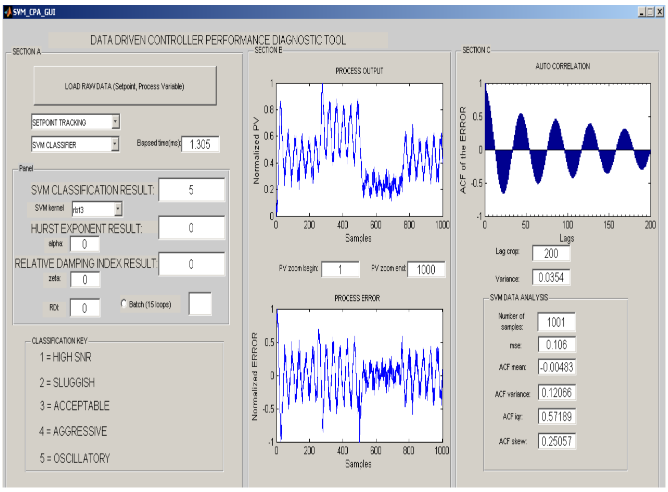

Fig. 7. Data driven CPA diagnostic GUI used in the experiments.

4.3 Simulation Case Study

Given the satisfactory performance of the MC-SVM classifier with RBF ($g = 3$) kernel mapping function, the methodology was applied to typical process models found in industry (Spinner et al., 2014). Table 3 shows the plant and corresponding disturbance models used in the experiments with varying degrees of control difficulty as indicated by the controllability ratio $\left(0 \le \frac{\theta_p}{T_p} \le 10\right)$.

Effects of control valve static friction were also considered with a stick band of $s_v = 5$ and jump band in the range of

$0 \le j_v \le 20$. Disturbance models were driven with a zero mean white noise sequence of variance $\sigma = 1\times10^{-3}$.

The proposed method was compared to the Relative Damping Index (RDI) (Howard and Cooper, 2010) and Hurst Exponent (HE) (Srinivasan et al., 2012) controller performance benchmarks and presented in Table 4.

Table 3. Simulated process and disturbance models used in the study.

Case study	Plant model	Disturbance model
1	$G_p(s) = \dfrac{1\exp^{-5s}}{(10s+1)}$	$D(s) = \dfrac{1}{(10s+1)}$
2	$G_p(s) = \dfrac{1}{(s+1)^4}$	$D(s) = \dfrac{1}{(s+1)^4}$
3	$G_p(s) = \dfrac{1\exp^{-3s}}{(15s+1)(5s+1)(2s+1)}$	$D(s) = \dfrac{1}{(17s+1)(4s+1)(s+1)}$
4	$G_p(s) = \dfrac{1\exp^{-5s}}{(s+1)^3}$	$D(s) = \dfrac{1}{(s+1)^3}$
5	$G_p(s) = \dfrac{1\exp^{-10s}}{(s+1)(s+2)(s+3)}$	$D(s) = \dfrac{1}{(s+1)(s+2)(s+3)}$

Table 4. Results of 65 simulated experiments.

	MC-SVM	HE	RDI
Average test time (ms)	1.3	823	609
% Misclassified	3.1 %	30.4 %	12.3 %

Results indicate an improved performance over the set of simulated data with the proposed classification scheme. The HE method failed to detect loop excessive oscillations which were caused by valve static fiction. Average computational time for the MC-SVM method is very fast in comparison to the other methods. This is due to the efficiency of the MC-SVM algorithm even with 15 support vectors defining the model. The other methods are hampered by computational burden in model fitting and data integration complexities over varying finite window lengths.

4.4 Full Scale Pilot Plant Data Assessment

In this section, CPA using the proposed methodology is experimented on real closed loop data acquired from a pH neutralization pilot plant as shown in Fig. 8. The following three PI control loops were studied in this work; acid flow control (FIC101), continuous stirred reactor tank (CSTR) level control (LIC100) and the reactor pH control (AIC100). The conventional PI controller output is given as:

$$U(t) = k_c\left(E(t) + \frac{1}{\tau_i}\int E(t)dt\right) \qquad (24)$$

where k_c and τ_i represents the controller proportional gain and integral reset time respectively. For the purpose of evaluating the MC-SVM CPA tool, the PI controller parameters was adjusted accordingly to obtain different closed loop responses for three different types of processes on the real-time plant used in the study. These processes were selected because of their unique behaviour under closed-loop dynamical conditions. Information exchange between the pilot plant DCS and the monitoring PC was achieved using an open process control (OPC) server as illustrated in Fig.9. ABB® FREELANCE® software was used to establish the OPC link to MATLAB™ environment where the developed CPA GUI interpreted batches of plant data sets automatically. Corresponding plant data for the loops are shown in Fig.10-Fig.12.

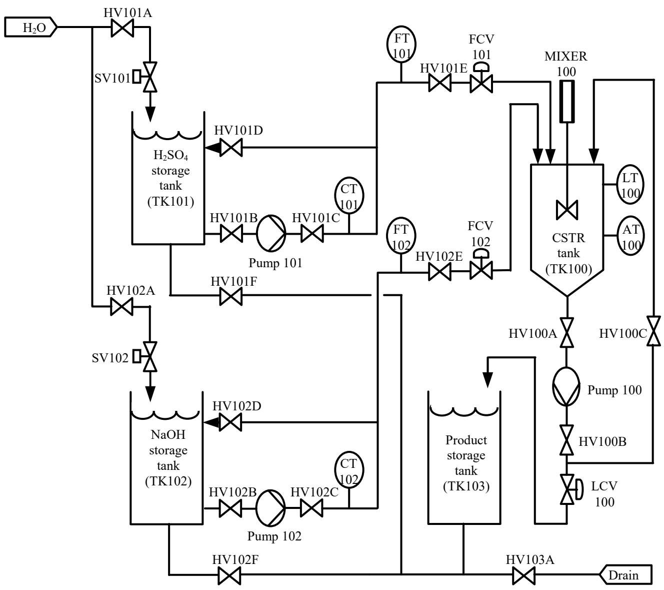

Fig. 8. P&ID of the full scale pH neutralization CSTR.

CSTR tank level control (LIC100) was regulated at 40% with constant inlet flow rate of 20 litres per minute. Level control valve (LCV100) controls the outlet flow rate to maintain tank level at SP. For the pH and flow controllers, SP step changes were made to ensure sufficient process activity in the data sets since the controllers were designed for SP tracking. Results of the experiments for the plant loops are given in Table 5.

The results shown in Fig.10-Fig.12 reveal that the proposed CPA classification tool has acceptable generalisation capabilities for both SP tracking and regulatory controller designs. With regards to Fig.10-Fig.12, the left column indicates poor performance; the right column represents satisfactory performance. Even though the MC-SVM framework was developed using a simple FOPDT process model, it was successful in categorizing plant data derived from nonlinear pH process and integrating level control loops.

Table 5. Experimental results of the pilot pH neutralization plant.

MC-SVM CPA results for pH CSTR plant loops LIC100-CSTR level, FIC101-acid flow rate, AIC100-CSTR pH							
Loop	k_c	τ_i [s]	Class	$\bar{\rho}$	$\mathrm{var}(\rho)$	$iqr(\rho)$	$\gamma(\rho)$
LIC 100	2.0	255	C2	0.364	0.061	0.371	0.492
	8.0	255	C3	0.105	0.055	0.137	2.531
FIC 101	4.1	1.37	C5	0.004	0.054	0.223	0.206
	5.5	1.54	C3	0.052	0.023	0.079	2.921
AIC 100	5.0	687	C5	-0.016	0.234	0.832	0.282
	1.2	787	C3	0.013	0.078	0.192	2.008

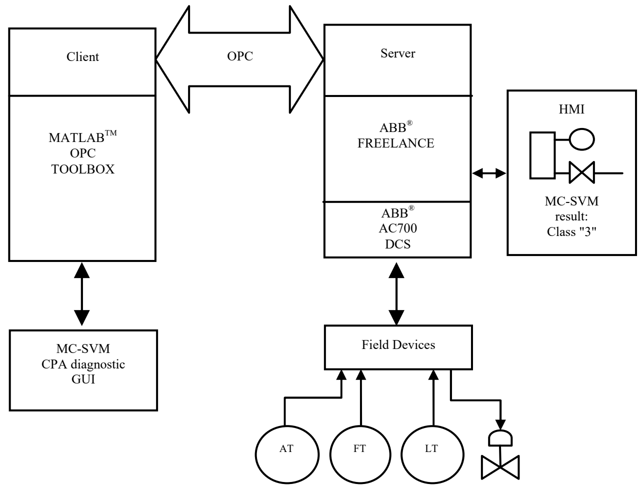

Fig. 9. ABB®FREELANCE® and MATLAB™ OPC architecture.

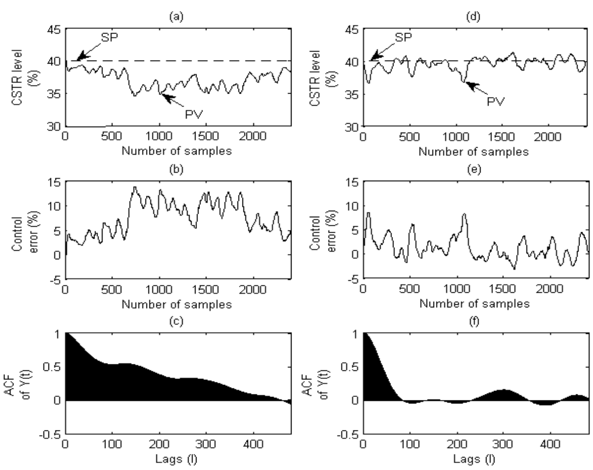

Fig. 10. CSTR tank level regulatory control. (a) and (d) – n=2400 samples of tank level % data, (b) and (e) – control error, (c) and (f) – ACF(l=480) of the process variable.

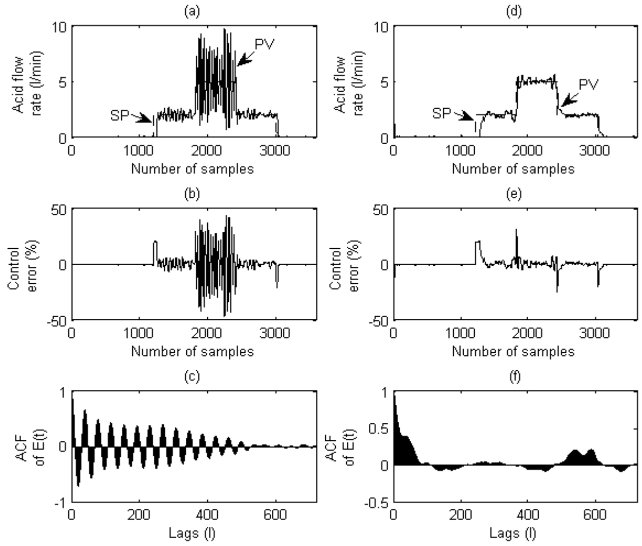

Fig 11. Acid flow rate SP tracking control. (a) and (d) – n=3500 samples of flow rate data, (b) and (e) – control error, (c) and (f) – ACF (l=700) of the control error.

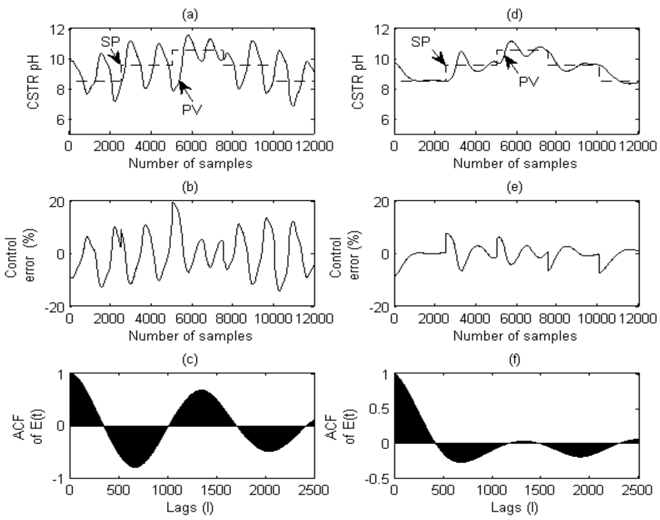

Fig 12. CSTR pH SP tracking control. (a) and (d) – n=12000 samples of pH data, (b) and (e) – control error, (c) and (f) – ACF (l=2500) of the control error.

4.5 Application to Industrial Steam Desuperheater Control

The presented method was also applied to real industrial data sets from a steam temperature desuperheater control from the utility supply section of a pulp and paper mill. Desuperheater control is often regarded as challenging due to its high order nonlinearity and load dependency. Steam temperature is regulated at 200 °C using a desuperheater unit, cooling water valve and a PI temperature controller with a temperature sensor located downstream.

Fig. 13(a-c) highlights the findings of initial steam temperature data acquired from the plant DCS following a SP change. 250 data points of the steam temperature ($Y(t)$) and control error ($E(t)$) were captured at a sample rate of 15 seconds. With a SP step increase to 210 °C, the steam temperature showed steady oscillatory behaviour. Applying the through the MC-SVM classifier tool identified the closed loop as belonging to "Class 5" category as given in Table 6.

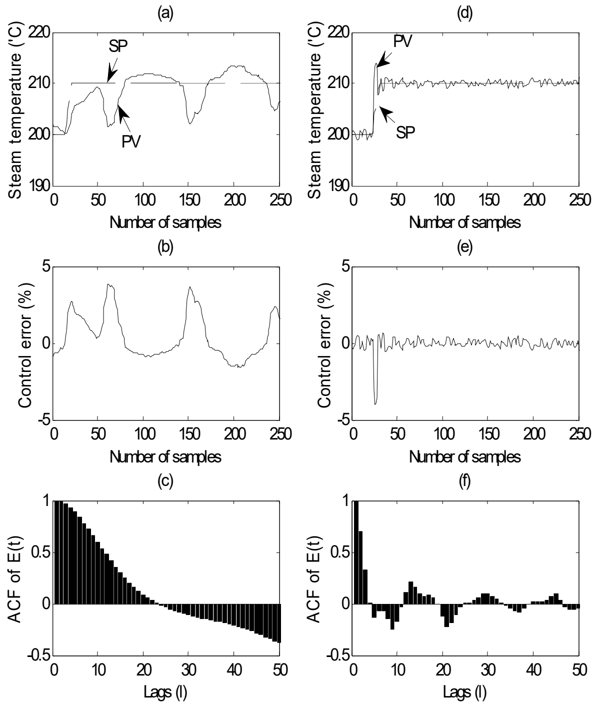

Fig 13. Steam temperature desuperheater SP tracking control. (a) and (d) – n=250 samples of temperature data, (b) and (e) – control error, (c) and (f) – ACF (l=50) of the control error.

Upon further investigation of the control loop, the automatic controller was discovered to have inappropriate parameter settings and was later fine-tuned, the results of which are illustrated in Fig. 13(d-f). Improved controller performance using the new PI parameters was verified by the MC-SVM classifier output indicating "Class 3" performance in Table 6. The mean square error (MSE) for the control loop after controller fine tuning indicates substantial improvement.

Table 6. Feature extraction and MC-SVM results for steam temperature datasets before and after fine tuning.

MC-SVM CPA results for steam temperature control						
	Class	MSE	$\bar{\rho}$	$\text{var}(\rho)$	$iqr(\rho)$	$\gamma(\rho)$
Before fine tuning	C5	0.368	0.115	0.178	0.597	0.849
After fine tuning	C3	0.171	0.029	0.039	0.130	3.004

5. CONCLUSION

In this paper, a novel data driven automated CPA tool using MC-SVMs to classify closed loop performance is proposed. This is achieved using only ACF signatures from routine data records and rudimentary statistical feature extraction. No other knowledge is required other than the fact that process dynamics following setpoint changes and disturbances are fully captured by the ACF pattern.

For the data sets examined in the simulation experiments, a high degree of accuracy from the MC-SVM classification method was achieved. A level of 96.9 % classification accuracy was obtained using the RBF kernel. In pursuit of further improving the accuracy of the method, the problem of feature selection is an open issue.

Application of the method on real world data sets yielded promising results. Automatic batch processing of multiple control loop data sets is currently being investigated and incorporated into the GUI tool. This will enable the simultaneous diagnosis of multiple control loop performance.

The main limitations of the method are:

- To determine an apt ACF, the method requires adequate process data in order to correctly classify plant behavior;
- For slow processes (e.g. pH control), the method requires a 'window of data' which may take a while to acquire due to the relatively large sampling intervals used in processes having slow dynamics;
- The method is confined to SISO systems and is not suited to open-loop unstable systems.

ACKNOWLEDGEMENTS

This research has been supported by the National Research Foundation (NRF) of South Africa under grant number: TTK1206121201. Technical support from ABB® South Africa, Department of Control Technologies is gratefully appreciated.

REFERENCES

Chapelle, O., Haffner, P. and Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10 (5), 1055-1064.

Choudhury, M. S., Thornhill, N. F. and Shah, S. L. (2005). Modelling valve stiction. *Control Engineering Practice*, 13 (5), 641-658.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20 (3),273-297.

Ender, D. B. (1993). Process control performance: Not as good as you think. *Control Engineering*, 40 (10), 180-190.

Howard, R. and Cooper, D. (2010). A novel pattern-based approach for diagnostic controller performance monitoring. *Control Engineering Practice*, 18 (3), 279-288.

Jelali, M. (2013). *Control Performance Management in Industrial Automation: Assessment, Diagnosis and Improvement of Control Loop Performance*. Springer.

Kampouraki, A., Manis, G. and Nikou, C. (2009). Heartbeat time series classification with support vector machines. *IEEE Transactions on Information Technology in Biomedicine,* 13 (4), 512-518.

Karra, S. and Karim, M. N. (2009). Comprehensive methodology for detection and diagnosis of oscillatory control loops. *Control Engineering Practice*, 17 (8), 939-956.

Rengaswamy, R., Hägglund, T. and Venkatasubramanian, V. (2001). A qualitative shape analysis formalism for monitoring control loop performance. *Engineering Applications of Artificial Intelligence*, 14 (1), 23-33.

Rinehart, N. (1997). The impact of control loop performance on the successful application of advanced control. In Proceedings -*Aspen World Conference*, Boston, MA.

Spinner, T., Srinivasan, B. and Rengaswamy, R. (2014). Data-based automated diagnosis and iterative retuning of proportional-integral (PI) controllers. *Control Engineering Practice*, 29, 23-41.

Srinivasan, B., Spinner, T. and Rengaswamy, R. (2012). Control loop performance assessment using detrended fluctuation analysis (DFA). *Automatica*, 48 (7), 1359-1363.

Widodo, A. and Yang, B.S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21 (6), 2560-2574.

Yang, B.S., Han, T. and Hwang, W.W. (2005). Fault diagnosis of rotating machinery based on multi-class support vector machines. *Journal of Mechanical Science and Technology*, 19 (3), 846-859.