**University of Alberta**

# Bayesian Methods for Control Performance Monitoring

by

**David Oluwaseyi Akande** ©

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

Edmonton, Alberta
Fall 2008

*Your file Votre référence*
*ISBN: 978-0-494-47176-0*
*Our file Notre référence*
*ISBN: 978-0-494-47176-0*

*Dedicated to* ***God Almighty*** *for His faithfulness, mercy and love, "He raiseth up the poor out of the dust, He lifteth up the needy from the dunghill, To make them sit with princes, And inherit the throne of glory: For the pillars of the earth are The Lord's, And he hath set the world upon them (1 Samuel 2:8)".*

*To my father,* ***Adebola Oladimeji Akande****, to the memory of my late mother,* ***Deborah Oyelekan Akande****, to my little nephew,* ***Daniel Morayooluwa Akande****, and to my one and only sunshine* ***Bukola****- my true love.*

*The saying is true: "eniyan ni aso mi (my people are my covering)".*

# Abstract

Monitoring and assessment of control systems have become an integral part of industrial process control applications due to their usefulness in meeting target objectives and increasing process productivity. In this work, we propose new approaches to controller monitoring by investigating the use of run lengths, Markov chains and ultimately, Bayesian analysis. Bayesian analysis is important for making decisions in the presence of uncertainty. Using model predictive controllers as a case study, we have addressed the issue of controller constraint tuning via a continuous-valued objective function within a Bayesian probabilistic framework. The benefits of this approach includes: a more generalized definition of *quality variables*; the development of a mathematically elegant formulation of the problem to address linear and quadratic objective functions, thereby obtaining closed form solutions; and maximum-likelihood location determination of the *quality variables* in the decision making process. The approaches are illustrated with simulations and pilot-scale experiments.

# Acknowledgements

Writing this thesis has been both a challenging and stimulating experience. I wish to express my sincere thanks to my supervisor, Dr Biao Huang, for his support, deep understanding and strong commitment to excellence. Working with him has been a great learning experience. I appreciate the "push" he consistently gave during the course of this graduate program.

I also consider myself fortunate to be a member of the Computer Process Control (CPC) group at the University of Alberta. I wish to thank other members of faculty (Dr Sirish Shah, Dr Fraser Forbes, Dr Jong Min Lee, Dr Amos Ben-Zvi) for their active research work and their contributions to making the research environment rich and conducive.

This thesis and the whole Masters experience could not have been possible without the love, sacrifice and support of my family. They have always been a source of strength and encouragement. Only God can bless and reward them.

My thanks also goes to the members of my research group: Nikhil Agarwal and Kwanho Lee for their assistance, support and collaboration on this project, Salim Ahmed for his friendship and assistance. Fei Qi, Moshood, Xinguang also deserve my thanks. My friends and office mates: Venkat, Barath, Sankar, Hussam, Hector, Deng Hua, Qi Li for their friendship and support, for all the times we laughed together even when the going was difficult.

I was fortunate to have some meaningful interaction with Engineers and Managers at Syncrude Canada (Aris Espejo, Dan Brown, Edgar Tamayo, John MacGowan and all others). This was the high point of my research and I am grateful for the experience.

My story will not be complete without mentioning Dr Sola Adeyinka and family, Emem Madu and Ifeoma Okoye (nee Onyerikam) for their assistance at the start of my program. Your support was much needed in facing the challenges of settling down after resuming late. I will do my best to mention my Naija people by name: Idowu, Kingsley, Ayo, Bukky, Collins, Ejike, Godwin ... having you guys around made life enjoyable. Thanks to the love and friendship of John and Didi Masiala; Dale and Heather Aceron; Jonathan, Marcia and Carol Krenz; Edmonton has become my "home away from home".

Finally, I would like to thank the Department of Chemical and Materials Engineering at University of Alberta and the Natural Sciences and Engineering Research Council (NSERC) for providing financial support for the period of my graduate studies.

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## 1.1 Motivation

Process control strategies and applications have become an intricate part of the operations in process and manufacturing industries, with stabilization of the process operation, quality and safety considerations, asset management/utilization and increasing production efficiency and profitability as the major objectives (Bauer and Craig 2007, Harris and Yu 2003, Jelali 2006).

In current industrial scenarios, the average process industry could have hundreds (sometimes thousands) of controllers, most of them being simple PID controllers. For example, Paulonis and Cox made reference to a case of 14,000 PID loops in the Eastman Chemical Company (Paulonis and Cox 2003). In practice, however, there are usually multivariable controllers, which in most cases are model predictive controllers (MPC), that act as supervisory controllers for the PID loops (Qin and Badgwell 1997, Harris *et al.* 1999). This concept of the supervisory behaviour of the MPC is illustrated in Figure 1.1 (Qin and Badgwell 1997). In view of the multiple interactions between control loops, it is inevitable that the overall performance of a process will change whenever a single controller or several loops are tuned. In addition to this, poor performance could arise from deterioration of equipment (e.g due to fouling), wearing out of actuators, change in production objectives or setpoints, etc (Huang and Shah 1999, Harris *et al.* 1999, Jelali 2006). Such real-life scenarios are evidently very complex systems and tools that assist in assessing performance and/or monitoring of such systems become of great value.

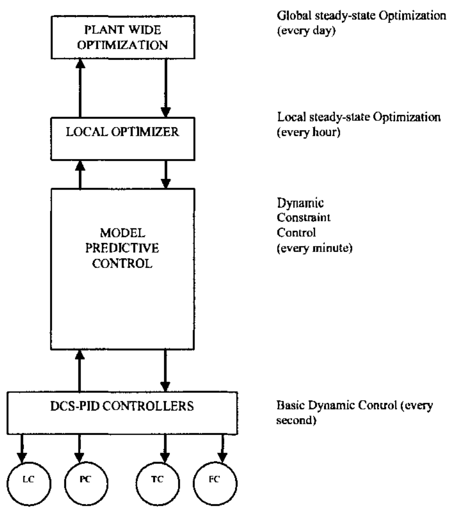



Figure 1.1: Schematic illustrating supervisory status of MPC

It is widely held that the work by Harris (Harris 1989) has proven to be seminal to much of the advances in controller performance assessment (Xu *et al.* 2007, Jelali 2006) and subsequently, the assessment of controller performance has grown from the use of the minimum variance (MVC) benchmark, which was initially applicable for univariate (SISO) systems, to the the development of other benchmarks for multivariable (MIMO) systems, such as those based on linear quadratic gaussian(LQG), and model predictive control (MPC) algorithms (Huang and Shah 1999, Qin and Yu 2007, Harris *et al.* 1999).

The applications of performance assessment tools have extended from a simple single-loop PI controller (Harris 1989) to plant-wide applications such as applications to refinery-wide performance assessment (Paulonis and Cox 2003, Thornhill *et al.* 1999). The scope of controller performance assessment (CPA) has also extended to areas like fault, nonlinearity and oscillation detection (Jelali 2006, Harris *et al.*

1999). There are excellent reviews on controller performance assessment technology and applications in the literature (Jelali 2006, Bauer and Craig 2007).

It has been stated that up to 60% of all industrial controllers have performance-related problems (Harris *et al.* 1999). An important cause of poor performance of industrial controllers is poor tuning or lack of maintenance. Others include design or equipment malfunction, inappropriate control structure, and poor (or the lack of) feed-forward compensation (Harris *et al.* 1999, Jelali 2006).

The use of statistical analysis in the assessment of control loops cannot be overemphasized. MVC as a benchmark for controller performance, which is a foundation for much of the performance assessment activity in the literature, is based on a time series model (Harris 1989). The use of various other statistical tools is common in the literature on controller performance assessment. The review by Jelali (2006) gives numerous examples and references. The statistical tools can be parametric or non-parametric in nature. Certain non-parametric approaches of interest, such as Markov chains and *run length distributions*, have been applied to controller performance assessment in recent literature (Harris and Yu 2003, Lu 2007, Li *et al.* 2004).

In the field of controller performance assessment, the work by Xu *et al.* (2007) has provided a framework for performance assessment for MPCs based on constraints and variability tuning. This work uses linear matrix inequalities (LMI) to formulate and solve the problem. The optimization objective function under consideration is the economic objective function of the MPC. The quality variables are the variables in the economic objective function with non-zero linear and quadratic coefficients. These variables affect the amount of potential benefits or profits that can be obtained from the process. The approach has been termed LMIPA (Linear Matrix Inequality *based* Performance Assessment) (Xu *et al.* 2007). The potential benefits that can be extracted from the process is estimated by considering scenarios where the mean values of the quality variables under consideration are moved closer to their optimum values by tuning the constraints or by variability reduction. A more realistic scenario would be considered where the mean value and the variance (i.e. the distribution) of the variable are used.

An extension of the work of Xu *et al.* (2007) has been proposed by Argawal (2007). Reference can also be made to his two papers in IECR, (Agarwal *et al.* 2007*a*, Agarwal *et al.* 2007*b*) in which the distributions (based on the means and variances) of the quality variables are considered. This extension was considered under a probabilistic framework using Bayesian statistics. The MPC economic objective function (as used by Xu *et al.* (2007)) was discretized into zones of profit and each zone of profit has associated probabilities for the quality variables to be above, within or below desired product specifications. Associated with the discretization step is some loss of information since the objective function is actually continuous-valued in nature. Subsequent analysis was shown using the Bayesian network for inferential purposes. The Bayesian based analysis was classified as *decision making* and *decision evaluation*. Decision evaluation means that the expected returns from the process are inferred if certain decisions regarding limit changes are made. Decision making means that the combination of constraint changes of relevant variables that will be needed to achieve a user-specified target value of the expected return is determined, based on the maximum a posteriori explanation obtained from the network when the target value is supplied as evidence.

In light of the above, ample room exists for research and investigation in controller performance assessment with respect to a new set of non-parametric statistics and the consideration of uncertainty in performance assessment using a Bayesian framework. This thesis aims to extend the scope of the previous works (Xu *et al.* 2007, Agarwal 2007) in this area. This thesis is primarily driven by the need to develop industrially applicable algorithms and to address certain open areas in the research on controller performance assessment.

As extensions of previous work in the literature, we propose the use of Markov chains and Bayesian networks to analyze *run length distributions*, for controller performance assessment, in the initial part of this thesis. In the latter part of this thesis, in particular, we propose an extension of the LMIPA approach by considering the economic objective function of the MPC as a continuous valued function within a Bayesian statistics framework (Xu *et al.* 2007, Agarwal 2007). By using a continuous valued objective function, the associated loss of information due to discretization is avoided. The occurrence of statistical dependence between quality variables is also a noteworthy consideration which we seek to address in this thesis. We also propose a methodology for modeling the dependence of quality variables within a Bayesian framework based on the continuous valued function proposed. Although one of the major strengths of the Bayesian approach is its applicability to cases of non-gaussianity, for the purposes of this thesis, we have assumed Gaussian probability distributions due to the fact that generally, a sum of random variables with *any* distribution tends toward a normal (Gaussian) distribution based on the central limit theorem; and the normal distribution has certain properties that are attractive and mathematically tractable (Bishop and Welch 2003). This Gaussian assumption also allows us obtain closed-form solutions.

The proposed algorithms as discussed in this thesis have been used to develop tools (graphical user interfaces) that are used for controller performance assessment. These tools were developed using MATLAB (MathWorks 2007*a*). This software is extensively used in research, industrial and academic circles. MATLAB codes are written using functions that are both easy to comprehend and implement. They are high level codes and are easy to manipulate. A major advantage of MATLAB is that it incorporates numerous toolboxes that can be used for various forms of analysis. In this work, the toolboxes we have used include the *System Identification*, *control*, *statistics* and *optimization* toolboxes. Other toolboxes used are the *Bayesian network toolbox* (BNT) (Murphy 2007, Murphy 2004*a*, Murphy 2004*b*, Murphy 2001), *SeDuMi* (Sturm 1999) and *SDPT3* (Toh *et al.* 1999) toolboxes, and the *YALMIP* (Lofberg 2004) interface. *YALMIP* is an easy to use interface that utilizes numerous optimization toolboxes.

In the sections that follow, we provide brief introductions to the concepts that are used in this thesis, which will form a basis for the subsequent overview of the thesis that will be provided.

## 1.2 Run-Length distributions

A run is defined as *an uninterrupted sequence* of a particular data value or state (Balakrishnan and Koutras 2002). The use of the term "*run length*" is quite common in certain areas of image and data processing. Run lengths are used as a means for data and image processing and compression, and various applications are available in the literature. Since data or image encodings can be broken down into particular sequences with inherently repeated patterns, they can be grouped as runs and subsequently analyzed. (Fatemi 1992, Chandra and Chakrabarty 2001, Galloway 1974, Tsai *et al.* 1996). Figure 1.2 illustrates the concept of *run lengths*. From Figure 1.2, the *run lengths* are as follows: $1, 3, 2, 3, 4, 4, 1, 2, 1, 1, \cdots$, etc.

*Run lengths* (more commonly the average *run lengths*, which is defined as the average time taken for a process to fall outside the specified control limits (Wardell *et al.* 1994)) have also been extensively applied in some areas of statistical process control with application to control charts (e.g Shewhart charts, CUSUM charts etc) (Robinson and Ho 1978, Brook and Evans 2003, Wardell *et al.* 1994). The *run length* is said to be one of the most important properties of any statistical process control (SPC) chart (Wardell *et al.* 1994). *Run lengths* have also been used as a test of randomness with applications in production processes via a *Runs test.* They are also useful in detecting non randomness in quantitative measurements over time (i.e. time series) (Wackerly *et al.* 2002).

The use of *run lengths* in engineering applications, particularly in controller performance monitoring, has been relatively sparse. Recent work by Li *et al.* (2004), has introduced possibilities for further application of this non-parametric statistic for controller performance monitoring applications beyond the previous scope of conventional statistical process control.

The basic idea of the use of *run lengths* to indicate *an uninterrupted sequence* is common to all the aforementioned areas of application. The particular method of analysis using *run lengths* is generally adapted to the end-user's personal objective. For example, the method of analysis required when using *run lengths* to determine randomness in data is different from the analysis required when control charts are considered. In this work we illustrate the application of *run length distributions* for controller performance assessment as introduced by Li *et al.* (2004) and then we introduce two new applications of *run lengths* to controller performance using Markov chains and Bayesian networks.

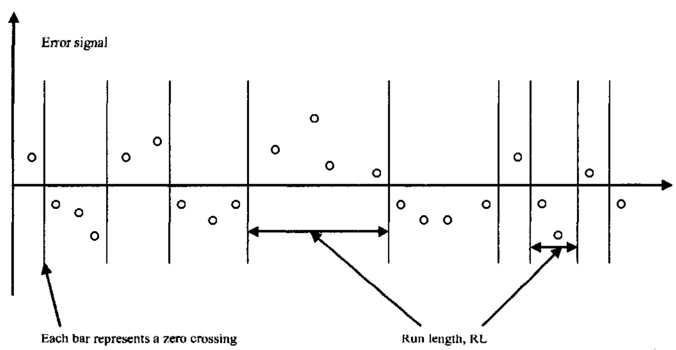



Figure 1.2: Illustration of *run lengths* and zero crossings

## 1.3 Markov chains - overview

A Markov process is a random process for which the current state of the process completely determines the state of the process at the next time instant. The Markov property implies that the past does not influence the future behaviour/state of the process. A Markov chain is a Markov process with a finite or countable number of states (Kemeny and Snell 1976).

Assuming that we have a state space, S, which contains q possible outcomes, the *i*th element of s is described as $s_i$, $i = 1, 2, \ldots, q$. Also, let the sequence of n observations of the Markov Process be $X_t, t = 0, 1, 2, \ldots, n$.

The states of a Markov chain can be classified into transient and ergodic sets. A transient set is a set which once left, is never again re-entered. For an ergodic set, however, once it is entered, it can never be left. An absorbing state is a state which is the only element of an ergodic set. For this state, $s_i$, the probability, $p_{ii}$ must be 1 and all other probability values in that row of the P matrix are 0.

If a Markov chain has all non-transient states as absorbing states, it is called an absorbing chain (Kemeny and Snell 1976, Lu 2007). An ergodic Markov chain has no transient sets, and has a single ergodic set. For a homogeneous Markov chain, the presence of only one eigen value of the P matrix with magnitude of unity, is a necessary and sufficient condition for a Markov Chain to be ergodic (Seneta 1973, Harris and Yu 2003).

## 1.4 Bayesian-based performance assessment

This approach to performance assessment of MPC is based on the LMIPA algorithm (Xu *et al.* 2007). The algorithm optimizes the MPC economic objective function which is both deterministic and quadratic in nature. The sign of the linear coefficients of the objective function refers to the decision as to whether the variable should be minimized or maximized. A positive linear coefficient indicates minimization of the variable while a negative linear coefficient indicates maximization of the variable. The quadratic coefficients address the question regarding keeping the process at specified target values. The magnitude of the quadratic coefficient indicates how desirable it is to keep the process at its target values. The constrained optimization problem is solved using the method of linear matrix inequalities (LMI) via the *Schur complement*

(Xu *et al.* 2007).

The LMIPA approach obtains the optimum mean operating point and since the optimization is deterministic, the associated results (the mean operating points and the subsequent constraint tunings analysis) are also deterministic. In reality, however, industrial process variables operate *around* a mean operating point and *not at* a mean operating point; therefore there is some degree of uncertainty associated with using deterministic results.

Bayesian analysis has been proposed as a means to accommodate the associated uncertainty and the distribution of the data (Agarwal 2007). This is achieved by using the probability distributions associated with the constraint change decisions. The Bayesian based LMIPA allows the data distribution to be considered in the analysis and assessment of the effects of constraint change of the variables of the MPC on the expected returns from the process. The improvements that this approach provides will be discussed in this thesis.

## 1.5 Thesis outline and relationship between thesis chapters

Before going into the details of the contents of this thesis, the intricate relationship between the Chapters needs to be explained. Chapters 2 and 3 are based on the concept of *run lengths* (which has been briefly introduced and will subsequently be further discussed), and the use of *run length distributions* as a statistic for controller performance monitoring leads us to consider certain parametric and non-parametric statistics in its analysis.

The multi-tank (pilot scale) experimental system which will be used throughout the thesis is also introduced and described in Chapter 2. This pilot scale experiment forms a link between Chapters 2, 4 and 5 because it is used to illustrate and experimentally validate the various methods proposed in the thesis. Finally, the use of Bayesian inference as a means of analysis in the presence of uncertainty also links Chapters 3, 4 and 5 of the thesis. It is important to note (as the title of the thesis indicates) that Bayesian analysis is the main theme of this thesis; the preceding chapters (Chapters 2 and 3) are provided to describe the "journey" that was taken to arrive at this desired end. The outline of this thesis is illustrated in Figure 1.3 and it shows the overlap between the chapters.

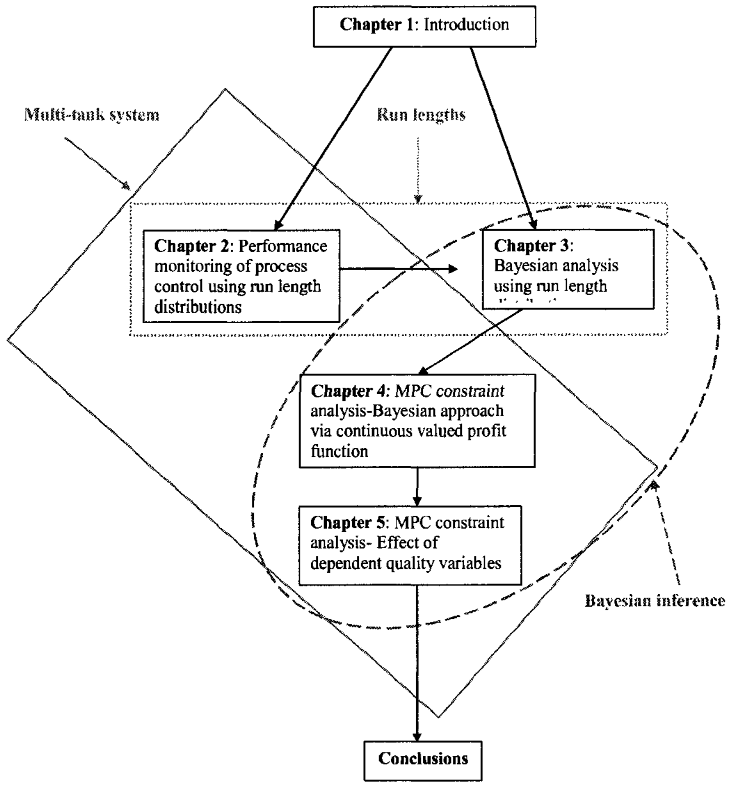



Figure 1.3: Schematic illustrating relationship between chapters of the thesis

In Chapter 2, we introduce the concept of *run lengths* and illustrate its use for dynamic performance monitoring based on the work of Li *et al* (2004), and we also introduce the use of Markov chains as another means of analyzing controller performance using *run lengths*. The multi-tank (pilot scale) experiment which will be used for illustrations throughout the thesis is also introduced and described in this chapter. Chapter 3 then builds on the concept of *run lengths* and introduces the application of Bayesian methods as a means to analyze run length data. Chapter 3 is the link between the first part of the thesis (Chapter 2) and the rest of the thesis, which focuses on Bayesian methods for controller performance monitoring.

In Chapter 2, we begin by reviewing the use of run length for controller performance and we indicate the advantages of using the *run lengths* approach proposed by Li *et al* (2004). We propose the use of Markov chains for analyzing *run lengths* as a means to contribute to the theoretical background for using *run lengths* in controller monitoring, while retaining the intuitive, computationally and conceptually simple nature of *run lengths*, in our analysis and discussion. By means of examples we also illustrate the use of this non-parametric statistic in controller performance assessment/monitoring.

In Chapter 3, we will address the possibility of using Bayesian networks to show the effect of controller tuning when many possible tuning combinations are available. The use of Bayesian statistics deals with the effect of uncertainty as a result of process, measurement and other sources. Bayesian analysis uses all the available information to make the best possible decision. The Bayesian method proposed serves as a tuning library for decision making purposes. User specified *run lengths* are supplied as evidence and the Bayesian tool determines which controller in the library would most likely satisfy the specifications provided. Examples are provided to illustrate the proposed approach.

Chapter 4 continues the consideration of Bayesian analysis for controller performance assessment and it builds on the previous work addressing MPC constraint analysis under a Bayesian statistics framework (Agarwal 2007). The previous work considered the controlled variables in the economic objective function as the quality variables under consideration. It also used discretization of the profit function into different zones of profit in its problem formulation and analysis. The considerations were also limited to a linear profit function. In this chapter, we propose an approach based on the continuous-valued profit function for the MPC constraint analysis. The consideration is extended to include both linear and quadratic forms of the objective function. This approach allows us to extend the definition of quality variables to include all the variables in the original economic objective function of the MPC controller which have non-zero linear and quadratic coefficients, without being limited to just the controlled variables and avoids the associated loss of information due to the discretization. Two illustrative examples are discussed using the proposed approach.

Chapter 5 follows on from Chapter 4 and considers the effect of dependence between quality variables in the economic objective function of the MPC controller. The presence of dependence between quality variables also affects the amount of potential benefits that can be achieved because when the quality variables are dependent, the mean value of any individual quality variable cannot arbitrarily be moved towards its optimum value without considering the other quality variable(s). This issue of dependence is addressed using Factor analysis within a Bayesian statistics framework.

Two case studies are also provided. Finally in Chapter 6 we provide our conclusions on the analysis carried out.

In Appendix A, we show by way of illustrative examples, the functionalities of the Bayesian Network Toolbox (BNT) as developed by Kevin Murphy (Murphy 2007, Murphy 2004*a*, Murphy 2004*b*, Murphy 2001). The examples are limited to the applications used in this thesis. This provides a reference or tutorial for future users.

This thesis has been presented in a paper-format according to the requirements of the Faculty of Graduate Studies and Research (FGSR), University of Alberta. In order to connect the materials in different chapters and at the same time ensuring completeness and cohesiveness of individual chapters, there is some over lap between chapters and there may be some redundancy of material. The aim, however, is to make the material easily comprehensible to the reader.

# 2

# Performance monitoring of process control using run lengths

## 2.1 Introduction

Process and/or controller performance monitoring (or assessment) algorithms have received considerable interest in recent times and various algorithms and their applications to various sectors of petrochemical, chemical process, pulp and paper and other industries have been reviewed in recent literature (Jelali 2006, Bauer and Craig 2007). The tools and algorithms which were reviewed have been developed (largely independent of each other) by several individuals, research groups and organizations, and they generally have some advantages and disadvantages relative to each other (Jelali 2006). It is noteworthy however, that most of the methods use model-based and/or use parametric statistic and there is room for approaches that are data-based and also use some non-parametric statistics. Such approaches could be very intuitive and computationally simple.

In this work, we investigate the use of methods that may not require process knowledge and that can incorporate non-parametric statistics in controller performance monitoring. Recent literature has laid the foundation and has indicated new possibilities for such approaches (Harris and Yu 2003, Li *et al.* 2004). In this chapter, we will further explore the applications and definitions of *run lengths* and *run length distributions* as proposed by Li *et al* (2004).

### 2.1.1 Advantages of using *run lengths* for controller monitoring

The use of popular approaches like the minimum variance control (MVC) and linear quadratic Gaussian (LQG) benchmarks require some process knowledge. The MVC approach requires knowledge of the process time delay while the LQG approach requires a process model (apart from the challenges involved if the possibility of online applications is considered). In reality, process delays sometimes change during routine operation in some chemical processes (Owusu and Rhinehart 2004) and the true process model is not always known. This indicates the usefulness of an approach that does not depend much on process knowledge. It has also been recommended that a method based on the *distribution* of the controller performance index value and not a single value should be used for controller monitoring because such distribution-based methods are more realistic and representative of controller performance (Li *et al.* 2004).

An approach using the chi-squared goodness-of-fit statistic to compare the distribution of the *run lengths* of current operational data to a reference *run length distribution* as a means of assessing the performance of a controller has been proposed. Since the technique is data based and requires only the routine plant data, it has advantages over the MVC and LQC approaches as it can be used when there is little or no process knowledge. It can also be used online due to its relatively simple computational requirements (Li *et al.* 2004). Since the value of the *run length* is not affected by the magnitude of the error data, it is not affected by the presence of outliers. This makes this approach quite robust. This approach is applicable to both nonlinear and time-varying processes(Li *et al.* 2004).

### 2.1.2 Contributions

We propose the use of Markov chains to analyze process controllers based on their *run length distribution*. This approach maintains the intuitive and computationally simple analysis of data and *run length distributions* but also introduces a new perspective for the analysis of controller behaviour.

A drawback of the proposed approach by Li *et al* (2004) is that there is no strong theoretical/analytical framework behind it. In order to address this issue, the use of Markov chains has been proposed to provide a basis for fundamental analysis (Owusu and Rhinehart 2004). In this work, we also propose a theoretical framework for the use of *run lengths* in controller performance monitoring/assessment by considering the use of Markov chains for the analysis, but from a different perspective from that previously proposed by Owusu *et al* (2004). This approach is based on the work by Harris *et al* (2003) and it provides an intuitive interpretation of *run lengths* as a means of assessing controller performance and at the same time provides similar insight into controller performance as the previous approaches (Li *et al.* 2004, Owusu and Rhinehart 2004).

### 2.1.3 Run length distributions - history and applications

A run is defined as *an uninterrupted sequence* of a particular data value or state (Balakrishnan and Koutras 2002). The use of the term "*run length*" is quite common is certain areas of image and data processing. Run lengths are used as a means for data and image processing and compression, and various applications are available in the literature. Since data or image encodings can be broken down into particular sequences with inherently repeated patterns, they can be grouped as runs and subsequently analyzed (Fatemi 1992, Chandra and Chakrabarty 2001, Galloway 1974, Tsai *et al.* 1996).

Run lengths (particularly the average *run length*, which is defined as the average time taken for a process to fall outside the specified control limits (Wardell *et al.* 1994)) have also been seen extensively in some areas of statistical process control with application to control charts such as the Shewart and CUSUM charts (Robinson and Ho 1978, Brook and Evans 2003, Wardell *et al.* 1994). The *run length* is said to be one of the most important properties of any statistical process control (SPC) chart (Wardell *et al.* 1994). Run lengths have also been used as a test of randomness with applications in production processes via a *Runs test.* They are also useful in detecting non randomness in quantitative measurements over time (i.e. time series) (Wackerly *et al.* 2002).

The use of *run lengths* in engineering applications, however (particularly in controller performance monitoring) has relatively been sparse. Recent work by (Li *et* *al.* 2004) has introduced possibilities for further application of this non-parametric statistic for controller performance monitoring applications beyond the previous scope of conventional statistical process control. Subsequent works (Owusu and Rhinehart 2004, Owusu *et al.* 2005) have built on this and provided some theoretical background to the use of *run lengths* in controller performance via Markov chains.

The basic idea of the definition of *run lengths* to indicate an uninterrupted sequence is common to all the aforementioned areas of application. The particular method of analysis (or interpretation of results) using *run lengths* is generally adapted to the end-user's personal objective. For example, the method of analysis required to use *run lengths* to determine randomness (Wackerly *et al.* 2002) is different from the analysis required when control charts are used to monitor processes (Robinson and Ho 1978, Brook and Evans 2003, Wardell *et al.* 1994). In this work we illustrate the application of *run length distributions* for controller performance assessment as introduced by Li *et al* (2004) and then we will introduce a new application of *run lengths* to controller performance using Markov chains.

## 2.2 Performance monitoring using *run lengths*

### 2.2.1 Overview of controller monitoring using *run lengths*

For a process that is being controlled, the controller works to keep the process at the specified setpoint and therefore, the values of the error signal (the setpoint minus controlled variable/output values) would typically be distributed above and below the mean value. Using the zero value as a reference, we can count the number of data points (above or below the zero line) just before the data changes sign from + to - (and then goes above or below the zero line, or vice versa) as the *run length.* Each time the data changes sign from + to - or vice versa (from - to +), we have a *zero crossing.* A *run length* is the number of data samples between zero crossings (Li *et al.* 2004).

As shown in Chapter 1, Figure 1.2 illustrates the definition of *run lengths* and zero crossings. From Figure 1.2, the *run lengths* are 1, 3, 2, 3, 4, 4, 1, 2, 1, 1. We consider the absolute values of the *run lengths* and the distribution can then be built by summing up and grouping the number of times that *run lengths* 1, 2, 3 and 4 appear. Therefore, *run length* 1 (RL1) will have a value $RL1 = 4$, *run length* 2 (RL2) will have a value $RL2 = 2$, likewise, $RL3 = 2$ and $RL4 = 2$. The *run length distribution* is usually represented in the form of histograms as will be shown subsequently.

In the use of *run length distributions* for controller monitoring, based on the fact that a process variable will typically be distributed above and below a particular mean value, the data can be classified according to run-lengths (the length or number of *runs* of a particular data value or state). The use of *run length* as a control performance index is based on the work by Li *et al* (2004) and this approach involves the creation of a reference based on the prior knowledge that a particular process is in a condition of good control, and subsequently the current data is checked against that reference to determine whether or not there are deviations from the reference as an indication of a change in the performance of the controller. The procedure developed by Li *et al* (2004) is both conceptually and computationally simple and can be applied to nonlinear and time-varying processes as well as for online controller monitoring (Li *et al.* 2004).

### 2.2.2 Multi-tank process: process description and controller design

The Multi-tank System (Inteco 2006) consists of three tanks (upper, middle, and lower tanks) arranged above each other, each equipped with drain valves. A pump fills the upper tank from a storage tank located at the bottom of the third tank as shown in Figure 2.1. Due to the vertical arrangement of the tanks, the liquid outflows all tanks due to gravity. The control objective is to maintain desired liquid levels in each of the tanks. Level sensors are mounted in the tanks to measure these levels. Each tank has a manual and an automatic valve. These valves can be used to stabilize the desired levels. The horizontal cross section of the first tank is constant while those of the second and third tanks are variable (prismatic and spherical, respectively) as shown in Figure 2.1.

The system can be used as a single input-single output (SISO), single input multiple output (SIMO), multiple input single output (MISO), or a multiple input-multiple output (MIMO) system, depending on the choice and/or combination of input and output variables chosen. For our purpose in this work, we consider the system as a $2 \times 2$ MIMO system with the first two tank levels as the outputs (or controlled variables) and the corresponding automated valves are operated as the inputs (or

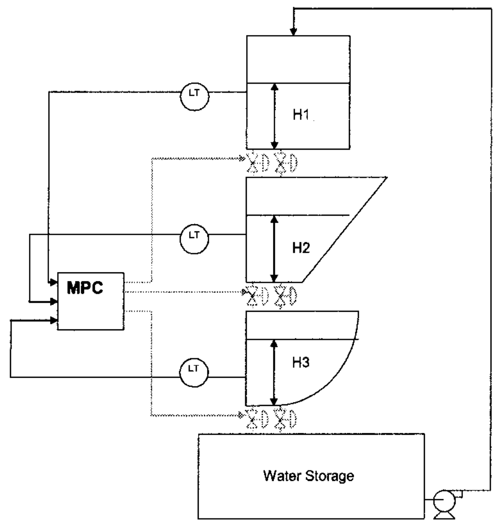



Figure 2.1: Schematic of multi-tank system configuration

manipulated variables). The pump flowrate is used as an unmeasured disturbance.

Real-time experiments are performed using MATLAB and Simulink to develop and run various controllers. For the real-time experiments, MATLAB's Real-Time Workshop (RTW) and Real-Time Windows Target (RTWT) toolboxes (MathWorks 2007*c*) are used for the implementation. This setup can be used to develop and validate various level control strategies as well as implement fault detection strategies (in the case where the manual valves are used to simulate leaks).

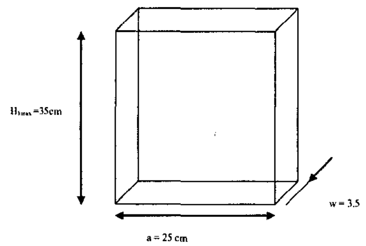
### Mathematical model of the multi-tank system

Based on first principles, the mathematical model of the multi-tank system as follows (Inteco 2006):

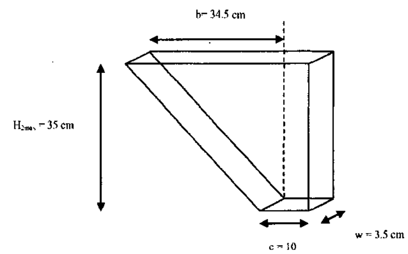
$$\begin{aligned} \frac{dH_1}{dt} &= \frac{1}{A_1(H_1)}q - \frac{1}{A_1(H_1)}C_1H_1^{\alpha_1} \\ \frac{dH_2}{dt} &= \frac{1}{A_2(H_2)}C_1H_1^{\alpha_1} - \frac{1}{A_2(H_2)}C_2H_2^{\alpha_2} \\ \frac{dH_2}{dt} &= \frac{1}{A_3(H_3)}C_2H_2^{\alpha_2} - \frac{1}{A_3(H_3)}C_3H_3^{\alpha_3} \end{aligned} \tag{2.1}$$

where:

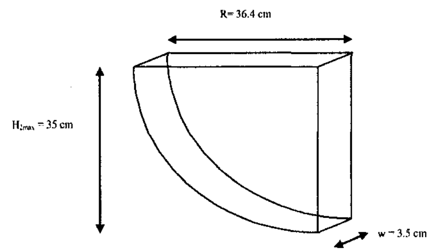
$H_i$ refers to the fluid level in tank i, $i = 1, 2, 3$




(a) tank 1 parameters




(b) tank 2 parameters




(c) tank 3 parameters

Figure 2.2: Schematic diagram showing geometric parameters of the tanks

$A_i(H_i)$ refers to the cross sectional area of tank i at level $H_i$, as follows:

$$\begin{aligned} A_1(H_1) &= aw \\ A_2(H_2) &= cw + \frac{H_2}{H_{2max}}bw \\ A_3(H_3) &= w\sqrt{R^2 - (R - H_3)^2} \end{aligned} \tag{2.2}$$

$C_i$ refers to the valve characteristics of tank i
$\alpha_i$ refers to the flow coefficient of tank i. Generally, we assume $\alpha_i = 0.5$ for laminar flow. This assumption ($\alpha_i = 0.5$) will be used throughout this work. Figure 2.2 shows the geometric parameters of the tanks.

The MATLAB MPC toolbox (Bemporad *et al.* 2007) was used for the controller design. The linearized model used is shown in Equation 2.4 and Figure 2.3 shows the corresponding block diagram arrangement. The parameters used for the MPC simulations are as follows:

- Control interval : 0.4
- Prediction horizon: 30

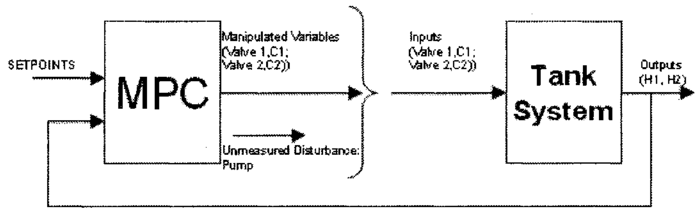



Figure 2.3: Block diagram representing MPC control of multi-tank process

- Control horizon: 3
- Manipulated variables: Controlled valves $(C_1, C_2)$
- Measured outputs: Liquid levels: $(H_1, H_2)$
- Unmeasured disturbance: Pump flowrate $(q_0)$
- Input constraints: Valves 1 and 2: $min = 0, max = 1$ (values given in percentage of valve opening)
- Output constraints: Heights 1 and 2: min=0.02, max= 0.30

For the MPC controller design, we specify the state space model of the process according to Equation 2.3.

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX \end{aligned} \tag{2.3}$$

Y represents the outputs which are the tank levels, in this case. The parameters of

Equation 2.3 are shown in Equation 2.4:

$$
\begin{aligned}
A &= \begin{bmatrix} \frac{-C_1\alpha_1}{awH_1^{1-\alpha_1}} & 0 \\ \frac{-C_1\alpha_1}{w(c+b\frac{H_2}{H_{2max}})H_1^{1-\alpha_1}} & \frac{-C_2\alpha_2}{w(c+b\frac{H_2}{H_{2max}})H_2^{1-\alpha_2}} \end{bmatrix} \\
X &= \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \\
B &= \begin{bmatrix} \frac{1}{aw} & \frac{-H_1^{\alpha_1}}{aw} & 0 \\ 0 & \frac{H_1^{\alpha_1}}{w(c+b\frac{H_2}{H_{2max}})H_1^{1-\alpha_1}} & \frac{H_2^{\alpha_2}}{w(c+b\frac{H_2}{H_{2max}})H_1^{1-\alpha_1}} \end{bmatrix} \\
U &= \begin{bmatrix} q_0 \\ C_1 \\ C_2 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
D &= 0
\end{aligned}
\tag{2.4}
$$

Using the MATLAB "ss" command, the state space model is generated:
$>> TankModel = ss(A, B, C, D)$
The pump inflow rate ($q_0$) is included in the input vector U to make it a measured or unmeasured disturbance as required for the MPC controller. Figure 2.4 shows the input and output data for the process. There is a 25% increase in pump flowrate (unmeasured disturbance) at 200 seconds (500 samples). At 250 seconds (625 samples), we introduce simultaneous step set-point changes of 0.1m to both tanks. Overall, the servo and regulatory performance of the controller is satisfactory.

### 2.2.3 Controller performance monitoring of multi-tank system

The objective here is to implement a data-based and intuitive controller monitor on the multi-tank system, that does not depend on process knowledge and therefore is suitable for monitoring time-varying and non-linear processes (Li *et al.* 2004). From the definition of *run lengths* , we can infer that having large *run length* values implies the possible presence of offsets and therefore, the greater the number of *run lengths* with large values (in the *run length distribution*), the poorer the controller in the steady state performance (Li *et al.* 2004). The *run length distribution* under good

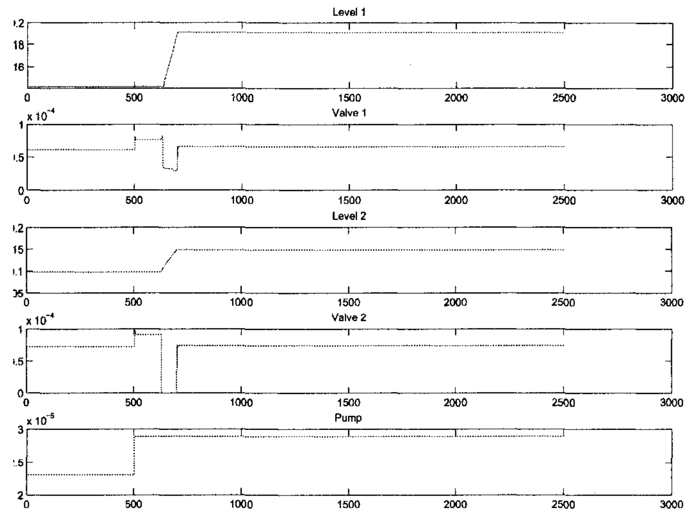



Figure 2.4: Input and output data for process under MPC control

control is not sensitive to changes in magnitude or variance of the errors because we only consider the position of the data relative to a zero crossing and not its actual magnitude.

### 2.2.4 Methodology

The *run length* is used as a control performance index. Its distribution at any point in time is compared to that of a reference, built from data collected during a period of good control, to determine controller performance. The reference distribution is built from the process data according to the procedure outlined below (Li *et al.* 2004):

- **Build the reference *run length distribution***: Using (setpoint (SP) and controlled variable (CV)) data collected during a representative good control period, the error signal is obtained and the *run length distribution* is determined as illustrated in Figure 1.2.
- **Divide the range of *run lengths* of the reference distribution into classes** : This work uses 5 classes to satisfy the requirements for Chi-square

test as shown in Equation 2.5.

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{2.5}$$

Let $O_i$ be the observed number of *run length distributions* in the $i$th class. The expected number of *run lengths* in the $i$th class, $E_i$, is obtained from the reference distribution. The number of classes is represented by $k$. Following the procedure outlined by Li *et al* (2004), we use 5 classes and check the chi-square value against a threshold value of 16.17. If the chi-square value obtained is greater than 16.17, we have a significant difference.

- **Choose a sampling window length $N$**: A sampling window size is chosen for building the current *run length distribution*.
- **Choose a grace period**: This step is optional and allows the monitor to check if the change is transient or actually a substantial change. This feature is useful for reducing false alarm rates.
- **Build the *run length distribution* for current data**: For monitoring purposes, a window of data is collected and the *run length distribution* is built as described above and checked against the reference. A moving window of data is used to check the entire data set against the reference.
- **Compare Distribution with reference**: If the current data is significantly different by Chi-square test, the Monitor Flags '1' to indicate the difference and flags '0' otherwise. If a grace period is included, a violation counter begins to increase until the threshold is exceeded and the monitor flags '1' when the violation counter exceeds the grace period, and '0' otherwise (if the difference does not persist beyond the grace period). The procedure for choosing the grace period is outlined by Li *et al* (2004).

### 2.2.5 Simulation results

In all of the results shown in this Section there is no grace period included in the algorithm, so the monitor flags whenever there is a significant difference between the reference data and the data under observation. This could lead to lots of false alarms

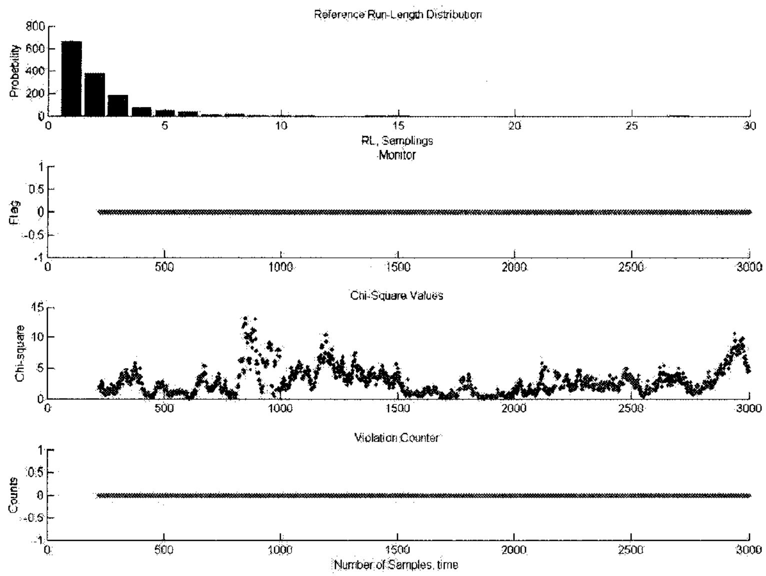



Figure 2.5: Results obtained with no process change

under practical conditions but we have not included the grace period here because the aim is to show the activity of the monitor and to illustrate its usefulness. When the change in controller performance is transient (e.g small step changes), the grace period becomes useful because the rate of false alarms will decrease significantly.

As shown in Figure 2.5, when there is no disturbance or change to the process, the monitor flags '0' throughout the simulation indicating that there is no significant change in the current data compared to the reference data. Figure 2.6 indicates that the monitor identifies the step change at about 200 seconds (500 samples) and indicates the controllers ability to correct the disturbance by flagging "0" after the effect of the step change has been overcome.

Since each tank has a manual and a controlled/automated valve, we can simulate a leak by opening the manual valve. As shown in Figure 2.7, after the leak was introduced at 300 seconds (750 samples), the process never returned to its original state. The monitor clearly indicates this change in the process and flags '1' from 300s (750 samples) onwards, as shown in Figure 2.8. In Figure 2.9 we illustrate that the approach can also be readily applied to non-linear processes by applying the approach to the second tank. The monitor also identifies the change in the controller

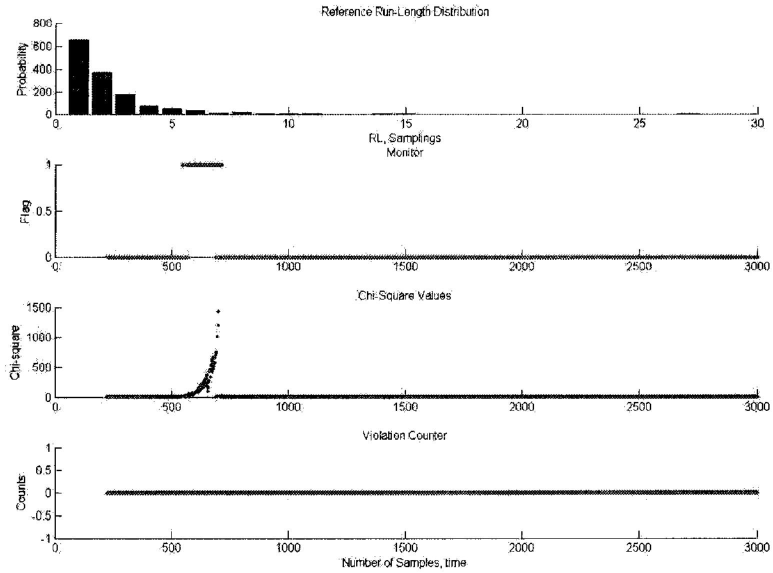



Figure 2.6: Results showing effect of step on level of tank 1

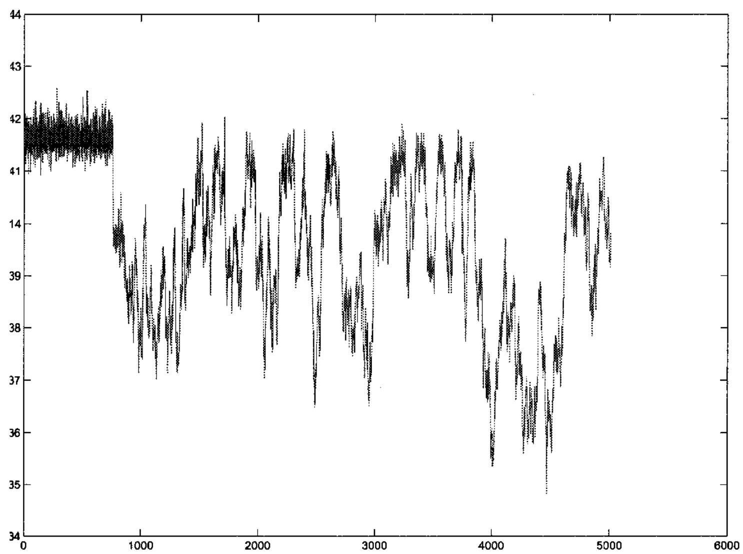


Figure 2.7: Tank 1 data showing the effect of a leak

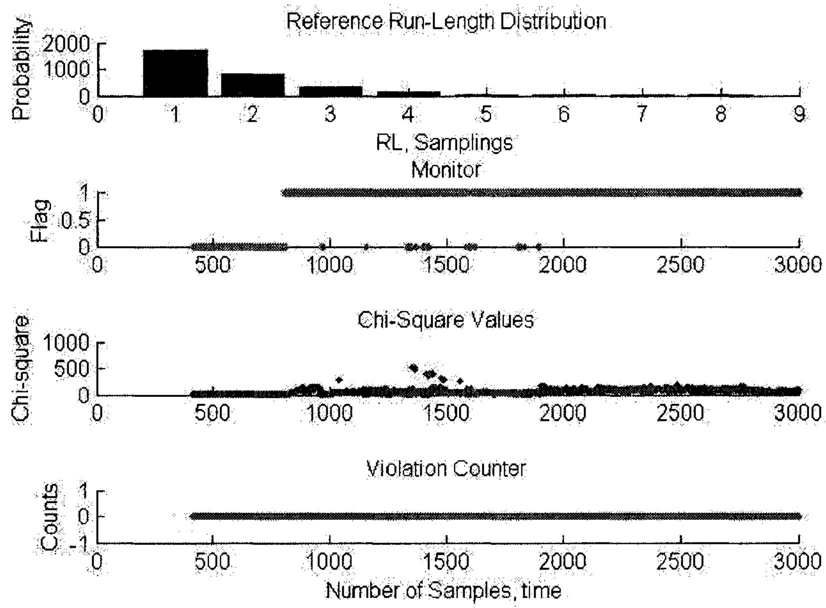



Figure 2.8: Results showing the effect of a leak on controller performance

performance due to the step change and ceases to flag the change when the effect of the step change has been overcome.

### 2.2.6 System identification and real-time experiments

The mathematical model shown above was not satisfactory as a model for the MPC experiments. A new model was identified using the MATLAB System Identification toolbox (Ljung 2007). The input signal was chosen to be a random binary sequence (RBS) so that sufficient persistent excitation could be achieved. The process operating conditions for the experiments were chosen such that overflow or underflow conditions were not encountered, as this will introduce errors in the identified models. The experimental conditions during the system identification are similar to those of the normal process operation such that a realistic model, which is truly representative of the process can be obtained. The upper and lower limits of the input signals based on these considerations are then used in the design of the RBS signal. There is a trade-off between the input signal-to-noise ratio (SNR) and nonlinearity in the system. Finally, the input signal was designed such that it has its energy in the frequency band of this

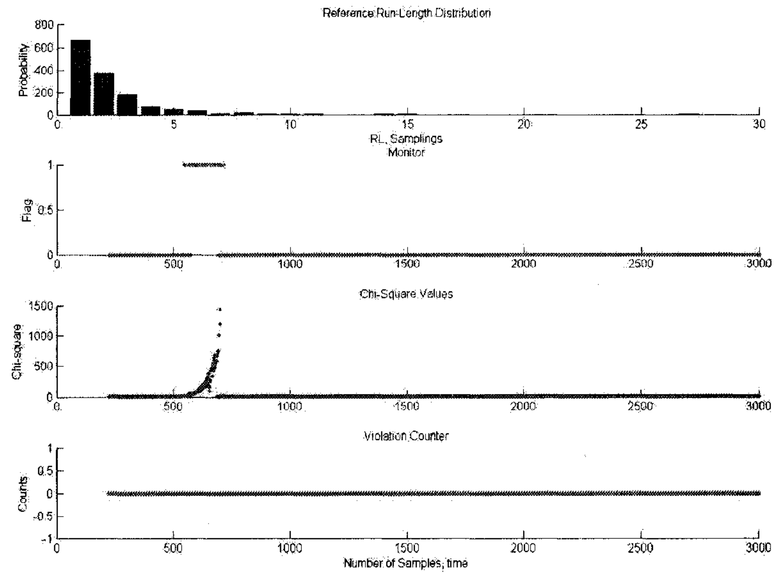



Figure 2.9: Results showing effect of step on level of tank 2

system. The following operating conditions were chosen:
Pump inflow: 34%: (7.0353e-5 $m^3$/s)
Valve 1: 75% open
Valve 2: 75% open
Valve 3: 75% open
There is a linear relationship for the valves between 50% and fully open (100%) and so, 75% opening was chosen as the operating point to give allowance for movement in both directions. The experimental setup is operated at its nominal operating point and this allows us use a linear model for our analysis. The final model obtained is an output error (OE) model which is shown in Equation 2.6 below. This model is subsequently used in the design of the MPC.

$$\begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix} = \begin{pmatrix} \frac{-0.04967}{z-0.8904} & 0 & 0 \\ \frac{0.03994z-0.04013}{z^2-1.764z+0.7789} & \frac{-0.04411z+0.04201}{z^2-1.841z+0.847} & 0 \\ \frac{0.0139z-0.01395}{z^3-1.897z^2+0.9023z} & \frac{0.03782z-0.03758}{z^2-1.778z+0.79} & \frac{-0.02095z+0.0211}{z^2-1.891z+0.8908} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} \quad (2.6)$$

the sampling time is 8 seconds.

The process input and output constraints are set as $0.5 \leq C_i \leq 1.0$, and $0.05 \leq H_i \leq 0.25$, where $C_i$ represent the valve opening, $H_i$ represents the level in the tanks,

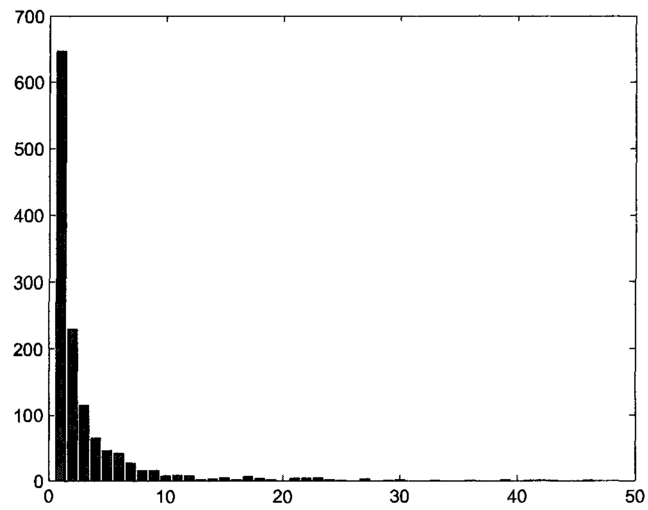



Figure 2.10: Reference *run length distribution* for Tank 2

and $i = 1, 2, 3$. The input rate weights are given as $[0.5, 0.5, 0.5]$, and the output weights are $[1, 0.8, 0.9]$. The control interval is 8 seconds, and the prediction and control horizons are as 15 and 2, respectively. The MATLAB MPC toolbox (Bemporad *et al.* 2007) was used to design the controller. For the following illustration, a $2 \times 2$ system consisting of the first and second tanks is considered and the data for the second tank will be used to show that the approach can readily be applied to an experimental nonlinear process. Figure 2.10 shows the reference *run length distribution* against which the process will subsequently be checked for deviations. Figure 2.11 shows the results obtained by the monitor when there are no changes to the process operation.

Figure 2.12 shows the input and output data for the second tank and we observe that there are step setpoint changes at 3000s and 6000s. The monitor does not flag poor performance due to setpoint change because the error signal is based on the setpoint value and as shown in Figure 2.12, the setpoint tracking of the controller is good. However, towards the end of the experiment, at approximately 8000s, we observe that the control valve reaches it's lower limit (and remains there) while trying to maintain the tank level at its set point, and consequently, the performance of the controller decreases. As shown in Figure 2.13, the monitor flags the poor performance towards the end of the experiment, as desired. These illustrations from both simu-

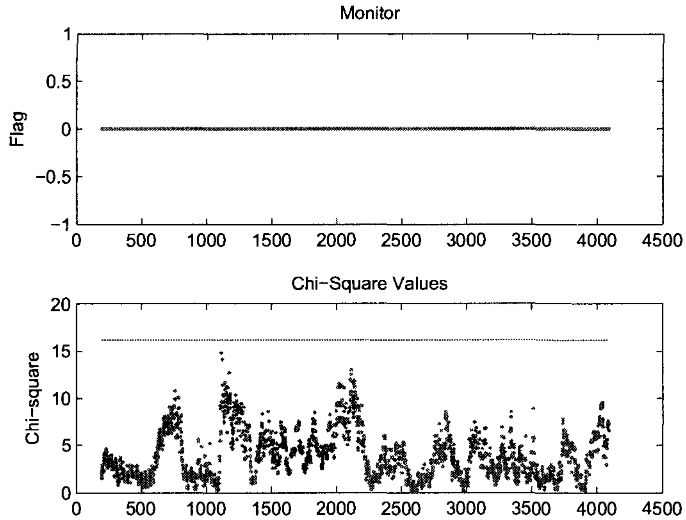



Figure 2.11: Monitor results for Tank 2 with no process change

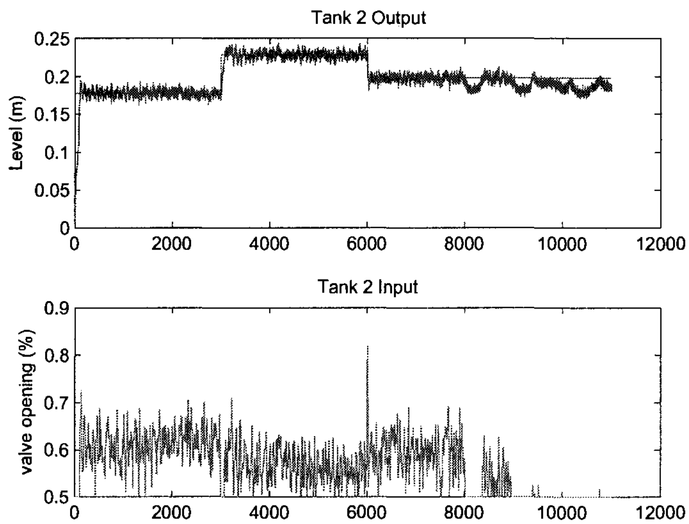



Figure 2.12: Input and output data for Tank 2 with step inputs

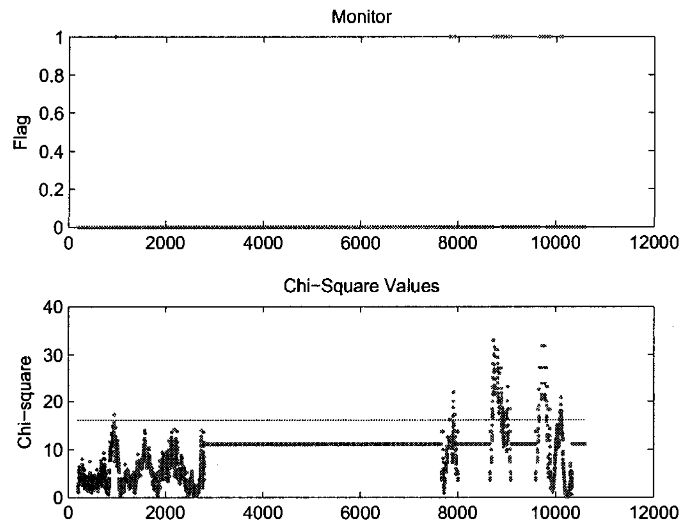



Figure 2.13: Controller monitoring for Tank 2 indicating change in controller perfomance

lation and experimental studies indicate that the algorithm is effective for controller monitoring.

## 2.3 Analysis of Run-Length Data using Markov Chains

The work by Li *et al* (2004) provided a useful framework for using *run length distributions* for controller performance assessment but it did not provide a suitable theoretical background for the proposed procedure. Owusu *et al* (2004) proposed the use of Markov chains to model the *run length distributions*, so as to provide a tractable theoretical basis for using *run lengths* for performance monitoring.

In this work, we explore a different approach of using Markov chains for modeling *run lengths* with the aim being to retain the intuitive interpretation of *run lengths* with respect to controller monitoring as well as working within a framework that has a sound theoretical basis.

In this section, the analysis of *run length distributions* using Markov chains and the interpretations with regards to controller performance are investigated. The output data from an MPC controlled process is considered in this section. The error data (set point minus controlled variable) is used to obtain the *run length distribution* (RLD) and this in turn, is used to obtain the transition probability matrix for the *run lengths* (Lu 2007). In this section, we propose an application of Markov chains to controller performance monitoring. Some applications of Markov chains to monitor controlled processes can be found in recent literature (Harris and Yu 2003, Lu 2007, Owusu and Rhinehart 2004, Owusu *et al.* 2005).

### 2.3.1 Markov chains- background

A Markov process is a random process from which the current state of the process completely determines the state of the process at the next time instant. The Markov property implies that the past does not influence the future behaviour/state of the process. A Markov chain is a Markov process with a finite or countable number of states (Kemeny and Snell 1976).

Assuming that we have a state space, $S$, which contains $q$ possible outcome, the $i$th element of $S$ is described as $s_i$, $i = 1, 2, \ldots, q$. Also, let the sequence of $n$ observations of the Markov Process be $X_t, t = 0, 1, 2, \ldots, n$. The transition probability matrix, P contains the probabilities of the process moving from one state to another in one step. The $p_{ij}$ element of the P matrix indicates the probability of the process to move from state $i$ to state $j$ in one step. $p_{ij} \geq 0$ for all $i$ and $j$ and each row of P adds up to one.

The states of a Markov chain can be classified into transient and ergodic sets. A transient set is a set which once left, are never again re-entered. For an ergodic set, however, once it is entered, it can never be left. An absorbing state is a state which is the only element of an ergodic set. For this state, $s_i$, the probability, $p_{ii}$ must be 1 and all other probability values in that row of the P matrix are 0. If a Markov chain has all non-transient states as absorbing states, it is called an absorbing chain (Kemeny and Snell 1976, Lu 2007). An ergodic Markov chain has no transient sets, and has a single ergodic set. For a homogeneous Markov chain, the presence of only one eigen value of the P matrix with magnitude of unity, is a necessary and sufficient condition for a Markov Chain to be ergodic (Seneta 1973, Harris and Yu 2003).

- **Equilibrium distribution**: For a regular transition matrix P, $P^n$ approaches a probability matrix A as n tends to infinity. This matrix A, has a unique characteristic in that all the rows are the same probability vector $\alpha = a_1, a_2, ..., a_n$.

The matrix A is called the limiting matrix and $\alpha$ is called the equilibrium or stationary distribution.

For a given transition matrix P, the equilibrium distribution is unique as shown in Equation 2.7.

$$\alpha P = \alpha \text{ and } a_1 + a_2 + \cdots + a_n = 1 \tag{2.7}$$

This equilibrium distribution is not affected by the initial distribution $\pi_0$

- **Mean first passage time**: If an ergodic Markov chain is started in state $s_i$, the expected number of steps to reach state $s_j$ for the first time is called the mean first passage time from $s_i$ to $s_j$ . It is denoted by $m_{ij}$ . By convention $m_{ii}$ = 0. If an ergodic Markov chain is started in state $s_i$, the expected number of steps to return to $s_i$ for the first time is the mean recurrence time for $s_i$. It is denoted by $r_i$.

- **Absorption probabilities**: Let $b_{ij}$ be the probability that an absorbing chain will be absorbed in the absorbing state $s_j$ if it starts in the transient state $s_i$. Let B be the matrix with entries $b_{ij}$. Then B is an t-by-r matrix, and $B = NR$, Where N is the fundamental matrix and R is in the canonical form.

- **Time to absorption**: Let $t_i$ be the expected number of steps before the chain is absorbed, given that the chain starts in state $s_i$, and let t be the column vector whose $i$th entry is $t_i$. Then

$$t = Nc \tag{2.8}$$

Where c is a column vector all of whose entries are 1.

- **State holding time**: This is the average time interval the sequence spends in state $s_i$ before it transits to another state.

- **Occupational time**: This is the number of times that the sequence is found in a particular state $s_i$.

At any time, t, the probability of the process being in any of the q states is described by the vector $\pi_t$. $\pi_t = Probability(X_t = S_i)$. With the condition that the sum of its elements equals one.

$$\Sigma_{i=1}^{q}(\pi_t)_i = 1 \tag{2.9}$$

$\pi_0$ is called the initial distribution.

For this analysis, the absolute values of the *run lengths* are considered as the individual states. That is, a *run length distribution* with maximum *run length n* will have $n$ states and the transition probability matrix will be an *n by n* matrix. The *run length distribution* can be interpreted by considering low *run lengths* as indicating a tendency towards good control and high *run length* values as indicating a tendency towards poor control (or the presence of offsets and sluggish behaviour), therefore a *run length* value 1 can be seen as a tendency towards the best control condition and *run length* value n as the tendency towards worst control condition. With the transition probability matrix (P), the mean first passage time matrix and the passage details are calculated (similar applications of Markov chains can be found in the work by Sien Lu (Lu 2007)). The mean first passage time matrix is denoted as M. Each entry $m_{ij}$ is the mean first passage time from state $s_i$ to $s_j$. The mean passage matrix M is given by

$$M = (I - Z + EZ_{dg})D \tag{2.10}$$

Where I is an identity matrix, $Z = (I - (P - A))^{-1}$ is called the fundamental matrix for a regular Markov chain, $Z_{dg}$ denotes the diagonal elements of the fundamental matrix, E is a matrix with all entries 1, D is the diagonal matrix with diagonal elements $d_{ii} = 1/a_i$ and A is called the limiting matrix. Each row of A is the same probability vector $\alpha = a_1, a_2, \ldots, a_n$, where $\alpha$ is the limiting vector or equilibrium distribution.

More specific information can be obtained by changing the process from a regular Markov process to an absorbing process. This tells us the average number of times that the process will be in each of the other states before reaching $s_j$ for the first time. To do this, we need to change state $s_j$ to an absorbing state. We can then obtain information which would ordinarily not be observed from the regular Markov chain. By this transformation, the behaviours of the original process and the newly created absorbing process remain the same as before the absorbing states were reached. The two extreme states: state 1 (symbolic of the best control) and state n, (symbolic of the poorest control) were made to be absorbing states. To create an absorbing chain matrix, the transition probability Matrix, P is rearranged as shown below.

$$P = \begin{bmatrix} S & 0 \\ R & Q \end{bmatrix} \tag{2.11}$$

Where the sub-matrix S is the transition probability matrix of all absorbing states and Q is that of all transient states. The fundamental matrix of an absorbing Markov chain is denoted as N and the elements of N, $n_{ij}$ , is the number of times that the process is in transient state $j$ after it leaves the initial state $i$ and before it reaches an absorbing state. N is calculated as:

$$N = (I - Q)^{-1} \tag{2.12}$$

## 2.3.2 Illustrative example - simulations

The above described procedures were applied to simulated data and the results are shown and discussed below.

Table 2.1: Transition probability matrix

| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.497 | 0.26 | 0.122 | 0.06 | 0.039 | 0.013 | 0.006 | 9E-04 | 0 | 9E-04 |
| **2** | 0.445 | 0.254 | 0.136 | 0.083 | 0.045 | 0.017 | 0.005 | 0.008 | 0.002 | 0.006 |
| **3** | 0.409 | 0.308 | 0.157 | 0.088 | 0.019 | 0.013 | 0.003 | 0 | 0.003 | 0 |
| **4** | 0.415 | 0.295 | 0.125 | 0.08 | 0.045 | 0.011 | 0.017 | 0.011 | 0 | 0 |
| **5** | 0.515 | 0.186 | 0.155 | 0.082 | 0.052 | 0.01 | 0 | 0 | 0 | 0 |
| **6** | 0.429 | 0.286 | 0.143 | 0.057 | 0.057 | 0 | 0 | 0.029 | 0 | 0 |
| **7** | 0.143 | 0.5 | 0.143 | 0.071 | 0 | 0.071 | 0 | 0.071 | 0 | 0 |
| **8** | 0.5 | 0.2 | 0 | 0.1 | 0.2 | 0 | 0 | 0 | 0 | 0 |
| **9** | 0.5 | 0 | 0 | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0.2 | 0 | 0.2 | 0.2 | 0.2 | 0.2 | 0 | 0 | 0 | 0 |

An eigen value analysis of transition matrix P shown in Table 2.1 shows that there is only one eigen value with magnitude unity, as shown in Table 2.2, and this implies that the transition matrix is ergodic.

The values of the equilibrium distribution and the *run length* probabilities are basically the same (see Table 2.3). This can be understood because the *run length*

Table 2.2: Eigenvalues of transition matrix

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.041 + .0254i | 0.041 - .0254i | 0.026 +0.050i | 0.026 - 0.050i | -0.016 +0.06i | -0.016 -0.06i | -0.05 | 0.009 | -0.02 |

Table 2.3: Equilibrium distribution and *run length distribution*

| Equil Dist | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.463 | 0.265 | 0.132 | 0.073 | 0.04 | 0.014 | 0.006 | 0.004 | 8E-04 | 0.002 |
| RL prob | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| | 0.463 | 0.265 | 0.132 | 0.073 | 0.04 | 0.014 | 0.006 | 0.004 | 8E-04 | 0.002 |

probability is the probability based on the overall distribution of the data (as the number of data tends to infinity). This piece of information could prove to be a useful link between Markov chains and Run length distributions. The data in the first

Table 2.4: Matrix of mean first passage times

| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 2.161 | 3.741 | 7.875 | 14.02 | 25.18 | 68.05 | 171.3 | 241.1 | 1207 | 481.9 |
| **2** | 2.284 | 3.769 | 7.767 | 13.68 | 24.99 | 67.75 | 171.6 | 239.4 | 1205 | 479.3 |
| **3** | 2.369 | 3.55 | 7.597 | 13.59 | 25.71 | 68.12 | 171.8 | 241.3 | 1203 | 482.2 |
| **4** | 2.355 | 3.601 | 7.852 | 13.73 | 24.99 | 68.17 | 169.5 | 238.3 | 1207 | 482.2 |
| **5** | 2.117 | 4.019 | 7.621 | 13.72 | 24.91 | 68.33 | 172.4 | 241.6 | 1207 | 482.5 |
| **6** | 2.307 | 3.661 | 7.73 | 14.03 | 24.63 | 69.03 | 172.5 | 234.6 | 1207 | 482.3 |
| **7** | 2.964 | 2.848 | 7.75 | 13.75 | 25.81 | 64.13 | 172.6 | 223.4 | 1207 | 481.7 |
| **8** | 2.116 | 4.034 | 8.8 | 13.49 | 21.09 | 69.06 | 172.4 | 241.6 | 1208 | 482.5 |
| **9** | 2.178 | 4.671 | 8.863 | 8.012 | 26.08 | 69.11 | 171.4 | 240.7 | 1208 | 483 |
| **10** | 2.83 | 4.714 | 7.215 | 12.07 | 21.1 | 55.53 | 172.5 | 240.4 | 1207 | 483.2 |

column in Table 2.4 gives the average time the process takes to return to the good control state $s_1$. For example, if the process starts from $s_{10}$ (poorest control), it would take about 2.83 time instants to get to state $s_1$. We also notice that $M_{ij} \quad (i < j)$ is much larger than $M_{ji}$. For instance, it will take about 481.9 time instants for the process to reach $s_{10}$ from $s_1$; however, just 2.83 time instants from $s_{10}$ to $s_1$. This trend can also be observed when lower and larger states are considered. This means that the process has fewer tendencies to stay at the poorer states and more tendencies to stay in the good control state. This confirms our previous observation that the process is in a state of satisfactory control.

From Tables 2.5 and 2.6, we can get more information concerning how much time

Table 2.5: Passage details with state 1 as an absorbing state

| **State** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|
| **2** | 1.599 | 0.315 | 0.192 | 0.1 | 0.037 | 0.012 | 0.017 | 0.003 | 0.01 |
| **3** | 0.679 | 1.348 | 0.204 | 0.074 | 0.033 | 0.011 | 0.009 | 0.005 | 0.004 |
| **4** | 0.662 | 0.313 | 1.193 | 0.102 | 0.032 | 0.024 | 0.021 | 0.002 | 0.004 |
| **5** | 0.488 | 0.312 | 0.176 | 1.096 | 0.027 | 0.006 | 0.007 | 0.002 | 0.003 |
| **6** | 0.633 | 0.323 | 0.168 | 0.115 | 1.019 | 0.007 | 0.036 | 0.002 | 0.004 |
| **7** | 1.023 | 0.407 | 0.236 | 0.094 | 0.1 | 1.01 | 0.086 | 0.003 | 0.006 |
| **8** | 0.484 | 0.157 | 0.193 | 0.249 | 0.016 | 0.006 | 1.007 | 0.001 | 0.003 |
| **9** | 0.331 | 0.157 | 0.597 | 0.051 | 0.016 | 0.012 | 0.011 | 1.001 | 0.002 |
| **10** | 0.493 | 0.459 | 0.348 | 0.277 | 0.222 | 0.01 | 0.015 | 0.002 | 1.003 |

Table 2.6: Passage details with state 10 as an absorbing state

| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 224.3 | 128.1 | 63.34 | 34.96 | 19.18 | 6.799 | 2.799 | 1.991 | 0.399 |
| **2** | 222.1 | 128.4 | 63.02 | 34.8 | 19.09 | 6.767 | 2.783 | 1.988 | 0.399 |
| **3** | 223.3 | 128.2 | 64.41 | 35.01 | 19.17 | 6.803 | 2.798 | 1.992 | 0.403 |
| **4** | 223.4 | 128.2 | 63.39 | 36.01 | 19.2 | 6.803 | 2.812 | 2.004 | 0.399 |
| **5** | 223.6 | 128.2 | 63.45 | 35.03 | 20.22 | 6.805 | 2.797 | 1.992 | 0.4 |
| **6** | 223.4 | 128.2 | 63.41 | 34.99 | 19.22 | 7.791 | 2.795 | 2.02 | 0.399 |
| **7** | 222.9 | 128.3 | 63.34 | 34.97 | 19.15 | 6.854 | 3.791 | 2.064 | 0.399 |
| **8** | 223.6 | 128.2 | 63.3 | 35.05 | 19.37 | 6.794 | 2.797 | 2.992 | 0.399 |
| **9** | 223.8 | 128.2 | 63.36 | 35.48 | 19.19 | 6.801 | 2.805 | 1.998 | 1.399 |

was spent in other states between, for example, moving from state $s_1$ to state $s_{10}$ and vice versa, by considering states 1 and 10 as absorbing states, respectively. For instance, while in Table 2.4, the tenth row, first column tells us that it takes 2.83 time instants for the process to move from state 10 to state 1, Table 2.5 gives us a more detailed explanation by showing that the process spends 0.493 time instants in state 2, 0.459 time instants in state 3, 0.348 time instants in state 4, and so on. The same detailed explanation can be made for the tenth item on the first row of Table 2.4 by considering the first row of Table 2.6. Similar explanations can also be made when other states are considered as absorbing chains.

Let $t_i$ be the expected number of steps before the chain is absorbed, given that the chain starts in state $s_i$, and let t be the column vector whose $i$th entry is $t_i$. The vector t can be found using Equation 2.8 (Kemeny and Snell 1976). Comparing $t_1$ and $t_{10}$ (see Table 2.7), it is obvious that regardless of what state the process starts from,

Table 2.7: Times to absorbtion with states 1 and 10 as absorbing states

| **State** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|
| $t_1$ | 2.284 | 2.369 | 2.355 | 2.117 | 2.307 | 2.964 | 2.1157 | 2.1776 | 2.83 |
| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| $t_{10}$ | 481.9 | 479.3 | 482.2 | 482.2 | 482.5 | 482.3 | 481.734 | 482.513 | 483 |

it has a higher tendency to be absorbed (to end up in state 1 than state 10). The same explanation could be extended to say that the process will more likely end up in lower states than in higher states, which implies that the process is in a condition of good control.

### 2.3.3 Illustrative example - experiments

In this section, we illustrate the proposed approach by analyzing experimental data based on the multi-tank system previously discussed. The level in the second tank will be considered for this illustration, to show that the approach can readily be applied to experimental data from a nonlinear process.

The controller previously used in section 2.2.6 is also used in this analysis. For simplicity, the *run lengths* have been limited to a maximum of 10, as shown in Figure 2.14, to contain the size of the matrices that result and for ease of discussion. However, for practical purposes, it is not compulsory to impose an upper limit on the maximum number of run lengths and the analysis that follows still applies.

Table 2.8: Transition probability matrix for experimental case

| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.452 | 0.193 | 0.111 | 0.077 | 0.048 | 0.042 | 0.033 | 0.017 | 0.019 | 0.008 |
| **2** | 0.578 | 0.196 | 0.087 | 0.035 | 0.026 | 0.035 | 0.013 | 0.013 | 0.013 | 0.004 |
| **3** | 0.617 | 0.183 | 0.078 | 0.017 | 0.026 | 0.052 | 0.017 | 0 | 0.009 | 0 |
| **4** | 0.615 | 0.246 | 0.062 | 0.015 | 0.031 | 0.015 | 0 | 0 | 0 | 0.015 |
| **5** | 0.674 | 0.174 | 0.065 | 0.043 | 0.043 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0.643 | 0.119 | 0.119 | 0.048 | 0.024 | 0 | 0.024 | 0.024 | 0 | 0 |
| **7** | 0.815 | 0.111 | 0.037 | 0 | 0.037 | 0 | 0 | 0 | 0 | 0 |
| **8** | 0.813 | 0.125 | 0 | 0 | 0 | 0 | 0 | 0.063 | 0 | 0 |
| **9** | 0.688 | 0.250 | 0.063 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0.857 | 0.143 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

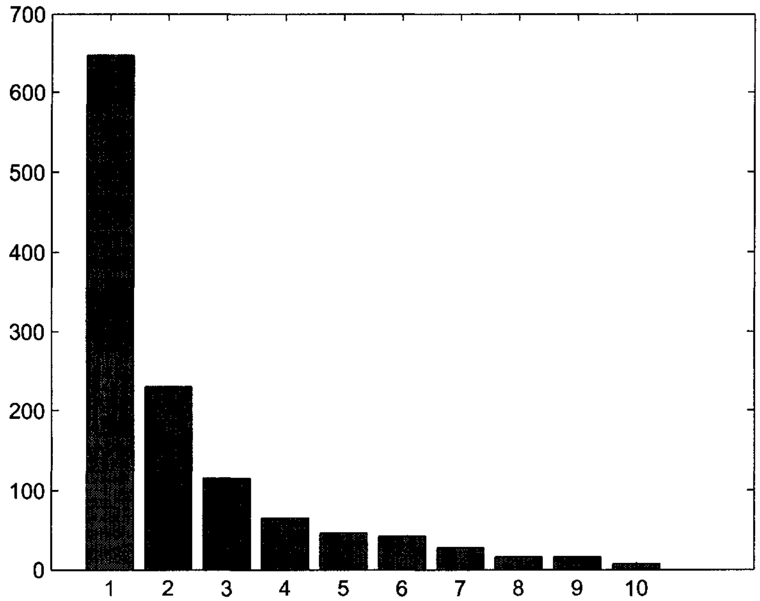



Figure 2.14: *Run length* distribution for Markov-based analysis - experimental case

Table 2.9: Eigenvalues of transition matrix - experimental case

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | -0.21 | 0.06 | 0.03+ 0.03i | 0.03- 0.03i | -0.05 | -0.03 | -0.02 | 0.01- 0.01i | 0.01- 0.01i |

The transition probability matrix for the *run length* distribution used in this analysis is shown in Table 2.8. Also, from the eigen values of this transition matrix (as shown in Table 2.9) we observe that there is only one eigen value with a magnitude equal to unity. This implies that it is ergodic in nature. In Table 2.10, we again show that the run length distribution and the equilibrium (or stationary) distribution of the Markov chain are essentially the same.

Considering Table 2.11 at a glance, we observe that the values at lower states are

Table 2.10: Equilibrium and *run length* distributions for the experimental case

| **Equil Dist** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.53 | 0.19 | 0.10 | 0.05 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 | 0.01 |
| **RL Prob** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| | 0.53 | 0.19 | 0.09 | 0.05 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 | 0.01 |

Table 2.11: Matrix of mean first passage times for experimental case

| State | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1.87 | 5.27 | 10.26 | 17.62 | 26.31 | 27.74 | 43.67 | 79.58 | 74.52 | 171.78 |
| **2** | 1.68 | 5.26 | 10.46 | 18.30 | 26.84 | 27.86 | 44.41 | 79.76 | 74.80 | 172.32 |
| **3** | 1.62 | 5.34 | 10.52 | 18.58 | 26.80 | 27.35 | 44.16 | 80.75 | 75.10 | 173.08 |
| **4** | 1.63 | 4.99 | 10.71 | 18.62 | 26.69 | 28.37 | 44.96 | 80.75 | 75.71 | 170.38 |
| **5** | 1.54 | 5.35 | 10.66 | 18.06 | 26.30 | 28.80 | 44.94 | 80.79 | 75.71 | 172.95 |
| **6** | 1.57 | 5.66 | 10.14 | 18.04 | 26.85 | 28.81 | 43.89 | 78.95 | 75.76 | 173.02 |
| **7** | 1.30 | 5.69 | 10.92 | 18.75 | 26.41 | 28.77 | 44.81 | 80.68 | 75.61 | 172.93 |
| **8** | 1.29 | 5.64 | 11.35 | 18.78 | 27.44 | 28.82 | 44.83 | 75.62 | 75.62 | 172.92 |
| **9** | 1.52 | 4.96 | 10.67 | 18.85 | 27.47 | 28.74 | 44.88 | 80.69 | 75.62 | 173.00 |
| **10** | 1.24 | 5.52 | 11.29 | 18.72 | 27.38 | 28.75 | 44.77 | 80.60 | 75.56 | 172.86 |

much smaller than the values at larger states for this controller. This observation indicates that the controller tends to "operate" at lower *run lengths* than at higher *run lengths.* From our previous discussions based on simulations, we can also conclude here that the controller performance is satisfactory because there is less probability for sluggish behaviour or the occurrence of offsets.

Tables 2.12 and 2.13 show the details of the passage "times" when the process moves from state 1 to state 10 (when state 1 is considered as an absorbing state), and when the process moves from state 10 to state 1 (when state 10 is considered as an absorbing state), respectively. For example, considering the first column in Table 2.11, we observe that if the process is in state 10 (poorer control), it will take 1.24 time instants for it to go to state 1 (better control). Whereas, if the process is in state 1, it will take 171.78 time instants for it to go to state 10. This implies that the process has greater probability to go to state 1 (and lower states similarly), than to state 10 (generally higher states).

The details of the "journey" from state 1 to 10 (i.e how much "time" the process spends in all other states between 1 and 10, when state 1 is an absorbing state) are found in the *tenth* row of Table 2.12. This row sums up to the *first* item in the *tenth* row of Table 2.11. Likewise, the details of the "time" spent in intermediate states between 10 and 1 when state 10 is an absorbing state are found in the *first* row of Table 2.13. The sum of the elements of this row is equal to the *tenth* item of the first row in Table 2.11. This procedure of analyzing the "time" spent in intermediate states can be applied to all the states to obtain more information of the behaviour of

Table 2.12: Passage details with state 1 as an absorbing state - experimental case

| **State** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|
| **2** | 1.322 | 0.140 | 0.054 | 0.044 | 0.054 | 0.021 | 0.020 | 0.018 | 0.007 |
| **3** | 0.296 | 1.129 | 0.036 | 0.043 | 0.070 | 0.025 | 0.006 | 0.014 | 0.002 |
| **4** | 0.364 | 0.112 | 1.034 | 0.047 | 0.034 | 0.008 | 0.006 | 0.006 | 0.017 |
| **5** | 0.277 | 0.108 | 0.059 | 1.058 | 0.016 | 0.006 | 0.004 | 0.005 | 0.002 |
| **6** | 0.225 | 0.161 | 0.062 | 0.039 | 1.017 | 0.030 | 0.029 | 0.004 | 0.002 |
| **7** | 0.168 | 0.061 | 0.009 | 0.046 | 0.009 | 1.003 | 0.003 | 0.003 | 0.001 |
| **8** | 0.176 | 0.019 | 0.007 | 0.006 | 0.007 | 0.003 | 1.069 | 0.002 | 0.001 |
| **9** | 0.349 | 0.106 | 0.016 | 0.014 | 0.018 | 0.007 | 0.005 | 1.005 | 0.002 |
| **10** | 0.189 | 0.020 | 0.008 | 0.006 | 0.008 | 0.003 | 0.003 | 0.003 | 1.001 |

Table 2.13: Passage details with state 10 as an absorbing state - experimental case

| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 92.372 | 32.699 | 16.424 | 9.287 | 6.571 | 5.998 | 3.858 | 2.285 | 2.285 |
| **2** | 91.766 | 33.805 | 16.456 | 9.279 | 6.572 | 6.013 | 3.853 | 2.290 | 2.289 |
| **3** | 92.203 | 32.935 | 17.523 | 9.305 | 6.602 | 6.057 | 3.876 | 2.287 | 2.295 |
| **4** | 90.758 | 32.488 | 16.248 | 10.159 | 6.504 | 5.927 | 3.798 | 2.251 | 2.251 |
| **5** | 92.177 | 32.907 | 16.497 | 9.326 | 7.616 | 6.001 | 3.855 | 2.284 | 2.285 |
| **6** | 92.195 | 32.861 | 16.553 | 9.331 | 6.598 | 7.004 | 3.880 | 2.310 | 2.285 |
| **7** | 92.291 | 32.838 | 16.471 | 9.288 | 6.611 | 6.002 | 4.858 | 2.286 | 2.286 |
| **8** | 92.291 | 32.847 | 16.428 | 9.286 | 6.571 | 6.000 | 3.857 | 3.352 | 2.286 |
| **9** | 92.210 | 32.990 | 16.501 | 9.286 | 6.573 | 6.005 | 3.858 | 2.286 | 3.287 |

the controller via the *run lengths* or states.

Finally, in Table 2.14, we observe that irrespective of what state the process starts in, there is a much more higher probability that it will end up in state 1 than in state 10, in much shorter time.

## 2.4 Conclusion

In the above examples we have illustrated the use of non-parametric statistics in controller performance assessment/monitoring. The use of *run length distributions* has been shown as a tool for dynamic performance monitoring. Markov chains have been used to model *run length distributions* with an aim to investigate the intuitive interpretation of *run lengths* within a framework that has good theoretical basis. Two illustrative examples have been provided for each of the algorithms discussed.

Table 2.14: Time to absorbtion with states 1 and 10 as absorbing states - experimental case

| **State** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|
| **t1** | 1.68 | 1.62 | 1.63 | 1.54 | 1.57 | 1.30 | 1.29 | 1.52 | 1.24 |
| **State** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **t10** | 171.8 | 172.3 | 173.1 | 170.4 | 172.9 | 173.0 | 172.9 | 172.9 | 173.0 |

# 3

# Bayesian analysis using run length distributions

## 3.1 Introduction

The use of the term *logic*, both in mathematics and in common speech, is based on assumed clear notions of truth and falsity. Information that can be categorized as either true or false is known as Boolean logic. For example, a statement such as *"Unless I turn the lights on, the room will be dark"*, leaves no room for uncertainty. In contrast to this, events that are encountered in the activities of daily life present us with many situations in which we come to conclusions based on the *accumulation* of evidence. For example, if you notice a small amount of coffee on the table after a meal, there may be no great cause for concern if you remember that you recently poured yourself a fresh cup of coffee, and spilled some on the table while doing so. In this case, the coffee spill evidence may be unimportant; otherwise (if you were absolutely sure you did not spill any coffee), you may need to have your coffee pot checked for a crack or a leak, or wait to monitor the situation a bit more (by waiting to see if another coffee stain shows up on the table when you place the coffee pot in a different place). Situations such as those described above are very common in day to day life and the need therefore arises to have a means of dealing with uncertainty in situations where Boolean logic fails (MSBN 2007).

Bayesian probability theory is a branch of mathematical probability theory that allows us to model the uncertainties associated with outcomes of interest that we find in the world by combining common-sense knowledge and available observational evidence (MSBN 2007).

A belief network is described by:

- A set of relevant variables
- A graphical structure connecting the variables, and
- A set of conditional probability distributions

### 3.1.1 Types of variables

A continuous variable is a variable for which, within the limits of the variable ranges, any value is possible. For example, the variable *time to solve a puzzle* is continuous because it could take 3 minutes, 4.24 minutes etc. to complete.

A discrete variable is a variable which has a limited number of values; each of the set of possible values is called a state. It cannot take on all values within the limits of the variable. For example, responses to a four-point rating scale can only take on the values 1, 2, 3, and 4. The variable cannot have the value 2.3. (MSBN 2007).

### 3.1.2 Relationships within a Bayesian network

One major importance of a Bayesian model is that it allows the user to include prior knowledge in the model (available common sense and real-world knowledge), and thereby producing a simpler model. To build on the previous example, the model builder's prior knowledge that the time of day (for example) would usually not directly influence the occurrence of a coffee stain on a table will eliminate *time of day* as a variable and allows him to focus on more direct factors, such as the presence of other people at the table, etc. Essentially, variables which cause changes in the system are associated with those that they influence. These influences are represented by *connecting arcs* between nodes. Each arc represents a causal relationship between a preceding variable (known as the *parent*) and the variable whose outcome is influenced by it (known as the *child*).

A Bayesian belief network is commonly described with the use of a graph, having a set of vertices and edges. The vertices, or nodes, represent the variables and the edges or arcs, represent the conditional dependencies in the model. The absence of an arc between two variables implies that they are conditionally independent; that is, under no situations will the probabilities of one of the variables be affected directly by (or affect) the state of the other variable. In this work, we use directed acyclic graphs (DAG). Simply speaking, a DAG "flows" in a single direction. $a \rightarrow b$ implies that $a$ affects $b$ but $b$ does not affect $a$ directly. A source in a DAG is a vertex with no incoming edges, while a sink is a vertex with no outgoing edges. A finite DAG has at least one source and at least one sink (MSBN 2007, Korb and Nicholson 2004).

A belief network is built as follows:

- Include all important variables that model the system
- Connect the variables based on (causal) knowledge of the system
- Use prior knowledge to specify the conditional distributions

A simple definition of causal knowledge in this context means linking variables in the model in such a way that arcs lead from causes to effects (MSBN 2007, Murphy 2001).

### 3.1.3 Inference

Inference, also known as model evaluation, is the process by which probabilities of outcomes are updated based on the relationships in the model and the currently available evidence. During the use of a Bayesian model, the end user supplies evidence based on recent events or observations and this information is then applied to the model via the appropriate variable. Based on a mathematical algorithm, the probabilities of all other variables that are connected to the variable representing the new evidence are also updated. The new level of belief ( or probabilities) of possible outcomes are now included in the model. The beliefs initially supplied to the model before any evidence is available about the situation are known as *prior* probabilities. The beliefs computed after new evidence is supplied to the model are known as *posterior* probabilities.

In the subsequent use of Bayesian belief networks (or Bayesian Networks), each continuous variable is represented by an ellipse while a discrete variable is represented by a rectangle. These graphical representations are called nodes. Causal influence is indicated by a line (or an arc) linking the influencing node to the influenced node. The influence arc has an arrowhead that points to and terminates at the influenced variable (MSBN 2007).

The Bayesian analysis is based on the probability distribution of the data. For this work, the data is assumed to be Gaussian. The Bayesian analysis is carried out using the Bayesian Network toolbox (BNT) by Kevin Murphy (Murphy 2004*b*, Murphy 2004*a*, Murphy 2001).

The Gaussian distribution is shown as follows:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(z-\overline{z})^2}{2\sigma^2}\right) \tag{3.1}$$

where z is the data, $\overline{z}$ is the mean and $\sigma$ is the standard deviation. If $p(z)$ is plotted against z, we obtain the probability density function. The integral of $p(z)$ in the range $-\infty$ to z gives the cumulative distribution function. The Cumulative distribution function is mathematically be represented as follows:

$$cdf(z) = \frac{1}{2}\left(1 + erf\left(\frac{z-\overline{z}}{\sigma\sqrt{2}}\right)\right) \tag{3.2}$$

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z exp(-t^2)dt \tag{3.3}$$

## 3.2 Analyzing Run length distributions using Bayesian networks

Since the *run length distribution* shows the behaviour of a particular of a process with a particular controller, we can use Bayesian networks to model the relationship between different control tunings and their *run length distribution*, based on the fact that the *run length distribution* of a process will change as the controller changes. The *run length distribution* can be used as a means of intuitively interpreting the control performance, as has been discussed previously. For a process with different PID control tunings, a Bayesian network can be built with the P, I and D components of the controller as the parent nodes and the *run length distributions*, grouped into classes, as the child nodes. This allows us to infer the change in control performance as the controller parameters are changed, within a probabilistic framework. Since there is always uncertainty associated with a process due to process and measurement noise, the usefulness of Bayesian statistics become readily apparent.

In addition, the Bayesian tool may also serve as a tuning library. The library is built initially by simulating different controller tunings, with each tuning having a corresponding *run length distribution.* A user specified *run length distribution* can then be supplied to the library to determine a corresponding controller tuning. The tool becomes very useful when the desired distribution is not the same as that in the library. In this case, the tool will determine the most probable explanation of that tuning, i.e., the most likely tuning (of all the available tunings) that will produce such a *run length distribution.* This is similar to interpolation to the nearest element in the library.

### 3.2.1 Illustrative example

Consider a simple closed loop example (Seborg *et al.* 2003) for which the responses to unit step disturbances of a FOPTD (first order plus time delay) process under PI control are shown in Figure 3.1. The process model is shown below:

$$G(s) = \frac{e^{-4s}}{20s+1} \tag{3.4}$$

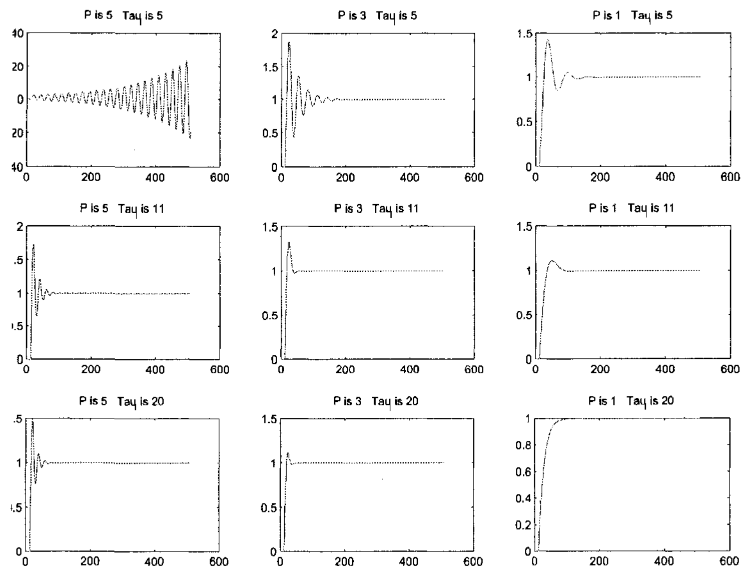



Figure 3.1: Unit step responses of a PI controller with different tunings

The effects of the increase of the controller gain $K_c$ (P component) and the decrease of the integral time constant (I component) are shown in Figure 3.1. As the value of $K_c$ increases, or $\tau_I$ decreases, we have more aggressive responses from the controller. Generally, conservative controller tunings require small controller gains ($K_c$) and large values of integral time ($\tau_I$) (Seborg *et al.* 2003). For this example, we have three values each for the P and I components. This results in $3^2$ (nine) possible combinations of controller tunings (or nine controllers) with each set of tunings producing a different response to the unit disturbance (see Figure 3.1). In the presence of process and measurement noise, each controller output will not be exactly the same even though the underlying behaviour remains still the same. This presence of noise introduces uncertainty into the system and therefore provides an opportunity for Bayesian statistics to be applied. Using the *run length distribution* will accommodate the underlying behaviour of the process while the Bayesian network will handle the problem of making inference in the presence of uncertainty.

For a process with different PI control tunings, a Bayesian network can be built with the P and I components of the controller as the parent nodes and the *run length*

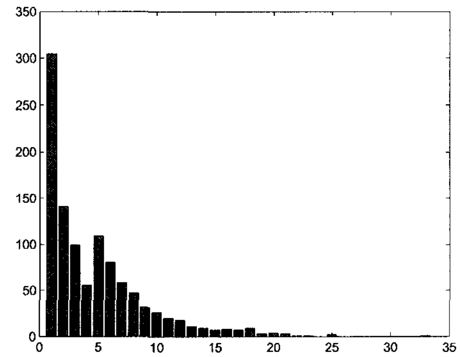
Table 3.1: Tuning values of Proportional and Integral components of the PI controller

| | **Proportional Gain** | **Integral time** |
|---|---|---|
| **State 1** | 5 | 5 |
| **State 2** | 3 | 11 |
| **State 3** | 1 | 20 |

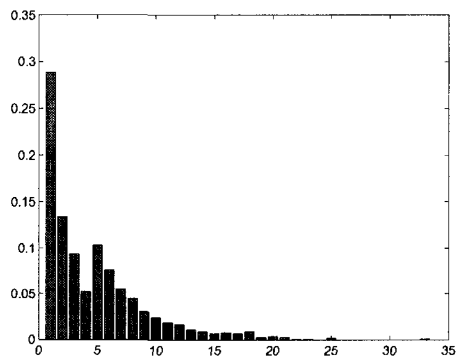
*distributions*, grouped into classes, as the child nodes. The states of the parent nodes are the parameters of the P and I components of the controller. These parents are discrete while the child nodes are assumed to be continuous and Gaussian in nature. In this example, we have three states each for the P and I parent nodes (see Table 3.1) and the child nodes (the *run lengths*) are grouped into five classes following the procedure previously discussed (Li *et al.* 2004). However, since we cannot assume that the group of 5 classes will be reasonably similar for all the controllers, without providing a standardized reference, we require a benchmark for grouping the run lengths into classes. To achieve this, we use the minimum variance control (MVC) method to design a controller for the process and obtain its corresponding *run length distribution*, and then we use the class groupings for this MVC controller as a reference for grouping the *run length distributions* of the controllers under subsequent consideration. This is done for two main reasons: first, we want to group the data such that each group has probabilities of occurrence that are as even as possible, so that subsequent comparison against a reference class of data, and the use of Chi-square statistics as a measure of goodness-of-fit is made reasonable (Li *et al.* 2004). Second, we want to provide a framework for reproducibility of the results rather than grouping the data arbitrarily, in the event that further investigation is necessary. The design of the minimum variance controller will be shown subsequently.

There is no formal rule for grouping the data into classes. A generally accepted rule of thumb is that each class should have at a minimum, one expected *run length* observation and 80% of all the classes should have at least 5 expected *run length* observations (Li *et al.* 2004). We will use Figure 3.2(b) and Table 3.2 to illustrate how the probabilities of the *run lengths* are grouped into classes.

Run length 1 (RL1) has a probability of 0.288152 which is greater than 1/5 (i.e.,




(a) Run length distribution




(b) Run length probabilities

Figure 3.2: Plots showing *run lengths* of FOPTD process under minimum variance control

Table 3.2: Run length distributions and class groupings

| **Run-Length** | **Probability** |
|---|---|
| 1 | 0.2881517 |
| 2 | 0.1336493 |
| 3 | 0.0938389 |
| 4 | 0.0521327 |
| 5 | 0.1033175 |
| 6 | 0.0758294 |
| 7 | 0.0549763 |
| 8 | 0.0445498 |
| 9 | 0.0303318 |
| 10,... | 0.0246,... |

| **Class** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **RL** | **1** | **2,3** | **4,5,6** | **7,8,9** | **10, 11,...** |
| **Prob** | 0.2881517 | 0.22748815 | 0.2312796 | 0.129858 | 0.123223 |

the even value of probabilities among 5 classes) therefore, RL1 forms class 1. Next, we check the value of RL2 against the remaining probability for the $(1 - 0.288152)/4 = 0.1780$. RL2 is less than 0.1780 but the sum of the probabilities of RL2 and RL3 are greater than 0.1780 therefore, RL2 and RL3 form class 2. Similarly, because RL4 has a value less than $(1 - 0.228152 - 0.133649 - 0.093839)/3 = 0.1815$ but the sum of the probabilities of RL4, RL5 and RL6 are greater than 0.1815, RL4, RL5 and RL6 form class 3. Following the above procedure, we form the 5 classes. Subsequently, we group the *run lengths* for each controller according to the classes in the reference distribution as shown above. Class 1 will consist of RL1, class 2 will consist of RL2 and 3, and so on (see table 3.2).

### 3.2.2 Minimum variance controller for FOPTD process

The use of minimum variance control as a benchmark for controller performance assessment can be traced to the work by Harris (1989). In this Section, we show the procedure for designing a controller using the MVC control law as outlined in Huang and Shah (1999).

In the following formulations we will omit the back-shift operator $q^{-1}$ and thus we will denote a transfer function, for example, $T(q^{-1})$ as $T$. Consider a SISO process under regulatory control (as shown in Figure 3.3) with $d$ as the time delay, $\tilde{T}$ as the plant transfer function which is delay-free, $N$ is the disturbance transfer function, $e_t$ is a white noise sequence with zero mean, and $Q$ is the controller transfer function. It can be seen from Figure 3.3 that

$$y_t = Tu_t + Na_t \tag{3.5}$$

$$y_t = \frac{N}{1 + q^{-d}\tilde{T}Q} a_t \tag{3.6}$$

using the Diophantine identity:

$$N = f_0 + f_1 q^{-1} + ... + f_{d-1} q^{-d+1} + Rq^{-d} \tag{3.7}$$

$$\text{where} \quad F = f_0 + f_1 q^{-1} + ... + f_{d-1} q^{-d+1}$$

where $f_i$(for $i$=1,...,$d$-$i$) are constant coefficients and $R$ is the remaining rational, proper transfer function. Equation 3.6 can be written as

$$\begin{aligned} y_t &= \frac{F + q^{-d}R}{1 + q^{-d}\tilde{T}Q} a_t \\ &= [F + \frac{R - F\tilde{T}Q}{1 + q^{-d}\tilde{T}Q} q^{-d}] a_t \\ &= F a_t + L a_{t-d} \end{aligned} \tag{3.8}$$

$$\textit{where} \quad L = \frac{R - F\tilde{T}Q}{1 + q^{-d}\tilde{T}Q}$$

is a proper transfer function. The two terms on the right hand side of Equation 3.8 are independent since $Fa_t = f_0 + f_1 q^{-1} + ... + f_{d-1} q^{-d+1}$. Thus,

$$Var(y_t) = Var(Fa_t) + Var(La_{t-d})$$

Therefore

$$Var(y_t) \geq Var(Fa_t)$$

The equality holds when $L = 0$, *i.e.*,

$$R - F\tilde{T}Q = 0$$

Thus we obtain the minimum variance control law:

$$Q = \frac{R}{\tilde{T}F} \tag{3.9}$$

The term $Fa_t$ (the process output under minimum variance control) is feedback controller-invariant because F is independent of the controller transfer function $Q$. Consequently, if a stable process output $y_t$ is modelled by an infinite moving-average model, then its first $d$ terms constitute an estimate of the minimum variance term $Fa_t$ (Huang and Shah 1999).

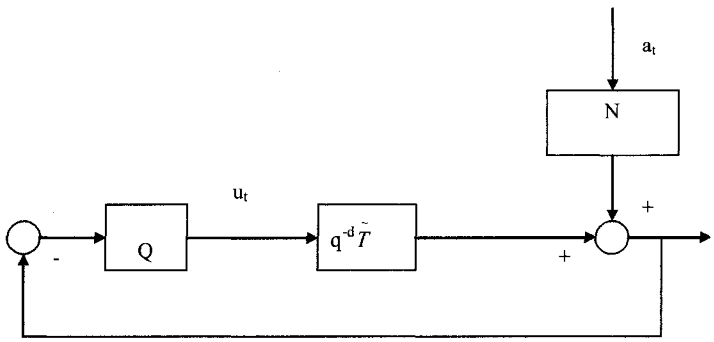



Figure 3.3: Schematic diagram of a SISO process under feedback control

Considering the process model described in Equation 3.4, we can design the minimum variance controller. First of all, we obtain the discrete time equivalent of the process (Seborg *et al.* 2003, Ogunnaike and Ray 1994) using a sampling time of $\delta t = 1s$.

$$\begin{aligned} G(s) &= \frac{Ke^{-\theta s}}{\tau s + 1} \\ G(q^{-1}) &= \frac{K(1-A)q^{-N-1}}{1-Aq^{-1}} \\ where \quad A &= e^{\frac{-\delta t}{\tau}} \\ and \quad N &= \frac{\theta}{\delta t} \end{aligned} \tag{3.10}$$

using Equation 3.10,

$$\begin{aligned} G(s) &= \frac{e^{-4s}}{20s+1} \\ G(q^{-1}) &= \frac{0.04877q^{-5}}{1-0.9512q^{-1}} \end{aligned} \tag{3.11}$$

from Equation 3.10, we have the following:
$\tilde{T} = \frac{0.04877}{1-0.9512q^{-1}}$
assuming a disturbance model, $N = \frac{1}{1-q^{-1}}$, $d = 5$

$N = 1 + q^{-1} + q^{-2} + q^{-3} + q^{-4} + \frac{q^{-5}}{1-q^{-1}}$

thus $F = 1 + q^{-1} + q^{-2} + q^{-3} + q^{-4}$

and $R = \frac{q^{-5}}{1-q^{-1}}$

Using Equation 3.9, we obtain the MVC controller Q as:

$$Q = \frac{1 - 0.9512q^{-1}}{1 + q^{-1} + q^{-2} + q^{-3} + q^{-4}} \frac{1}{0.04877(1 - q^{-1})} \tag{3.12}$$

Finally, using Equation 3.6, we can verify that the process is under minimum variance control with the controller Q.

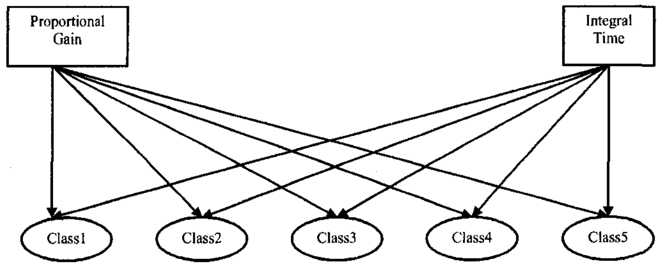
$$y_t = (1 + q^{-1} + q^{-2} + q^{-3} + q^{-4})a_t \tag{3.13}$$

Figure 3.2 shows the *run length distribution* of the FOPTD process under minimum variance control.

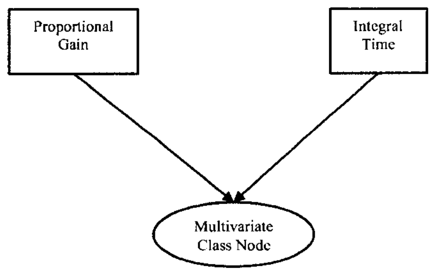
### 3.2.3 Bayesian analysis of *run length distributions*: results

With the information provided in Section 3, 3.2 and 3.2.2, we can proceed to build a Bayesian network following the outlined procedures and subsequently make inferences using the network. With the process model as shown in Equation 3.4, and the PI tuning parameters as shown in Table 3.1. The proportional (P) and integral (I) parts of the controller are considered to be the parent nodes. Each parent node is discrete with three states (see Table 3.1) while the child nodes are assumed to be continuous. In the network, we can have five univariate child nodes or one multivariate child node with a size equal to five (i.e., a node with five variables). The Bayesian network is shown in Figure 3.4.

In order to build the network, data is generated and the network is *trained* in order to obtain the conditional probability distributions for each node. For each controller (or each tuning combination), we generate data by simulating the process 500 times with a different noise seed each time. The *run length distributions* are obtained from the simulation data and the distribution obtained for each simulation is then grouped into 5 classes as previously discussed. Since each of the parent nodes has three states, we have $3^2 = 9$ tuning combinations and therefore we have $9 \times 500 = 4500$ simulations.




(a) Univariate child node




(b) Multivariate child node

Figure 3.4: Bayesian network for PI controller showing use of univariate and multivariate nodes

The network is trained with the data and the conditional probability distributions are estimated using the expectation maximization algorithm (Murphy 2007, Murphy 2001).

Figure 3.5 shows the data plots for the entire experiment. The average values of the classes of each simulated controller was used as demonstrated in Figure 3.6. Class1 to Class5 refers to the *run length* data in each of the class groupings for the 4500 simulations, RefC1 to RefC5 in the figure refers to the values of Class1 to Class5 when the process was under minimum variance control. As we observed previously in Figure 3.1, the performance of the controller is worst when P is in state 1 ($K_c = 5$) and I is in state $1(\tau_I = 5)$. As $K_c$ decreases and $\tau_I$ increases, we observe that the performance of the controller improves. We can make inference as to which of the nine controllers is most likely to produce *run length distributions* with class groupings similar to those of the minimum variance controller by supplying the class groupings of the minimum variance controller to the network as evidence. Based on the supplied

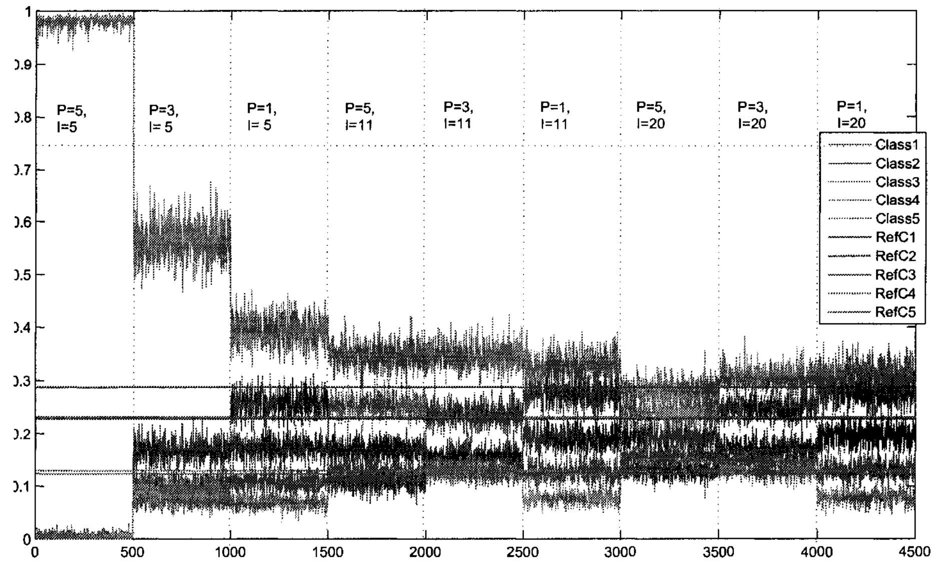



Figure 3.5: Plot showing change in *run length* classes with controller tuning

evidence, we then determine the most probable explanation for the evidence based on the combination of the states of the parent nodes.

- **Example 1**

  Using the evidence (class groupings data for the MVC controlled process), we can obtain as inference from the network. The evidence for the child nodes is supplied as shown in Table 3.3:

Table 3.3: Evidence supplied to network

| P state | I state | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|---|---|---|---|---|---|---|
| | | 0.28815 | 0.22749 | 0.23128 | 0.12986 | 0.12322 |

After the inference, we have the following result (Table 3.4):

This indicates that the controller with P-node being in state 2 and the I-node in state 3 (i.e $P = 3$ and $I = 20$, see Table 3.1 for details), will most likely

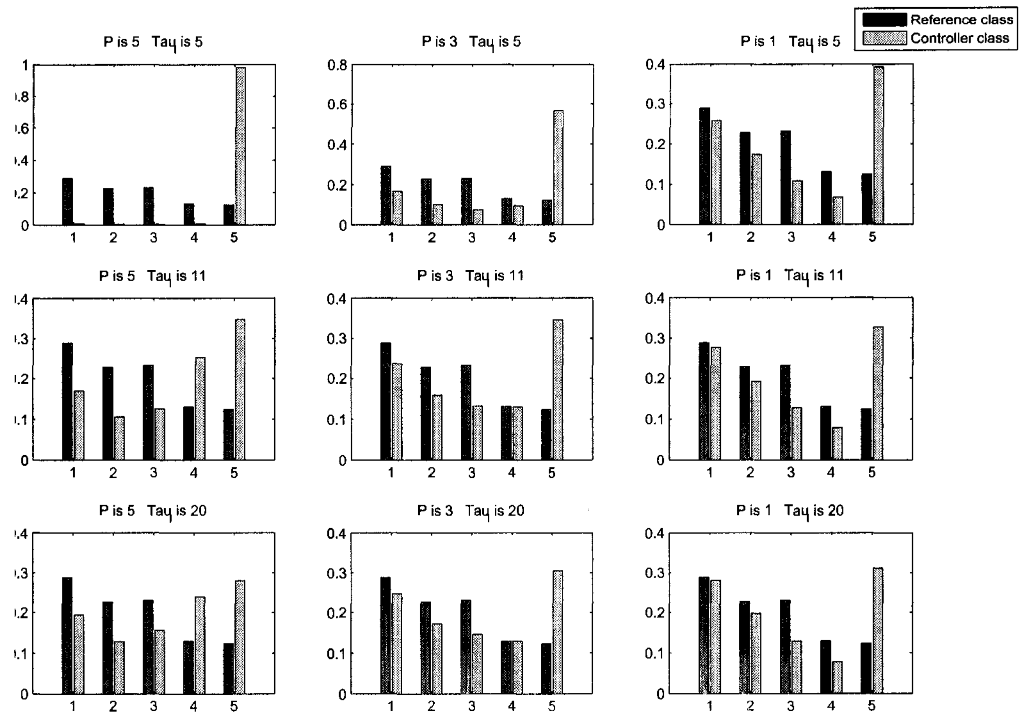



Figure 3.6: Bar plots showing change in *run length* classes with controller tuning

produce a *run length distribution* with class groupings similar to those of the process under minimum variance control.

- **Example 2**

  With a user specified *run length distribution* as shown in Table 3.5, we can investigate the choice of tunings that will most likely satisfy this requirement. From Table 3.2, we can interpret the evidence supplied in Table 3.5. The evidence supplied implies that we would like a controller that does not have too much random behaviour. With the assumption that a very high value of RL1 could imply that an actuator could wear out easily (opening and closing continuously) we desire a *run length distribution* which is not too random/erratic, hence the value of Class 1 (which is made up of the *run length* with value of 1 (RL1) only) is chosen 0.120. At the same time, we do not desire a controller with much offset so we choose a value for class 5 to be 0.125, this is approximately the

Table 3.4: Inference based on evidence supplied to network

| P state | I state | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|---|---|---|---|---|---|---|
| 2 | 3 | 0.28815 | 0.22749 | 0.23128 | 0.12986 | 0.12322 |

same as in the reference class (see Table 3.2). The values of Classes 2, 3 and 4 are chosen likewise as shown in Table 3.5. We ensure that the probabilities of all classes add up to 1.

Table 3.5: User specified evidence supplied to network

| P state | I state | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|---|---|---|---|---|---|---|
| | | 0.120 | 0.229 | 0.409 | 0.117 | 0.125 |

This decision made from the library is as shown in Table 3.6. Comparing the controller chosen (P in state 1 and I in state 3) with the controllers shown in Figure 3.1, we observe a reasonable agreement. The controller chosen ($P = 5$ and $I = 20$) shows some oscillatory behavior when compared to the controller chosen in example 1 ($P = 3$ and $I = 20$). If the output plot for this controller ($P = 5, I = 20$) shown in Figure 3.1 was described by *run lengths*, we would observe that the presence of oscillations would imply less *run lengths* of low values but higher *run lengths* of intermediate values (since the oscillations are rather transient). Also, no sustained offsets are present. This result provided by the Bayesian tool corresponds to the specifications we outlined in the user specified target *run length distribution*.

Table 3.6: Inference based on user specified evidence supplied to network

| P state | I state | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|---|---|---|---|---|---|---|
| 1 | 3 | 0.120 | 0.229 | 0.409 | 0.117 | 0.125 |

In the two examples provided, we have shown that the tuning library based on a Bayesian framework can be utilized to determine controller tunings that agree with user specified *run length* objectives.

## 3.3 Conclusion

In this chapter, we have proposed an approach for using a Bayesian network to describe the changes in controller tunings when many possible tuning combinations are available. The use of Bayesian statistics deals with the effect of uncertainty as a result of process, measurement and other sources of noise and the Bayesian tool serves as a tuning library for decision making purposes. Examples have been provided to illustrate the proposed approach.

Although we have used the *run length distribution* as the index for our analysis, a similar approach could be applied to auto-covariances of processes (and possibly to any process that has a repeatable pattern) and similar reference could be made to MVC as a benchmark (or any other appropriate algorithm).

# 4

# MPC constraint analysis - Bayesian approach via a continuous-valued profit function[1]

## 4.1 Introduction

The current academic and industrial interest in performance monitoring and assessment of industrial process controllers is motivated by the use of control systems to achieve goals related to quality, safety and asset utilization(Huang and Shah 1999, Bauer and Craig 2007, Jelali 2006, Harris and Yu 2003). As a technology for asset-management, performance assessment helps industrial automation systems in maintaining high efficiency in production operations. Certain unexpected events can lead to disruptions in output and/or quality of processes and eventually, the process suffers in terms of loss of quality control of products, reduced machine efficiency and ultimately, increased costs. A common example is the malfunctioning of sensors and/or actuators in an industrial process. It becomes necessary to quickly detect

[1]A version of this chapter has been submitted for publication in Akande, S and Huang, B., *MPC constraint analysis - Bayesian approach via a continuous-valued profit function*, Industrial & Engineering Chemistry Research, 2008

and correct such malfunctions and reduce associated variability. Consequently, it is beneficial to provide an automated procedure that would enable plant operators to effectively monitor and evaluate industrial processes and thereby increase profit margins. This is the main objective of controller performance monitoring/assessment(Jelali 2006). A 0.1% improvement in an oilsands extraction/upgrading process, for example, could result in savings of millions of dollars per year (given a yearly output of about $10^8$ barrels) in a typical oil industry.

The use of advanced process control (APC) strategies in industries has become crucial to achieving economic performance objectives. Among them, model predictive control (MPC) stands out as a key strategy for dealing with multivariable systems. "MPC refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behaviour of a plant"(Qin and Badgwell 1997). MPC is a multivariable model-based controller and therefore it inherently considers the interaction between the process variables. One major advantage of MPC is its ability to handle input and output constraints. Tuning of an MPC is highly dependent on process knowledge and the particular control philosophy chosen for the process(Qin and Badgwell 1997) . Most approaches to tuning MPCs involve one or more of the following: 1) the penalty matrix, 2) the prediction horizon and 3) the control horizon. All these parameters are important to the performance of the MPC, but the constraint limits and the variability of the process variables also play critical roles in controller performance. In general, however, there is no specific procedure or framework for choosing or tuning MPC constraints. Most approaches could be said to be largely subjective(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*).

In an MPC controller, the controlled variables (CVs) are often said to indicate the desired product qualities which are to be considered for optimization and the manipulated variables (MVs) are generally used to control and optimize the process within a constraint set. CV and MV constraints are often set conservatively in practice(Singh and Seto 2002); therefore, the probability that adjusting constraints could provide more degrees of freedom for the controller and improving profit margins by doing so, typically exists(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*).

A method to estimate the effect of the constraints on the performance of an MPC will provide more insight and thus reduce conservativeness when setting up the constraints during commissioning or otherwise. In trying to achieve optimal MPC economic performance, the following questions arise: which constraints should be changed, and if constraint change is indeed possible, what profits can be obtained by doing so? These questions are answered by studying the relationship between variability and constraints within a Bayesian probabilistic framework. The details are described as follows(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*):

**Decision evaluation:** To infer the expected return (or profit) that could be obtained, if certain decisions regarding constraint limits change are made.

**Decision making:** To obtain the combination of constraint changes of relevant variables, based on the maximum-a-posteriori explanation, that will most likely achieve a user-specified target profit.

This work considers the case where the expected return is obtained from a continuous-valued function. To make Bayesian inference and for the *decision making/evaluation*, the plant routine operating data, the plant steady state gains, which CVs and MVs are possibly allowed to change their limits, and the preference to make such changes, are required.

This chapter builds on the previous work addressing the issue of MPC constraint analysis under a Bayesian statistics framework (Agarwal 2007) and as a result, we will frequently make reference to it for the sake of consistency and proper comparison. The previous work considered the controlled variables in the economic objective function as the quality variables under consideration. It also assumed that the profit function was discretized into different zones of profit in its problem formulation and analysis. The considerations were also limited to a linear profit function.

The main contributions of this chapter include the use of a continuous-valued profit function for the MPC constraint analysis and maximum-likelihood location determination of the quality variable(s) in the decision making process. In addition, the profit function consideration is extended to include both linear and quadratic forms. This approach allows us to extend the definition of quality variables to include all the variables in the original economic objective function of the MPC controller, without being limited to just the controlled variables. The derivation of the results is more mathematically elegant and, as a result, closed forms of solutions are obtained.

Using only controlled variables (CVs) as the quality variables is restrictive because the consideration is limited to only the product qualities being within or outside the specification. Other important variables in the MPC economic objective function (which could also be quality variables) cannot be considered in the same way as the product qualities and their specifications. Using the continuous-valued function gives us the ability to use all the variables in the economic objective function and this leads to a more complete form of solutions as will be shown subsequently. Also, as we will see in Chapter 5, using the continuous-valued function allows for a natural extension to the consideration of statistical dependence between quality variables.

The work by Agarwal (2007) was an extension of previous work by Xu *et al* (2007). Xu *et al* (2007) formulated the LMIPA algorithm and that approach considers only the mean operating points of the quality variables. It is however, more realistic to consider the distribution (such as the mean and standard deviation or variance in the Gaussian distribution case) of the quality variables, rather than just the mean operating points. This gives a more practical understanding of the process. Agarwal (2007) developed an algorithm for considering the distribution of the quality variables and their effect on the potential benefit to be obtained from the process. This previous work, however, assumed that the continuous-valued function for the expected returns could be discretized into zones of profit related to the process being above, within, or below the desired product specifications. While this was a commendable first step, the resolution from this kind of discretization is rather low, the information lost due to the discretization could prove valuable in the analysis, and the dimension of the problem can grow quickly, making its application in large-scale process impossible. In this work, we show that the continuous-valued function can be used without discretization and thus we avoid loss of information due to poor resolution, and the results are compact. Two case studies of MPC applications are provided to illustrate the proposed method. The new results are also compared to that obtained from the previously proposed discrete method (Agarwal 2007).

## 4.2 LMIPA revisited

Linear matrix inequality *based* performance assessment (LMIPA) is a tool for MPC performance assessment based on the work of (Xu *et al.* 2007). For a particular process, the assessment of yield is calculated for various cases as follows (Agarwal 2007):

1. Assessment of ideal yield
2. Assessment of optimal yield without tuning the controller
3. Assessment of improved yield by variability reduction

4. Assessment of improved yield by relaxing constraints

5. Constraint tuning for desired yield

These aspects have extensively been described and discussed by Xu *et al* (2007) and Agarwal (2007). In this work, we will consider the effect of constraint adjustment (or tuning) of the MPC on the potential yield of the plant and make some inferences on the importance of constraint tuning within a Bayesian statistics framework. Chapter 3 has given a brief background on Bayesian analysis and networks upon which we will build in this chapter.

For an MPC application with $n$ inputs and $m$ outputs, let $K$ be the steady state gain matrix and $(\overline{y_{i0}}; \overline{u_{j0}})$ be the current mean, (or base case) operating points. This state of the process based on the current steady state operation, prior to the benefit analysis, is called the base case operation. Also let the number of controlled variables (CVs) and the number of manipulated variables (MVs) for which constraint limit changes are allowed (by relaxation) be $a$ and $b$ respectively. This makes a total of $N = a + b$ variables for which we can make limit changes.

With *yes* and *no* as the options for applying the limit change to these $N$ variables, there will be $2^N$ combinations for applying the constraint changes. Each combination of limits change will have a specific optimal return and the optimum values of the mean operating point can be obtained through optimization, which will affect the MPC performance. Also, let $(Ly_i; Hy_i)$ be the low and the high limits for $y_i$, and $(Lu_j; Hu_j)$ be the low and the high limits for $u_j$ , respectively. The optimization is carried out for each of the possible combinations, with the real time CV/MV data collected. The objective function is the economic objective function of the MPC controller and the constraints are the CV and MV constraint limits, while also taking into account the variability and the steady state gain relations. The quality variables are defined *as all the variables (CVs and MVs) in the MPC economic benefit function which have non-zero linear and quadratic coefficients.*

The optimization problem can be defined as the linear-quadratic function:

$$J = \sum_{i=1}^{m}(\epsilon_i y_i + \theta_i^2(y_i - \eta_i)^2) + \sum_{j=1}^{n}(\delta_j u_j + \gamma_j^2(u_j - \nu_j)^2) \tag{4.1}$$

where, $(y_i, u_j)$ and $(\eta_i, \nu_j)$ are the mean operating points and target values for the $i^{th}$ CV $(y_i)$ and $j^{th}$ MV $(u_j)$, respectively. $\epsilon_i, \theta_i$ are the linear and quadratic coefficients for $y_i$ (outputs) and $\delta_j, \gamma_j$ are the linear and quadratic coefficients for $u_j$ (inputs), respectively. We also assume that within the constraints of the MVs and CVs, the derivative of the objective function is nonzero, i.e. the optimum does not occur inside the constraint limits. With $(\overline{y}_{i0}; \overline{u}_{j0})$ as the base case mean operating point, $(y_i; u_j)$ as the optimum operating point, when the base case operating points are moved by $(\Delta y_i; \Delta u_j)$, the equality and inequality constraints that need to be satisfied for the economic objective function are shown in Equations 4.2 to 4.8 below (Xu *et al.* 2007):

$$\Delta y_i = \sum_{j=1}^{n} (K_{ij} \times \Delta u_j) \tag{4.2}$$

$$y_i = \overline{y}_{i0} + \Delta y_i \tag{4.3}$$

$$u_j = \overline{u}_{j0} + \Delta u_j \tag{4.4}$$

$$Ly_i \leq y_i \leq Hy_i \tag{4.5}$$

$$Lu_j \leq u_j \leq Hu_j \tag{4.6}$$

$$Ly_i - y_{holi} \times r_{y_i} \leq y_i \leq Hy_i + y_{holi} \times r_{yi} \tag{4.7}$$

$$Lu_j - u_{holj} \times r_{u_j} \leq u_j \leq Hu_j + u_{holj} \times r_{uj} \tag{4.8}$$

where $i = 1, 2, \ldots, m$, and $j = 1, 2, \ldots, n$. We allow up to 5% constraint violation for the constrained variables in the desire to maximize the profit potential (Latour *et al.* 1986, Martin *et al.* 1991). $K_{ij}$ represents the elements of the gain matrix of the MPC. $Ly_i$ and $Hy_i$ are the low and the high limits for $y_i$ respectively. $Lu_i$ and $Hu_i$ are the low and the high limits for $u_j$ respectively. $r_{y_i}$ and $r_{u_j}$ are the allowable change (percentage of the range) in the constraint limits for the process variables, i.e. 5% for process variables with changeable constraints and 0% for others; $y_{holi}$ , $u_{holj}$ are half of the constraint limits for $y_i$ and $u_j$ respectively.

Thus, the economic objective function for each constraint tuning can be specified as (Xu *et al.* 2007):

$$\min_{\overline{y}_i, \overline{u}_j} J \text{ subject to Equations } 4.2; 4.3; 4.4; 4.5; 4.6; 4.7; 4.8 \tag{4.9}$$

$2^N$ optimum operating points are obtained for the constraint change cases. Superimposing the $2^N$ optimum operating points with the base case variability and assuming the data to be Gaussian distributed, the probability distribution for the data can be obtained (Agarwal 2007).

The Bayesian network is subsequently created with $N$ parent nodes, $q$ child nodes (where $q$ refers to the number of quality variables, i.e. all variables in the economic objective function with non-zero linear and quadratic coefficients). For each of the cases of limit change considerations, the parent nodes have two states *(yes, no)* where *yes* implies *change the limits* and *no* implies *do not change the limits.* By changing the constraint limits, the quality variables are optimized to operate as close as possible to their maximum-return values. At the same time, other non-quality variables are also moved due to the interaction between the process variables. Each optimal value taken by the quality variable is called its state and a state can take any value within the operating range and is continuous. The expected return from the process can therefore be calculated using Equation 4.10. At this point, we introduce a new variable $z$ that represents all the quality variables (both CVs and MVs). This new variable will be useful for the formulations that follow

$$E(F) = \int_{z_1,\ldots,z_q} p(z_1,\ldots,z_q)F(z_1,\ldots,z_q)dz_1,\ldots,dz_q \tag{4.10}$$

$E(F)$ is the expected cost, $q$ represents the number of quality variables in the economic objective function and $F(z_1,\ldots,z_q)$ is an economic cost function which is continuous-valued in nature. The profit is estimated as the difference between the value of the base case cost function and the value of the cost function obtained after constraint limits have been adjusted. For this reason, the continuous-valued economic objective function is referred to as the *continuous-valued profit function* in this thesis.

## 4.3 Continuous-valued profit function

From the preliminary information on LMIPA previously provided in section 4.2 and Equation 4.10, and with $F(z_1,\ldots,z_q)$ having the following additive form (Agarwal *et al.* 2007*b*):

$$F(z_1,\ldots,z_q) = \sum_{i=1}^{q} F_i(z_i) \tag{4.11}$$

We can re-write Equation 4.10 as:

$$E(F) = \int_{z_1,\ldots,z_q} p(z_1,\ldots,z_q)[\sum_{i=1}^{q} F_i(z_i)]dz_1,\ldots,dz_q \tag{4.12}$$

Previously this problem was solved using a discretization of the above function into finite zones of profits based on the probabilities of the quality variables (specifically the products) to be above, within or below desired specifications (Agarwal 2007). However, we propose a solution to this problem by considering the continuous-valued function directly and thereby derive a closed form of solutions. The results will be shown to apply to both output and input variables in Equation 4.1.

**Proposition 1**

$$E(F) = \int_{z_1,\ldots,z_q} p(z_1,\ldots,z_q)[\sum_{i=1}^{q} F_i(z_i)]dz_1,\ldots,dz_q = E[\sum_{i=1}^{q} F_i(z_i)] \tag{4.13}$$

if

$$F_i(z_i) = \alpha_i z_i \text{ (Linear case), then} \tag{4.14}$$

$$E(F) = E[\sum_{i=1}^{q} F_i(z_i)] = \sum_{i=1}^{q} \alpha_i E(z_i) = \sum_{i=1}^{q} \alpha_i \overline{z_i} \tag{4.15}$$

if

$$F_i(z_i) = \alpha_i z_i + \beta_i^2 (z_i - \mu_i)^2) \text{ (Quadratic case), then} \tag{4.16}$$

$$E(F) = \sum_{i=1}^{q} (\alpha_i \overline{z}_i + \beta_i^2 (\sigma_{z_i}^2 + \overline{z}_i^2 - 2\overline{z}_i \mu_i + \mu_i^2)) \tag{4.17}$$

where $z_i$ are the $q$ quality variables, $\alpha_i$ and $\beta_i$ are the linear and quadratic coefficients, $\sigma_{z_i}$ are the variances of the quality variables and $\mu_i$ are the target values. The above results (for the linear and quadratic cases) hold whether the quality variables are dependent or independent (we will further discuss the formulation for the case when the quality variables are dependent in Chapter 5). When the quality variables are dependent, the proof is shown below. For simplicity, we use two quality variables, and we will consider the quadratic case for generality:

**Proof of Proposition 1**

From Equation 4.12, using q=2 for simplicity

$$E(F) = \int_{z2}\int_{z_1}(p(z_1,z_2)[F_1(z_1)+F_2(z_2)])dz_1dz_2$$
$$= \int_{z_2}\int_{z_1}(p(z_1,z_2)[F_1(z_1)]+p(z_1,z_2)[F_2(z_2)])dz_1dz_2$$

From equation 4.13, for i=1,2,

$$F_1(z_1) = \alpha_1 z_1 + \beta_1^2(z_1-\mu_1)^2$$
$$F_2(z_2) = \alpha_2 z_2 + \beta_2^2(z_2-\mu_2)^2$$

Therefore

$$\begin{aligned}
E(F) &= \alpha_1\int_{z_2}\int_{z_1}p(z_1,z_2)z_1dz_1dz_2 + \beta_1^2\int_{z_2}\int_{z_1}p(z_1,z_2)(z_1^2+\mu_1^2-2z_1\mu_1)dz_1dz_2\\
&+\alpha_2\int_{z_2}\int_{z_1}p(z_1,z_2)z_2dz_1dz_2 + \beta_2^2\int_{z_2}\int_{z_1}p(z_1,z_2)(z_2^2+\mu_2^2-2z_2\mu_2)dz_1dz_2\\
&=\alpha_1\int_{z_1}\int_{z_2}[p(z_1,z_2)dz_2]z_1dz_1 + \beta_1^2\int_{z_1}\int_{z_2}[p(z_1,z_2)dz_2]z_1^2dz_1\\
&+\beta_1^2\mu_1^2\int_{z_1}\int_{z_2}[p(z_1,z_2)dz_2]dz_1 - 2\beta_1^2\mu_1\int_{z_1}\int_{z_2}[p(z_1,z_2)dz_2]z_1^2dz_1\\
&+\alpha_2\int_{z_2}\int_{z_1}[p(z_1,z_2)dz_1]z_2dz_2 + \beta_2^2\int_{z_2}\int_{z_1}[p(z_1,z_2)dz_1]z_2^2dz_2\\
&+\beta_2^2\mu_2^2\int_{z_2}\int_{z_1}[p(z_1,z_2)dz_1]dz_2 - 2\beta_2^2\mu_2\int_{z_2}\int_{z_1}[p(z_1,z_2)dz_1]z_2^2dz_2
\end{aligned} \tag{4.18}$$

From the definition of marginal probability

$$\int_{z_1}p(z_1,z_2)dz_1 = p(z_2) \text{ and } \int_{z_2}p(z_1,z_2)dz_2 = p(z_1) \tag{4.19}$$

Therefore Equation 4.18 becomes:

$$\begin{aligned}
E(F) &= \alpha_1\int_{z_1}p(z_1)z_1dz_1 + \beta_1^2\int_{z_1}p(z_1)z_1^2dz_1 + \beta_1^2\mu_1^2\int_{z_1}p(z_1)dz_1 - 2\beta_1^2\mu_1\int_{z_1}p(z_1)z_1^2dz_1\\
&+\alpha_2\int_{z_2}p(z_2)z_2dz_2 + \beta_2^2\int_{z_2}p(z_2)z_2^2dz_2 + \beta_2^2\mu_2^2\int_{z_2}p(z_2)dz_2 - 2\beta_2^2\mu_2\int_{z_2}p(z_2)z_2^2dz_2\\
&=\alpha_1E(z_1)+\beta_1^2E(z_1^2)+\beta_1^2\mu_1^2-2\beta_1^2\mu_2E(z_1)\\
&+\alpha_2E(z_2)+\beta_2^2E(z_2^2)+\beta_2^2\mu_2^2-2\beta_2^2\mu_2E(z_2)
\end{aligned} \tag{4.20}$$

Thus

$$E(F) = \alpha_1\overline{z_1} + \beta_1^2(\sigma_{z_1} + \overline{z_1}^2 - 2\overline{z_1}\mu_1) + \alpha_2\overline{z_2} + \beta_2^2(\sigma_{z_2} + \overline{z_2}^2 - 2\overline{z_2}\mu_2)$$

and in general form,

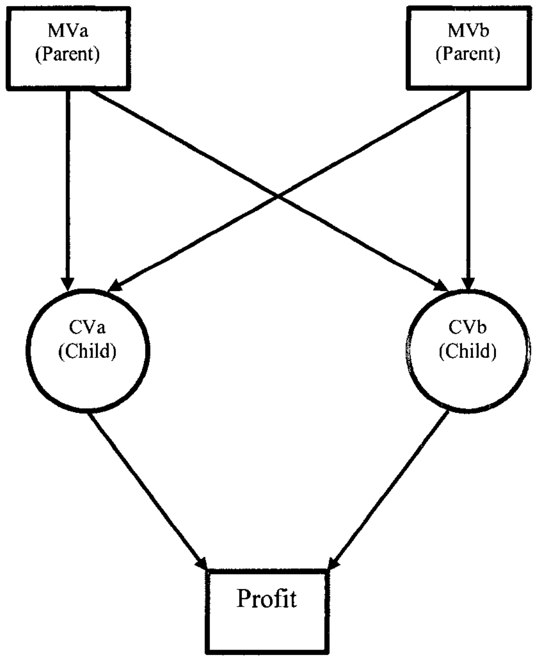
$$E(F) = \sum_{i=1}^{q}(\alpha_i\overline{z}_i + \beta_i^2(\sigma_{z_i}^2 + \overline{z}_i^2 - 2\overline{z}_i\mu_i + \mu_i^2)) \tag{4.21}$$

This proves Equation 4.17. The above formulation can be applied to Equation 4.1 to give the general form when the input and output variables are considered separately, as follows:

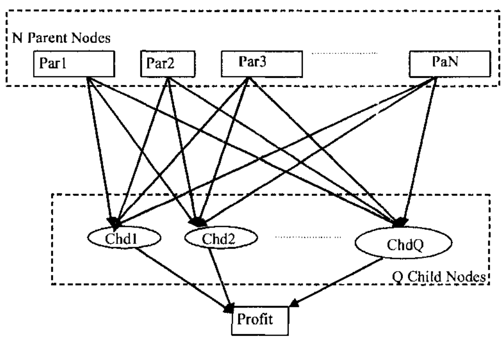
$$E(F) = \sum_{i=1}^{m}(\epsilon_i\overline{y}_i + \theta_i^2(\sigma_{y_i}^2 + \overline{y}_i^2 - 2\overline{y}\eta_i + \eta_i^2)) + \sum_{j=1}^{n}(\delta_j\overline{u}_j + \gamma_j^2(\sigma_{u_j}^2 + \overline{u}_j^2 - 2\overline{u}\nu_j + \nu_j^2)) \tag{4.22}$$

### 4.3.1 Bayesian approach for MPC constraint analysis

For the cases of constraint change only, the covariance remains the same as that of the base case operation since the dynamic control is not changed. Optimizing the objective function subject to the new constraints provides the optimum operating points that are used as the new mean operating points for the benefit analysis. The parent nodes are defined as the variables that have the possibility of having their limits changed. The prior probability or the *a priori* for the parent nodes can be user defined or obtained from historical data. It indicates the preference to change or not to change the limits. For example, if a parent node has *a priori* of 0.7 for making a change to the limits, this means that the constraint limits for this variable has 70% tendency to change and 30% not to change. In this work, to define the conditional probability table for the parent nodes, we have used equal probability to change or not change the constraints when performing the optimization and, as a result, there is an equal probability to change or not to change the limits (i.e. 50% to change or not to change the limits). In Agarwal *et al.* (2007)(Agarwal *et al.* 2007*a*, Agarwal *et al.* 2007*b*), the selection of equal prior probabilities has been shown to give the maximum-likelihood solution. Figure 4.1 show the DAGs (directed acyclic graphs) of two Bayesian networks. Figure 4.1(a) shows the structure for two parent nodes and two child nodes, while Figure 4.1(b) shows the structure for a network with multiple parent and child nodes and how they affect the obtainable profit. Reference can be




(a) Two parents and two child nodes




(b) N parents and Q child nodes

Figure 4.1: Illustrative Directed Acyclic Graphs (DAG) of Bayesian networks

made to the work by Agarwal *et al* (2007) (Agarwal *et al.* 2007*b*) for the procedure for building Bayesian networks for this type of analysis.

The child nodes are defined as the quality variables, i.e, the variables which affect the profit function. These are the variables with non-zero linear and/or quadratic coefficients in the MPC economic objective function. The conditional probability distributions of the child nodes are specified based on the optimum operating points obtained from the optimization step and the base-case standard deviations from the data.

The Bayesian analysis is based on the probability distribution of the data. For this work, the data is assumed to be Gaussian. The Bayesian analysis for *decision evaluation* is carried out using the Bayesian Network toolbox (BNT) developed by Kevin Murphy (Murphy 2004*b*, Murphy 2004*a*, Murphy 2001) and the *decision making* is directly solved by optimization.

### Decision evaluation

This refers to inferring the achievable profit, if certain decisions regarding limits change are made. For *decision evaluation*, the decision whether to change or not to change the limits is provided. This is equivalent to the evidence for the Bayesian network conditional on which, the probabilities of the locations of the quality variables are then estimated. Thus the profit can be evaluated based on the relation specified in Equation 4.17.

### Decision making

This refers to obtaining the maximum-a-posteriori explanation that will help to achieve a target value of the profit. For *decision making* purposes the target return is provided and the corresponding optimum values for the quality variables affecting the profit function are estimated. Based on the optimum values for the quality variables obtained, the algorithm suggests a combination of variables for which changing their constraint limits will most likely achieve the specified profit. It is not necessary however that, given a profit value, there is a unique set of quality CVs/MVs location. Multiple sets of locations may exist to yield the same desired profit. This is shown in Figure 4.2 and will be elaborated shortly.

Based on the probability distribution assumption for the base case variability, we use an optimization approach to find the locations of the quality variables (i.e their values) that will have the maximum probability of occurring while also yielding the desired profit. As a result, the maximum-a-posteriori estimate of the states of the parent nodes (i.e. change the limits or not) can then be obtained.

Based on the formulae for the continuous-valued profit function, we proceed to show how the *decision making* can be carried out. First consider the case when the quality variables are independent, i.e.

$$p(z_1, \ldots, z_q) = \prod_{i=1}^{q} p(z_i) \tag{4.23}$$

Thus:

$$\prod_{i=1}^{q} p(z_i) = \prod_{i=1}^{q} \left[ \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left( -\frac{(z_i - \overline{z_i})^2}{2\sigma_i^2} \right) \right] \tag{4.24}$$

where Gaussian probability distributions of the variables have been assumed for each of the $q$ quality variables. $\bar{z}_i$ is a set of optimal operating points for a specific constraint tuning, obtained from the *decision evaluation* step.

To obtain the most likely locations of the quality variables that will give the desired return, we maximize the probability density function:

$$p = \max_{z_i, \cdots, z_q} \prod_{i=1}^{q} \left[ \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left( -\frac{(z_i - \overline{z_i})^2}{2\sigma_i^2} \right) \right] \tag{4.25}$$

using log-likelihood:

$$\max_{z_i, \cdots, z_q} \prod_{i=1}^{q} p(z_i) = \max_{z_i, \cdots, z_q} \{\log(\prod_{i=1}^{q} \left[ \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left( -\frac{(z_i - \overline{z_i})^2}{2\sigma_i^2} \right) \right] )\} \tag{4.26}$$

thereby obtaining

$$\begin{aligned} \max_{z_i, \cdots, z_q} \prod_{i=1}^{q} p(z_i) &= \max_{z_i, \cdots, z_q} (\sum_{i=1}^{q} \left( -\frac{(z_i - \overline{z_i})^2}{2\sigma_i^2} \right)) \text{ or equivalently} \\ \max_{z_i, \cdots, z_q} \prod_{i=1}^{q} p(z_i) &= \min_{z_i, \cdots, z_q} (\sum_{i=1}^{q} \left( \frac{(z_i - \overline{z_i})^2}{2\sigma_i^2} \right)) \end{aligned} \tag{4.27}$$

subject to the specified profit:

$$\sum_{i=1}^{q} (\alpha_i z_i + \beta_i^2(\sigma_{z_i}^2 + z_i^2 - 2z_i\mu_i + \mu_i^2)) = R_T \tag{4.28}$$

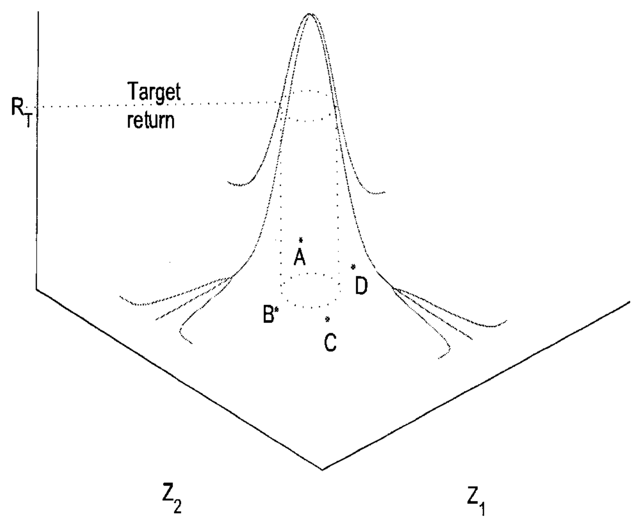



Figure 4.2: Schematic illustrating choice of combination of quality variables to achieve a target return

and also subject to the constraints on the quality variables.

As illustrated in Figure 4.2, all points falling on the circle in the $z_1$ and $z_2$ plane satisfy the profit constraints. Thus the solution to this problem is non-unique and the above optimization is justified. In Figure 4.2 the locations A, B, C and D are examples of these points. These locations could all satisfy the specified target return as shown in Equation 4.35. The probability associated with all the possible locations is determined by the joint distribution of $z_1$ and $z_2$ and the optimization is carried out to find the location that has the highest probability of occurring.

Equation 4.34 is an optimization problem with a quadratic objective function and a quadratic equality constraint (Equation 4.35). This optimization problem can be solved using YALMIP(Lofberg 2004), an open-source optimization toolbox for MATLAB(MathWorks 2007*a*). YALMIP is an interface for SeDuMi, SDPT3 and many other optimization algorithms. It is used for rapid prototyping of optimization problems and although it initially focussed on semi-definite optimization problems, its scope has been significantly extended to include various other optimization problems. Other optimization toolboxes used are the MATLAB optimization toolbox(MathWorks 2007*b*), SeDuMi toolbox(Sturm 1999) and SDPT3 (Toh *et al.* 1999), which are also useful for semi-definite programming and optimization over symmetric cones.

Equations 4.27 and 4.28 are used $2^N$ times, once for each of the combinations of limits change, where N is the number of variables chosen for limits change, and a set of optimum values are obtained in each case. This results in $2^N$ sets of optimum values. The *decision making* process is then completed by using Equation 4.29 to determine which set of optimum values out of the $2^N$ cases has the highest probability of occurring.

$$\max_{(z_1,...,z_q)^*} \{p(z_1,...,z_q)_1,...,p(z_1,...,z_q)_{2^N}\} \tag{4.29}$$

We will illustrate the above formulations using an example where $q = 2$ for the case when $z_1$ and $z_2$ are independent, i.e.

$$p(z_1, z_2) = p(z_1)p(z_2) \tag{4.30}$$

Thus:

$$p(z_1)p(z_2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} exp\left(-\frac{(z_1-\overline{z_1})^2}{2\sigma_1^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma_2^2}} exp\left(-\frac{(z_2-\overline{z_2})^2}{2\sigma_2^2}\right) \tag{4.31}$$

As stated above, to obtain the most likely locations of the quality variables that will give the desired return, we maximize the probability density function:

$$p = \max p(z_1, z_2) \tag{4.32}$$

Using log-likelihood:

$$\max p(z_1)p(z_2) = \max\{\log(p(z_1)p(z_2))\} \tag{4.33}$$

this gives:

$$\begin{aligned} &\max_{z_1,z_2}\left(-\frac{(z_1-\overline{z_1})^2}{2\sigma_1^2}\right) - \left(\frac{(z_2-\overline{z_2})^2}{2\sigma_2^2}\right) \text{ or equivalently} \\ &\min_{z_1,z_2}\left(\frac{(z_1-\overline{z_1})^2}{2\sigma_1^2}\right) + \left(\frac{(z_2-\overline{z_2})^2}{2\sigma_2^2}\right) \end{aligned} \tag{4.34}$$

subject to

$$\sum_{i=1}^{2}(\alpha_i z_i + \beta_i^2(\sigma_{z_i}^2 + z_i^2 - 2z_i\mu_i + \mu_i^2)) = R_T \tag{4.35}$$

and subject to Equations 4.5 and 4.6.

The values of $\bar{z}_1$ and $\bar{z}_2$ in Equation 4.34 are obtained from the optimization step previously described. In this illustration, since we have $2^2 = 4$ combinations of limits change, we will have 4 pairs of mean values ($\bar{z}_1$ and $\bar{z}_2$) for each case of constraints change. For each pair of means, the optimization described in Equation 4.34 and its associated constraints will give the corresponding optimum values of $z_1$ and $z_2$. Overall, for all the cases considered, four pairs of optimum values for $z_1$ and $z_2$ are obtained (i.e. $(z_1, z_2)_1, (z_1, z_2)_2, (z_1, z_2)_3$ and$(z_1, z_2)_4$). Each of these pairs will satisfy Equation 4.34. Finally, using Equation 4.36, the pair of optimum values (i.e. $(z_1, z_2)_1, (z_1, z_2)_2, (z_1, z_2)_3$ or $(z_1, z_2)_4$) that has the maximum probability of occurring is chosen, i.e.,

$$\max_{(z_1,z_2)^*} \{p((z_1, z_2)_1), p((z_1, z_2)_2), p((z_1, z_2)_3), p((z_1, z_2)_4)\} \tag{4.36}$$

where $(z_1, z_2)^*$ is the pair of optimum values of all four cases, with the maximum probability of occurrence. This solution is also the maximum a posteriori with the specified profit value given as the evidence.

### 4.3.2 Illustrative example of a Bayesian network for MPC constraint analysis

The following example illustrates the procedure for building a Bayesian network to carry out the procedures for MPC constraint analysis as described previously. We consider a 2 input-2 output multivariable system with a steady state gain matrix as follows:

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{4.37}$$

In the following discussion we will differentiate between the process variable (CV or MV) and the logical variable (the decision whether to change limits or not) by labeling them as CV or MV (process variables) and $CV^*$ or $MV^*$ (logical variables), respectively.

Table 4.1: Linear and quadratic coefficients of variables and limits change specifications

| Variable | L Coeff | Q Coeff | Limits Change |
|---|---|---|---|
| **CV1** | 0 | 0 | Yes |
| **CV2** | $\alpha_{CV2}$ | 0 | No |
| **MV1** | $\alpha_{MV1}$ | 0 | No |
| **MV2** | 0 | 0 | Yes |

Table 4.2: Specifications of parent and child nodes

| **Parent Nodes** | $CV1^*$ | $MV2^*$ |
|---|---|---|
| **States** | (Yes, No) | (Yes, No) |
| **Child Nodes** | **CV2** | **MV1** |
| **States** | $(\mu, \sigma^2)$ | $(\mu, \sigma^2)$ |

Let CV2 and MV1 be quality variables (i.e variables in the MPC economic objective functions with non-zero linear and/or quadratic coefficients) while CV1 and MV2 are constraint variables, see Table 4.1.

Since we have specified two variables for limits change, the Bayesian network will have two parent nodes ($CV1^*$ and $MV2^*$). Also, since we have two variables with non-zero linear and quadratic coefficients, we have two child nodes for the network (CV2 and MV1). Table 4.2 shows the specifications for the parent and child nodes. The conditional probability distributions (CPD) of the nodes are thus defined. The parent nodes will have conditional probability tables (CPT) as shown in Table 4.3. We assume that the prior probabilities of each of the parent nodes is 0.5, implying that there is an equal probability that we will make a change in the limits or leave the limits unchanged.

For this system, we will have $2^2 = 4$ (see Table 4.4) combinations of limits change for which we can carry out optimizations to obtain the optimum values of the quality variables based on our choice of limits change. The CPD's of the child nodes, being

Table 4.3: Specifications of parent and child nodes

| Parent Nodes | Change Limits | Do not Change Limits |
|---|---|---|
| $CV1^*$ | 0.5 | 0.5 |
| $MV2^*$ | 0.5 | 0.5 |

Table 4.4: Different cases of limits change combinations

| Parent Nodes | Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|---|
| $CV1^*$ | No | Yes | No | Yes |
| $MV2^*$ | No | No | Yes | Yes |

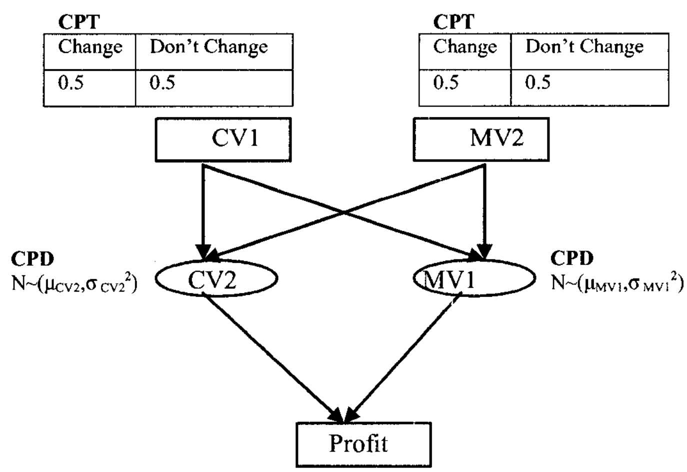



Figure 4.3: Bayesian network for illustrative example

continuous, are defined by mean values and variances for each node. The mean values of these child nodes are the optimum values obtained from the optimization step and the variances or covariances are the base case variances or covariances of the quality variables, obtained from data.

With CV2 and MV1 identified as quality variables and their CPD specified as discussed above, with the CPT for $CV1^*$ and $MV2^*$ as parent nodes given as shown in Table 4.3, we have a Bayesian network as shown in Figure 4.3, which can subsequently be used for *decision making* and *decision evaluation.*

**Decision Evaluation:** Here we supply our decision to change or not change limits as evidence to the network. This evidence is used to estimate the probabilities of the child nodes to have a particular expected value, based on their probability distributions. This estimated mean value is then used to calculate the expected return, and subsequently, the potential profit that can be obtained from the process. The expected return is calculated using Equation 4.15 (for this linear case) as follows:

$$E(F) = \alpha_{CV2}\overline{CV2} + \alpha_{MV1}\overline{MV1} \tag{4.38}$$

As an illustration, let us assume we make a decision to change the limits of CV1 and leave MV2 unchanged. For simplicity of notation, $P(CV2|CV1^* = Yes, MV2^* = No)$ will be written as $P(CV2|CV1^*, MV2^*)$. With this information, the probability associated with the child node CV2 (for example) is as follows, using Bayes rule (Korb and Nicholson 2004, Wikipedia 2007):

$$P(CV2|CV1^*, MV2^*) = \frac{P(CV1^*|CV2) \times P(MV2^*|CV2)P(CV2)}{P(CV1^*) \times P(MV2^*)} \tag{4.39}$$

Using likelihood ratios, we have:

$$P(CV2|CV1^*, MV2^*) = \frac{\Lambda_1\Lambda_2 P(CV2)}{[\Lambda_1 P(CV2) + P(\neg CV2)][\Lambda_2 P(CV2) + P(\neg CV2)]} \tag{4.40}$$

where

$$\begin{aligned} \Lambda_1 &= \frac{P(CV1^*|CV2)}{P(CV1^*|\neg CV2)} \\ \Lambda_2 &= \frac{P(MV2^*|CV2)}{P(CV1^*|\neg CV2)} \end{aligned} \tag{4.41}$$

In the above equations (Equations 4.39, 4.40 and 4.41), P(CV2) is the prior probability of the occurrence of CV2, also known as the *a-priori*. $P(CV1^*|CV2)$ is the likelihood of obtaining evidence $CV1^*$ given that the occurrence of CV2 is true, $P(MV2^*|CV2)$ is the likelihood of obtaining evidence $MV2^*$ given that the occurrence of CV2 is true and $P(CV2|CV1^*, MV2^*)$ is the posterior probability of CV2 being true given that the evidence $CV1^*$ and $MV2^*$ are both obtained. $\neg CV2$ represents the case when the hypothesis of obtaining CV2 is not true.

**Decision Making:** In this case, a specified value for the expected returns is provided to the algorithm. Using Equation 4.34 for the *decision making*, we have the objective function for optimization as:

$$\min_{CV2, MV1} \left( \frac{(CV2 - \overline{CV2})^2}{2\sigma^2_{CV2}} \right) + \left( \frac{(MV1 - \overline{MV1})^2}{2\sigma^2_{MV1}} \right) \tag{4.42}$$

subject to

$$L_{CV2} \leq CV2 \leq H_{CV2} \tag{4.43}$$

Table 4.5: Linear and quadratic optimization coefficients - linear case

| | CV Number | | | | | | | | | | MV Number | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 |
| **Linear** | 0 | 0.2364 | 0 | 0 | 0 | 0 | 0 | 0.1714 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Quadratic** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$L_{MV1} \leq MV1 \leq H_{MV1} \tag{4.44}$$

$$\alpha_{CV2}CV2 + \alpha_{MV1}MV1 = R_T \tag{4.45}$$

$$\max_{CV2,MV1^*}(p((CV2, MV1)_1), p((CV2, MV1)_2), p((CV2, MV1)_3), p((CV2, MV1)_4)) \tag{4.46}$$

where $R_T$ is the specified returns, $L_{CV2}$ and $L_{MV1}$ are the lower limits for CV2 and MV1, $H_{CV2}$ and $H_{MV1}$ are the upper limits for CV2 and MV1, respectively. These upper and lower limits will change for each of the four cases described above. $(CV2, MV1)^*$ are the optimum values for CV2 and MV1 thus obtained from the optimization.

### 4.3.3 Illustration of the Bayesian Method through a Case Study of a Binary Distillation Column

The binary distillation column shown in Figure 4.4 is a system with 10 Controlled Variables (outputs, y)and 4 Manipulated Variables (inputs, u). A detailed description of this system is available in the literature (Volk *et al.* 2005, Agarwal 2007). A study was carried out by Agarwal *et al.* (2007)(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*) for this system based on the control objective described in Volk *et al.* (2005)(Volk *et al.* 2005). The linear quadratic objective function coefficients required to carry out the analysis are provided in Table 4.5. This case study is included here to compare the proposed approach with the previous work.

The objective function can be defined using the information in Table 4.5 and substituting into Equation 4.1. $\epsilon_i$ and $\theta_i$ are the first ten values of the linear and quadratic rows in Table 4.5, respectively. Likewise, $\delta_j$ and $\gamma_j$ are the last four values of the linear and quadratic rows in Table 4.5, respectively.

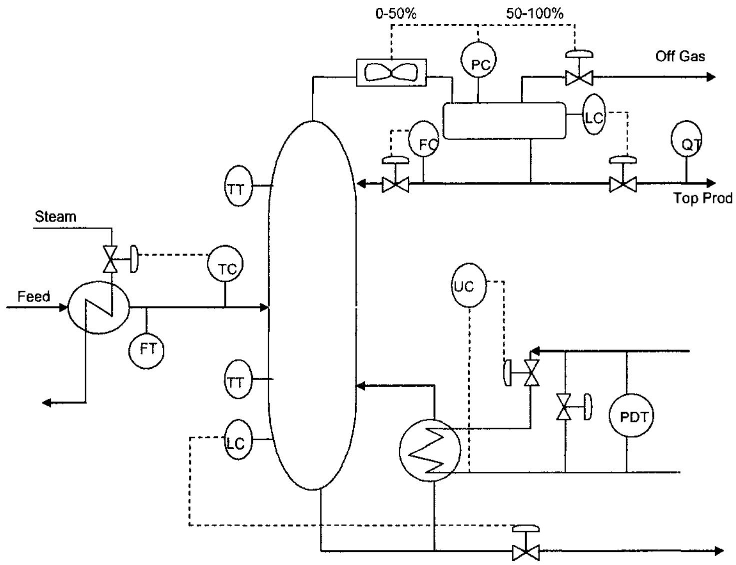



Figure 4.4: Schematic of simulated binary distillation Column

### 4.3.4 Results

From Equation 4.15, the expected returns are calculated and the results are shown to be the same as those of the deterministic LMIPA for the linear case. This is as expected since the LMIPA approach is based on the mean operating points only. For the quadratic case, we would expect a difference since the formulation (Equation 4.17) includes the variance (or covariance) term(s) as well as the mean values.

#### Linear objective function

In the following discussions we will differentiate between the process variable (CV or MV) and the logical variable (the decision whether to change limits or not) by labeling them as CV or MV (process variables) and $CV^*$ or $MV^*$ (logical variables), respectively. The results for the linear case are presented in Table 4.7 (Table 4.6 shows the different combinations of limits change).

The objective function can be defined using the information in Table 4.5 and substituting into Equation 4.1. $\epsilon_i$ and $\theta_i$ are the first ten values of the linear and quadratic rows in Table 4.5, respectively. Likewise, $\delta_j$ and $\gamma_j$ are the last four values of the linear and quadratic rows in Table 4.5, respectively.

Table 4.6: Different combinations of limits change

| CASE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $CV5^*$ | N | Y | N | Y | N | Y | N | N | Y | N | N | Y | N | Y | N | Y |
| $CV9^*$ | N | N | Y | Y | N | N | Y | Y | Y | N | Y | Y | N | N | Y | Y |
| $MV1^*$ | N | N | N | N | Y | Y | N | Y | Y | N | N | N | Y | Y | Y | Y |
| $MV3^*$ | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y | Y | Y | Y |

**a. Decision Evaluation:**

To calculate the potential profit that can be obtained from the change of limits. This is done as follows:

$$\text{Profit} = \textit{Base case cost - Cost from Limits Change} \tag{4.47}$$

*Base case cost:* Cost from operating at current mean values. In this case study, *Base case cost* = -69.6293

*Cost from Limits Change:* Cost from operating at optimum values due to limits change

Equation 4.47 is used to obtain Table 4.7. The notation in Table 4.7 is explained below:

- R-LMIPA: Returns based on LMIPA
- R-case: Returns calculated from the proposed Bayesian-based approach

From Equations 4.15 and 4.47, the profits are calculated and the results are shown to be the same as those of the deterministic LMIPA for the linear case, this is expected. For the quadratic case, we would expect a difference since the formulation in Equation 4.17 includes the variance (or covariance) term(s) as well as the mean values.

From Table 4.7, comparing case 1 and case 16 (and the cases in between), we observe that there is an overall trend between case 1 (where only one variable has its constraint limits changed) and case 16 (where all four process variables have their constraint limits changed). This trend indicates that the more the number of process variables that are chosen for constraint change, the more the profits that can be obtained from the process. This observation would intuitively be expected because we get more degrees of freedom as more variables are allowed to have their limits

Table 4.7: Profits obtained for combinations of limits change

| CASE | R-LMIPA | R-case |
|---|---|---|
| 1 | 1.7513 | 1.7513 |
| 2 | 2.0257 | 2.0257 |
| 3 | 1.7513 | 1.7513 |
| 4 | 2.0303 | 2.0303 |
| 5 | 1.9212 | 1.9212 |
| 6 | 2.2025 | 2.2025 |
| 7 | 1.9212 | 1.9212 |
| 8 | 2.2071 | 2.2071 |
| 9 | 1.7613 | 1.7613 |
| 10 | 2.0309 | 2.0309 |
| 11 | 1.7613 | 1.7613 |
| 12 | 2.0364 | 2.0364 |
| 13 | 1.9312 | 1.9312 |
| 14 | 2.2086 | 2.2086 |
| 15 | 1.9312 | 1.9312 |
| 16 | 2.2132 | 2.2132 |

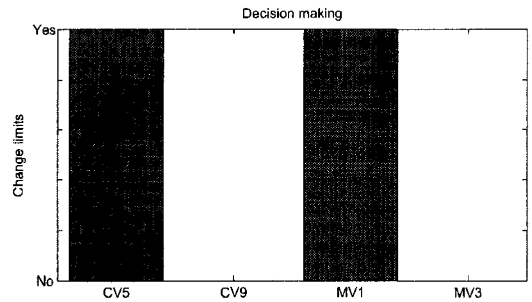
changed. Table 4.7 also shows that the profits are the same for the deterministic LMIPA and the Bayesian-based LMIPA, for the linear objective function.

The overall trend in Table 3 shows that the values of the profits for the even-numbered cases are generally larger than the values of the profit for the odd-numbered cases. While this may not always be so, the explanation for this observation is due to the fact that some variables contribute more to the total profit when their constraints are adjusted, than others; and whenever these variables are included in the combination of variables chosen for limits change as *yes* or *no*, this affects the value of the total profit obtained. For example, by comparing Tables 4.6 and 4.7, and comparing the results from cases 2 and 3 (with respect to the combination of limits change), we note that changing the limits for only CV5 produces some profit and so whenever CV5 is included in the combination as a *yes*, more profit would be expected. Whereas, in Case 3, changing CV9 alone does not produce any extra profit (same profit as case 1). In case 5, changing the limits of only MV1 produces some profit, but it is not up to the profit produced when the limits of *both* CV5 and CV9 are changed. Hence, although changing only CV9 did not produce any profit, we find that changing the limits for both CV5 and CV9 produces more profit than from only CV5. Therefore, although we find an overall trend indicating an increase in profits as more variables are considered for constraint limits change, the profit obtainable from each case depends on the combination of variables chosen and the individual contribution of each variable to the total profit.

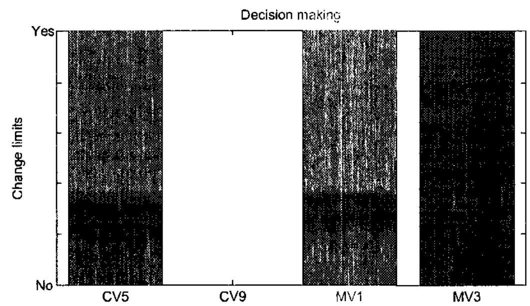
Table 4.8: Comparison of *decision evaluation* results for discrete and continuous algorithms

| Case | R-LMIPA | Discrete | Continuous |
|---|---|---|---|
| 1 | 1.7513 | 1.7513 | 2.2809 |
| 2 | 2.0257 | 2.0257 | 2.2809 |
| 3 | 1.7513 | 1.7513 | 2.2809 |
| 4 | 2.0303 | 2.0303 | 2.2809 |
| 5 | 1.9212 | 1.9212 | 2.2809 |
| 6 | 2.2025 | 2.2025 | 2.2809 |
| 7 | 1.9212 | 1.9212 | 2.2809 |
| 8 | 2.2071 | 2.2071 | 2.2809 |
| 9 | 1.7613 | 1.7613 | 2.5511 |
| 10 | 2.0309 | 2.0309 | 2.5117 |
| 11 | 1.7613 | 1.7613 | 2.5511 |
| 12 | 2.0364 | 2.0364 | 2.5511 |
| 13 | 1.9312 | 1.9312 | 2.5511 |
| 14 | 2.2086 | 2.2086 | 2.5511 |
| 15 | 1.9312 | 1.9312 | 2.5511 |
| 16 | 2.2132 | 2.2132 | 2.5511 |

Comparing this proposed approach using the continuous-valued function with the approach proposed by Agarwal *et al.* (2007)(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*) for the discrete calculation and LMIPA by Xu *et al.* (2007)(Xu *et al.* 2007) for the deterministic calculation, Table 4.8 shows that the LMIPA and the proposed continuous version of the Bayesian approach agree perfectly, whereas the the results obtained from using the previous discrete approach(Agarwal *et al.* 2007*b*) does not clearly differentiate between the cases of limits change due to poor resolution of the zones of profits.




(a) Target profit = 2.14




(b) Target profit = 2.21

Figure 4.5: Decision making results for specified target profit

## b. Decision Making:

Here, we seek to determine the process variables which should have their constraints changed based on a user specified profit. Equation 4.27 is used for the optimization step to obtain the locations of the quality variables that will give the desired profit and then the algorithm determines the most probable combinations of variables for which changing their limits will give the desired results.

It is important to note that the *decision making* algorithm involves the determination of a specific location of the quality variable as evidence and not just the mean values as in the case of *decision evaluation.* Thus, the profit used in decision making is a point-wise profit while the profit used in decision evaluation is the averaged one.

To achieve a profit of 2.21 units, the results are shown in Figure 4.5(b), showing that we will need to change the limits of CV5, MV1 and MV3. Whereas, changing the limits of CV5 and MV1 will allow us to achieve our target value of 2.14 units of profit as shown in Figure 4.5(a).

We illustrate a comparison between the *decision making* results for the discrete

Table 4.9: Decision making: mean values of quality variables (continuous approach)

| | **Mean Values** | **Operating points (*decision making*)** |
|---|---|---|
| CV2 | 67.9986 | 60.5912 |
| CV8 | -500.025 | -500.0181 |

Table 4.10: Linear and quadratic optimization coefficients - quadratic case

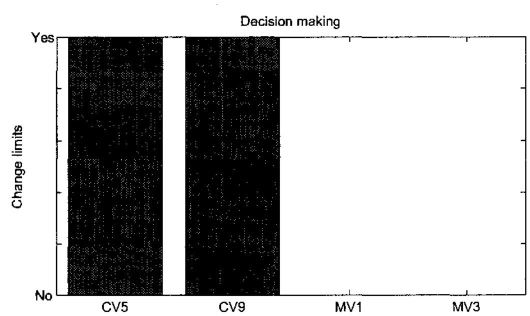
| | CV Number | | | | | | | | | | MV Number | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 |
| **Linear** | 0 | 0.2364 | 0 | 0 | 1 | 1.5 | 0 | 0.1714 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Quadratic** | 0 | 0.5 | 0 | 0 | 2 | 0 | 0 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 |

and continuous algorithms, with the following example. To achieve a target profit of 1.75 units, Figure 4.6 shows the results using both approaches. Figure 4.6(b) suggests that changing the limits of only CV9 will satisfy our objective while Figure 4.6(a) suggests that both CV5 and CV9 should have their limits changed. Since, we desire to achieve *the same target profit*, the result from continuous approach is more reasonable than the result from the discrete approach (the discrete approach being an approximation of the continuous approach).

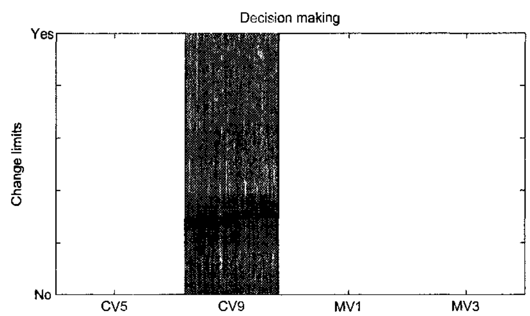
For the given target profit value (1.75), the calculated locations for the quality variables are shown in Table 4.9. The mean values (as obtained from the base case data) are compared with the suggested operating points obtained from the *decision making* analysis. We observe that the operating point of CV2 is lowered, compared to its mean value, thereby creating room for potential profit.

### Quadratic objective function

The linear and quadratic coefficients used for this case study are shown in Table 4.10. From Equation 4.17, the expected return is based on the mean operating points and the variances of each quality variable.




(a) Discrete approach




(b) Continuous approach

Figure 4.6: Comparison of discrete and continuous *decision making* results

Table 4.11: Decision evaluation: profits for quadratic case

| Case | R-LMIPA | R-case |
|---|---|---|
| 1 | 4014.04 | 4013.32 |
| 2 | 4983.54 | 4982.83 |
| 3 | 4014.72 | 4014.01 |
| 4 | 5075.68 | 5074.97 |

**a. Decision Evaluation:**

There is a difference between the results obtained by the deterministic LMIPA and the Bayesian-based methods. This difference can be accounted for by the product of the quadratic coefficients of the objective function and the variance of the corresponding quality variable. As shown in Equation 4.48:

$$\sum_{i=1}^{q}(\beta_i^2\sigma_i^2) \tag{4.48}$$

**b. Decision Making:**

The results using the Bayesian approach are as follows: To achieve a profit of 4500 units, the result is shown in Figure 4.7. It indicates that changing the limits of CV5 will allow us to achieve our desired profit. The suggested locations of the quality variables from *decision making* are shown in Table 4.12 (these values are most likely achieved when the limits of CV5 have been changed by 5%).

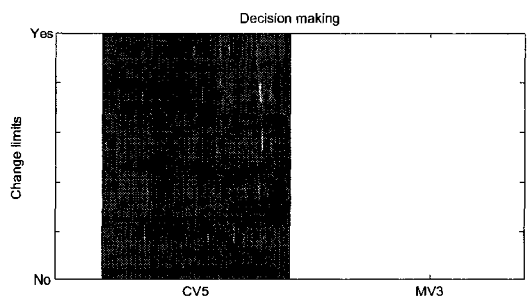



Figure 4.7: Decision making results for quadratic case

Table 4.12: Decision making results for quadratic case: locations of quality variables

| Quality Variables | Base case Mean | Operating points (decision making) |
|---|---|---|
| CV2 | 67.97 | 61.11 |
| CV5 | 86.32 | 79.93 |
| CV6 | 0.157 | 0.209 |
| CV8 | -500.11 | -497.49 |

## 4.4 Experimental validation

Reference will be made here to the three-tank system previously described in section 2.2.2 of Chapter 2. To avoid unnecessary repetition, we will not get into a detailed discussion of the setup. The model used for the MPC design is show in Equation 4.49. In this section, we will only describe the details of the MPC design and illustrate how this pilot-scale setup allows us to implement validation experiments.

$$\begin{pmatrix} CV_1 \\ CV_2 \\ CV_3 \end{pmatrix} = \begin{pmatrix} \frac{-0.04967}{z-0.8904} & 0 & 0 \\ \frac{0.03994z-0.04013}{z^2-1.764z+0.7789} & \frac{-0.04411z+0.04201}{z^2-1.841z+0.847} & 0 \\ \frac{0.0139z-0.01395}{z^3-1.897z^2+0.9023z} & \frac{0.03782z-0.03758}{z^2-1.778z+0.79} & \frac{-0.02095z+0.0211}{z^2-1.891z+0.8908} \end{pmatrix} \begin{pmatrix} MV_1 \\ MV_2 \\ MV_3 \end{pmatrix} \tag{4.49}$$

The sampling time is 8 seconds.

### 4.4.1 Linear objective function

Given that the control objective in this case is to maintain desired liquid levels in the tanks, we can further specify our economic objective function as the maximization of the liquid level in the third tank (i.e. $\max(100CV_3)$ or $\min(-100CV_3)$ this implies that CV3 has a linear coefficient value of $-100$). All other conditions are left unchanged unless further specified.

#### MPC controller parameters

The constraints are set as $0.5 \leq MV_i \leq 1.0$, and $0.05 \leq CV_i \leq 0.25$, where $i = 1, 2, 3$. The input rate weights are given as $[0.5, 0.5, 0.5]$, and the output weights are $[1, 0.8, 0.9]$. The control interval is 8 seconds, and the prediction and control horizons are specified as 15 and 2, respectively.

Table 4.13: Four combinations of limits change for linear validation example

| Parent Nodes | Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|---|
| $CV1^*$ | No | Yes | No | Yes |
| $MV3^*$ | No | No | Yes | Yes |

By applying the designed MPC controller, and ensuring that all of the controlled variables and manipulated variables are running within their corresponding constraints, the real time experimental data is collected and regarded as the base case operation in this study. The variables chosen for limits change are $CV_1$ and $MV_3$. This implies that we will have four ($2^2 = 4$) possible combinations of limits change (see Table 4.13).

### a. Decision Evaluation

Table 4.14: Decision evaluation: profits for pilot-scale experiment - linear case

| Case | R-LMIPA | R-Case |
|---|---|---|
| 1 | 1.6837 | 1.6837 |
| 2 | 1.7046 | 1.7046 |
| 3 | 2.8795 | 2.8795 |
| 4 | 2.8795 | 2.8795 |

Using the Bayesian method developed in this work, we obtain the *decision evaluation* shown in Table 4.14. The results are shown to be the same as the LMIPA results as previously discussed, for this linear case (R-Case). Also, as we would expect intuitively, the results show that with more degrees of freedom due to limits change, more profits can potentially be obtained from the process.

The *decision evaluation* results obtained and the associated optimum operating values for each of the four cases were then used to conduct four validation experiments to ascertain the practicability of the suggested constraint tunings and determine if the potential profits can be actualized in reality. As shown in Figure 4.8, there is a strong agreement between the calculated and experimentally achieved profits.

### b. Decision Making

When target profits of 1.7 and 2.5 are specified respectively, the algorithm determines the most probable combination of variables for limits change that will satisfy

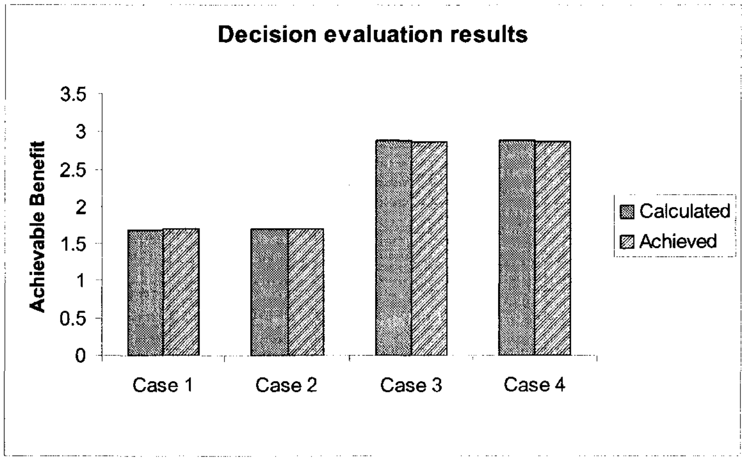



Figure 4.8: Experimental validation of *decision evaluation* results - linear case

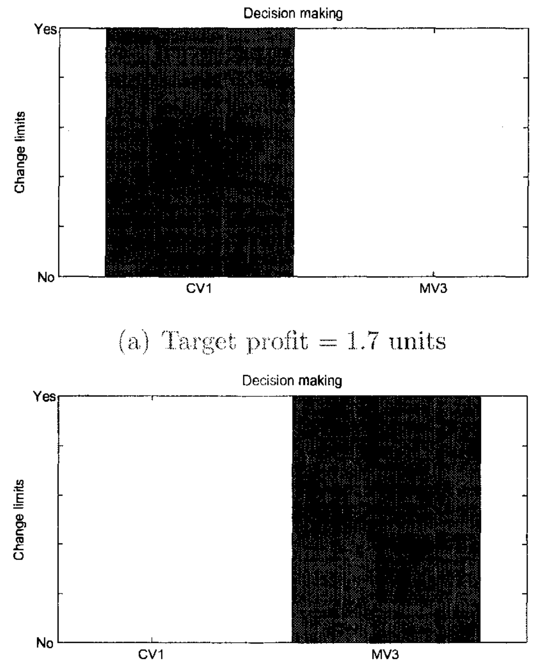
the target profits. Figure 4.9 shows the results for *decision making.* We have then implemented the suggested operation points and constraint changes in the experiment. Figure 4.10 shows that the *decision making* results have been experimentally evaluated. Note that there are some disagreements since the experimental profits are calculated as an averaged value while the decision making gives a point-wise profit. In addition, the maximum a posterior decision making only gives the most likely constraint tuning among a given set of tunings to achieve the desired profit target.

### 4.4.2 Quadratic objective function

Following the above specifications for the linear case, we can also illustrate the case of a quadratic economic objective function by specifying that $CV3$ has a linear coefficient value of $-100$ and quadratic coefficient value of 50. We also choose $MV1$ and $MV3$ for limits change, indicating four possible cases of potential profits.

#### a. Decision Evaluation

The results for *decision evaluation* are shown in Table 4.15 to compare the LMIPA and Bayesian-based approaches for the quadratic case. As shown in Figure 4.11, there is indeed profit to be obtained from constraint relaxation, even though the experimentally achieved profit is slightly less than that which was calculated.




(b) Target profit = 2.5 units

Figure 4.9: Decision making results for specified target profit for pilot-scale experiment-linear case

Table 4.15: Decision evaluation: profits for pilot-scale experiment - quadratic case

| Case | R-LMIPA | R-Case |
|---|---|---|
| 1 | 23.6001 | 23.8619 |
| 2 | 23.7438 | 24.0056 |
| 3 | 34.4680 | 34.7297 |
| 4 | 34.5975 | 34.8592 |

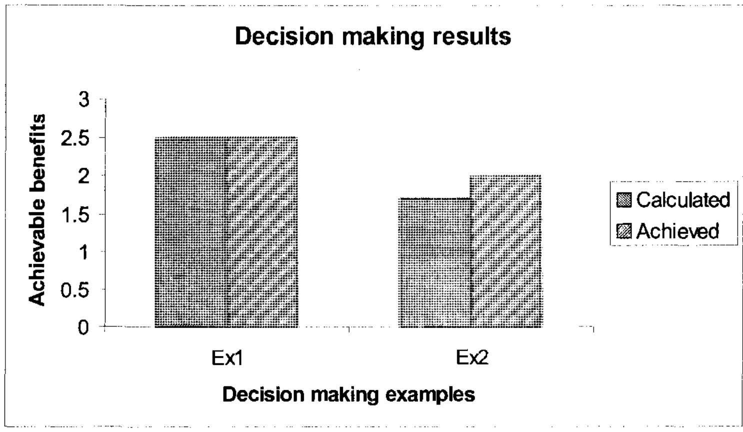



Figure 4.10: Experimental validation of *decision making* results - linear case

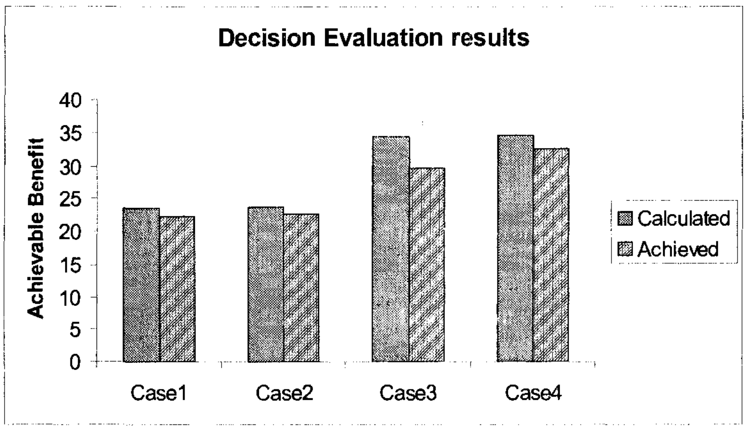
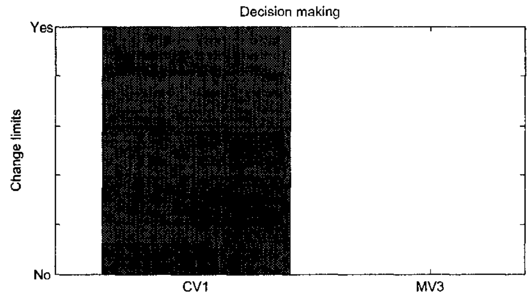


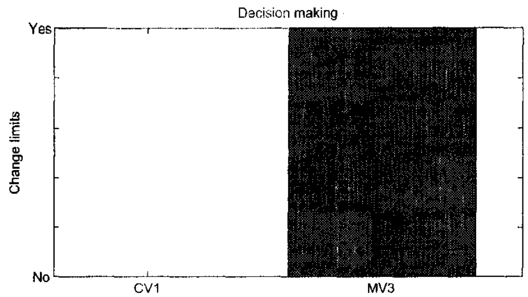

Figure 4.11: Experimental validation of *decision evaluation* results - quadratic case

## b. Decision Making

When target profits of 25 and 31 are specified respectively, the algorithm determines the most probable combination of variables for limits change that will satisfy the target profits. Figure 4.12 shows the *decision making* results and as shown in Figure 4.13, this result is reasonable given that the experimental profits are calculated as an averaged value while the decision making gives a point-wise profit, as previously noted.




(a) Target profit = 25 units




(b) Target profit = 31 units

Figure 4.12: Decision making results for specified target profit for pilot-scale experiment - quadratic case

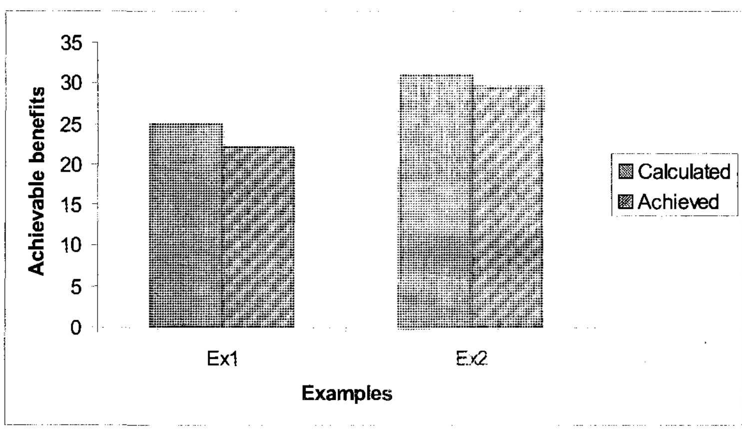



Figure 4.13: Experimental validation of *decision making* results - quadratic case

Table 4.16: Comparing features of discrete and continuous algorithms

| | **Discrete** | **Continuous** |
|---|---|---|
| Compared with LMIPA | Approximation | Mathematically accurate |
| Quality Variables | Only CVs considered for computational simplicity | ALL variables in economic objective function |
| Decision making approach | Discrete solution from Bayesian software | Direct solution through optimization |

## 4.5 Conclusion

A method based on the continuous-valued linear and quadratic considerations for the profit function has been developed. It provides the constraint analysis for the process variables of the MPC controller based on a Bayesian framework, and the profit that can be achieved based on limits change (*decision evaluation*) is calculated. For *decision making*, the profits are specified and corresponding decisions concerning variables for limits change are inferred from the optimization.

We have compared the results obtained using the approach proposed by Agarwal *et al* (2007)(Agarwal *et al.* 2007*b*, Agarwal *et al.* 2007*a*) and the results obtained using the proposed continuous-valued approach. The comparisons show that with this method, we can achieve calculations without loss of information due to discretization of profits, as in the case of the discrete algorithm. Table 4.16 summarizes the improvements of the continuous algorithm over the discrete version.

The benefit of using the Bayesian approach, as shown in this paper, includes the ability to handle uncertainties by dealing with the probability distributions associated with the quality variables. The illustrations provided indicate the usefulness of this method for constraint analysis and tuning during process operation and maintenance of MPC controllers. It is also applicable for accessing controller performance and evaluates the opportunity for improvement and increasing profit margins. Finally, the utility of this method has been evaluated using pilot-scale experiments.

# 5
# MPC constraint analysis-effect of dependent quality variables

## 5.1 Introduction

Since the advent of modern control algorithms and model-based controllers in the mid-seventies, the use of advanced process control (APC) strategies in industries has become crucial to the achievement of economic performance objectives. These concerns have generated a great deal of interest and research activity both in industry and academia. Among the available APC strategies, model predictive control (MPC) stands out as a key strategy for dealing with multivariable systems. Model predictive control (MPC) is a model-based algorithm which predicts the behaviour of the process outputs (controlled variables) based on the control action that minimizes a specified cost function (Qin and Badgwell 1997).

The MPC algorithm seeks to satisfy economic, safety, equipment, product quality or human preferences, or even a combination of any of the above criteria (Agarwal 2007, Backx *et al.* 2000). In addition to the MPCs objective of optimizing the future behaviour of a plant (Qin and Badgwell 1997), there is also the economic objective of an MPC controller. In a sense, since MPC is usually used at the supervisory control level (see Figure 1.1) (Qin and Badgwell 1997), the economic objective could be said to take priority over servo and regulatory objectives.

In the previous chapter, we have outlined a framework for estimating the effect of changes in constraint limits of process variables on the performance of an MPC thereby providing more insight for proper constraint analysis and tuning. In the quest to achieve optimal MPC economic performance, the following questions have been addressed: which constraints should be changed, and if constraint change is possible, what benefits can be obtained by doing so? These questions are answered by studying the relationship between variability and constraints within a Bayesian probabilistic framework. The analysis has been discussed as *decision evaluation* and *decision making* processes. Considering the profit as a continuous-valued function, we have shown that the analysis can be carried out without resorting to discretization.

In a the previous analysis, we assumed independence between the quality variables. This is a fundamental assumption. In reality, however, it is also possible for the quality variables to be dependent. In view of this, a question arises naturally: what happens when the quality variables are correlated or statistically dependent? Does the analysis break down or can it be extended to deal with this consideration? In this chapter, we address this problem using a multivariate projection method called factor analysis and we show that with certain modifications, we can still handle the MPC constraint issue within a Bayesian statistics framework.

## 5.2 Main Contributions

A fundamental assumption in the previous analysis was that the quality variables were independent. This assumption often holds but in reality the quality variables could be dependent. This issue is addressed in this chapter. In dealing with this, we use factor analysis to model the dependence between quality variables. The algorithm will be extended to deal with the cases where there are correlations between quality variables in the Bayesian network (controlled and/or manipulated Variables, i.e. all variables with non-zero linear and/or quadratic coefficients in the MPC economic objective function).

The decision evaluation results using the modified network (dealing with correlations among quality variables) will be shown to be the same as in the previous analysis, when the variables were assumed to be independent. This is so because the modified network deals with the correlations between the quality variables but the mean and variance, which are used to calculate the profit, remain the same.

## 5.3 Continuous-valued profit function-revisited

In Chapter 4, sections 4.3 and 4.3.1, we have introduced the mathematical formulations that were proposed for this analysis but the discussion was restricted to the case of assumed independence between the quality variables. In section 4.3 we showed via Proposition 1 that the *decision evaluation* formulation holds for both the cases when the quality variables are dependent and when they are independent. Therefore, in this section, we will focus on the *decision making* aspect (please refer to section 4.3.1 of Chapter 4 for more background information). When the quality variables are dependent, we consider the joint probability distribution of the variables and the following formulation applies. For clarity, we consider a case of two quality variables.

$$p(z_1, \dots, z_q) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(Z - \overline{Z})^T \Sigma^{-1} (Z - \overline{Z})\right) \tag{5.1}$$

where $Z$ is a vector of quality variables for optimization, $\overline{Z}$ is the vector of corresponding mean values, and $\Sigma$ is the corresponding covariance matrix.

$$(Z - \overline{Z}) = \begin{pmatrix} z_1 - \overline{z_1} \\ \vdots \\ z_q - \overline{z_q} \end{pmatrix} \text{ and } \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \sigma_{q2} & \cdots & \sigma_q^2 \end{bmatrix} \tag{5.2}$$

where $\Sigma$ can be estimated from the base case operation data. The objective function for this case can also be formulated as shown previously using log-likelihood:

$$\begin{aligned} &\max_{z_1, z_2} \left(-\frac{1}{2}(Z - \overline{Z})^T \Sigma^{-1} (Z - \overline{Z})\right) \text{ or equivalently} \\ &\min_{z_1, z_2} \left(\frac{1}{2}(Z - \overline{Z})^T \Sigma^{-1} (Z - \overline{Z})\right) \end{aligned} \tag{5.3}$$

subject to Equations 4.5, 4.6 and subject to the specified profit:

$$\sum_{i=1}^{2} (\alpha_i z_i + \beta_i^2 (\sigma_{z_i}^2 + \overline{z}_i^2 - 2\overline{z}\mu_i + \mu_i^2)) = R_T \tag{5.4}$$

$\alpha$ and $\beta$ represent each of the linear and quadratic coefficients, respectively, $\mu$ represents the specified target values, $q$ is the number of the quality variables (i.e. all the controlled and manipulated variables with non-zero linear and/or quadratic coefficients in the economic objective function).

## 5.4 Modeling Dependence between Quality Variables

Among many other issues, the effect of dependence between quality variables has led to many applications of multivariate projection methods for Statistical Process Control (SPC). These methods have great advantages when issues such as the dependence between quality variables, missing data, measurement errors, etc; arise (MacGregor and Kourti 1995, Kourti and MacGregor 1995, Kourti *et al.* 1996).

For our purpose, we will consider one of the multivariate projection methods: factor analysis (FA). When quality variables are dependent (or correlated), the location of one quality variable will be affected by those of the other correlated quality variable(s) and thus they cannot be arbitrarily shifted to their optimum values to achieve the desired objective of increasing the profitability of the plant. This relationship must be carefully considered in the algorithm to prevent the possibility of taking action based on wrong and/or misleading information. When variables are correlated, we use Equation 5.3, subject to Equation 5.4 and also subject to Equation 5.5.

$$L_{z_i} \leq z_i \leq H_{z_i} \tag{5.5}$$

$H_{z_i}$ is a vector of the upper limits for the quality variables, $L_{z_i}$ is a vector of the lower limits for the quality variables.

In dealing with this issue of dependence, via the use of *factor analysis*, the algorithm as described in Chapter 4 will be extended to deal with the cases where there is dependence (or correlations) between the quality variables in the Bayesian network.

### 5.4.1 Factor analysis

In this work, we refer to the common approach (orthogonal factor analysis) as Factor analysis. The aim of factor analysis (FA) is to obtain a number of underlying processes or factors that represent the correlation between variables in a data set. The results of FA form a new set of variables that are mutually independent of each other.

FA can be used directly to identify groups of inter-related variables, and thereby reduce the number of variables in a data set (dimension reduction); or indirectly to transform data to identify certain properties of interest which the original data may not have. FA can be used for exploratory or confirmatory purposes. Exploratory factor analysis seeks to identify underlying factors among variables while confirmatory factor analysis is used for hypothesis testing.

The objective of using FA is to transform the set of variables into a linear combination of the underlying components (factors). These factors are called common factors if they are associated with 2 or more of the original variables. They are called unique factors if they are associated with an individual variable. The relationship between the original variables and the factors is contained in the loadings. These loadings are in turn associated with the magnitude of the eigen values associated with the individual variables.

To properly interpret the factors, there is usually a rotation step which positions the factors in such a way that only the variables related to such factors will be associated with them (various orthogonal and oblique rotations are used in this step). In this work, we do not consider rotation of factors since our reason for using FA is different from the conventional uses (Wulder 2007, Tabachnick and Fidell 1989, Johnson and Wichern 1992, Moore *et al.* 1993).

The factor model postulates that $X$ is linearly dependent on a few unobservable random variables, $f_1, f_2, \cdots, f_m$, called common factors, and $p$ additional sources of variation $\epsilon_1, \epsilon_2, \cdots, \epsilon_p$ called errors or specific factors, where the observable random vector $X$ ( a $p \times 1$ vector with $p$ components), has mean $\mu$ and covariance matrix $\Sigma$ (Johnson and Wichern 1992):

$$X - \mu = \Lambda f + \epsilon \tag{5.6}$$

$\Lambda$ is the $p \times m$ matrix of factor loadings, $f$ is a $m \times 1$ vector of independent, standardized common factors, and $\epsilon$ is a $p \times 1$ noise vector of independent specific factors. Through this factor analysis, a set of correlated variables $X$ can be decomposed into a linear combination of a set of mutually uncorrelated variables $f$. Alternatively, the factor analysis model can be specified as

$$Cov(X) = \Lambda\Lambda^T + \Psi \tag{5.7}$$

where $\Psi = Cov(e)$ is a $p \times p$ matrix of specific variances.

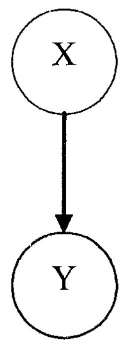
In this work we use the BNT toolbox (Murphy 2001, Murphy 2004*a*, Murphy 2004*b*) to perform the factor analysis between quality variables and thereby model dependence between them. In Figure 5.1(a), we consider factor analysis under a graphical model framework. $F$ is the factor node with a prior of $N(0, I)$ and $Y|F = f \sim N(\mu + \Lambda f, \Psi)$. $Y$ is the child node under consideration (in the context of this illustration, we will consider $Y$ as a multivariate node, $Y = [CV2 \quad MV1]^T$), $\Lambda$ is the factor loading matrix and $\Psi$ is the covariance of the noise or specific variances. Also, because the noise on both $F$ and $Y$ is diagonal, the components of these vectors are uncorrelated, and can then be represented as individual scalar nodes, as we show in Figure 5.1(b). Generally, we specify that the size of $F$ (say size of $F = k$) is less than the size of $Y$ (let size of $Y = D$). Thus, the factor analysis model seeks to explain $D$ observations using a k, lower-dimensional subspace (Murphy 2007). Specifying the prior probability of the factor node $F$ as $N(0, I)$ requires centering the data by deducting the mean values from the data . This is done for simplification purposes.

As shown in Figure 5.2, we can use factor analysis to create a new parent node $f$, which can be a multivariate node, depending on the number of factors to be considered. The use of factor analysis enables us to convert a set of correlated variables into a set of uncorrelated source variables. The factor loadings are used as weights or effects on the child nodes (quality variables). This procedure will subsequently be illustrated by an example.

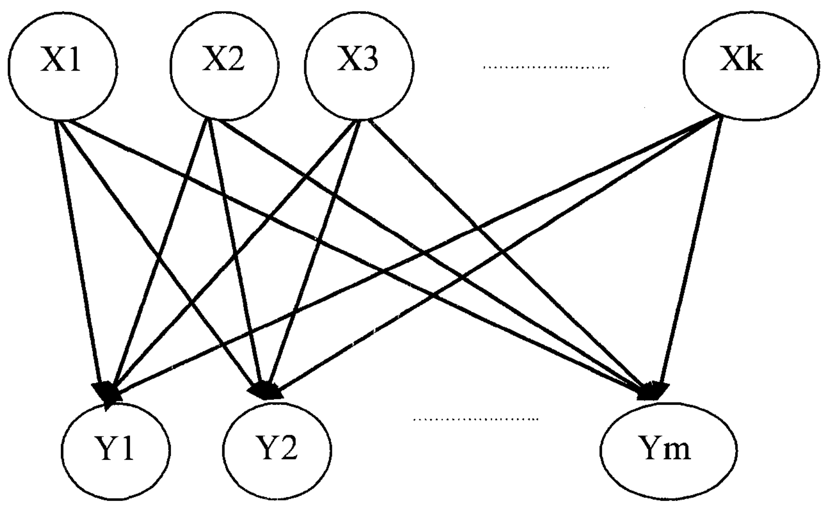
### 5.4.2 Illustrative example of a Bayesian network for MPC constraint analysis using factor analysis

The following example illustrates the procedure for building a Bayesian network to carry out the procedures for MPC constraint analysis as described previously. We consider a 3 input-3 output multivariable system with a steady state gain matrix as follows:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{5.8}$$

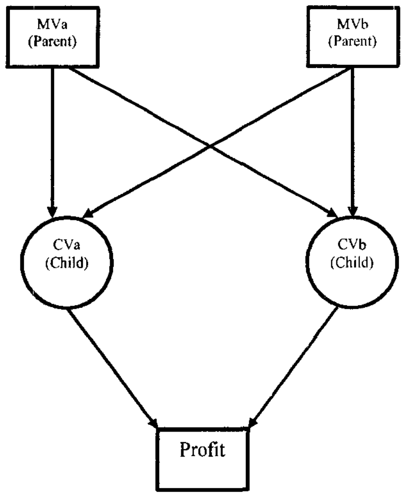



(a) Using multivariate nodes

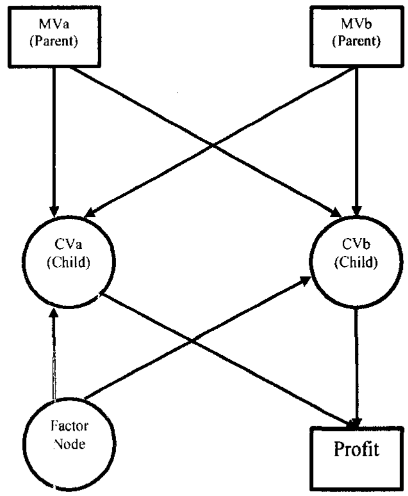



(b) Using univariate nodes

Figure 5.1: Factor analysis as graphical model




(a) Independent child nodes




(b) Dependent child nodes

Figure 5.2: Directed Acyclic Graphs with dependence modeled by a factor node

Table 5.1: Linear and quadratic coefficients of variables and specifications for limits change

| Variable | L Coeff | Q Coeff | Limits Change |
|---|---|---|---|
| **CV1** | 0 | 0 | No |
| **CV2** | $\alpha_{CV2}$ | 0 | No |
| **CV3** | 0 | 0 | Yes |
| **MV1** | $\alpha_{MV1}$ | 0 | No |
| **MV2** | 0 | 0 | Yes |
| **MV3** | 0 | 0 | No |

From Table 5.1, the variables in the MPC economic objective functions with non-zero linear and/or quadratic coefficients are CV2 and MV1; therefore, these are our quality variables. All the other variables could be considered as constraint variables with possibilities for limits change. For simplicity, we assume that only CV1 and MV2 are chosen for limits change.

In the following discussion we will differentiate between the process variable (CV or MV) and the logical variable (the decision whether to change limits or not) by labeling them as CV or MV and $CV^*$ or $MV^*$, respectively.

Since we have specified two variables for limits change, the Bayesian network will have two parent nodes ($CV1^*$ and $MV2^*$). Also, since we have two variables with

Table 5.2: Specifications for parent and child nodes

| **Parent Nodes** | **CV3** | **MV2** |
|---|---|---|
| **States** | (Yes, No) | (Yes, No) |
| **Child Nodes** | **CV2** | **MV1** |
| **States** | $(\mu, \sigma^2)$ | $(\mu, \sigma^2)$ |

Table 5.3: Different cases of limits change combinations

| **Parent Nodes** | **Case 1** | **Case 2** | **Case 3** | **Case 4** |
|---|---|---|---|---|
| $CV1^*$ | No | Yes | No | Yes |
| $MV2^*$ | No | No | Yes | Yes |

non-zero linear and quadratic coefficients, we have two child nodes for the network (CV2 and MV1). Table 5.2 shows the specifications for the parent and child nodes. The conditional probability distributions (CPD) of the nodes are thus defined. We assume that the prior probabilities of each of the parent nodes is 0.5, implying that there is an equal probability that we will make a change in the limits or leave the limits unchanged.

For this system, we will have $2^2 = 4$ (see Table 5.3) combinations of limits change for which we can carry out optimizations to obtain the optimum values of the quality variables based on our choice of limits change. The CPD's of the child nodes, being continuous, are defined by mean values and variances for each node. The mean values of these child nodes are the optimum values obtained from the optimization step and the variances are the base case variances of the quality variables, obtained from data.

With the CV2 and MV1 identified as quality variables and their CPD specified as discussed above, with the CPT for $CV1^*$ and $MV2^*$ as parent nodes given as 0.5 for each of their prior probabilities, we have a Bayesian network as shown in Figure 5.3, which can subsequently be used for *decision making* and *decision evaluation.*

**Decision Evaluation:** Here we supply our decision to change or not change limits as evidence to the network. This evidence is used to estimate the probabilities of the child nodes to have a particular mean value, based on its distribution. This estimated mean value is then used to calculate the expected cost, and subsequently, the potential profit that can be obtained from the process. The expected cost is calculated using Equation 4.14 as follows:

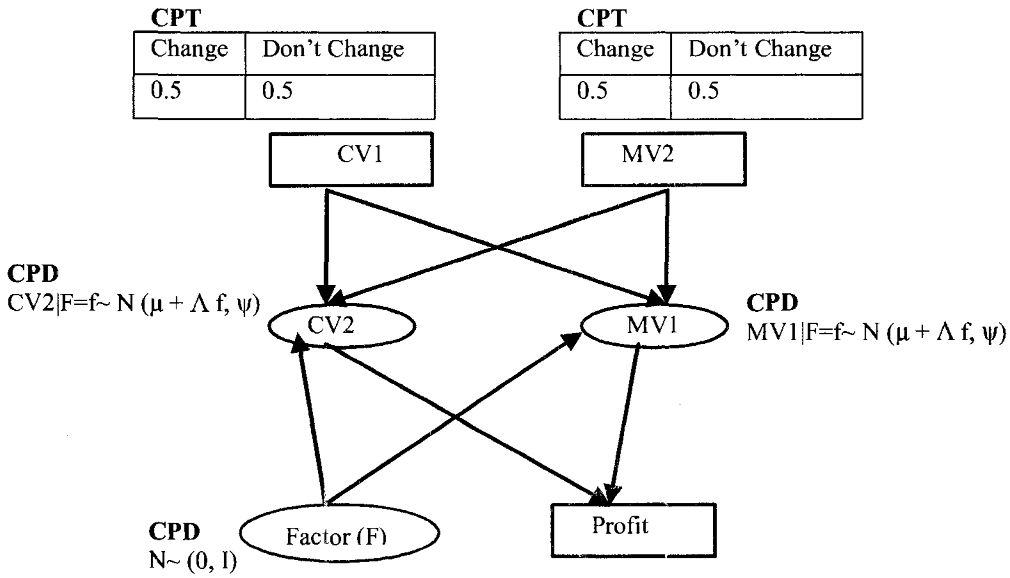



Figure 5.3: Bayesian network with factor node for illustrative example

$$E(F) = \alpha_{CV2}\overline{CV}2 + \alpha_{MV1}\overline{MV}1 \tag{5.9}$$

**Decision Making:** In this case, a specified value for the profit is provided to the algorithm. Using Equation 5.3 we have the objective function for optimization as:

$$\min_{CV2,MV1} \left( \frac{1}{2}(Z - \overline{Z})^T \Sigma^{-1} (Z - \overline{Z}) \right) \tag{5.10}$$

subject to

$$L_{CV2} \leq \overline{CV}2 \leq H_{CV2} \tag{5.11}$$

$$L_{MV1} \leq \overline{MV}1 \leq H_{MV1} \tag{5.12}$$

$$\alpha_{CV2}\overline{CV}2 + \alpha_{MV1}\overline{MV}1 = R_T \tag{5.13}$$

where $R_T$ is the specified returns, $L_{CV2}$ and $L_{MV1}$ are the lower limits for CV2 and MV1, $H_{CV2}$ and $H_{MV1}$ are the upper limits for CV2 and MV1, respectively, $\Sigma$ is the covariance matrix as shown below, and $\alpha_{CV_2}$ and $\alpha_{MV1}$ are the linear coefficients of the economic objective function. $Y = \begin{bmatrix} CV2 \\ MV1 \end{bmatrix}$, $\overline{Y} = \begin{bmatrix} \overline{CV}2 \\ \overline{MV}1 \end{bmatrix}$, and

$$\Sigma = \begin{bmatrix} \sigma^2_{CV2} & \sigma_{CV2MV1} \\ \sigma_{MV1CV2} & \sigma^2_{MV1} \end{bmatrix}$$

The optimum values for CV2 and MV1 thus obtained from optimization are then supplied as evidence to the network. The network then determines the most probable explanation of this evidence based on the probabilities of the four available combinations of limits change and the associated distributions of CV2 and MV1 previously provided to the network.

## 5.5 Experimental validation

The proposed approach was implemented on the pilot scale process described in Chapter 2, section 2.2.2. The MPC design parameters are as previously described in Chapter 4, section 4.4.1. Since we have previously illustrated that the *decision evaluation* results are obtained by the same formulation for both the dependent and independent cases(see section 4.3), we will focus on the experimental validation of the *decision making* results.

### 5.5.1 Decision making

As previously explained, this refers to obtaining the maximum *a posteriori* explanation for *decision making* that will help to achieve a target value of profit. For *decision making* purposes the target expected return (or profit) is provided and the corresponding optimum values for the quality variables affecting the benefit function are estimated.

The quality variables can be checked for significant correlations between them as using Matlab (MathWorks 2007*a*): $[R, P] = corrcoef(X)$ returns P, a matrix of p-values for testing the hypothesis of no correlation. Each p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero.

For this analysis, we consider the case where the liquid levels in all three tanks (CV1, CV2, CV3) are the quality variables. Equation 5.14 shows the corresponding correlation matrix. Since there is some reasonable correlation between the quality

Table 5.4: Decision making - suggested operating points

| Profit = 9.5 | | |
|---|---|---|
| Quality variable | Dependence | Independence |
| CV1 | 0.2337 | 0.2331 |
| CV2 | 0.1802 | 0.1793 |
| CV3 | 0.2142 | 0.2132 |

| Profit = 11 | | |
|---|---|---|
| Quality variable | Dependence | Independence |
| CV1 | 0.2337 | 0.2331 |
| CV2 | 0.1802 | 0.1793 |
| CV3 | 0.2292 | 0.2282 |

variables, we can use the proposed algorithm to accommodate the dependence.

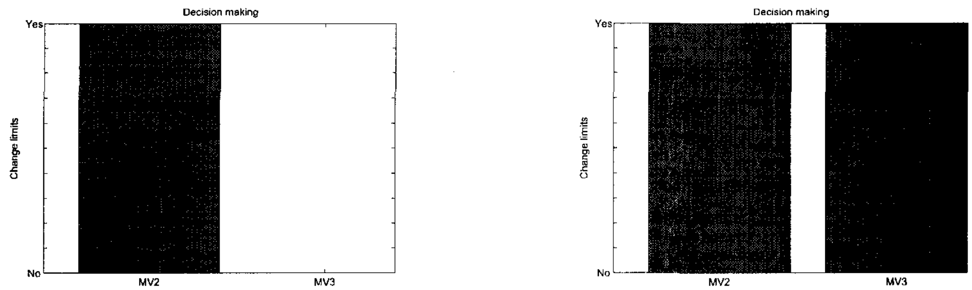
$$\begin{bmatrix} 1 & 0.354055 & 0.109908 \\ 0.354055 & 1 & 0.256687 \\ 0.109908 & 0.256687 & 1 \end{bmatrix} \tag{5.14}$$

### 5.5.2 Results

#### Linear function

For the linear case, we specify the linear coefficients to be $-100$ for all three quality variables. The variables considered for constraint limits change are MV2 and MV3. Based on this information and following the procedures set forth in Chapter 4, the decision evaluation step can be performed. The decision evaluation results remain the same even when the variables are correlated.

To obtain profits of 9.5 and 11 units, respectively, the *decision making* results are shown in Figure 5.4. Figure 5.5 shows that it is indeed possible to obtain more benefit, based on the *decision making* results obtained. It was observed that the same *decision making* results were obtained when independence between quality variables was assumed in the algorithm; however when the actual suggested operating points are checked, we observe that the two algorithms actually suggest different operating points for the quality variables. Both sets of suggested values can be achieved by changing the limits of MV2, which in turn allows CV3 to shift towards its optimum operating point to achieve the desired profit, as shown in Figure 5.4.




(a) Target profit = 9.5 units (b) Target profit = 11 units

Figure 5.4: Decision making results with dependence modeled-linear case

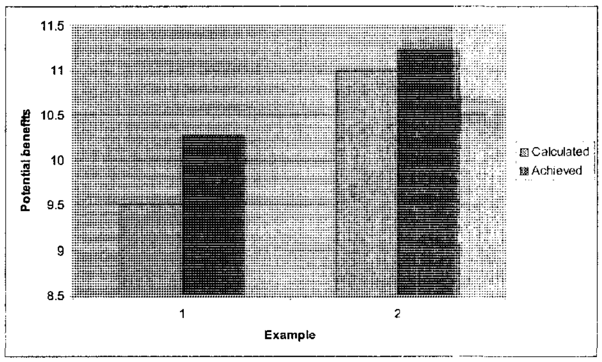
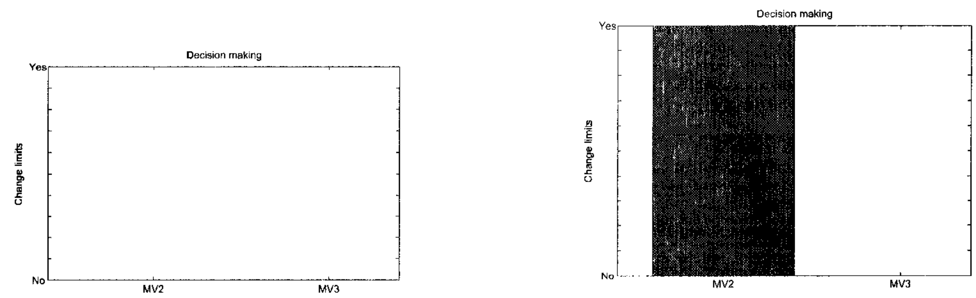



Figure 5.5: Experimental validation of *decision making* results-linear case

### Quadratic function

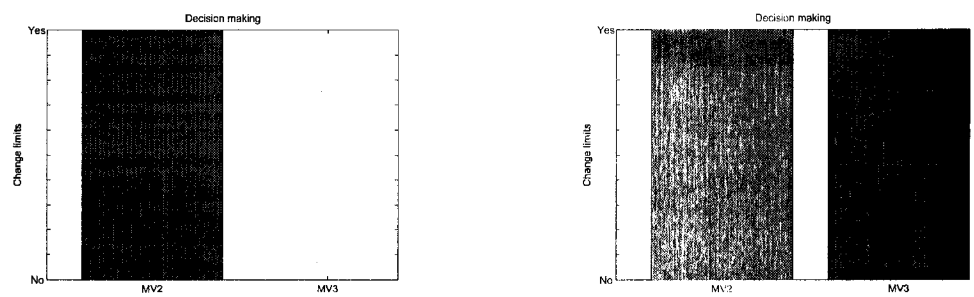
Similarly, for the quadratic case, we specify the linear and quadratic coefficients to be −100 and 50, respectively, for all the quality variables. When target profits are specified as 43 and 52 units, respectively, the *decision making* results are shown in Figures 5.6 and 5.7. Figure 5.6 shows the results when the dependence is not modeled while Figure 5.7 shows the results with the dependence modeled. Figure 5.8 also shows that the *decision making* results obtained, are indeed feasible. The difference between using the algorithm proposed in Chapter 4 and this algorithm is shown in Figures 5.6 and 5.7. We observe that when the dependence between the quality variables is modeled, the results obtained (as shown in Figures 5.7(a) and 5.7(b)) suggest that more variables should have their constraint limits changed, than in the case where independence is assumed (see Figures 5.6(a) and 5.6(b)).

From the experimental results, we also observe that in general, there is better




(a) Target profit = 43 units (b) Target profit = 52 units

Figure 5.6: Decision making results without dependence modeled-quadratic case




(a) Target profit = 43 units (b) Target profit = 52 units

Figure 5.7: Decision making results with dependence modeled-quadratic case

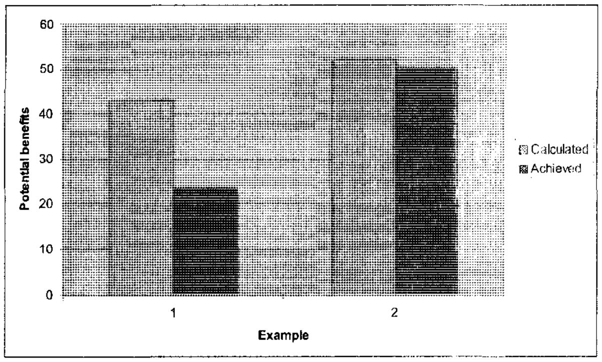



Figure 5.8: Experimental validation of *decision making* results-quadratic case

agreement with the calculated results when the limits of the two constrained variables are changed, rather than when only one process variable has its constraint limits changed.

## 5.6 Conclusion

In this chapter, we have shown the importance of, and provided a framework for modeling the dependence between quality variables in the Bayesian analysis. The importance of determining and modeling the dependence between quality variables for Bayesian analysis has been illustrated. This tool assists the user in making decisions and provides a new level of understanding of the relationship between the quality variables and the variables chosen for limits change.

The results obtained by using the modified network (dealing with correlations among quality variables) are the same as when the quality variables were assumed to be independent, when used for *decision evaluation*. This is because the modified network deals with the correlation between the variables but the mean and variance, which are used to calculate the profits for the *decision evaluation* step in both cases, remain the same. The importance of the modified algorithm is shown when we consider *decision making* for a case where there is strong dependence between the quality variables, has been illustrated.

# 6 Conclusions

This thesis has proposed new approaches for controller performance assessment based on the use of non-parametric statistics and Bayesian analysis. We have proposed some methods based on *run lengths* and more importantly, based on the previously proposed work by Xu *et al* (2007) and Agarwal (2007), we have developed a Bayesian-based framework that increases the functionalities of the previous algorithms.

In Chapter 2 of the thesis we reviewed the use of run length distributions for controller performance assessment and proposed new approaches based on the use of Markov chains. The proposed methods were discussed with the aid of illustrative examples. The use of Markov chains provided an approach that retained the intuitive appeal of the use of run lengths while at the same time also providing a theoretical background that is amenable to analysis. In Chapter 3, we developed an algorithm based on the Bayesian network that provides a new framework that serves as a tuning library which is useful for decision making purposes.

In Chapters 4 and 5 of this thesis, an approach for the constraint analysis and tuning of Model Predictive Controllers (MPC) using Bayesian networks (Murphy 2007, Murphy 2004*a*, Murphy 2004*b*, Murphy 2001), has been proposed as an extension of previous work in this area (Agarwal 2007, Xu *et al.* 2007). We have addressed this problem using a continuous valued objective function and have provided a framework for considering the effect of dependence between quality variables as discussed in Chapter 5.

Using a continuous valued function has allowed us to consider all the variables in the objective function as quality variables (without being limited to the situation where only the output variables were being used as quality variables), without loss of information due to discretization. We have also proposed a methodology that covers both linear and quadratic objective functions. Two case studies (a simulated binary distillation column and an pilot scale process) have been considered to illustrate the proposed approaches.

The results obtained using the continuous valued function has been shown to be the same as those of the LMIPA for the linear case. For the quadratic case, the formulation of the objective function takes the variance of the quality variables into consideration and the presence of the variances in the formulation constitutes the difference between the LMIPA and the Bayesian-based algorithm using the continuous valued function.

In Chapter 5, we have considered the special case where the quality variables are dependent and we have illustrated how this affects the decisions that can be made from the analysis. The dependence was modeled using factor analysis within a Bayesian statistics framework. We have shown that the presence of dependence results in more conservative decisions than in the case where there was no dependence. This is because the optimum value of each quality variable can not be easily attained when it is statistically dependent on the optimal values of other quality variables.

It is expected that the effect of dependence between quality variables will be even more useful when the decisions as to change constraint limits are further investigated. In the present work, we have considered the decision to change or not to change the limits *by* 5%. This implies that the parent nodes have only two states (*yes or no*). However, if the limits change were considered *up to* 5%, then the parent nodes would be continuous nodes with a range of 0% to 5% and the effect of dependence should be more clearly seen. This work has provided the framework for such possible further investigation.

Finally, in appendix A, we have provided some guidelines on how to use the BNT toolbox (Murphy 2007, Murphy 2004*a*, Murphy 2004*b*, Murphy 2001). We have included sample codes that explain how we have used the toolbox in this thesis. The chapter is a simple tutorial that complements the tutorial in the BNT toolbox.

# Bibliography


Agarwal, N. (2007). Probabilistic approach for mpc performance assessment. Master's thesis. University of Alberta, Edmonton, Canada.

Agarwal, N., B. Huang and E.C. Tamayo (2007*a*). Assessing model prediction control (mpc) performance. 1. probabilistic approach for constraint analysis. *Industrial & Engineering Chemistry Research* **46**(24), 8101–8111.

Agarwal, N., B. Huang and E.C. Tamayo (2007*b*). Assessing model prediction control (mpc) performance. 2. bayesian approach for constraint tuning. *Industrial & Engineering Chemistry Research* **46**(24), 8112–8119.

Backx, T., O. Bosgra and W. Marquardt (2000). Integration of model predictive control and optimization of processes. *Proceedings ADCHEM 2000* **1**, 249–260.

Balakrishnan, N. and M.V. Koutras (2002). *Runs and Scans with Applications.* J. Wiley New York.

Bauer, M. and I.K. Craig (2007). Economic assessment of advanced process control A survey and framework. *Journal of Process Control* **18**(1), 2–8.

Bemporad, A., M. Morari and N.L. Ricker (2007). Model Predictive Control Toolbox user's guide. *The MathWorks Inc., Natick, MA.*

Bishop, G. and G. Welch (2003). An Introduction to the Kalman Filter. *Department of Computer Science, University of North Carolina at Chapel Hill.*

Brook, D. and D.A. Evans (2003). An approach to the probability distribution of cusum run length. *Biometrika* **59**(3), 539–549.

Chandra, Anshuman and Krishnendu Chakrabarty (2001). Frequency-directed run-length (fdr) codes with application to system-on-a-chip test data compression. In: *VTS '01: Proceedings of the 19th IEEE VLSI Test Symposium.* IEEE Computer Society. Washington, DC, USA. p. 42.

Fatemi, M.; Kelarestaghi, M. (1992). Application of run-length coding in reprojection. *Nuclear Science Symposium and Medical Imaging Conference, 1992., Conference Record of the 1992 IEEE* pp. 1166–1168 vol.2.

Galloway, M.M. (1974). Texture analysis using grey level run lengths. *Unknown.*

Harris, T.J. (1989). Assessment of control loop performance. *Canadian Journal of Chemical Engineering* **67**, 856–861.

Harris, T.J. and W. Yu (2003). Analysis of multivariable controllers using degree of freedom data. *International Journal of Adaptive Control and Signal Processing* **17**(7-9), 569–588.

Harris, T.J., C.T. Seppala and L.D. Desborough (1999). A review of performance monitoring and assessment techniques for univariate and multivariate control systems. *Journal of Process Control* **9**(1), 1–17.

Huang, B. and S.L. Shah (1999). *Performance assessment of control loops: Theory and applications.* Springer-Verlag, London.

Inteco (2006). Multitank system user's manual. *Inteco Ltd Poland, www.inteco.com.pl.*

Jelali, M. (2006). An overview of control performance assessment technology and industrial applications. *Control Engineering Practice* **14**(5), 441–466.

Johnson, R.A. and D.W. Wichern (1992). *Applied multivariate statistical analysis.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Kemeny, J.G. and J.L. Snell (1976). *Finite Markov Chains.* Springer Verlag.

Korb, K.B. and A.E. Nicholson (2004). *Bayesian Artificial Intelligence.* Chapman & Hall/CRC.

Kourti, T. and J.F. MacGregor (1995). Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems* **28**(1), 3–21.

Kourti, T., J. Lee and J.F. Macgregor (1996). Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering* **20**, 745–750.

Latour, P.H., J.H. Sharpe and M.C. Delaney (1986). Estimating benefits from advanced control. *ISA Transactions* **25**(4), 13–21.

Li, Q., J.R. Whiteley and R.R. Rhinehart (2004). An automated performance monitor for process controllers. *Control Engineering Practice* **12**(5), 537–553.

Ljung, L. (2007). MATLAB System Identification Toolbox: User's Guide. *The Mathworks, Inc., Nantucket, MA.*

Lofberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD Conference.* Taipei, Taiwan.

Lu, S. (2007). Multivariate process and control monitoring - practical approaches and algorithms. Master's thesis. University of Alberta. Edmonton, Canada.

MacGregor, J.F. and T. Kourti (1995). Statistical process control of multivariate processes. *Control Engineering Practice* **3**(3), 403–414.

Martin, G.D., L.E. Turpin and R.P. Cline (1991). Estimating control function benefits. *Hydrocarbon Processing* **June**, 68–73.

MathWorks (2007*a*). MATLAB User's Guide. *The MathWorks Inc., Natick, MA.*

MathWorks (2007*b*). Optimization Toolbox user's guide. *The MathWorks Inc., Natick, MA.*

MathWorks (2007*c*). Real Time Workshop Toolbox user's guide. *The MathWorks Inc., Natick, MA.*

Moore, D.S., G.P. McCabe and M.J. Evans (1993). *Introduction to the Practice of Statistics.* WH Freeman & Co. New York, NY, USA.

MSBN (2007). http://www.research.microsoft.com/adapt/MSBNx/msbnx/Basics_of_Bayesian_Inference.htm (Accessed November 2007).

Murphy (2007). http://www.cs.ubc.ca/ murphyk/Software/BNT/usage.html#pca (Accessed November 2007).

Murphy, K.P. (2001). The Bayes Net Toolbox for MATLAB. *Computing Science and Statistics* **33**, 331–351.

Murphy, K.P. (2004*a*). The bayes net toolbox for matlab. Technical report. Department of Computer Science,University of California, Berkeley.

Murphy, K.P. (2004*b*). How to use bayes net toolbox.

Ogunnaike, B.A. and W.H. Ray (1994). *Process dynamics, modeling, and control.* Oxford University Press New York.

Owusu, S.O. and R.R. Rhinehart (2004). A controller performance monitor. *American Control Conference, 2004. Proceedings of the 2004* **6**, 5046–5051.

Owusu, S.O., Q. Li and R.R. Rhinehart (2005). Run length-based control performance monitors: a comparison between two techniques. *American Control Conference, 2005. Proceedings of the 2005* **2**, 1042–1047.

Paulonis, M.A. and J.W. Cox (2003). A practical approach for large-scale controller performance assessment, diagnosis, and improvement. *Journal of Process Control* **13**(2), 155–168.

Qin, S.J. and J. Yu (2007). Recent developments in multivariable controller performance monitoring. *Journal of Process Control* **17**(3), 221–227.

Qin, S.J. and T.A. Badgwell (1997). An overview of industrial model predictive control technology. *Fifth International Conference on Chemical Process Control* **93**, 232–256.

Robinson, P.B. and T.Y. Ho (1978). Average Run Lengths of Geometric Moving Average Charts by Numerical Methods. *Technometrics* **20**(1), 85–93.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp (2003). *Process dynamics and control.* Wiley Hoboken, NJ.

Seneta, E. (1973). *Non-Negative Matrices: An Introduction to Theory and Applications.*. Allen & Unwin: London,.

Singh, P. and K. Seto (2002). Analyzing APC performance. *Chemical engineering progress* **98**(8), 60–66.

Sturm, J.F. (1999). Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* **11**(1), 625–653.

Tabachnick, B.G. and L.S. Fidell (1989). *Using multivariate statistics.* Allyn and Bacon Boston, MA.

Thornhill, N.F., M. Oettinger and P. Fedenczuk (1999). Refinery-wide control loop performance assessment. *Journal of Process Control* **9**(2), 109–124.

Toh, K.C., M.J. Todd and R.H. Tutuncu (1999). SDPT3a MATLAB software package for semidefinite programming. *Optimization Methods and Software* **11**(12), 545–581.

Tsai, M.J., J.D. Villasenor and F. Chen (1996). Stack-run image coding. *Circuits and Systems for Video Technology, IEEE Transactions on* **6**(5), 519–521.

Volk, U., D.W. Kniese, R. Hahn, R. Haber and U. Schmitz (2005). Optimized multivariable predictive control of an industrial distillation column considering hard and soft constraints. *Control Engineering Practice* **13**(7), 913–927.

Wackerly, D.D., W. Mendenhall and R.L. Scheaffer (2002). *Mathematical statistics with applications.* Duxbury Pacific Grove, CA.

Wardell, D.G., H. Moskowitz and R.D. Plante (1994). Run-Length Distributions of Special-Cause Control Charts for Correlated Processes. *Technometrics* **36**(1), 3–17.

Wikipedia (2007). http://en.wikipedia.org/wiki/Bayesian_analysis (Accessed November 2007).

Wulder (2007). http://www.pfc.forestry.ca/profiles/wulder/mvstats/pca_fa_e.html (Accessed November 2007).

Xu, F., B. Huang and S. Akande (2007). Performance assessment of model predictive control for variability and constraint tuning. *Industrial & Engineering Chemistry Research* **46**(4), 1208–1219.

# A

# Exploring the Bayesian Network Toolbox

## A.1 Introduction

This chapter focuses on providing a step-by-step overview of the various possibilities that exist using the Bayesian network toolbox (BNT) developed by Murphy (Murphy 2001, Murphy 2004*a*, Murphy 2004*b*, Murphy 2007). We will restrict ourselves mainly to the applications of the toolbox that have been used in the previous chapters of this thesis. A complete tutorial on how to use the BNT toolbox can be found on the Murphy's webpage (Murphy 2007). In the following sections, we will illustrate the process of building a Bayesian network, defining the nodes and their parameters. We will also provide examples of how to use the BNT toolbox for Factor analysis (or Principal component analysis). Since there is a scarcity of references on using the BNT toolbox, we will draw extensively from the examples provided by Murphy (Murphy 2007), but we will adapt them to our specific problems and explain how the toolbox could be use for similar problems. The structure of this chapter is more of a step by step tutorial on how to use the BNT toolbox, than a well-structured paper article. All the commands provided in this section are given as they would be written in the MATLAB environment.

## A.2 Building a Bayesian network

The instructions for downloading and installing the BNT toolbox are available at http://www.cs.ubc.ca/murphyk/Software/BNT/bnt.html (this link can also be obtained by typing "bnt toolbox" in google or any search engine). The tutorial available on the website indicated is, in itself, very detailed. What we have set out to do here is to illustrate our use of the BNT toolbox in the approach and algorithm discussed in this thesis.

### A.2.1 Illustrative example

In this section, we examine the procedure for building and parameterizing a Bayesian network for an MPC process with two variables chosen for limits change and two quality variables that affect the potential profit that can be obtained from the process. A directed acyclic graph (DAG) for this process is shown in Figure A.1. If the analysis was for changing the limits of N variables and there were Q quality variables, we would use Figure A.2 for the analysis.

To build a Bayesian network for the example shown in Figure A.1, we need to specify the following commands:

```

nNode = nParent + nChild;
dag = zeros(nNode);
for  i = 1:nParent,
     for  j = 1:nChild,
     dag(i,nParent + j) = 1;
     end
end
```

Where *nParent* and *nChild* are the number of parent and child nodes respectively. In the above segment of code, we have used *for* loops for simplicity but essentially in defining the DAG, we need to put a *zero* where there is no connecting arc between

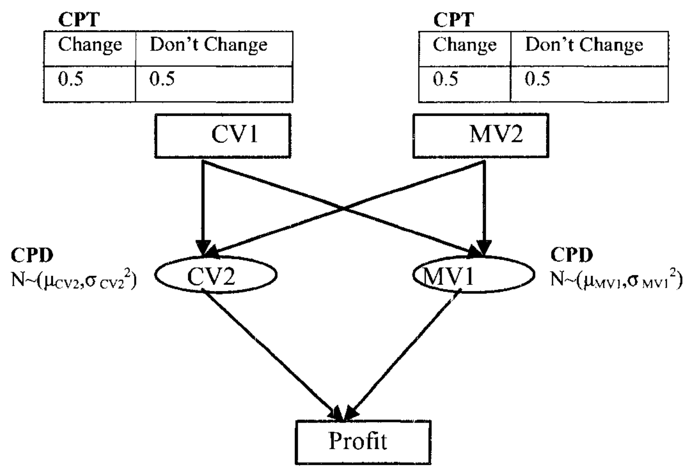



Figure A.1: Bayesian network for illustrative example

nodes and a *one* where there is a connecting arc between nodes. Specifying the values of *nParent* and *nChild* differentiates between figures A.1 and A.2.

### A.2.2 Parameterizing and creating the Bayesian network

To parameterize the network, we need to specify the conditional probability distributions of each node. The BNT toolbox can support different types of conditional probability distributions such as (Murphy 2007):

- Tabular (multinomial)
- Gaussian
- Softmax (logistic/ sigmoid)
- Multi-layer perceptron (neural network)
- Noisy-or
- Deterministic

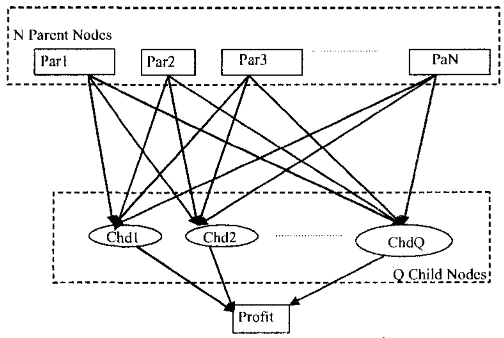



Figure A.2: Directed Acyclic Graphs (DAG) for illustrative example

In this work, we have used only tabular and gaussian conditional probability distributions for discrete and continuous nodes respectively. As shown in Figure A.1, the parent nodes (CV1 and MV2) have two states each (discrete nodes) while the child nodes (CV2 and MV1) have continuous distributions (continuous nodes). The conditional probability distributions for the parent and child nodes are specified using tabular and Gaussian distributions, respectively, as follows:

```

    ParentNodeSizes = 2*ones(1,nParent);
    % this indicates that each parent node has two states
    ChildrenNodeSizes = ones(1,nChild);
    % this indicates that each child node is univariate
    NodeSizes = [ParentNodeSizes  ChildrenNodeSizes];
    Net=mk_bnet(dag, NodeSizes,'discrete',[1:ParentNumber]);
    % the function mk_bnet builds the Bayesian Network
    % Next, parent nodes are specified as discrete nodes
    % and the child nodes are specified as continuous nodes:
     for i=1:size(dag,1)
      if  i ≤ ParentNumber
          CPT=[0.5  0.5];
          Net.CPD{i}=tabular_CPD(Net,i,CPT);
          else
             Net.CPD{i}=gaussian_CPD(Net,i,'mean',...
             mu(:,i-ParentNumber),'cov',cov(:,i-ParentNumber));
          end
     end
```

The above segment of code specifies the conditional probability table (CPT) of the parent nodes and the conditional probability distributions (CPD) of the child nodes (continuous nodes). The probabilities are specified in the CPT while the distributions (means and variances/covariances) are specified in the CPD. *"mu"* is a vector with as many elements as the states of the parent nodes. *"cov"* is the covariance matrix (or the variance vector) with corresponding elements to the mean vector. The Bayesian network is now ready to be used for inference and analysis.

### A.2.3 Supplying evidence and inference

One-dimensional cell arrays are used to specify evidence supplied to the network. This allows each cell to accommodate vectors of different lengths, particularly when there are multivariate nodes present (Murphy 2007).

```

    % first we specify the inference engine
    engine = jtree_inf_engine(bnet);
    % then we define the cell array for evidence
    evidence = cell(1,nNode);
    % We specify evidence stating do not change limits for...
    % parent node 1 (CV1)
    evidence{1} = 1;
    % We specify evidence stating change limits for parent...
    % node 2 (MV2)
    evidence{2} = 2;
    [engine, loglik] = enter_evidence(engine, evidence);
    % We can infer the updated CPD based on the evidence ...
    % supplied above
    for i = 1:nChild,
    % mean values for each child node is obtained as follows:
         Marg_Dist = marginal_nodes(Engine,nParent+i);
        mu(i)=Marg_Dist.mu;
        % other parameters can be obtained likewise
    end
```

## A.3 Factor analysis using the Bayesian network

In this section, we will carry out factor analysis using the Bayesian network. We will illustrate the procedure using a multivariate child node in this example, this is for illustrative purposes to show how multivariate nodes are defined and used. The DAG under consideration is shown in Figure A.3. Since each child node has both discrete and continuous parents, the CPD of each child node is defined as $CV2|X = x \sim N(\mu + \Lambda x, \Psi)$ and $MV1|X = x \sim N(\mu + \Lambda x, \Psi)$ . This has been discussed in Chapter 5.

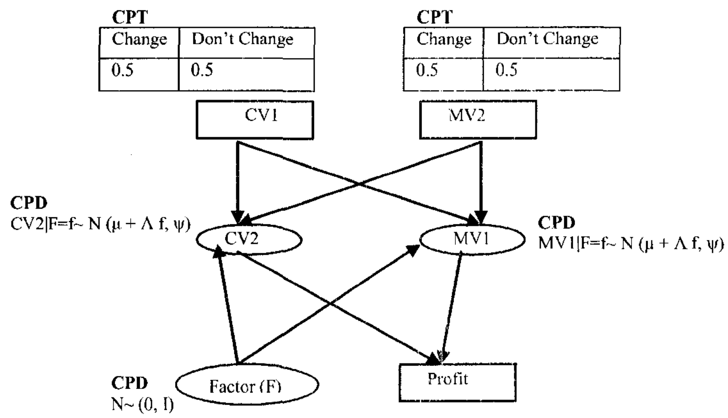



Figure A.3: DAG with dependence modeled using factor node

```

load Data; % data of quality variables
K=2;% Number of factors to be used
M=2;% Number of states of the discrete nodes
D = size(Data, 2);% Number of quality variables
N = size(Data, 1);% Size of data set
nodeSizes = [M M K D];
% each parent node has M states, K factors are used and...
% the child node is multivariate with 5 variables
dag = zeros(length(nodeSizes));
% again we build the DAG we need the Lambda matrix...
% as weights for the network
[L1, Psi1, LL1] = ffa(Data,K,max_iter);
% this function performs the factor analysis
% LL1 is the Λ matrix and Psi1 is the Ψ matrix
```

```

    for  i=1:length(nodeSizes)-1
        dag(i,length(nodeSizes))=1;
    end
    nNode = length(nodeSizes);

    Net=mk_bnet(dag, nodeSizes,'discrete',[1:nParent]);

    for  i=1:nParent
        CPT=[0.5 0.5];
      Net.CPD{i}=tabular_CPD(Net,i,CPT);
    end
    i=length(nodeSizes);
    Net.CPD{i-1}=gaussian_CPD(Net,i-1,'mean', zeros(K, 1),...
     'cov', eye(K),'cov_type', 'diag','cov_prior_weight', 0,...
     'clamp_mean', 1, 'clamp_cov', 1);
    Net.CPD{i}=gaussian_CPD(Net,i,'mean', Mean1, 'cov',...
     repmat(diag(Psi1),[1 1 nParent*M]),'weights', L11,...
     'cov_type', 'diag', 'cov_prior_weight', 0, 'tied_cov', 1);
```

In this section, we have provided an overview of the BNT toolbox and we have illustrated the procedure for building a Bayesian network for various forms of analysis by providing snippets of MATLAB code. Examples have been included to show how the network can be used to make inference. We have also illustrated that the network can be used for a statistical procedure like factor analysis.

Other uses and applications of the BNT toolbox have been shown by Murphy (Murphy 2001, Murphy 2004*a*, Murphy 2004*b*, Murphy 2007), but we have restricted our examples to the scope of the algorithms included in the previous chapters of this thesis.