A MODELLING-ORIENTED SCHEME FOR CONTROL CHART PATTERN RECOGNITION

by

HÉCTOR DE LA TORRE GUTIÉRREZ

A thesis submitted to the University of Birmingham for the degree of DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering
School of Engineering
College of Engineering and Physical Sciences
University of Birmingham
December 2016

ABSTRACT

Control charts are graphical tools that monitor and assess the performance of production processes, revealing abnormal (deterministic) disturbances when there is a fault. Simple patterns belonging to one of six types or compound patterns made up of more than one type can be observed when a fault is occurring, and a Normal pattern when the process is performing under its intended conditions. Machine Learning (ML) algorithms have been implemented in this research to enable automatic identification of these patterns. The first phase in the operation of a supervised learning ML algorithm consists in training the algorithm by presenting patterns similar to those to be classified afterwards. If real control chart data is not available for training purposes, synthetic patterns must be generated.

Two pattern generation schemes (PGS) for synthesising patterns are proposed in this work. These PGSs ensure generality, randomness, and comparability, as well as allowing the further categorisation of the studied patterns. One of these PGSs was developed for processes that fulfil the NIID (Normally, identically and independently distributed) condition, and the other for three first-order lagged time series models. This last PGS was used as base to generate patterns of feedback-controlled processes.

Using the three aforementioned processes, control chart pattern recognition (CCPR) systems for these process types were proposed and studied. Furthermore, taking the recognition accuracy as a performance measure, the arrangement of input factors that achieved the highest accuracies for each of the CCPR systems was determined. It is worth to note that two of the

time series used for studying autocorrelated processes had never been studied before. Furthermore, a CCPR system for feedback-controlled processes was developed and several possible combinations of inherent factors of this process type were studied, namely, the feedback controller type, the time series model that describes the control chart noise and the signal to monitor.

DEDICATION

This PhD research Thesis is dedicated to all members of my family, friends and girlfriend, especially to my: father, Jose Luis; mother, Maria de Lourdes; brothers, nephews, nieces; and my girlfriend, Giovanna.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Duc Truong Pham for the guidance, support and supervision unconditionally offered along this PhD. His guidance helped me in all the time since the research and writing of this thesis. Without his priceless support, this journey could not even have started.

Also, I gratefully acknowledge the CONACyT (Consejo Nacional de Ciencia y Tecnologia) for enabling me to pursue graduate studies at the University of Birmingham under grant No. 381672.

Furthermore, I want to acknowledge the high performance computing support given by the Birmingham Environment for Academic Research (BEAR).

I also thank my fellow officemates for all the support and feedback given. Also, I express my sincere gratitude to my girlfriend, Giovanna Villalobos, for all her unconditional and priceless support given from the very first day.

Last but not least, I would like to express my thankfulness and love to my family for supporting me mentally, physically and emotionally during this entire journey. Without them, I could have achieved nothing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ANOVA	Analysis of variance
AR(1)	First-order autoregressive
ARMA(1,1)	First-order autoregressive moving-average
BA	Bees algorithm
BESSEL	Bessel kernel
CCPR	Control chart pattern recognition
CI	Confidence interval
CYC	Cyclic pattern
DS	Downward shift
DT	Downward trend
D_t	Disturbance magnitude at time t
EPC	Engineering process control
e_t	Random error (white-noise) at time t
IMA (1,1)	First-order integrated moving-average
IRT	Input representation technique
k_D	Coefficients for the derivative part of the controller
k_I	Coefficients for the integral part of the controller
k_P	Coefficients for the proportional part of the controller
LAPLA	Laplace kernel
MA(1)	First-order moving-average model
ML	Machine Learning
MMSE	Minimum mean square error
MSE	Mean square error

n	Sample size
NIID	Normally, identically and independently distributed
NLM	Nonlinear model
NLM-ARMA	Nonlinear regression model with autoregressive moving-average errors
NORM	Normal pattern
N_t	Inherent noise at time t
PGS	Pattern generation scheme
PGS-1	Pattern generation scheme with $\alpha = 0.01$
PGS-C	Conventional pattern generation scheme
PI	Proportional integral
PID	Proportional integral derivative
PNN	Probabilistic Neural Network
POLY	Polynomial
r_1	Autocorrelation coefficient
RBF	Radial basis function
sin	Sine function
SPC	Statistical process control
SSE	Sum of squared error
St	Scaled variable at time t
SVM	Support Vector Machine
SYS	Systematic
T	Target value
t	Time
TANH	Hyperbolic Tangent

US	Upward Shift
UT	Upward trend
X_t	Controller compensation at time t
Y_t	Measurement of the quality characteristic under study at time t
Z_t	Observed quality characteristic after the controller compensation
α	Significance level
β	Abnormal pattern parameter
ε_t	Random error of the fitted regression model at time t
μ	Mean value
ϕ	Autoregressive coefficient
σ_e	Standard deviation of the white noise
σ_N	Standard deviation of the variable N
σ_Y	Estimated standard deviation of the variable Y
τ	Time when a break point occurs
θ	Moving-average coefficient

1 INTRODUCTION

1.1 BACKGROUND

Root Cause Analysis is an important task in most quality assurance systems. A good quality assurance system is one that can quickly and precisely find and address quality failures. To achieve this, quality assurance is assisted by Statistical Quality Control in order to monitor the production system and its critical quality features. In recent production systems, the identification of causes of poor quality has been a comprehensive task that sometimes can be uncertain because of human intervention. Statistical Quality Control is used to sort out this uncertainty problem, but even so, human intervention is required in conventional Statistical Quality Control systems. An example of this is the identification of patterns in Statistical Process Control Charts and the one-one matching of these patterns with their assignable causes.

Control charts are graphical tools that monitor and assess the performance of production processes, revealing abnormal (deterministic) disturbances when there is a fault. When the process is operating normally, a “Normal” (NORM) control chart pattern can be observed (Figure 1.1). If an assignable cause is affecting the process, the control chart can exhibit one or more of fourteen types of patterns (see Western Electric Company, 1956), six of these being considered as basic patterns: Upward/Downward Trends (UT/DT), Upward/Downward Shifts (US/DS), Cycles (CYC) and Systematic (SYS) (Figure 1.1). The remaining eight patterns are either particular cases or combinations of these basic patterns.

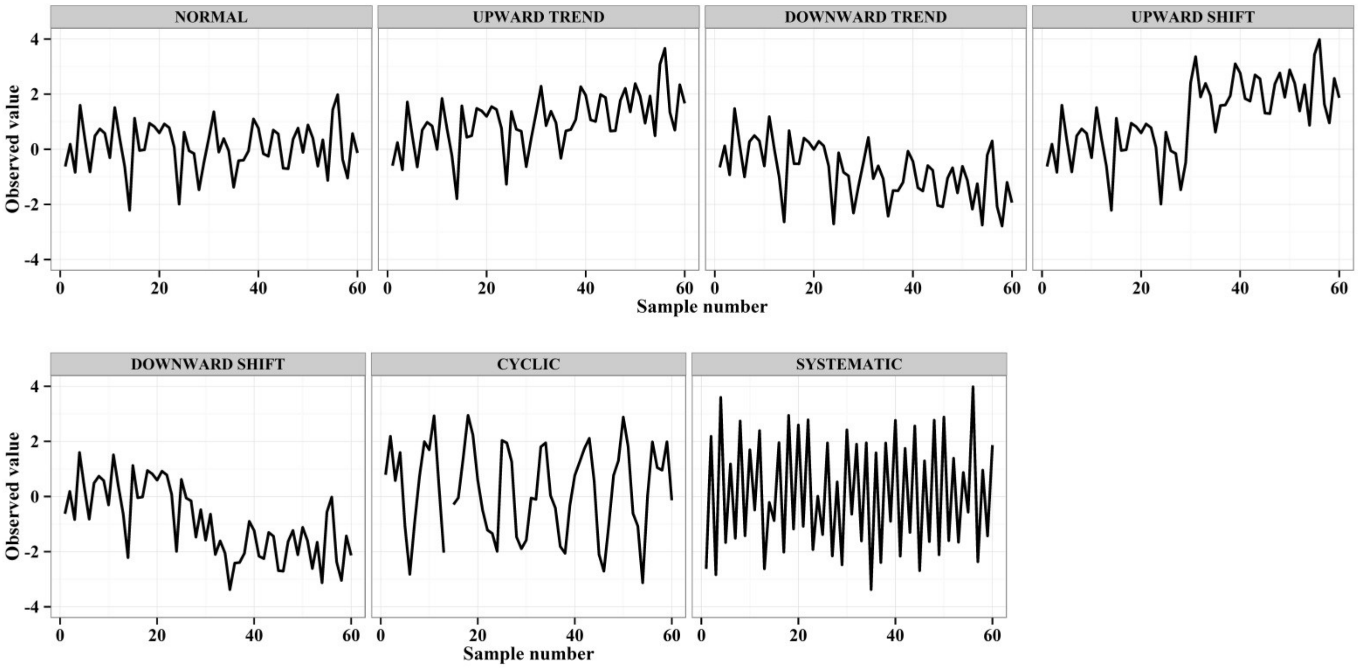

Figure 1.1: Seven simple control chart patterns

For pattern identification and cause assignment, it is necessary to identify abnormalities in the current control chart and extract information such as frequency, magnitude, time when a certain abnormality happened, etc. This data is obtained to help the root cause analysis in the efficient identification of assignable causes of poor quality in the production system; such information can help to distinguish between patterns that are identified as the same but have different root causes, e.g., a Cyclic pattern with period of 12 might be produced due to wear of a tool, while another pattern with period equal to 24 might be caused by variations in the input voltage.

In recent years, Machine Learning (ML) algorithms have been implemented to enable automatic identification of control chart patterns. The first phase in the operation of a supervised learning ML algorithm consists in training the algorithm by presenting patterns similar to those to be classified afterwards. Ideally, observation samples (process data) should be collected from the real process environment and be used as inputs during the training. However, since a large amount of data is required for control chart pattern recognition (CCPR), synthetic samples need to be generated. This is commonly done using Monte-Carlo simulation (Hachicha and Ghorbel, 2012).

A drawback of Monte-Carlo simulation is that this method is not sufficiently robust to noise which can greatly affect the generalisation ability of the algorithm if special care is not paid to the patterns used for training. Consequently, the statistical properties of the patterns generated for training the algorithm can be altered, e.g., a positive slope can appear shallower or even disappear altogether due to noise, and so the ML algorithm is trained with a-priori misclassified patterns, thus setting incorrect pattern classification boundaries.

One of the objectives of this research was to develop pattern generation schemes (PGS) for CCPR that are robust to variations in the pattern parameters used for training ML algorithms. Thus, synthetic patterns generated from PGSs for purposes of training CCPR systems should fulfil the following conditions:

- Being generated from a broad range of fully randomised training pattern parameters (slope, shift magnitude, break point position, systematic departure (Hachicha and Ghorbel, 2012), amplitude and frequency of cycles). This is to ensure that the CCPR system is able to identify a wide variety of patterns.
- Having parameters that are statistically significant. This is to ensure that correct decision boundaries for pattern classification are obtained by the ML algorithm. For

example, the slope of an UT pattern could be reduced or completely removed by noise. The statistical significance of the slope must be tested and found to be positive for a pattern to be classified as a Trend. If this condition is fulfilled, correct decision boundaries for pattern classification can be obtained by the ML algorithm.

- Being generated by an automatic and reproducible procedure, so allowing other researchers to reproduce the same patterns for comparison purposes.

The efficient and accurate identification of the seven simple patterns has been a problem studied by researchers during the last two decades. Most of those studies have related to control chart pattern recognition (CCPR) when the inherent disturbance of the process is normally, independently and identically distributed (NIID). Unfortunately, the assumption of uncorrelated (independent) observations is not even approximately satisfied in many real processes (for example, chemical processes where consecutive measurements of process or product characteristics are often highly correlated), the first-order autoregressive model (AR(1)) being the only utilised so far. Another purpose of this work was to develop new CCPR systems where the inherent disturbance is modelled by other stationary time series. Therefore, it is also necessary to determine the effect of the PGS on the recognition accuracies of the CCPR system, and the type of system and input representation technique (IRT) giving the highest accuracies.

Reduction of the variance in the monitored variables is essential in modern manufacturing due to the need to increase quality and reduce scrap. In discrete part manufacturing, SPC has shown to be effective at addressing assignable causes. However, when the aim of monitoring is the reduction of the variance of the process, SPC has not always succeeded (see the discussion provided in Box et al. (2009), pages 1-3). On the other hand, in continuous processes, different process conditions are observed. The most common control charts cannot be used to monitor a continuous process as the assumption of time independence of quality characteristics does not hold in most continuous processes (Montgomery, 2009). In these processes, it is usual that the variance of the output variable (the quality characteristic of interest) is reduced by means of a compensation or regulation scheme of manipulatable variables. The main aim of such a scheme is to reduce the variance around the nominal value of the characteristic (target). These process compensation or regulation schemes are known as Engineering Process Control (EPC), stochastic control, or feedback/feedforward control, depending on the nature of the adjustments (Montgomery, 2009). An ideal quality control system is one that efficiently identifies patterns due to assignable causes and keeps the variance of the process at the minimum level, always around the nominal value of the process. Quality control systems that simultaneously monitor manipulatable variables and compensate for their effects are known as SPC-EPC or synergistic control systems.

EPC schemes assume that changes in the manipulatable variables will have repercussions on the output variable given a specific dynamic model that links these variables. If this dynamic model is correct, the variance of the output variable is reduced. However, when certain types of external disturbances or assignable causes occur that are outside the framework of this dynamic model, then the compensation rules will not completely account for them. As a result, variability will be increased. By applying SPC in a specific way, these assignable causes can be detected and the combined SPC-EPC procedure will be more effective than EPC alone (Montgomery, 2009).

CCPR systems for SPC-EPC schemes must take into account how the seven simple patterns can be affected by the control scheme implemented by EPC. Another objective of this work was to develop CCPR systems for cases where the seven simple patterns are affected by a control scheme, as well as determining which variable, the output variable or the control signal, is suitable for efficient identification of assignable causes through CCPR systems.

Two of the most common and effective controllers were implemented and compared, namely, the Proportional Integrative Derivative (PID) and the Minimum Mean Squared Error (MMSE) controllers. As a consequence of applying these controllers, two signals were produced: the controller output and the plant output after application of the controller. As aforementioned, another aim of this work was to determine which signal to monitor to accurately recognise patterns.

As the design of CCPR systems for feedback-controlled processes is a problem that has never been formally studied, the combination of input factors that achieves the highest accuracies is an unresolved topic. The relevant input factors are: PGS, IRT and the type of ML algorithm.

1.2 OBJECTIVES OF THE RESEARCH

In summary, the main objectives of this research were:

1. To develop a systematic method to set the range of parameter values used during the pattern generation that fulfils the three PGSs characteristics listed on pages 3 and 4. PGSs were developed for NIID, autocorrelated and feedback-controlled processes.

2. To study and develop CCPR for autocorrelated processes that had not yet been studied in the CCPR literature. These were first-order moving-average (MA(1)) and first-order autoregressive moving-average (ARMA(1,1)) time series models.
3. To measure the impact of the proposed PGSs on the recognition accuracy of the CCPR systems.
4. Develop and study CCPR systems for different feedback-controlled processes. These schemes had never been formally studied in the CCPR domain.
5. To apply a popular optimisation algorithm, the Bees Algorithm (BA), to determine the best set of free parameters in order to achieve the highest recognition accuracies.
6. To use a state-of-the-art and efficient ML algorithm as pattern identifier in the CCPR system.
7. To determine the best arrangement of input factors in each of the three CCPR system types developed.

1.3 RESEARCH METHODOLOGY

The PGSs presented in this work comprise three steps: initial pattern generation, mean change classification and pattern categorisation. In the first step, the generality of the CCPR system is ensured thanks to the wide-range of training pattern parameters values used and the random assignment of these during pattern generation. In the second step, break points (if there are any) are detected. Due to noise, this step is not straightforward. First, the most likely position of a potential break point is determined. Then, the statistical significance of the magnitude of this most likely break point is computed to decide whether there really is a break point. If so, the amount of shift of the pattern is determined and the pattern is categorised as US or DS.

Determining the regression or time series model that best explains the behaviour of the variable to monitor was necessary for the development of the proposed PGSs. A Nonlinear Regression model precisely represented the monitored variable when the inherent noise was NIID. A dynamic regression model precisely described the studied variables when the inherent noise was autocorrelated. This model type also ensured the unbiased estimation of the common-cause and abnormal pattern parameters.

The recognition accuracies of the ML algorithm were used as measures of the performance of the PGSs, i.e., the impact on the accuracies when correct decision boundaries were studied. The best arrangement of input factors involved in the design of the CCPR systems was determined using post-hoc Tukey and Analysis of Variance (ANOVA) tests. Two ML algorithms were utilised as recognition systems when the NIID process was studied. As the Support Vector Machine (SVM) algorithm showed the highest accuracy for that process type, only that ML algorithm was employed in the following CCPR systems.

1.4 OUTLINE OF THE THESIS

The next chapter commences with an extensive literature review regarding the three process types previously mentioned and other topics related to monitoring and pattern recognition for these process types. In that chapter, some theoretical background is also provided on the regression models used to develop the proposed PGSs, ML algorithms, feature extraction techniques and the BA.

Chapter 3 presents the CCPR developed for NIID processes. Furthermore, the effect of the PGS on recognition accuracies is described. Two different ML algorithms were used as recognition systems of the CCPR; the accuracies achieved by those two ML algorithms with different kernel types and two IRTs are also presented.

In the work reported in chapter 4, the inherent noise of the control chart was modelled by three different time series models. In that chapter, a detailed analysis of the accuracies obtained is provided to find the overall accuracies and the best arrangement of factor input factor of the CCPR system.

A new CCPR system for feedback-controlled processes is presented in chapter 5. In this system, special attention is paid to the generation of patterns and signals to monitor by the SPC system. As this topic had not been formally studied for cases where ML algorithms were applied to identify patterns, it was necessary to determine the best set of input factors and then find the highest accuracy achieved by the CCPR system for the two time series models utilised.

Finally, chapter 6 summarises the work and presents the conclusions reached. Suggestions for further research are also provided in the chapter.

2 BACKGROUND

2.1 PRELIMINARIES

This chapter covers the background required for developing this research. Firstly, an extensive review of the literature on monitoring and pattern recognition for NIID, autocorrelated and feedback-controlled processes is provided. Secondly, the theoretical background concerning the models fitted to the control chart data is outlined; the models are Nonlinear, time series and Nonlinear with ARMA errors models (NLM-ARMA).

The procedure for obtaining the seven shape features used in this work is also shown in this chapter. A mathematical explanation of pattern classification using SVMs and Probabilistic Neural Networks (PNN) is given. Finally, the BA is presented and explained.

This chapter is organised as follows: section 2.2 reviews the key publications relevant to this work. The theory relating to the models fitted to the control chart data during pattern generation is provided in section 2.3. The mathematical explanation of the two ML algorithms is given section 2.4. Section 2.5 shows the procedure to obtain the shape features used in this work. Finally, the BA is presented in section 2.6.

2.2 LITERATURE REVIEW

This section reviews previous work on pattern synthesis for training CCPR systems, application of PNN and SVM in CCPR and estimation of abnormal pattern parameters when ML algorithms are used for identifying patterns in control charts.

2.2.1 NIID PROCESSES

2.2.1.1 *Pattern generation for CCPR systems*

Little attention has been paid to the pattern parameters used during training. Most authors have adopted the pseudorandom number generator proposed by Matsumoto and Nishimura (1998) and implemented as default in software such as R and MATLAB® for the generation of the inherent noise.

It was found that the values of the abnormal pattern parameters used for training and testing the recognition systems varied greatly. Table 2.1 shows the range of the parameter values used for generating patterns in the most relevant recent works. However, in the table, it can be observed that there are two parameters that have not been randomised during pattern generation, namely, the break point position and the period of the Cyclic patterns. These two parameters must be randomly generated in a proper PGS in order to enable to the recognition system to identify a broader variety of pattern types and magnitudes.

So far, Barghash and Santarisi (2004) have been the only authors to have studied the training parameter problem. They used design of experiments to assess the effect of some of the parameter ranges used during pattern generation, finding that the values of maximum and minimum shifts and slopes greatly affect Type-1 and Type-2 errors. Other authors such as Pacella et al. (2004) and Guh (2004) studied other parameters like Neural Network configuration and inspection window length and how they influenced the performance of the learning algorithm.

Table 2.1: Maximum and minimum values of the training pattern parameters

Pattern type	**UT, slope**		**DT, slope**		**US / DS break point**		**US, Magnitude**		**DS, Magnitude**		**CYC, Amplitude**		**CYC, Frequency**		**SYS, systematic departure**	
Range / Author (s)	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Kazemi et al. (2015)	0.015	0.025	-0.025	-0.015	One point		0.7	2.5	-2.5	-0.7	0.0	2.5	10		--	--
Chompu-inwai and Thaiupathump (2015)	0.05	0.10	-0.10	-0.05	One point		1.5	3.0	-3.0	-1.5	1.5	3.0	15		--	--
Xanthopoulos and Razzaghi (2014)	0.005	0.605	-0.605	-0.005	One point		0.005	1.805	-1.805	-0.005	0.005	1.805	Fixed		0.005	1.805
Xie et al. (2013)	0.05	0.10	-0.10	-0.05	One point		1.5	2.5	-2.5	-1.5	1.5	2.5	8, 16		1.0	3.0
Gu et al. (2013)	0.05	0.10	-0.10	-0.05	One point		2.5	3.0	-3.0	-2.5	1.5	2.5	20		1.0	3.0
Du et al. (2013)	0.0	0.22	-0.22	0.0	One point		0.0	2.5	-2.5	0	1.0	2.5	8		1.5	3.0
Bag et al. (2012)	0.05	0.10	-0.10	-0.05	Three points		1.5	2.5	-2.5	-1.5	1.5	2.5	8, 16		1.0	3.0
Gauri (2012)	0.05	0.10	-0.10	-0.05	Three points		1.5	2.5	-2.5	-1.5	1.5	2.5	8, 16		1.0	3.0
Yu (2012)	0.05	0.15	-0.15	-0.05	One point		1.5	2.5	-2.5	-1.5	1.5	3.0	12		1.5	3.0
Lu et al. (2011)	0.05	0.10	-0.10	-0.05	Three		1.5	2.5	-2.5	-1.5	1.5	2.5	8,16		1.0	3.0

Pattern type	**UT, slope**		**DT, slope**		**US / DS break point**		**US, Magnitude**		**DS, Magnitude**		**CYC, Amplitude**		**CYC, Frequency**		**SYS, systematic departure**	
Range / Author (s)	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
					points											
Hassan et al. (2011)	0.015	0.025	-0.025	-0.015	One point		0.7	2.5	-2.5	-0.7	0.5	2.5	10		--	--
Ranaee et al. (2010)	0.20	0.50	-0.50	-0.20	One point		1.5	4.0	-4.0	-1.5	0	3	4	12	--	--
Gauri (2010)	0.05	0.10	-0.10	-0.05	Three points		1.5	2.5	-2.5	1.5	1.5	2.5	8,16		1.0	3.0
Jiang et al. (2009)	0.10	0.26	-0.26	-0.10	One point		1.0	3.0	-3	-1	Not given[1]		7,8,9		--	--
Cheng and Cheng (2009)	0.10	0.30	-0.30	-0.10	One point		0.5	3.0	-3.0	-0.5	0.5	3.0	12		--	--
Gauri and Chakraborty (2009)	0.05	0.10	-0.10	-0.05	Three points		1.5	2.5	-2.5	-1.5	1.5	2.5	8, 16		--	--
Wang et al. (2008)	0.05	0.10	-0.10	-0.05	One point		1.5	3.0	-3.0	-1.5	1.5	3.0	15		--	--
Wang and Kuo (2007)	0.10	0.25	-0.25	-0.10	One point		1.5	3.0	-3.0	-1.5	1.5	3.0	16		--	--
Pham et al. (2006)	0.2	0.5	-0.5	-0.2	One point		1.5	4.0	-4.0	-1.5	0.0	3.0	4	12	--	--
Yang and Yang	0.05	0.12	-0.05	-0.12	One point		1.0	2.5	-2.5	-1.0	0.5	2.5	8		0.5	2.5

[1] This pattern range is not provided in the paper.

Pattern type	**UT, slope**		**DT, slope**		**US / DS break point**		**US, Magnitude**		**DS, Magnitude**		**CYC, Amplitude**		**CYC, Frequency**		**SYS, systematic departure**	
Range / Author (s)	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
(2005)																
Pham and Chan (2005)	0.04	0.10	-0.10	-0.04	One point		1.5	4.0	-4.0	-1.5	0.0	3.0	4	12	--	--
Guh (2005)	0.10	0.26	-0.26	-0.10	One point		1.0	3.0	-3.0	-1.0	1.0	3.0	8		1.0	3.0
Al-Assaf (2004)	0.0	0.22	-0.22	0.0	10	27	0.0	2.5	-2.5	0.0	0.1	2.5	8		--	--
Guh and Shiue (1999)	0.0	0.14	-0.14	0.0	One point		0.0	7.0	-7.0	0.0	1.0	3.0	10	50	--	--

2.2.1.2 *Estimation of parameters of abnormal patterns*

In the literature, models that deal with the recognition and classification of patterns in addition to estimating their corresponding parameters are very rare (Lesany et al., 2013).

Guh and Tannock (1999) developed an intelligent system capable of recognising common abnormal patterns and identifying their characteristics. However, their method was not statistically robust as they used small ranges of abnormal patterns parameters during the training of the recognition system, and did not estimate the parameters at the end of the system, considering them only as another characteristic to identify. Other authors such as Guh (2003, 2005), Jiang et al. (2009) and Shaban and Shalaby (2012) created a sequence of intelligent sub-systems where in a first step, the pattern was identified by a sub-system and the magnitude of the pattern subsequently determined by another sub-system. Training and testing were carried out with patterns generated from a narrow range of pattern parameters.

Possible misclassification due to the effect of noise during pattern generation was ignored in all the reviewed papers; thus, the potential increase in the probability of Type 1 and Type 2 errors was neglected.

2.2.1.3 *CCPR for NIID processes*

The design of CCPR systems for processes where the inherent disturbance is NIID is the most common problem studied in recent research, as this NIID condition is important for monitoring production processes by means of traditional control charts, such as X and $\bar{X}$ charts (Hachicha and Ghorbel, 2012).

Pham and Wani (1997) proposed a set of shape features to be extracted directly from the CC data and to be used as inputs for CCPR systems, increasing the pattern recognition accuracies and recognition stability. Based on these features, Gauri and Chakraborty (2006b, 2009) proposed another set of shape features that not only increased the pattern recognition accuracies and recognition stability but also were independent of the scale of the CC data.

As mentioned above, models that deal with the recognition and classification of patterns in addition to estimating their corresponding parameters are very rare (Lesany et al., 2013). Barghash and Santarisi (2004) studied the effect of training parameters on the performance of the CCPR system, finding that the values range used during pattern generation greatly affects Type 1 and Type 2 errors. Cheng and Cheng (2008) highlighted the importance of the same parameter in the generalisability of the CCPR model.

In more recent work, authors such as Lu et al. (2011), Du et al. (2013) and Xie et al. (2013) have focused on the application of signal analysis techniques to pre-process the CC data in order to enhance the performance of the CCPR system.

Xanthopoulos and Razzaghi (2014) proposed the use of weighted Support Vector Machines for CCPR. Ranaee et al. (2010) introduced a CCPR system using a SVM as the recognition algorithm and the particle swarm optimisation algorithm to improve the overall performance of the SVM by finding the best set of free parameters.

2.2.1.4 *Feature extraction techniques*

Feature extraction techniques have been developed in order to improve the recognition accuracy, reduce the dimensionality of the input data and decrease the time in training and testing algorithms. Such techniques can be grouped into the following three different classes (Hachicha and Ghorbel, 2012):

i. *Statistical Features:* Values such as the mean, standard deviation, skewness, kurtosis and other statistical measures are taken from the control chart data and used as input to the algorithm. The use of these features has the difficulty that patterns with similar statistical properties, such as shifts and trends, can be confused and thus, misclassified. Moreover, statistical features lose information on the order of the data (Gauri and Chakraborty, 2006b).

ii. *Wavelet Denoise features:* In recent years, Multi-resolution Wavelet analysis has been used to denoise and extract distinct features from CCPs by providing distinct time frequency characteristics (Hachicha and Ghorbel, 2012). Studies such those performed by Al-Assaf (2004), Wang et al. (2007), Cheng and Cheng (2009) and Du et al. (2013) have proved that Wavelet Denoise features help to increase the pattern recognition accuracy. A drawback of this method is that feature extraction is computationally intensive and not totally automated, and feature interpretation is not easily comprehensible to users.

iii. *Shape features:* These represent the main characteristics of the original data in a condensed form. Thus, they facilitate accurate and efficient pattern recognition (Pham and Wani, 1997). Nine shape features were proposed by Pham and Wani (1997), having some drawbacks; their scale depends on the data scale and, if the data are scaled, some important information related to the patterns may be lost; also these features have shown significant correlation between them. Furthermore, the extraction of some of these features requires users' inputs and, thus, this CCP recognition system is not truly automated (Gauri and Chakraborty, 2006a). Gauri (2010) proposed a set of seven shape features based on those proposed by the aforementioned authors and attending to the drawbacks that these suffered. These features achieved a better pattern recognition accuracy, reduction of the training time; the scale of the features is independent to the data scale, and other benefits.

2.2.2 AUTOCORRELATED PROCESSES

This section reviews work related to CCPR for autocorrelated processes, synthesis of patterns for training CCPR systems and estimation of parameters of control charts using CCPR systems.

2.2.2.1 *Monitoring Autocorrelated processes*

The literature on monitoring autocorrelated processes is extensive. This subsection summarises work that is most relevant to this research.

There is a substantial discussion in the literature on choosing the disturbance model for the inherent noise of the process (Wang and Tsung, 2007). Authors such as Zhang and Pollard

(1994), Nembhard and Kao (2003) and Hwarng (2004) used the AR(1) model to describe the inherent disturbance of some industrial processes, Tsung et al. (1998), Jiang and Tsui (2002) and Wang and Tsung (2007) employed the ARMA (1,1) model. Montgomery et al. (2000) and Jiang and Tsui (2002) modelled the inherent disturbance using an integrated moving-average model (IMA (1,1)).

The most common procedure for monitoring autocorrelated processes consists of plotting the residuals of a fitted time series model. If the fitted model is adequate, these residuals will be NIID and thus traditional control charts can be used to monitor them. Recent research found that monitoring residuals affects the detection of mean shifts and highly depends on the ability to fit time series models and a-priori knowledge of the process (Longnecker and Ryan, 1992; Zhang, 1997; Lu and Reynolds, 1999).

Recent advances have been made by forecast-based monitoring schemes to address this problem. Dyer et al. (2003) proposed a forecast-based monitoring scheme for three stationary processes, AR (1), MA (1) and ARMA(1,1). Alwan (1991) investigated the effect of autocorrelation on masking the effect of special causes and also studied how run-rules and static control limits can increase the number of false alarms. Boyles (2000) studied the problem of splitting common-cause signals from assignable-cause signals by means of a standard estimator for first-order autocorrelated processes.

For a literature review and broader discussion of Statistical Process Control methods for monitoring autocorrelated processes, see Psarakis and Papaleonida (2007).

2.2.2.2 *CCPR for Autocorrelated processes*

The identification of patterns where the inherent noise is not NIID is an infrequently studied issue, so few papers dealing with this topic were found. The task of CCPR for autocorrelated processes can be divided into two: identifying changes in process mean, and identifying abnormal patterns like those proposed in Western Electric Company (1956). Furthermore, all CCPR systems developed so far have assumed that the inherent disturbance can be represented by an AR(1) model. CCPR systems for disturbances modelled by other types of times series such as MA(1) and ARMA(1,1) are needed.

The first attempts to apply ML algorithms for CCPR with autocorrelated processes were intended to detect only mean shifts. Chiu et al. (2001) utilised a back-propagation neural network (BPNN) to identify mean shifts in AR(1) processes with varying autocorrelation levels. Hwarng (2004) monitored the mean value of an autocorrelated process by means of a BPNN, comparing the monitoring capability with those achieved using Exponentially Weighted Moving Average (EWMA), special cause control and other charts. Zobel et al. (2004) developed a BPNN-based technique for CCPR for recognising process mean shifts, incorporating a data processing classification algorithm. Hwarng (2005) proposed a neural-network-based identification system for detecting mean shifts and correlation changes. Wu and Yu (2010) used a selective network ensemble approach called Discrete Particle Swarm Optimisation for detecting both mean and variance shifts. Guh (2008) was the first author formally to study the identification of simple patterns in processes of which observations are not independent, using the AR(1) model to describe the inherent noise. He developed an on-line CCPR for each of the nineteen autocorrelation levels studied. The designed on-line CCPRs neglected the biasing factor of the abnormal pattern over the estimated common-cause parameters (Boyles, 2000). The study showed encouraging results, introducing a new path in CCPR research.

Cheng and Cheng (2008) used a multi-resolution analysis approach based on the Haar Discrete Wavelet Transform to denoise, decorrelate and extract features from AR(1) processes. They studied five pattern types. A multi-layer neural network was used as recognition system. The recognition accuracies of the proposed CCPR system were compared with those obtained using raw data as input. The best combination of coefficients of the Haar discrete wavelet transform, approximation and detail, was determined by trial and error, causing long processing times and fuzziness. The proposed feature extraction technique increased the accuracy compared to those achieved using the raw data as input.

Noorossana et al. (2003) were the first to apply neural networks for detecting and classifying non-random disturbances. The patterns under study were referred to as *level shifts, additive outliers* and *innovation outlier.* The inherent disturbance was modelled by an AR(1) model.

Most recent authors have focused on recognition of simple patterns by on-line ensembles. Lin et al. (2011) designed an on-line real-time CCPR using SVMs as pattern classifiers, training one SVM for each autocorrelation level. Yang and Zhou (2015) developed an Learning-Vector-quantisation-based ensemble for on-line pattern recognition of the seven simple patterns, providing the autocorrelation level as additional information about the processes.

Cheng and Cheng (2008), Lin et al. (2011) and Yang and Zhou (2015) used an estimator of autocorrelation for common-cause charts in their recognition systems. They neglected the biasing effect on the autocorrelation coefficient caused by the abnormal disturbance (Woodall and Faltin, 1993; Dyer et al., 2003), so training the ML algorithms based on erroneous autocorrelation coefficients.

In this work, it is proposed a methodology for generating training patterns for CCPR systems taking into account the possible existence of assignable causes. A recognition system will be presented for each of the following stationary time series models: AR(1), MA(1) and ARMA(1,1). In this research, it is assumed independence between the inherent noise distribution the pattern effect.

2.2.2.3 *Generation of patterns of CCPR systems for autocorrelated processes*

As in CCPR systems for NIID processes, the magnitude of the parameters used for generating abnormal patterns for CCPR systems of autocorrelated processes also varied greatly.

Table 2.2 shows the ranges used in the most important papers published. However, in the table, it can be observed that most of the research done so far has mainly focused on the identification of Shift pattern types and their magnitudes. The range of magnitude of these pattern types greatly varied. Furthermore, there are two parameters that have not been randomised, namely, the break point position and the period of the CYC patterns. These two parameters must be randomly selected in a proper PGS in order to enable to the recognition system to identify a broader variety of pattern types and magnitudes.

Table 2.2: Maximum and minimum values of the training pattern parameters

Pattern type	**UT, slope**		**DT, slope**		**US / DS break point**		**US, Magnitude**		**DS, Magnitude**		**CYC, Amplitude**		**CYC, Frequency**		**SYS, systematic departure**	
Range / Authors (s)	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Yang and Zhou (2015)	0.10	0.26	-0.26	-0.10	One point		1.0	3.0	-3.0	-1.0	1.0	3.0	8		1.0	3.0
Lin et al. (2011)	0.10	0.26	-0.26	-0.10	One point		1.0	3.0	-3.0	-1.0	1.0	3.0	8		1.0	3.0
Wu and Yu (2010)	--	--	--	--	One point		0.0	3.0	-3.0	0.0	--	--	--	--	--	--
Cheng and Cheng (2008)	0.10	0.22	-0.22	-0.10	One point		1.0	4.0	-4.0	-1.0	1.0	4.0	12		--	--
Guh (2008)	0.10	0.26	-0.26	-0.10	One point		1.0	3.0	-3.0	-1.0	1.0	3.0	8		1.0	3.0
Hwarng (2005)	--	--	--	--	One point		0.0	3.0	-3.0	0.0	--	--	--	--	--	--
Zobel et al. (2004)	--	--	--	--	One point		0.5	2.0	-2.0	-0.5	--	--	--	--	--	--
Hwarng (2004)	--	--	--	--	One point		0.0	3.0	-3.0	0.0	--	--	--	--	--	--
Noorossana et al. (2003)	--	--	--	--	Four points		1.0	4.0	-4.0	-1.0	--	--	--	--	--	--

Pattern type	**UT, slope**		**DT, slope**		**US / DS break point**		**US, Magnitude**		**DS, Magnitude**		**CYC, Amplitude**		**CYC, Frequency**		**SYS, systematic departure**	
Range / Authors (s)	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Chiu et al. (2001)	--	--	--	--	One point		0.0	3.0	-3.0	0.0	--	--	--	--	--	--

2.2.3 SPC-EPC PROCESSES

This section reviews the most relevant publications related to SPC-EPC monitoring schemes and CCPR for NIID, autocorrelated and feedback-controlled processes.

2.2.3.1 *SPC-EPC monitoring schemes*

The design of quality systems where SPC is integrated with EPC techniques has been an issue studied by several researchers during the last two decades. This subsection reviews work that is most relevant to this work.

In SPC-EPC schemes, monitoring is typically conducted on the output of a controlled process (Del Castillo 2006), but which signal to monitor in SPC-EPC control systems is still an unresolved issue. Jiang and Tsui (2002) demonstrated that monitoring either the output or the control action can be more efficient depending on the autocorrelated process dynamics. Other authors such as Box and Kramer (1992) and Capilla et al. (1999) suggested that monitoring controller actions may improve the chances of early detection of shifts in the mean.

Kandananond (2010) quantified the effect of factors such as types of controllers, control charts and monitored signals on integrated SPC-EPC systems for non-stationary inherent disturbances when the Mean squared error (MSE) and average run length are measured as responses. Wang and Tsung (2007) proposed the use of the T^2 control chart for detecting dynamic patterns in mean shifts of proportional-integral (PI) controlled and MMSE controlled processes for inherent noise modelled by an ARMA(1,1) time series. For a broader discussion and review of the integration of SPC and EPC, see Jiang and Farr (2007).

2.2.3.2 *CCPR for feedback-controlled processes*

The identification of assignable causes of variations in feedback-controlled processes through control chart has been studied by few researchers. The most relevant work is reviewed in this subsection.

The first authors to state the issue of pattern recognition and categorisation were Shao and Chiu (1999). A neural network was used as pattern recognition system. A IMA(1,1) model was adopted to model the inherent noise and a proportional-integral (PI) controller to adjust the process. The Neural network was trained to identify step-changes and linear disturbances and their magnitudes, achieving good recognition accuracies.

Lu et al. (2008) developed a Neural-network-based model with independent component analysis to recognise shifts in the correlated process parameters, again considering only step-changes and linear disturbances. The authors assumed that the process can be modelled by an IMA(1,1) time series and can be adjusted using a proportional-integral (PI) controller.

The issue of detecting the start time of some abnormal patterns was addressed by Shao et al. (2011). The inherent disturbance was modelled by using an AR(1) time series and a MMSE controller was employed to tune the process. SVM and Neural network were used for detecting the start time of step-change patterns with different magnitudes. The proposed system achieved good accuracies.

Shao (2014) trained three SVMs to recognise patterns types by pairs, i.e., studied three pattern types, having three possible pairwise combinations and training one SVM for each pair. The author assumed that the inherent noise can be modelled by an AR(1) series with a given autocorrelation level (ϕ=0.9) and the process can be adjusted by a MMSE controller.

2.2.4 SVM AND PNN

SVM is a relatively recent algorithm in the field of ML. Within less than two decades of being created, many of its advantages with respect to the best existing methods have become evident: generalisation capacity, ease of use and solution uniqueness (De Tejada and Martìnez-Echevarrìa, 2007). SVMs can deal with nonlinear formulations, provide a trade-off between dimensionality (space complexity) and accuracy and have shown good results in pattern recognition applications. Further details on SVMs can be found in Cortes and Vapnik (1995), Burges (1998) and Hsu and Lin (2002).

SVMs have been applied to diverse problems, from text classification (Tong and Koller, 2001), object recognition (Pontil and Verri, 1998), image classification (Chapelle et al., 1999), and bioinformatics (Furey et al., 2000; Hua and Sun, 2001).

As a classification system, SVMs have also been used for CCPR and abnormal patterns parameter identification. Authors such as Xanthopoulos and Razzaghi (2014) and Chinnam (2002b) achieved good pattern recognition accuracies with SVMs. Other authors such as Lu et al. (2011), Du et al. (2013) and Xie et al. (2013) have utilised signal processing techniques such as Independent Component Analysis and Wavelet transforms to pre-process the control chart data, and also achieved good pattern recognition accuracies.

The PNN is a feed-forward neural network based on the Bayesian Criterion and Parzen Window for Probability Distribution Function estimation, also showing good pattern recognition accuracies. The most important advantage of the PNN is that training is easy and instantaneous. Other advantages offered by the PNN are that only one parameter (so-called *smoothing* parameter) has to be set by the user, good accuracy can be achieved even with small samples, and the network is tolerant to erroneous data and operates completely in parallel without requiring feedback from the individual neurons to the inputs (Specht 1990). For further details, see Specht (1990, 1992) and Mao et al. (2000).

PNNs have been applied to diverse tasks such as pattern recognition (Kramer et al., 1995; Mao et al., 2000; Musavi et al., 1994; Romero et al., 1997; Sun et al., 1996), image processing (Quan et al., 2008; Song et al., 2007) and many others (Gerbec et al., 2005; Kim et al., 2008; Pande and Abdel-aty, 2008; Übeyli, 2010). Wu (2006) is the only author to have applied PNN to CCPR, combining it with Wavelet transforms to identify some abnormal pattern parameters.

2.3 REGRESSION AND TIME SERIES MODELS

2.3.1 LINEAR REGRESSION MODEL

The normal linear regression model may be written as follows:

$$Y_t = f\left(\beta, {X_t}^T\right) + e_t \qquad 2.1$$

where Y_t represents the expected value of the dependent variable *Y* at time *t,* X_t is a row or matrix of observations of the independent variable(s); β is a p-dimensional vector of parameters to be estimated by Maximum Likelihood (MLE), Least Square Error (LSE) or other estimation methods; e_t is a random error which, that in the simplest case, it is assumed to be independent (not autocorrelated), identically and normally distributed (NIID). Therefore, $e_t \sim Normal(\mu = 0, \sigma^2)$; where σ^2 variance of e_t and is assumed to be constant.

2.3.2 NONLINEAR REGRESSION MODEL (NLM)

In a more general case, the relationship between the independent variable (X) and the model parameters can be non-linear, i.e.

$$Y_t = f\left(\beta, {X_t}^T\right) + e_t \qquad 2.2$$

As in linear regression, X_t is a vector or matrix of independent variables and β is a vector of parameters to be estimated by Maximum Likelihood, Weighted Maximum Likelihood (WMLE), Least Squares or any other suitable estimation technique. e_t represents the white noise, $e_t \sim Normal(\mu = 0, \sigma^2)$ with σ^2 constant.

$f(\cdot)$ is a nonlinear function that relates the dependent variable to the model parameters. Two examples of such models are:

$$Y_t = \frac{\beta_1}{\beta_1 - \beta_2}\left(e^{-\beta_2 X_t} - e^{-\beta_1 X_t}\right) + e_t \qquad 2.3$$

$$Y_t = \beta_1 x_t + \beta_2 sin\left(\frac{2\pi x_t}{\beta_3}\right) + \beta_4 (-1)^{x_t} + e_t \qquad 2.4$$

where e_t is NIID, with $\mu = 0$.

2.3.3 TIME SERIES MODELS

Time series are a sequence of random variables indexed by the time, t. In time series models, it is studied the performance of random variables along the time. The most commonly used time series model is known as Autoregressive moving-average (ARMA).The order of the ARMA model is determined by the lag of each of the two components of the model. The first component corresponds to the autoregressive model (AR) part. In the AR, the current value is expressed as a finite linear aggregate of the previous observations of the process. Let ϕ_t the level of autocorrelation of the current observation and the t-lagged observation. The p-lagged AR model is expressed as:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t \qquad 2.5$$

Therefore, the first-lagged AR model (AR(1)) is as following:

$$Y_t = \mu + \phi_1 Y_{t-1} + e_t \qquad 2.6$$

where ϕ_1 is range in (-1, 1)

The second part of the model, the q-lagged MA(q), can be expressed as following:

$$Y_t = \mu + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \qquad 2.7$$

Thus, the first-lagged MA(1) is the following:

$$Y_t = \mu + e_t + \theta_1 e_{t-1} \qquad 2.8$$

In order to achieve a greater effectiveness, these two time series models are merged into one, the ARMA(p,q). The first-lagged ARMA(1,1) model being represented as follows:

$$Y_t = \mu + \phi_1 Y_{t-1} + \theta_1 e_{t-1} + e_t \qquad 2.9$$

2.3.4 NLM-ARMA

At this point, the random noise in NLM models, e_t, has been assumed to be NIID. But in a more general formulation, the assumption of independence (no-autocorrelation) cannot be correct; thus, the inherent noise is modelled by a time series model. Therefore, a NLM with autocorrelated noise can be represented as following:

$$Y_t = \mu + f\left(\beta, {X_t}^T\right) + N_t \qquad 2.10$$

The inherent noise model of this model, N_t, is defined as:

$$N_t = \mu + \phi_1 N_{t-1} + \theta_1 e_{t-1} + e_t \qquad 2.11$$

Thus, Yt can be modelled by a Dynamic Regression model (see Hyndman and Athanasopoulos (2014) and Pankratz (1991)). In this research, no possible feedback between Y_t and X_t, and X_t and N_t are assumed. Therefore, the Dynamic Regression model as presented in Pankratz (1991) is reduced to a Nonlinear regression with ARMA errors model (NLM-ARMA).

2.4 PNN AND SVM

2.4.1 PROBABILISTIC NEURAL NETWORK (PNN)

The PNN is a four layer feed-forward Neural network that can map any input pattern to any number of classes. Proposed by Specht (1990), this Neural network is based on the theory of Bayesian decision criteria and uses the Parzen window method for nonparametric estimation of Probability Density Function (PDF).

The four PNN layers are (See Figure 2.1):

- Input layer: This layer presents the pattern to be classified.

- Pattern layer: This estimates the likelihood φ_{km} of the pattern to be classified with each vector in the training data set separated into classes (1,…,k). This likelihood is calculated by applying the Normal Probability Distribution.

- Summation layer: This layer sums the likelihoods of the windows and computes the PDF for each class, $p(y_i|x_k)$. In this work, the kernel used for Parzen window PDF estimation was the Normal kernel. In the particular case of the Normal kernel, the multivariate estimates can be expressed as:

$$\sum \varphi_{km} = p(y_i|x_k) = \frac{1}{(2\pi)^{n/2}\sigma^n\, m_k}\sum_{j=1}^{m_k} e^{\left[-\frac{\left(y_i-y_j^{(k)}\right)^T\left(y_i-y_j^{(k)}\right)}{2\sigma^2}\right]} \qquad 2.12$$

where

i,j = pattern number

m_k = total number of training patterns of class k

σ = “smoothing parameter”

$y_j^{(k)}$= j-th pattern of class k

y_i = i-th pattern to be classified

n = dimensionality of the feature space

k = k-th class

- Decision layer: This layer classifies the input vector into the class with the highest estimated probability in the previous layer.

The best value of σ, the smoothing parameter, is the one that yields the best pattern recognition accuracy.

The main disadvantage of the PNN stems from the fact that it requires one node or neuron in the pattern layer for each training pattern (Specht 1992). This was the reason for implementing the shape features described previously. The effect of such features was to reduce the dimensionality in the Pattern layer and accelerate pattern recognition without loss of information.

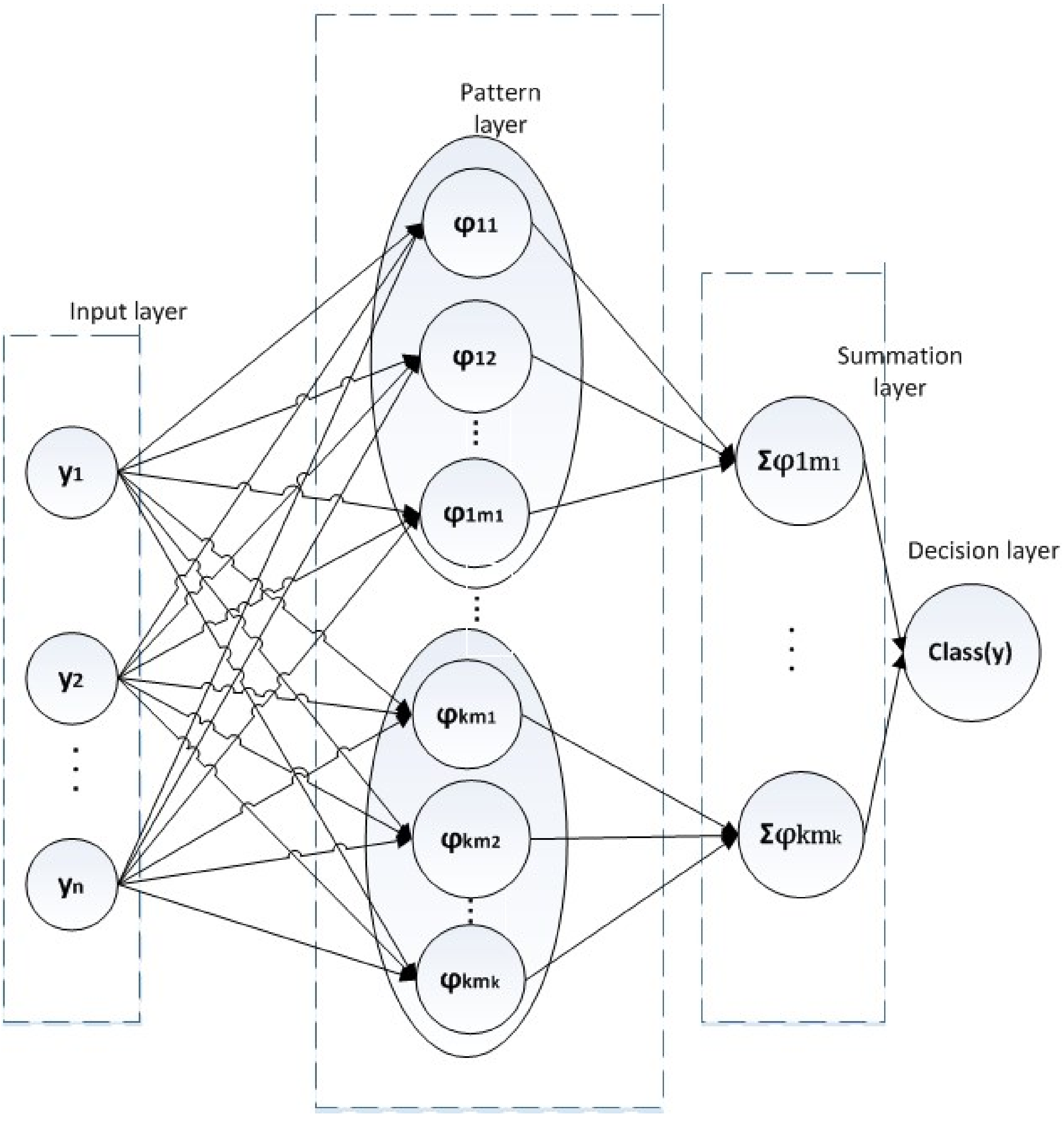

Figure 2.1: PNN topology

2.4.2 SUPPORT VECTOR MACHINE (SVM)

As a supervised learning algorithm, SVMs work with a given set of examples (x_i, y_i), *i=1, 2, 3,…,m,* where $y_i \in R^n$ and $x_i \in \{-1, 1\}$ is a previously defined class to which y_i belongs.

2.4.2.1 *Two-class SVM*

Separable case

The purpose of SVM is to find the optimal hyperplane in a high-dimensional transformation of the data that maximises the margin between two classes.

Thus, the function $f(y) = \boldsymbol{y} \cdot \boldsymbol{w} + b = 0$ divides the space into two, and the sign of this function indicates the side (group) to which the observation belongs. This function is known as the *Decision Boundary* of the classifier.

The vector $\boldsymbol{w}$ is known as the weight vector and $\boldsymbol{b}$ as the *bias.*

For the linearly separable case, this problem can be formulated as follows (Burges 1998):

$$\begin{aligned} \boldsymbol{y_i} \cdot \boldsymbol{w} + b \geq +1 \;\; for\, x_i = +1 \\ \boldsymbol{y_i} \cdot \boldsymbol{w} + b \leq -1 \;\; for\, x_i = -1 \end{aligned} \qquad 2.13$$

Combining the two previous inequalities gives:

$$x_i(\boldsymbol{y_i} \cdot \boldsymbol{w} \; + b) - 1 \geq 0 \qquad 2.14$$

SVM looks for the optimal hyperplane by solving the following optimisation problem:

$$\underset{w,b}{\text{minimise}} \quad \frac{1}{2}\|w\|^2 \qquad 2.15$$
$$subject\ to: \quad x_i(\boldsymbol{y_i} \cdot \boldsymbol{w} + b) - 1 \geq 0 \quad i = 1, 2, 3, \ldots, n$$

Lagrange multipliers α_i $(\alpha_i \geq 0)$, $i = 1, 2, 3, \ldots, n$ are introduced, one for each inequality constraint in order to have the following Lagrangian:

$$L_P \equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i x_i(\boldsymbol{y_i} \cdot \boldsymbol{w} + b) + \sum_{i=1}^{n} \alpha_i \qquad 2.16$$

The saddle point can be obtained by minimising the Lagrange function (L_p) with respect to the weights ***(w)*** and the bias ***(b)*** and maximising L_p and with respect to the dual variables, α_i (further details in Burges (1998)). By differentiating 2.16 with respect to *w* and *b*, the following equations are obtained:

$$\frac{\partial}{\partial w} L_P = 0\,, \qquad w = \sum_{i=1}^{n} \alpha_i x_i y_i \qquad 2.17$$

$$\frac{\partial}{\partial d} L_P = 0\,, \qquad \sum_{i=1}^{n} \alpha_i x_i = 0 \qquad 2.18$$

Substituting 2.17 and 2.18 into 2.16, the following dual formulation L_D is yield:

$$\underset{\alpha}{\text{maximise}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} x_i x_j \alpha_i \alpha_j \boldsymbol{y_i} \cdot \boldsymbol{y_j} \qquad 2.19$$
$$subject\ to: \sum_{i=1}^{n} x_i \alpha_i = 0, \ \ \alpha_i \geq \ 0$$

Thus, to find the optimal hyperplane has become in a quadratic optimisation problem. This optimisation problem can be solved by using standard optimisation program.

Non-separable case

In real problems, data are often not linearly separable, so, s*lack variables* ε_i are introduced to deal with margin errors $(0 \leq \varepsilon_i \leq 1)$ and misclassification problems $(\varepsilon_i > 1)$. Then, the constraints become:

$$\begin{aligned} \boldsymbol{y_i} \cdot \boldsymbol{w} + b &\geq +1 - \varepsilon_i \ \ for\, x_i = +1 \\ \boldsymbol{y_i} \cdot \boldsymbol{w} + b &\leq -1 + \varepsilon_i \ \ for\, x_i = -1 \end{aligned} \qquad 2.20$$

Since a bound of misclassification penalties the $\sum_i \varepsilon_i$, maximising the margin in the hyperplane becomes minimisation of $\frac{1}{2}\|w\|^2$ augmented with $C\sum_i \varepsilon_i$, where *C* is a parameter to be chosen by the user and is a penalty parameter for the *slack variables.* Thus, the optimisation problem is:

$$\begin{aligned} &\underset{w,b}{\text{minimise}} \quad \frac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^{n} \varepsilon_i \\ &subject\ to\text{:} \quad x_i(\boldsymbol{y_i} \cdot \boldsymbol{w} + b) \geq 1 - \varepsilon_i \quad \varepsilon_i \geq 0 \end{aligned} \qquad 2.21$$

This formulation is called *soft-margin SVM* and was proposed by Cortes and Vapnik (1995).

Therefore, by again introducing positive Lagrange multipliers, the following *dual* formulation is obtained:

$$\underset{\alpha}{\text{maximise}} \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} x_i x_j \alpha_i \alpha_j \boldsymbol{y_i} \cdot \boldsymbol{y_j} \qquad 2.22$$

$$subject\ to\colon \textstyle\sum_{i=1}^{n} x_i \alpha_i = 0,\ \ 0 \le \alpha_i \le C$$

2.4.2.2 *Multiclass SVM*

As seen above, SVMs are inherently two-class classifiers, so a *one-against-one* voting process is performed for multiclass classification. With *k* classes, *k(k-1)/2* binary classifiers are trained, giving a vote to the class into which the pattern is classified by each pairwise binary SVM, and selecting as the *final class* the one with most votes. In other words, let the decision function for class *i* against class *j*, with the maximum margin (Abe 2003), be:

$$D_{ij} = w_{ij}\, y + b_{ij} \qquad 2.23$$

where $D_{ij} = -D_{ji}$.

For the input vector *y*:

$$D_i(y) = \sum_{j \neq i}^{k} sign\left(D_{ij}(y)\right) \qquad 2.24$$

y is classified into the class:

$$\arg \max_{i=1,..,k} D_i(y) \qquad 2.25$$

This one-against-one classification has shown satisfactory results in various classification problems such as handwriting recognition (Milgram et al., 2006), object recognition (Hsu and

Lin, 2002) and image classification (Chapelle et al. 1999). Therefore, this method was chosen to deal with multiclass classification problem in this work.

2.4.2.3 *Kernels in SVM, nonlinear case*

As previously seen, the *Decision Boundary Function* is a linear function. In order to generalise the idea of this function, a nonlinear mapping function is presented: Kernels.

Kernels are used in SVMs as a function ϕ mapping the input data into a high-dimensional feature space using an inner product $[\phi(y), \phi(y')]$ and are represented as follows:

$$k(y_i, y_j) = [\phi(y_i), \phi(y_j)] \qquad 2.26$$

Some popular kernels are:

- Polynomial kernel (POLY):

$$k(y_i, y_j) = (\alpha y_i y_j + c)^d \qquad 2.27$$

where α, *d* and *c* are parameters to be adjusted by the user.

- Radial Basis Function kernel (RBF):

$$k(y_i, y_j) = exp\left(-\sigma \|y_i - y_j\|^2\right) \qquad 2.28$$

where σ is a parameter to be adjusted by the user.

- Hyperbolic tangent kernel (TANH):

$$k(y_i, y_j) = tanh(\alpha y_i y_j + c) \qquad 2.29$$

where α and are *c* are parameters to be set by the user.

- Laplace kernel (LAPLA):

$$k(y_i, y_j) = exp(-\sigma\|y_i - y_j\|) \qquad 2.30$$

where σ is a parameter to be set by the user.

- Bessel kernel (BESSEL):

$$k(y_i, y_j) = Bessel^{n}_{(v+1)}\left(\sigma\|y_i - y_j\|^2\right) \qquad 2.31$$

where ν and σ are free parameters to be adjusted by the user.

Including the kernel function in the *Decision Boundary Function:*

$$f(x) = \boldsymbol{k(y_i, y_j)} \cdot \boldsymbol{w} + b = 0 \qquad 2.32$$

gives the following *dual* formulation:

$$\underset{\alpha}{\text{maximise}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} x_i x_j \alpha_i \alpha_j k(y_i, y_j)$$

$$subject\ to\text{: } \sum_{i=1}^{n} x_i \alpha_i = 0,\ \ 0 \leq \alpha_i \leq C \qquad 2.33$$

The new formulation contains more free parameters to be determined by the user, the *Cost Parameter (C)* of the slack variables and the kernel parameters.

2.5 FEATURE EXTRACTION TECHNIQUE

In pattern recognition problems, it is common to deal with large amounts of data. Such data is used as input to train the ML algorithms in order to identify the pattern under study. Most of the ML algorithms are sensitive to the dimensionality of the vector used as inputs, causing large processing times when the input vector is of high dimensions. A challenge in recent research has been how to reduce the dimension of the input data without loss of information.

Gauri (2010) proposed seven shape features attending to the drawbacks found in the statistical and shape features proposed by Pham and Wani (1997). They were:

1) Sign of slope of the least square (LS) line representing the overall pattern (AB): AB represents the categorical variable correspondent to slope of the LS line fitted to the data and is represented by the following mathematical expression:

$$AB = \begin{cases} 1, & \sum_{i=1}^{n} y_i(t_i - \bar{t}) \Big/ \sum_{i=1}^{n}(t_i - \bar{t})^2 > 0 \\ 0, & \sum_{i=1}^{n} y_i(t_i - \bar{t}) \Big/ \sum_{i=1}^{n}(t_i - \bar{t})^2 < 0 \end{cases} \qquad 2.34$$

where t_i represents the time when the quality characteristic was taken $(i = 1,2, \dots n)$.

So, creating a categorical variable that indicated “1” when the magnitude of AB is greater than zero and “0” otherwise.

This feature can aid to discriminate between UT versus DT and US versus DS.

2) Area between the overall pattern and mean line per interval in terms of square of standard deviation (SD) (ACLPI):

ACL is the area between the pattern and the pattern and the mean line, it can be computed by summing the areas of the triangles and trapeziums that are formed by the mean line and overall pattern using simple algorithm. For further details regarding how to estimate this feature, refer to Gauri (2010).

Thus, ACLPI can be computed by means of the following equation:

$$ACLPI = [ACL/(N-1)]/SD^2\,; \quad SD = \left[\sum_{i=1}^{n}(y_i - \bar{y})^2 \Big/ (n-1)\right]^2 \qquad 2.35$$

This feature can discriminate the SYS pattern from the others.

3) Proportion of the sum of the numbers of crossovers of the mean line and the LS line (PSMLSC):

$$PSMLSC = \sum_{i=1}^{n-1}(o_i - o_i') \Big/ 2(n-1) \qquad 2.36$$

where $o_i = \begin{cases} 1, & (y_i - \bar{y})(y_{i+1} - \bar{y}) < 0 \\ 0, & otherwise \end{cases}$ and $o_i' = \begin{cases} 1, & (y_i - y_i')(y_{i+1} - y_{i+1}') < 0 \\ 0, & otherwise \end{cases}$

$\bar{y}$ is the arithmetic mean of the *n* observations, and y_i' is the Least Square prediction for time *i*.

The magnitude of PSMLSC is the lesser for CYC patterns and the maximum for SYS patterns.

***Features based on windows segmentation:**

For the case of the following features, the Control Chart is divided in four windows of n/4 = 15 observations each.

The midpoint of each segment is given by:

$$\left[\left\{\sum_{i=k}^{k+(n/4)-1} t_i/(n/4)\right\},\left\{\sum_{i=k}^{k+(n/4)-1} y_i/(n/4)\right\}\right] \qquad 2.37$$

where, $k = 1, \left(\frac{n}{4}+1\right), \left(\frac{2n}{4}+1\right), \left(\frac{3n}{4}+1\right)$ for the first, second, third and fourth segments, respectively. A combination of two midpoints can be obtained in $C_2^4 = 6$ ways implying that six straight lines can be drawn passing through the midpoints of these four segments. Similarly, six subsets of n/2 data points can be formed taking a combination of two segments in six ways. So six LS lines can also be fitted to six subsets of n/2 data points.

4) Absolute average slope of the straight lines passing through six pairwise combinations of midpoints of four segments (AASL):

$$AASL = \left|\sum_{j,k} s_{jk}/6\right|;\ (j = 1,2,3; k = 2,3,4; j < k) \qquad 2.38$$

The magnitude of this feature is lesser for NORM, CYC and SYS, and higher for patterns that can depict a slope.

5) Range of slopes of straight lines passing through six pair-wise combinations of midpoints of four segments (SRANGE):

$$SRANGE = maximum(s_{jk}) - minimum(s_{jk})$$
$$(j = 1,2,3; k = 2,3,4; j < k) \qquad 2.39$$

The magnitude of SRANGE will be lesser for NOR and SYS patterns.

6) Ratio of mean sum of squares of errors (MSE) of the LS line fitted to overall data and average MSE of the LS lines fitted to six subsets of N/2 data points (REAE):

$$REAE = MSE \Big/ \left[\sum_{j,k} MSE_{jk}/6\right];\ (j = 1,2,3; k = 2,3,4; j < k) \qquad 2.40$$

where MSE_{jk} is the mean sum of squares of errors of the *Least Square line* fitted to the observations in j^{th} and k^{th} segments.

The magnitude of this feature is the highest for Shift patterns and the lesser for CYC and SYS patterns.

7) Absolute slope difference between the LS line representing the overall pattern and the line segments representing the patterns within the two segments (ABDPE).

$$ABDPE = \left| B - \left(\sum_{j=1}^{2} B_j/2 \right) \right| \quad 2.41$$

where B_j is the slope of the Least Square line fitted from the j_{th} segment, and B represents the Least Square line fitted to overall pattern.

The magnitude of this feature is the highest for CYC patterns, intermediate for SYS and lesser for the remaining patterns.

Gauri (2010) noted that the above shape features had low correlation and improved the recognition stability. Furthermore, these features are independent of the scale of the data. Only one parameter is needed to be set by the user, namely, the number of window segments.

Therefore, these features were extracted from the generated patterns in order to reduce the dimensionality and the computational effort for the pattern recognition.

2.6 BEES ALGORITHM

A drawback of ML algorithms is the lack of rules for determining the values of the free parameters. In this research, an efficient optimisation algorithm, the BA was used for tuning the free parameters of the ML algorithms in order to minimise the pattern misclassification rate during the training and testing stages. The algorithm was chosen due to its simplicity and proven search and optimisation capabilities (Pham and Castellani, 2014).

2.6.1 HONEYBEES BEHAVIOUR IN NATURE

Honeybees can exploit a large number of flower patches as food sources by extending their foraging over more than 10 km and in multiple directions (Frisch, 1971; Shuttleworth and Johnson, 2010). Scout bees are those in charge of exploring the first patches and collecting nectar or pollen then returning to the hive to deposit it. The collected nectar or pollen is evaluated according to a certain quality threshold (i.e. sugar content) and then the scout bee goes to the “dance floor” and performs a “waggle dance”. This dance transmits information to the hive about the patch such as: the direction in which it will be found, its distance from the hive and its quality rating (Frisch, 1971; Camazine et al., 2001). This information helps the colony to send bees to flower the patches discovered by the scout bee, without using guides or maps. Each individual’s knowledge of the outside environment is gleaned solely from the waggle dance. Thus, more bees are sent to more promising patches, and fewer follower bees (or none) are sent to less promising patches.

2.6.2 BEES ALGORITHM OPTIMISATION

The BA has both local and global search capability utilising exploitation and exploration strategies, respectively (Yuce et. al., 2013). Without loss of generality, it will be assumed henceforth that the optimisation problem entails the maximisation of a given performance index. The solutions are encoded as n-dimensional vectors of decision variables, where each variable is bound within a pre-defined interval (Pham et al., 2012).

The BA is a population-based search algorithm that mimics the food foraging of swarms of honey bees. It was developed by Pham et al. (2006). At the beginning of the search, a population of *N* scout bees is generated and designated for random global search and their fitness is assessed. The *N* patches are ranked according to their fitness and the m fittest patches are selected for neighbourhood search. From these m best patches, e sites (called *elite* sites) with the best fitness and n_e scout bees are called for more detailed exploration for neighbourhood search in these patches. For the remaining m-*e* patches, neighbourhood search is also performed, this time with n_b bees carrying out local exploitation; n_b is taken to be less than n_e, i.e., this search is less intense.

The optimisation process can be described using the following pseudocode (Pham et al., 2012):

```
1) Initialise N random solutions.
2) Iterate until a stopping criterion is met
       o  Evaluate the fitness of the population.
       o  Sort the solutions according to their fitness.
       *waggle dance (recruitment)
       o  Select the highest-ranked $m$ solutions for neighbourhood search.
       o  Assign $x = n_e$ foragers to each of the $e \leq m$ top-ranking elite solutions.
       o  Assign $x = n_b \leq n_e$ foragers to each of the remaining m-e selected solutions.
       * local search
       o  For each of the $m$ selected solutions:
            ▪ Create x new solutions by perturbing the selected solution randomly or
              otherwise.
            ▪ Evaluate the new solutions.
            ▪ Retain the best solution amongst the selected and new solutions.
            ▪ Adjust the neighbourhood of the retained solution.
      o Re-initialise sites where search has stagnated for a given number of iterations.
      * global search
      o Repeat N-m times
            ▪ Generate randomly a new solution.
            ▪ Evaluate the new solution.
Form the new population (m solutions from local search, N-m from global search).
```

2.7 SUMMARY

In this chapter, the theoretical background and literature review were presented. Firstly, relevant works were reviewed and gaps in the knowledge were highlighted, some of them to be filled in this work. The theoretical background concerning the modelling of control chart data was given; Nonlinear, time series and dynamic models were explained. Finally, the other elements needed for the development of CCPR systems for NIID processes were presented, namely, ML algorithms, the BA and the shape features used as data input to the ML algorithm.

3 CCPR FOR NIID PROCESSES

3.1 PRELIMINARIES

This chapter presents a CCPR system for NIID processes that comprises two main stages. In the first stage, a new PGS fulfilling all the desirable conditions described in the Introduction chapter is employed. This proposed PGS randomises all the parameters and fully categorises the pattern based on the statistical significance of the parameter related to the pattern.

The second stage of the CCPR system uses two ML algorithms, SVM and PNN. These algorithms are based on statistical learning theory. The main difference in their learning methods relates to their risk minimisation (Gunn, 1998). In the case of SVM, structural risk minimisation is used to minimise an upper bound based on the expected risk, whereas in PNN, traditional empirical risk minimisation is adopted to minimise the error during training (Du et al., 2013). Another advantage of SVMs and PNNs is the number or free parameters to be chosen, being only one in the case of PNNs, and at most four for SVMs. The BA was adopted to find the best set of free parameters.

The CCPR system trained using the best arrangement of factors was utilised to identify abnormal patterns in data from a real example, namely, the flow width of the resist of a hard-bake process (Montgomery, 2009; Ross, 2014).

The remainder of the chapter is organised as follows. In section 3.2, the proposed PGS and the training of the CCPR are presented. The recognition accuracies of all the possible combinations of input factors are presented in section 3.3. In that section, the recognition accuracies are analysed and the best arrangement of input factors are given. Finally, section 3.4 reports the application of the CCPR trained using the best arrangement of factors to identify patterns in the resist flow width example.

3.2 PROPOSED CCPR SYSTEM

In this section, it is presented a PGS scheme that fulfils all the desirable conditions mentioned in chapter 1. This scheme consists of three steps: initial generation of patterns, most likely break point detection and final pattern classification. Secondly, the analysis of the results is divided into two parts: results of the PGS and CCPR accuracies to determine the best arrangement of input factors.

Furthermore, the CCPR that fulfilled the highest accuracies was used to analyse a real data example, in this case, the flow width in microns of the resist of a hard-bake process.

3.2.1 PATTERN GENERATION SCHEME

3.2.1.1 *Initial pattern generation*

Patterns are generated with only one possible change point in the time window examined; e.g. only one shift pattern can occur in the time window examined.

Firstly, a random vector, N_t, normally distributed with zero mean and variance σ_N^2 is generated by applying the pseudorandom number generator proposed by Matsumoto and Nishimura (1998).

$$N_t \sim Normal\ (\mu,\ \sigma_N^2\ ) \qquad 3.1$$

The following mathematical expressions were used initially to generate the patterns to be detected:

- Normal Pattern (NORM):

$$D_t = N_t \qquad 3.2$$

where D_t represents the pattern magnitude sampled at time t, μ is the mean value of the process, fixed to zero, and N_t is a normally distributed variable with mean equal to zero and variance σ_N equal to one, and represents the inherent noise in the process.

- Upward/Downward Trend (UT/DT):

$$D_t = \pm\beta_1 t \qquad 3.3$$

where β_1 is the Trend slope.

- Upward/Downward Shift (US/DS):

$$D_t = \ \pm\beta_2 d \qquad 3.4$$

where β_2 is the shift magnitude in the mean; *d=1* after the shift*,* and *d=0* before the shift, being the break point when the shift occurred randomly chosen between τ = 16 and τ = n-15.

- Cyclic (CYC):

$$D_t = \beta_3 \, sin\left(\frac{2\pi t}{\beta_4}\right) \qquad 3.5$$

where β_3 and β_4 are the amplitude and frequency of the CYC pattern respectively.

- Systematic (SYS):

$$D_t = \beta_5(-1)^t \qquad 3.6$$

where β_5 represent the systematic departure.

Therefore, the control chart pattern data is represented by:

$$Y_t = D_t + N_t \qquad 3.7$$

where Y_t represents the quality characteristic sampled at time *t*.

The values of the parameters in equations 3.2 to 3.6 are chosen randomly between the maxima and minima shown in Table 3.1.

Table 3.1: Parameters used during pattern generation (values given as a proportion of σ_N)

Pattern type	**Name**	**Parameter**	**Minimum**	**Maximum**
UT	Slope	β_1	0.001	0.30
DT	Slope	β_1	-0.30	-0.001
US / DS	Break point position	*d*	0	1
US	Magnitude of the mean shift	β_2	0.01	3.0
DS	Magnitude of the mean shift	β_2	-3.0	-0.01
CYC	Amplitude	β_3	0.01	3.0
CYC	Frequency	β_4	3	16
SYS	Systematic departure	β_5	0.01	3.0

Each of the parameters shown in Table 3.1 was designed to follow a uniform distribution in the specified range. The maximum and minimum values of the slope were determined to remain within the 6σ limits after 10 observations. The values for the period of the CYC patterns were set to have at least four recurrences in the observation window. The minimum for the other patterns was set near to zero, and the maximum to stay inside the 6σ limits in the observation window.

3.2.1.2 *Mean change classification*

Every pattern created in the previous step is employed in the proposed mean change classification.

As mentioned above, the proposed methodology for determining the type of mean change occurring in the control chart is based on the identification of potential change points and nested NLMs. A change point estimator focuses on finding the point in time where the process parameters have changed because of some assignable cause(s), i.e. it estimates the time when a change in the mean occurred. For example, consider a normal process where $Y_i \sim N(\mu_0, \sigma^2)$, $i = 1, 2, 3, \dots \tau$ and $Y_i \sim N(\mu_1, \sigma^2)$, $i = \tau + 1, \tau + 2, \dots, n$. That is, the process follows a normal distribution with mean μ_0 and variance σ^2, until the change point, τ. Following the change point τ, one parameter of the process has changed (from $\mu_o\ to\ \mu_1$). The aim is to estimate τ and the difference between μ_0 and μ_1, τ being the time when the mean changed Upward/Downward Shifts are considered the only patterns where a change point is detected.

The proposed mean change point classification methodology consists of two stages: identifying the most likely change point and evaluating that change point by means of an F-test for nested models.

- *Most likely change point fitting all the possible NLMs.*

Considering the number of parameters to be estimated and the degrees of freedom for the significance tests, a minimum sample of *15* is desirable to have a good estimation of regression parameters. Therefore, all possible piecewise regression models are fitted assuming change points at: $\tau = 16, 17, \dots, (n - 15)$. The Bayesian Information Criterion (McQuarrie and Tsai, 1998) is extracted from each possible fitted model in order to determine which of the fitted models is the most likely to have a change point, with τ chosen to correspond to the least Bayesian Information Criterion value. The following is the NLM assuming the existence of a change point:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 d + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5(-1)^t + \varepsilon_t \qquad 3.8$$

where *d* and the parameters β_1 to β_5 are as defined in Table 3.1. β_0 and ε_t represent the intercept with the y-axis of the regression model and the random error at time t, respectively.

- *Nested models and model selection*

The model represented by equation 3.9 below corresponds to a NLM not assuming the existence of change points, i.e., only continuous change in mean is considered. It is observed that the model that does not take account of the existence of change points is fully contained in equation 3.8 which relates to a model with the most likely change point. Thus, the mean change categorisation problem becomes a selection between two nested NLMs, considering the model fitted under the supposition of no change point as the reduced model and the model fitted with the most likely change point as the full model.

$$Y_t = \beta_0 + \beta_1 t + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5(-1)^t + \varepsilon_t \qquad 3.9$$

As it is unknown if there is a break point and if its magnitude is statistically significant, the selection of which model best fits the patterns is a model selection problem with nested models, raising the following two hypotheses:

$$H_0: There\ are\ no\ break\ points\ (the\ reduced\ model\ fits\ better)$$
$$H_1: A\ break\ point\ is\ detected\ (the\ full\ model\ fits\ better)$$

An *F-test* for nested models is used (Draper, Smith and Pownell, 1966; Rawlings, Pantula and Dickey, 2006) in order to determine which hypothesis to reject, i.e.

$$F_{v_2}^{v_1} = \left(SSE_{full} - SSE_{reduced}\right) / \left(SSE_{full} / (n - k - 1)\right) \qquad 3.10$$

where k is the number of parameters of the full model excluding the intercept, and F belongs to an F-distribution with one degree of freedom in the numerator $(v_1 = 1)$ and n - k - *1* degrees of freedom in the denominator $(v_2 = n - 6)$. Three significance levels of *α=0.01, 0.02* and *0.03* are established in this research to determine whether the control chart under study has a break point or not.

3.2.1.3 *Pattern classification*

If it is decided that the studied pattern has a change point, a model of type (1) (represented by equation 3.8) is fitted to the data; otherwise, a model of type (2) (represented by equation 3.9) is adopted.

The statistical significance of the parameters βs related of the model fitted in the previous step determines the class to which the pattern belongs. In this work, three significance levels were chosen, *α = 0.01*, *0.02* and *0.03*. The pattern parameter whose p-value is less than the significance level will determine the class of the pattern. For further information about statistical testing significance in NLMs, refer to Gallant (1987).

Figure 3.1 shows a flowchart that summarises the scheme proposed for the generation of patterns.

Once the CCPR system has been trained, it will be able to recognise a broad range of patterns. If the system identifies a pattern as abnormal, the parameter(s) related to this pattern can be statistically estimated by applying equation 3.8 or 3.9.

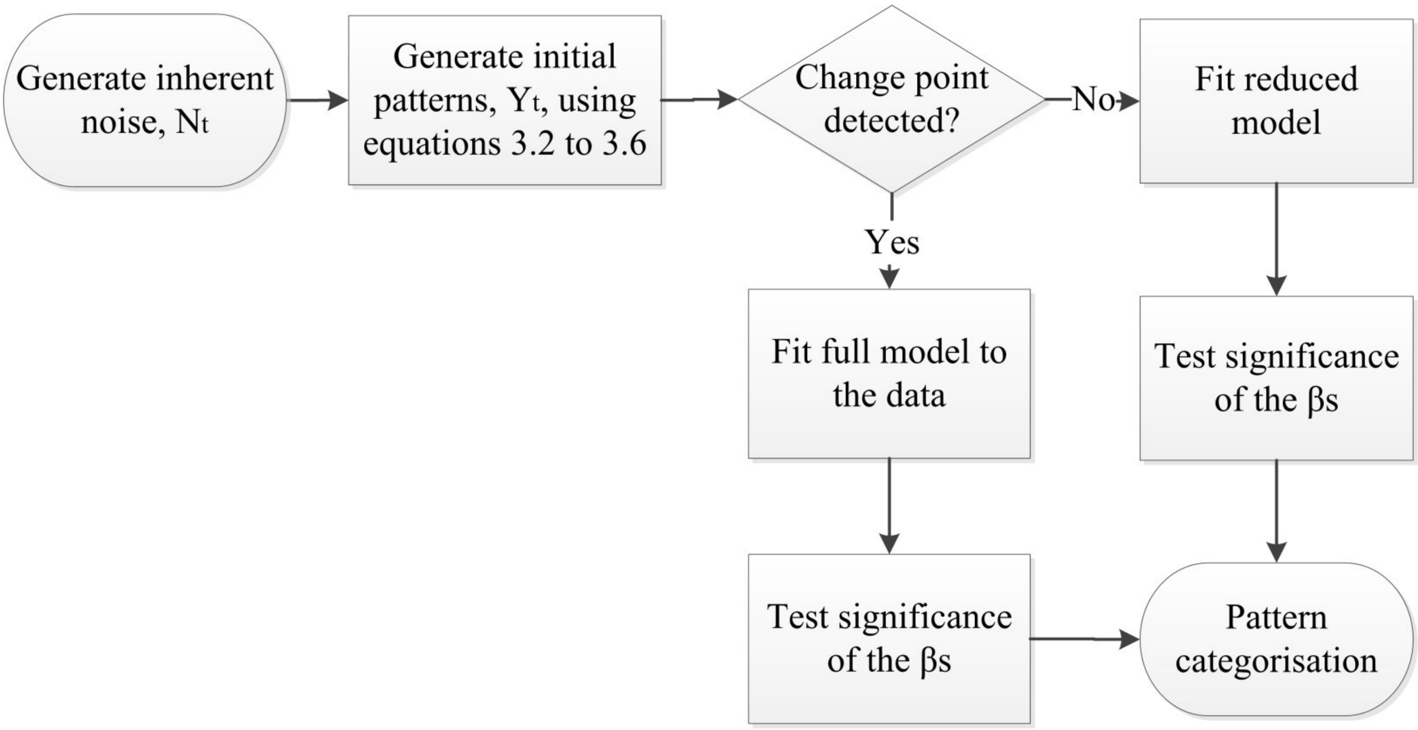
Generate inherent noise, Nt
Generate initial patterns, Yt, using equations 3.2 to 3.6
Change point detected?
No
Fit reduced model
Yes
Fit full model to the data
Test significance of the βs
Test significance of the βs
Pattern categorisation

Figure 3.1: Flowchart of the proposed pattern generation scheme

3.2.2 ILLUSTRATIVE EXAMPLE OF PATTERN CLASSIFICATION

Seven patterns were generated using equations 3.2 to 3.6 sampled at times t=1, 2, …, n=60, Figure 3.2 depicts these patterns.

The third column of Table 3.2 gives the parameter values used for the initial pattern generation. The fourth column shows which model fits each pattern better and the p-values obtained from the F-test. To obtain the fourth column, it was necessary to fit two models to each pattern. Then, using the SSEs from these, the F-value and its respective p-value were obtained to determine which model fits better. The fifth column lists the p-values of the significant term of the model that was determined to fit better in the previous step. Finally, the last column gives the final class of the pattern according to the proposed PGS. The significance level chosen for this example is α=0.01.

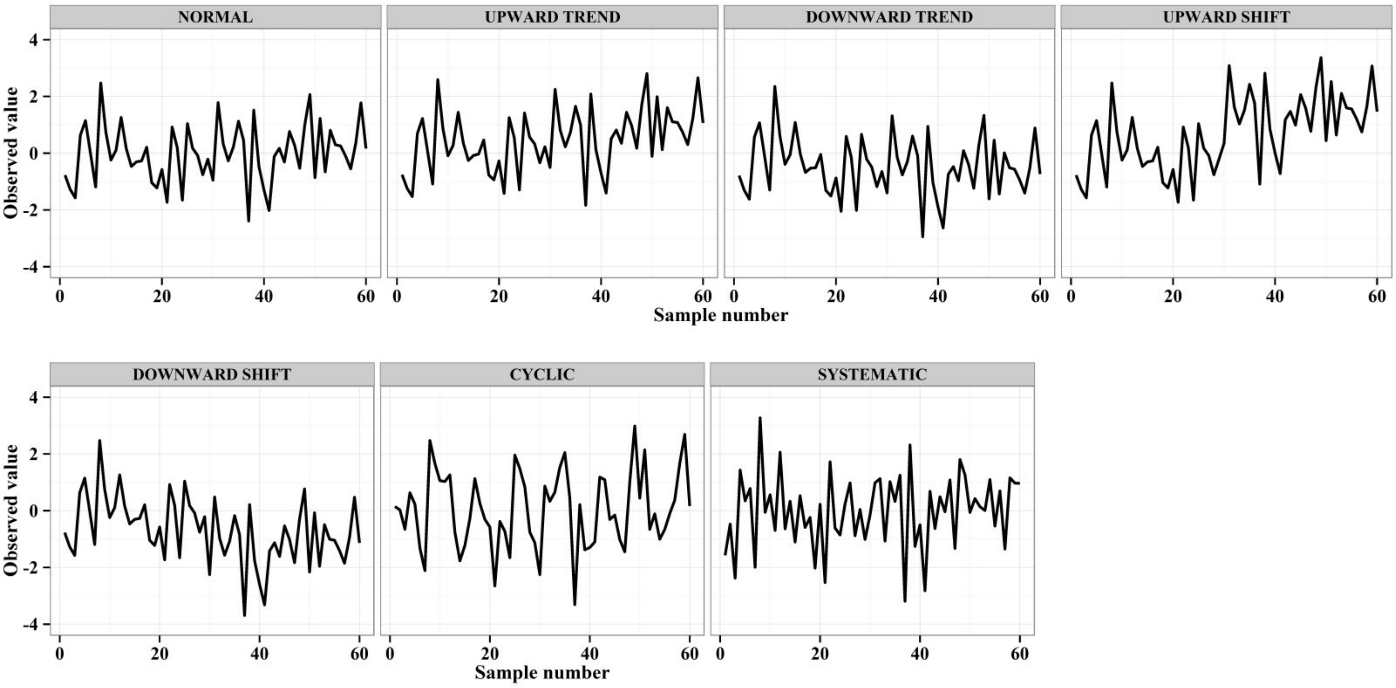

Figure 3.2: Seven generated patterns

Table 3.2: Analysis of the seven patterns as illustrative examples

Initial pattern class	Parameter	Initial magnitude	Best model fitted (p-value of F-test)	Estimated parameter (p-value)	Final pattern class
NORM	-	-	Reduced (0.0477)	-	NORM
UT	Slope	0.015σ	Reduced (0.0467)	0.0219 (0.0056)	UT
DT	Slope	0.015σ	Reduced (0.0407)	-0.0080 (0.0296)	NORM
US	Shift magnitude (τ)	1.3σ (τ=30)	Full (0.0098, τ =29)	1.4098 (0.0098)	US
DS	Shift magnitude (τ)	1.3σ (τ=30)	Reduced (0.0243)	-0.0253 (0.0024)*	DT
CYC	Amplitude (frequency)	1.3σ (8)	Reduced (0.0975)	1.2421 (0.0000)	CYC
SYS	Systematic departure	0.8σ	Reduced (0.0940)	0.7830 (0.0000)	SYS

Regarding the Trend patterns, it was observed that DT was reclassified as NORM since none of the pattern parameters of the model that fitted it better was statistically significant, this being opposite to the case of the UT pattern where the estimated slope of 0.0219 was statistically significant. For the US pattern, it was observed that the Full model fitted it better and the most likely break point of magnitude 1.4098 was observed at τ =29.

For the case of DS, it is worth showing the operation of the PGS step by step. Firstly, a full model and a reduced model were fitted, giving the models shown in equations 3.11 and 3.12.

$$Y_t = -0.23 + 0.005t - 1.23d - 0.41 sin\left(\frac{2\pi t}{13}\right) - 0.02(-1)^t + \varepsilon_t \qquad 3.11$$

$$Y_t = 0.11 - 0.03t - 0.36\, sin\left(\frac{2\pi t}{13}\right) - 0.03(-1)^t + \varepsilon_t \qquad 3.12$$

Nesting these two models, the following F-test value was obtained: $\mathrm{F}_{54}^{1} = (61.183 - 55.652)/(55.652/54) = 5.3661$

As the p-value associated with the F-value was less than the established significance level (α = 0.01), it was determined that the reduced model fitted the pattern better; therefore, equation 3.12 was used for categorising the pattern. In this equation, the only statistically significant parameter was the one related to the Trend pattern. Therefore, this pattern was categorised as DS.

3.2.3 TRAINING OF THE RECOGNITION ALGORITHM

For the SVM training, four sets of 2800 patterns were generated, 400 for each pattern class; one of these sets was created not using the proposed PGS and the other three generated using the proposed PGS, setting the significance level to α=0.01, 0.02 and 0.03. In order to deal with nonlinear decision boundaries, five different kernels implemented in the *kernlab* (Karatzoglou et al., 2004) library of *R*-software (Team, 2014) were tested. These were the BESSEL, LAPLA, POLY, RBF and TANH kernels.

Five different sample sizes were considered for training PNNs. These were n = 60, 80, 100, 120 and 140 patterns of each type. Using these five sample sizes, four data sets were generated, one with conventionally produced patterns, and the other three sets using the PGS with the three aforementioned significance levels (α=0.01, 0.02 and 0.03).

To reduce the dimension of the input vectors for training the SVMs and PNNs, the shape features the shape features initially proposed by Pham and Wani (1997) and then improved by Gauri (2010) were adopted since they are independent of the scale and length of the data, reduce the training time significantly and increase the pattern recognition accuracy. The number of window segments was set to four.

The function *nls* implemented in *R-software* (R Core Development Team 2015)*,* was used to fit NLMs according to the proposed methodology.

A five-fold cross validation and hold-one-out validation for the SVM and the PNN, respectively, were employed for model validation, and the misclassification rate under these schemes was used as the cost function to be minimised during the training. The BA, proposed by Pham et al. (2006) was implemented to find the best sets of free parameters of the SVM and the PNN, the aforementioned cost function being the objective function value to be minimised. This algorithm was selected for its proven ability to determine globally optimal solutions to complex optimisation problems (Pham and Castellani, 2014). Table 3.3 shows the values adopted for the BA parameters. For a definition of those parameters, see Pham et al. (2006).

Table 3.3: Parameter values used in the BA

Parameter	Symbol	Value in PNN	Value in SVM
Initial population	*n*	30	30
Number of "best" sites	*m*	5	4
Number of "elite" sites	*e*	3	2
Patch size (Smoothing parameter σ in PNN)	*ngh*	0.01	-
Patch size (Cost parameter *C* in SVM)	*ngh-c*	-	0.5
Patch size (Kernel parameters in SVM)	*ngh-k*	-	0.02
Number of elite bees for the elite sites	*ne*	4	4
Number of bees for the remaining "best" points	*nb*	2	2

3.3 RESULTS

3.3.1 PATTERN GENERATION

To measure the performance of the proposed PGS at different significance levels, 10,000 patterns of each type were initially generated, being sampled at 60 equal time intervals $t_1, \dots, t_{60}$. As mentioned previously, three different significance levels were used, α= 0.01, 0.02 and 0.03.

Table 3.4 shows the allocation of these 70,000 patterns. They fall into three categories: retained in initial class, reclassified or discarded. It is was found that when the significance level used was α= 0.01, 65.68% of the patterns remained in the classes that were initially generated, 14.71% were reclassified and 19.61% were discarded due to two or more patterns produced by the PGS not being related to any of the classes in a statistically significant way. It is worth noting that 19.23% of the patterns initially generated as Normal were classified as CYC. This could be due to a possible periodic behaviour of the RNG. It can also be observed that around 15% of the patterns initially generated as Shift patterns were reclassified as Normal. This could be due to the small amounts of shift in these patterns. For a significance level of α=0.02, 54.26% of the patterns retained their initial classes, 13.41% were reclassified, and the remaining 32.33% were discarded. When the significance level was chosen to be α=0.03, 42.17% of the patterns were discarded, 12.40% were reclassified, and the other 45.43% remained in the same pattern class as initially generated. Using this significance level, it was found that 28.69% of the patterns initially created as Normal were reclassified. Reclassification rates for Normal patterns were higher at the three significance levels. It was also noted that the pattern with the lowest discarding percentage was the Normal pattern. As for the Shift patterns, it was found that the reclassification percentage decreased as the significance level increased. Therefore, as shown in Table 3.4, the proposed PGS was more efficient when α = 0.01, only 19.60% of the patterns were discarded.

3.3.2 RECOGNITION ACCURACIES

The four SVMs trained in 3.2.3 were put through 100 test runs. A test run consisted of applying 100 patterns of each type to the trained SVM. The pattern recognition accuracies obtained in the one hundred runs were compared against those achieved with a SVM trained using conventionally generated patterns.

Figure 3.3 shows the 95% CI for the mean accuracy of each kernel tested.

Table 3.5 shows the accuracies achieved by the four designs (No-PGS and the three α levels of the proposed PGS) disaggregated by pattern type. ANOVA of three factors with single, double and triple interactions was used to analyse the results. The three factors were: kernel type, four pattern generators and pattern type. It was observed from this ANOVA that using the proposed PGS significantly increased the mean accuracy by 6.90%. Also, from Table 3.5, the kernel that achieved the best accuracy (92.92%) was the LAPLA kernel.

Table 3.4: Pattern classification using the proposed PGS (%)

	Initial class						
α level \ Final class	**NORM**	**UT**	**DT**	**US**	**DS**	**CYC**	**SYS**
α = 0.01							
NORM	65.30	1.47	1.49	15.19	16.06	2.11	5.49
UT	0.79	72.32	0.00	8.73	0.03	0.04	0.01
DT	0.68	0.00	71.99	0.02	9.02	0.04	0.07
US	2.44	0.70	0.00	49.28	0.14	0.06	0.22
DS	2.15	0.00	0.77	0.12	49.39	0.04	0.20
CYC	19.23	1.06	0.99	5.45	5.44	80.29	1.65
SYS	0.72	0.02	0.00	0.17	0.18	0.01	71.22
Discarded	8.69	24.43	24.76	21.04	19.74	17.41	21.14
α = 0.02							
NORM	48.55	0.88	0.93	9.29	9.88	1.25	3.43
UT	0.84	57.65	0.00	6.11	0.04	0.03	0.07
DT	0.83	0.00	56.51	0.01	6.49	0.05	0.09
US	3.04	0.80	0.00	44.43	0.12	0.08	0.20
DS	2.94	0.00	0.96	0.17	44.71	0.05	0.33
CYC	25.90	1.26	1.08	6.81	6.41	69.66	1.85
SYS	1.14	0.02	0.00	0.23	0.22	0.00	58.34
Discarded	16.76	39.39	40.52	32.95	32.13	28.88	35.69
α = 0.03							
NORM	37.31	0.57	0.76	6.63	6.78	0.71	2.46
UT	0.88	46.69	0.00	4.71	0.02	0.03	0.06
DT	0.86	0.00	45.20	0.03	4.95	0.06	0.09
US	3.40	0.90	0.00	39.61	0.13	0.08	0.13
DS	3.31	0.00	0.97	0.18	40.19	0.05	0.25
CYC	28.69	1.28	1.13	6.59	6.38	60.67	1.91
SYS	1.39	0.02	0.00	0.18	0.22	0.00	48.37
Discarded	24.16	50.54	51.94	42.07	41.33	38.40	46.73

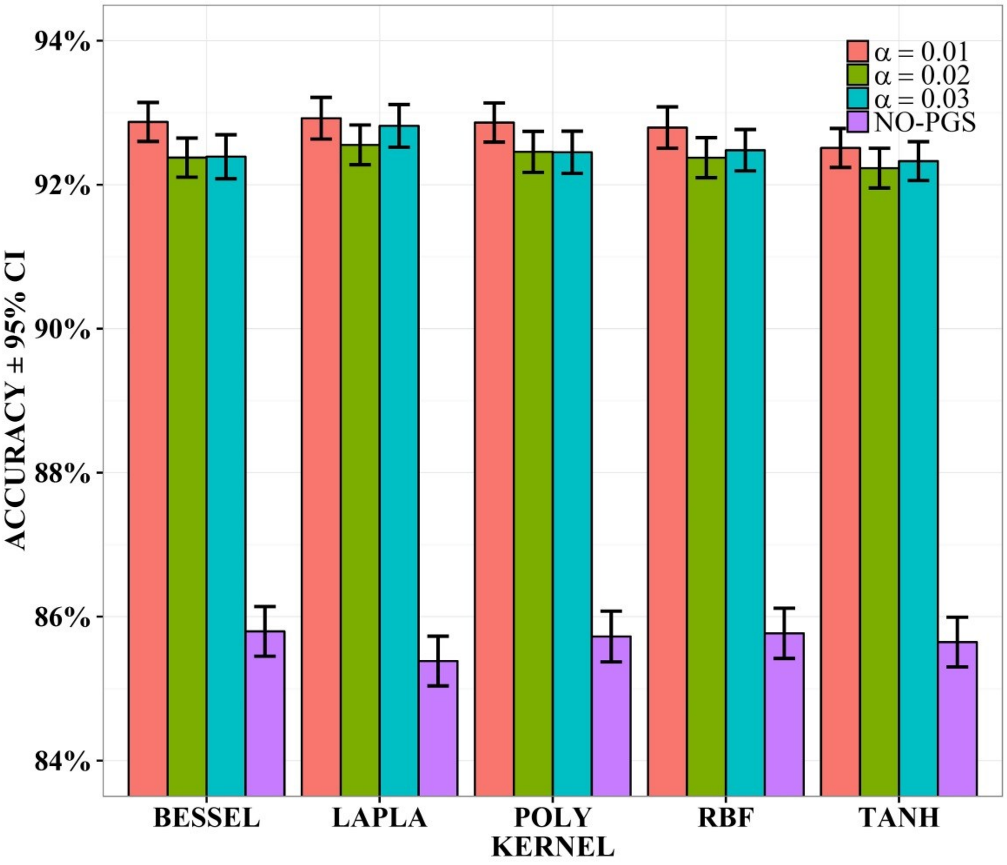

Figure 3.3: Accuracies achieved with five different kernels and three α levels

Table 3.5: Results for different types using five different kernels in the SVM for three different values of α (%)

Kernel \ Pattern	**Without the PGS**	**With PGS, α=0.01**	**With PGS, α=0.02**	**With PGS, α=0.03**
OVERALL	**85.66**	**92.79**	**92.40**	**92.49%**
BESSEL				
TOTAL	**85.65**	**92.51**	**92.23**	**92.33**
NORM	85.42	86.54	86.49	87.02
UT	82.51	93.53	93.32	93.81
DT	89.80	94.46	94.10	94.33
US	83.39	92.57	92.82	92.35
DS	85.57	92.28	90.93	91.05
CYC	82.99	93.43	92.80	92.61
SYS	89.85	94.76	95.15	95.12

Kernel \ Pattern	**Without the PGS**	**With PGS, α=0.01**	**With PGS, α=0.02**	**With PGS, α=0.03**
LAPLA				
TOTAL	**85.38**	**92.92**	**92.55**	**92.82**
NORM	84.72	86.43	86.81	86.06
UT	82.31	94.12	93.94	94.65
DT	88.97	95.26	95.23	94.92
US	85.68	92.54	92.67	93.77
DS	85.39	92.33	91.19	90.91
CYC	81.35	94.56	93.22	93.77
SYS	89.26	95.22	94.81	95.64
POLY				
TOTAL	**85.77**	**92.79**	**92.37**	**92.48**
NORM	85.29	86.43	86.50	86.19
UT	82.57	93.91	93.07	93.78
DT	89.76	95.27	94.28	94.81
US	85.74	92.35	93.63	93.27
DS	85.23	91.75	90.71	90.48
CYC	81.85	94.17	93.20	93.78
SYS	89.94	95.67	95.23	95.04
RBF				
TOTAL	**85.80**	**92.87**	**92.38**	**92.39**
NORM	85.46	87.43	87.09	85.90
UT	82.53	93.31	92.71	93.92
DT	89.74	95.44	95.01	95.08
US	85.50	91.50	92.64	93.03
DS	85.56	92.74	90.87	90.14
CYC	82.08	94.01	92.65	92.60
SYS	89.70	95.67	95.66	96.05
TANH				
TOTAL	**85.72**	**92.86**	**92.46**	**92.45**
NORM	85.42	86.78	86.44	85.92
UT	82.35	94.18	93.35	93.92
DT	89.96	94.74	94.79	94.75
US	85.42	92.62	93.47	93.64
DS	85.14	92.54	90.73	90.46
CYC	81.80	94.04	93.00	93.30
SYS	89.98	95.14	95.41	95.16

For the PNN, one hundred test runs were carried out, each involving 50 patterns of each type.

Figure 3.4 shows the 95% CI for the mean accuracy obtained using the proposed PGS as well as the results for a PNN trained conventionally.

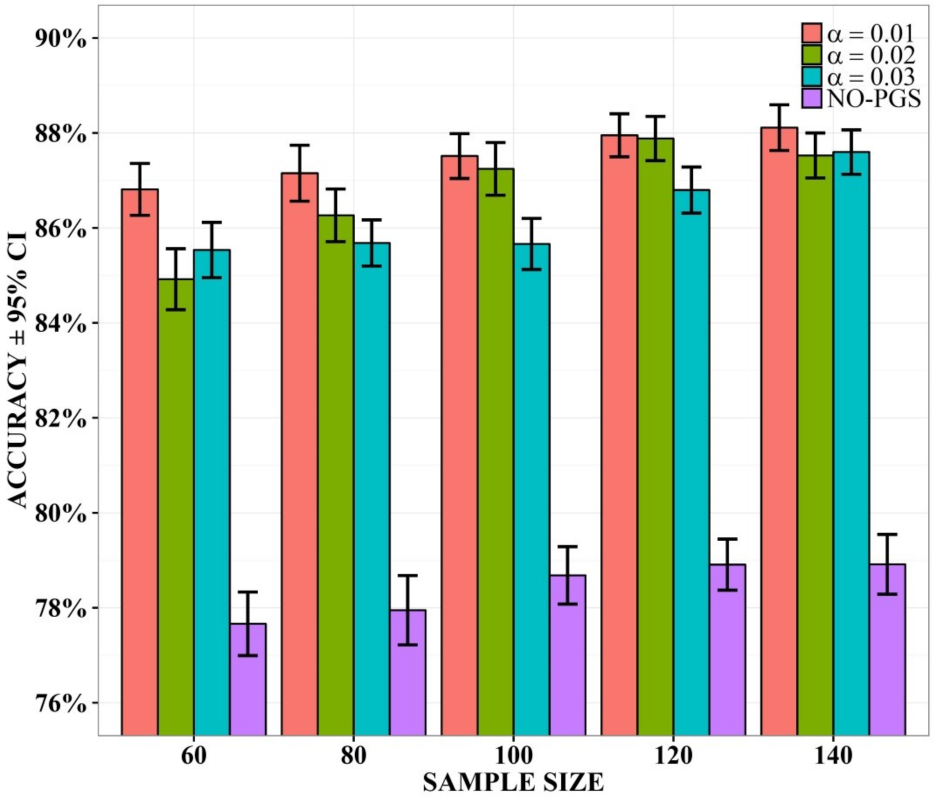

Figure 3.4: PNN trained with and without patterns created using the proposed PGS

Table 3.6 shows the mean accuracies achieved during the testing of the PNNs disaggregated by each of the four designs, the seven simple patterns and the five sample sizes. ANOVA of three factors with up to triple interactions was also employed to analyse the accuracies achieved by the PNNs. The three factors considered were: sample size, pattern generation design and pattern type. It was found that the mean accuracy was significantly increased when the PGS scheme was employed. Also, from Table 3.6, the highest recognition (88.11%) accuracy was achieved when the proposed PGS (α = 0.01) is used and the sample size was n=140.

Table 3.6: Results for different pattern types using five different sample sizes in the PNN (%)

Sample size\ Pattern	**Without the PGS**	**With PGS, α=0.01**	**With PGS, α=0.02**	**With PGS, α=0.03**
OVERALL	**78.42**	**87.51**	**86.77**	**86.25**
n=60				
TOTAL	**77.66**	**86.81**	**84.92**	**85.53**
NORM	62.48	74.34	68.34	72.92
UT	79.18	85.06	86.40	84.90
DT	82.90	88.64	89.88	87.44
US	75.74	91.42	83.94	93.50
DS	82.38	90.44	91.50	91.34
CYC	76.18	87.14	87.30	86.24
SYS	84.78	90.64	87.08	82.40
n=80				
TOTAL	**77.95**	**87.15**	**86.27**	**85.68**
NORM	64.84	71.76	73.82	75.14
UT	73.18	87.30	91.22	88.72
DT	86.90	87.88	84.94	88.48
US	71.74	94.24	90.84	86.30
DS	83.66	88.38	89.24	89.66
CYC	77.38	90.04	88.86	85.26
SYS	87.94	90.46	84.94	86.22
n=100				
TOTAL	**78.68**	**87.51**	**87.24**	**85.66**
NORM	68.76	77.44	73.54	73.82
UT	78.40	88.54	85.92	84.30
DT	83.68	86.44	91.36	88.18
US	71.94	91.24	89.08	83.82
DS	83.54	88.72	90.68	90.94
CYC	79.08	90.76	90.92	87.94
SYS	85.38	89.46	89.20	90.64
n=120				
TOTAL	**78.91**	**87.95**	**87.88**	**86.80**
NORM	71.72	78.24	77.46	77.14
UT	79.54	87.40	89.38	86.42
DT	80.96	89.80	89.72	89.76
US	73.72	90.00	89.56	88.28
DS	83.56	88.22	90.68	92.80
CYC	77.86	92.42	88.36	84.48
SYS	85.00	89.58	90.02	88.70

Sample size\ Pattern	**Without the PGS**	**With PGS, α=0.01**	**With PGS, α=0.02**	**With PGS, α=0.03**
n=140				
TOTAL	**78.91**	**88.11**	**87.53**	**87.60**
NORM	73.76	77.56	77.30	77.74
UT	76.74	88.16	87.56	86.88
DT	85.90	89.42	89.12	88.40
US	69.66	91.76	89.04	90.34
DS	84.10	90.04	90.42	90.82
CYC	75.52	92.06	87.46	88.08
SYS	86.72	87.78	91.78	90.92

As mentioned in the Introduction, it was also of interest in this study to assess the proposed scheme with two very different ML algorithms, namely, SVM and PNN. The performance of the proposed PGS was measured for the three aforementioned α levels. Figure 3.5 shows the mean and the 95% CI of the accuracies achieved. It can be seen that with both ML algorithms the accuracy was increased when the proposed PGS was used. These accuracies marginally increased when the α level was changed to 0.01 for both algorithms.

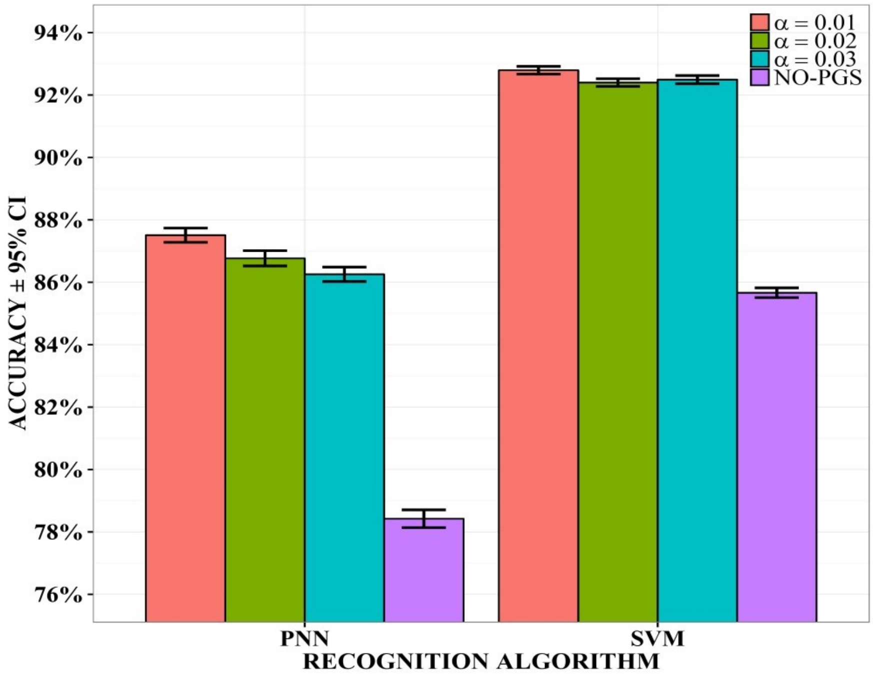

Figure 3.5: Accuracies achieved by PNN and SVM with different pattern generators

3.4 REAL DATA APPLICATION

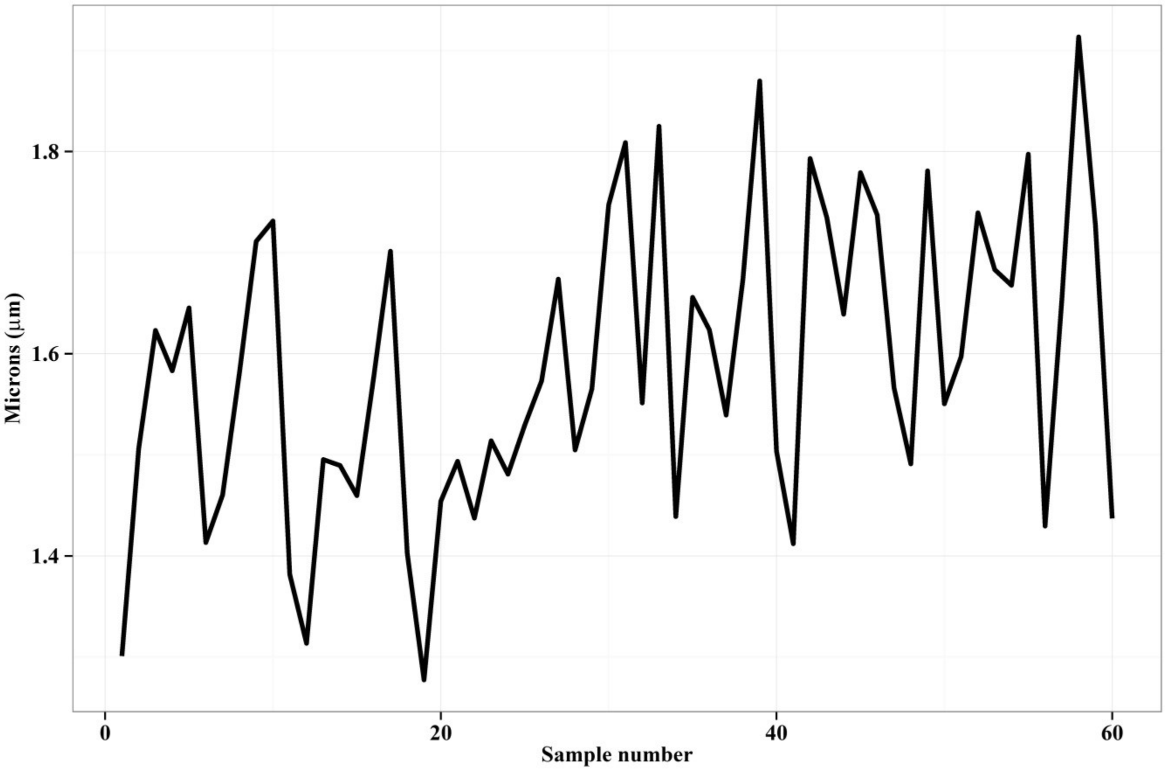

Figure 3.6: Flow width in microns of the resist of a hard-bake process

To demonstrate the ability of the proposed CCPR system to handle real data, the data taken from Montgomery (2009) and analysed in Ross (2014) were utilised (see Figure 3.6). These represent the flow width in microns of the resist of a hard-bake process (corresponding to observations 161 to 220). This data set was utilised by Ross (2014) to detect shifts in the mean width. He detected a mean shift after 25 observations.

The trained CCPR was applied to these data and a US pattern was identified. The proposed NLM was applied to the data, a p-value of 0.0118 was obtained from the F-test for nested models, therefore, a break-point at time $\tau = 25$ was observed and the full model fits better to the data.

Table 3.7 shows the ANOVA obtained from the NLM fitted to the data

Table 3.7:ANOVA obtained from the flow width of the resist of a hard-bake process

	Estimate	Std. Error	t value	p-value
Intercept	1.5005	0.0336	44.6313	0.0000
Slope (Trend)	-0.0003	0.0018	-0.1890	0.8508
Shift magnitude (Shift)	0.1625	0.0624	2.6070	0.0118
Amplitude (Cyclic)	0.0629	0.0425	1.4796	0.0695
Frequency (Cyclic)	7.1497	0.0835	85.6030	0.0000
Departure (Systematic)	-0.0238	0.0159	-1.4950	0.1407

It can be observed from Table 3.7 that the parameter related to the Shift patter was statistically significant and its magnitude is greater than zero, so the ANOVA confirms that an US pattern exists in the control chart.

3.5 SUMMARY

A CCPR for NIID processes was proposed in this chapter. This system consisted of two main stages. The first stage paid special attention to the right setting of class boundaries. For that, a PGS was proposed. That PGS fulfilled all the desirable conditions mentioned in the Introduction section and its effect on recognition accuracies was measured. In the second stage, the ML algorithms were trained using all the possible combinations of input factors in order to determine the combination yielding the highest accuracies. A popular optimisation algorithm named the BA was used to find the best combination of free parameters for the training of the ML algorithms.

The performance of the PGS was studied by analysing the numbers of patterns that remained in the same pattern class, were reclassified to another class or were discarded.

The highest recognition accuracy was achieved when the CCPR system was trained with patterns synthesised using the proposed PGS. Nearly the same accuracies were obtained for three alpha levels.

Finally, the CCPR system trained using the best arrangement of input factors was used to identify patterns in real data on the flow width of the resist of a hard-bake process. A US pattern was correctly identified when the aforementioned CCPR system was applied. Furthermore, by fitting the NLM models proposed in the PGS, it was possible to estimate a shift magnitude of 0.0118 at $\tau = 25$.

4 CCPR FOR AUTOCORRELATED PROCESSES

4.1 PRELIMINARIES

In the work reported here, three time series models were used to synthesise control chart patterns (and therefore the inherent noise). The proposed CCPR system took into account the possible combination of effects between the shape of the autocorrelated process and the abnormal patterns, i.e., some patterns could be masked or distorted by the intrinsic shape of the noise and the shape and the final chart may not resemble the original pattern.

The proposed CCPR consisted of two stages: the generation of patterns and the training of the ML algorithm. For the first stage, a new PGS scheme was employed. This PGS scheme was based on a dynamic regression model called NLM-ARMA. This model avoids the biasing effect of abnormal patterns on the common-cause parameters as the model parameters are estimated simultaneously. The final pattern categorisation in this stage was based on the statistical significance of the parameters related to each of the patterns. In the second stage, the CCPR system was trained using a wide variety of input factors in order to analyse the recognition accuracies and determine the arrangement of these factors yielding the highest accuracies.

Finally, the CCPR system that achieved the highest accuracy was used to detect patterns in data from a real example, namely, the monthly numbers of accidental deaths in the United States from January 1973 to December 1978. The example was used to demonstrate the ability of the proposed CCPR system to handle real rather than synthetic data.

The remainder of the chapter is organised as follows. Section 4.2 gives an example illustrating simultaneous parameters estimation and the biasing effect of an abnormal pattern on common-cause parameters. Section 4.3 presents the proposed CCPR system. The results are analysed in section 4.4. Section 4.5 reports on applying the CCPR system trained using the best arrangement of input factors to analyse the monthly numbers of accidental deaths in the United States.

4.2 INITIAL EXAMPLE

In the proposed PGS, unbiased estimation of common-cause parameters (autocorrelation coefficients, standard deviation and mean value) and pattern parameters is ensured. If the autocorrelation and pattern parameters are not estimated simultaneously, an abnormal pattern can badly affect their estimation, i.e., they can be biased due to the mixing of the effects of the two types of noise. For instance, Figure 4.1(a) shows a Normal pattern generated from an AR(1) process with autocorrelation coefficient ϕ=0.5. Figure 4.1(b) shows the same pattern added to a trend of slope $0.05\sigma_y$. Using equation 4.1 for estimating ϕ_1 from Figure 4.1(a) data, the estimated value of autocorrelation is ϕ=0.50.

$$\phi_1 = \frac{\sum_{t=1}^{n-1}(Y_t - \bar{Y})(Y_{t+1} - \bar{Y})}{\sum_{t=1}^{n}(Y_t - \bar{Y})^2} \qquad 4.1$$

On the other hand, in the control chart shown in Figure 4.1(b), the estimated value of ϕ is 0.81. Thus, the estimation of ϕ is biased by the positive trend.

Such biasing of parameters when assignable causes are present was studied by Woodall and Faltin (1993) and Boyles (2000) who found that if pattern recognition is employed to detect assignable causes, estimators of common-cause, in this case the autocorrelation level, are highly biased when an abnormal pattern is present.

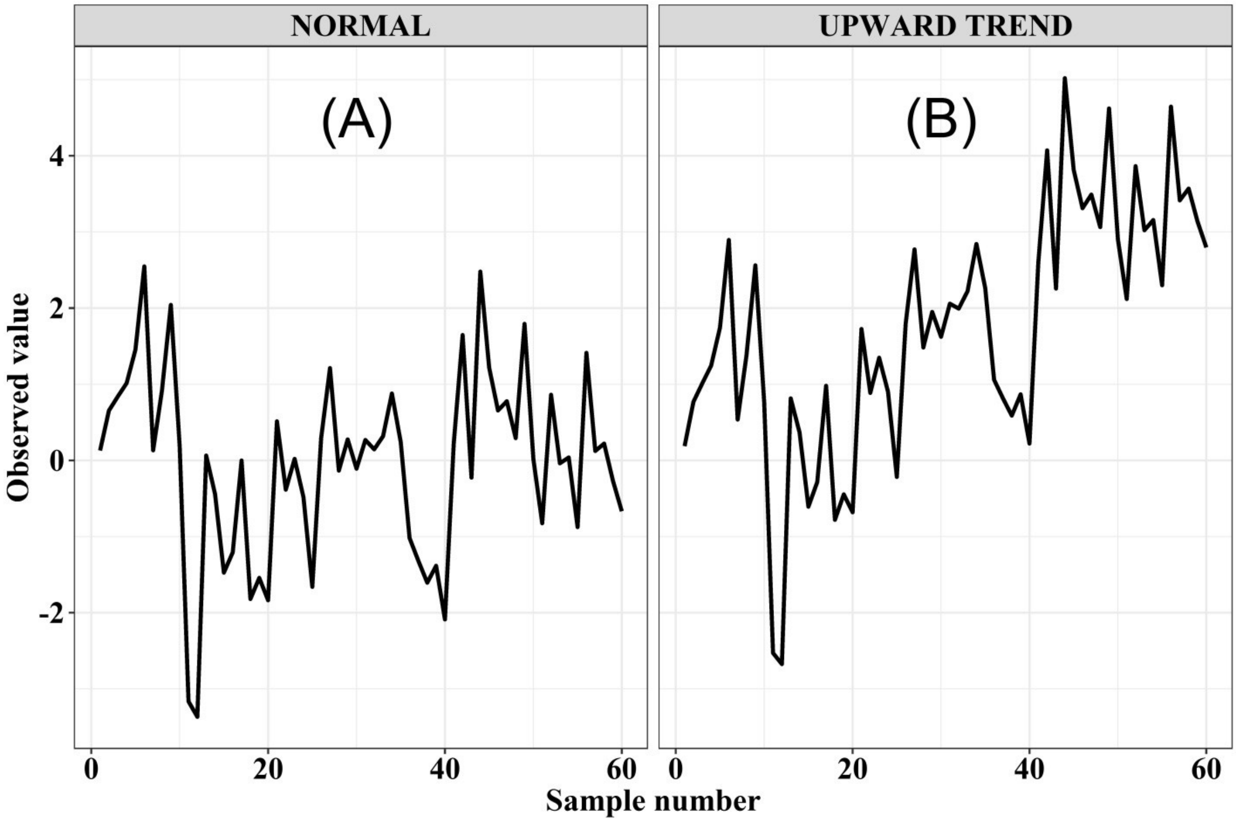

Figure 4.1: Two simple patterns with $\boldsymbol{\phi}$=0.50

In the literature, CCPR models that deal with the recognition and classification of significant patterns and that can estimate the corresponding parameters while minimising misclassification errors are very rare (Lesany et al., 2013). In this work, it is proposed to employ a PGS that is able to separate common-cause and assignable-cause effects and estimate them simultaneously. However, the estimation is carried out independently as the two disturbances are assumed to be independent due to their different sources. The dynamic regression model named *Nonlinear regression model with autoregressive moving-average errors* (NLM-ARMA) used in the proposed PGS ensures such conditions of parameter estimation (Pankratz, 1991; Hyndman and Athanasopoulos, 2014).

Once the synthetic patterns are generated by applying the proposed PGS, the next step is to train the ML algorithm. In this chapter, three factors relating to the input of the CCPR system are studied: input representation technique (IRT), PGS and kernel of the pattern recognition system. Finding arrangements of these factors that achieve the highest accuracies when the inherent noise is autocorrelated is another aim of this research.

4.3 PROPOSED CCPR SYSTEM

4.3.1 PATTERN GENERATION SCHEME

4.3.1.1 *Initial pattern generation*

The first step in producing patterns for the proposed PGS is to generate the inherent noise according to one of the three models. To generate such an inherent disturbance, it is necessary to create a white-noise vector. Firstly, an NIID vector, e_t, is produced based on the pseudo-random number generator proposed by Matsumoto and Nishimura (1998), i.e.

$$e_t \sim Normal(\mu = 0, \sigma_e = 1) \qquad 4.2$$

The methodology for generating ARMA (1,1) processes from normal white-noise proposed by Box et al. (2009) was adopted in this work.

Therefore, the inherent noise represented as an ARMA(1,1) process is represented as:

$$N_t = \phi N_{t-1} - \theta e_{t-1} + e_t \qquad 4.3$$

where if $\phi = 0,$ an MA(1) process is obtained; and if $\theta = 0$, an AR(1) model is obtained. Both ϕ and θ are in the range (-1, +1).

The standard deviation of this ARMA(1,1) process can be estimated by means of the following expression:

$$\sigma_N = \sqrt{\frac{1+\theta^2-2\phi\theta}{1-\phi^2}}\sigma_e \qquad 4.4$$

The model for disturbances in control charts, assuming independence between the inherent noise and deterministic disturbance, is the following:

$$Y_t = D_t + N_t \qquad 4.5$$

where Y_t represents the process to be monitored, D_t the deterministic disturbance and N_t the inherent noise. The deterministic part or each patter is used as shown in equations 3.2 - 3.6. Without loss of generality, in this work μ is set to zero.

The meaning and range of the pattern parameters used are shown in Table 4.1.

Table 4.1: Parameters used during pattern generation

Pattern type	Parameter	Range	Meaning
UT	β_1	Between $0.001\sigma_N$ and $0.30\sigma_N$	Slope
DT	β_1	Between $-0.30\sigma_N$ and $-0.001\sigma_N$	Slope
US / DS	*d*	0 or 1, 0 when $t < \tau$ and 1 when $t \geq \tau$ τ between 16 and n-15	τ is the time when the mean shift is observed.
US	β_2	From $0.01\sigma_N$ to $3.0\sigma_N$	Magnitude of the mean shift
DS	β_2	From $-3.0\sigma_N$ to $-0.01\sigma_N$	Magnitude of the mean shift
CYS	β_3	From $0.01\sigma_N$ to $3.0\sigma_N$	Amplitude
CYC	β_4	From 3 to 16	Frequency
SYS	β_5	From $0.1\sigma_N$ to $3.0\sigma_N$	Systematic departure

The magnitude shift, slope, systematic departure and cyclic amplitude were kept within in $6\sigma_N$ control limits in the inspection window. The frequency of the CYC pattern was determined to show at least four cycles in the inspection window.

The break point position was randomly chosen between τ=16 and τ = *n-15.* This is due to the number of parameters and degrees of freedom available during the estimation of the parameters when NLM-ARMA is utilised.

4.3.1.2 *Model selection*

In order to synthesise patterns that fulfil the desirable conditions mentioned in chapter 1, the proposed PGS comprises the following steps:

i. Generate one of the seven simple patterns using expressions as shown in the previous subsection.

   Thus, the control chart data, Y_t, can be modelled by a NLM-ARMA. In this work, independence is assumed between the causes of the inherent disturbance and the causes of patterns in the control chart, thus retaining the same seven patterns observed in NIID processes.

   The autocorrelation level is expected to remain constant during the pattern recognition window and is also assumed to be unknown.

ii. Determine the type of mean change in the pattern. Following the methodology proposed in chapter 3, the determination of the type of mean change is considered as a problem of selecting between two models, in this case, two NLM-ARMA models when Y_t is as shown in equation 4.6. The most likely break point is found by fitting piecewise NLM-ARMA models at time $\tau = 16, 17, \ldots, (n-15)$ (see previous subsection).

$$Y_t = \beta_0 + \beta_1 t + \beta_2 d + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5 (-1)^t + N_t \qquad 4.6$$

The Bayesian Information Criterion (BIC) is extracted from each fitted model and the one with the minimum BIC value is selected. Such a selected model represents the model with the most likely break point.

iii. Compare the model representing the most likely break point with the one where no break points are assumed. Equation 4.7 represents the model to be used for Y_t when there is no break point. Models 4.6 and 4.7 represent two NLM-ARMA nested models; the full model is that with the most likely break point and the reduced model the one that does not consider the existence of break points.

$$Y_t == \beta_0 + \beta_1 t + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5(-1)^t + N_t \qquad 4.7$$

The hypotheses for this problem of model selection are:

$$H_0: There\ are\ no\ break\ points\ (the\ reduced\ model\ fits\ better)$$

$$H_1: A\ break\ point\ is\ detected\ (the\ full\ model\ fits\ better)$$

The F-statistic shown in 4.8 is used to determine which model better fits to the control data.

$$F_{v_2}^{v_1} = \left(SSE_{full} - SSE_{reduced}\right)/\left(SSE_{full}/(n-k-1)\right) \qquad 4.8$$

where SSE_{full} and $SSE_{reduced}$ represent the Sum of Squared Error (SSE) of the full and the reduced model, respectively. *F* represents an F-distribution with one degree of freedom in the numerator ($v_1 = 1$) and *n – k - 1* degrees of freedom in the denominator ($v_2 = n - k - 1$), *k* being the number of parameters in the full model.

iv. Determine pattern class. Once it has been decided whether 4.6 or 4.7 fits the pattern better, the pattern class is determined by the corresponding β value that is statistically significant. Three significance levels were utilised α=0.01, 0.02 and 0.03 in this work.

Figure 4.2 summarises the proposed PGS.

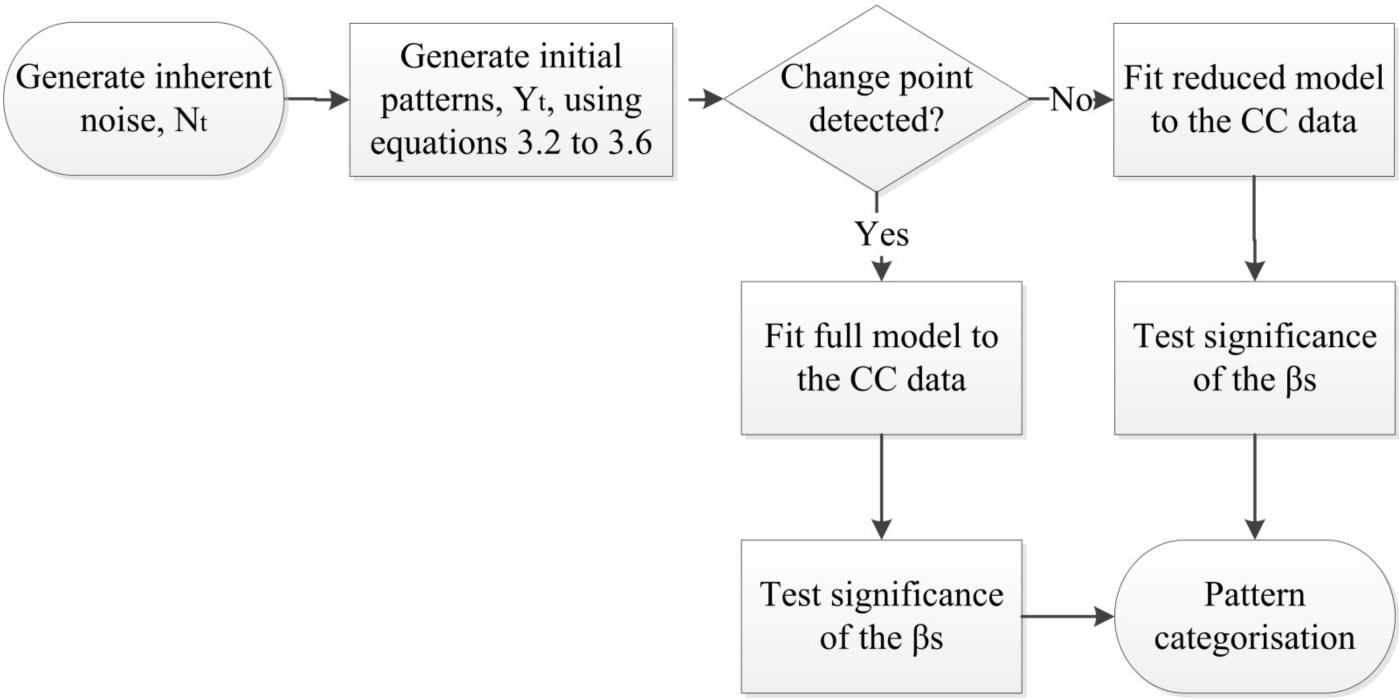
Generate inherent noise, N_t
Generate initial patterns, Y_t, using equations 3.2 to 3.6
Change point detected?
No
Fit reduced model to the CC data
Yes
Fit full model to the CC data
Test significance of the βs
Test significance of the βs
Pattern categorisation

Figure 4.2: Flowchart of the proposed PGS

4.3.2 TRAINING OF THE ML

Three sets of 5,600 training patterns, 800 of each type, were generated using α = 0.01, 0.02, 0.03 using the proposed PGS and another set was created conventionally (Pham et al., 2006; Pham and Oztemel, 1996).

The BA proposed by Pham et al. (2006) was used to find the best sets of free parameters that ensure the minimum misclassification rate with the 5-fold cross-validation technique.

The BA is a popular population-based optimisation algorithm inspired by the foraging activities of bees. Table 4.2 shows the parameters of the BA used in this work. For more explanation of the algorithm and parameters, see Pham et al. (2006) and Pham and Castellani (2013).

Table 4.2: Parameter values used in the BA

Parameter	**Symbol**	**Value**
Initial population	*n*	30
Number of "best" sites	*m*	5
Number of "elite" sites	*e*	2
Patch size for Cost parameter *C*	*ngh-c*	0.5
Patch size for Kernel parameters	*ngh-k*	0.02
Number of elite bees for the elite sites	n_e	4
Number of bees for the remaining "best" points	n_b	2

Once the synthetic patterns are generated by applying the proposed PGS, the next step is to train the ML algorithm. In this work, three factors relating to the input of the CCPR system are studied: input representation technique (IRT), PGS and kernel of the pattern recognition system. Finding arrangements of these factors that achieve the highest accuracies when the inherent noise is autocorrelated is another objective of this research.

4.3.2.1 *Input factors*

For testing purposes, one hundred sets of 700 patterns, 100 of each type, were generated using the best arrangement, and the pattern recognition accuracies obtained.

The three factors studied in the first step of the analysis are as follows:

- IRT: Standardised raw data and shape features. To standardise the raw data, the control chart data are rescaled using the following expression:

$$S_t = \frac{Y_t - \bar{Y}}{\hat{\sigma}_y} \qquad 4.9$$

where S_t represents the scaled variable to be used as input for the training of the SVM, $\bar{Y}$ is the estimated mean value of the current control chart and $\hat{\sigma}_y$ is the estimated standard deviation.

The shape features utilised here are those initially proposed by Pham and Wani (1997) and improved by Gauri (2010). These features have advantages in CCPR problems where the process to be monitored is NIID. These are reduction in the training time, increase in pattern recognition accuracy and independence from the data scale. The performance of these features has not been assessed in CCPR systems where the inherent disturbance is not NIID. Therefore, this is a factor to be considered during the first step of the analysis.

- PGS. The PGS adopted in this work deals with the correct categorisation of training patterns before they are input to the CCPR system. The performance of the recognition systems trained using patterns generated at three α levels is assessed, and compared with the recognition achieved without the PGS.
- SVM kernel. Four kernels were tested: RBF, LAPLA, TANH and BESSEL.

4.4 RESULTS

4.4.1 ANALYSIS OF THE PGS

Three sets of 70,000 random vectors, 10,000 of each pattern type, were initially generated. Each vector $(Y_1, Y_2, \dots, Y_{60})$ represents a quality characteristic sampled at time $t_1, t_2, \dots, t_{60}$. Each set of random vectors was created using one of the three first-order stationary models.

The patterns were passed through the PGS. Three significance levels were set, namely, α = 0.01, 0.02 and 0.03. The allocation of the patterns to the different classes is shown in Table 4.3. In Table 4.3, for each of the models, the column "Retained" gives the percentages of patterns for which the final classification by the PGS agrees with the classification when the patterns were initially produced. The column "Reclassified" shows the percentages of patterns for which the classification was changed after they were passed through the PGS. The column "Discarded" gives the percentages of patterns rejected by the PGS as not recognisable due to the low statistical significance of the parameters that characterise them.

It can also be observed that when α = 0.01, 24.26%, 32.91% and 43.40% of the patterns were discarded for AR, MA and ARMA processes, respectively. When this α level was used, the pattern with the highest reclassification rate was the NORM pattern for the MA and ARMA models, and the US pattern for the AR model. This was probably due to the initial blend of inherent noise and abnormal disturbance. With the significance level set to α = 0.02, 35.96%, 44.09% and 53.89% of the patterns were discarded when the AR, MA and ARMA models were used, respectively. In the case of α = 0.03, the percentage of discarded patterns increased to 44.43%, 51.93% and 60.78% for the aforementioned three processes. Therefore, as shown in Table 4.3, the proposed PGS was more efficient when α = 0.01, 19.60% of the patterns were discarded.

Table 4.3: Allocation of patterns passed through the proposed PGS (%)

	AR			**MA**			**ARMA**		
	Retained	Reclassified	Discarded	Retained	Reclassified	Discarded	Retained	Reclassified	Discarded
α = 0.01	**64.45**	**11.29**	**24.26**	**57.12**	**9.97**	**32.91**	**46.67**	**9.92**	**43.40**
NORM	62.70	20.53	16.77	52.17	22.94	24.89	38.08	26.51	35.40
UT	67.57	2.60	29.83	58.79	1.80	39.41	46.25	3.17	50.58
DT	67.65	2.83	29.52	58.78	1.86	39.37	46.16	3.18	50.66
US	59.78	21.43	18.80	57.36	17.85	24.79	54.80	13.62	31.58
DS	59.89	21.27	18.85	57.38	18.10	24.52	54.78	13.46	31.76
CYC	69.63	1.49	28.88	56.94	2.14	40.92	41.95	2.15	55.90
SYS	63.96	8.88	27.16	58.41	5.10	36.49	44.71	7.36	47.93
α = 0.02	**54.30**	**9.74**	**35.96**	**47.34**	**8.57**	**44.09**	**37.83**	**8.28**	**53.89**
NORM	48.20	25.52	26.28	38.74	26.16	35.10	26.98	27.29	45.73
UT	53.55	2.51	43.94	45.87	1.76	52.37	34.85	2.72	62.43
DT	53.77	2.68	43.55	45.98	1.81	52.21	34.87	2.77	62.36
US	58.12	14.74	27.15	54.47	12.33	33.20	51.23	8.87	39.90
DS	58.31	14.58	27.11	54.47	12.64	32.89	51.17	8.86	39.96
CYC	56.28	0.96	42.76	45.34	1.39	53.28	31.69	1.47	66.84
SYS	51.90	7.17	40.93	46.52	3.89	49.59	34.01	5.98	60.02
α = 0.03	**46.85**	**8.72**	**44.43**	**40.48**	**7.59**	**51.93**	**32.01**	**7.21**	**60.78**
NORM	38.58	27.48	33.94	30.35	26.81	42.84	20.54	26.43	53.02
UT	43.77	2.39	53.84	37.14	1.69	61.18	27.72	2.42	69.87
DT	43.89	2.53	53.58	37.35	1.76	60.88	27.72	2.49	69.79
US	55.78	11.02	33.20	51.55	9.28	39.18	48.00	6.39	45.61
DS	55.93	10.84	33.23	51.41	9.38	39.21	47.93	6.47	45.60
CYC	46.79	0.69	52.52	37.28	0.99	61.73	25.06	1.12	73.82
SYS	43.23	6.09	50.68	38.25	3.22	58.53	27.10	5.16	67.74

4.4.2 OVERALL ACCURACIES

To simplify the analysis of the results, it will be carried out in two steps. First, the best arrangement amongst the input factors for each disturbance model is determined; these factors are IRT, PGS and kernel of the recognition system. Using the best arrangement determined in the first step of the analysis, the performance of the CCPR system for a set of patterns generated with a specific autocorrelation level is then assessed.

The accuracies found for the AR inherent noise model are shown in Figure 4.3, 4.4 and 4.5. These accuracies are disaggregated by IRT, Kernel and pattern type, the PGS being the constant factor in each plot. In these plots, it can be observed that the three lines corresponding to the three α levels are nearly coincident; this indicates that the proposed PGS is robust to changes in α values.

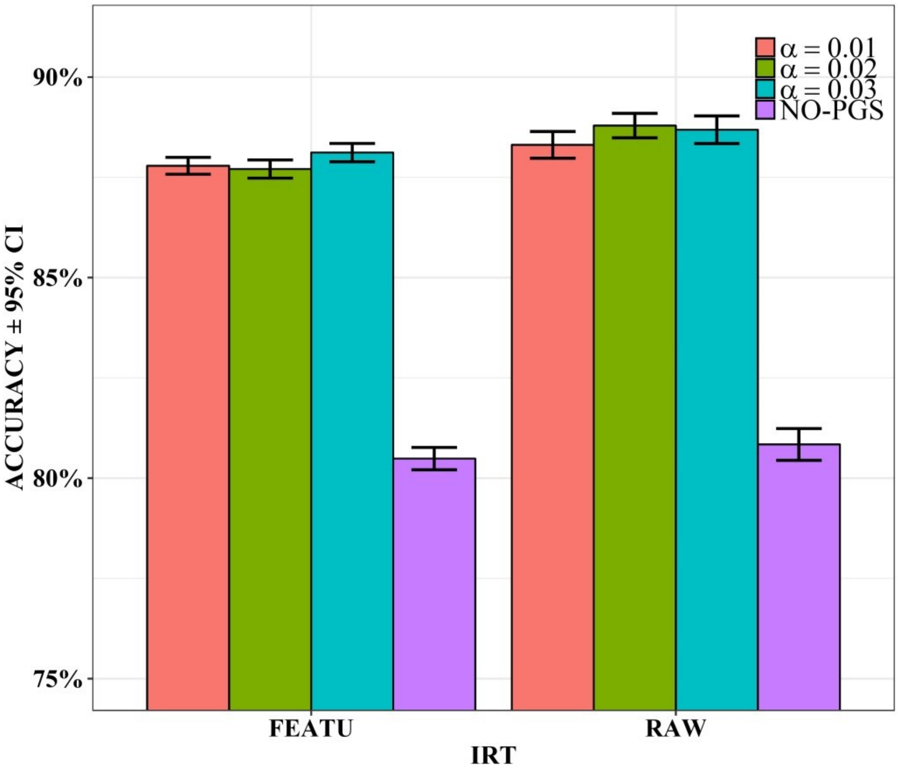

Figure 4.3: Accuracies achieved from the AR process, disaggregated by IRT

The accuracies achieved for the MA inherent noise model are shown in Figure 4.6, 4.7 and 4.8. These accuracies are also disaggregated by pattern type, IRT, kernel and PGS.

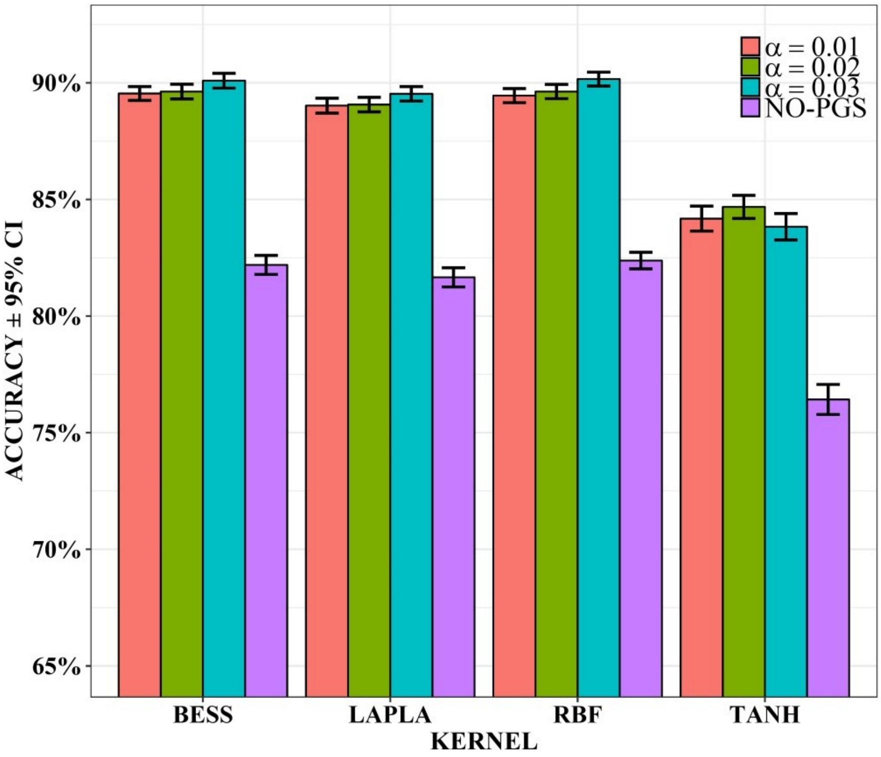

Figure 4.4: Accuracies achieved from the AR process, disaggregated by kernel

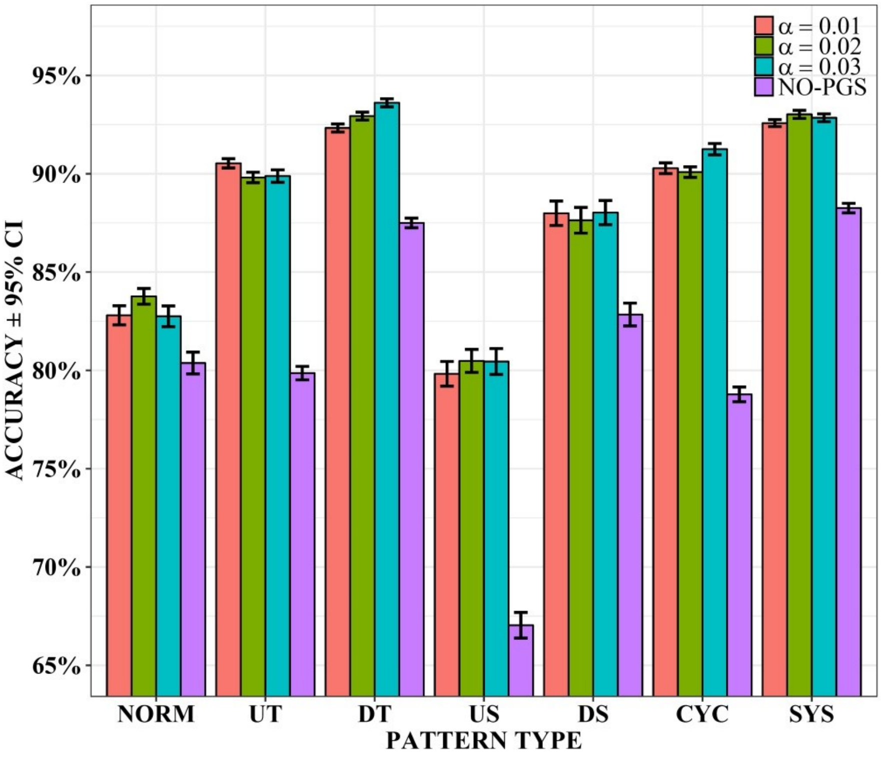

Figure 4.5: Accuracies achieved from the AR process, disaggregated by pattern type

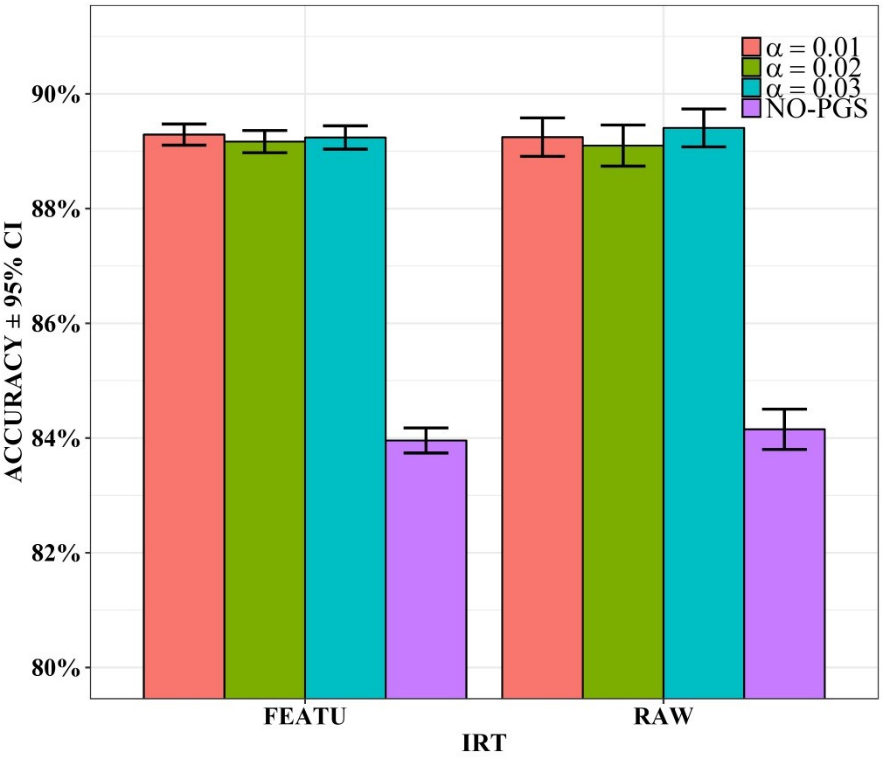

Figure 4.6: Accuracies achieved from the MA process, disaggregated by IRT

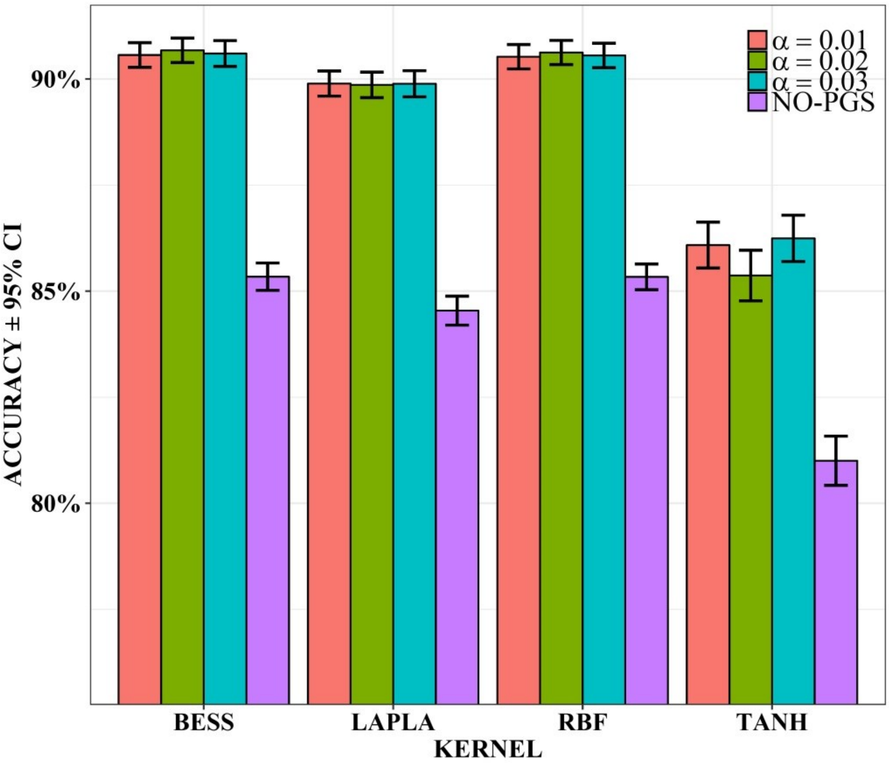

Figure 4.7: Accuracies achieved from the MA process, disaggregated by kernel

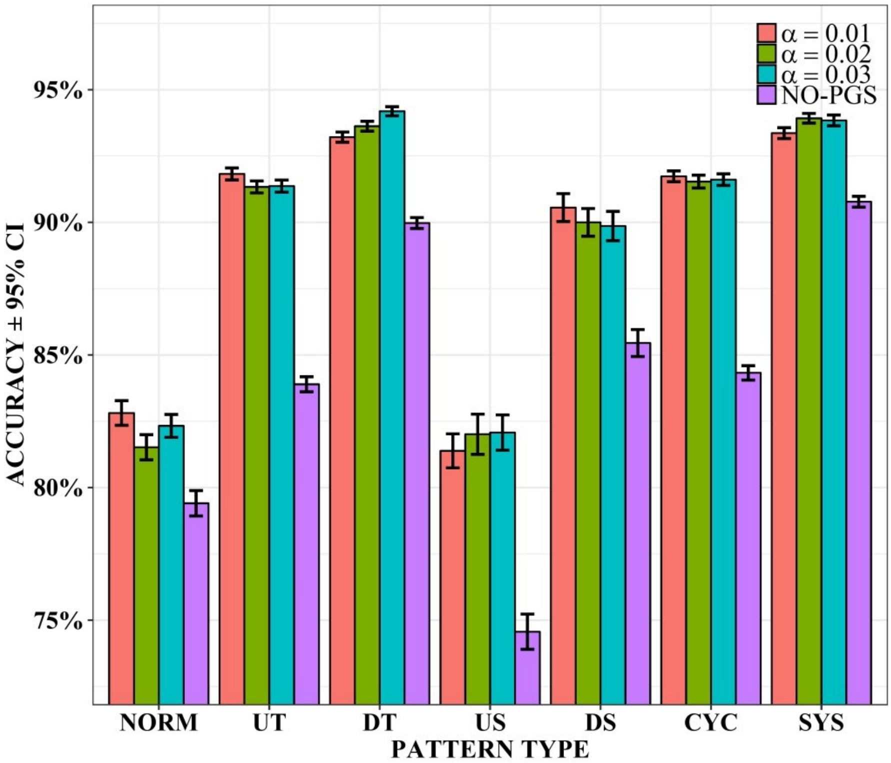

Figure 4.8: Accuracies achieved from the MA process, disaggregated by pattern type

The accuracies achieved for the ARMA inherent noise model are shown in Figure 4.9, 4.10 and 4.11. These accuracies are disaggregated by pattern type, IRT, kernel and PGS.

In Figures 4.3 to 4.11, it can be observed that the three lines corresponding to the three α levels are nearly coincident; this indicates that the proposed PGS is robust to changes in α values.

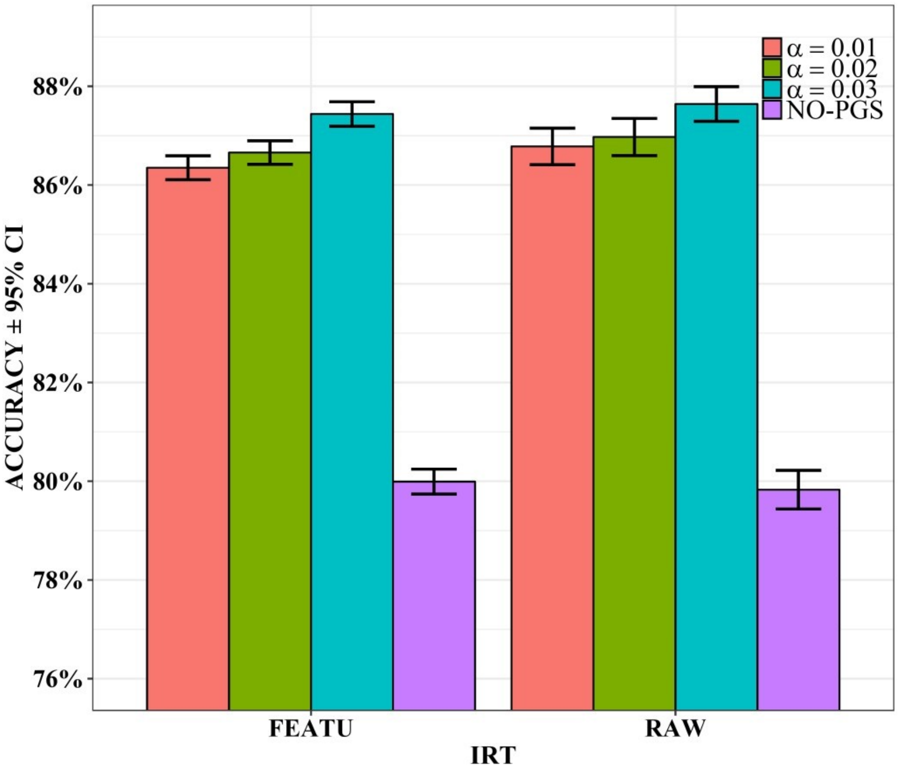

Figure 4.9: Accuracies achieved from the ARMA process, disaggregated by IRT

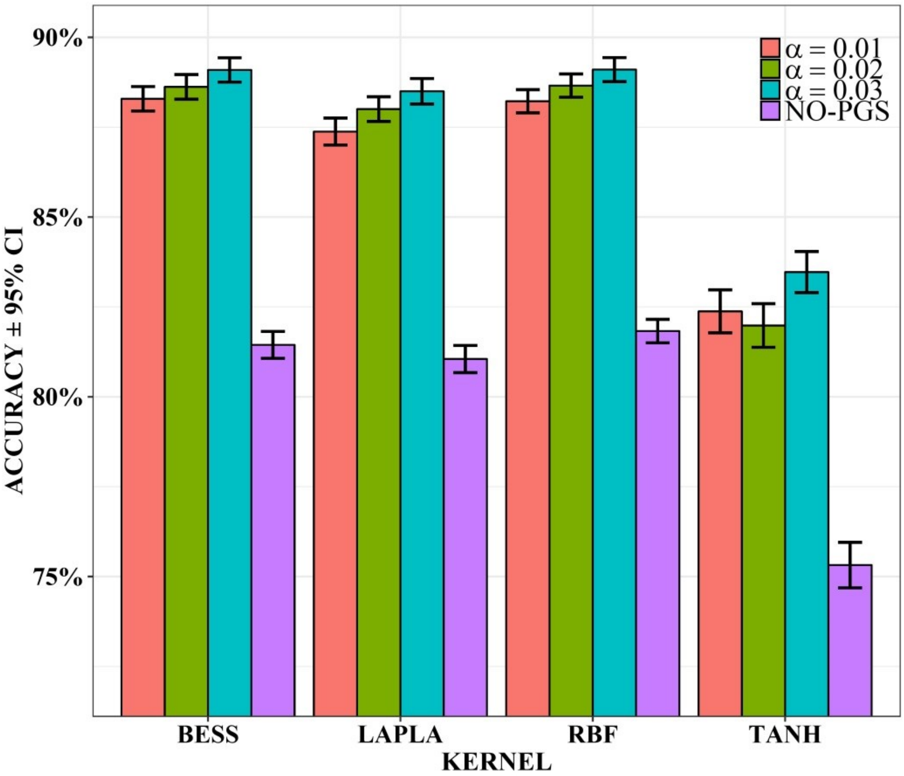

Figure 4.10: Accuracies achieved from the ARMA process, disaggregated by kernel

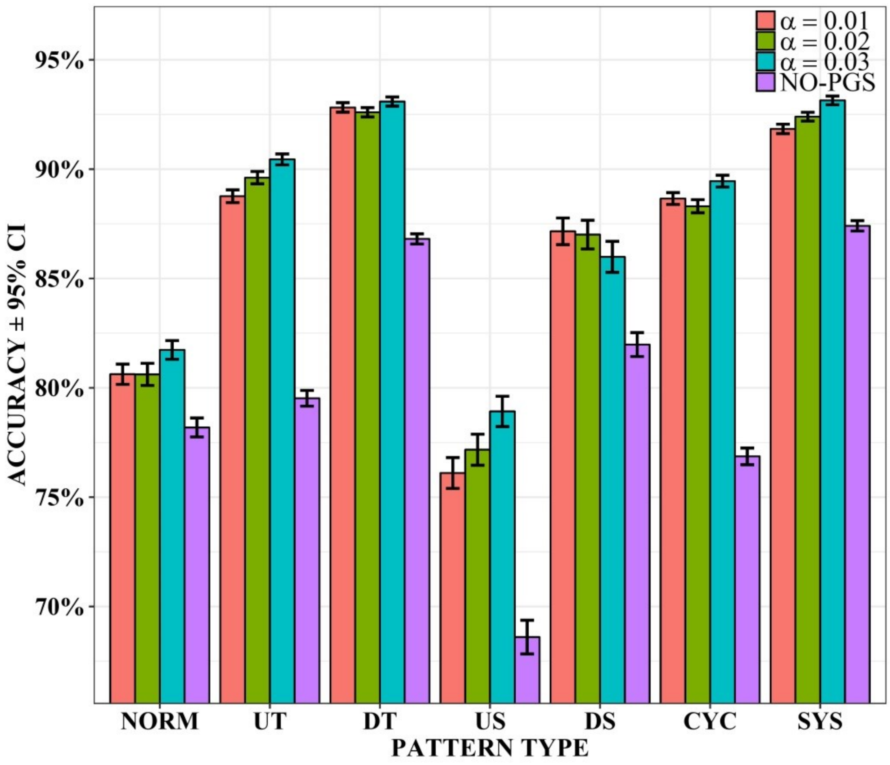

Figure 4.11: Accuracies achieved from the ARMA process, disaggregated by pattern type

ANOVA of type 4x2x4 with single, double and triple interactions was utilised to determine which factors were significant for the pattern recognition accuracies. The p-values of each factor are shown in Table 4.4.

Table 4.4: P-values of the of the three factors obtained from the ANOVA

p-values			
Factor	**AR**	**MA**	**ARMA**
PGS	0.0000	0.0000	0.0000
KERNEL	0.0000	0.0000	0.0000
IRT	0.0000	0.5150	0.0638
PGS*KERNEL	0.0186	0.1800	0.1510
KERNEL*IRT	0.0000	0.0000	0.0000
PGS*IRT	0.0583	0.6680	0.2197
PGS*KERNEL*IRT	0.0000	0.0000	0.0000

Tukey post-hoc tests were used to determine the best arrangement for each disturbance model; results are summarised in Table 4.5.

Table 4.5: Best arrangement for AR, MA and ARMA processes

Disturbance model	**AR**	**MA**	**ARMA**
PGS	α = 0.01, 0.02, 0.03	α = 0.01, 0.02, 0.03	α = 0.01 , 0.02
IRT	Raw data	Raw data, Shape features	Raw data, Shape features
Kernel	Bessel/RBF	Bessel/RBF	Bessel/RBF

4.4.3 ANALYSIS OF THE BEST ARRANGEMENT OF INPUT FACTORS

To assess the performance of the SVM trained with the best combination of factors, one hundred sets of 700 patterns, 100 of each type were generated for each autocorrelation level. The combination used is: PGS with α = 0.01, Raw data and RBF kernel.

Table 4.6 gives the accuracies found for the AR and ARMA models disaggregated by pattern type and autocorrelation level. For the case of the AR process, an overall accuracy of 90.03% was found. The US pattern type was the one with the least recognition accuracy. It was also found that the accuracy for all pattern types was lowest with $\phi \geq 0.70$.

The accuracies achieved for the MA and ARMA processes are shown in Table 4.7, disaggregated by pattern type and moving-average level. For the MA process, the US pattern class and patterns of all types generated with $\theta \geq 0.70$ yielded the lowest recognition accuracies.

To obtain the accuracies for the ARMA process of Table 4.6, the ϕ level was fixed and the θ values were randomised. It can be observed that the overall accuracy for the ARMA process considering all the ϕ levels is 89.08%, the Normal pattern having the lowest accuracy. Again, an autocorrelation level of $\phi \geq 0.70$ produced the least accuracy.

Likewise in Table 4.7, to obtain the accuracies for the ARMA process, the θ was fixed and the ϕ levels were randomised. It can be seen that the lowest accuracies were obtained for Normal patterns and $\theta \geq 0.70$.

Therefore, as shown in Table 4.6 and Table 4.7, the three proposed CCPRs showed the least efficiency when the autocorrelation level in the inherent noise was strong (greater than 0.5).

Table 4.6: Accuracies for AR and ARMA processes by ϕ values (%)

ϕ	OVERALL	NORM	UT	DT	US	DS	CYC	SYS
AR	**90.03**	**83.92**	**91.08**	**93.60**	**79.70**	**96.64**	**90.97**	**94.28**
$\phi \leq -0.7$	91.86	77.50	91.49	94.21	93.17	99.82	91.80	95.00
$-0.7 < \phi \leq -0.5$	92.97	80.00	93.51	93.65	96.54	99.78	93.13	94.33
$-0.5 < \phi \leq -0.3$	92.98	82.63	93.93	93.46	94.92	98.52	93.46	93.83
$-0.3 < \phi \leq 0.0$	92.19	87.17	93.07	93.07	87.94	97.44	92.51	94.12
$0.0 < \phi \leq 0.3$	90.47	89.52	91.94	93.20	77.37	95.46	91.69	94.10
$0.3 < \phi \leq 0.5$	89.22	90.24	91.29	93.79	68.76	95.06	91.08	94.31
$0.5 < \phi < 0.7$	87.39	86.13	89.15	93.71	65.27	94.54	88.99	93.91
$\phi \geq 0.7$	83.15	78.15	84.26	93.68	53.65	92.53	85.09	94.66
ARMA	**89.08**	**80.61**	**89.63**	**93.17**	**82.43**	**95.55**	**89.44**	**92.77**
$\phi \leq -0.7$	88.22	67.83	88.72	90.76	90.94	99.03	87.40	92.83
$-0.7 < \phi \leq -0.5$	90.26	75.62	89.78	92.16	93.68	98.56	90.33	91.67
$-0.5 < \phi \leq -0.3$	91.75	82.37	92.35	93.95	93.02	97.13	91.26	92.16
$-0.3 < \phi \leq 0.0$	91.64	86.73	92.05	94.08	88.27	96.53	91.63	92.17
$0.0 < \phi \leq 0.3$	91.22	88.64	91.79	94.68	83.17	95.77	91.68	92.81
$0.3 < \phi \leq 0.5$	90.55	88.27	91.74	94.49	79.30	94.97	91.17	93.93
$0.5 < \phi < 0.7$	88.00	84.72	88.42	93.98	72.63	93.83	88.66	93.73
$\phi \geq 0.7$	81.05	70.70	82.15	91.28	58.39	88.60	83.36	92.85

Table 4.7: Accuracies for MA and ARMA processes by θ values (%)

θ	TOTAL	NORM	UT	DT	US	DS	CYC	SYS
MA	**91.47**	**84.86**	**92.92**	**94.39**	**82.66**	**97.75**	**93.15**	**94.59**
$\theta \leq -0.7$	91.88	80.39	89.36	94.24	95.90	99.99	88.76	94.52
$-0.7 < \theta \leq -0.5$	92.56	80.75	92.40	93.82	94.15	99.97	92.32	94.54
$-0.5 < \theta \leq -0.3$	93.22	81.64	94.59	94.16	93.18	99.96	94.39	94.60
$-0.3 < \theta \leq 0.0$	92.86	83.07	95.20	94.15	88.41	99.32	95.41	94.43
$0.0 < \theta \leq 0.3$	92.18	86.99	95.13	94.29	80.71	98.14	95.29	94.73
$0.3 < \theta \leq 0.5$	90.41	87.48	93.54	94.84	72.76	96.06	93.65	94.51
$0.5 < \theta < 0.7$	89.85	89.27	92.99	95.07	68.99	94.53	93.33	94.79
$\theta \geq 0.7$	88.83	89.29	90.18	94.52	67.21	94.03	92.03	94.57
ARMA	**89.48**	**81.47**	**89.98**	**93.42**	**83.65**	**95.71**	**89.50**	**92.65**
$\theta \leq -0.7$	89.21	74.55	87.04	91.52	96.34	98.40	85.57	91.07
$-0.7 < \theta \leq -0.5$	90.23	77.16	89.67	93.29	93.99	98.13	87.87	91.47
$-0.5 < \theta \leq -0.3$	90.84	80.42	91.20	93.74	89.77	97.55	91.00	92.18
$-0.3 < \theta \leq 0.0$	90.71	80.43	92.67	94.59	85.38	96.66	92.61	92.61
$0.0 < \theta \leq 0.3$	90.34	83.97	92.64	94.07	80.50	96.20	91.66	93.35
$0.3 < \theta \leq 0.5$	89.58	84.75	91.04	94.24	76.37	94.78	91.33	94.56
$0.5 < \theta < 0.7$	88.29	85.62	89.30	93.41	74.22	93.68	89.16	92.65
$\theta \geq 0.7$	86.67	84.84	86.29	92.47	72.66	90.29	86.81	93.30

4.5 REAL DATA APPLICATION

4.5.1 GLOBAL MEAN LAND-OCEAN TEMPERATURE

To demonstrate the ability of the proposed CCPR system to handle real data, the data provided by Shumway and Stoffer (2011) used for proving the Global warming phenomenon are utilised. The data are shown in Figure 4.12 and represent the global mean land–ocean temperature from 1950 to 2010. This data were used by Shumway & Stoffer (2011) as an argument for global warming hypothesis and he spotted a trend with positive slope on it.

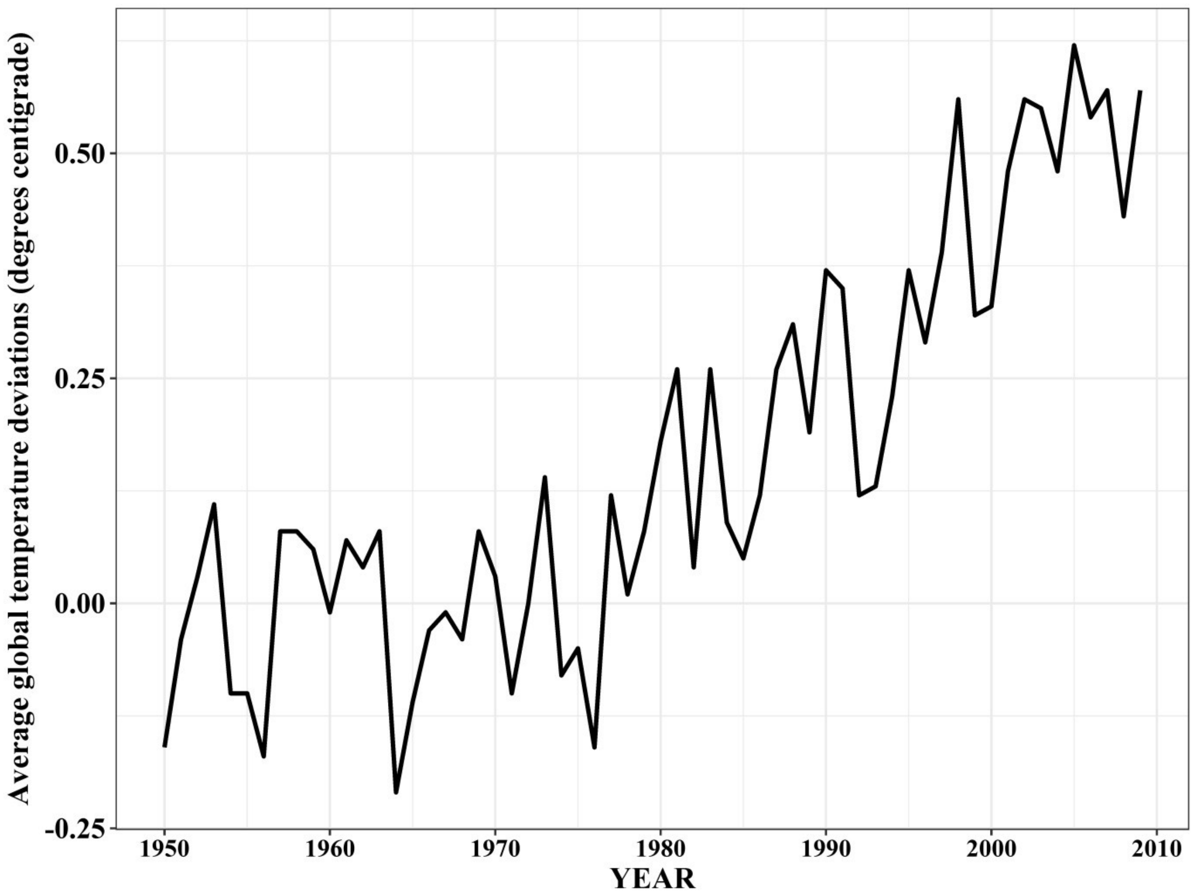

Figure 4.12: Global mean land-ocean temperature from 1950 to 2010

Analysing the autocorrelation structure of the aforementioned data, it was determined that a AR(1) fits right to the data, therefore, the CCPR system trained for AR(1) inherent noise was applied. This CCPR system identified a UT pattern in the data. Applying the methodology here proposed, the most likely breakpoint is detected at τ = 44 (1994). The reduced NLM-ARMA model was fitted to the data as the p-value of the F-test for nested models was large (0.0735).

Table 4.8 presents the ANOVA obtained when the reduced NLM-ARMA was fitted to the data.

Table 4.8: ANOVA of the NLM-ARMA model fitted to the Gloabal warming data

	Estimate	**Std. Error**	**t value**	**p-value**
ϕ	0.4630	0.1129	4.1005	0.0000
Intercept	-0.5845	0.1863	-3.1366	0.0017
Slope (Trend)	0.0434	0.0053	8.2247	0.0000
Amplitude (Cyclic)	-0.0895	0.0907	-0.9858	0.3242
Frequency (Cyclic)	7.0231	0.0922	76.17245	0.0000
Departure (Systematic)	0.1038	0.4514	0.2299	0.5909

It can be observed that the parameter related to the Trend pattern is statistically significant, so the ANOVA confirms the presence of an UT of magnitude 0.0434

4.5.2 MONTHLY TOTAL OF ACCIDENTAL DEATHS IN THE UNITED STATES OF AMERICA

Figure 4.13 shows the monthly total of accidental deaths in the United States (January 1973–December 1978) taken from Hyndman and Athanasopoulos (2014). By analysing the autocorrelation structure of the data, a MA(1) was determined to fit properly. Therefore, the CCPR trained using the best combination of input factor for inherent noise modelled by a MA(1) is utilised. This CCPR system identified a CYC pattern in the data.

The reduced and the full NLM-ARMA models were fitted to the data and the p-value of the F test obtained. The value was 0.0798, thus the reduced model fitted better. The most likely breakpoint was detected in $\tau = 52$ (May 1977). Table 4.9 shows the values of the ANOVA obtained from the reduced model. In that Table can be observed that the magnitude of the parameter related to the CYC pattern is statistically significant. It can also be observed that the frequency of this pattern is 12.

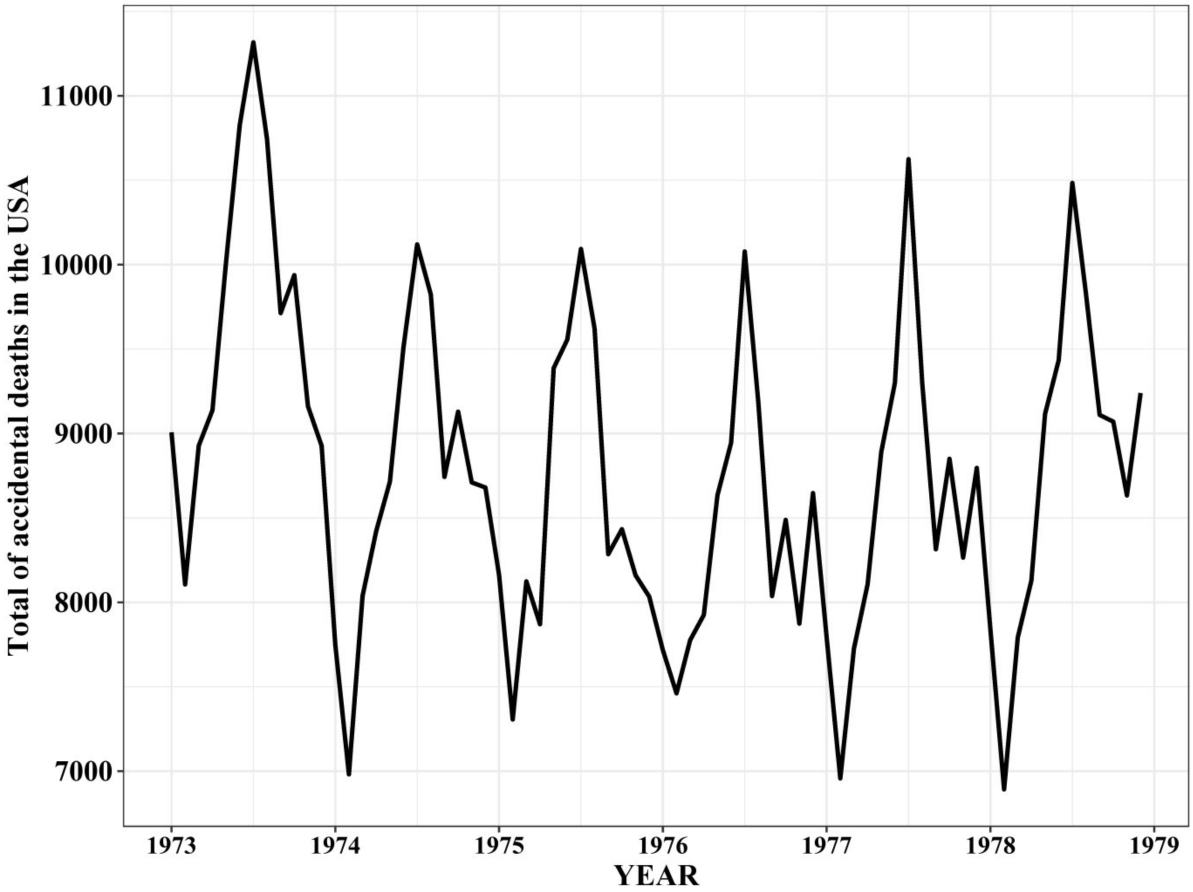

Figure 4.13: Monthly total of accidental deaths in the United States

Table 4.9: ANOVA of the NLM-ARMA model fitted to the monthly total of accidental deaths in the USA

	Estimate	**Std. Error**	**t value**	**p-value**
θ	0.6298	0.1506	4.1819	0.0000
Intercept	9215.1970	244.8792	37.6316	0.0000
Slope (Trend)	-11.3478	5.8244	-1.9483	0.0514
Amplitude (Cyclic)	-745.4094	168.2036	-4.4316	0.0000
Frequency (Cyclic)	12.0021	0.9987	12.01772	0.0000
Departure (Systematic)	-19.5413	28.4055	-0.6879	0.4915

4.6 SUMMARY

In this chapter, the assumptions regarding the inherent noise in the CC pattern were relaxed, the condition of independence (non-autocorrelation) being the one altered. Thus, three time series models were used to represent the inherent noise, the MA(1) and ARMA(1,1) models having never been used in the CCPR domain previously.

Also, a PGS based on a dynamic regression model was proposed. Using that model, the common-cause and pattern parameters were simultaneously estimated, thus avoiding biasing. This PGS consists of three stages: initial pattern generation, break point detection and final pattern categorisation.

Once the CCPR system had been trained using all possible combinations of factors, the recognition accuracies were analysed and the best combination of factors showing the highest accuracies were obtained. Furthermore, the effect of the autocorrelation levels was assessed when the best arrangement of factors was used to train the CCPR.

Finally, the CCPR system trained with the best arrangement of factors for the AR(1) model was employed to identify patterns in an example involving real data.

5 CCPR FOR FEEDBACK-CONTROLLED PROCESSES

5.1 PRELIMINARIES

This chapter presents a CCPR system for feedback-controlled processes. This CCPR again includes a method for generating patterns of feedback-controlled processes and a ML algorithm. The reason for using pattern generators was given in Chapter 1.

The results are analysed using the ANOVA and Tukey tests, and the best arrangement of input factors for the two time series models seen in previous chapters are determined.

Finally, real data on the thickness of a very thin metallic film in the early stages of the development of an electronic device was put though a CCPR system trained to identify patterns in an AR-PID process.

The remainder of the chapter is organised as follows. Section 5.2 introduces the proposed method for pattern synthesis. In section 5.3, the ML algorithm and its training are described. The results obtained are analysed in section 5.4. Finally, section 5.5 presents the results of using a CCPR system trained with the best arrangement of factors on the metallic film thickness data.

5.2 PATTERN GENERATION

This section presents a method for synthesising patterns for feedback-controlled processes using two different controllers and two time series models.

5.2.1 INITIAL PATTERN GENERATION

Two time series models were used to model the inherent noise of the monitored process, ARMA(1,1) and AR(1). The ARMA (1,1) model is shown in equation 5.1.

$$N_t = \phi N_{t-1} + \theta e_{t-1} + e_t \qquad 5.1$$

where if θ=0, an AR(1) process is obtained. e_t represents white noise which is NIID, with mean zero and standard deviation of one. For further information about generating stationary processes starting from white noise e_t, see Box et al. (2009).

The seven simple patterns, represented by D_t, were generated using expressions 3.2 to 3.6 and the PGS proposed in chapter 4 were utilised to fully synthesise autocorrelated patterns. Therefore, Y_t be the quality characteristic under study without the effect of the controller, defined as follows:

$$Y_t = N_t + D_t \qquad 5.2$$

5.2.2 SPC-EPC PROCESSES

The proposed scheme for recognising patterns of feedback-controlled processes is divided into three steps:

i. Initial generation of in-control processes and estimation of the controller parameters. An autocorrelated process, $N_{t,}$ is generated according to one of the two time series models employed in this work (see section 5.2.1). Using N_t, the parameters of the following two controllers are estimated:

   ❖ Minimum-Mean-Squared-Error (MMSE) (Jiang and Tsui, 2002):

$$X_t = \phi X_{t-1} + (\theta - \phi)(N_t - T) \qquad 5.3$$

   The parameters ϕ and θ are estimated by fitting an ARMA(1,1) or AR(1) model to N_t. In this research, the target value, T, is set to zero without loss of generality.

❖ Proportional Integral Derivative (PID) (Montgomery, 2009):

$$X_t = -k_P N_t - k_I \sum_{i=1}^{t} N_i - k_D (N_t - N_{t-1}) \qquad 5.4$$

The parameters k_P, k_I, k_D are estimated by minimising the MSE of the output variable, N_t, constrained to the following stability region for stationary processes (Box, 1994; Tsung and Shi, 1999):

$$(k_P, k_I, k_D) = \begin{cases} k_I \geq 0 \\ k_P + {k_I}/{2} + 2k_D < 1 \\ -1 < k_D < 1 \\ -k_D(1 + k_P + k_I) - k_P < 1 \end{cases} \qquad 5.5$$

ii. Using the autocorrelated data, N_t, generated in (i), two methods were followed to generate patterns: the conventional method adopted in most of the work on CCPR (denoted here by PGS-C) and the scheme proposed in chapter 4 based on testing the statistical significance of the pattern parameters (denoted by PGS-1). In the first scheme, patterns are generated directly from the equations shown in 3.2 - 3.6 and no further treatment is made to the data. In the second scheme, by fitting a dynamic regression model to the CC data, the statistical significance of the pattern parameters is tested. The chosen significance level in the latter scheme is $\alpha = 0.01$.

iii. Once the patterns have been generated and the parameters of the PID and MMSE controllers have been estimated, the controller is applied and two different variables are obtained from the original, the feedback-controlled output, Z_t, and the controller compensation, X_t. To determine which of these two signals to monitor and analyse in order to achieve the highest pattern recognition accuracy is a key point in this work. Z_t can be obtained using the following expression:

$$Z_t = Y_t + X_{t-1} \qquad 5.6$$

where Y_t represents the control chart patterns as previously described.

5.2.3 INPUT FACTORS OF THE CCPR SYSTEM

In addition to the pattern generation methods explained in the previous subsection, the CCPR systems also involve applying an IRT and a recognition system.

The two IRTs tested in this work are:

- ❖ Standardised raw data: The length of the input vector is preserved and the data are scaled to maintain the capability of the CCPR for recognising patterns irrespective of the process mean and standard deviation (Zorriassatine and Tannock, 1998). The following expression was used for standardisation:

$$S_t = \frac{Z_t - \bar{Z}}{\hat{\sigma}_Z} \qquad 5.7$$

  where $\bar{Z}$ and $\hat{\sigma}_Z$ represent the mean and standard deviation of the output variable Z_t, respectively. This standardisation was applied to both signals to monitor, X_t and Z_t

- ❖ Shape features: To reduce the dimension of the input vector, the shape features initially proposed by Pham and Wani (1997) and then improved by Gauri (2010) for NIID observations are extracted. These features have the following characteristics: independence from the data scale, enhancement of the pattern recognition accuracy and reduction of training time.

As mentioned previously, SVM was used as the pattern recognition algorithm in this work. SVM is a relatively recent ML algorithm. It has many advantages compared to other existing methods: generalisation capacity, ease of use and solution uniqueness (De Tejada and Martìnez-Echevarrìa, 2007). SVMs can also deal with nonlinear formulations, provide a trade-off between dimensionality (space complexity) and accuracy and produce good results in pattern recognition applications. Three SVM kernels for nonlinear decision classification were tested, namely, RBF, LAPLA and Bessel.

Figure 5.1 depicts the proposed scheme for CCPR of feedback-controlled process for each inherent noise model.

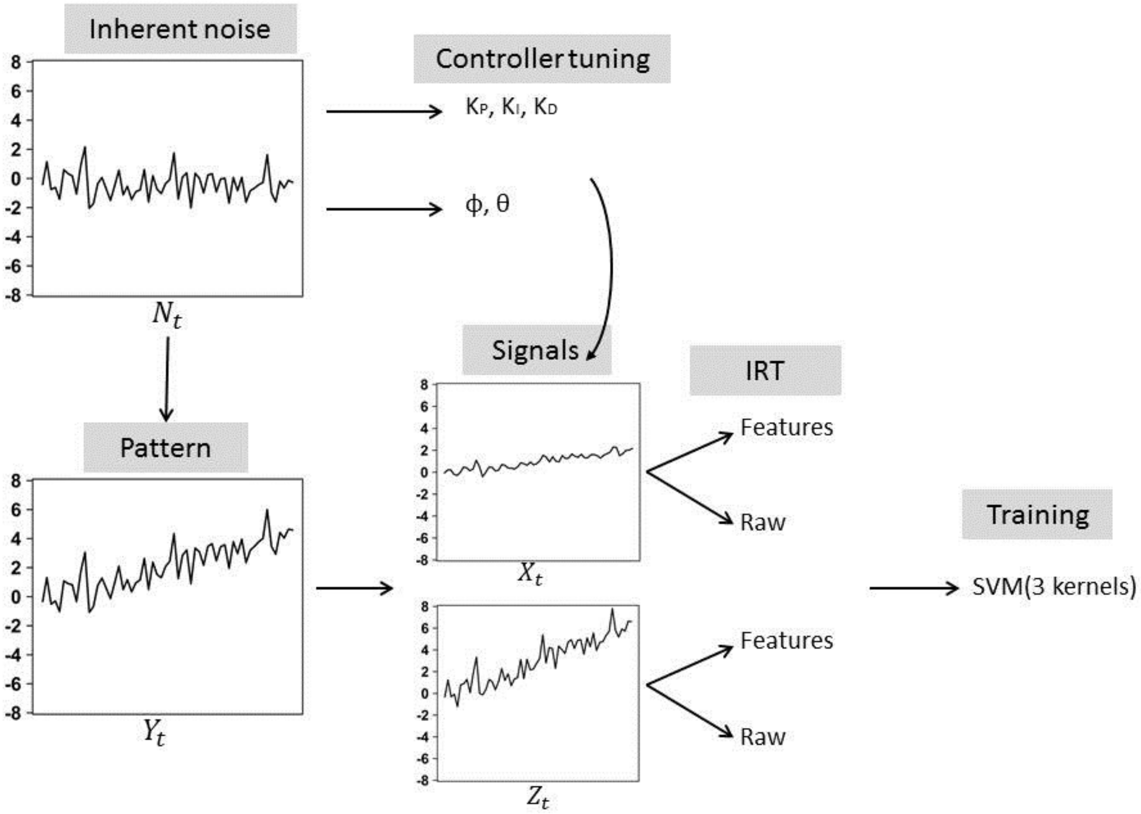

Figure 5.1: Proposed scheme for CCPR of feedback-control processes

5.3 TRAINING OF THE RECOGNITION SYSTEM

Two sets of 5600 patterns (Y_t), 800 of each type, were generated; one of the sets was generated using the methodology proposed in chapter 4 for autocorrelated patterns, and the other following the conventional PGS adopted in most of the CCPR literature (Pham and Oztemel, 1996; Pham et al., 2006). The two aforementioned feedback controllers were applied to both pattern sets.

A population-based algorithm, the BA, proposed by Pham et al. (2006), was chosen to determine the best sets of SVM parameters that minimise the misclassification rate during training. The misclassification rate under five-fold cross-validation was selected as the loss function to be minimised. The BA algorithm was chosen for its proven ability to find globally optimal solutions in diverse complex optimisation problems, using both local and global search techniques (Castellani, 2013). Table 5.1 shows the BA parameter values used. For further information regarding this algorithm and its parameters, see Pham et al. (2006) and Yuce et al. (2013).

Table 5.1: Parameters of the BA used during SVM training

Parameter	Symbol	Value
Initial population	n	20
Number of "best" sites	m	3
Number of "elite" sites	e	2
Patch size for Cost parameter *C*	*ngh-c*	0.5
Patch size for Kernel parameters	*ngh-k*	0.02
Number of elite bees for the elite sites	n_e	3
Number of bees for the remaining "best" points	n_b	2

5.4 RESULTS

For test purposes, for each inherent noise model, 200 sets of 700 patterns were generated, 100 sets created using PGS-1 and the remaining using PGS-C. Each synthesised pattern consisted of a random sample of length *n=60*, collected at time t_1, t_2,....,t_{60}. For simplicity, the analysis of accuracies has been divided into two parts; in the first part, the five aforementioned factors are analysed and the best combination is determined. In the second part, the performance of the best arrangement is studied and disaggregated by controller type and pattern type.

5.4.1 ANALYSIS OF INPUT FACTORS

The mean accuracies and their 95% confidence intervals (CI) of the AR(1) process for the five studied factors are shown from Figure 5.2 to Figure 5.6, these are disaggregated by pattern type.

The recognition accuracies and 95% CI of the ARMA model for the five studied factors are shown from Figure 5.7 to Figure 5.11. In these figures, the accuracies are disaggregated by pattern type.

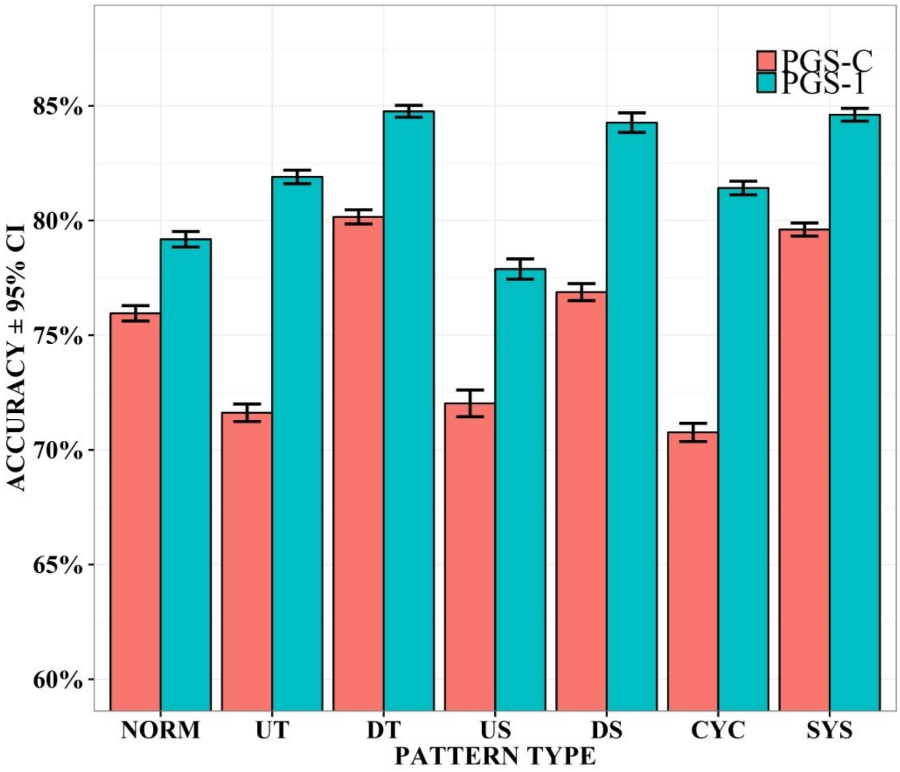

Figure 5.2: Accuracies achieved from the AR process, disaggregated by PGS

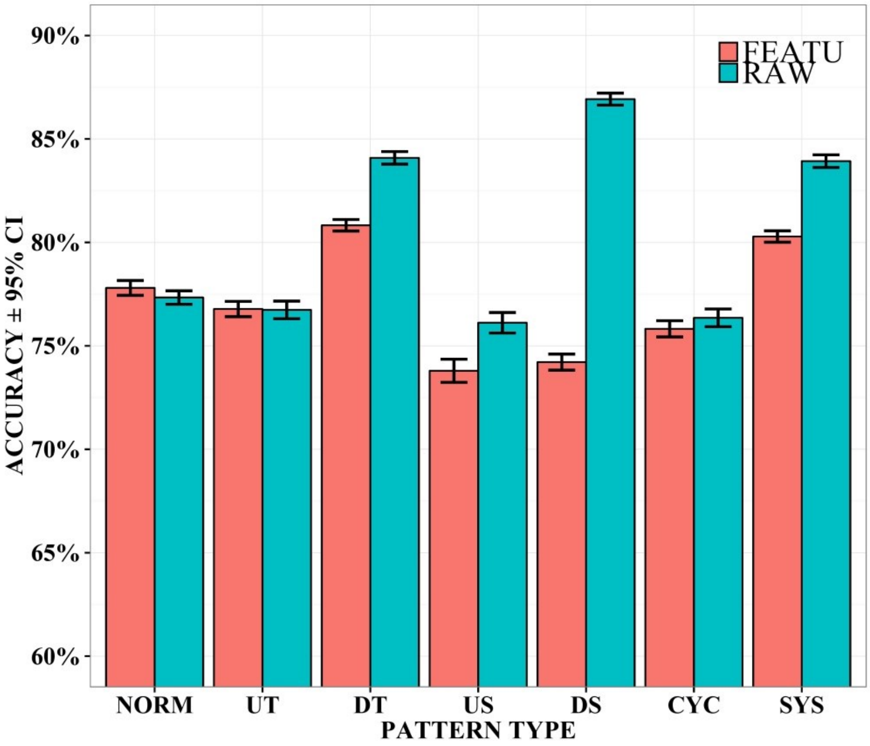

Figure 5.3: Accuracies achieved from the AR process, disaggregated by IRT

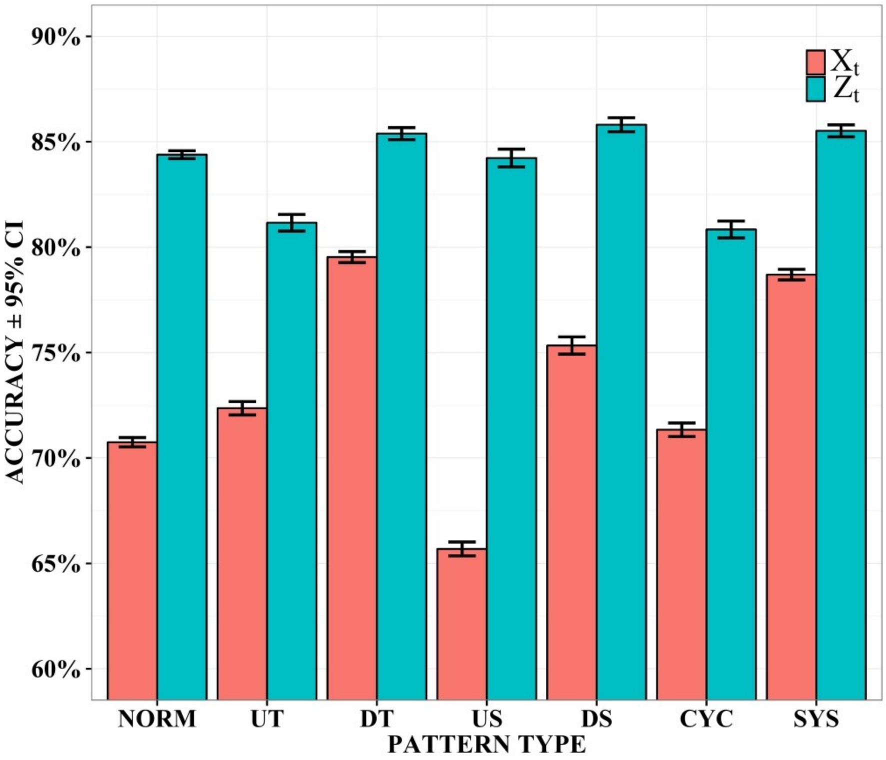

Figure 5.4: Accuracies achieved from the AR process, disaggregated by signal to monitor

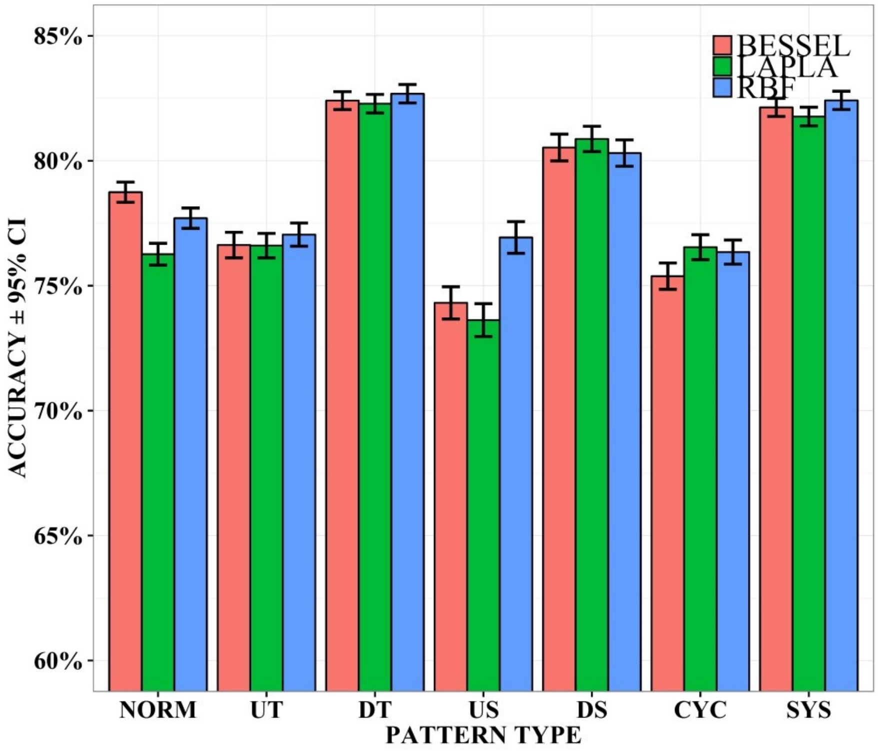

Figure 5.5 : Accuracies achieved from the AR process, disaggregated by kernel

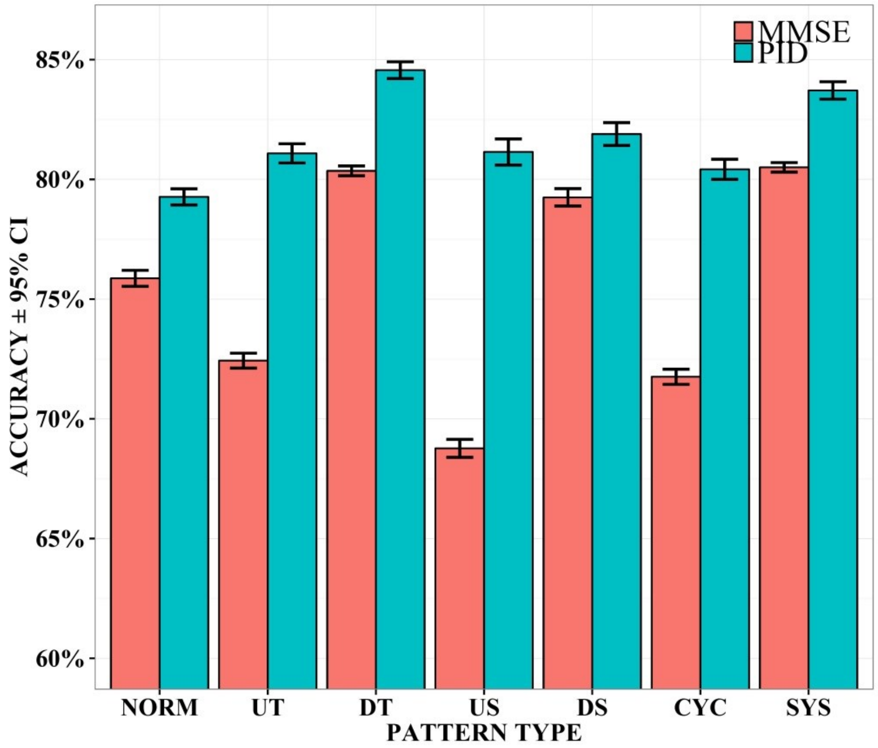

Figure 5.6: Accuracies achieved from the AR process, disaggregated by kernel

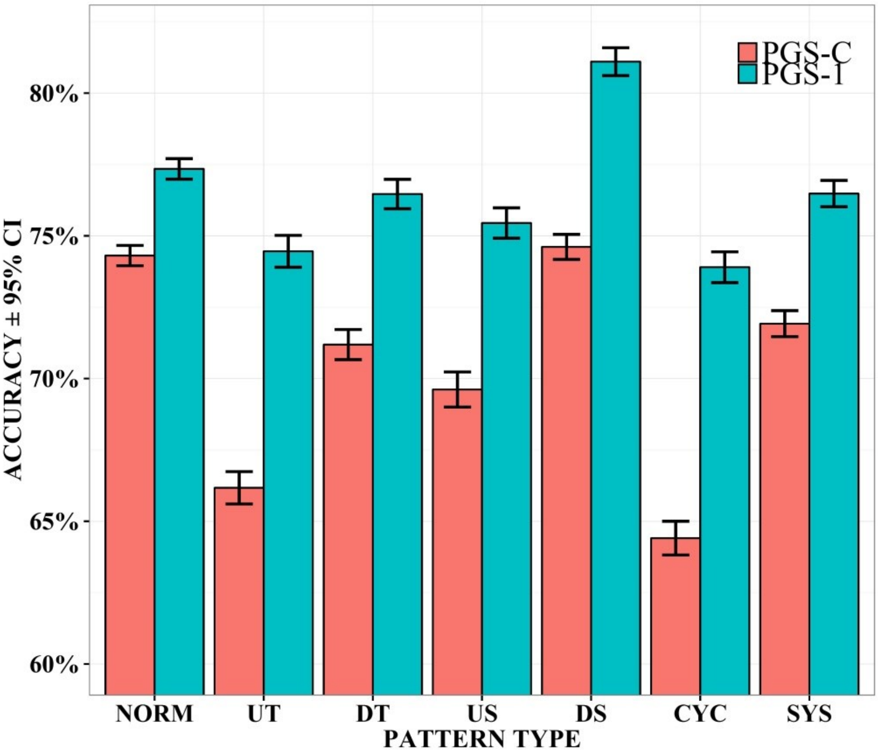

Figure 5.7: Accuracies achieved from the ARMA process, disaggregated by PGS

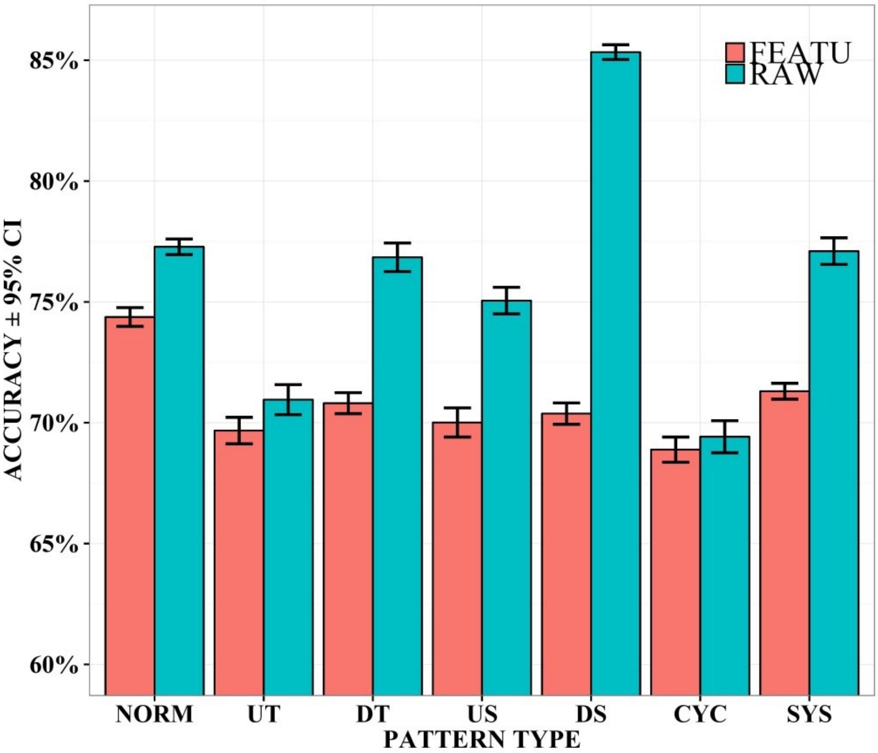

Figure 5.8: Accuracies achieved from the ARMA process, disaggregated by IRT

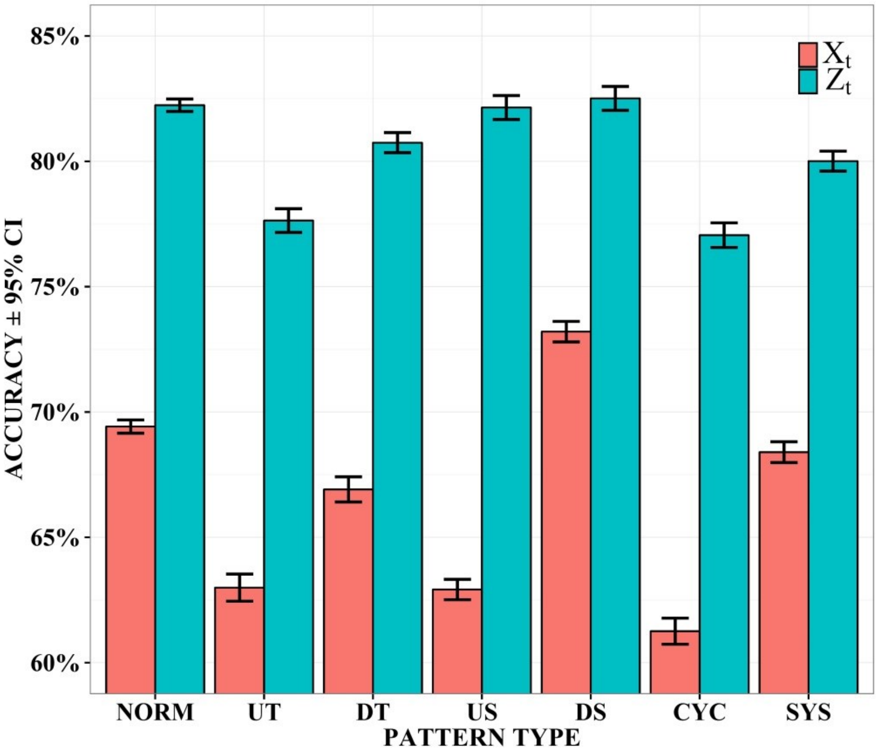

Figure 5.9: Accuracies achieved from the ARMA process, disaggregated by signal to monitor

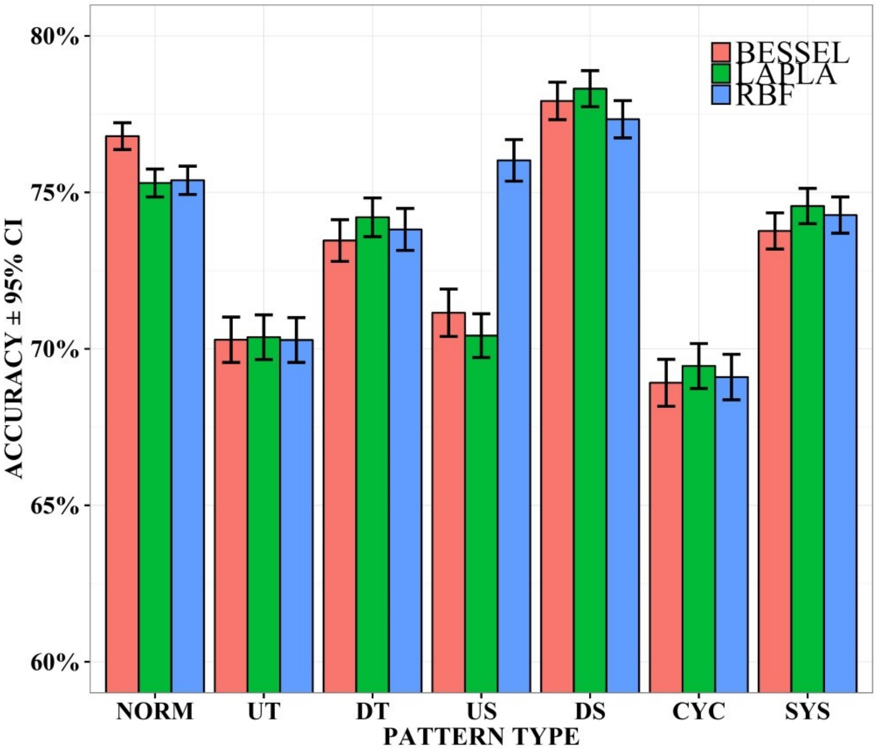

Figure 5.10: Accuracies achieved from the ARMA process, disaggregated by kernel

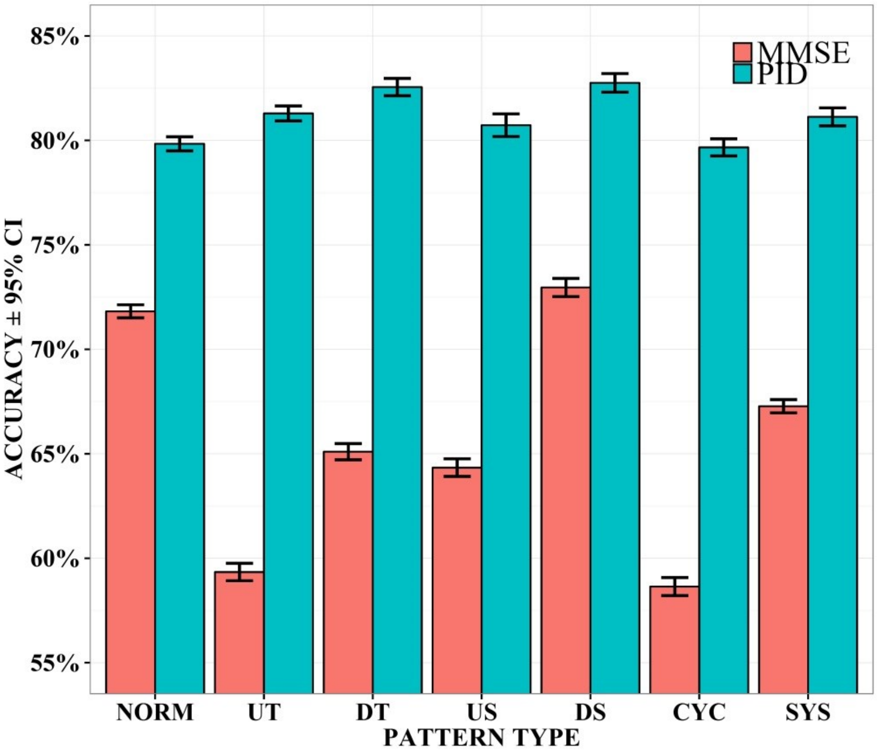

Figure 5.11: Accuracies achieved from the ARMA process, disaggregated by controller type

5.4.2 ANALYSIS OF THE BEST ARRANGEMENT

In order to analyse the accuracies and determine which of the aforementioned five factors affects the recognition accuracy, a 2^4 x3 ANOVA with up to quintuple interactions was utilised. in Table 5.3.

Table 5.2 shows the p-values obtained from the aforementioned ANOVA for the AR(1) and ARMA(1,1) models; the triple, quadruple and quintuple interactions have been omitted.

Based on the results of the ANOVA shown in Table 5.2, a post-hoc Tukey test was obtained. The controller type was taken as the fixed factor and the interaction of the other factors with the inherent noise model was studied. The performance of each model-controller type has been analysed and the results obtained are shown in Table 5.3.

Table 5.2: p-values obtained from ANOVA for the five input factors

Factor	**AR(1)**	**ARMA(1,1)**
PGS	≈0	≈0
IRT	≈0	≈0
SIGNAL	≈0	≈0
KERNEL	≈0	≈0
CONTROLLER	≈0	≈0
PGS*CONTROLLER	≈0	≈0
IRT*CONTROLLER	≈0	≈0
SIGNAL*CONTROLLER	≈0	≈0
KERNEL*CONTROLLER	0.4893	≈0
PGS*IRT	0.1723	≈0
PGS*SIGNAL	≈0	≈0
PGS*KERNEL	0.3730	0.2849
IRT*SIGNAL	≈0	≈0
IRT*KERNEL	≈0	≈0
SIGNAL*KERNEL	0.3614	0.0008

Table 5.3: Best factor arrangements for AR(1) and ARMA(1,1) models

Model-Controller	**Best arrangement**			
	PGS	**IRT**	**Kernel**	**Signal**
AR-PID	PGS-1	Raw	RBF	Z_t
AR-MMSE	PGS-1	Raw / Features	RBF	Z_t
ARMA-PID	PGS-1	Raw	RBF	Z_t
ARMA-MMSE	PGS-1	Raw / Features	RBF / LAPLA	Z_t

Table 5.4 shows the accuracies for each optimal arrangement, disaggregated by controller type and pattern. It can be observed that the accuracies for NORM patterns when a PID

controller was applied were the lowest. Regarding the MMSE controller, the lowest accuracies were observed for CYC patterns.

Table 5.4: Accuracies using the best arrangements of factors (%)

	AR		ARMA	
	PID (%)	MMSE (%)	PID (%)	MMSE (%)
TOTAL	**94.18**	**81.53**	**94.14**	**76.93**
NORM	88.13	82.93	88.95	81.32
UT	93.00	77.66	92.70	70.66
DT	95.99	81.58	96.48	73.24
US	94.12	78.21	93.58	82.79
DS	98.60	90.64	98.04	87.66
CYC	92.63	77.41	93.10	70.12
SYS	96.82	82.25	96.15	72.72

5.5 APPLICATION OF THE CCPR SYSTEM TO REAL DATA

To demonstrate the ability of the proposed CCPR system to handle real data, the thickness of a very thin metallic film in the early stages of the development of an electronic device data taken from Box et al. (2009) were utilised, and the SPC-EPC approach was followed to adjust and monitor this quality characteristic (see Figure 5.12). Box et al. (2009) highlighted the existence of an assignable cause that abruptly increased the metallic film thickness after 30 observations.

The CCPR trained with the best arrangement of input factors for the two signals (X_t and Z_t) were used, and NORM and US patterns were identified for X_t and Z_t, respectively (see Figure 5.13). As mentioned in Box et al. (2009), in the process before the adjustment, it can be observed an US pattern, so this pattern is expected to be recognised in one of the two signals, in this case, X_t.

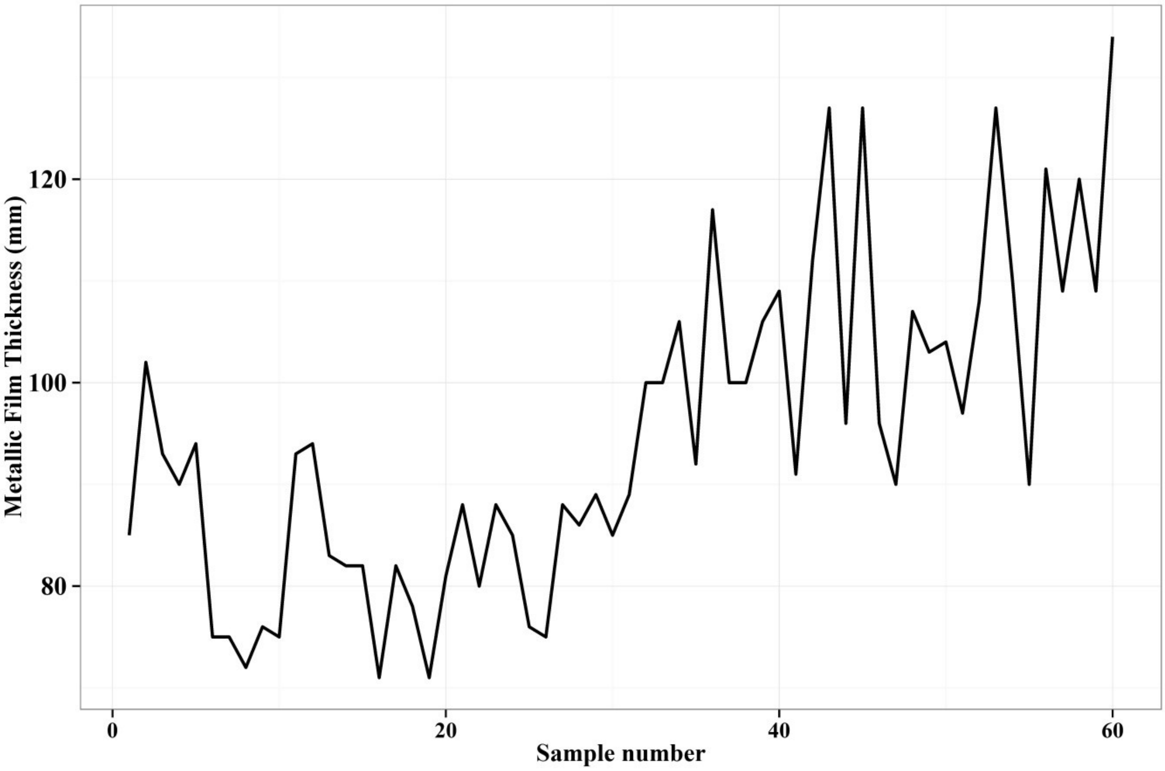

Figure 5.12: Thickness of metallic film in the early stages of the development of an electronic device

In order to further categorise the pattern recognised by the aforementioned CCPR system, the methodology proposed for autocorrelated inherent noise is applied, i.e., two NLM-ARMA proposed for autocorrelated patterns is fitted to the original data. The p-value corresponding to the F-test for nested models is 0.000002; therefore, the full model fits better to the data. The most likely breakpoint was detected at τ = 30. In Table 5.5 are shown all the values regarding the full model fitted. It can be observed that the parameter related to the Shift pattern is statistically significant and greater than zero, thus likewise the CCPR, the NLM-ARMA classifies the pattern as US pattern.

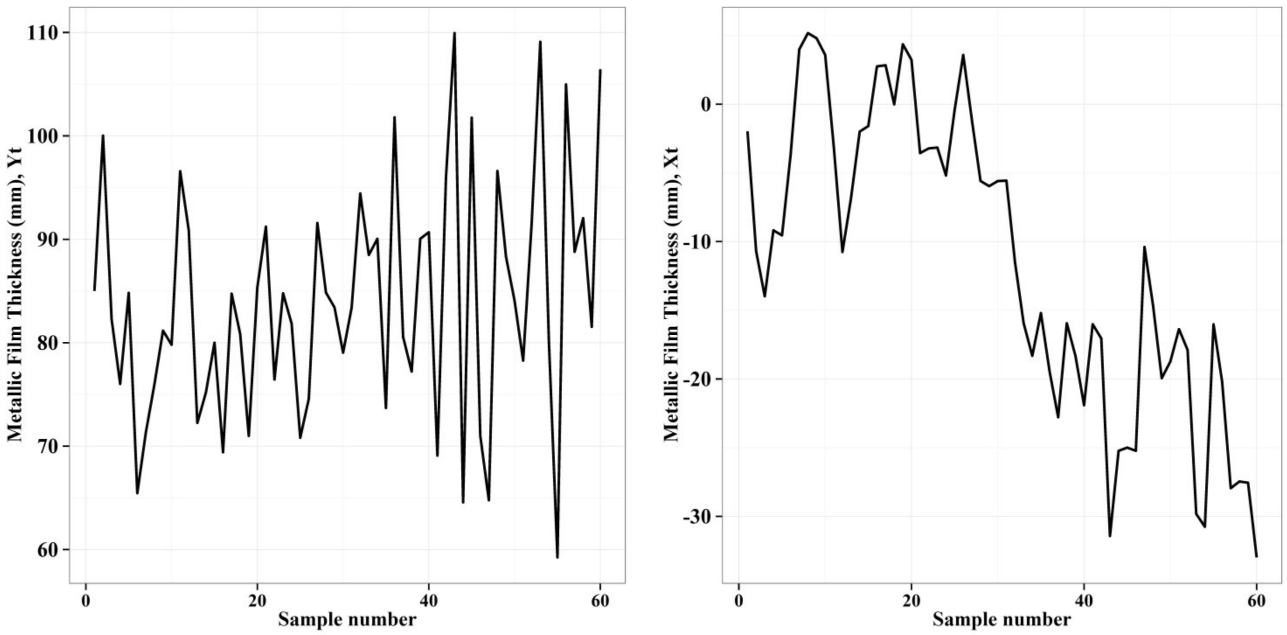

Figure 5.13: Output and controller signals obtained from the SPC-EPC process

Table 5.5: ANOVA of the NLM-ARMA model fitted to the Thickness of metallic film

Parameter	Estimate	Std. Error	t value	p-value
ϕ	-0.2632	0.1075	-2.4484	0.0144
Intercept	80.4461	2.1699	37.0736	0.0000
Slope (Trend)	0.1750	0.0986	1.7748	0.0760
Shift magnitude (Shift)	23.0429	4.0195	5.7328	0.0000
Amplitude (Cyclic)	2.1916	1.4738	1.4870	0.1370
Frequency (Cyclic)	8.2311	1.0233	8.0436	0.0000
Departure (Systematic)	0.1576	1.3958	0.1129	0.9101

5.6 SUMMARY

In this chapter, an entirely new CCPR system for feedback-controlled processes has been proposed. As in previous chapters, special attention was paid to the synthesis of patterns and the effect of the proposed PGS on recognition accuracies. As found in the literature review, the formal identification of patterns in this process type is still an unstudied problem; therefore, it was necessary to determine the best arrangement of input factors of the CCPR system.

Firstly, a method for synthesising patterns for feedback-controlled processes is presented. In this method, the importance of the PGS scheme proposed in chapter 4 is highlighted. Also, two time series and two feedback controllers were studied, to cover a wide variety of patterns and make each of the proposed CCPR systems more robust. The PID and MMSE controllers were used and compared. The CCPR system trained when the process was controlled by the PID controller achieved the highest accuracy.

Also, as a consequence of the application of controllers, two new signals were produced, namely the output of the process after the controller and the controller performance. The determination of which of these two signals to monitor in order to identify patterns was another contribution of this work shown in this chapter. The output of the feedback-controlled process achieved the highest accuracies.

Finally, the CCPR system trained with the PID-controlled AR(1) process was used to identify patterns in a real example, namely, the control of the thickness of a metallic film in the early stages of the development of an electronic device. A US pattern was correctly identified when the X_t signal was monitored and the best arrangement of factors was used. By fitting a

NLM-ARMA model to the original data, a US pattern of magnitude $\beta_2 = 23.0429$ at time τ = 30 was observed, thus confirming what was observed by the CCPR system.

6 CONCLUSION

6.1 PRELIMINARIES

This chapter summarises the conclusions of this research and highlights the contributions made. The chapter also provides suggestions for further research.

6.2 CONCLUSIONS

6.2.1 CCPR FOR NIID PROCESSES

The literature review identified that there were no standard methods for generating control chart patterns. In the proposed PGS, all the pattern parameters are randomly assigned to the patterns, i.e., even parameters such as break point position and cycle period were randomised. Also, in the literature review, it was found that these two parameters had been ignored by all authors, despite that fact they are of interest in root cause analysis.

Another issue studied in chapter 3 was the determination of the minima and maxima of pattern parameters during pattern generation. Finding an objective method to set the range of parameter values was an aim of this research. This issue was addressed by nesting two NLMs, with the p-value of the related parameter determining the pattern class. Without the objective method proposed here, by using different parameter ranges during pattern generation, different decision boundaries would be estimated. This makes the recognition accuracies achieved not comparable and is a common mistake found in the CCPR literature.

To design the proposed PGS, it was necessary to develop a robust procedure for identifying and categorising break points in the mean value in control charts. Such a method not only detects the potential existence of sudden changes in the mean but also statistically estimates the magnitude of these changes. The proposed scheme is also able to handle noise as the estimation of the p-values employed during pattern categorisation is based on the ratio of the estimated parameters and their standard errors.

The performance of the proposed PGS during the initial generation of patterns was measured. It was found that, as the significance level was increased, the percentage of discarded patterns also increased. The percentage of reclassified patterns remained approximately constant for the three α values tested. Thus, the significance level mainly affects the number of patterns to be discarded and it is necessary to generate more patterns initially if a significance level is required.

During the analysis of the performance of the proposed PGS at three different significance levels, two alternative pattern recognition systems were presented: SVM and PNN. In the case of SVM with the proposed PGS, the pattern recognition accuracy was statistically significantly increased. In the case of PNN, the mean accuracy also increased. Furthermore, it was observed in both recognition systems that increases of 0.01 in the significance level did not significantly affect the pattern recognition accuracy. However, when the increment was from $\alpha=0.01$ to $\alpha=0.03$, a statistically significant decrease in the mean accuracy was observed in the PNN.

As previously noted, the significance level set during pattern generation mainly affected the number of patterns discarded, and had a small effect on the pattern recognition accuracy of the two tested ML algorithms. It is recommended to use low significance levels such as 0.01 in order to generate fewer patterns and reduce computational efforts.

As mere identification of patterns is sometimes not enough for efficient root cause analysis, further information related to the identified pattern needs to be extracted. Since the CCPR system was trained using patterns that ensured the estimation of correct decision boundaries, generality of the model and statistical significance of the model parameters, details such as cycle amplitude, periodicity, slope, shift magnitude, change point position and systematic departure can be obtained by fitting a NLM to the control chart data as implemented in the proposed PGS.

6.2.2 CCPR FOR AUTOCORRELATED PROCESSES

The literature review uncovered very few papers dealing with monitoring of non-NIID production processes, the AR(1) model being the only model studied. First-order stationary models are used to represent many continuous production processes. It is necessary to develop robust CCPR systems that allow the automated identification of abnormal patterns when these types of processes are monitored using control charts.

The proposed PGS fulfils three conditions: generality, comparability and facilitation of the extraction of further information from the pattern. Generality was achieved by employing a broad range of pattern parameters during the initial pattern generation and the total randomisation of parameters, including the break point position and the amplitude of CYC patterns. Comparability between different studies is possible as the proposed PGS is a standard technique for producing the same data for training CCPR systems, requiring only the significance level to be set. Further information (for example, the amplitude of cycles and the autocorrelation level) regarding the abnormal pattern identified can be extracted as the decision boundaries were estimated using patterns for which the statistical significance of the parameters was tested. Therefore, the CCPR system can categorise the presented patterns into the class for which the related parameter is the most significant when a NLM-ARMA model is fitted. Furthermore, by fitting this dynamic model to the control chart data, the PGS is able to divide signals into inherent noise and fault signal.

From the first part of the recognition accuracy analysis, of the four different PGS ($\alpha = 0.01$, 0.02, 0.03 and conventional PGS), the proposed PGS significantly increased the pattern recognition accuracy for the three models used. Regarding the IRT, the shape features and raw data achieved similar accuracies in the MA and ARMA models, but lower accuracies were obtained with shape features when the AR process was utilised. Two kernels, RBF and Bessel, showed the greatest recognition accuracies.

In the second part of the analysis, it was found that the values of θ and ϕ greatly affected the pattern recognition accuracy. For the MA and ARMA processes, the lowest accuracy was found when $\theta \geq 0.70$, and the highest accuracy was achieved when $-0.5 < \theta \leq -0.3$. For AR and ARMA processes, the same behaviour was observed, with the lowest accuracy obtained when $\phi \geq 0.70$, and the highest accuracy when $-0.5 < \phi \leq -0.3$. Regarding the patterns, the Normal and US types were those that gave the lowest accuracies.

To measure the impact of the autocorrelation level (ϕ) on the accuracies achieved by CCPR systems, the accuracies obtained by the CCPR trained with autocorrelated processes (see Table 4.6, RBF kernel) can be compared to those obtained with the CCPR trained with only non-autocorrelated (NIID) patterns (see Table 3.5, RBF kernel). Table 6.1 shows the differences in accuracies between these two process types, disaggregated by pattern type and time series models. It can be observed that Shift patterns are those greatly affected in both time series models. Regarding the US pattern, the accuracy decreased when the CCPR was trained only with NIID patterns for the AR and ARMA models. For the DS pattern, an accuracy increase was observed for the AR and ARMA models.

Table 6.1: Differences between NIID model and autocorrelated models, ϕ parameter (%)

	NIID	ϕ			
		AR		ARMA	
		ACCURACY	DIFFERENCE	ACCURACY	DIFFERENCE
OVERALL	**92.87**	**90.03**	**2.84**	**89.08**	**3.79**
NORM	87.43	83.92	3.51	80.61	6.82
UT	93.31	91.08	2.23	89.63	3.68
DT	95.44	93.6	1.84	93.17	2.27
US	91.5	79.7	11.80	82.43	9.07
DS	92.74	96.64	-3.90	95.55	-2.81
CYC	94.01	90.97	3.04	89.44	4.57
SYS	95.67	94.28	1.39	92.77	2.90

On the other hand, to measure the impact of the moving-average level (θ) on the recognition accuracy of the CCPR system, the accuracies obtained from MA and ARMA processes shown in Table 4.7 (RBF kernel) are compared to those obtained when the CCPR was trained only with NIID pa tterns (see Table 3.5). These accuracies are disaggregated by pattern type and time series model and shown in Table 6.2. It can be observed that the Shift pattern types are again the most affected. Regarding the DS pattern, the accuracy increased when the CCPR was trained only with NIID patterns for the MA and ARMA models. For the US pattern, a decrease in accuracy was found for the MA and ARMA models.

Table 6.2: Accuracies and differences between NIID model and autocorrelated models, θ parameter (%)

	NIID	θ			
		MA		ARMA	
		ACCURACY	DIFFERENCE	ACCURACY	DIFFERENCE
OVERALL	**92.87**	**91.47**	**-1.4**	**89.48**	**-3.39**
NORM	87.43	84.86	-2.57	81.47	-5.96
UT	93.31	92.92	-0.39	89.98	-3.33
DT	95.44	94.39	-1.05	93.42	-2.02
US	91.5	82.66	-8.84	83.65	-7.85
DS	92.74	97.75	5.01	95.71	2.97
CYC	94.01	93.15	-0.86	89.5	-4.51
SYS	95.67	94.59	-1.08	92.65	-3.02

6.2.3 FEEDBACK-CONTROLLED PROCESSES

An important aspect to pay attention to when developing CCPR models is that they must be general, i.e., able to identify a wide variety of patterns. The generation of training patterns to ensure generality of the CCPR model and to create benchmarks for recognition accuracies is an issue discussed in chapters 3 and 4 which describe the development of PGSs with that specific aim. The PGS proposed for autocorrelated patterns was the one adopted in chapter 4 for pattern generation for feedback-controlled processes. The control chart patterns, Z_t, synthesised using of two different PGSs, PGS-1 and PGS-C. PGS-1, proposed in chapter 4, ensured that the correct decision boundaries were estimated and the patterns were correctly categorised before the application of the controller. Therefore, the models developed here represent general CCPR models and can be used in real applications.

Synergistic control (also known as SPC-EPC control), which combines SPC and feedback control, is a quality improvement technique that has shown good performance in a variety of production systems. However, automatic recognition of control charts patterns has not been formally studied for cases where feedback controllers are used to reduce the variability of the output. The design of CCPR for feedback-controlled processes is described in chapter 5. This task involves several factors such as IRT and signals to be monitored. Determining the best arrangement of these factors was the focus of that chapter. Another factor that can be set by production personnel and also studied there is the controller type, with PID and MMSE being the two most commonly adopted.

In the case of the AR(1) model, employing a PID controller and monitoring the output variable using the raw data as IRT was the arrangement that yielded the highest pattern recognition accuracies. In the case of the MMSE controller, the raw data used as IRT also gave the best accuracies. The RBF, LAPLA and Bessel kernels showed similar accuracies for both controllers.

When the ARMA(1,1) model was used, employing raw data also produced the highest pattern recognition accuracy for both controllers. As in the case of the AR(1) model, the RBF, LAPLA and Bessel kernels showed similar accuracies for both controllers.

To measure the impact of the controllers, the accuracies achieved using the best arrangement of factors (see Table 5.4) were compared with those achieved where no controllers were utilised (see Tables 4.6 and 4.7). The overall accuracy of the ARMA model represents the mean accuracy considering the two parameters tables (Table 4.6 and Table 4.7). The results of the comparison are shown i n Table 6.3.

Table 6.3 Comparison of accuracies with and without controller application (%)

	AR			ARMA		
	No controller	PID	Difference	No controller	PID	Difference
OVERALL	**90.03**	**94.18**	**4.15**	**89.28**	**94.14**	**4.86**
NORM	83.92	88.13	4.21	81.04	88.95	7.91
UT	91.08	93	1.92	89.81	92.7	2.9
DT	93.6	95.99	2.39	93.3	96.48	3.19
US	79.7	94.12	14.42	83.04	93.58	10.54
DS	96.64	98.6	1.96	95.63	98.04	2.41
CYC	90.97	92.63	1.66	89.47	93.1	3.63
SYS	94.28	96.82	2.54	92.71	96.15	3.44

As shown in Table 6.3, the overall accuracy increased when the controller was used with the AR and ARMA processes. Regarding the pattern type, the US pattern showed the largest improvement in recognition accuracy. This could be caused by the amplification effect of the controller on the magnitude of abnormal patterns.

6.3 CONTRIBUTIONS

- Two PGSs where correct decision boundaries are set and all the pattern parameters are randomised are presented. These PGSs are robust to the pattern parameter ranges used in the initial pattern generation as the reclassification of patterns is based on the statistical significance of the pattern parameter. In order to develop these PGSs, it was necessary to fit a NLM and a NLM-ARMA model to the CC data (this addresses to objective no. 1).
- The best arrangement of input factors of the CCPR systems was determined for each of the three studied processes (this addresses to objective no. 7).
- The impact of the proposed PGSs on the recognition accuracies achieved by two CCPR systems for NIID processes was measured. One system employed SVMs as recognition algorithm, and the other used PNNs (this addresses to objective no. 6 and 7).
- The effect of pattern autocorrelation on the recognition accuracies was also measured. The evaluation was performed when three time-series models were used to represent the inherent noise. Two of the models (MA and ARMA) had never been investigated in the CCPR domain (this addresses to objective no. 1 and 2).
- A novel CCPR system for feedback-controlled processes was proposed. As the identification of the seven simple patterns had never been formally studied, it was necessary to develop a standard procedure to generate the patterns to be identified by the ML algorithm (this addresses to objective no. 1, 3, 4, 5 and 6).
- The best combination of input factors of the CCPR system for feedback-controlled processes was determined (this addresses to objective no. 7).

6.4 FURTHER RESEARCH

Based on the results obtained and the literature review, new research directions in the CCPR domain have been identified. Some of those are listed below.

- Research concerning recognition/prediction schemes for patterns that are "slow" or "weak" in a specific inspection window, thus enabling the appearance of a pattern to be anticipated and pre-emptive actions, such as predictive maintenance or repairs, to be taken.
- Work aimed at increasing recognition accuracy beyond the currently achievable 90-95%. Signal processing techniques such as Independent Component Analysis and Multiresolution filters could help this effort.
- Development of PGSs and CCPR systems for processes with inherent noise modelled by non-stationary time series models, such as IMA and autoregressive integrated moving-average (ARIMA) models. The effect of autocorrelation and moving-average levels should also be measured.
- Application of other ML algorithms such as Random Forest and Deep Learning and comparison of the results obtained with those reported here.
- Application of the T^2 control chart to simultaneously monitor the output and controller performance in feedback-controlled processes.
- Identification of combined control chart patterns, e.g. an upward trend with cyclic variations.
- Identification of patterns in multivariate control charts. This research should be extended to the three pattern types studied in this work, NIID, autocorrelated and feedback-controlled processes. Furthermore, the effect of the covariance matrix should be studied.

BIBLIOGRAPHY

Abe, S. (2003). Analysis of multiclass support vector machines. In Thyroid (Vol. 21, pp. 385–396).

Al-Assaf, Y. (2004a). Multi-resolution wavelets analysis approach for the recognition of concurrent control chart patterns. Quality Engineering, 17(1), 11–21.

Al-Assaf, Y. (2004b). Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks. Computers & Industrial Engineering, 47(1), 17–29.

Alwan, L. C. (1991). Autocorrelation: Fixed versus variable control limits. Quality Engineering, 4(2), 167–188.

Bag, M., Gauri, S. K., & Chakraborty, S. (2012). An expert system for control chart pattern recognition. International Journal of Advanced Manufacturing Technology, 62(1-4), 291–301.

Barghash, M. A., & Santarisi, N. S. (2004). Pattern recognition of control charts using artificial neural networks—analyzing the effect of the training parameters. Journal of Intelligent Manufacturing, 15(5), 635–644.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). Time Series Analysis: Forecasting & Control. Pearson Education India.

Box, G. E. P., Luceño, A., & Paniagua-Quiñones, M. del C. (2009). Statistical control by monitoring and adjustment. John Wiley & Sons.

Box, G., & Kramer, T. (1992). Statistical Process Monitoring and Feedback Adjustment: A Discussion. Technometrics, 34(3), 251–267.

Boyles, R. A. (2000). Phase I Analysis for Autocorrelated Processes. Journal of Quality Technology, 32(4), 395–409.

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 121–167.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). Self-organisation in biological systems. Princetown studies in complexity. Princeton University Press.

Capilla, C., Ferrer, A., Romero, R., & Hualda, A. (1999). Integration of Statistical and Engineering Process Control in a Continuous Polymerization Process. Technometrics, 41(1), 14–28.

Castellani, M. (2013). Evolutionary generation of neural network classifiers-An empirical comparison. Neurocomputing, 99, 214–229.

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

Cheng, H. P., & Cheng, C. S. (2008). Denoising and feature extraction for control chart pattern recognition in autocorrelated processes. International Journal of Signal and Imaging Systems Engineering, 1(2), 115–126.

Cheng, H. P., & Cheng, C. S. (2009). Control chart pattern recognition using wavelet analysis and neural networks. Journal of Quality, 16(5), 311–321.

Chinnam, R. B. (2002). Support vector machines for recognizing shifts in correlated and other manufacturing processes. International Journal of Production Research, 40(17), 4449–4466.

Chiu, C. C., Chen, M. K., & Lee, K. M. (2001). Shifts recognition in correlated process data using a neural network. International Journal of Systems Science, 32(2), 137–143.

Chompu-Inwai, R., & Thaiupathump, T. (2015). Improved ICA-based mixture control chart patterns recognition using shape related features. In 2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings (pp. 484–489). IEEE.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

De Tejada, J. G. S., & Martìnez-Echevarrìa, J. S. (2007). Support Vector Machines. In Computational Intelligence (pp. 147–191). Boston, MA: Springer US.

Del Castillo, E. (2006). Statistical process adjustment: A brief retrospective, current status, and some opportunities for further work. Statistica Neerlandica, 60(3), 309–326.

Draper, N., Smith, H., & Pownell, E. (1966). Applied regression analysis (Vol. 3). Wiley New York.

Du, S. C., Huang, D. L., & Lv, J. (2013). Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines. Computers & Industrial Engineering, 66(4), 683–695.

Dyer, J. N., Adams, B. M., & Conerly, M. D. (2003). The Reverse Moving Average Control Chart for Monitoring Autocorrelated Processes. Journal of Quality Technology, 35(2), 139–152.

Frisch, K. Von. (1971). Bees: their vision, chemical senses, and language., 157.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D.

(2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16(10), 906 –914.

Gallant, A. R. (1987). Nonlinear Statistical Models. New York, (1980), 595–600.

Gauri, S. K. (2010). Control chart pattern recognition using feature-based learning vector quantization. International Journal of Advanced Manufacturing Technology, 48(3), 1061–1073.

Gauri, S. K. (2012). Improved feature-based test statistic for assessing suitability of the preliminary samples for constructing control limits of $\bar{X}$ chart. International Journal of Advanced Manufacturing Technology, 58(9-12), 1171–1187.

Gauri, S. K., & Chakraborty, S. (2006a). A study on the various features for effective control chart pattern recognition. The International Journal of Advanced Manufacturing Technology, 34(3-4), 385–398.

Gauri, S. K., & Chakraborty, S. (2006b). Feature-based recognition of control chart patterns. Computers and Industrial Engineering, 51(4), 726–742.

Gauri, S. K., & Chakraborty, S. (2009). Recognition of control chart patterns using improved selection of features. Computers and Industrial Engineering, 56(4), 1577–1588.

Gerbec, D., Gašperič, S., Šmon, I., & Gubina, F. (2005). Allocation of the load profiles to consumers using probabilistic neural networks. IEEE Transactions on Power Systems, 20(2), 548–555.

Gu, N., Cao, Z. Q., Xie, L. J., Creighton, D., Tan, M., & Nahavandi, S. (2013). Identification of concurrent control chart patterns with singular spectrum analysis and learning vector quantization. Journal of Intelligent Manufacturing, 24(1), 1241–1252.

Guh, R. S. (2003). Integrating artificial intelligence into on-line statistical process control. Quality and Reliability Engineering International, 19(1), 1–20.

Guh, R. S. (2004). Optimizing Feed Forward Neural Networks for Control Chart Pattern Recognition Through Genetic Algorithms. International Journal of Pattern Recognition and Artificial Intelligence, 18(02), 75–99.

Guh, R. S. (2005). A hybrid learning-based model for on-line detection and analysis of control chart patterns. Computers & Industrial Engineering, 49(1), 35–62.

Guh, R. S. (2008). Real-time recognition of control chart patterns in autocorrelated processes using a learning vector quantization network-based approach. International Journal of Production Research, 46(14), 3959–3991.

Guh, R. S., & Shiue, Y.-R. (1999). A neural network based model for abnormal pattern recognition of control charts. Computers & Industrial Engineering, 36(1), 97–108.

Guh, R. S., & Tannock, J. D. T. (1999). A neural network approach to characterize pattern parameters in process control charts. Journal of Intelligent Manufacturing, 10(5), 449–462.

Gunn, S. R. (1998). Support Vector Machines for Classification and Regression. University of Southhampton, Technical Report. Retrieved from

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991-2010) based on a new conceptual classification scheme. Computers and Industrial Engineering, 63(1), 204–222.

Hassan, A. (2011). An improved scheme for online recognition of control chart patterns. International Journal of Computer Aided Engineering and Technology, 3(3/4), 309.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of Methods for Multiclass Support Vector Machines. Neural Networks, IEEE Transactions on, 13(2), 415–425.

Hua, S., & Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. Bioinformatics, 17(8), 721–728.

Hwarng, H. B. (2004). Detecting process mean shift in the presence of autocorrelation: a neural-network based monitoring scheme. International Journal of Production Research, 42(3), 573–595.

Hwarng, H. B. (2005). Simultaneous identification of mean shift and correlation change in AR(1) processes. International Journal of Production Research, 43(9), 1761–1783.

Hyndman, R. J., & Athanasopoulos, G. (2014). Forecasting: principles and practice. OTexts.

Jiang, P., Liu, D., & Zeng, Z. (2009). Recognizing control chart patterns with neural network and numerical fitting. Journal of Intelligent Manufacturing, 20(6), 625–635.

Jiang, W., & Farr, J. V. (2007). Integrating SPC and EPC Methods for Quality Improvement. Quality Technology & Quantitative Management, 4(3), 345–363.

Jiang, W., & Tsui, K.-L. (2002). SPC Monitoring of MMSE- and PI-Controlled Processes. Journal of Quality Technology, 34(4), 384–398.

Kandananond, K. (2010). Effectively monitoring the performance of integrated process control systems under nonstationary disturbances. International Journal of Quality, Statistics, and Reliability, 2010.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab – An S4 Package for Kernel Methods in R. Journal of Statistical Software.

Kazemi, M. S., Kazemi, K., Yaghoobi, M. A., & Bazargan, H. (2015). A hybrid method for estimating the process change point using support vector machine and fuzzy statistical clustering. Applied Soft Computing, 40, 507–516.

Kim, D., Kim, D. H., & Chang, S. (2008). Application of probabilistic neural network to design breakwater armor blocks. Ocean Engineering, 35(3-4), 294–300.

Kramer, C., Mckay, B., Belina, J., & Hall, P. (1995). Probabilistic Neural Network Array Architecture for ECG Classification. In Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference (Vol. 1, pp. 807–808). IEEE.

Lesany, S. A., Koochakzadeh, A., & Fatemi Ghomi, S. M. T. (2013). Recognition and classification of single and concurrent unnatural patterns in control charts via neural networks and fitted line of samples. International Journal of Production Research, 52(6), 1771–1786.

Lin, S.-Y., Guh, R.-S., & Shiue, Y.-R. (2011). Effective recognition of control chart patterns in autocorrelated data using a support vector machine based approach. Computers & Industrial Engineering, 61(4), 1123–1134.

Longnecker, M. T., & Ryan, T. P. (1992). Charting correlated process data. Texas A&M University Department of Statistics Technical Report, 166.

Lu, C.-J., Shao, Y. E., & Li, P.-H. (2011). Mixture control chart patterns recognition using independent component analysis and support vector machine. Neurocomputing, 74(11), 1908–1914.

Lu, C.-J., Wu, C.-M., Keng, C.-J., & Chiu, C.-C. (2008). Integrated Application of SPC/EPC/ICA and neural networks. International Journal of Production Research, 46(4), 873–893.

Lu, C.-W., & Reynolds Jr., M. R. (1999). Control Charts for Monitoring the Mean and Variance of Autocorrelated Processes. Journal of Quality Technology, 31(3), 259–274.

Mao, K. Z., Tan, K. C., & Ser, W. (2000). Probabilistic neural-network structure determination for pattern classification. IEEE Transactions on Neural Networks, 11(4), 1009–1016.

Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation, 8(1), 3–30.

McQuarrie, R. A. D., & Tsai, C.-L. (1998). Regression and time series model selection (Vol. 43). World Scientific Singapore.

Milgram, J., Cheriet, M., & Sabourin, R. (2006). “One Against One” or “One Against All”: Which One is Better for Handwriting Recognition with SVMs? In Tenth International Workshop on Frontiers in Handwriting Recognition (pp. 1–6).

Montgomery, D. (2009). Introduction to statistical quality control. John Wiley & Sons Inc.

Montgomery, D. C., Keats, J. B., Yatskievitch, M., & Messina, W. S. (2000). Integrating Statistical Process Monitoring. Quality and Reliability Engineering International, 16(September 1999), 515–525.

Musavi, M. T., Hummels, D. M., Kalantri, K., & Chan, K. H. (1994). On the Generalization Ability of Neural Network Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 659–663.

Nembhard, H. B., & Kao, M. S. (2003). Adaptive Forecast-Based Monitoring for Dynamic Systems. Technometrics, 45(3), 208–219.

Noorossana, R., Farrokhi, M., & Saghaei, A. (2003). Using Neural Networks to Detect and Classify Out-of-control Signals in Autocorrelated Processes. Quality and Reliability Engineering International, 19(6), 493–504.

Pacella, M., Semeraro, Q., & Anglani, A. (2004). Manufacturing quality control by means of a Fuzzy ART network trained on natural process data. Engineering Applications of Artificial Intelligence, 17(1), 83–96.

Pande, A., & Abdel-Aty, M. (2008). A computing approach using probabilistic neural networks for instantaneous appraisal of rear-end crash risk. Computer-Aided Civil and Infrastructure Engineering, 23(7), 549–559.

Pankratz, A. (1991). Forecasting with Dynamic Regression Models. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Pham, D.T., & Oztemel, E. (1996). Intelligent quality systems. Springer Science & Business Media.

Pham, D. T., & Castellani, M. (2014). Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms. Soft Computing, 18(5), 871–903.

Pham, D. T., & Chan, A. B. (2005). Control chart pattern recognition using a new type of self-organizing neural network. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 212(2), 115–127.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The Bees Algorithm - A Novel Tool for Complex Optimisation Problems. In Intelligent

Production Machines and Systems - 2nd I*PROMS Virtual International Conference 3-14 July 2006 (pp. 454–459).

Pham, D. T., Otri, S., Ghanbarzadeh, A., & Koc, E. (2006). Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition. 2006 2nd International Conference on Information & Communication Technologies, 1, 1–5.

Pham, D. T., & Wani, M. A. (1997). Feature-based control chart pattern recognition. International Journal of Production Research, 35(7), 1875–1890.

Pham, Q. T., Pham, D. T., & Castellani, M. (2012). A modified bees algorithm and a statistics-based method for tuning its parameters. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 226(3), 287–301.

Pontil, M., & Verri, A. (1998). Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6), 637–646.

Psarakis, S., & Papaleonida, G. (2007). SPC procedures for monitoring autocorrelated processes. Quality Technology & Quantitative Management, 4(4), 501–540.

Quan, J., Wen, X., & Xu, X. (2008). Multiscale probabilistic neural network method for SAR image segmentation. Applied Mathematics and Computation, 205(2), 494–499.

R Core Development Team. (2015). R: A language and environment for statistical computing, 3.2.1.

Ranaee, V., Ebrahimzadeh, A., & Ghaderi, R. (2010). Application of the PSO-SVM model for recognition of control chart patterns. ISA Transactions, 49(4), 577–86.

Rawlings, J. O., Pantula, S. G., & Dickey, D. A. (2006). Applied Regression Analysis: A Research Tool. Springer Science & Business Media.

Romero, R., Touretzky, D., & Thibadeau, R. (1997). Optical Chinese character recognition using probabilistic neural networks. Pattern Recognition, 30(8), 1279–1292.

Ross, G. J. (2014). Sequential change detection in the presence of unknown parameters. Statistics and Computing, 24(6), 1017–1030.

Shaban, A., & Shalaby, M. A. (2012). A double neural network approach for the identification and parameter estimation of control chart patterns. International Journal of Quality Engineering and Technology, 3(2), 124.

Shao, Y. E. (2014). Recognition of Process Disturbances for an SPC / EPC Stochastic System Using Support Vector Machine and Artificial Neural Network Approaches. In Abstract and Applied Analysis (Vol. 2014). Hindawi Publishing Corporation.

Shao, Y. E., & Chiu, C. C. (1999). Developing identification techniques with the integrated use of SPC/EPC and neural networks. Quality and Reliability Engineering International, 15(4), 287–294.

Shao, Y. E., Lu, C. J., & Chiu, C. C. (2011). A fault detection system for an autocorrelated process using SPC/EPC/ANN AND SPC/EPC/SVM schemes. International Journal of Innovative Computing, Information and Control, 7(9), 5417–5428.

Shumway, R. H., & Stoffer, D. S. (2011). Time Series Analysis and Its Applications With R Examples. Media.

Shuttleworth, A., & Johnson, S. D. (2010). The missing stink: sulphur compounds can mediate a shift between fly and wasp pollination systems. Proceedings. Biological sciences / The Royal Society (Vol. 277). Harvard University Press.

Song, T., Jamshidi, M. M., Lee, R. R., & Huang, M. (2007). A modified probabilistic neural network for partial volume segmentation in brain MR image. IEEE Transactions on Neural Networks, 18(5), 1424–1432.

Specht, D. F. (1990). Probabilistic neural networks. Neural Networks, 3(1), 109–118.

Specht, D. F. (1992). Enhancements to Probabilistic Neural Networks. In Proceedings of the International Joint Conference on Neural Networks (Vol. 1, pp. 761–768). IEEE.

Sun, Y.-N., Horng, M.-H., Lin, X.-Z., & Wang, J.-Y. (1996). Ultrasonic lmaae Analysis for Liver Diagnosis. IEEE Engineering In Medicine and Biology, 15(6), 93–101.

Tong, S., & Koller, D. (2001). Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 2, 45–66.

Tsung, F., & Shi, J. (1999). Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection. IIE Transactions (Institute of Industrial Engineers), 31(6), 517–527.

Tsung, F., Wu, H., & Nair, V. N. (1998). On the Efficiency and Robustness of Discrete Proportional-Integral Control Schemes. Technometrics, 40(3), 214–222.

Übeyli, E. D. (2010). Lyapunov exponents/probabilistic neural networks for analysis of EEG signals. Expert Systems with Applications, 37(2), 985–992.

Wang, C.-H., Guo, R.-S., Chiang, M.-H., & Wong, J.-Y. (2008). Decision tree based control chart pattern recognition. International Journal of Production Research, 46(17), 4889–4901.

Wang, C.-H., & Kuo, W. (2007). Identification of control chart patterns using wavelet filtering and robust fuzzy clustering. Journal of Intelligent Manufacturing, 18(3), 343–350.

Wang, C.-H., Kuo, W., & Qi, H. (2007). An integrated approach for process monitoring using wavelet analysis and competitive neural network. International Journal of Production Research, 45(1), 227.

Wang, K., & Tsung, F. (2007). Monitoring feedback-controlled processes using adaptive T2 schemes. International Journal of Production Research, 45(23), 5601–5619.

Western Electric Company. (1956). Statistical Quality Control Handbook.

Woodall, W. H., & Faltin, F. W. (1993). Autocorrelated data and SPC. ASQC Statistics Division Newsletter, 13(4), 18–21.

Wu, B., & Yu, J. B. (2010). A neural network-based on-line monitoring model of process mean and variance shifts. Proceedings of the International Conference on E-Business and E-Government, ICEE 2010, 37(6), 2615–2618.

Wu, S. (2006). Abnormal Pattern Parameters Estimation of Control Chart Based on Wavelet Transform and Probabilistic. In Intelligent Computing in Signal Processing and Pattern Recognition (pp. 112–119). Springer.

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. Computers & Industrial Engineering, 70, 134–149.

Xie, L., Gu, N., Li, D., Cao, Z., Tan, M., & Nahavandi, S. (2013). Concurrent control chart patterns recognition with singular spectrum analysis and support vector machine. Computers & Industrial Engineering, 64(1), 280–289.

Yang, J.-H. J.-H., & Yang, M.-S. M.-S. (2005). A control chart pattern recognition system using a statistical correlation coefficient method. Computers & Industrial Engineering, 48(2), 205–221.

Yang, W. A., & Zhou, W. (2015). Autoregressive coefficient-invariant control chart pattern recognition in autocorrelated manufacturing processes using neural network ensemble. Journal of Intelligent Manufacturing, 26(6), 1161–1180.

Yu, J. B. (2012). Gaussian mixture models-based control chart pattern recognition. International Journal of Production Research, 50(23), 6746–6762.

Yuce, B., Packianather, M. S., Mastrocinque, E., Pham, D. T., & Lambiase, A. (2013). Honey bees inspired optimization method: The bees algorithm. Insects, 4(4), 646–662.

Zhang, N. F. (1997). Detection capability of residual control chart for stationary process data. Journal of Applied Statistics, 24(4), 475–492.

Zhang, N. F., & Pollard, J. F. (1994). Analysis of Autocorrelations Processes in Dynamic. Technometrics, 36(4), 354–368.

Zobel, C. W., Cook, D. F., & Nottingham, Q. J. (2004). An augmented neural network classification approach to detecting mean shifts in correlated manufacturing process parameters. International Journal of Production Research, 42(4), 741–758.

Zorriassatine, F., & Tannock, J. D. T. (1998). A review of neural networks for statistical process control. Journal of Intelligent Manufacturing, 9(3), 209–224.