University of Alberta

Plant-wide Performance Monitoring and Controller Prioritization

by

Samidh Pareek

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Process Control

Department of Chemical and Materials Engineering

Spring 2011
Edmonton, Alberta

To Mummy, Papa and Tinni

Abstract

Plant-wide performance monitoring has generated a lot of interest in the control engineering community. The idea is to judge the performance of a plant as a whole rather than looking at performance of individual controllers. Data based methods are currently used to generate a variety of statistical performance indices to help us judge the performance of production units and control assets. However, so much information can often be overwhelming if it lacks precise information. Powerful computing and data storage capabilities have enabled industries to store huge amounts of data. Commercial performance monitoring softwares such as those available from many vendor companies such as Honeywell, Matrikon, ExperTune etc typically use this data to generate huge amounts of information. The problem of data overload has in this way turned into an information overload problem. This work focuses on developing methods that reconcile these various statistical measures of performance and generate useful diagnostic measures in order to optimize process performance of a unit/plant. These methods are also able to identify the relative importance of controllers in the way that they affect the performance of the unit/plant under consideration.

Acknowledgements

The two years that I have spent at the University of Alberta will be etched in my memory forever. Along with being a rich experience academically, these years have helped shape my character and grow as a person. I am highly indebted to the University of Alberta and people of Canada for all the love & support that they have given me during my time here.

I have had several mentors during my course of study at the university. First and foremost, Dr. Sirish L. Shah has been a wonderful supervisor. It was largely due to his guidance that I have been able to give my research project its final shape. His teaching style makes even the toughest of concepts easy to learn. I will always be indebted to him for allowing me to take my time in understanding things and giving me freedom to explore the research area while always being there to provide support and guidance. I got to learn a lot about teaching while working as a Teaching Assistant with him for CH E-358 (Process Data Analysis) course. I am a great admirer of his patient teaching, work ethics and his vast knowledge in Process Control and Chemical Engineering. If I have been able to absorb even a small percentage of these things from him, I would consider myself highly successful. Dr. Christopher McNabb from Matrikon Inc. is another person that I would like to thank. Discussions with him were very wonderfully thought provoking and they always kept me on my toes. I learnt a lot during my discussions with him and

he has been like a co-supervisor for me in this research work. His vast wealth of knowledge and cheerful demeanour made working with him an enriching experience. Without his help, this work would not be in its current form.

I would also like to thank Dr. Biao Huang, Dr. Vinay Prasad, Dr. David Shook, Dr. Ravindra Gudi, Dr. Sachin Patwardhan, Dr. M.A.A. Shoukat Choudhury for all their help with different parts of my academic life and research work.

I would like to thank, from the bottom of my heart, all my friends who stuck with me throughout all the ups and downs in my life while at the university. You guys made it possible for me to overcome the tough periods of my life. Special thanks go to Anuj Narang, Arjun Shenoy, Vivek Bhushan, Debanjan Chakrabarti, Xing Jin, Yashasvi Purwar, Johnny Razdan, Satadru Kashyap, Ganesh Gujarathi, Saneej B.C., Sandeep Reddy, Venkat Raghavan, Karteek Popuri, Fariborz Kiasi, Siddhartha Kumar, Dipen Deshpande, Aditya Tulsyan, Rajesh Hegde, Nitesh Goyal, Sunil Ravinder, Abhinav Mohan and Vishnu.

Special thanks are due to Anuj Narang and Arjun Shenoy for patiently helping me with all of my process control and coding related questions.

As I complete my academic journey and move to work life, I cannot help but think of all the love and support my family has provided me throughout my life. I am highly indebted to them for the trust they have had in me. My mom has been my pillar of faith and strength throughout my life. No words can express my gratitude for her love and trust.

Contents

List of Tables

List of Figures

List of Abbreviations

PID	Proportional Derivate Integral (Controller)
SP	Set Point (for a controller)
OP	Output (of a controller)
PV	Process Variable (in a control loop)
TE	Tennessee Eastman Process
KPI	Key Performance Indicator
SVM	Support Vector Machines
RFE	Recursive Feature Elimination
CV	Cross Validation
HAZOP	Hazard and Operability Study
P&ID	Process and Instrumentation Diagram
IDV	Disturbance Variable in Tennessee Eastman Challenge problem
XMV	Manipulated Variable in Tennessee Eastman Challenge problem

Chapter-1

Introduction

1.1 Plant-wide performance monitoring

Plant-wide performance monitoring refers to wide-spectra monitoring activities of different aspects of a process plant, for example: control performance, process performance and equipment or asset monitoring. This is an area that has generated much interest in process systems engineering research because it allows one to take stock of the performance of a plant and identify gaps in performance which could be improved upon to result in higher long term profitability. In this work, we have focused our attention upon the relationship between optimum process performance and the performance of controllers on the said process. We also develop a correlation between the two to identify gaps in controller performance that could result in an optimum process performance. In the following part of the thesis, we first define the meaning of process performance monitoring and controller performance monitoring in general before investigating their relationship in detail in the subsequent chapters of this thesis.

Process performance monitoring is concerned with the evaluation of performance of production assets such as reactors, furnaces and distillation columns. There are usually two aspects to measuring the process performance of a plant:

- Short term performance: how well is the process doing with respect to meeting its target of production, energy consumption or other key performance indicators
- Long term performance: setting the targets so that the plant is able to avoid unplanned outages. For example, if the plant is run at full

capacity at all times, it may increase short term profits but may also lead to equipment failure and could cause unplanned shutdowns resulting in big losses.

Controller performance monitoring, on the other hand, relates to calculation of statistical measures of performance of controllers and judging how they are performing with respect to user-specified benchmarks. Often times, poor performance as indicated by controller performance metrics suggests that tuning changes be made on the controller. Sometimes, this may happen because of valve or other equipment problems as well. Controller performance monitoring is typically used to monitor performance of PID controllers in industry but can also be extended to multivariable controllers.

With the recent advancements in computational technology, it has become much easier for industries to store process data and use them for a historical analysis and to optimize process performance based on past information.

In the scope of this study, we first focus on process and control performance monitoring and attempt to correlate the two using statistical techniques. The results obtained through developing this correlation also help us prioritize controller maintenance activities. In essence, we attempt to prioritize controller maintenance activities relative to their influence on the overall business/objective performance of the plant. We also discuss the merits of this method over methods that are conventionally used in industry.

1.2 Motivation

The technology of controller performance monitoring has matured to the point where it is routinely used by process engineers to identify controllers that are not performing well. However this advancement has also led to an information overload in the sense that for each control loop there are many

more performance metrics, often as many as 20 different indices! These metrics are definitely useful in the diagnosis of poor control performance; however they can often overwhelm the broad process performance picture.

Typical PID controllers have essentially one objective and that is to keep the controlled variable close to the desired set point while being subject to set point changes and external disturbances. Specific controllers are 'tuned' to provide a reasonable compromise between that objective and the constraints imposed by process dynamics and controller output activity. A growing number of performance and diagnostic metrics are available to help quantify various aspects of that single objective and to provide clues to performance problems. But that information alone is not sufficient to prioritize maintenance activities. Consider a hypothetical example of a temperature controller with a rapid but small oscillation that causes the output to swing by a couple percent. Is that control performance acceptable or unacceptable? Should it be given a high priority to repair or a low priority? The dilemma is depicted in figure 1.1.

Figure 1.1 - The challenge of prioritizing controllers (Pareek et al. [20])

Obviously, the answer is 'it depends'. If this controller is involved in heat recovery from a source of low grade steam then we probably consider this performance perfectly acceptable. However if this temperature controller is causing a critical reactant stream to fluctuate by 2% this could be considered poor or perhaps even dangerous! The point is that specific control performance issues cannot be prioritized without first putting them in the context of the overall process or business performance.

In a typical process plant, if one starts to look for poorly performing controllers, typically hundreds of controllers light up as ones that may require maintenance. In such a scenario, it becomes imperative that the maintenance of such loops be prioritized.

Therein lies the challenge of automatically prioritizing control performance problems. Controller performance metrics are fairly generic across processes

and even industries; however, the same is not true of process performance metrics. They tend to be very process and industry specific. Our challenge is to develop an algorithm which can relate control performance to process or business performance without requiring a huge amount of customization by the system integrator.

Previous studies have aimed at ranking of controllers through empirical methods such as those discussed in Trenchard et al. (2005)[1] and through their potential to affect other controllers in the unit/plant[2]. Trenchard et al. provide users with a checklist through which loop criticality can be determined. The checklist involves various parameters such as importance of loops with respect to safety of the operation. Choudhury et al.[3] discuss ways of prioritizing controllers through simple perturbation tests on the subject system. Their approach is focused upon quantifying the potential of a controller to affect other controllers in the unit as well but through process model based methods. In this work we analyze data based techniques for prioritizing controllers based on their process and business impact. The methods discussed in this thesis are remote analysis methods which require little or no interaction with the process. We will also discuss the merit of these methods in this thesis. Naturally, the methods are ideally carried out in the plant by process engineers who would have an intimate knowledge of the process and the plant. However, we did not have access to such information and as a result the reconciliation of controller performance with process performance had to be carried out remotely but on real industrial data.

1.3 Quantifying plant performance

A composite plant performance index is a variable that can be a weighted combination of several process variables (measured or calculated) or one high level variable of the overall plant/unit operation. The constituents of

plant performance index can be variables such as conversion rate, yield, fuel efficiency etc. There are various ways in which one can define the overall process/plant performance index. One simple idea is to find a region of acceptable value(s) of the objective function (comprising process variable(s) of interest). One can then reward the objective function when it lies in the region of acceptable values. Elsewhere, it can be penalized accordingly. For instance, suppose that we want to keep the process yield close to 90%; however values lying in the region of 85% to 95% are acceptable. For this situation, the objective function could take the form of a parabola. The maximum value of the parabola could be achieved at a yield of 90% at which point the objective function takes a value of 1. At points corresponding to 85% and 95% yields, the objective function takes a value of 0. For yields of less than 85% and more than 95%, the objective function takes negative values. It may seem counter-intuitive to penalize yields of more than 95% but high yield may not always be desirable. Operating process plants at high yield may result in severe degradation of process equipments as a result of pushing them to the extreme limits of their performance. This may cause unscheduled shut downs for maintenance. For the situation described above, an objective function is illustrated in figure 1.2:

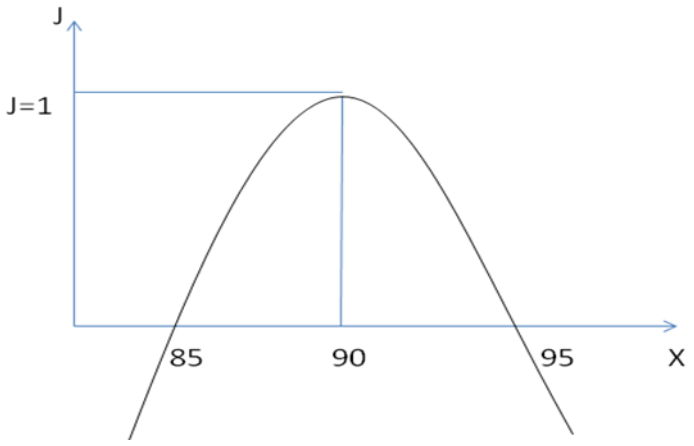


Figure 1.2 - Objective performance metric definition

For many reactors, conversion is the high level variable of primary importance to the operators of the plant. Conversion is defined as amount of feed converted per unit amount of feed into the unit. There is usually no optimal conversion level for a reactor; therefore defining the objective function as explained above may not be suitable. Usually the optimal conversion level is decided based on financial parameters such as cost of fuel, feedstock availability, purity of the feed and market conditions. Many process industries make use of an offline optimizer to decide upon an optimal conversion level for a finite time-horizon (in terms of days or weeks) for the plant. This is done keeping in mind the long term performance of the plant. Once the optimizer computes the conversion target (conversion SP), it is then expected that the reactor will operate close to this target using a supervisory control strategy. Therefore in this case, the objective function could be defined as the standard deviation of conversion from its set point value over a day: J=std (Conv SP – Conv PV). High standard deviation value means that the conversion value was off target by a great margin indicating that the unit performance on that day cannot be considered as a particularly good operational day. On days when conversion PV stayed close to the conversion SP value, standard deviation would be low, thus indicating that the day was a good operational day. However, it is not always a black or white case; there are days that don't belong to either of these categories. They are referred to as 'in-between' days in this study.

1.4 Definition of good and bad operational days

In the two case studies that are discussed in this work: case study of an industrial reactor and the Tennessee Eastman problem (TE problem), we define the objective function in terms of standard deviation of a Key Performance Indicator (KPI). For the industrial reactor, conversion was identified as the KPI. For the TE problem, ratio of the two main products (G &

H) in the product line was identified as the KPI. For both case studies, we define three classes of days: Good days, bad days and 'in-between' days. For the purpose of classifying days into these categories, we calculate the J function value, plot a time series graph and either visually inspect the thresholds (as in the case of the TE problem) or get inputs from plant engineers (for the industrial reactor case study). The J function value is calculated for each 24 hours period so as to judge the performance of the process in that one day.

For the industrial reactor case study, good and bad operational days are defined in terms of deviation of conversion value from its target over a 24 hour period. Therefore we calculate the error value (conversion SP – conversion PV) for 24 samples in a day and find the standard deviation of the error for each day. Next, we plot a time series of the standard deviation value of error for each day over the entire period of data available to us. Looking at the trends in this time series graph and after consultation with plant engineers, we propose a threshold or a limit to divide the data into good days (low standard deviation), bad days (high standard deviation) and 'in-between' days.

In the TE problem, it is known that high variability in ratio of products in the product line is an undesirable situation. Therefore we define std(flow rate G/flow rate H) in the product line as our objective function. As in the industrial case study, we calculate J value over a 24 hour period and plot a time series graph. After visual inspection we draw the threshold values in order to classify days into one of the three categories.

For the purpose of prioritization of controllers, we only consider the data in two classes (good days and bad days) and leave out the 'in-between' days. By doing so we ensure that we only find the most discriminating factors among

these two classes. The more discriminating a factor is, the higher priority it gets, thus helping us in ranking of the control loops.

The classification method that we use for the purpose of this study is the Support Vector Machines (2-class SVM and multi-class SVM) technique. Through means of feature selection strategies we find the most discriminating controllers among these two classes. A brief description of Support Vector Machines and the feature selection methods used in this study is provided in Chapter-2 of this thesis.

The plot in figure 1.3 shows the time-series of the objective function, J=std(error) for one of the reactors:

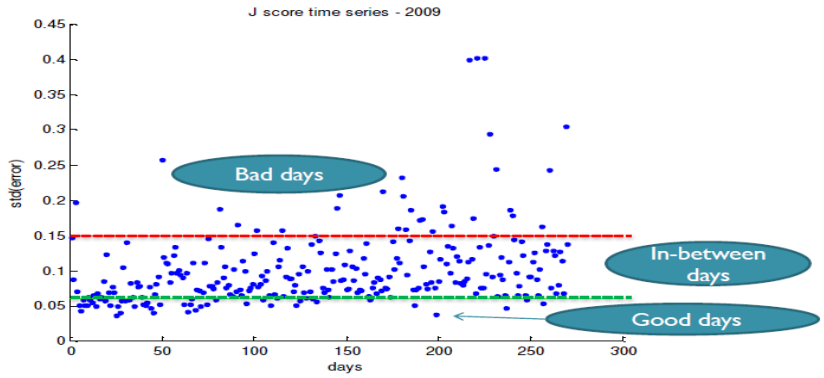


Figure 1.3 - J score time series for one of the industrial reactors for the year 2009

1.5 Outline of this thesis

This thesis is organised as follows:

- In chapter 2 we introduce Support Vector Machines (SVM) that is the main method of classification used in this work. Thereafter, we introduce feature selection methods that help us identify important contributors to any given classification model.
- In chapter 3 we discuss the first industrial reactor case study. This is carried out on data obtained from three identical industrial reactors. We apply the controller prioritization algorithm and rank order the controllers according to their impact on a business objective. Also, we compare the results from the three industrial reactors and discuss the similarities and differences in the results.
- Chapter 4 looks at the second case study, the well known Tennessee Eastman Challenge problem (TE problem). We simulate data from a TE simulation model and apply controller prioritization algorithm in order to identify important controllers in the TE process.
- Chapter 5 discusses concluding remarks and future work that could be done in further automating and improving methods of controller prioritization.

Chapter-2

Support Vector Machines and Feature Selection Methods

'Support Vector Machines' (or SVM) is a relatively new supervised machine learning algorithm proposed by Vapnik et al. The foundation for SVM was laid by Vapnik in the year 1982[4]. It was formally proposed by Boser, Guyon and Vapnik in their paper in 1992[5]. Since that time, SVMs have been successfully used as methods of classification and regression. It has been particularly successful in applications in the fields of handwritten digit recognition and pattern recognition.

Given data points from two classes, SVM finds the optimal separating hyperplane boundary between points of the two given classes. The optimal separating hyerplane boundary is the one that has the maximum margin of separation between the two classes. Figure 2.1 explains the concept of optimal separating hyperplane.

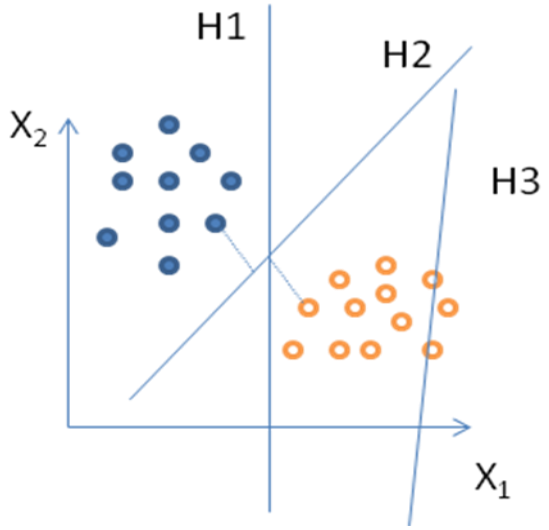


Figure 2.1 - SVM separating hyperplanes

In figure 2.1, there are points belonging to two classes. Points from class 1 are represented by solid dots and those from class 2 are represented by blank dots. As can be seen, hyperplanes H1 and H2 both act as separation boundaries between the points from two classes. Hyperplane H3 does not do a good job of separating these points. H2 is really the optimal separating hyperplane because it is the maximum margin classifier. Margin is defined as the width that the boundary can be extended by before it hits a data point. The points that are the first to hit the separating boundary when its width is extended are called support vectors. Maximal margin hyperplanes are considered optimal separation boundaries because they lead to less generalization errors[5]. They are more robust and less prone to the presence of noise in data. In the following sections, we will look at how SVM finds the maximal margin hyperplanes for two cases, i) linear SVM and ii) nonlinear SVM. We will only discuss the case of 2-class SVM methods where the goal is to separate points belonging to two different classes. In section 2.4 we will show how this idea can be extended to multi-class SVM.

2.1 Linear Support Vector Machines

Linear SVM, as the name suggests, aims at constructing maximal margin linear separation boundaries (hyperplanes) to separate points from two different classes. Consider that the set of training data available to us is in the form

D=$\{x_i,y_i\}$ where $i=1,.....l$

$y_i=\{-1,+1\}$ represents the class of the data point. The nomenclature is arbitrary and any of the two classes can be labeled +1 or -1. These labels are required for SVM model building exercise during training.

Let us assume that the separating hyperplane can be represented as:

F(x) = wx+b=0

Since, we assume that the hyperplane is linear, this is a fair assumption. When we extend the margin of the separating hyperplane until it touches points on both sides (belonging to +1 and -1 classes respectively), we get two more planes: plus-plane and minus-plane. Points lying on the other side of plus-plane (away from the classifier boundary) will be predicted as points belonging to class +1. Similarly, points lying on the other side of minus-plane (away from the classifier boundary) will be predicted as belonging to class -1. If we assume that the points are completely linearly separable, we have to maximise the distance between the plus planes and minus planes subject to the condition that all points are correctly classified.

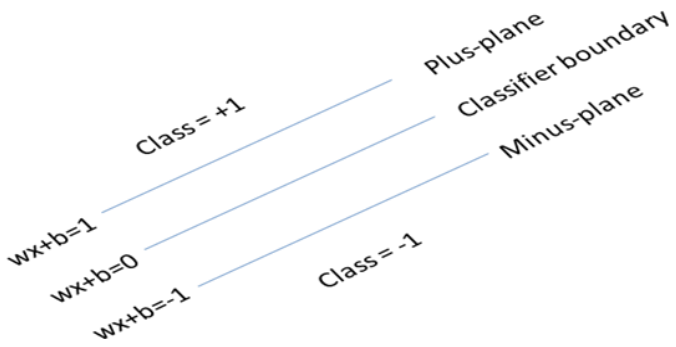


Figure 2.2 – SVM classification boundary and labeling of classes

It is easy to understand that the distance between these two planes is $d = \frac{2}{\|w\|}$. Hence the goal is to minimize //*w*//. Mathematically,

Minimise J=$\frac{1}{2}$//*w*//^{2} under the following constraint:

$$y_i(w.x_i + b) - 1 \geq 0 \quad \forall\ i \tag{2.1}$$

The factor of $\frac{1}{2}$ is included for mathematical convenience as can be seen when differentiating equation (2.2).

In order to make this problem a little easier to handle and also to generalize the problem formulation for linear and non linear cases, this optimization problem is re-written using Lagrange multipliers.

Let us introduce a set of positive Lagrange multipliers α_i, *i=1,…,l,* one for each inequality in (2.1). For inequalities of the form $c_i \geq 0$, inequalities are multiplied with a positive Lagrange multiplier and subtracted from the objective function to form the Lagrangian[6]. Therefore, the Lagrangian becomes:

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i . w + b) + \sum_{i=1}^{l} \alpha_i \tag{2.2}$$

The SVM algorithm solves this optimization problem subject to the constraints that the derivative of *Lp* vanishes with respect to *w* and *b* and $\alpha_i \geq 0$. Therefore, we get:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{2.3}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.4}$$

Substituting (2.3) and (2.4) in (2.2), we get the dual form of the problem:

$$L_D = \sum_{i}^{l} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i . x_j \tag{2.5}$$

Minimizing *Lp* is the same as maximizing L_D with respect to α_i, subject to the conditions

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad \text{and } \alpha_i \geq 0 \tag{2.6}$$

The SVM algorithm solves the above quadratic optimization problem and finds the maximal margin hyperplane for a given training data. This kind of SVM classifier is also known as hard margin classifier because we assume that none of the training data points is misclassified. For this optimization to work as intended, it is essential that the points be indeed linearly separable. Section 2.2 looks at cases when the points are not completely separable using linear SVM.

2.2 Linear SVM for non-separable classes

Suppose the classification problem is so posed that a linear separation boundary can do a fairly good job of separating points from two different classes but it cannot separate them completely. These cases are fairly common in real life. In fact it is uncommon to find cases that are completely linearly separable. It is easy to understand that solving the quadratic optimization problem that we formulated in section 2.1 is not going to be helpful in this case. Therefore we construct a soft-margin classifier by adding a cost function term penalizing misclassification in training data set to the term that maximizes the margin of the separation hyperplane.

In figure 2.3, one can observe that solid dots (class 1) and hollow dots (class -1) are not completely separable, therefore a different objective function has to be formulated.

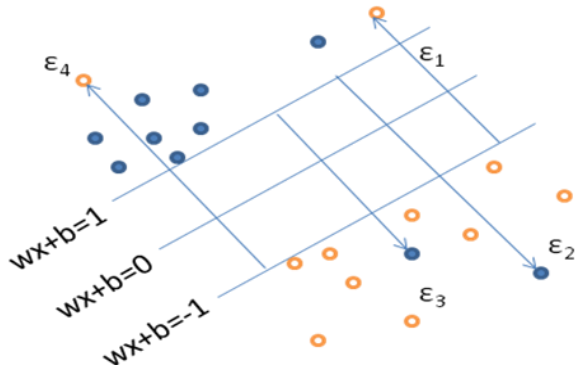


Figure 2.3 – Linear SVM for non-separable classes

Therefore the objective function can be formulated as follows:

$$J = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\varepsilon_i \tag{2.7}$$

J has to be minimized under the following constraints:

$$y_i(wx_i + b) \geq 1 - \varepsilon_i \tag{2.8}$$

Here, ε_i denotes the distance of incorrectly classified points from the correct plane and C is a cost function that penalizes misclassification error. C is a parameter that can be tuned to give best accuracy over k-fold cross validation. In principle, high value of C implies a higher penalty for misclassification during training. Excessively high C may result in over-fitting the data where one may get low misclassification rates on training data set but the classification model may not have a good generalization capability. Excessively low C may result in a large number of training points getting misclassified and may even cause high misclassification rates in testing data set as a result of poor model learnt.

Using Lagrangian formulation, we can re-write the objective function as follows:

Maximize $$L_D = \sum_i \alpha - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j x_i . x_j \quad (2.9)$$

Subject to

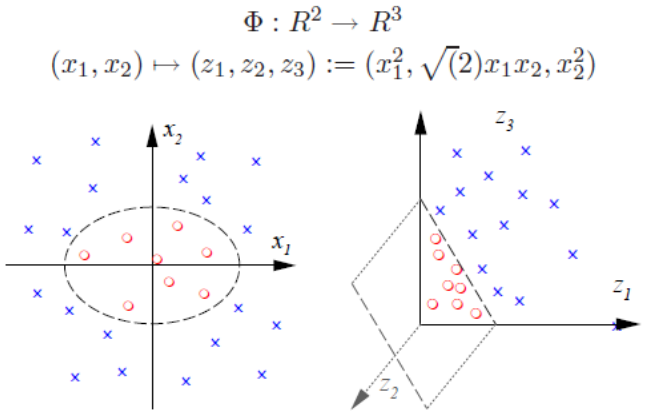
$$0 \leq \alpha_i \leq C \quad (2.10)$$

$$\sum_i \alpha_i y_i = 0 \quad (2.11)$$

Soft margin classifiers give reasonable results only if the training data set is nearly linearly separable. In cases, when constructing a linear classifier will result in large number of training errors, it is advisable to build non-linear SVM classification models. This is usually done through transforming data points from the original input space to a higher dimensional feature space. Section 2.3 looks at developing non-linear SVM classifiers.

2.3 Non-linear SVM classifiers

Application of SVM is not limited only to cases when a linear separating boundary can be drawn between points belonging to different classes[6]. They can also be used to draw non-linear separating boundaries using the 'kernel trick'. Kernel transformation maps data points from input space (original dimension) on to a higher dimensional feature space. This transformation is so done that points that are linearly non-separable in the input space become linearly separable in the feature space. SVM then constructs a separating hyperplane in the feature space which when transformed back to the original input space translates to a non-linear separating boundary. A simple illustration of the kernel trick is presented in figure 2.4:



(Courtesy of: Jason Weston, NEC Labs America)

Figure 2.4 – Kernel transformation of linearly non-separable points

In figure 2.4, *(x_1, x_2)* is the original space. Data points (circles and crosses) seem to be linearly non-separable in the input space but when transformed to a 3 dimensional space by introducing the following transformation:

$z_1 = x_1^2$, $z_2 = \sqrt{2}x_1x_2$ and $z_3 = x_2^2$ (2.12)

These points can be easily separated by a linear hyperplane in the feature space.

It can be observed from equations (2.5) & (2.9) that the optimization problem only depends on the inner product $< x_i . x_j >$. Now let us assume that we use a kernel mapping function φ to transform the data on to a higher dimensional space. In this space, the optimization problem would only depend on $\varphi(x_i).\varphi(x_j)$ If there exists a kernel function K such that

$K(x_i, x_j) = \varphi(x_i).\varphi(x_j)$, then we would only need *K* in the training algorithm and would never need to know φ explicitly. Commonly used kernel functions are as follows:

- Linear kernel: $x_i . x_j$
- Polynomial kernel: $(\gamma x_i . x_j + constant)^d$
- Radial Basis Function kernel: $e^{-\gamma \left\| x_i - x_j \right\|^2}$

Kernel functions should be positive definite and should obey Mercer's theorem in order to be legitimate to represent a dot product space[6].

After the data points are transformed to a higher dimensional space, we can formulate an optimization problem similar to the one formulated in section 2.2 to find the maximal margin hyperplane in feature space.

If we use the radial basis kernel function for instance, we see that there are two parameters that need tuning: C and γ . When using polynomial kernel functions, one would have to tune 3 parameters. The way tuning is done in this work is through a grid search method where different combination of values of C and γ are tried and the pair that gives the best accuracy over 5 fold cross validation is chosen. 5 fold cross validation refers to segmenting the data randomly into 5 folds. For a given pair of (C, γ), we train the SVM model over 4 folds and test it on the fifth. We repeat this exercise 5 times such that a different fold is left out of model building each time. We average the misclassification rate for each pair of (C, γ) over these 5 runs and select the pair that gives the lowest misclassification rate.

Readers are referred to C.J. Burges[6] and Hsu et al.[7] for a more detailed tutorial on SVM theory and applications respectively.

2.4 Multi-class SVM

Until now, we have only looked at cases where classification is done between two classes. SVMs were originally proposed as a classification method for

two class problems. Several modifications, however, allow for them to be used for multi-class problems.[10, 11.]

In this thesis, we have used one of the several possible modifications to develop 3-class SVM models. This technique is called one-versus-one. Here, we develop three 2-class SVM models taking two classes at a time and then use a voting technique to decide the true class of the data point. Section 3.3 explains in detail the 3-class SVM models employed in this work.

2.5 Feature selection through Recursive Feature Elimination

Feature selection refers to selecting a subset of variables (features) that are most relevant to the classification models. Recursive Feature Elimination is one of the many ways to do that. The various ways to do feature selection for a classification strategy are broadly categorized as: Filter method, Wrapper method and Embedded methods[8]. Following is a brief overview of these strategies. Readers are referred to Guyon[8] for a more detailed account of feature selection strategies.

Filter methods usually involve calculation of a statistical metric corresponding to all features and rank order them according to the value of the metric. Such methods are independent of the model building exercise. Fisher score calculation is one such way[9]. Fisher scores calculate a discriminant value corresponding to each feature in the data set. The higher the discriminant value, the more statistically important a feature is perceived to be. Filter methods are useful in cases when there are a large number of features to work with and the cost of including all of them is prohibitive. One can use filter methods to exclude some features that are deemed non-important from the model building exercise. Depending only on filter methods can be disadvantageous because they do not take into account how

different features interact with the classification model. In this work, we use Fisher scores only as a tie-breaking criterion in Recursive Feature Elimination algorithm that is used to rank controllers in the reactor unit. A more detailed account of the feature selection algorithm as used in this thesis is provided in section 3.4.

Wrapper methods are a class of feature selection methods that are usually dependent on the classification model that is learned from the data. There are various ways of employing wrapper methods in feature selection[6]. Wrapper methods test various possible subsets of features and their effect on model accuracy. Wrapper methods are generally computationally very intensive but using greedy search strategies such as backward elimination and forward selection, the computational load can be reduced.

Another way to reduce the computational effort that is usually required for wrapper methods but still give an accurate feature selection result is to use embedded methods of feature selection. Embedded methods use techniques of feature selection with the model building exercise. Guyon et al. suggest different ways to do feature selection in an embedded manner.

In this study, we have used Recursive Feature Elimination which belongs to the class of wrapper methods of feature selection. RFE can be used to find the optimal feature set as well as to rank order the features in a classification problem. RFE as used in this case study works as a greedy backward selection algorithm eliminating in each step the variables that contribute the least to the classification accuracy. The step-wise elimination can be stopped when the desired accuracy is reached and can also be continued until all features have been ranked in the order of their removal. A stepwise approach to performing RFE as was done in this case study is explained in section 3.4 of chapter 3.

An important thing to remember while applying feature selection is that one should ensure that the classification models built on the data set give a reasonably good 5 fold cross validation accuracy. If they don't, one should try to fit better models, try another technique, adjust model parameters etc. Only when it is ensured that the generalizable performance of the model is good should one proceed to find the important contributors to that classification model. Throughout this work, we achieved SVM classification accuracies in excess of 90% therefore we could apply feature selection and develop the controller prioritization algorithm.

In this work, we have used the SPIDER toolbox[18] in MATLAB for developing classification models using SVM.

Chapter-3

Case Study: Unit-wide performance assessment of an industrial reactor using SVM for Controller Prioritization

As the first case study for this paper, we analyzed data from three industrial reactors. The main aim of this case study is to correlate overall unit process performance with controller performance and identify those control performance problems most correlated to reactor performance. These important controllers are the ones that are the most discriminating factors between good days and bad days. As we will see in the distribution histograms presented in this section, good performance of the higher ranked important controllers meant it was a good day for the plant and vice versa. When a subset of few important controllers has been identified, maintenance priority can be given to these controllers whenever process performance appears to degrade. In any case these are the controllers that should be examined first in case unit performance appears to be degrading. Statistical analysis tells us that these controllers have the biggest impact on process performance and hence have the higher maintenance priority. We will also see through the application three-class SVM classification that these important controllers if maintained properly can significantly improve the performance of the reactor unit.

3.1 Description of the Industrial Reactor Unit

The reactor studied for this case study is a fairly common process in hydrocarbon processing industries. The feed gets treated inside the reactor and the outlet stream is a mixture of products, one important desirable

product plus by-products. Some part of the feed remains unconverted and comes out of the outlet line along with other products. The conversion rate tells us how much of feed has been converted into other products. The desirable component on the outlet line is a specific product, 'X'. The percentage of this product in the outlet is the variable yield. As identified in Section 1.2, the objective function of the supervisory control strategy is standard deviation of conversion. This is also the measure of goodness of a day for our study.

We performed the analysis on three separate reactors (R1 – R3). For each of the three reactors, we gathered data on conversion SP, conversion PV, reactor on/off days, supervisory control on/off days and several performance metrics for the controllers in the reactor unit. The reactor unit has a total of 17 controllers. Ten of these controllers regulate the feed flow into the reactor (W1, F1-F4, W5 and F5-F8). There are three pressure control loops, two of which control the pressure of fuel gases (P1 and P2) and one that controls the inlet air (P9). The fuel gas pressure loops (slave controllers) are cascaded with two temperature control loops (T1 and T2: master controllers). The outlet products are cooled by a stream of boiler feed water. There is a flow control loop (F9) that regulates the flow of boiler feed water. One controller that is very important with regards to safe operation of the reactor is the one that regulates the amount of Oxygen inside the reactor. The inlet air pressure loop (P9) is cascaded with the excess Oxygen control loop (A9).

The performance metrics data that was collected corresponded to three indices I_1, I_2 and I_3. I_1 corresponds to the settling time of the controller compared to a benchmark controller. I_2 gives the standard deviation of the control error and I_3 is a metric that quantifies the amount of oscillation present in the control loop.

To keep the analysis unbiased, we removed certain days of data from our analysis. These were days when conversion SP changes were made on days when supervisory control was fully or intermittently operational. Also removed were days when the supervisory control was completely off during the day. This was done because it is observed that the standard deviation of conversion error tends to be higher on days when Conversion SP change was made and on days when the supervisory control was off. The reason for this high standard error on these days may not just be bad control. Since, we assume that there is a causal relationship in control performance with process performance, we exclude these days from the SVM model building. Therefore now, calculation of standard error is equivalent to calculating standard deviation of conversion PV.

3.2 SVM classification models and feature selection

In this case study, we have assumed that the following factors: performance of controllers, incipient faults in the process and process operating conditions have a causal relationship with process performance. It is easy to understand why controller performance has a causal relationship with process performance. If a controller oscillates significantly or has a high standard error, it will result in the value of a particular process variable being very different from its desired set point and thus overall process performance will be poor. There have been studies that have reported the effect of incipient faults in the process as a factor influencing process performance as well[12]. Also, it is often seen that industrial processes behave differently in different operating conditions, i.e. the process model used as a basis for supervisory control may work very well for certain operating conditions but may not work as well for other operating conditions. For this reason, we select these three factors: controller performance (3 indices for each of the 17 controllers), incipient fault data (which relates to scaling of

reactor tubes) and mean of feed flow set point and conversion SP for the day. This means we have a total of 54 variables as our predictor variables for this case study.

The strategy that we have followed in this work can be described as follows. We decide upon the good day and bad day criteria based on a process variable value (standard deviation of conversion). The important point to note here is that these boundaries are not fixed and can be moved around to some extent. They can be different for different reactors. The only idea behind drawing these separation boundaries is that when we draw separation boundaries using SVM on predictor variables we want a good level of separation so that the boundaries are more robust and wide. Once the good days and bad days are decided upon based on the standard deviation of conversion, we take the 54 predictor variables for good days and bad days and develop a 2-class SVM model for classifying good day data from bad day data. 2-class SVM models between good days and bad days built on the predictor variables give a mean accuracy over 5 fold cross validation of around 90-95%. 5 fold cross validation, as used in this study, can be explained as follows:

The entire data set is partitioned into 5 folds of almost equal sizes. SVM model is built on 4 of these folds and tested on the 5^{th} partitioned segment of data. This exercise is repeated 5 times leaving a different fold out of the model building process each time. Mean accuracy is calculated by taking the mean of 5 accuracy values obtained in each of the five iterations. 5 fold cross validation ensures that the model does not over-fit the data and remains robust. This robustness can be judged from the fact that a mean accuracy of 90% indicates that the SVM models were able to classify correctly 90% of the points that were not used in the model building exercise.

After this, we perform feature selection as described in step 2 to prioritize controllers for the reactor unit. The working algorithm is explained through the flowcharts presented in figures 3.1 and 3.2:

Step1: Overall data processing algorithm:

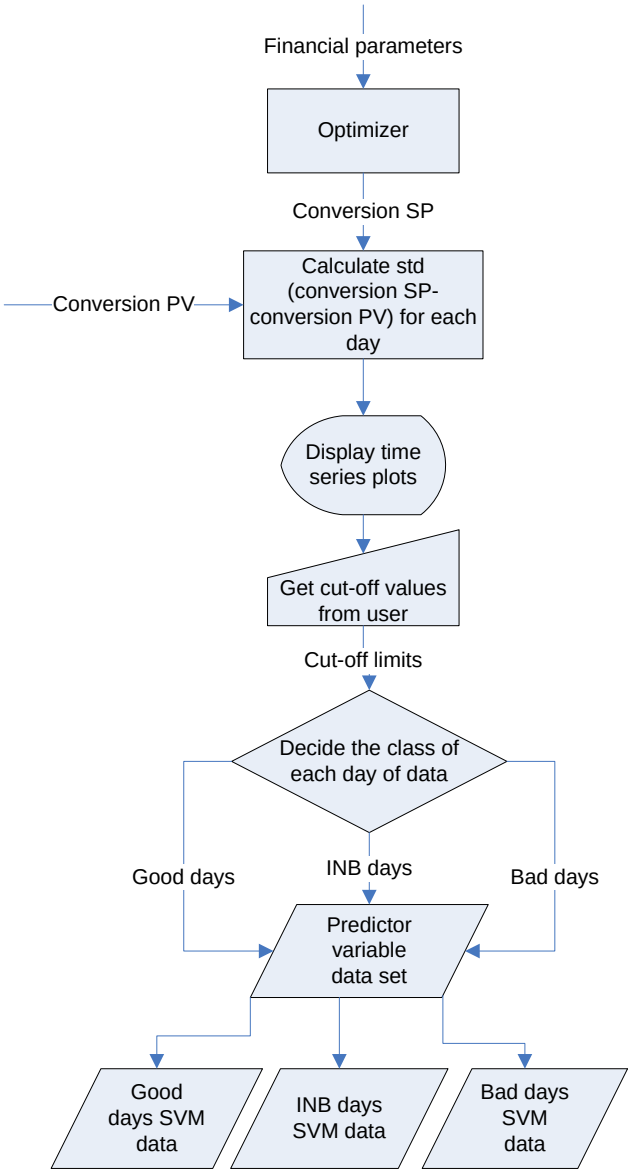


Figure 3.1 – Data processing algorithm

Step 2: Recursive Feature Elimination algorithm for identifying important control loops, this is the Controller prioritization algorithm

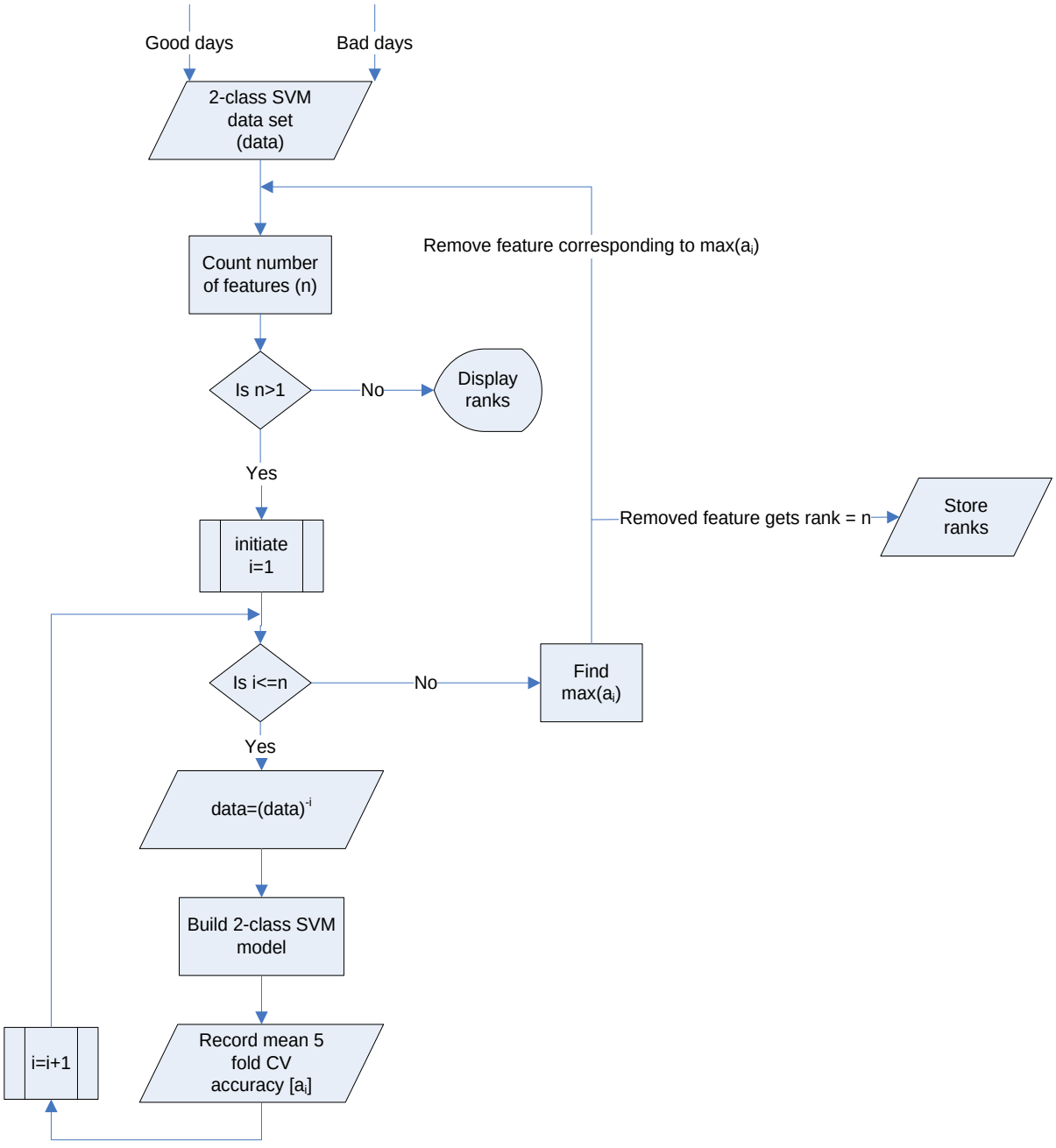


Figure 3.2 – Recursive Feature Elimination algorithm

Recursive feature Elimination for SVM models can be done in many ways depending upon the problem at hand. In this study, we used a methodology described in Maldonado et al.[13] with slight modifications suiting the needs of this study. A brief description of the Feature selection methodology as used in this case study is as follows:

1) Count the number of variables (features) remaining in the problem (n).

2) Remove one variable at a time (x_i); develop a two class SVM model (m_i) of the reduced dimension good day and bad day data sets; record mean accuracy over 5-fold cross validation (a_i).

3) Find the maximum a_i value. Note that, this maximum accuracy was achieved by the removal of variable x_i. It means that variable x_i is probably not an important contributor to the good day-bad day classification.

4) If there is not a unique maximum value and there are two or more maximum accuracy values (a_i, a_j etc. for instance). Compare the Fisher scores of these variables. Select the variable with a higher Fisher score.

5) Remove the variable x_i identified as the least important contributor to the SVM model.

6) Repeat steps 1-5 until all variables are ranked (variables eliminated earlier are ranked lower)

One must realize that this sort of feature selection procedure is best applied by partitioning the data set differently and repeating this calculation multiple times. For this case study, we partitioned the data set into 5 folds differently for 32 times. We applied the feature selection algorithm for each of the 32 times and the final rankings that we arrived at were the mean of the rankings over 32 runs. It is observed that mean rankings do not change even if the feature selection algorithm is applied for more than 32 times. The number 32 was chosen because the seed to the MATLAB random number generator

using 'twister' algorithm by Matsumoto et al.[14] can be varied from 0 to 2^32 – 1. For computational ease, in this study, the seed to the random number generator were given as 2^s where s varies from 0:31.

In this case study, there are 17 controllers with 3 indices each in addition to 3 process variables as mentioned in section 3.2. To prioritize the controller maintenance effort, the controllers need to be ranked in order of their relative importance to the good day-bad day classification. Therefore when removing a variable in the feature selection algorithm, we remove all 3 indices corresponding to a controller. In effect, we have 20 variables to be rank ordered. Therefore steps 1-5 of feature selection need to be repeated 19 times. The entire exercise is repeated 32 times. In each of the 32 runs, the features are ranked according to the order of their removal. Variables removed earlier get a lower rank as compared to variables removed after them. These rankings are then summed up over 32 runs in order to identify the top few contributors to the classification between good days and bad days.

The 5 most important controllers as identified by feature selection for this case study are as follows:

Reactor R1	Reactor R2	Reactor R3
T2	T1	T1
T1	T2	T2
P1	W1	P9
W5	A9	A9
F1	P9	F7

Table 3.1 – Results from feature selection

These results show a great deal of similarity between the three reactors. For all the three reactors studied in this work, the temperature loops are identified as two of the top 5 most important controllers. This signifies the importance of proper maintenance of temperature control loops for these

reactors. In the feature selection results for R1, one can see that loop F1 is identified as being very important. When the plant engineers were asked about these results, they identified F1 as being one controller that has been difficult for them to tune. Feature selection results on other reactors were also corroborated by the plant engineers. In short, the essence of this study is to use data based methods to identify important controller based on a performance or business based objective function so as to help plant engineers prioritize the maintenance effort for these controllers. In the next section, we show results from three-class SVM classification.

3.3 Three-class SVM

Three-class SVM models are built on the complete data sets, i.e. inclusive of good days, in-between days and bad days. It should be noted that we had excluded the in-between days from the model building process for feature selection. The reason for their exclusion was to ensure high separation between the two classes.

Three-class SVM models as used in this study are really a combination of three, 2-class SVM models built on

1) Good days and bad days (model1),

2) Good days and in-between days (model2), and

3) Bad days and in-between days (model3)

Predicted class of a data point is decided by a voting scheme. For instance, on passing a novel data point through the three models:

Model1 predicts: it is a good day

Model 2 predicts: it is a good day

Model 3 predicts: it is an in-between day

In such a scenario, we have two votes for it being a point belonging to the good day class. Hence, we predict it to belong to the 'good day' class. If the day is actually a good day, we consider the point to have been correctly classified. Otherwise, we take it as a misclassified point.

There is a possibility that all three models give a different prediction for the class of a data point. In such a case, we can't decide the predicted class of the data point. Such cases are few and far-between. We consider these points to be misclassified regardless of their actual class.

This kind of 3-class SVM model is known as one-versus-one 3-class SVM models. They are named so because the SVM models that are developed in this scheme are all 2-class models developed between two classes at a time. A detailed description of three-class SVM can be found in Crammer[9] and Ulrich Kreßel[10]

For this case study, five-fold cross validation of the 3-class SVM models give a mean accuracy of almost 90% for all three reactors studied in this paper. In this case study, three-class SVM models are built for two purposes: a) to underline the significance of feature selection results, b) to identify regions of good performance of the reactors. These results are best understood by the numerical experiments presented in sections 3.3.1 and 3.3.2.

3.3.1 Numerical Experiment 1: Importance of the top 5 controllers

In numerical experiment 1, we replace the controller indices values of the top 5 controllers (i.e. a total of 15 variables) of the bad days with that of the mean value of the good day data for the corresponding 5 controllers and parse them through the 3-class SVM model. We repeat this experiment now with the controller performance indices of top 10 controllers changed. INB

denotes the percentage of bad days that changed to in-between days and Good denotes the percentage of bad days that changed to good days. The results are tabulated in table 3.2:

Number of controllers	R1	R2	R3
5	INB= 88% Good=12%	INB= 94% Good=6%	INB= 95% Good=5%
10	INB= 37% Good=63%	INB= 59% Good=41%	INB= 55% Good=45%

Table 3.2 – Results to illustrate the importance of feature selection

We observe that if the identified important controllers perform well, bad days can almost disappear. For the reactor to give a good performance however, significantly larger number of controllers should perform optimally.

3.3.2 Region of optimal performance for the reactors

As a starting point, take mp_1 to denote the mean of good day data, mp_2 to denote the mean of bad day data and mp_3 to represent the mean of in-between day data. Each of these three means is a 1x54 vector. Find 100 points equally spaced between two means taken at a time. In effect, we simulate a total of 300 data points. These points are then parsed through the 3-class SVM model. Transition of data points, taken 100 at a time, from one class to another is plotted in figures 3.3, 3.4 and 3.5:

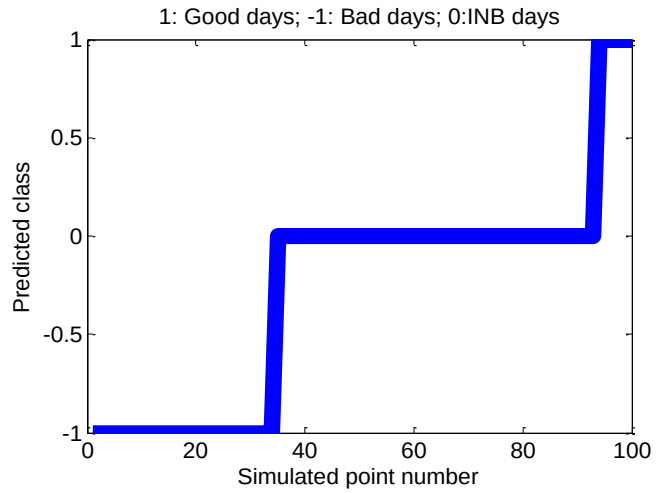


Figure 3.3 – Class transition from bad days to good days

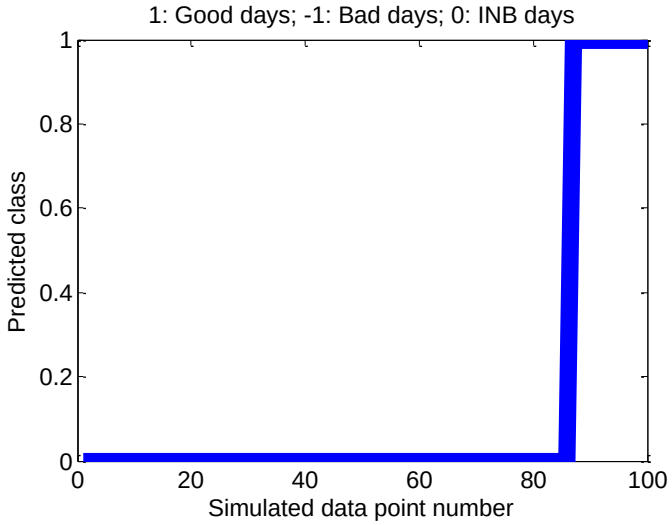


Figure 3.4 – Class transition from INB days to good days

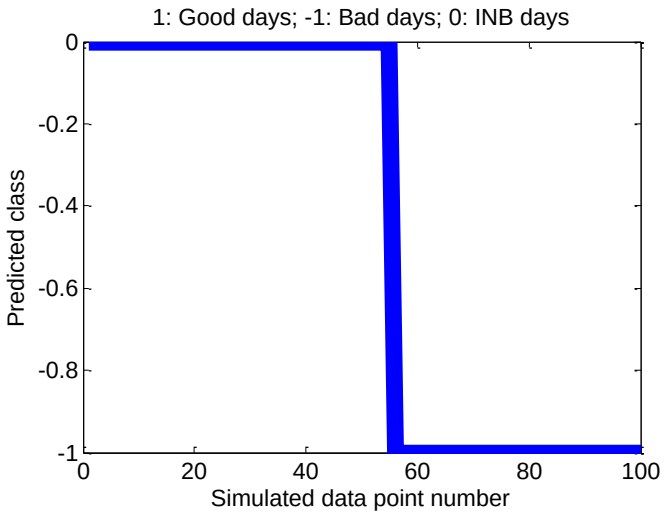


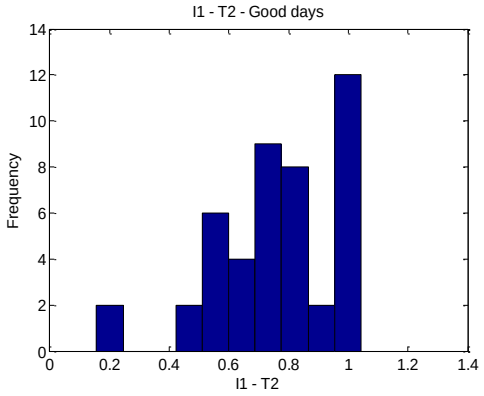
Figure 3.5 – Class transition from INB days to bad days

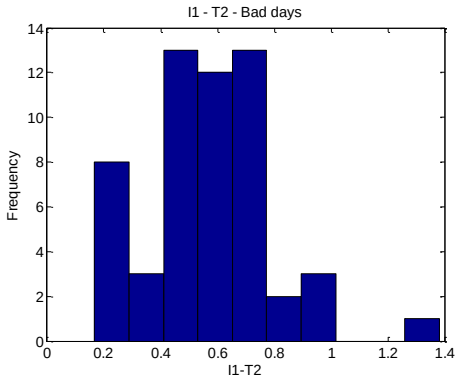
The key observation in these class transition graphs is that there is a region of in-between days between the regions of good days and bad days. When we move from optimal controller index values (i.e. mean of good day data set) to sub-optimal controller index values (i.e. mean of bad day data set), we encounter a region of INB days. Similarly from other transition graphs we can conclude that there is no good day region between bad days and INB days; also there is no region of bad days between regions of good days and INB days. The implication here is that we know where the transition from one class to another class occurs. Therefore, we can figure out the optimal controller performance index value region wherein if we maintain our controllers, we will have more optimal plant performance. If it is possible to translate the controller index values back to tuning parameters for the controllers, we can identify optimal tuning parameters for the controllers.

3.4 Distribution Histograms

So far, we have identified the important controllers for each of the three reactors. In order to somewhat validate our results, we present here the distribution histograms of the controller performance indices for the identified important controllers for each unit. The important observation here is the clear difference in performance of the identified important controllers over good days as compared with bad days. It should be noted that these distributions are not normal or follow a clear pattern because the number of sample days available to us are very few. Presented in figures 3.6 – 3.12 are distribution histograms from a few controllers for each reactor

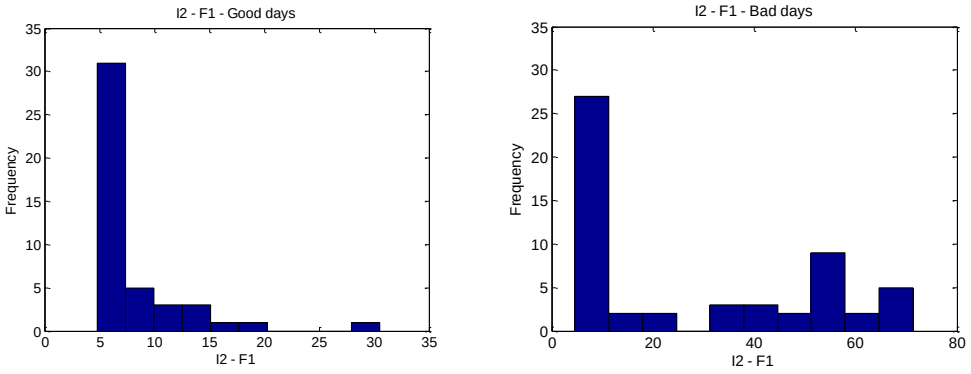
R1:





Mean: 0.76 Mean: 0.57

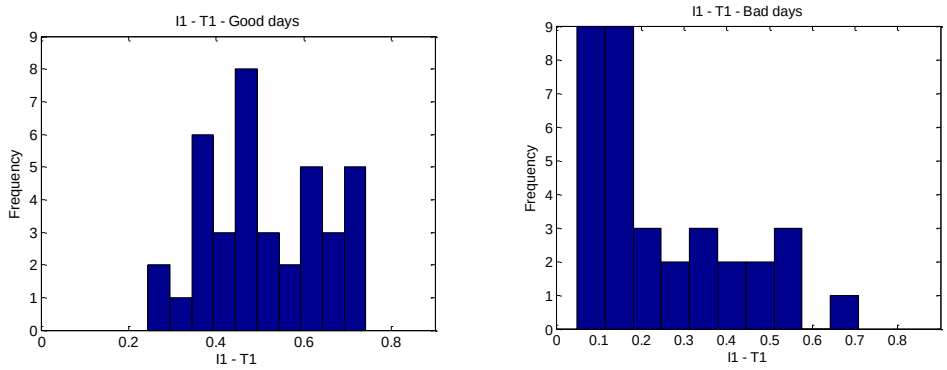
Figure 3.6 – Distribution histograms of index I1 for controller T2 over good days and bad days



Mean: 8.5 Mean: 27.9

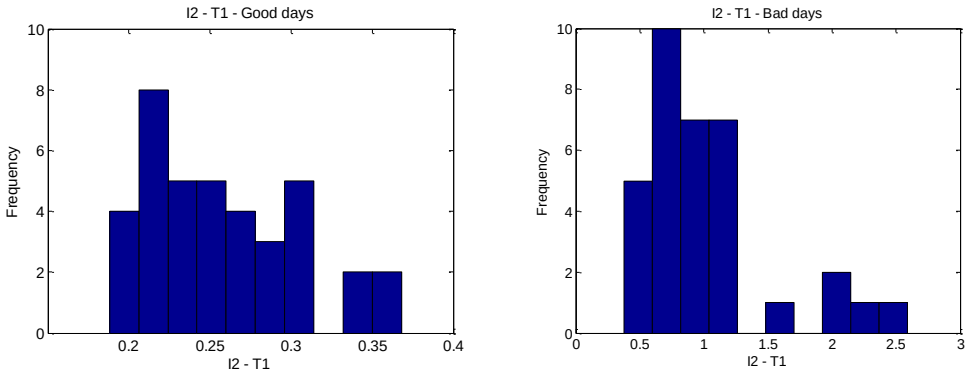
Figure 3.7 – Distribution histograms of index I2 for controller F1 over good days and bad days

R2:



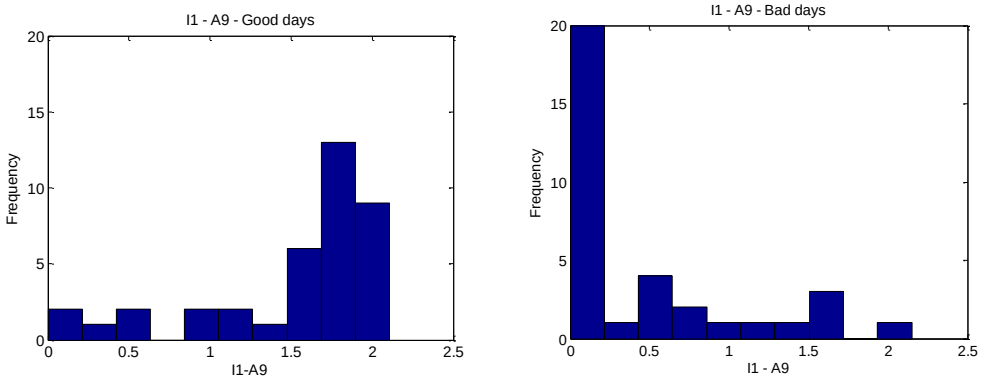
Mean: 0.51 Mean: 0.25

Figure 3.8 – Distribution histograms of index I1 for controller T1 over good and bad days



Mean: 0.25 Mean: 1.01

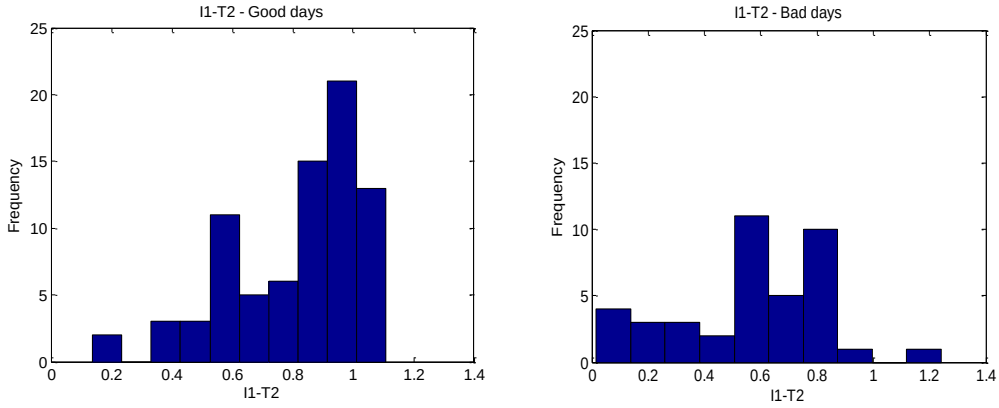
Figure 3.9 – Distribution histograms of index I2 for controller T1 over good days and bad days



Mean: 1.50 Mean: 0.46

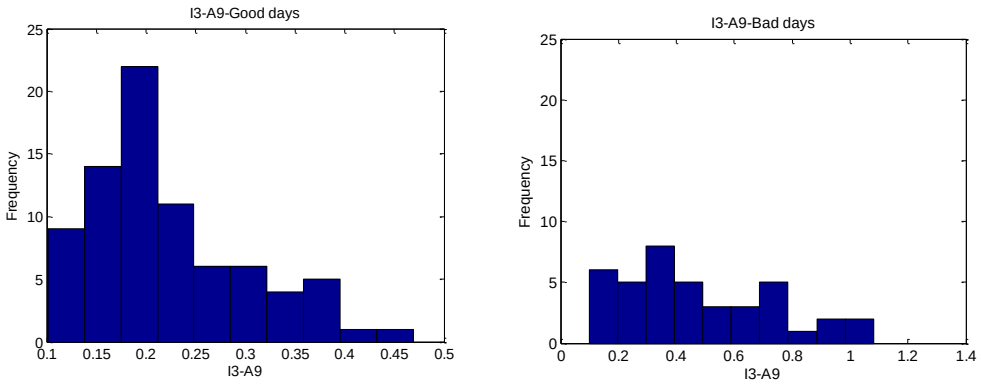
Figure 3.10 – Distribution histograms of index I1 for controller A9 over good days and bad days

R3:



Mean: 0.81 Mean: 0.56

Figure 3.11 – Distribution histograms of index I1 for controller T2 over good days and bad days



Mean: 0.22 Mean: 0.47

Figure 3.12 – Distribution histograms of index I3 for controller A9 over good days and bad days

We observe that in all of these histograms: index I_1 (corresponding to settling time of the controller) tends to have a higher value on good days as compared with bad days (high value indicating a lower settling time) and indices I_2 (corresponding to standard deviation of control error) and I_3 (quantifying oscillation the control loop) tend to be lower on good days compared with bad days. This observation also verifies our results because it shows that when important controllers had a good day, the reactor unit had a good day as well. Therefore, these important controllers impact the performance of the reactor more than others.

3.5 Control Loop Digraph and Reachability Matrix

Control Loop Digraph is a technique of capturing information flow between control loops in a plant/unit operation. They have been used for a variety of purposes. Some of them include root cause analysis of plant oscillations[15] and HAZOP[2] analysis. The concept of Adjacency Matrix and Reachability

Matrix are associated with control loop digraph and can help us develop a static ranking of controllers in order of their potential to affect other controllers in a unit operation.

A control loop digraph has control loops depicted as nodes and directed arrows connecting these nodes. The directed arrows and their direction are decided by a logic explained as follows:

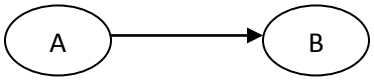


Fig. 3.13 – Construction of a Control Loop Digraph

In figure 3.13, we can see a directed arrow from control loop A towards control loop B. This indicates that OP of controller A can directly affect PV of control loop B. In many cases, there are two way arrows between two control loops; this would mean OP of each controller directly affects the PV of the other controller. It should be kept in mind that only direct interactions between control loops are depicted by an arrow connection. 'Indirect' connections are not. A detailed explanation is provided in the example in section 3.5.1.

3.5.1 Example illustrating construction of a Control Loop Digraph

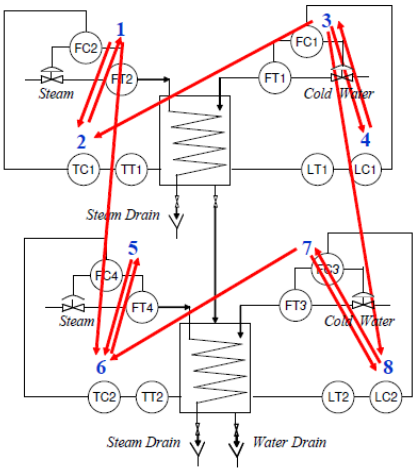


Fig. 3.14 – Example illustrating Control Loop Digraph (courtesy of: Jiang et al. [15])

Figure 3.14 shows a two tank system in which the top tank drains freely into the second tank[15]. FCs represent flow controllers, LCs represent level controllers and TCs represent temperature controllers. There are a total of 8 controllers in this example. Steam flow loops are cascaded with temperature loops and cold water flow loops are cascaded with level loops. All the 8 controllers have been marked from 1 to 8. All the connections are shown by red arrows.

It can be seen that there is a direct two-way interaction between nodes 1 and 2. This is because OP of controller 1(FC2.OP) can change the PV of node 2 (TC1.PV). Also, the OP of controller 2 (TC1.OP) can change the PV of node 1(FC2.PV). Node 1 also interacts directly with node 6 because FC2.OP will affect TC2.PV since the top tank drains freely into the bottom tank. It can also be seen that node 1 will affect node 5 because when TC2.PV changes, it will change FC4.PV but since the interaction between node 1 and node 5 is not direct, we do not connect them in the control loop digraph. If node 5 were not cascaded with node 6, node 1 will not directly influence node 5. In short, control loop digraphs only show connections between nodes when there is a direct interaction between OP of one node with the PV of another.

In figure 3.14, further distinction can be made by way of drawing a dashed arrow to indicate that increase in OP of controller A causes decrease of PV of control loop B and a solid arrow indicating that increase in OP of controller A would cause an increase in PV of control loop B. This distinction can be very useful when a root cause analysis is to be done. However, for the purpose of this study we are interested only in defining a 'span of influence' for each of the 17 controllers in the reactor unit. Span of influence will be defined in the section that talks about Reachability Matrix.

Currently there are no good automated ways of constructing a Control Loop Digraph. Therefore when constructing a Control Loop Digraph, one has to

understand fully well the process information and topography of the unit/plant. Process and Instrumentation diagrams can help to make this process easier. Decisions have to be made by qualitatively understanding the process. It can then be validated by finding correlation between OP and PV of the control loops where a connection has to be verified. However, one has to be careful about drawing inferences based on correlation between OP and PV of two control loops. Correlation does not always imply causality; hence a high correlation between OP and PV is not sufficient to suggest a connection. Therefore, the Control Loop Digraph must first be constructed based on process knowledge. The connections can then be verified by correlation value[2]. If the correlation is high enough (0.7 or more), one can be reasonably confident about a connection. If the correlation is not very high (0.4 or less), there are reasons to doubt a connection in Control Loop Digraph. Anything between 0.4 and 0.7 is a grey area and perhaps P&ID diagrams should be referred to in order to validate a connection.

Presented in figure 3.15 is the Control Loop Digraph for the industrial reactor units. It is the same for all three reactor units because they are all structurally similar with a similar control scheme. In this control loops digraph we have not made the distinction between positive and negative connections because a root cause analysis is not of interest to us. In the following sections, we will describe the concepts of Adjacency and Reachability matrices which will help us identify those controllers that have more of a potential to affect other control loops in the unit/plant.

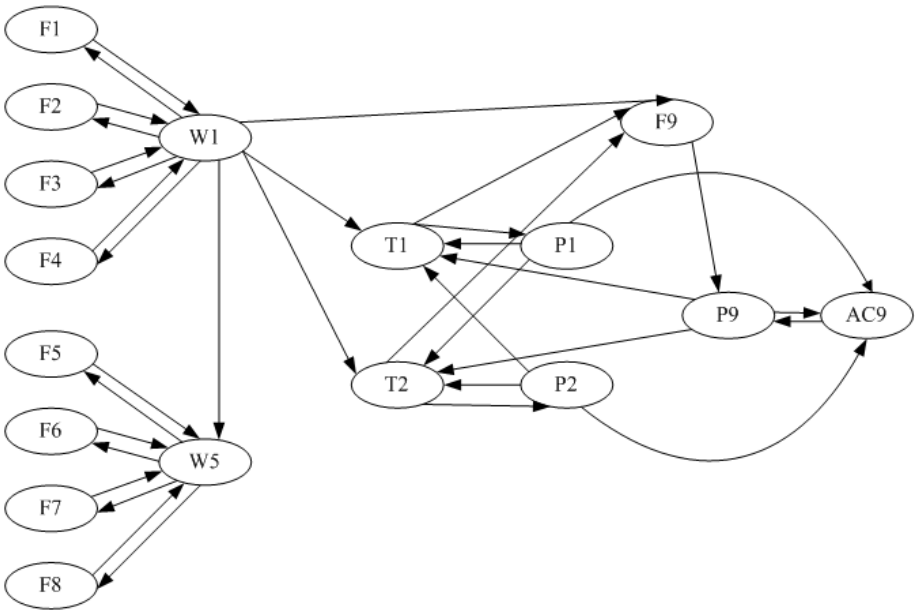


Fig. 3.15 – Control Loop Digraph of the reactor unit with 17 controllers

3.5.1 Adjacency Matrix and Reachability Matrix for the reactor unit

The Adjacency Matrix (X) has 17 rows and 17 columns (corresponding to the 17 controllers). If there is a directed arrow from controller A to controller B, we enter '1' in the (A, B) cell of the matrix, otherwise we enter '0'.

The Reachability Matrix (R) takes into account all possible paths between a set of controllers and not necessarily direct ones as is the case with the adjacency Matrix. R can be calculated as:

$R=(X+X^2+X^3+\ldots+X^n)^{\#}$

n is the total number of controllers in the unit. '#' is a boolean (algebra) operator which returns a value of 1 if the sum is greater than 1 and returns 0 if the sum is zero. Results from the Reachability Matrix show that the controller with the highest reach is W1. It can reach all the other controllers in the reactor. The other feed flow controllers (F1-F4) also have an equally big reach. All controllers other than W5, F5-F8 have a reach of 7. The steam flow controllers have a reach of 5.

Table 3.3 shows the Reachability matrix for the controllers in these reactors. Highlighted in green are those controllers that were identified as important for reactor R1 based on feature selection results. One will observe that the controllers with highest reach are not often predicted as ones being important by the prioritisation algorithm. This indicates that static ranking of controllers is probably not the optimal way of rank ordering controllers to prioritise their maintenance effort. As can be seen from the distribution histograms, you have more good days when the important controllers perform well. The same cannot be said of controllers with highest reach.

	W1	W5	F1	F2	F3	F4	F5	F6	F7	F8	AC9	P9	P1	P2	T1	T2	F9	Reach
W1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
W5	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
F1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
F2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
F3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
F4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
F5	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
F6	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
F7	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
F8	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
AC9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
P9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
P1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
P2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
T1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
T2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
F9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7

Table 3.3 – Reachability Matrix for the three Reactors

Chapter-4

Case study: Controller Prioritization for the Tennessee Eastman Challenge problem

The Tennessee Eastman Challenge problem (TE problem) was first proposed by Downs and Vogel (1993)[16] as a test-bed problem on which to test control schemes, fault detection techniques and other process control technologies in general. The simulation for this problem was provided by Downs and Vogel in the form of FORTRAN sub-routines. Since then, it has been adapted into MATLAB, SIMULINK and other programming languages. This simulation is very useful because it imitates a real industrial process. It has a total of 5 sub-units: Reactor, Condenser, Compressor, Vapour/liquid separator and Stripping column. This gives the process control community a chance to test out their ideas on a simulated case study before applying it on a real process. A brief description of the Tennessee Eastman problem follows:

The process produces two products (G & H) from four reactants (A, C, D & E). There is one byproduct (F) and one inert (B). Therefore, this process has a total of 8 components. The reactions describing this process are as follows:

$$A(g) + C(g) + D(g) \rightarrow G(liq) \quad (4.1)$$

$$A(g) + C(g) + E(g) \rightarrow H(liq) \quad (4.2)$$

$$A(g) + E(g) \rightarrow F(liq) \quad (4.3)$$

$$3D(g) \rightarrow 2F(liq) \quad (4.4)$$

All reactions are irreversible and exothermic.

The process has a total of 41 measured variables and 12 manipulated variables thus giving the control scheme designers a total of 12 degrees of freedom. In their paper, Downs and Vogel also specify some control objectives that the designers should adhere to. Most of these objectives relate to minimizing variability in some key process variables. A more detailed description of the TE process follows in section 4.1.

4.1. Description of the TE process

Gaseous reactants are fed to the reactor where they react to form liquid products. Since the reactions are exothermic, the reactor is provided with a cooling bundle to remove the heat of reaction. Products and unconverted reactants leave the reactor in gaseous form and enter a condenser to condense the products. The stream then enters the vapour liquid separator where the gaseous components recycle back through a centrifugal compressor to the reactor feed. Condensed components move to a product stripping column where they are mixed with stream 4 (as shown in figure 4.1) comprising reactants A & C. Products G & H exit the system from stream 11 and enter a downstream refining section which is not included in the TE process. Inert (B) and byproduct (F) exit the process through purge stream from the vapour liquid separator. Readers are referred to Downs and Vogel [16] for a more detailed description of the process.

Shown in figure 4.1 is the schematic of this process depicting the various unit processes and streams that are a part of the Tennessee Eastman process.

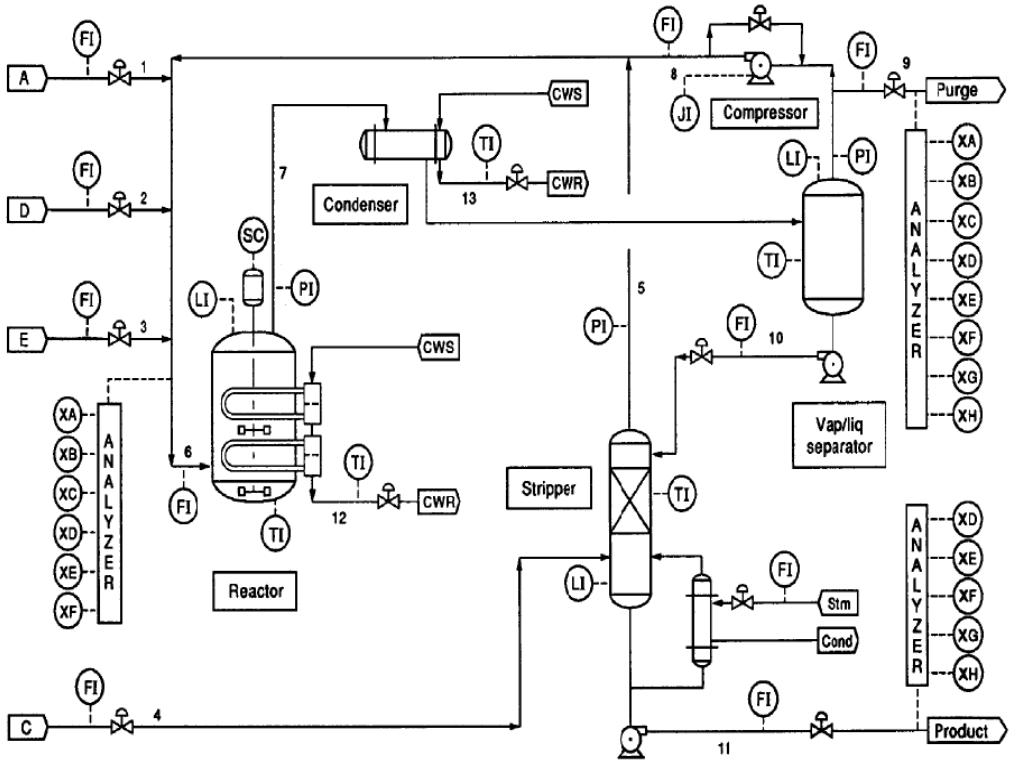


Figure 4.1 – Schematic of the Tennessee Eastman Process

This process has six specified modes of operation with varying production rates and G/H mass ratio in the product line (stream 11 in the figure 4.1). In order to make the process as realistic as possible to a real industrial process, the TE problem is also provided with 20 disturbance variables. The disturbance variables are summarized in table 4.1:

Disturbance	Process Variable	Type
IDV(1)	A/C feed ratio, B composition constant (stream 4)	Step
IDV(2)	B composition, A/C ratio constant (stream 4)	Step
IDV(3)	D feed temperature (stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (stream 1)	Step
IDV(7)	C header pressure loss-reduced availability (stream 4)	Step
IDV(8)	A, B, C feed composition (stream 4)	Random
IDV(9)	D feed temperature (stream 2)	Random
IDV(10)	C feed temperature (stream 4)	Random
IDV(11)	Reactor cooling water inlet temperature	Random
IDV(12)	Condenser cooling water inlet temperature	Random
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	Unknown	Unknown
IDV(17)	Unknown	Unknown
IDV(18)	Unknown	Unknown
IDV(19)	Unknown	Unknown
IDV(20)	Unknown	Unknown

Table 4.1 – List of disturbances in TE Process

As mentioned earlier, Downs and Vogel have specified multiple control objectives that any attempted control scheme should work towards attaining. In the literature, there are multiple control schemes available for this process as proposed by different authors. These control schemes often vary in their

choice of manipulated variable for a given controlled variable. For the purpose of this case study, we chose the control scheme developed by Ricker et al[17]. Ricker has also provided a SIMULINK model of their control scheme. The controller prioritization algorithm remains unchanged from the previously discussed SVM-classifier based algorithm in the industrial case study in chapter 3. A brief description of Ricker's control scheme for the TE process is described in section 4.1.

4.2. Description of Ricker's SIMULINK model

The model consists of 19 PID controllers. The design of this control scheme is based on a decentralized approach wherein the plant is partitioned into sub-units and controllers are designed for each sub-unit.

As mentioned earlier, the TE process has 12 degrees of freedom, i.e. 12 manipulated variables. These are:

Variable number	Variable name
xmv(1)	D feed flow (stream 2)
xmv(2)	E feed flow (stream 3)
xmv(3)	A feed flow (stream 1)
xmv(4)	A and C feed flow (stream 4)
xmv(5)	Compressor recycle valve
xmv(6)	Purge valve (stream 9)
xmv(7)	Separator pot liquid flow (stream 10)
xmv(8)	Stripper liquid flow (stream 11)
xmv(9)	Stripper steam valve
xmv(10)	Reactor cooling water flow
xmv(11)	Condenser cooling water flow
xmv(12)	Agitator speed

Table 4.2 – List of manipulated variables in TE process

According to the control objectives specified by Downs and Vogel, six measured variables must be controlled at specified setpoints:

1. Production rate
2. Mole %G in product
3. Reactor pressure
4. Reactor liquid level
5. Separator liquid level
6. Stripper liquid level

This leaves us with just 6 degrees of freedom. One of these, the agitation rate is fixed at 100% to maximise the heat transfer in the reactor. The remaining 5 degrees of freedom are utilized in controlling the following variables:

7. Reactor temperature
8. y_{AC} – the combined %A + %C in the reactor feed
9. y_A – amount of A in the reactor feed relative to the amount of A+C (as a percent)
10. Recycle valve position
11. Steam valve position

Presented in table 4.3 is a summary of the 19 PID loops used in Ricker's control scheme. It should be noted that 17 of these loops assist the normal operation of the plant. Two loops are described as override loops that assist with abnormal operation. These loops (Loop 18 & 19) control maximum reactor pressure and maximum reactor level respectively.

Loop	Controlled variable	Manipulated variable	Gain	Integral time (min)
1	A feed rate (stream1)	xmv(3)	0.01	0.001
2	D feed rate (stream2)	xmv(1)	1.6×10^{-6}	0.001
3	E feed rate (stream3)	xmv(2)	1.8×10^{-6}	0.001
4	C feed rate (stream4)	xmv(4)	0.003	0.001
5	Purge rate (stream 9)	xmv(6)	0.01	0.001
6	Separator liquid rate (stream 10)	xmv(7)	4.0×10^{-4}	0.001
7	Stripper liquid rate (stream 11)	xmv(8)	4.0×10^{-4}	0.001
8	Production rate	Production index, Fp	2.0	400
9	Stripper liquid level	Ratio in loop 7	-2.0×10^{-4}	200
10	Separator liquid level	Ratio in loop 6	-1.0×10^{-3}	200
11	Reactor liquid level	Setpoint of loop 17	0.8	60
12	Reactor pressure	Ratio in loop 5	-1.0×10^{-4}	20
13	Mol %G in stream 11	Eadj	-0.4	100
14	yA	Ratio in loop 1	2.0×10^{-4}	60
15	yAC	Sum of r1+r4	3.0×10^{-4}	120
16	Reactor temperature	Reactor coolant valve	-8.0	7.5
17	Separator temperature	Condenser coolant valve	-4.0	15
18	Maximum reactor pressure	Production index, Fp	2.0×10^{-6}	0.001
19	Reactor level override	Recycle valve, xmv(5)	1.0×10^{-6}	1.0×10^{5}

Table 4.3 – Summary of PID controllers used in Ricker's control scheme

Please note:

1. Fp is the production index which has a value of 100 at mode 1 of operation.
2. Eadj is an adjustment factor that is the signal from the feedback controller. It is used to adjust the flow rates of streams 1 and 4.
3. r1 - r4 are flow rates of streams 1-4 respectively

4.3. Generate process data from the SIMULINK model

For this case study, we need the following data:

1) Process data: using which an objective function for the performance evaluation of the process will be defined, and

2) Controller data: i.e. controller output (OP), process variable in the loop (PV) and controller set point (SP) data. This will be used to generate performance indices for the controllers.

For the purpose of data generation, we introduce a random disturbance generator function into Ricker's SIMULINK model. This disturbance generator function introduces different disturbances at different times by generating random integers between 0 and 20. 0 would introduce no disturbance and any other number would introduce the corresponding disturbance from table 4.2. In the propagation of disturbances, we have followed the suggestions made by Downs and Vogel that disturbances 14-20 should be used in conjunction with another disturbance and that they should be kept on for at least 24-48 hour time period. Random disturbances are generated every 6 hours otherwise.

Using this SIMULINK model, we were able to generate 166 days of process and controller data. All of the controller data was then passed through a commercially available software to generate performance indices of these controllers. The idea here is to simulate a real plant as closely as possible.

However, to keep things simple, we did not change the mode of operation of the TE process. Changing the mode often requires changing of controller tuning parameters. It is suggested that people following up on this study try the controller prioritization algorithm by operating at different modes and by trying out different control schemes as proposed by different authors.

4.4. Objective performance monitoring of the TE process

Downs and Vogel, in their paper, suggest that it is important to minimize the variance of G/H mass ratio in the product line. High variability affects the performance of the downstream process which utilizes the product stream from the TE process. This downstream process is, however, not a part of the TE process. Therefore, we select standard deviation of mass flow rates of G & H in the product line as our objective function for this case study. In keeping with the terminology used for the previously discussed industrial case study, one can observe in figure 4.3 that good days would be when the standard deviation is low (points below the green line), bad days are points above the red line and points between the two lines would be the 'in-between' days.

Figures 4.2 and 4.3 show the distinction between good and bad days for the TE process:

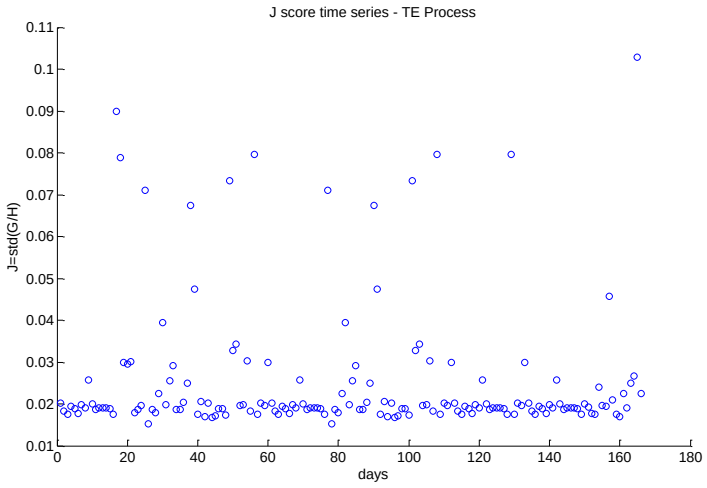


Figure 4.2 – J score time series for TE process

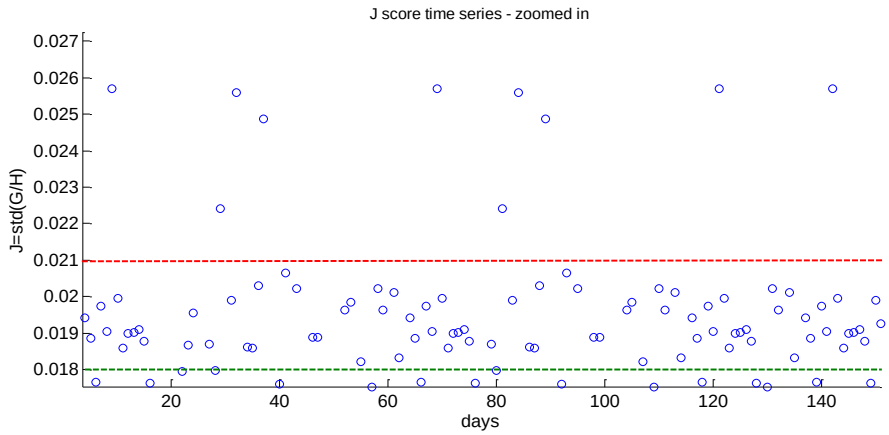


Figure 4.3 – J score time series for TE process (zoomed in)

We find that in 166 days, 68 days are operationally good days, 47 are operationally bad days and the rest are in-between days.

Next step, as with the previous case study, is to segregate the predictor variable data set upon which SVM classification models will be built. Unlike the previous case study, we do not change the process operating conditions for the process because we operate the plant only in mode 1. Therefore, process operating conditions are not included in the predictor variable data set. Also, there is no variable specified by Downs and Vogel which could be thought of as the incipient fault variable. Therefore, in this case, we only have the Controller Performance Monitoring data obtained from the commercial software as our predictor variable data set.

4.5. Two-class SVM and Controller Prioritization algorithm applied to TE Process data

The commercial software that we used to generate controller performance indices needs controller OP, SP and PV to generate performance indices for the controllers for each operational day. Due to the proprietary nature of the software, we can not disclose the algorithm that it uses to calculate the indices. Like the previous case study, the indices that we used for the controller prioritization algorithm stay the same:

I1: Settling time of controller compared with desired settling time

I2: Standard deviation of control error

I3: Oscillation in the control loop

We build 2-class SVM models on the good day and bad day data sets and observe that a 5-fold cross validation accuracy of more than 99% is achieved. This means that SVM is able to correctly predict good and bad days from their respective controller performance data with an accuracy of more than 99%. Since the classification model works well, we can proceed with the application of controller prioritization algorithm on the data set. As with the previous case study, the algorithm is applied in a Monte-Carlo fashion with 32 repeated runs and the results aggregated for these runs.

The Controller prioritization algorithm tells us that the 5 most important controllers in the TE process are as follows:

Stripper bottoms liquid level (C9)

Separator liquid level (C10)

A+C control in reactor feed (C15)

Separator temperature control (C17)

E feed flow controller (C3)

Hereafter, we change the good day and bad day thresholds to some different values to see if this makes a difference in the controller prioritization results. It is observed that when controller prioritization algorithm is run with different threshold boundaries in a Monte-Carlo fashion, the feature selection results do not change at all. In fact, when the separation boundaries are made wider, it is observed that 5 fold cross validation accuracy with 2-class SVM is 100%. This implies, that as we make the separation boundaries wider, we make the distinction between good and bad day controller performance values even wider thereby making the generalizable performance of the model even better.

4.6 Discussion of Controller Prioritization results

The controller prioritization algorithm results tell us that the five most important controllers with respect to lowering the variance in the product line are the ones listed above. Unlike the previous case study, here we have to resort to literature search and understanding of the process to try and validate the results. In section 4.7, we also present distribution histograms that are in tune with our findings: when important controllers perform well, the plant tends to have more good days.

Shown in figure 4.4 is a schematic (P&ID) of the control scheme developed by Ricker et al.

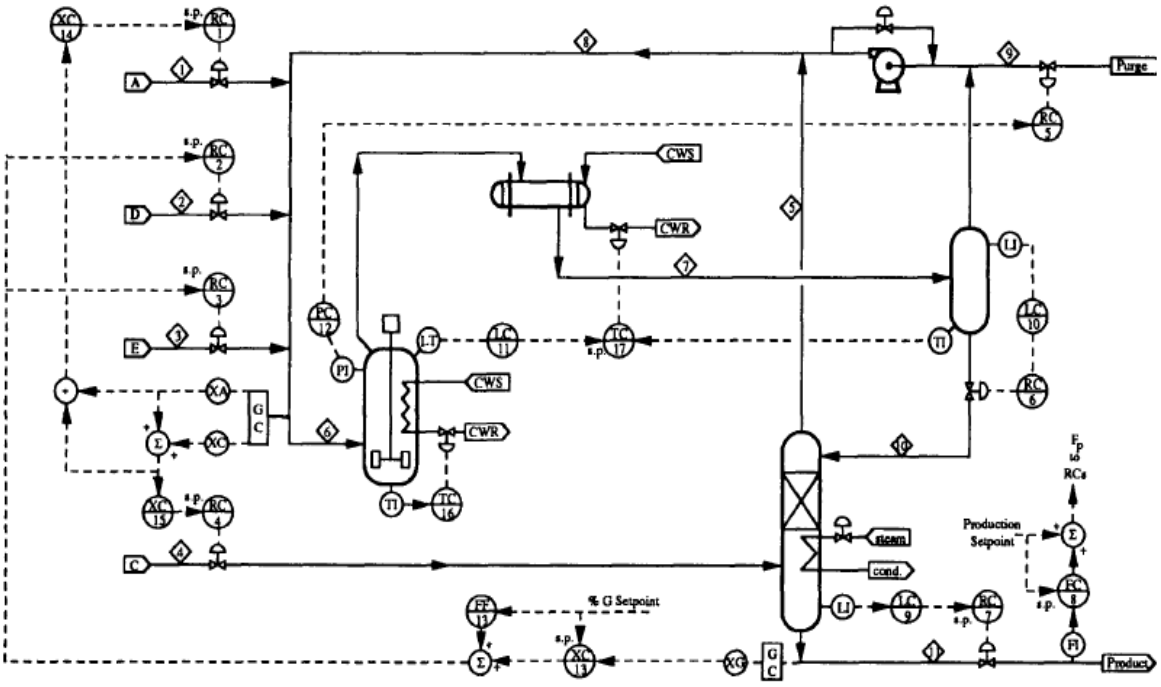


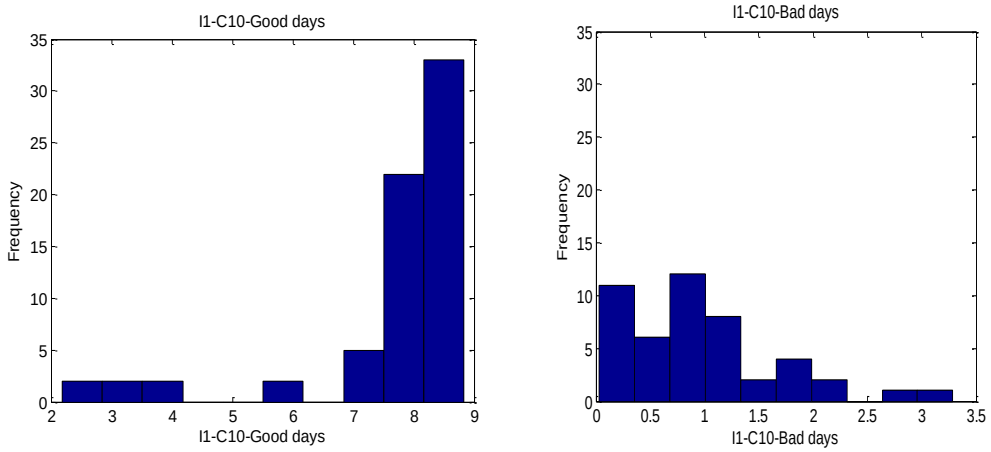
Figure 4.4 –P&ID diagram of the Control scheme (Ricker et al.)

One can clearly see that C9, C10 and C17 are the three controllers that directly impact the ratio in the product line. If the separator temperature control does not work properly, it is easy to see that it is going to affect the amount of G & H that go into the product line. It may also cause more by-products to enter the product line. Similar argument holds true for the level controllers C9 and C10. Price et al.[19] suggest that controller C17 does have an effect on product variability and removing this controller from the control scheme will increase product variability.

In table 6, controller C13 is mentioned as the one that controls %G in the product line. It seems to have the most direct relation with our objective function that controls the variability in product line. The manipulated variable for this controller is E_{adj} which actually manipulates flow rates of D and E entering the reactor. We see that the flow controller for E is identified as an important controller in the feature selection results. C13 is found to be an important controller as well and is ranked as one of the 10 most

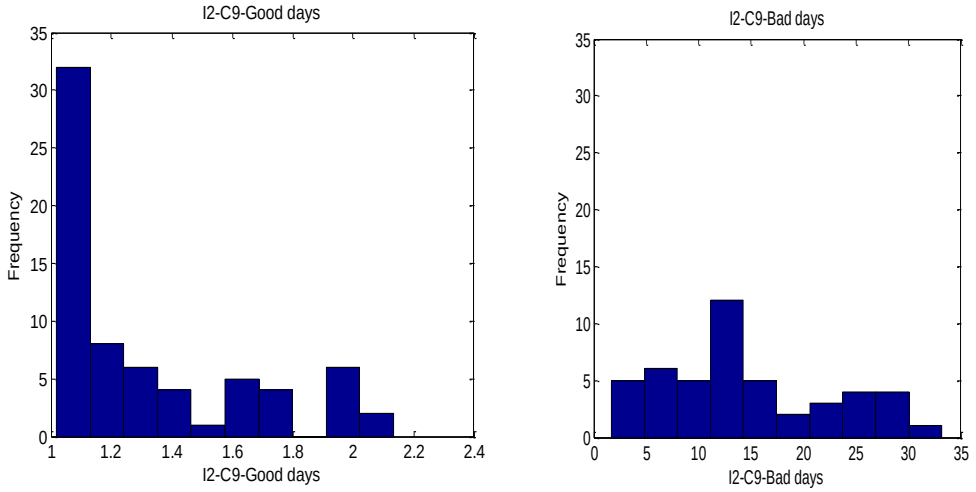
important controllers. However, it is not amongst the top 5. Through the distribution histograms presented below, we see that there is a marked distinction in the performance of the top 5 controllers on good days as compared with bad days.

Presented in figures 4.5 – 4.10 are distribution histograms of controller performance indices for some of the most important controllers:



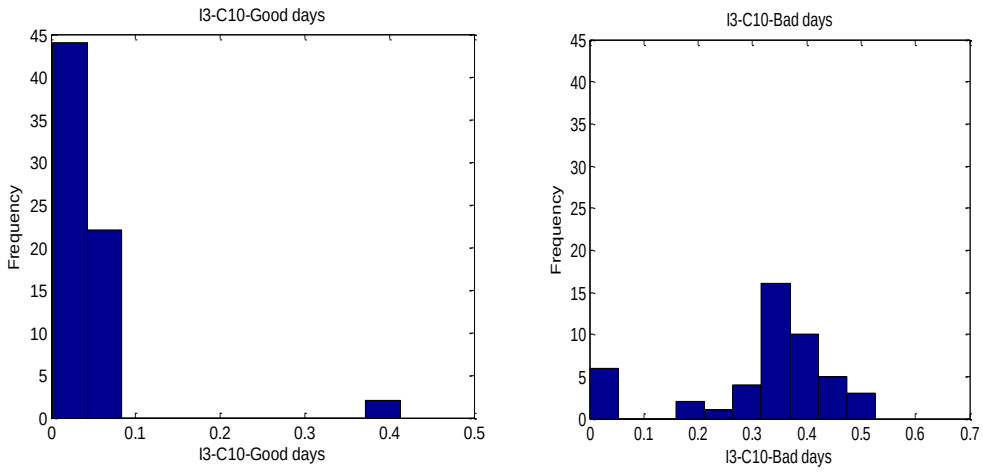
Mean: 7.67 Mean: 0.95

Figure 4.5 – Distribution histograms of index I1 for controller C10 over good days and bad days



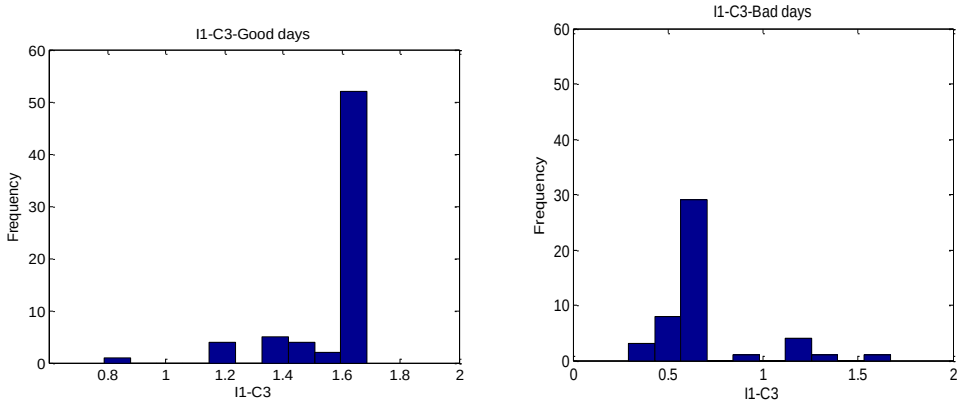
Mean: 1.31 Mean: 14.61

Figure 4.6 – Distribution histograms of index I2 for controller C9 over good days and bad days



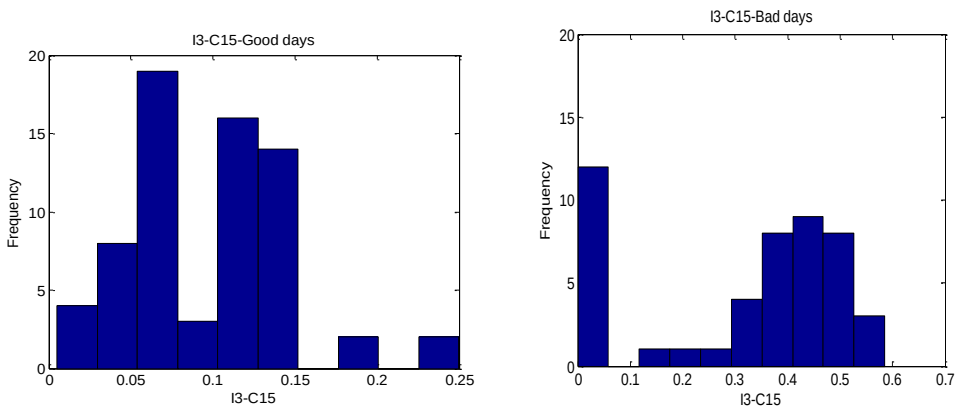
Mean: 0.04 Mean: 0.32

Figure 4.7 – Distribution histograms of index I3 for controller C10 over good days and bad days



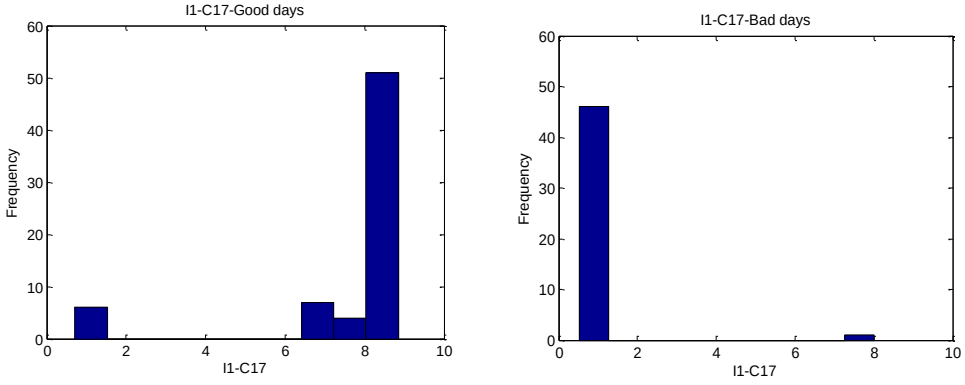
Mean: 1.59 Mean: 0.67

Figure 4.8 – Distribution histograms of index I1 for controller C3 over good days and bad days



Mean: 0.10 Mean: 0.31

Figure 4.9 – Distribution histograms of index I3 for controller C15 over good days and bad days



Mean: 7.75 Mean: 0.94

Figure 4.10 – Distribution histograms of index I1 for controller C17 over good days and bad days

These histograms again prove our point that when 'important controllers' perform well, you tend to have more good days. As can be seen, for all the important controllers, consistently, I_1 has a higher value; I_2 and I_3 have a lower value on good days as compared with bad days.

Chapter-5

Concluding Remarks and Future Work

In this work we have developed a framework for plant-wide performance monitoring by introducing the concept of objective performance monitoring of a unit operation/plant and correlating it with the controller performance indices. We discussed that it is important for us to find out controllers that affect process performance more than others so that plant engineers can focus their efforts in ensuring that these controllers are optimally tuned and well maintained at all times. We also saw through the distribution histograms and 3-class SVM methods that when the important controllers perform well, we have more good days than bad days. Therefore, in an industrial plant with more than a thousand controllers, it makes business sense to prioritize a subset of the most important controllers so that more energy can be spent in the optimal maintenance of these controllers.

Through the method of Control Loop Digraphs, we saw that it is possible to find controllers that have a high reach, i.e. greater potential to affect other controllers in the unit/plant. However, it turns out that these controllers are not necessarily the ones that cause good days and bad days.

We observed that Support Vector Machines classifiers are able to provide us with good classification accuracy for good days and bad days predictor variable data set. Therefore, we were able to carry out the application of a feature selection algorithm in order to identify controllers that contribute more towards the classification boundaries. We identified these controllers as important controllers and we were able to see through distribution histograms that good performance of these controllers meant good days for the plant. Therefore the plant engineers should focus on improving their performance.

All techniques that were discussed in this thesis are non-intrusive in nature. We were able to carry out this analysis remotely without running any experiments on the real plant. The idea is to make use of the wealth of information typically available to process control practitioners through data historian to extract useful information for better operation of the plant.

We have shown that Support Vector Machines show promise as a classification method for this purpose. It has worked very well on the two case studies that we looked at in this work. SVM is able to identify a subset of important controllers that are responsible for causing good and bad days. In its present form, this algorithm needs some customization in the sense that Objective performance is highly unit dependent. Therefore, in order to find important controllers, one needs to identify the objectives of the unit/plant and then apply the controller prioritization algorithm. The algorithm is fairly generic in the sense that it can accommodate any number/kind of performance indices generated by one/more types of controller performance monitoring softwares.

Interested researchers following up on the Tennessee Eastman case study are encouraged to apply the controller prioritization algorithm on the various control schemes available in literature and compare the results.

However, there remains more to be done in order to make controller prioritization applicable in process industries. For instance, one could try and make this algorithm online. The idea is that it is not always the same controllers that cause bad process performance. In order to identify controller(s) performing poorly at any given time there is a method that we would like to suggest. The method makes use of 3-class SVM. People following up on this work are recommended to try it out along with any other approach that they develop.

5.1 Online identification of poorly performing control loops

In section 3.3, we had discussed three-class SVM for identification of optimal performance regions for each controller. In order to make this algorithm online, the following approach is suggested:

At any given time, if the need arises to identify poorly performing loops which could be causing bad plant performance, do the following:

1. Run the controller performance monitoring software on data obtained from the last 24 hours of operation.
2. Calculate the performance indices for each controller
3. Use three class SVM to judge the class of this fictitious day (these 24 hours are not really one day and could be spread over two days)
4. Compare the performance indices values of this fictitious day with the optimal performance values and identify controllers that need to perform well in order to make this day a good day.
5. Once the controllers causing bad performance are identified, it is suggested that they be re-tuned or be checked for valve problems etc. in order to progress towards improving the process performance.

Bibliography

[1] A.J. Trenchard, H. Boder, How do you know: which control loops are the most important?, IEE Computing and Control Engineering, vol. 16 (4), pp. 24-29, 2005.

[2] F. Yang, D. Xiao, S.L. Shah, Qualitative Fault Detection and Hazard Analysis Based on Signed Directed Graphs for Large-Scale Complex Systems, in: Wei Zhang (Ed.), Fault Detection, ISBN: 978-953-307-037-7, INTECH, Available from: http://sciyo.com/articles/show/title/qualitative-fault-detection-and-hazard-analysis-based-on-signed-directed-graphs-for-large-scale-comp, 2010.

[3] A. Rahman and M.A.A.S. Choudhury, Detection of Control Loop Interactions and Prioritization of Control Loop Maintenance, Proceedings of ADCONIP 2011 (in press)

[4] V. Vapnik, Estimation of dependences based on empirical data, ISBN: 978-0-387-30865-4, Springer Verlag, New York, 1982.

[5] B.E. Boser, I.M. Guyon, and V.N. Vapnik, A training algorithm for optimal margin classifiers, Proceedings of the 5th annnaul ACM Workshop on Computational Learning Theory, pp. 144–152, 1992.

[6] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2, vol. 2, pp. 121–167, 1998.

[7] C.W. Hsu, C.C. Chang and C.J. Lin, A Practical Guide to Support Vector Classification, available through

http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[8] I.M. Guyon, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[9] Y.W. Chen, C.J. Lin, Combining SVMs with Various Feature Selection Strategies, http://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf.

[10] K. Crammer and Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machine, Journal of Machine Learning Research, vol. 2, pp. 265-292, 2001.

[11] U. H.-G. Kreßel, Pairwise classification and support vector machines, in: B. Scholkopf, C.J.C. Burges, and A.J. Smola(eds.), Advances in Kernel Methods-Support Vector Learning, ISBN: 978-0-262-19416-7 MIT Press, Cambridge, MA, pp. 255–268, 1999.

[12] V. Hölttä, Plant Performance Evaluation in Complex Industrial Applications, ISBN: 978-951-248-092-7 (pdf), Available from: http://lib.tkk.fi/Diss/2009/isbn9789522480927/isbn9789522480927.pdf

[13] S. Maldonado, R. Weber, A Wrapper Method for Feature Selection Using Support Vector Machines, Journal of Information Sciences, 179 (2009) 2208–2217

[14] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator", ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998, pp 3--30.

[15] H. Jiang, R. Patwardhan, S.L. Shah, Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix, Journal of Process Control, Volume 19, Issue 8, September 2009, Pages 1347-1354.

[16] J.J. Downs and E.F. Vogel, A Plant-wide Industrial Process Control Problem, Computers & Chemical Engineering, Vol. 17, No. 3, pp. 245-255, 1993.

[17] N. L. Ricker, Decentralized Control of the Tennessee Eastman Challenge Process, Journal of Process Control, Vol. 6, No. 4, pp. 205-221, 1996.

[18] SPIDER Machine Learning Toolbox for MATLAB, available through:

http://www.kyb.mpg.de/bs/people/spider/

[19] R.M. Price, P.R. Lyman and C. Georgakis, Throughput Manipulation in Plantwide Control Structures, Ind. Eng. Chem. Res., 1994, 33 (5), pp 1197–1207.

[20] S. Pareek, C. McNabb, S.L. Shah, Why do I care? A novel approach for relating control performance to business value, Proceedings of ISA Automation Week conference 2010.