



2810642743



REFERENCE ONLY

UNIVERSITY OF LONDON THESIS

Degree *Ph.D.* Year *2006* Name of Author *GRIFFITHS, P.*

COPYRIGHT

This is a thesis accepted for a Higher Degree of the University of London. It is an unpublished typescript and the copyright is held by the author. All persons consulting this thesis must read and abide by the Copyright Declaration below.

COPYRIGHT DECLARATION

I recognise that the copyright of the above-described thesis rests with the author and that no quotation from it or information derived from it may be published without the prior written consent of the author.

LOANS

Theses may not be loaned but may be consulted within the library of University College London upon application.

REPRODUCTION

University of London theses may not be reproduced without explicit written permission from Library Services, University College London. Regulations concerning reproduction vary according to the date of acceptance of the thesis and are listed below as guidelines.

- A. Before 1962. Permission granted only upon the prior written consent of the author. (The Senate House Library will provide addresses where possible).
- B. 1962-1974. In many cases the author has agreed to permit copying upon completion of a Copyright Declaration.
- C. 1975-1988. Most theses may be copied upon completion of a Copyright Declaration.
- D. 1989 onwards. Most theses may be copied.

This thesis comes within category D.

This copy has been deposited in the library of University College London, Gower Street, London, WC1E 6BT.

SENATE HOUSE
MALET STREET
UNIVERSITY OF LONDON
SENATE HOUSE
MALET STREET
LONDON WC1E 7HU

**Developing Methodologies for Determining Operating
Strategies for Bioprocesses**

by

Paul Griffiths

Thesis submitted for the degree of
Doctor of Philosophy

in

The Advanced Centre for Biochemical Engineering
University College London
Torrington Place
London
WC1E 7JE

UMI Number: U592886

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U592886

Published by ProQuest LLC 2013. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

This thesis examines techniques for analysing bioprocess flowsheet simulations so as to determine operating strategies. Currently approaches cited in the literature for analysing bioprocesses employ visualisation of two-dimensional subsets of the feasible region. However this approach is restricted to two control variables and relies heavily on the engineer's judgement to estimate the potential impact of uncertainties in both the model and the process operation. The objective of this research was to generate methods capable of locating robust operating points for multivariate bioprocesses.

Increasingly the biopharmaceutical firms are under economic pressure to speed up process development. This has led to an increased interest in computer simulation as a tool to develop robust bioprocess. Whilst simulation has been applied extensively in the process industries it has not often been applied to bioprocesses as these tend to be more complex to model and frequently only a partial understanding of behaviour exists. Recent work has led to a capacity to simulate complete bioprocess sequences using models that capture the interactions between the unit operations. However, a major limitation is the interpretation of results from such simulations.

In conventional process engineering studies optimisation routines have been used to identify the best operating conditions for a given set of objectives. Such techniques have not been applied effectively to bioprocesses due to limitations in the reliability of the models. These limitations mean that results obtained via such an approach are unlikely to be useful as, in practice, the optimal points found are unlikely to be robust.

The work in this thesis also looks at defining methodologies that are able to analyse multivariable bioprocesses. It looks at the application of techniques developed in the chemical process industry that can be used to account for the variability in the control variables and process parameters and at the application of statistical techniques for analysing bioprocess robustness. Overall work highlights the nature of the bioprocess insights that can be obtained through simulation and explores the utility of the application of the developed methods of analysis.

Table of Contents

ABSTRACT.....	2
TABLE OF CONTENTS.....	2
LIST OF TABLES	8
LIST OF FIGURES	9
1 INTRODUCTION AND SCOPE.....	13
1.1 ABSTRACT.....	13
1.2 BACKGROUND	14
1.3 THE BIOPHARMACEUTICAL INDUSTRY.....	15
1.3.1 <i>Introduction</i>	15
1.3.2 <i>Research and Development</i>	16
1.3.3 <i>Manufacturing Process Development</i>	17
1.4 BIOPROCESS DESIGN.....	19
1.4.1 <i>Introduction</i>	19
1.4.2 <i>Process Economics</i>	21
1.4.3 <i>Interactions and Process Robustness</i>	24
1.5 IMPROVED BIOPROCESS DEVELOPMENT	26
1.6 AIMS OF RESEARCH.....	28
1.7 SUMMARY OF THE THESIS.....	29
2 SIMULATION AND MODELLING.....	34
2.1 ABSTRACT.....	34
2.2 INTRODUCTION.....	35
2.3 COMPUTER AIDED DESIGN IN PROCESS DEVELOPMENT	35
2.3.1 <i>Introduction</i>	35
2.3.2 <i>Software in Early Design</i>	37
2.3.3 <i>Process Simulation</i>	38
2.3.4 <i>Bioprocess Simulation Packages</i>	41
2.3.5 <i>Alternative Bioprocess Simulation</i>	43
2.4 PROCESS ANALYSIS	46

2.4.1	<i>"Windows of Operation"</i>	46
2.4.2	<i>Optimisation</i>	48
2.4.3	<i>Robustness</i>	52
2.4.4	<i>Multi-Dimensional Integration</i>	56
2.4.5	<i>Application to Bioprocesses</i>	58
3	BIOPROCESS CASE STUDY AND SIMULATION DEVELOPMENT	61
3.1	ABSTRACT	61
3.2	INTRODUCTION	62
3.3	CASE STUDY	63
3.4	SIMULATION PACKAGES AND LANGUAGES	64
3.4.1	<i>Introduction</i>	64
3.4.2	<i>SuperPro Designer</i>	65
3.4.3	<i>Labview</i>	65
3.4.4	<i>Matlab</i>	66
3.4.5	<i>C++</i>	67
3.4.6	<i>Development of the Simulations</i>	67
3.5	SIMULATION ARCHITECTURE	70
3.5.1	<i>Introduction</i>	70
3.5.2	<i>Modelling the Streams</i>	71
3.5.3	<i>The Unit Operations</i>	72
3.6	MODEL DEVELOPMENT	73
3.6.1	<i>Introduction</i>	73
3.6.2	<i>Fermentation</i>	74
3.6.3	<i>Enzyme Degradation</i>	76
3.6.4	<i>Centrifugation and Homogenisation</i>	77
3.6.5	<i>Dilution</i>	77
3.7	RESULTS	78
3.8	CONCLUSIONS	80
4	ANALYSING BIOPROCESS OPERATING SPACE	89
4.1	ABSTRACT	89
4.2	INTRODUCTION	90
4.3	THEORY	91

4.4	COMPUTATIONAL METHODS	92
4.4.1	<i>Calculating Feasibility</i>	92
4.4.2	<i>Size of the Feasible Region</i>	93
4.4.3	<i>Determining Operating Ranges</i>	94
4.5	RESULTS	95
4.5.1	<i>Introduction</i>	95
4.5.2	<i>Distribution of the Feasible Region</i>	96
4.5.3	<i>Interactions in the Feasible Region</i>	98
4.5.4	<i>Size in Process Analysis</i>	98
4.5.5	<i>Monte Carlo Simulation</i>	100
4.5.6	<i>Operating Ranges</i>	101
4.6	CONCLUSIONS	103
5	SCENARIO-BASED APPROACHES.....	116
5.1	ABSTRACT.....	116
5.2	INTRODUCTION.....	116
5.3	THEORY	118
5.4	COMPUTATIONAL METHODS	121
5.5	RESULTS	124
5.5.1	<i>Introduction</i>	124
5.5.2	<i>Uncertainty in Control Variables</i>	124
5.5.3	<i>Uncertainty in the Parameters</i>	127
5.6	CONCLUSIONS	128
6	STOCHASTIC SIMULATION AND ANALYSIS.....	140
6.1	ABSTRACT.....	140
6.2	INTRODUCTION.....	141
6.3	THEORY	142
6.3.1	<i>Stochastic Modelling</i>	142
6.3.2	<i>Analysis of Uncertainty</i>	144
6.4	COMPUTATION METHODS	145
6.4.1	<i>Introduction</i>	145
6.4.2	<i>Stochastic Simulation</i>	146
6.4.3	<i>Analysis of the Stochastic Simulation</i>	151

6.5	RESULTS	153
6.5.1	<i>Introduction</i>	153
6.5.2	<i>Visualisation</i>	153
6.5.3	<i>The Trade-off between Robustness and Performance</i>	156
6.6	CONCLUSIONS	162
CONCLUSIONS		175
ANALYSING MULTIDIMENSIONAL FEASIBLE REGIONS		176
SCENARIO BASED APPROACHES.....		176
STOCHASTIC MODELLING.....		177
FUTURE WORK.....		179
APPENDIX A.....		183
A.1	INTRODUCTION.....	183
A.2	FERMENTATION.....	183
A.2.1	<i>Overview</i>	183
A.2.2	<i>Wang Cooney Fed-Batch Model</i>	184
A.2.3	<i>Intracellular Stream Properties</i>	185
A.2.4	<i>Solid Properties</i>	186
A.2.5	<i>Liquid Properties</i>	186
A.2.6	<i>Other Properties</i>	186
A.3	CENTRIFUGATION	187
A.3.1	<i>Introduction</i>	187
A.3.2	<i>Separation Performance</i>	188
A.3.3	<i>Mass Balance</i>	190
A.3.4	<i>Volume</i>	191
A.3.5	<i>Concentration</i>	191
A.3.6	<i>Time Taken</i>	192
A.4	HOMOGENISATION.....	192
A.4.1	<i>Introduction</i>	192
A.4.2	<i>Mass Balance</i>	193
A.4.3	<i>Solid Properties</i>	194
A.4.4	<i>Liquid Properties</i>	195
A.4.5	<i>Time Taken</i>	195

A.5 DILUTION	196
<i>A.5.1 Introduction</i>	196
<i>A.5.2 Liquid Properties</i>	196
APPENDIX B: HYPER-RECTANGLES INSIDE A HYPER-SPHERE.....	198
NOMENCLATURE.....	203
REFERENCES.....	207

List of Tables

Table 3.1: A summary of the components that define the streams in the ADH simulation.....	83
Table 4.1: The five largest hyper-rectangles that will fit inside a feasible region defined by an alcohol dehydrogenase constraint of 4×10^6 units and debris concentration of 0.5g.L^{-1}	102
Table 6.1: The parameters used during optimisation	158
Table 6.2: The best results for each of the different levels of risk of the debris constraint being broken.....	160
Table 6.3: The best results for each of the different levels of risk of the debris constraint being broken when the number of homogeniser passes is treated as an integer variable.....	161

List of Figures

Figure 1.1: The clinical process required to bring a drug to the market	31
Figure 1.2: Steps involved in process development.....	32
Figure 1.3: A Schematic of Different processing options.....	33
Figure 2.1: Maximum scaled hyper-rectangle within the feasible region.....	60
Figure 3.1: The alcohol dehydrogenase process.....	84
Figure 3.2: Two bar charts showing the levels of alcohol dehydrogenase and cell wall in the different streams in the alcohol dehydrogenase process.....	85
Figure 3.3: Two bar charts showing showing the levels of alcohol dehydrogenase and cell wall in the different stream in the alcohol dehydrogenase process	86
Figure 3.4: The ‘Window of Operation’ showing the interactions between pressure and debris removal flowrate.....	87
Figure 3.5: The ‘Window of Operation’ showing the interactions between debris removal flowrate and the number of passes.....	88
Figure 4.1: An example where analysing the size distribution of the feasible region can lead to the selection of a robust operating point.	106
Figure 4.2: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different growth rates.....	107
Figure 4.3: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of productivity, with different growth rates.....	108
Figure 4.4: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different harvest centrifuge flowrates.....	109
Figure 4.5: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different homogeniser pressures.	110
Figure 4.6: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different debris removal centrifuge flowrates.....	111

Figure 4.7: Variation in the size of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a minimum levels of ADH productions of 4×10^6 units, for different homogeniser pressures and flowrates in the debris removal centrifuge.....	112
Figure 4.8: Variation in the size of the six-dimensional feasible regions.....	113
Figure 4.9: Variation in the size of the six-dimensional feasible regions, under different maximum debris removal constraints and minimum levels of ADH of constraints.	114
Figure 4.10: A graph showing the results of a series of Monte Carlo integrations showing predicted size against the number of simulations.	115
Figure 4.11: A graph showing the error when different numbers of simulations are used for calculating the size of the feasible region. The error is calculated using the root mean square.....	115
Figure 5.1: A schematic showing a two-dimensional feasible region (R_{FR}) and the subset of the feasible region that is robust (R_{OFR}) given a pre-determined level of uncertainty in the process control. The schematic also shows two points that are feasible, however one is not robust.	130
Figure 5.2: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate and homogeniser pressure	131
Figure 5.3: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate and homogeniser pressure and given a higher ADH production constraint.	132
Figure 5.4: The robust regions defined by homogeniser pressure and homogeniser passes given imprecision in the centrifuge flowrate and homogeniser pressure.....	133
Figure 5.5: The robust regions defined by debris removal centrifuge flowrate and number of passes given imprecision in the centrifuge flowrate and homogeniser pressure	134
Figure 5.6: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate, homogeniser pressure, dilution ratio and growth rate.	135

-
- Figure 5.7:** The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate, homogeniser pressure, dilution ratio and growth rate. 136
- Figure 5.8:** The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in debris density parameter. 137
- Figure 5.9:** The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the cell breakage coefficient. 138
- Figure 5.10:** The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the debris density and the cell breakage coefficient. 139
- Figure 6.1:** A collaboration diagram showing how each of the objects in the stochastic simulation, using random variables, interacts. 165
- Figure 6.2:** A collaboration diagram showing how each of the objects in the stochastic simulation, using quasi random variables, interacts. 166
- Figure 6.3:** The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure with a graph generated using 20x20 points, each calculated using 100 trials with random numbers generated using the Box-Muller algorithm. 167
- Figure 6.4:** The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure. This graph was generated using 100x100 points, each calculated using 100 trials with random numbers generated using the Box-Muller algorithm. 168
- Figure 6.5:** The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation. This graph was generated using 100x100 points, each calculated using 1000 trials with random numbers generated using the Box-Muller algorithm. 169
- Figure 6.6:** The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure. This graph was generated using 100x100 points, each calculated using 100 trials with quasi random numbers generated using the Sobol sequence and an inverse normal distribution algorithm. 170
- Figure 6.7:** The robust region defined by variation in number of parameters and control variables. This graph was generated using 100x100 points, each

calculated using 10,000 trials with quasi random numbers generated using the Sobol sequence and an inverse normal distribution algorithm. 171

Figure 6.8: The results showing the maximum amount of alcohol dehydrogenase (units of activity) in the product stream against predicted probability of failure to meet debris constraint where the Number of Passes is treated as a continuous variable. 172

Figure 6.9: The results showing the maximum amount of alcohol dehydrogenase (units of activity) in the product stream against predicted probability of failure to meet debris constraint where the Number of Passes is treated as an integer variable. 173

Figure 6.10: A Comparison between the results obtained looking at the trade-off between maximising alcohol dehydrogenase and probability of failure to meet the debris constraint, using the technique treating number of passes as a continuous variable (blue line) and as an integer variable (purple line) 174

1 Introduction and Scope

1.1 Abstract

This thesis looks at issues and solutions surrounding the development of methods for selection of operating strategies within particular bioprocess flowsheets. It does so against a backdrop of an industry subject to extensive regulation which means that many of the features of a process will be fixed during the development phase whilst the drugs are still being investigated for their clinical efficacy. This can lead to either a suboptimal process or delays in getting the drug to market, both of which can result in a loss of earnings. Consequently it is important that an effective process design is found rapidly and early in the development phase. This necessitates better strategies for evaluating process designs that do not increase the time to bring the product to market. Furthermore, effective strategies are required to ensure that the objectives of the process, once in operation, are met consistently when the process is operated.

This chapter examines the cost pressures bearing on the biopharmaceutical industry which are leading to a renewed interest in techniques that could be applied to reducing production costs and as an aid process development.

Computer simulations are often used in the traditional chemical industry. However, they are less common in the bioprocessing industry. Some work has been carried out into evaluating the costs associated with specific flowsheets and scheduling of task to reduced manufacturing costs. However little work has been carried out into methods for analysing process interactions to determine robust operating strategies. The inability to quantify these effects means that bioprocess engineers need to rely on pilot plant trials. This chapter demonstrates a need within the industry for better simulation and analysis techniques in order to reduce both the time and costs associated with developing a robust and efficient bioprocess.

1.2 Background

The healthcare industry is being revolutionised by developments in biotechnology. The genetic revolution has led to the discovery of a new generation of biological macromolecular drugs that have both enormous pharmacological and commercial potential. These drugs however are significantly more complex than the small chemical entities that have traditionally been developed by the pharmaceutical industry. This complexity means that the development and commercialisation of such products is difficult and takes a long time, as the drugs and their manufacturing processes are subject to a large amount of regulation (Werner *et al*, 1988, Baird & De Santis, 1994).

The manufacturing process is a critical bridge between the science and the market, as without a robust, scalable and economic manufacturing process drugs would be unlikely to reach the market. The manufacturing costs, associated with macromolecular drugs, are typically around 20% - 25% of the sales (Farid, 2002), although the manufacturing costs are heavily dependent on scale (Werner, 1998a). By contrast the costs of manufacturing small chemical entities are significantly cheaper. In addition the development of the manufacturing process needs to be carried out as quickly as possible since longer development times would reduce the profit generated if the drug is successfully commercialised.

Traditional bioprocess design methods have been very empirical, relying heavily on pilot plant trials, which are both expensive in labour and capital resources (Zhou & Titchener-Hooker, 1999). One potential methodology for improving bioprocess design is the use of computer tools to speed up the process design and reduce manufacturing costs. Computer tools such as process modelling, simulation and optimisation have already been extensively used in chemical engineering design. However, in bioprocesses the phenomena underlying process behaviour are generally less well understood and the use of computer tools more limited.

This chapter looks at the biopharmaceutical industry in more detail and in particular the importance of manufacturing. A short review will be given of the typical process

steps involved in bioprocess manufacturing. The chapter will then examine the trade-offs inherent in bioprocessing and the need for better design methodologies in order to develop more robust and economic bioprocesses.

1.3 The Biopharmaceutical Industry

1.3.1 Introduction

According to a recent report from Price Waterhouse Coopers (Arlington *et al*, 1998) the pharmaceutical industry is facing some serious challenges. The costs of research and development are soaring. Meanwhile research productivity, in terms of the number of new pharmaceutical products reaching the market, has declined. The report claims that the number of new substances that will be produced by the pharmaceutical companies is likely to remain short of that necessary for their revenue generating targets. Therefore their total shareholder returns are likely to drop significantly (Arlington *et al*, 1998). As a result there is increasing interest in biotechnology and its potential for providing new medicines.

Current developments in the fields of genomics and proteomics have generated a large number of candidate therapeutics. A significant proportion of these are likely to be large molecule drugs or biopharmaceuticals (Dvorin *et al*, 2001). Such drugs are based on understanding diseases at a molecular level. Currently the biopharmaceuticals in the market can be divided into three distinct groups (Werner, 1998b): -

1. Products that had previously been available from a natural source and can now be manufactured using recombinant technology. These include insulin, factor VIII and human growth hormone. Often the natural sources for these drugs are scarce and there are also potential risks with products, so derived (e.g. Insulin taken from bovine or pork islet cells).
2. Biological drugs whose modes of action are well understood, but are unobtainable from natural sources. Such drugs include tissue plasminogen activator, erythropoietin and DNase.

3. Biopharmaceuticals as rationally designed biologicals such as monoclonal antibodies designed to interact with specific enzymes or receptors.

Increasingly genetic engineering is being used to modify protein structure in order to optimise the pharmacokinetic and pharmacological behaviour and this too is leading to new products (Werner, 1998b). There is also increased interest in non-protein based biopharmaceuticals such as gene therapy, whereby DNA is introduced into cells so that they are able to produce a deficient or absent protein *in vivo*. Finally antisense oligonucleotides are being investigated for their ability to block the formation of undesirable proteins (Werner, 1998b).

Despite the apparent potential of new candidates, many pharmaceutical companies have been reluctant to invest in biopharmaceuticals. One factor, suggested by Dvorin (2001), is that biological drugs need to be delivered intravenously, and hence were not seen as being of sufficient commercial value. However recent analysis predicts that in the next 10-15 year 50% of new active substances could be biologics and therefore pharmaceutical companies need to invest in capabilities to develop and manufacture these drugs (Dvorin, 2001).

1.3.2 Research and Development

In order to get any drug to market a significant amount of time and money needs to be invested in research and development. Before drugs are able to reach the market they have to have clinical proof of their safety and efficacy. The clinical process required to bring a drug to the market is shown in figure 1.1. The first stage of this process is testing the toxicity of the drugs using animal models. After the safety has been assessed, the drugs are tested on health volunteers in Phase I clinical trials to evaluate their potential side effects. Finally in Phase II and Phase III clinical trials the drugs are tested on patients prove their efficacy. Typically of 2,500 screened compounds only 250 will enter pre-clinical trials, of which only 5 will ever enter clinical trials and only 1 will reach the market (Pharmaceutical Industry Profile, 2002).

The development times for biopharmaceuticals have been generally shorter than those for small chemical entities (Werner 1998b). According to Foo *et al* (2001), the average time to market for biopharmaceuticals is around 7.8 years. However, as Werner (1998b) point out, this could be in part attributed to the early biopharmaceuticals already having clinical proof of concept. Critically recent biopharmaceuticals whose mode of action is not understood have shown longer development times, between 9-17 years similar to that of chemical entities (Werner, 1998b). Such long development times mean that the time on the market under patent is reduced and potential revenues are lost.

Another important factor is the development cost of drugs. According to Arlington *et al* (1998) the research and development costs per approved drug are between \$350m - \$600m. The PhRMA report estimates the research and development cost per approved drug at over \$800m (Pharmaceutical Industry Profile, 2002). However, 90% of drugs have annual revenues of less than \$180m. Hence if the industry is to increase its profitability then it will need to find methods of reducing substantially the cost of drug research and development (Arlington *et al*, 1998).

A number of technologies could potentially reduce research and development costs. Better drug discovery methodologies should lead to more focused drug discovery to reduce the attrition rates in the clinical trial phases. Equally many delays occur during clinical trials and better clinical trial procedures could potentially reduce the time to market. However, as the next section will demonstrate, better process development strategies could have a significant impact on the costs throughout the lifecycle of the product.

1.3.3 Manufacturing Process Development

Traditionally process development in the pharmaceutical industry was done at a late stage and process modifications were only considered later when the patent life of the product was running out and the threat of generic competition had increased. There was a good economic logic for this strategy as manufacturing cost were typically less than 10% of revenues and inefficient processes would have a limited impact on the product's profitability (Pisano and Wheelwright, 1995). However, the cost of

manufacturing a biologic is typically 20-25% of revenue (Farid, 2002), significantly higher than for small chemical entities. Piscano and Wheelwright (1995) showed, through analysing a number of case studies, that treating process development as an after-thought can lead to delays in the product reaching the market. Such delays can lead to the loss of potential revenue, therefore highlight the importance of effective process development.

It is difficult to determine the exact impact that process development plays on the overall development time. However, according to Foo *et al* (2001) one of the critical roles of process development in the early phases of drug development is to devise a process to manufacture enough product to support the clinical trials. Insufficient material for clinical trials can lead to delays in the trials and ultimately delays in product launch. Early stage investment in process development can potentially reduce the time to market. Karri *et al* (2001) showed how time spent improving process efficiency could improve the productivity of the process, thereby reducing the time required to manufacture material for clinical trials. In addition to this, effective process development is also advantageous due to the extensive regulation in the biotechnology sector.

In contrast to the chemical synthesis of drugs, the cellular synthesis of biological products cannot be directly controlled, but only indirectly through environmental parameters. As a result small changes in the environmental conditions have a significant impact on the quality of the product (Werner, 1994). Before biopharmaceuticals are tested on human subjects the manufacturing process needs to be validated by the regulatory authority, such as the EMEA in Europe and the FDA in America. The validation procedure involves the planning and carrying out of a testing procedure to prove a process will constantly produce a product to the predetermined specifications. The results of the testing are documented to provide evidence to the regulatory agencies that the process is suitable for manufacturing (Baird & De Santes, 1994).

Any significant process changes after validation would require bioequivalence studies to compare the product of the modified process to that of the original process

(Werner, 1994). As a result, bioprocesses technology gets locked-in at an early phase, usually before clinical testing. Ideally any process used in the phase I clinical trials, should be suitable for scale-up, otherwise the company could end up with a manufacturing process that is difficult to operate at industrial scale (Foo *et al*, 2001). One such example given by Foo *et al* (2001) is that of Amgen whose process for manufacturing Epogen is done in roller bottle cultures, which although economic for that particular molecule, would not likely be so for a drug with a lower market value. Therefore, effective process development in the early stages of clinical development should result in a more efficient and economic manufacturing process that will produce better gross margins.

However a recent paper by Byrom (2000) acknowledges that although extensive process development at an early stage may reduce the manufacturing costs, it will require a large investment in a high-risk project. Ideally process development should be done in parallel with clinical development. A methodology for achieving this is suggested by Byrom (2000), in which the development was looked at in three stages (figure 1.2). The strategy described tries to link the resources invested in process development at each stage to the risk of clinical failure.

In conclusion the ultimate aim of process development is to design a robust and economic bioprocess within the time and resource constraints available. The next section will focus on the design of bioprocesses, with particular reference to unit operations that are included in the manufacturing processes examined in this thesis. The section following will then highlight new techniques being developed that could potentially reduce process development costs and enable more advanced process development to be achieved at an earlier stage.

1.4 Bioprocess Design

1.4.1 Introduction

Typically processes for producing bioproducts can be divided into an upstream stage, where the product is produced by fermentation, and the downstream stage, used to recover and purify the product. When designing a bioprocess an engineer will need to

develop an effective fermentation process that will produce the product in the quality and quantity that is required. They will then need to develop a downstream process to obtain the product at the right purity and concentration. Often the downstream process will critically depend on how the product is expressed in the upstream stage (Kelley and Hatton, 1991).

The basic structure of most downstream flowsheets is set out by Gandikota *et al* (1992). Figure 1.3 shows three basic routes to purify a protein, depending on how it is expressed at the fermentation stage: -

1. If the product is extracellular, the cells are separated from the broth, and the cell free liquor is then concentrated before the product is purified from it by a sequence of high-resolution purification steps and finally by a polishing step.
2. If the product is intracellular and soluble the cells are separated from the broth and passed through a cell rupture step to release the intracellular product. Initial product recovery step or steps are used to remove key contaminants before the concentration, high-resolution purification steps and the polishing steps.
3. Finally if the product is intracellular and insoluble then often it will amass inside the cell in the form of large aggregates called inclusion bodies. Cell rupture is used to release the inclusion bodies from the cells. The inclusion bodies are then separated from the intracellular compounds using a physical separation technique that takes advantage of their relatively high density and size compared to the cell debris. In order to obtain the protein in a useful form, the inclusion bodies are then resolubilised and a refolding step is used to convert the protein into its active form. Finally a concentration step is used before the high-resolution purification steps and the polishing step.

The three flowsheets are each based on a sound logic; intracellular products need to be released from the cells; proteins in the form of inclusion bodies need to be resolubilised and refolded to be in a useful form; reducing the amount of feed via

concentration before the high-resolution purification steps reduces capital cost in such steps; finishing on a polishing step ensures the purity of the final product. However, the challenge is establishing the appropriate unit operations and their order to accomplish these tasks.

In work by Bonnerjea *et al* (1986), 100 different protein purification processes were examined to determine the typically order of unit operations. The work looked at the position of each unit operation within a sequence and showed that a typical order of unit operations was: -

- Homogenisation – Cell rupture
- Protein Precipitation - Purification and production concentration
- Ion Exchange - High resolution purification
- Affinity Chromatography - High resolution purification
- Gel Filtration - Polishing

Ultimately the design of a bioprocess needs to consider two fundamental factors, the process economics and the reliability of the process. A significant body of work has been carried out looking at process economics. Very little work has been carried out to look at developing models to determine the reliability of bioprocess. The next section looks in greater detail at the bioprocess economics and discusses how factors that affect the economics may have a detrimental effect on the process reliability.

1.4.2 Process Economics

Any new bioprocess will require an initial capital investment in process equipment. The cost involved will depend on the complexity of the product being manufactured and the unit operations required. For example, a complex biological therapeutic is likely to involve considerably higher downstream processing costs, due to the high level of purity required. This was demonstrated by Datar (1986) who showed that the ratio of recovery to fermentation cost was 0.16 for ethanol and 2 for an enzyme. For a drug to be profitable it must be able to provide a return on both this investment and the research and development costs. The profitability of the drug will depend on the throughput per year, the cost of production (or cost of goods) and the value of the end product.

The costs associated with operating a biopharmaceutical plant can be divided into variable costs that are dependant on the amount produced and fixed costs that are independent of the amount produced, although as pointed out by Reisman (1999) most costs will have a fixed and variable element. Many of the fixed costs such as maintenance, insurance and depreciation of capital equipment can be estimated as a proportion of the capital investment. Work done by Petrides (1994) and Petrides *et al* (1995, 1996) using the simulation package, Biopro Designer, showed that in the insulin production process the fixed costs form the largest proportion of the costs. Significant variable costs are waste treatment and disposal, raw materials and consumables, such as matrices for chromatography and membranes (Petrides, 1996). However, both labour costs and utilities were shown to make up a small proportion of the total costs.

In another study Farid *et al* (2000) developed a cost of goods models for biopharmaceutical production using Rethink (Gensym, MA). Here the variable costs were computed on the basis of the utilisation of the material, utilities, and staff resources and the fixed costs were derived from the capital investment. The model was used to compare the costs of producing batches of a high value biopharmaceutical with a conventional stainless steel plant or a plastic disposable plant, where the items of equipment are thrown away after use. This work showed that disposable equipment can be particularly cost effective for carrying out small scale manufacturing for clinical trials.

Three methods for increasing the productivity of a bioprocess are: better scheduling of unit operations, operating at a larger scale and the development of more efficient processes. Better scheduling of unit operations can increase the production rate without significantly increasing the operating cost (Petrides *et al*, 1999; Petrides *et al*, 2002). However, better scheduling can sometimes only be achieved by de-bottlenecking by the purchase of additional capital equipment. For example, fermentation steps usually take longer than the downstream processing so adding a second parallel fermentation train could result in double the utilisation of the downstream processing equipment. However in such situations the improvement in

the process performance needs to be weighed against the additional investment (Petrides, 2002).

Equally increasing the scale of production can dramatically reduce the production cost per amount of product. Werner (1998) demonstrated this by showing the cost of producing protein in mammalian cell culture was inversely proportion to the scale of operation. This was because the large increases in production rate were offset by relatively small increases in operating costs. However such an approach will only be cost effective, if there is a market for the additional product and will have little value for products with a small market. Equally the longer fermentation times associated with larger scale production could potentially mean a greater risk of contamination particularly for mammalian cell cultures (Werner *et al*, 1988).

Finally another method is to improve the process design. A typical protein production process may require as many as eight steps to obtain the crude enzyme (Wheelwright, 1991). Subsequent purification based on chromatography may need a significant number of additional stages. The effect of these additional stages is to decrease the process yield and significantly increase the costs. However as Zhou *et al*. (1997) acknowledge, systematic methods are required to ensure that any process changes will not result in a reduction in product quality or process reliability.

One factor that they note will be of particular importance is process interactions. Process interactions are where change in the conditions in one unit operation leads a change in the output of a later unit operation. These interactions are caused by changes in the properties of the process stream that have an impact on the performance of later unit operations. In extreme cases such interactions can lead to an apparent improvement in the performance of one step but actually reduce the efficiency of the whole process.

Equally such interactions can result in small fluctuation in parameters having a significant effect of process output making process performance unreliable. Typically engineers will want to design reliable processes and hence will be prepared to increase capital and operating costs to ensure this. However, greater systematic analysis of

interactions and uncertainty should enable engineers to make more informed decisions and hence reduce the production costs. The next section looks in more detail at problems associated with interactions and robustness in bioprocesses.

1.4.3 Interactions and Process Robustness

One of the main sources of interactions and uncertainty in a bioprocess will be the fermentation step. This is because the properties of the broth will have an impact on how the recovery operations can be performed. For the whole manufacturing process to be efficient the fermentation product should be in a form that can be easily purified in the downstream processing steps (Datar *et al*, 1993). Oolman and Liu (1991) give the example of a *P. chrysogenum* fermentation where changes in the morphology under different fermentation conditions have a significant impact on the recovery steps. It will also affect the amount of product that will be available for purification and the form of this product. This was demonstrated by Gregory *et al* (1996) who looked at the production of a number of intracellular enzymes in *Saccharomyces cerevisiae*.

Fermentation will also be a major source of uncertainty as there is no direct means of controlling the processes within the cells. Instead operators have to regulate the cells environmental parameters inside the fermentation vessel, such as temperature, oxygen supply, carbon source and pH to ensure the cells consistently produce the product (Werner and Langlouis-Gau, 1989).

If the product produced by the cell is intracellular then the cell wall will need to be ruptured to release the product. The most common method for this is by using a high-pressure homogeniser (Middelberg *et al*, 1992a; Middelberg *et al*, 1992b; Klenig and Middelberg, 1996). In a high-pressure homogeniser, cells are disrupted by forcing the cell suspension through an adjustable, restricted orifice valve under high pressure. Increasing the pressure and the number of times the cells are passed through the homogeniser will increase the release of product. However, it will also result in the cell wall being broken into micronised debris. Micronised debris is difficult to separate using normal solid/liquid separation methods. If intracellular product is soluble there is a trade-off between achieving a high level of recovery of intracellular product and removing the debris contaminant (Zhou & Titchener-Hooker, 1999). However, if the

intracellular product is in the form of inclusion bodies, the inclusion bodies will be easier to separate from the debris if the debris is micronised (Wong *et al*, 1997).

Once a solution has been obtained a step is used to get it in a higher concentration before chromatography. The main advantage of using protein enrichment techniques is that the scale of subsequent high-resolution operations is significantly reduced, thereby providing savings in both capital and operating costs. Commonly used enrichment operations include precipitation, extraction and ultrafiltration. Both precipitation and extraction utilise differences in protein solubility behaviour as a basis for separation. The efficiency of these steps will also be affected by upstream processing. One example of this was demonstrated by Zhou *et al*. (1996) who showed how protein solubilities are reduced when debris is present.

The final steps in most bioprocesses are high-resolution chromatography steps. The performance of these unit operations will be affected by the feed. Often when homogenised solutions are used the solid particle contaminants, particularly cell wall material, foul the columns by damaging the matrix and reducing separation efficiency. A methodology for experimentally determining the impact of such foulants on the performance of chromatography was developed by Hearle *et al* (1994).

Additionally chromatography will have a significant impact on the process economics. The capacity of the column will be proportional to its length; however, longer columns will require more matrix and hence cost more. Another approach to increase the capacity of the column is to reduce the load speed (Jungbauer and Kaltenbrunner, 1991, Ngiam *et al*, 2001). However this also reduces the productivity of the column. Fahrner *et al* (1998) looked at these trade-offs for a given system and determined that the optimum operating conditions for their specific column was a relatively short column length with a slow throughput. Equally for both adsorption and non-adsorption chromatography the yield will be affected by the elution stage. Here, there is a trade-off between purity and yield of product (Ngiam *et al*, 2001). Typically collecting a broad fraction will yield greater amounts of product but of a lower purity whereas a narrow fraction will increase the purity at the expense of yield.

One problem with bioprocess design has been the inability to quantify these effects. Currently the only way to investigate such interactions is through pilot scale experimentation (Zhou and Titchener-Hooker, 1999). This is because standard laboratory equipment does not replicate the performance of industrial scale processes (Maybury *et al.* 2000, Boychyn *et al.* 2000 and Boychyn *et al.* 2001). Additionally the current generation of bioprocess simulators are unable to accurately predict the performance of processes. This is because many of the interactions are not yet well understood. The ability to quantify these effects would enable the development of accurate whole process models and effective analysis of different strategies on the quality of the product.

1.5 Improved Bioprocess Development

In order to design better processes, methods are required for evaluating different process options at an early stage of development. This would enable different processing options to be evaluated and a reliable and economic manufacturing process to be selected. The next section looks at the problems associated with early stage process development and potential tools that can be used to help overcome these problems.

Currently there is a trade-off in investing resources in process development. If resources are allocated to process development at an early stage then there is a real chance of improving the efficiency of the manufacturing process and hence the profitability of the therapeutic agent. However during early stage development the risks associated with the project are high as large numbers of drugs fail. Investing process development effort on all early stage drugs, before it is known whether they will be clinically effective, would result in a large increase in overall development costs.

The critical problem is that there are few tools available for the predictive design of processes and for determining the most suitable operating conditions. Traditionally most process design in the biopharmaceutical industries has made recourse to existing processes and has relied on the use of expensive pilot plant facilities to test new designs (Zhou and Titchener-Hooker, 1999). This approach is expensive in both

resources and time. Increasingly companies are looking for systematic methodologies to speed up process development, cut development costs and design bioprocess more efficiently.

One approach that has been used in a variety of engineering disciplines is 'Scale-up'. In process engineering small scale laboratory equipment is used to mimic industrial scale process. However, one problem that has been traditionally encountered in biochemical engineering is that standard laboratory equipment cannot replicate industrial scale process equipment. Often laboratory scale equipment cannot be used to analyse the impact of shear forces and mixing when scaling up from a laboratory process to an industrial process. For example, production scale fermenters will not transfer oxygen to a culture as efficiently as laboratory scale fermenters (Titchener-Hooker *et al.*, 2001). Equally laboratory centrifuges do will not replicate the shear forces in continuous industrial centrifuges and hence cannot be used to emulate the recovery of "delicate" biological material (Boychyn *et al.* 2001).

One methodology that has been examined recently is the use of small experimental devices that more accurately replicate the actions of industrial unit operations. This approach is more commonly known as 'scale down'. This emulates the behaviour of industrial scale equipment (including the unwanted effects) at a smaller scale, meaning that industrial processes can be mimicked using small amounts of material and hence at lower cost. This enables processes to be investigated rapidly and at an earlier phase in the products development when the project is still risky. A number of 'scale-down' devices have been developed and more detail on this field is given by Maybury *et al.* (2000), Boychyn *et al.* (2000), Varga *et al.* (1997) and Boychyn *et al.* (2001).

Another tool with the potential to aid process development in the biotechnology industry is process simulation (Evans and Field, 1988; Gritsis and Titchener-Hooker, 1989; Petrides *et al.*, 1989; Narodoslowsky, 1991). Unlike in the traditional chemical industry, computers are not used extensively in the field of biochemical engineering for design and evaluation of processes. One of the main obstacles to a greater use of simulation in this area has been the lack of robust, predictive models (Gritsis and Titchener-Hooker, 1989). This is due to the complex nature of biochemical systems,

the lack of available physical property data and poorly understood unit operations. A more extensive review of process simulation and computer application in bioprocess design is given in the next chapter.

'Scale-down' models offer the potential to extract critical process parameters via experimentation. The data from such experiments can then be used to generate accurate process models of unit operations that capture key interactions between the steps. Such models can ultimately be used to generate simulation of whole processes and hence analyse process trade-offs (Titchener-Hooker *et al*, 2001).

1.6 Aims of Research

The aim of this project is to apply modelling and simulation techniques to the analysis and design of a whole bioprocess, which consists of production, recovery and purification stages. As previous work focuses more on the individual unit operation, this work will take an integrated approach as the interactions between unit operations should be taken into consideration in process analysis and design.

The first part of this thesis will look at the development of bioprocess simulation that is able to capture interactions between unit operations using models generated from 'Scale-down' experimental data. This will build upon earlier simulation and modelling work. This work will look at a number of different packages for developing such simulation and analyse their merits.

The second part of this work will develop methods to analyse bioprocess simulations. This will look at methods for evaluating the impact of different control variables so that process operating strategies can be determined that will fulfil a set of predetermined conditions such as minimum acceptable yields of product and maximum acceptable levels of purities. This work will also look at methods for determining likely performance given uncertainty in both control variables and model parameters. The objective here will be to determine operating strategies which will offer reliable performance.

1.7 Summary of the Thesis

The next chapter will look at computer simulation tools that have been developed for the chemical industry as well as simulation tools aimed specifically at bioprocess engineers. The chapter will examine recent advances in the simulation of uncertainty and dynamic processes that could have application for bioprocess engineers.

Chapter 3 will present a bioprocess case study and will look at how the unit operations models were developed from information in the literature including the assumptions that had to be made and the limitations of the models. The chapter will look at the various packages used to simulate the processes, from high-level packages to low-level programming languages.

Analysis methods for analysing the simulation outputs will be explored in Chapter 4. This chapter will look at how the feasible region is distributed in a multi-dimensional space defined by the operating variables. This work will show how this can be used in order to analyse the process interactions and potential operating points. This chapter will show how different methodologies can be used for evaluating the size of a multi-dimensional feasible region.

Chapter 5 investigates the impact of inaccurate control and inaccuracies in the underlying models on the simulation. This will be used to determine operating strategies that allow for these inaccuracies and ensure robust process performance. The work will seek to determine under what constraints a process design that had been feasible can be rendered infeasible by inaccurate control and parameters.

Chapter 6 shows how stochastic simulation techniques can be applied to generate probabilistic simulation. The work will then show how the results of such a simulation can be analysed to determine which process control strategies are likely to result in 'unacceptable' risk of a constraint being broken and how this should result in the selection of more reliable operating strategies. This chapter will also look at how stochastic simulations can be used in conjunction with optimisation techniques to look at the trade-off between probability of achieving a constraint and achieving

higher yields. The work will show both the potential benefits of using optimisation techniques as well as the difficulties of applying them effectively.

Finally the conclusions review the work methods applied in the thesis and examine their potential for analysing bioprocesses and determining operating strategies. The future work then looks at how these approaches could be improved as well as the need for more research examining the development of bioprocess models from experimental data.

Figure 1.1: The clinical process required to bring a drug to the market

(PhRMA, Pharmaceutical Industry Profile 2002, Chapter 2)

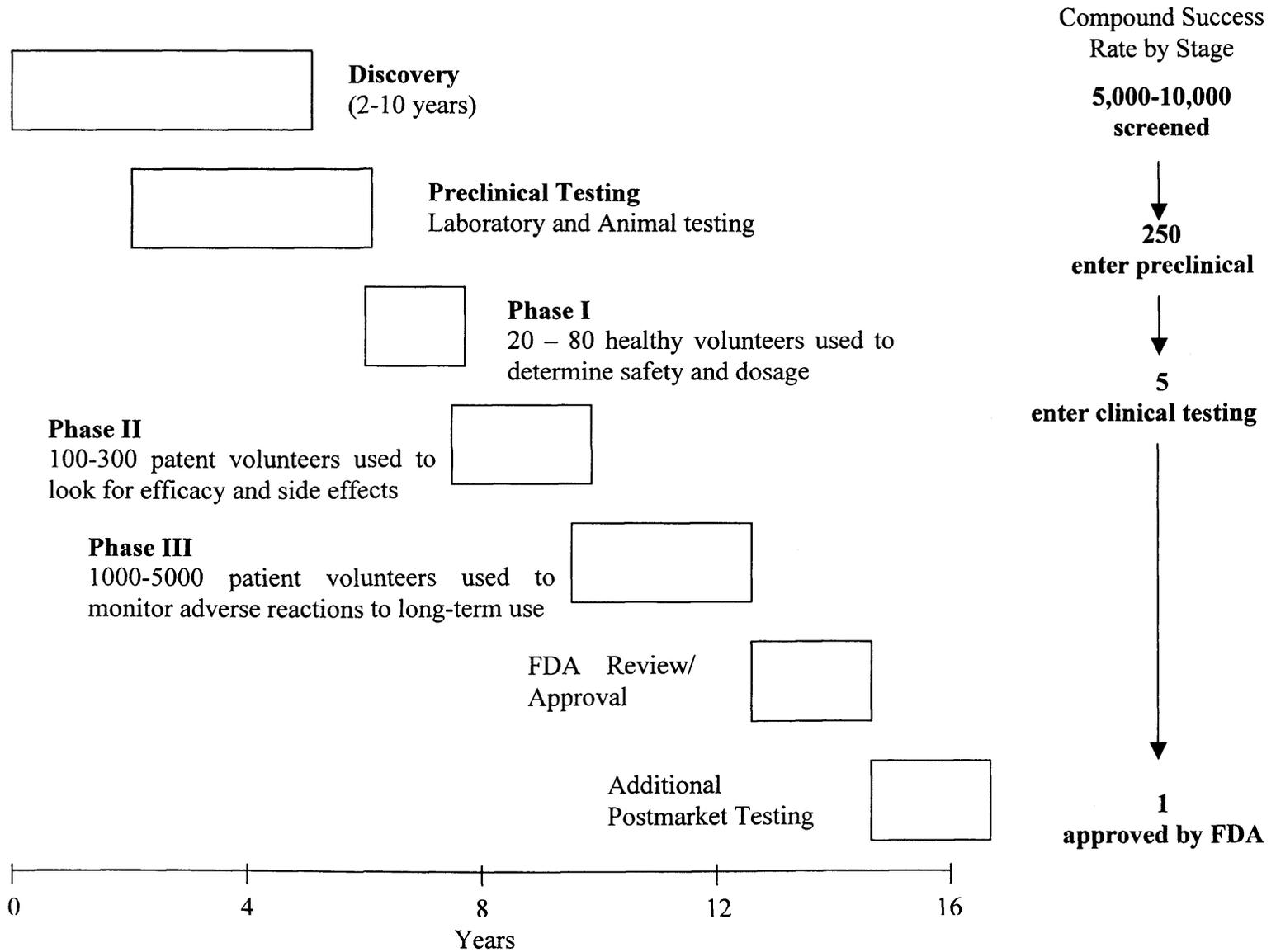


Figure 1.2: Steps involved in process development

(Bryon (2000), Pharmaceutical Technology Europe, 12, page 54)

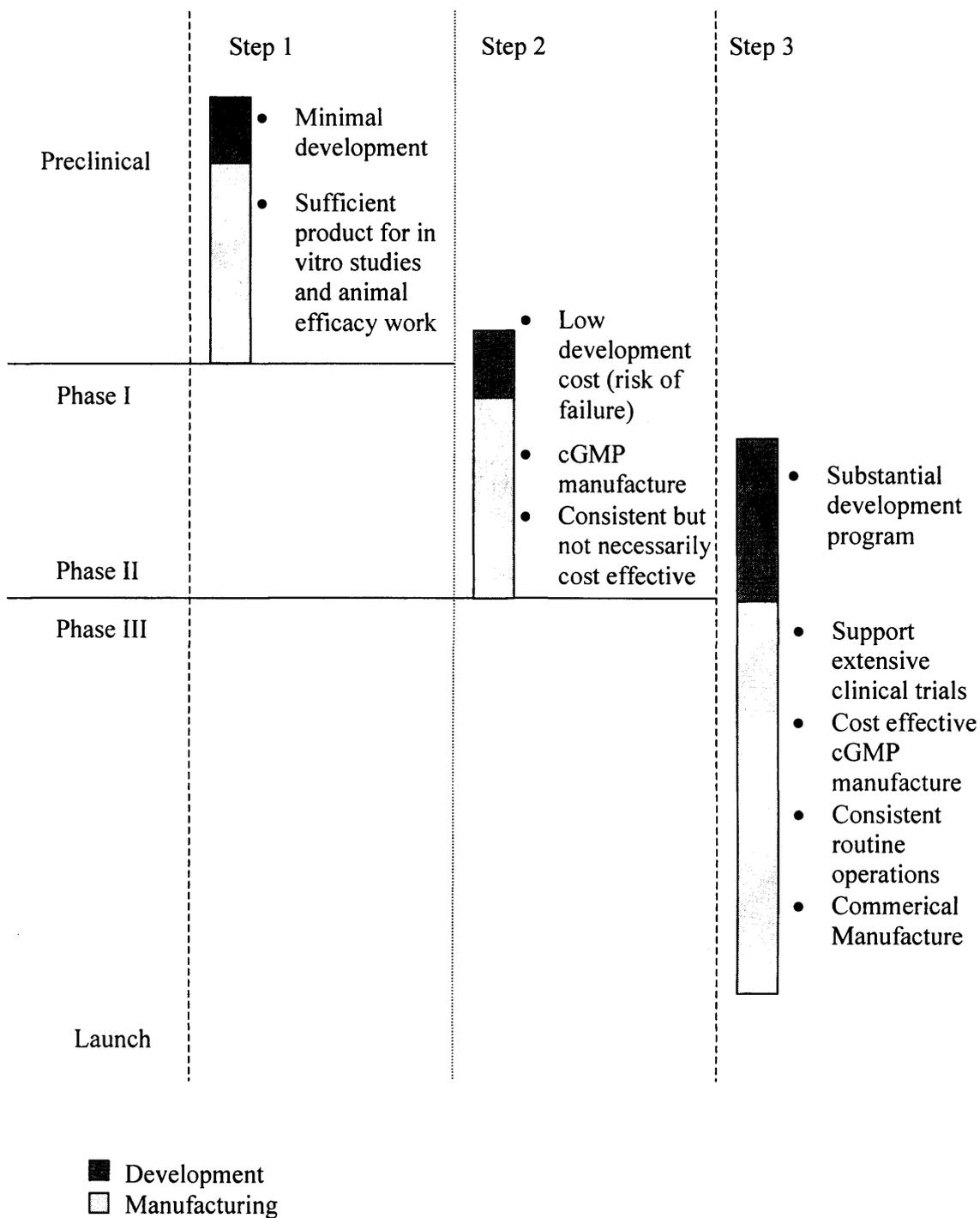
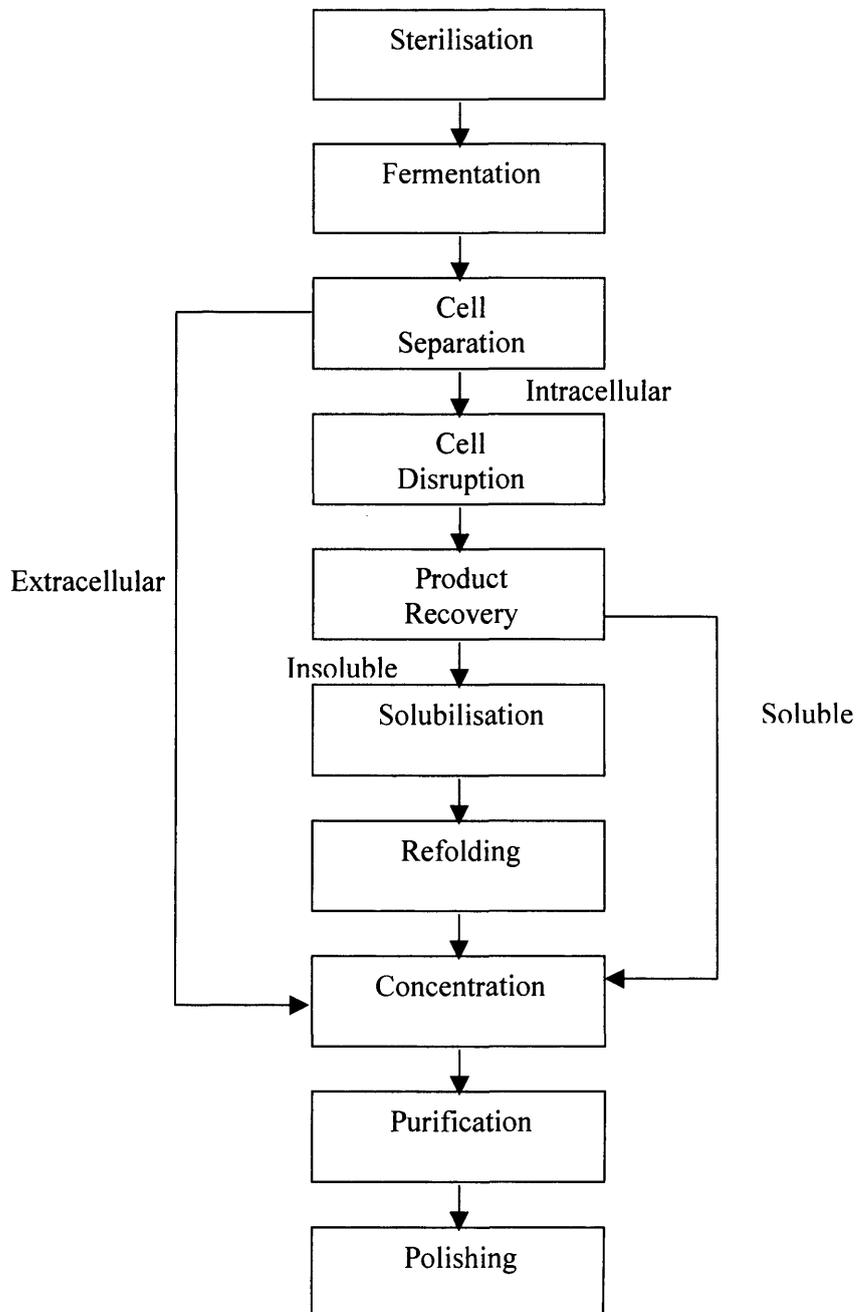


Figure 1.3: A Schematic of Different processing options

(Gandikota *et al* (1992), Chemical Technology, Vol 12 (11), pages 694-695)



2 Simulation and Modelling

2.1 Abstract

Chapter 1 reviewed the increased pressure to reduce operating costs, which is leading to a renewed interest in computer tools for bioprocess design. This chapter looks at computer tools that are used for the design of processes with particular focus on tools used for simulating flowsheets. The chapter then looks at methods used for analysing the results of such simulations.

In the chemical industry, both sequential modular and equation-orientated packages have been used for simulating processes. Sequential modular packages have the advantage that they can be built with unit operation libraries that can be reused. Equation orientated simulator do not usually contain libraries making it difficult to reuse models. In contrast with the traditional chemical industry, the biotechnology industry has not used simulation as extensively for process analysis. Currently the best-known commercial bioprocess simulation package is SuperPro Designer. This is a sequential modular simulation with relatively simple built in models that have a limited capacity to capture interactions in bioprocesses. Consequently, some academic researchers have investigated both the development of more detailed models and the implementation of them using generic simulation packages and programming languages.

Techniques exist for analysing simulations to obtain useful results. One technique that has found application in bioprocess simulation is the so-called “Window of Operation”, where a two-dimensional plot is drawn showing a feasible region defined by two control variables. By contrast, in the traditional chemical industry the technique that is typically used is mathematical optimisation, which in part reflects the greater confidence in the underlying models. However, other researchers have investigated techniques for determining robust operating points given variations in model parameters. One technique commonly applied in a number of areas is Monte Carlo integration, which can be used to analyse a multi-dimensional space. Both Monte Carlo integration and methods for finding robust operating points

could potentially be applied to the analysis of multivariable bioprocess simulation to determine robust operating points. The application of these approaches is investigated in later chapters.

2.2 Introduction

The last chapter demonstrated the need for new tools for developing bioprocesses that are both economically viable and can reliably manufacture the desired product. Existing strategies for bioprocess development make extensive use of experimentation and "rules-of-thumb". Currently the bioprocessing industry is in a situation that is similar to the traditional chemical industry during the 1970s (Evans *et al*, 1988). One tool that has the potential to improve process development is process modelling and simulation. Modelling can be defined as the mathematical presentation of a process and simulation as the use of the model to predict a plant's performance and ultimately its economics.

Computer simulation tools have been extensively used for the design of bulk chemical manufacturing processes. An overview of its applications in this field is given in section 2.3, looking at both existing commercial packages as well as recent developments in academia. This section then examines work that has been done on bioprocess simulation. Compared to bulk chemical processes less work has been carried out into bioprocess simulation because of the complexity of biological systems, which often involve many interactions that are not well understood and not easily simulated. Finally Section 2.4 looks at the techniques that have been used to analyse both chemical and bioprocess simulations in order to determine operating points.

2.3 Computer Aided Design in Process Development

2.3.1 Introduction

The key objectives in process design are sizing equipment and determining operating conditions to ensure the process will deliver enough of the product with the desired specification at an affordable cost. Before the development of computers the design of chemical engineering plants used simple strategies to find solutions to specific

problems. Consequently process design relied heavily on engineering judgement, rather than systematic methods (Sargent, 1967).

The use of computer simulation has enabled a more systematic approach to process development. By applying process models, that mathematically describe the unit operations in the process, mass and energy balances can be calculated over an entire process. Over the last four decades, the application of process simulation has been examined extensively by both industrial and academic groups.

The advantages that may be realised through using such packages are faster process design and improved process efficiency. Increasingly computer simulation is being seen by many large chemical companies as a core competency for gaining competitive advantage, in an increasingly global market (Kreiger, 1995).

By contrast, the biotechnology industry does not make significant use of simulation and carries out process development based on pilot plant experiments (Zhou and Titchener-Hooker, 1999), which is both expensive and time-consuming. The industry is only just beginning to make use of simulation for process development. According to one estimate, bioprocess simulation has the potential to reduce manufacturing costs by 10% to 30% by improving process recovery yields (Evans *et al*, 1988). With increasing competitive pressure in the pharmaceutical industry such benefits are likely to increase the interest in this technology.

There are, however, a number of fundamental differences between biochemical engineering and chemical engineering. These distinctions have an impact on the ability of traditional, chemical engineering simulation packages to tackle biochemical engineering processes. Evans *et al* (1988) suggested the three key issues impeding the building of a bioprocess simulator are:

- Models of novel unit operations.
- New types of materials and physical properties.
- Integrated batch, semi-batch, semi-continuous and cyclic operations.

Bioprocesses have many novel **unit operations** that are not used in traditional chemical engineering processes. An example is high-pressure homogenisation for rupturing cells. When analogous unit operations do not exist in chemical engineering, the bioprocess models tend to be inadequate for simulation. This is because such unit operations are often poorly understood and accurate predictive models frequently do not exist.

Physical property models and data are essential for the accurate simulation of chemical processes. Chemical simulations deal mainly with liquids and vapours. However, bioprocess simulations have to handle complex materials such as cells, cell debris, precipitates, proteins, and polymers. Ideally each of these biomaterials should be characterised according to biological activity, viscosity, partition coefficients, solubilities, diffusivity, etc (Evans *et al*, 1988). However they are difficult to characterise and the accuracy of the assays used is often limited.

Finally in biochemical processing, **batch, semi-continuous and continuous** unit operations are often encountered within one process (Gritsis and Titchener-Hooker, 1989). This is in stark contrast to many bulk chemical processes where each unit operation is operated continuously at a steady state. As a result bioprocess simulation often requires dynamic models. These differences mean that flowsheeting and design packages designed for the chemical industry are unsuitable for the bioprocess simulation. Therefore a number of attempts have been made at developing computer simulation tools for bioprocesses (Evan *et al*, 1988; Petrides *et al*, 1995). The next sections look how different types of software packages can be applied in process design.

2.3.2 Software in Early Design

The design of both traditional chemical plants and bioprocesses can be split up into several stages (Westerberg, 1989)

- Conceptualisation, where the basic premise of the design is focused
- Generation of designs
- Analysis and evaluation of the solutions

Early-stage process design is a critical step in getting a product to market. In the chemical industry process design usually constitutes 10-15% of the cost of generating a new product, but choices made at this stage will fix 70-80% of the manufacturing cost (Westerberg, 1989). As a result a number of computer tools aimed at automating process synthesis have emerged for both chemical processes and bioprocesses. These process synthesis packages are able to generate automatically potential process designs from specifications set by the user.

In chemical engineering, the most commonly used process synthesis packages are based on implicit enumeration. The approach commonly used is to generate a superstructure of all the potential process routes, and then formulate an optimisation problem to search for the best process flowsheets. By contrast, approaches used for bioprocess synthesis are generally based on expert systems and the use of artificial intelligence, which in part reflects the limitations in current bioprocess models. A number of synthesis tools based on these concepts have been developed by academic groups. For example, the approach taken by Gandikota *et al* (1992) was to use a set of rules to determine the appropriate steps in a downstream process (Fig 1.2). A similar rules-based approach was also used by Silletti *et al*, (1992) in BioSep designer. One notable exception is the work of Steffan *et al*, (1999), who chose to extend *Jacaranda*, a numerical process synthesis tool used by the chemical industry, to synthesis bioprocesses. However, in their work they used relatively simple models; therefore any results are likely to be subject to significant inaccuracy.

In both chemical engineering and biochemical engineering, process synthesis software is rarely able to produce detailed or accurate process designs. Typically such techniques are used in the early stage of design and then refined later using more detailed simulation packages. These are discussed in the next section.

2.3.3 Process Simulation

A significant amount of research has been carried out on the development of process simulators. Early work in process simulation looked at traditional continuous chemical processes operated at steady state. Early simulators were able to carry out mass and energy balances over entire processes. In such simple situations the application of simulation enabled engineers to examine multiple "what-if" scenarios and even to

determine the most effective mode of operation through mathematical optimisation (Section 2.4.2).

All process simulations are composed of the flowsheet topology defining the unit operations and their connectivity, a physical property database or calculation routines and feed stream data. Simulations can either be performance based for calculating the performance of a particular process, or design based for determining equipment sizes so that the process can meet a set of objectives.

Early research into computer simulation of processes evolved in two distinct directions, **sequential modular simulation** and **equation-orientated simulation** (Biegler, 1989). The sequential modular approach breaks down the simulation task into discrete modules each describing specific unit operations. By contrast equation orientated approaches in their purest form remove the distinction between stream connections, the unit operations, and physical property data and focus on solving the underlying equations.

In **sequential modular simulators**, the flowsheet is represented by a collection of modules, containing the equations that describe the mass and energy balance relationships. This modular nature makes developing new process simulations relatively easy and means the packages themselves can be adapted to include new unit operations. Consequently, this architecture remains popular amongst current commercial simulators. Examples of such commercial simulators are ChemCAD (Chemstations, 1998), ASPEN PLUS (Aspen Technology, 1998), and SuperPro Designer (Intelligen, 1999).

In contrast with **Equation Orientated simulators**, the process model is in the form of a set of equations (and inequalities). The process equations are solved simultaneously to find a solution within the constraints specified (Pantelides, 1988). Generally equation-oriented simulators offer better computational performance for solving problems with recycles and process constraints as less time is spent on iterative calculations. This improved performance is particularly critical for dynamic simulation and carrying out process optimisation (section 2.4.2). Two of the leading commercially available equation orientated simulation packages are Aspen Customer

Modeller (Aspentech, MA), formerly know as SpeedUp and gPROMs (Process System Enterprise, UK).

Dynamic simulation enables the examination of process performances that are subject to change over time. This is particularly critical when simulating batch processes. However, dynamic simulation is also a useful tool for examining continuous processes, which typically show time-varying behaviour during start-up, shut down, and when subject to set point changes and process disturbances. The demand for better dynamic simulation tools has been driven in part by increased concern over safety and environmental hazards as well as the re-emergence of batch processing for speciality chemicals (Naess *et al*, 1993). Improvements in computer hardware have enabled this subject area to develop significantly. Most industrial bioprocesses are batch and therefore dynamic in nature.

The research carried out in dynamic simulation has focused primarily on equation-orientated simulation. This is in part due to better computation performance of such approaches. It is also because there is such a diverse array of dynamic processes, that it would be impossible to generate a complete library of dynamic models (Pantelides & Barton, 1993). This limitation removes one of the key benefits of sequential modular simulation. The focus of much research has been on packages that enable engineers to decouple model formation from the numerical methods that provide the solution (Pantelides & Barton, 1993).

The basis of the equation-orientation system is to collect all of the equations describing the flowsheet and to solve them as a large system of non-linear differential and algebraic equations. This can be expressed mathematically as: -

$$f_1(x, \dot{x}, z, u, t) = 0 \quad \text{Equation 2.1}$$

$$f_2(x, z, u, t) = 0, \quad \text{Equation 2.2}$$

where f_1 is the set of differential equations, f_2 is the set of algebraic relationships, x is the set of differential state variables, \dot{x} is the set of derivatives of the state variables, u is the set of algebraic state variables, z is the set of Control variables and t is time.

Aspen Custom Modeller (Aspentech, MA) provides a comprehensive set of facilities for performing dynamic simulation. The input language enables a complex set of differential algebraic equation to be solved including problems with dynamically changing control variables (Pantelides, 1988). However, the package is unable to model discrete changes that are a consequence of changes within the system (e.g. the rupture of a safety valve due to excessive pressure). Clarkson (1994) reported convergence problems when modelling a bioprocess made up of a mixture of dynamic and steady state models using SPEEDUP, the precursor to Aspen Custom Modeller. Such mixed mode operation is common in bioprocesses and therefore this represents a significant limitation of this package.

Research has lead to the development of gPROMS, an equation orientated simulation package that is able to model processes that are subject to a combination of continuous and discrete actions (Barton & Pantelides, 1994; Pantelides & Barton, 1993). gPROMS also enables the development of simulation using techniques to break the model down into sub-components and derive models from existing models using inheritance. They argue that such approaches have the capacity to simplify the process of building complex dynamic models and enable the reuse of existing models.

Most of the current generation of commercial bioprocess simulators have limited dynamic capabilities. This is because they generally have sequential modular structure. The sequential modular structure means that they can have predefined libraries of unit operation models. This makes them easier to use and enables the rapid development of whole process simulations. The next section looks at the commercial packages that have been developed for bioprocess simulation.

2.3.4 Bioprocess Simulation Packages

A significant amount of work has been carried out on bioprocess simulation since the mid-1980s, driven in part by the commercialisation of the first high-value bioproducts

(Petrides *et al.*, 1995). BioProcess Simulators™ was the first commercially available process simulation tool specifically designed for the biotech industry (Petrides *et al.*, 1996). The package was an extension of the established chemical process simulator ASPEN PLUS and used the infrastructure and facilities provided by ASPEN PLUS. BioProcess Simulators™ was able to carry out mass and energy balances and could be used to estimate the size and cost of equipment and economic evaluations of processes (Petrides *et al.*, 1996). The utility of the package was demonstrated through a published case study showing how it could be applied to the modelling of the manufacture of porcine growth hormone from *E. coli* (Petrides *et al.*, 1989). However, a number of chemical engineering characteristics made it difficult to apply Bioprocess Simulator to bioprocesses and therefore Aspentech discontinued technical support for it in 1998. Since then Aspentech has developed two computer packages suitable for the biotechnology industry, Aspen Chromatography and Batch Plus™, which are used for simulating chromatography and complex batch processes respectively.

BioPro Designer® was the second commercial bioprocess tool (Petrides *et al.*, 1996). This was based on research carried out at MIT. BioPro Designer was then developed further by Intelligen, Inc. Petrides *et al.* (1996) showed that this tool handles material and energy balances, equipment sizing and costing, economic evaluation, process scheduling and debottlenecking of batch operations. The package has an intuitive user interface making it easier to use and faster to learn.

Intelligen recently introduced a new version of their software called SuperPro Designer, which also included other packages such as EnviroPro Designer®, for design and evaluation of water purification processes and BatchPro, for batch chemical production. A description of the architecture of the package is given by Petrides (1994). Later publications illustrated the use of BioPro Designer for the evaluation of the production of biosynthetic human insulin (BHI) from *E. coli* (Petrides *et al.*, 1996) and the production of a monoclonal antibody (Petrides *et al.*, 1999).

Aspen Batch Plus and SuperPro Designer were evaluated by an industrial research group at Merck (Shanklin *et al.*, 2002). They used both packages to simulate a vaccine

manufacturing process. From this research, they concluded that Aspen Batch Plus and SuperPro Designer can successfully perform specific simulation tasks but cannot model of all phenomena occurring within a bioprocess. Nevertheless such packages are useful for process management, determining material and energy balances, answering scheduling question, and performing economic calculations. However, the critical limitation of the software was the inability to predict accurately the impact of scale-up and changing the process operating strategy. For such challenges it was deemed that more sophisticated models would be required. The next section looks an alternative strategy for bioprocess simulation.

2.3.5 Alternative Bioprocess Simulation

Research at UCL has been focused on using the knowledge gained from experimental investigations to generate more accurate models able to predict the effects of changes in upstream conditions. As the current generation of bioprocess simulators cannot easily be extended or adapted to include such models, researchers have built bioprocess simulations in scientific programming languages and general-purpose simulation packages. Such simulations can then be used for detailed analysis that would otherwise be impossible.

The first stage of this work is using experimental data to develop a model. A mathematical model describes how a system will perform under a particular set of operating conditions. By creating models of unit operations, engineers are able to examine process performance and assess the impact of changes on the process without having to perform time-consuming experimentation. Model generation typically requires considerable effort although this will be dependent on the type of model generated.

The simplest models to generate are **Empirical models**. These models are produced simply by finding a relationship between the inputs and outputs usually from experimental data. The standard empirical model can be in the form of a correlation, however other techniques that can also be used are neural networks (Chen *et al*, 2001) and multivariate statistics (Pate *et al*, 1999). The advantage of empirical models is that they are relatively easy to develop, as they are not based on physical reasoning.

Theoretical Models are based on a fundamental understanding of the underlying physical, chemical and biological process in operation. Such models can often be extended to allow for changes in the process as the impact of the underlying phenomena are understood. However, true theoretical models often take a greater effort to develop. This is particularly true in complex bioprocesses. For example, a true theoretical model of a fermentation process would require detailed knowledge of the internal kinetics of the cells. Most bioprocess models are therefore **semi-empirical**. Such models require less knowledge of the underlying science and do not require detailed understanding of all the physical mechanisms.

Many researchers investigating bioprocess modelling have chosen to focus on specific unit operations e.g. fermentation (Pascal *at al.*, 1995, Naroclaslowsky, 1991), or chromatography (Jungbauer and Kaltenbrunner, 1991). Review of some of these models used for common bioprocessing step can be found in Bailey and Ollis (2002). However, there are often interactions between unit operations, such that changing the conditions in one unit operation leads to a change in the output of a later unit operation. This means that models cannot simply look at unit operations in isolation and as a result the whole process needs to be modelled to take such interactions into consideration. One of the main problems encountered with generating such bioprocess models is the collection of experimental data from whole process runs. Traditionally such data has been obtained from pilot plant work. However, this approach requires a significant investment in terms of both time and money.

More recently there has been significant effort in the development of 'scale down' devices that can accurately mimic full-scale process equipment. Such devices enable engineers to mimic the results from processes under different combinations of experimental conditions. This additional data can be used to generate models of bioprocesses that are better able to capture the interactions between unit operations.

In order to obtain value from the models in tasks such as process design, the models need to be incorporated into whole process simulations. This enables the impact of the entire process to be judged. One limitation of this approach is that the current generation of commercially available bioprocess simulation packages cannot be

extended to include new models. Therefore more generic simulation packages and programming languages have to be used in order to develop accurate bioprocess simulations.

Early work at UCL used SPEEDUP (AspenTech, MA) a general-purpose simulation tool. Its capabilities for dealing with batch operations made it ideal for developing bioprocess simulations. Gritsis & Titchener-Hooker (1989) used SPEEDUP for modelling ultrafiltration, precipitation and centrifugation in order to determine the minimum processing time. Later Clarkson (1994) used SPEEDUP to model precipitation and centrifugation steps in the production of alcohol dehydrogenase from *Saccharomyces cerevisiae* and the production of beta-galactosidase from *E. coli*. However difficulty was experienced simulating processes with both steady state and dynamic models (Clarkson, 1994). Such mixed mode operation is common in bioprocesses and therefore represents a significant limitation of this package.

More recently specialist-programming packages have been used for developing models, in particular MATLAB (Mathworks, MA) and Labview (National Instruments, TX). Both packages, whilst requiring skill to program, are very user friendly and designed for scientist and engineers to prototype programs. Both packages contain multiple built in subroutines for solving dynamic equations and producing graphical outputs. This significantly reduces the programming effort. Recent work by Varga (1997) used MATLAB and spreadsheets to model recovery of a protein-engineered enzyme (*pe*-ADH).

Simulations have also been developed in Labview, a program primarily designed for monitoring and control applications. Labview includes its own graphical programming language (G Language) that can be used for developing simulations. This programming language also facilitates the design of graphical user interfaces, thus enabling simulations to be developed by non-programmers. Labview has been used for developing a simulation of the alcohol dehydrogenase process (Zhou *et al*, 1997; Zhou and Titchener-Hooker, 1999) and more recently for simulating the plasmid gene process.

This work has shown that with enough experimental data, simulations of bioprocesses can be developed. The results generated from such simulations may then be analysed to determine operating strategies and evaluate process designs. The next section looks at methods for analysing processes and their applicability to the biotechnology sector.

2.4 Process Analysis

2.4.1 "Windows of Operation"

One of the primary reasons for carrying out process simulation is to understand the whole process behaviour. The knowledge and the insight gained from running the simulation can be used for improved process design. One approach that has been used for developing operating strategies for bioprocesses is the so-called "Windows of Operation" (Woodley and Titchener-Hooker, 1996). This work defines a "Window of Operation" as the region on a graph defined by two process control variables, where all the specified performance levels, or constraints, are met. Through visualising this region the inter-play, between two control variables and their effect on the capability of the process to achieve its objectives, can be examined. The advantage of this approach is that it shows all the solutions that satisfy the design requirement rather than a unique solution, which maximises or minimises an objective function. Therefore an engineer can easily add on or change the requirements without having to solve an optimisation problem again. It gives an engineer the scope for decision-making based upon their empirical knowledge about the unit operations and the reliability of the underlying models. The other advantage of the visualisation approach over traditional optimisation is that it gives insight into the interactions in the process and the importance of process trade-off. The ability of "Window of Operation" to communicate this information in an intuitive manner is of critical importance in an industry where often engineers are in a minority (Titchener-Hooker *et al*, 2001).

However one limitation of this approach is that most bioprocesses have multiple control variables. Consequently visualisation of two control variables will not enable a process engineer to see all the trade-off between all the other control variables. Processes with n control variables will have an n -dimensional feasible region, defined

as the combinations of set points that meet the process constraints. This can be written as: -

$$R_{FR}(d, c) = \{z | g(d, z, c) \leq 0\}, \quad \text{Equation 2.3}$$

where R_{FR} is the feasible region, g is the set of inequality constraints, d is the set of design variables and c are the levels of the constraints.

A "Window of Operation" can only be used to represent a two dimensional subset of the feasible region. The problem of visualising processes with multiple control variables was first encountered by Zhou and Titchener-Hooker (1999), in which they applied "Windows of Operation" to visualise the alcohol dehydrogenase process. The "Windows of Operation" were only able to highlight the interactions between two control variables. In order to look at the interplay between three control variables a series of "Windows of Operation" were plotted, where a third control variable was changed. This highlighted how trade-offs can exist between multiple control variables.

Another technique for analysing multi-dimensional feasible regions was developed by Samsatli *et al* (2001) in their work looking at batch processes. In this work they used a hyper-rectangle inside the feasible region to define the ranges of the control variables (z). Typically batch processes are operated manually and therefore subject to significant variability in the control. Hence the focus of this research was determining a range that can be operated in. The approach uses the geometry of the hyper-rectangle to de-couple each of the control variables to determine a range for each control variable such that operating at any point within the ranges defined will result in the process meeting its constraints.

These both approaches are in stark contrast to approaches typically used in standard chemical engineering simulation. Typically the emphasis is on trying to maximise the performance of the process or improve process design to attain better economic performance rather than analysing a feasible region. Consequently the focus is often on the use of optimisation. This is discussed in greater detail in the next section.

2.4.2 Optimisation

Optimisation is commonly used in traditional chemical process design. Here optimisation problems for process design are formulated and algorithms are used to determine the best designs or operating strategies according to an objective. Generally optimisation can be viewed as improving the economic returns of a process. However, non-monetary objectives can also be used e.g. the reduction of the release of a pollutant (Edgar *et al*, 1999). Most commercially available chemical process simulators now have built-in optimisation tools.

According to Edgar *et al* (1999) there are a number of situations in a process where optimisation can be very useful. These include: -

- Sales limited by production, where optimisation could lead to higher production.
- Sales limited by market, where optimisation could focus on cutting operating costs.
- High raw material or energy consumption, which can be reduced through optimisation.
- Losses of valuable components through waste streams and therefore loss of potential revenue
- High labour cost, which is often a problem in the biotechnology industry. Optimisation can lead to reduced labour requirements and more efficient processing.

Most optimisation problems in process engineering have three components: -

- The objective function, which will be maximised or minimised. The objective function could be, for example, maximising profit or minimising costs.
- The equality constraints, which include all the mathematical relations for the material and energy balances and the physical laws such as rate equations.
- The inequality constraints, which are limits placed on the feasible solutions and typically include: material flow limits; equipment operating limits; environmental stipulations; safety constraints etc.

This can be expressed mathematically as: -

Minimise or maximise: $f_{obj}(d, z, x)$ Equation 2.4

Subject to: $h(d, z, x) = 0$ Equation 2.5

$g(d, z, x) \leq 0$ Equation 2.6

where f_{obj} is the objective function, h is the set of equality constraints, g is the set of inequality constraints, x are the state variables and z are the control variables.

The simplest type of problem will have a linear objective function and constraints. In such a situation the problem can be solved using Linear Programming (LP) techniques such as the simplex method. However, most chemical engineering optimisation problems will have both a non-linear objective function and constraints. Classically such problems are solved by Non-Linear Programming techniques such as Successive Linear Programming (SLP), Successive Quadratic Programming (SQP) or Generalised Reduced Gradient (GRG) (Edgar *et al*, 1999).

Both SLP and SQP use Lagrange multipliers to incorporate the equality and inequality constraints. SLP solves the problem using first order differentials whereas SQP uses second order differentials or quasi-Newton approximations. The GRG method does not explicitly use a Lagrange multiplier. Instead it uses the equality constraints to reformulate the problem into dependent (basic) and independent variables (non-basic) variables. It can also be adapted to deal with inequality constraint by using slack variables. The advantage of GRG over SLP and SQP is that the solution is guaranteed to be lie within the feasible region (rather than close to it); however GRG can be slower to converge.

There are a number of drawbacks to these techniques. The first is they can only guarantee global optimal solutions to problems with convex objective functions and constraints that form a convex region. This means that for the majority of optimisation problems such techniques can only guarantee a local optima. However the main disadvantage of all of these techniques is they require knowledge of the derivative. When the analytical partial derivative cannot be easily be obtained, approaches commonly used are finite different approximation methods or quasi-Newton (secant)

methods. These methods however are not as robust as similar implementations using analytical derivatives (Pantelides, 1988). This is one of the motivators for integrating automated algebraic manipulation into SPEEDUP (Pantelides, 1988).

Such approaches can be difficult to implement for sequential modular simulators where the simulation can be regarded as a 'black box'. In such situations the most robust approach is often to use direct search methods, which do not calculate derivatives but instead determine the search direction based on application of rules or heuristics (Biegler and Hughes, 1982). However this is not always efficient.

An example of a heuristic approach is the Nelder-Mead algorithm (Nelder and Mead, 1965). This algorithm searches for the minimum or maximum point by generating a simplex. With each move it determines a new point through a series of expansions, reflections and contractions. This has the advantage that it is relatively simple to implement and will usually find an optimum point. However the disadvantage of this approach is that it can be slow to converge. This method can be extended for dealing with constrained problems by using a penalty function. However, a large penalty function could lead to an ill-conditioned problem that is difficult to solve. Another limitation of the Nelder-Mead technique is that it is a greedy algorithm, meaning that it will follow the path that gives the quickest payoff, often finding a local optima (Press et al, 2002).

One solution to this problem is to use meta-heuristic optimisation techniques such as Simulated Annealing, which have the capacity to find global optima amongst many local optimum points. Simulated Annealing is so named because it is based on an analogy to the annealing of metals. In most optimisation techniques the optimisation routine will look for the best downhill move. However, the simulated annealing algorithm also allows uphill moves. The probability of the simulation routine making an uphill move will be proportional to the 'Temperature' of the optimisation routine. As the optimisation progresses the temperature is lowered until eventually only downhill moves are allowed. The theory behind this approach is that it will allow the routine to 'escape' local optima in the early stages of the optimisation. Using the analogy of the annealing of metals, this approach can be regarded as the slow cooling

of a metal. The other techniques discussed so far are the equivalent of quenching a hot metal.

The technique was first applied by Metropolis *et al* (1953) who developed the techniques for solving combinatorial optimisation problems. More recently a version of this algorithm was developed by Press *et al* (2002) that can be applied to continuous space. Their algorithm is an adaptation of the Nelder-Mead algorithm, where the values of the vertices of the simplex are modified to reflect the temperature in the system. This means that the simplex will occasionally move in the opposite direction to the objective function.

This algorithm was used by Cardoso *et al* (1996), who looked at the application of two variants of this algorithm to chemical engineering problems. They concluded that this algorithm performed significantly better than a standard simplex search for a variety of constrained optimisation problems. In a later paper Cardoso *et al* (1997) developed an adapted version of this algorithm capable of dealing with Mixed Integer Non Linear Programming problems.

In bioprocess engineering there has been a reluctance to use modelling and simulation tools. Titchener-Hooker *et al.* (2001) suggested that this might be due to the relative infancy of the approach and the lack of accurate models. These factors lead to a lack of confidence in the results produced by such approaches. This scepticism is not without basis; the optimum point in a constrained optimisation problem is likely to lie on the boundary of the feasible region. This means small variations or inaccuracies in the process performance will result in the constraint being broken. Furthermore, as Edgar *et al* (1999) point out, the solution generated by optimisation can only be as accurate as the model it is based on.

By contrast, “Windows of Operation” provides a visual output, which can be used by an engineer to both assess the accuracy of the underlying model and determine a robust operating point given the level of variation that they would expect. In chemical engineering, where simulation is more widely used and accepted, a number of approaches have been developed for solving the robustness problem. The next section

looks at methods that have been applied to evaluating the robustness of processes subject to uncertainty. These methodologies enable the engineers to evaluate the process and determine optimum designs and operating strategies given the uncertainty in the process.

2.4.3 Robustness

There are many reasons why process simulations may not accurately represent process behaviour. The models of the unit operations in the process could be based on noisy data, making parameter estimations inaccurate. Alternatively the process may be subject to variations in the operation conditions that are not accounted for in the model. For example the model might not account for seasonal changes in the composition of a feed-stock and its impact on process performance.

The performance of a process is determined by the equality constraints, which define the mass and energy balances within the process. Most processes are also subject to inequality constraints that define limits on the process performance. These equations are functions of the design variables, the control variables and the state variables. However, if parameters within the process vary, equality and inequality constraints will also be functions of the parameter values (θ).

$$h(d, z, x, \theta) = 0 \quad \text{Equation 2.7}$$

$$g(d, z, x, \theta) \leq 0 \quad \text{Equation 2.8}$$

where θ is the set of parameter values

By assuming that the state variables are determined by the equality constraint equation 2.8 can be simplified to: -

$$\begin{aligned} x &= h(d, z, \theta) \\ \Rightarrow g(d, z, h(d, z, \theta), \theta) &\leq 0 \\ \Rightarrow g(d, z, \theta) &\leq 0 \end{aligned} \quad \text{Equation 2.9}$$

where h is the set of function for the state variables (based on the equality constraint)

In situations where parameters can vary, the output of the process will also vary meaning the location of the inequality constraint will change. One approach adopted in early work was to assume the parameter will vary within a range T , where $T = \{\theta | \theta_L \leq \theta \leq \theta_U\}$. Consequently an operating point was considered robust when all the constraints were met for all the parameter values. As the constraints are written with maximum values of zero, an operating point will be robust if the largest (most positive) constraint under all the possible parameter values within the range T is less than zero. This can be written as: -

$$\max_{\theta \in T} \max_{i \in I} g_i(d, z, \theta) \leq 0 \quad \text{Equation 2.10}$$

where I is the set of inequality constraints $\{i | 1, \dots, i_{\max}\}$ and i_{\max} is the number of constraints

However with chemical engineering problems this approach leads to conservative designs as it assumes that the control variables cannot be adapted to compensate for variations in the parameters. Grossmann and Sargent (1978) assumed that the control variables could be adjusted in light of a variation in process parameters. They proceeded to optimise the design variables on the basis that for any design there must be a set of control variables that will result in the constraints being met for any combination of parameters. Halemane and Grossmann (1983) showed that this can be expressed as: -

$$\forall \theta \in T \{ \exists z (\forall i \in I [g_i(d, z, \theta) \leq 0]) \} \quad \text{Equation 2.11}$$

Equation 2.11 states that for all potential values of the uncertain parameters (θ) within the range T , there must be a set of control variables (z) where all the constraints (g) are met. Halemane and Grossmann (1983) then showed that it was equivalent to: -

$$\max_{\theta \in T} \min_z \max_{i \in I} g_i(d, z, \theta) \leq 0 \quad \text{Equation 2.12}$$

This states that within the worst combination of variable parameters, there must be a set of control variables, which ensures that the most positive or 'worst' constraint is still satisfied.

Halemane and Grossman (1983) adapted this approach so that it could be used as a means to determine the feasibility of a design. The measure of feasibility (ψ) was calculated by determining the values of the largest constrained variable under different parameter conditions.

$$\psi(d, \theta) = \min_z \max_{i \in I} g_i(d, z, \theta) \quad \text{Equation 2.13}$$

where ψ is a measure of the feasibility of a design at a given set of parameters

For a design to be feasible under the specified set of the variable parameters (θ), ψ must be smaller than or equal to zero, as values greater than zero would mean that at least one of the constraints had been broken. Generally the lower (more negative) values of ψ correspond to more combinations of control variables that result in feasible process performance. However, feasibility studies do not directly address the issue of robustness.

This work was later built upon by Swaney and Grossman (1985) who looked at the flexibility of a design. They started by examining the region of flexibility defined as the combinations of variable parameters where control settings exist that result in the constraints being met. (See Figure 2.1): -

$$R_{flex} = \{\theta | \exists z | g(d, z, \theta) \leq 0\} \quad \text{Equation 2.14}$$

Within the region of flexibility lies a hyper-rectangle (T_{flex}) (see figure 2.1). The use of the hyper-rectangle enables each of the parameters to be decoupled and be defined as having upper and lower limits within which the process constraints can be met: -

$$T_{flex} = \{\theta | \theta^L \leq \theta \leq \theta^U\} \quad \text{Equation 2.15}$$

Such that: $T_{flex} \subset R_{flex}$ Equation 2.16

where R_{flex} is the Flexible Region and T_{flex} is a hyper-rectangle in the flexible region

The size of the hyper-rectangle is calculated by evaluating the allowable deviation within each parameter from a nominal fixed set of parameters (θ_N). This approach differs from earlier work on robustness as the ranges of the parameters are calculated from the constraints rather than being fixed. The flexibility of the design to cope with changes in the parameter values can then be calculated from the size of the largest hyper-rectangle within the constraints. This can be expressed as: -

$$f_{Flex} = \max \delta \quad \text{Equation 2.17}$$

Subject to: $\max_{\theta \in T(\delta)} \min_z \max_{i \in I} g_i(d, z, \theta) \leq 0$ Equation 2.18

$$T_{flex}(\delta) = \{\theta | (\theta_N - \delta \Delta \theta^-) \leq \theta \leq (\theta_N + \delta \Delta \theta^+)\} \quad \text{Equation 2.19}$$

where f_{Flex} is a measure of the flexibility of the process, θ_N is the nominal set of parameter values, $\Delta \theta^+$ and $\Delta \theta^-$ are vectors of expected parameter deviations and δ is the scaled parameter deviation.

This approach provides a useful indicator on the ability of processes to cope with variations of different combinations of parameter values and gives ranges for each of the parameters. However the approach can only be applied by engineers with a good understanding of the significance of the each of parameters.

If the uncertain variables can be assigned a probability distribution then the expected values and the variance can be calculated for the process outputs: -

$$\bar{y} = E_{\Theta}(f_u) = \int_{\Theta} f_u(\theta) \cdot j(\theta) d\theta \quad \text{Equation 2.20}$$

$$\sigma_y^2 = E_{\Theta} \left\{ (y - \bar{y})^2 \right\}$$

where \bar{y} is the expected or average value, $E_{\Theta}(f)$ is the expectancy function, $f_u(\theta)$ is the function defining process performance, subject to uncertainty, $j(\theta)$ is the joint probability density function for the uncertain parameters and Θ represents the entire parameter space.

This can be used to generate optimisation problems based on the expectancy function. It also enables the formulation of constraints based on the statistical moments. Bernardo *et al.* (2001) suggest a number of possible constraints such as constraints on the variation in a process and the expected value. Additionally, constraints can also be generated to reflect the probability of a constraint being broken: -

$$\bar{y} + \Phi^{-1}(\gamma)\sigma_y \leq 0 \quad \text{Equation 2.21}$$

where γ is the desired probability for a parameter and Φ^{-1} is the inverse normal distribution.

Using such techniques enables the user to determine an optimum point given an 'acceptable' risk of the constraints being broken. In many respects this reflects the situation in biochemical engineering where the objective, when selecting control variables, will be to maximise the productivity whilst ensuring critical quality constraints are met. However, one limitation of this approach is that it assumes the output variable will be approximately normally distributed.

2.4.4 Multi-Dimensional Integration

In order to carry the type of analysis proposed in the previous section, techniques are required to evaluate multi-dimensional integrals. A number of algorithms exist for evaluating such integrals and many have been applied to chemical engineering problems. One approach that is commonly used is quadrature. In this approach the

integration of each dimension is solved separately. In a case with two dimensions this would mean calculating the integral of one of the variables and then using the return values from this function as the basis for the integration with the second variable.

The work by Bernardo *et al* (2001) uses Gaussian quadrature. In such an approach the algorithm determines the integral by determining both the position of the interior points and their relative weights. This can give significantly improved performance particularly if the integrand is the product of two polynomials (Press *et al*, 2002). However, the limitation of this approach is that generally the number of points that need evaluation will increase exponentially with the number of dimensions in the problem (Bernardo *et al*, 2001).

An alternative strategy is to use Monte Carlo integration, which is based on sampling. If a complex shape is known to reside in a multi-dimensional space then the size of the shape can be estimated by selecting random points in this space and determining whether these points lie within the shape. The advantage of this approach is that it can easily be applied to complex regions defined by mathematical formulae that cannot be integrated analytically. Additionally the sampling technique can be adapted to reflect specific properties of the integral by choosing more sample points in a region that is likely to be of more importance.

An example of such an approach is demonstrated by Press *et al* (2002). This shows how Monte Carlo integration can be used to calculate the size of a region of a truncated torus. They then showed how a theoretical weight can be calculated given a variation of density in the object by skewing the sample set to denser regions that are likely to have a greater impact on the overall weight.

In integration problems such as the one described in Equation 2.20, the distribution of the uncertain parameters will be known. Hence the sampling set can be skewed to sample from the parameters that are more likely to occur. For normally distributed variables this can be achieved using the Box-Muller technique, which calculates random normally distributed variables (Press *et al*, 2002), or an inverse normal distribution function (Acklam, 1999).

In theory, the Monte Carlo technique is completely scalable with increasing dimensions as the number of sampling points could stay constant. In practice, the error on such a calculation is given by: -

$$err = \frac{\sigma}{\sqrt{m}} \quad \text{Equation 2.22}$$

where σ is the variance, and m is the number of sampling points.

Consequently in order to obtain increased accuracy a larger number of simulations is required. As equation 2.22 shows, increasing the number of samples (m) will decrease the error by $m^{-1/2}$. The error also has an impact of the scalability of the technique with a larger number of dimensions. This is because increased variance would be expected when a larger number of dimensions is used.

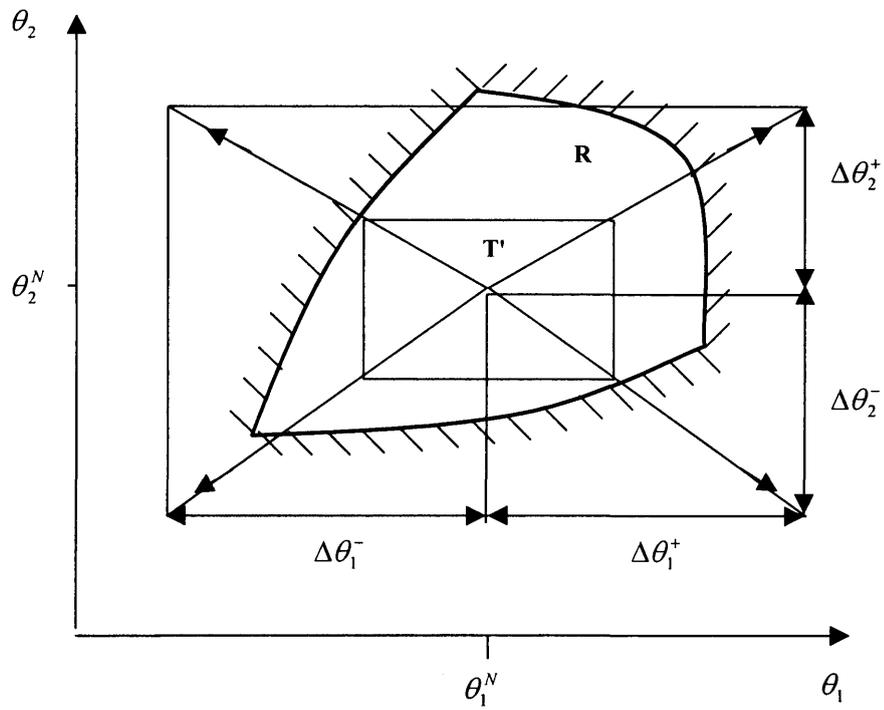
Another mechanism that has been reported in the literature is the use of quasi-random numbers in Monte Carlo simulations. Unlike 'true' random numbers, quasi-random numbers are generated to fill uniformly a multi-dimension cube. An example of such a sequence is the Sobol algorithm. A description of this algorithm and its performance is given by Bratley and Fox (1988). The advantage of this approach is that the error of the integral is proportional to $\log(m)/m$, meaning that more accurate estimates of a feasible region are obtained for less computational effort.

2.4.5 Application to Bioprocesses

Classical optimisation would not be suitable for analysing biochemical engineering processes due to the limitations in both the models and the process control. However, even robust optimisation techniques are not particularly useful for analysing the nature of the interactions in a process as they do not yield an intuitive output. The focus of the work reported in this thesis will be to study the use of using such techniques in conjunction with visualisation tools to determine more intuitive means for analysing bioprocess performance.

The next chapter looks in more detail at simulation development, including the use of different programming packages. A case study will be used as the basis for the work and is built upon an extension of that used by Zhou *et al* (1997) with a fermentation model included so as to provide a complete sequence for analysis.

Figure 2.1: Maximum scaled hyper-rectangle within the feasible region
(Swaney and Grossmann, 1985).



3 Bioprocess Case Study and Simulation Development

3.1 Abstract

This chapter looks at the development of a bioprocess simulation as the basis for an investigation into methods for determining bioprocess operating strategies. The objectives of the work were to develop a simulation, using an appropriate platform, capable of capturing process interactions and based on models in the literature. A secondary objective was to design the simulation so that it could be easily maintained.

Simulations were built using four platforms: SuperPro Designer, Labview, Matlab and C++. SuperPro Designer was the simplest to use but was unable to capture the interaction between the unit operations. Consequently the focus was directed to more generic programming languages such as Matlab and C++. Of these Matlab had the advantage of being simpler to use whereas C++ offered certain performance advantages.

Another consideration was the architecture of the simulation. In this work, modules were built for each unit operation and a common data structure was defined to hold all data on the material flowing between the unit operations. In Matlab and Labview the unit operations are defined using a series of functions for each unit operation whereas in C++ this was done using objected orientated design.

In order to build an accurate whole bioprocess simulation, it was necessary for new models to be generated. A fermentation model was built and a first-order degradation model was included to predict the impact of proteolytic enzymes on the product. The chapter concludes with an examination of the results that can be obtained through such simulations and highlights the need for improved forms of data visualisation in order to capture fully the insights contained within such datasets.

3.2 Introduction

The process of developing a simulation model will be heavily dependent on both the objectives of the simulation and the resources available. Simulations can be used to solve many different processing problems. In this work the objective is to understand the impact of interactions in processes and how they will affect the selection of an operating point in terms of the ability to achieve predefined performance metrics i.e. productivity, yield, etc.

Previous bioprocess simulation work has looked at two different approaches. These were using bioprocess simulation software or producing bioprocess simulations using programming languages and generic simulation tools. Bioprocess simulation packages have the advantage of ease of use. This is often advantageous if an organisation does not have programming skills or time and resources to develop such a skill set. However, as reported by Shankin *et al* (1999) such simulations are less suited for accurate simulation.

Bioprocess simulations can be developed using programming languages and generic simulation packages. Such approaches enable the use of unit operations models that can capture process interactions (Clarkson *et al.* 1993; Bulmer *et al.* 1996; Zhou *et al.*, 1997; Zhou and Titchener-Hooker, 1999). This is because these packages give developers the freedom to develop the simulation in the way they want. However, such packages often require either expert knowledge or time to master and many organisations may not have resources available for this. Additionally, simulations developed in such packages need to be designed and implemented in such a manner than they can be easily adapted or modified by other users.

In this chapter, a number of different approaches for developing a simulation for a typical bioprocess are examined. The aim was to obtain a simulation using available knowledge to predict the interactions between control variables. The next section looks in more detail at the process selected as the case study. Section 3.4 looks at the merits of different simulation packages and languages and section 3.5 examines how to ensure the resulting model can be maintained. Section 3.6 looks at some of the additional model development work carried out in the course of this PhD.

3.3 Case Study

The process that was selected as the case study was based on the use of a yeast to produce alcohol dehydrogenase. Its production features many manufacturing problems typically associated with labile intracellular enzymes, making it a good model system. Extensive work has already been carried out on unit operations comprising the alcohol dehydrogenase process. The process that was simulated is shown in Figure 3.1.

The process starts with 100L fed-batch fermentation of *Saccharomyces cerevisiae*. The fermentation follows the protocol described by Gregory et al (1996). The fermentation finishes when the fermenter reaches its full working volume, in this case 70L. After the fermentation finishes it was assumed that enzyme product was subject to first order degradation caused by the action of proteolytic enzymes. This is a major reason why processing has to be rapid in order not to sacrifice yields.

The broth from the fermentation then passes directly from the fermenter into a centrifuge. The simulation was based upon a disk-stack centrifuge (CSA1, Westfalia) with an equivalent settling area of 1465m² calculated using Stokes law. Previous work by Bulmer *et al.* (1996), Clarkson (1994) and Clarkson *et al* (1996) has shown that the performance of the centrifuge can be modelled using the grade-efficiency model. Details of this approach are described in Appendix A.

The cell paste from the centrifugation step is then moved into a storage vessel. It is then re-suspended by mixing it with buffer solution at a concentration to 450g/L. This solution is passed through a homogeniser, which ruptures the cells releasing product and producing cell debris contaminant. Previous work has shown that homogeniser operation affects both the release of product and the properties of the cell debris (Hetherington *et al*, 1971; Siddiqi 1996).

The resulting homogenate is then passed into a vessel where it is re-suspended using a buffer. This is then put into second disk stack centrifuge modelled using the same approach as described previously. The objective of this centrifuge is to remove the

cell debris. The efficiency of this process will depend on the properties of the homogenate. Efficient debris removal is a pre-requisite for later purification steps.

After these steps the volume of product is reduced using two-step precipitation before the ion-exchange and gel-filtration steps. These latter steps were not included in the simulation as there was limited information on their performance to model them successfully.

A simulation of the process had already been developed in Labview (Texas Instruments, TX), a package used for laboratory automation and basic calculations (Zhou *et al*, 1997; Zhou & Titchener-Hooker, 1999). However there were a number of limitations in the package, the architecture of the simulation as well as the underlying models used in this simulation. Consequently a specific simulation was built for this work with new models to capture more of the process and better architecture to enable future editing. The remainder of this chapter looks at the packages used, simulation design and additional modelling work carried out to develop the overall process simulation.

3.4 Simulation Packages and Languages

3.4.1 Introduction

When building a simulation the first problem is to determine which package is most suitable for solving the problem. Ideally a package needs to be easy to use. This serves two critical purposes. An easy to use package will cut down the time to produce a model as well as enabling the model to be passed on to other users. However, often ease of use comes at the expense of the flexibility necessary to adapt a simulation to fit an exact set of requirements.

The package used previously, Labview, had a number of limitations and therefore it may not be the best choice of package for such simulation development. These limitations are discussed in greater detail later in this section. As a result of these limitations, simulations were built in a number of different packages using a range of simulation languages, including specialised bioprocess simulation tools, and fundamental programming languages. This section looks at the relative merits of the

different approaches that were used, explaining both their advantages and drawbacks. The section finally looks at the development work carried out for this thesis.

3.4.2 SuperPro Designer

Probably the simplest method for producing a bioprocess simulation is to use SuperPro Designer (Petrides, 1994; Petrides *et al.*, 1995; Petrides *et al.*, 1996). This commercial package offers a relatively intuitive graphical user interface. The package enables the user to select the unit operations they wish to include in a simulation from menus as well as the materials they are using for the mass balance. This enables the user to generate a simple flowsheet of the process.

After this is completed the user can select the parameters for each of the items of equipment and run a mass balance. The mass balance is carried out using a series of built in models for each of the items of equipment. Additionally, the package can also be used to perform relatively simple scheduling and basic economic calculations. [Both of these are outside the scope of the current study.]

However the big disadvantage of this package is that there is no easy method to extend it to incorporate more sophisticated models of unit operations. For example, the model used for homogenisation does not incorporate any analysis of the size distribution of the debris. Instead it assumes the debris will have a fixed size, which needs to be defined by the user. Consequently the model is unable to give a realistic assessment of how well the debris removal centrifuge will work as the feed input changes with the homogenisation conditions selected. In addition, work by Siddiqi *et al* (1995) and Siddiqi (1996) showed that fermentation conditions would have a major effect on cell wall size and strength. The current set of models in Superpro Designer is unable to take such factors into account. To incorporate new models would involve either reverse engineering the product or obtaining the source code. This is impractical and consequently SuperPro Designer was deemed to be an unsuitable package for this work, principally because of its limited flexibility and adaptability.

3.4.3 Labview

The main focus of the package is connecting measuring and control devices to a PC to automate experimentation and to enable a user to log data from such experiments

(Labview Starter Guide). Labview has the advantage in that the programming is done through a graphical interface and therefore it does not require a high-level of skill to program. Each variable is represented by a wire and a series of blocks can be used to represent mathematical operations such as addition or subtraction. The other advantage of this package is that the programming interface automatically generates a graphical user interface. This means a relatively inexperienced user can build a simple application that is easy to use. The interface that is built can then be adjusted and configured by the developer.

However, the main problem with this package is actually the graphical programming language. When handling a large number of variables the resultant visual presentation of the model rapidly becomes complex and can get very confusing very quickly, particularly if the code is not structured well to start with.

3.4.4 Matlab

Matlab (Mathworks, MA) is a language primarily designed for engineers and scientists to develop and prototype simulations. In Matlab the user creates a text file containing the basic Matlab script that they wish to run. The code in the text file is interpreted by the Matlab engine at runtime. The advantage of such a text file is that it is easier to analyse such code than the diagrams produced in Labview. Additionally the Matlab package also contains extensive debugging tools that allow the code to be stepped through to spot errors.

Matlab is based around matrices and has built in functions for matrix arithmetic. This makes programming linear systems very simple. Additionally it also contains a number of easy to use visualisation functions for analysing the results of the simulation. The power of Matlab can be extended by using a series of toolboxes, which provide additional functions for solving specific problems (e.g. the Neural Network Toolbox and the optimisation toolbox).

One disadvantage of Matlab is that it does not lend itself particularly well to object-orientated programming. Although there are some basic capabilities in this area they are not particularly easy to use. This makes it hard to develop a simple modular

architecture. The main disadvantage of Matlab is that it can be quite slow, particularly when dealing with code that contains loops. One way of avoiding the generation of inefficient and slow code in Matlab is to replace loops with matrix calculations where possible. However when modelling non-linear systems this is not always possible.

3.4.5 C++

C++ is a modification of the C language to allow for object-orientated programming. C++ on first appearance would appear to solve some of the problems with Matlab. It can be used for object-orientated programming and therefore it is possible to develop a simple modular structure. Equally this work used Visual Studio 6 (Microsoft, WA) that includes an Integrated Development Environment which offers a number of features that make it easy to analyse the code.

Additionally, as it is a relatively low-level compiled language, faster performance would be expected from C++ than a language such as Matlab, which relies on an interpreter. However, the disadvantage of this is that C++ code can be harder to write and therefore requires a greater level of expertise. Another limitation of C++ when compared to Matlab is that it is difficult to visualise the results using a graph. To do so would require the simulation to be linked to a suitable graphical library. This would mean either finding a suitable graphical library or developing one, neither of which is a trivial challenge.

3.4.6 Development of the Simulations

Initially a new simulation was built in Labview. The objective of this was to remove some of the problems with the previous implementation through a more coherent design. This ensured that the simulation could be amended and updated later to allow for the impact of variation. Although a better design was able to reduce some of the confusion associated with the previous implementation, the resulting code was still difficult to understand.

A simulation was then developed in Matlab. This had the advantage that the code was stored in a series of text files that could be easily examined. In order to optimise the code a number of calculations were carried out using matrices. However, in the bioprocess used for the case study, there are many non-linear equations making it difficult to do many calculations using matrices. Moreover, much of the work carried out in this thesis relies on running the simulations many times within a high level program. Such repeat simulations need to be carried out using a loop, which is slow and results in a long runtime.

Consequently another simulation was developed in C++. The idea was to develop a simulation with a faster runtime enabling more advanced analysis. In order to compare the speeds of the two simulations they were each run 10,000 times using random numbers for the control variables. The tests were run on a laptop with a 2.66GHz Pentium 4 processor running Windows XP. The results for the simulations were: -

- C++ simulation - approx. 20secs
- Matlab simulation - approx. 90secs.

This indicated that the C++ was capable of fast performance, however, this was still slower than required for later analysis. Further analysis revealed that the slowest step in the C++ simulation was the code used to calculate the amount of solids collected by the centrifuge. The proportion of solids removed is calculated using the equation 3.1 below: -

$$F_{ss} = \int_0^{\lambda_{pU}} \phi(\lambda) \cdot \tau(\lambda) d\lambda \quad \text{Equation 3.1}$$

where F_{ss} is the amount of solid being collected in the sediment, λ the particle size, λ_{pU} is the largest particle size, $\tau(\lambda_p)$ is the solid of size (λ_p) collected according to the grade-efficiency model and $\phi(\lambda)$ is the distribution density of solids of a particular size.

In the Matlab simulation this calculation was done using vectors. By contrast there is no mechanism for handling vectors in the basic C++ language. Consequently the calculation was done using a 'for' loop. In order to speed up the C++ simulation, the integral in Equation 3.1 was evaluated using an algorithm demonstrated by Press *et al* (2002). The algorithm evaluates the integral by: -

- Calculating the integrated function at a series of evenly spaced points.
- Uses these values to calculate an approximate value of the integral.
- Determines the mid points between the previous sets of points.
- Uses these values in conjunction with the previous estimate of the integral to calculate a new estimate of the integral.
- It continues these steps until the difference between the two previous estimates of the integral is small.

In this work the integration was treated as having converged when the relative difference between two successive evaluations was $<0.1\%$. This procedure cut the simulation time significantly. This version of the simulation was tested, by calling the simulation 10,000 times; the time taken could not be measured as it was so short. Consequently a second test was performed using 100,000 points and this took 12secs.

In theory, reducing the number of points should also speed up the Matlab simulation. However the specific approach suggested by Press *et al* (2002) will not result in faster Matlab code. This is because the algorithm uses a loop and generally in Matlab vectors should be used instead of loops in order to attain optimal performance (Matlab User Guide, 2000). To investigate this further a second test was run using a modified version of the Matlab code with the number of elements in the vector reduced by a factor of twenty. However, interestingly when the test was performed the code ran at the same speed as the previous version of the Matlab simulation (90secs to complete 10,000 simulations). This result indicates that there is internal logic in Matlab that means that the processing speed is not directly proportional to number of instructions contained in the code.

As well as differences in the approaches used for the underlying calculations, the different languages needed different approaches to the design and architecture of the

simulation. The next section describes in more detail the architecture of each simulation.

3.5 Simulation Architecture

3.5.1 Introduction

In this work, a sequential modular simulation was built since this allows the complexity of each of the unit operation to be contained in discrete modules. One of the main challenges when developing such a simulation is to write the code in such a manner that it can easily be adapted and developed further at a later date and ideally by another user. The key to this is designing the code in such a way that it can be easily extended. This is typically done by breaking code down into logical units using data structures, functions and classes.

The simulation developed by Zhou and Titchener-Hooker (1999) was also sequential modular in nature. However, the simulation was simply a collection of procedures linked by variables and the code was not structured into logical units that corresponded to the unit operations. Equally, the inputs to the procedures were not always logical. For example, a procedure used to calculate the output of the homogeniser took the fermentation growth rate as an input. This made understanding the simulation difficult and meant that it could not easily be extended to incorporate additional unit operations or even adjust existing operations.

In this PhD the streams were modelled using a complex data structure designed to hold all the parameters that defined the properties of the liquid flowing from one unit operation to the next. [Details of the data held in these streams are given in the next section.]

Two different approaches were then used to model each of the unit operations. For the Labview and Matlab simulations the approach used was to define each unit operation as functions. The C++ simulation used classes to represent each of the unit operations. More details of this object-orientated approach are given in section 3.5.3.

3.5.2 Modelling the Streams

Details on the process streams were held in complex data structures containing the necessary information on the material flows and physical properties of the streams generated in the preceding unit operation. These were used throughout the simulation. A common stream data structure simplified the programming, since a universal data structure was passed between unit operation rather than sets of unstructured data. The resulting code was easier to understand. As the same data structure was used for each stream throughout the simulation, unit operation modules could also be re-used whenever a unit operation occurs more than once in a process.

In bioprocesses, components can exist in multiple phases. For example a protein can be intracellular, in aqueous solution, in organic solution or in a solid precipitant phase. This is analogous to bulk chemical processes where volatile components can often be in either liquid or vapour phases. The phase that a component is in will have an effect on how it is processed by a particular unit operation. For example a protein precipitate can be collected by a centrifuge whereas proteins in solution cannot. To capture this information, the stream data structure contained a series of component structures storing information on each component as follows: -

- intracellular phase
- precipitant phase
- total (in all phases)
- concentration of each component

The stream did not include a separate field for components in the aqueous phase and consequently all components are assumed to be in this phase by default. In the alcohol dehydrogenase simulation, five components were tracked through the processes; alcohol dehydrogenase (units), total protein (g), nucleic acids (g), cell wall/ cell debris (g) and whole cells (g). In the actual process there will be many more components present, however, these components provide an adequate description of the process.

The remaining fields in the stream structure store the physical properties of the stream and its contents. The physical properties of the stream may determine how later unit operations will perform. For example, increased viscosity will hinder a later

centrifuge separation step. The physical properties are often the underlying cause of interactions between unit operations. The following fields were included in the stream definition to characterise different aspects of the streams physical properties: -

- liquid properties – containing viscosity and density of the liquid phase.
- solid properties – storing average size and size distribution of the particles, the concentration and density of the particles.
- other properties – holding details such as cell wall strength and protein degradation rate.

Finally the streams also contained details of the volume of material per batch and the time spent to get to that point in the process. A summary of the structure of the stream data is given in Table 3.1, which shows the fields used in the data structure and the data stored within each of the fields.

3.5.3 The Unit Operations

Both Labview and Matlab have only a limited capacity for object-orientated coding. Therefore a series of functions were defined for each of the unit operations. These functions were designed to take the control variables and the input stream as arguments and generate output streams.

There are many drawbacks to this approach. In particular, functions that take multiple parameters can become difficult to use, as there is a danger that parameters will be entered in the wrong order. Equally, related functions cannot easily be grouped together to use a common set of data.

Such problems can be solved using object orientated programming. Object-orientated programming allows a programmer to define a class, which describes the data and functions associated with an object. The class definition is used as a template for creating objects. For example, a fermentation class could be created and used to create two fermentation objects. These objects could be the same or could have different properties (e.g. different growth rates).

There are many advantages to this approach. It means that all the data associated with one item can be stored with the functions associated with them. For example, an object can be used to store the necessary parameters for a particular unit operation thus ensuring that it is not necessary to pass these parameters as a long list of arguments into a function. Equally, functions may be grouped together in a logical fashion, hiding functions that will not be needed elsewhere and enabling functions to work with the data stored in the object.

The C++ simulation had an object-orientated structure with classes defined for each unit operation. Each unit operation class was designed to store the control variables and have accessor functions so that these values could be updated. Additionally they store the memory address of their respective input and output streams.

3.6 Model Development

3.6.1 Introduction

The utility of a simulation for analysing a process will depend on the level of detail within the models. At the simplest level stoichiometric models can be used to generate mass balance predictions based on simple assumptions. Such models have the advantage that they are simple to develop. Typically however such models are not suitable for detailed analysis. More detailed models based on theory and observations have the advantage that they can be used for much more detailed analysis. The drawback is that they will require more time and effort to develop.

The alcohol dehydrogenase process used as the case study in this work has been studied previously. The earlier work of Titchener-Hooker and Zhou, 1999 did not include a fermentation simulation. Instead a lookup table was used that simply stored values taken directly from experimental work. Equally the simulation did not incorporate the degradation of the alcohol dehydrogenase caused by the presence of proteolytic enzymes in the yeast. In order to achieve a better simulation, new models were developed. The next section gives details on these models, the sources used and

assumptions that were made. Details on these and other models used in the new process simulation are given in Appendix A.

3.6.2 Fermentation

The fermentation model was designed to capture the pertinent aspects of cell growth, product formation, as well as the properties of the cells. Of particular interest in this simulation were cell size and cell wall strength. Previous experimental work had shown that growth rate has a significant impact on the cell properties of *Saccharomyces cerevisiae* (Siddiqi *et al*, 1995) as well as the expression levels of different intracellular proteins (Gregory *et al*, 1996). Growth rate will affect the fermentation time, which has a critical impact on the production rate and process economics.

In this work a fermentation model was developed based on the experimental protocol described by Gregory *et al* (1994). In this work the feeding strategy used was based on the Wang-Cooney model (Wang *et al*, 1979), in which the feed rate is exponentially increased so as to ensure a constant growth rate (Equation 3.2): -

$$Q_{fed} = \frac{\mu C_X V}{C_N Y_{x/s}} \varepsilon = \frac{\mu C_{X0} V_0 e^{\mu t}}{C_N Y_{x/s}} \varepsilon \quad \text{Equation 3.2}$$

where Q_{fed} is the feed rate (L/hr), ε is the respiratory quotient, C_N is the concentration of substrate (g/L), C_X is the concentration of biomass (g/L), C_{X0} is the initial concentration of biomass (g/L), $Y_{x/s}$ is the yield of biomass on substrate (g/g), V is the Volume in the fermenter (L), V_0 is the initial volume in the fermenter (L) and μ is the Growth rate (h^{-1})

From previous chemostat experimentation, the yield of biomass on substrate for *Saccharomyces cerevisiae* was found to vary with growth rate (Gregory *et al*, 1996). Growth rates greater than 0.2h^{-1} resulted in a significant drop in yield. This was attributed to a switch to oxido-reductive growth when placed in a glucose rich environment. A correlation was developed in this work to capture the drop in the yield (Equation 3.3). The correlation included a hyperbolic tangent term to capture the

transition to oxido-reductive growth. The correlation's coefficients were calculated using a Matlab program using the Nelder-Meld simplex algorithm: -

$$Y_{x/s} = 0.3608 - 0.0742\mu - 0.4471\mu^2 - 0.144 \tanh(107.65(\mu - 0.2005))$$

Equation 3.3

Equations 3.2 and 3.3 enable the feed rate to be calculated for a range of different fermentation growth rates. This in turn can be used to calculate the volume of liquid present in the fermenter at any given time if it is assumed that the fermentation will proceed until the vessel reaches its full working volume. The fermentation time can be calculated as below: -

$$t = \frac{1}{\mu} \ln \left(\left(\frac{V_f}{V_0} - 1 \right) \cdot \frac{Y_{x/s} C_N}{C_{x0} R} + 1 \right)$$

Equation 3.4

This shows that the time taken is inversely proportional to the growth rate. However, both the growth rate and time will affect the level of biomass production. Hence equation 3.5 can be formed using the equation above: -

$$C_{x_f} = \frac{Y_{x/s} C_N}{\varepsilon} \left(1 - \frac{V_0}{V_f} \right) + \frac{V_0 C_{x0}}{V_f}$$

Equation 3.5

In addition to the amount of biomass, the simulation also needed to consider other factors. These include the amount of various components in the cells and the physical properties of the stream. Previous work by Gregory *et al* (1996) and Siddiqi (1996) examined the impact of growth rates on enzyme expression and cell properties. In most cases it was found that there was a transition when the cells entered oxido-reductive growth. Therefore correlations were developed that also included a hyperbolic tangent function. Parameters for each of these correlations were then estimated using a Nelder-Mead routine. Using this approach, correlations were developed for showing the relationship between growth rate and the following fermentation properties: -

- cell size

- cell breakage coefficient
- cell breakage exponent
- protein release coefficient
- ratio of protein to dry cell weight
- ratio of alcohol dehydrogenase to protein

Details of these correlations are given in Appendix A along with assumptions used to build the simulation model.

3.6.3 Enzyme Degradation

An additional feature of the simulation developed in this work was the inclusion of an enzyme degradation coefficient. This was included as experimental evidence suggests that alcohol dehydrogenase from *Saccharomyces cerevisiae* is subject to degradation caused by the action of proteolytic enzymes in the downstream processing operations (Smith, 1997).

The degradation rate of the enzyme often has a decisive effect on the process control strategies. A high degradation rate means that slower downstream processing options will be infeasible, as they will only yield small amounts of the enzyme. Based on data from Smith (1997), the degradation rate was estimated as 0.112h^{-1} and was assumed to be unaffected by growth rate.

To simulate the impact of this degradation after performing mass balances, each unit operation calculates the loss of alcohol dehydrogenase caused by degradation. The degradation process is a first order process. The amount of alcohol dehydrogenase degradation is given by the following equation: -

$$M_{ADH_deg} = M_{ADH} \cdot e^{-k_{deg} \cdot t} \quad \text{Equation 3.6}$$

where M_{ADH} is the mass of alcohol dehydrogenase (g), M_{ADH_deg} is the mass of alcohol dehydrogenase after degradation (g), k_{deg} is degradation rate (h^{-1}) and t is time (h)

3.6.4 Centrifugation and Homogenisation

In this work centrifugation was modelled using the grade-efficiency concept. This approach had previously been used in models by Bulmer (1996), Clarkson *et al* (1996) and Titchener-Hooker and Zhou (1999). Details of the model are given in Appendix A.

The one modification made to the model was to allow for the effects of discharge. Previous work had not allowed for the effect of loss of liquid when the solid is released from the centrifuge. This leads to errors in the mass balance predictions. The model developed in this work calculates both the number of times in a batch that a centrifuge will need to discharge and estimates the volume of liquid that will be discharged along with the solids.

Like centrifugation, homogenisation has also been extensively studied and no changes were made to the approaches reported in the literature that were used to calculate size distribution of the cell debris (Siddiqi *et al.*, 1995; Siddiqi *et al.*, 1996).

3.6.5 Dilution

The process under study also includes two dilution steps; before and after homogenisation. Before homogenisation a buffer is added to the cells collected by the centrifuge in order to obtain a cell concentration of 450g/L. The second dilution step (used after the homogenisation step) reduces the viscosity of the homogenate and aids debris removal.

In previous work by Zhou and Titchener-Hooker (1999) it had been assumed that both dilution steps were fixed. In this work, the dilution step before the second debris centrifuge was allowed to vary so that the effect of different amounts of buffer could be examined. To capture the impact of the dilution step equations 3.6 & 3.7 were generated to describe changes in viscosity and liquid density. The viscosity of the diluted cell stream is assumed to approach asymptotically the viscosity of the buffer. The density was given by linear interpolation reflecting the change in the volume of liquid.

$$\eta_{Dil} = \eta_{Buf} \left(\frac{\eta_{Conc}}{\eta_{Buf}} \right)^{V_{Conc}/V_{Conc}+V_{Buf}}$$

Equation 3.7

$$\rho_{L,Dil} = \frac{\rho_{L,Conc}V_{Conc} + \rho_{L,Buf}V_{Buf}}{V_{Conc} + V_{Buf}}$$

Equation 3.8

where V_{Conc} is the volume of feed in the dilution step (L), V_{Buf} is the volume of additional buffer added (L), η_{Stream} is the viscosity of the specified stream (Ns/m) and $\rho_{L,stream}$ is the density of the specified stream (g/L). The subscripts *Conc*, *Dil* and *Buf* represented the concentrated feed, the dilution output and the buffer solution, respectively.

In addition to the changes in viscosity and liquid density, the addition of buffer also causes the solids concentration to drop. It also has an impact on the centrifugation because of hindered settling. (See Appendix A).

3.7 Results

In order to evaluate the simulation a series of graphs were plotted looking at both the amount of product (alcohol dehydrogenase) and contaminant (debris) produced. Figures 3.2a and 3.2b show respectively the amount of ADH and debris in each stream in the process, when operated under the following conditions: -

- growth rate – $0.18h^{-1}$
- harvest centrifuge flowrate – $400L.h^{-1}$
- homogenisation pressure – 600bar
- number of passes through the homogeniser – 4
- dilution – 1:1
- debris removal centrifuge flowrate – $160L.h^{-1}$

Figure 3.2a shows that the yield of the alcohol dehydrogenase produced is high and very little is lost in the recovery. The total loss of product in the process operated under these conditions stated above is 40% and this is primarily from degradation. However, Figure 3.2b shows that around 10% of the cell debris is left in the

supernatant stream after the debris removal centrifuge. This is because the high pressure results in micronisation of the debris, which is then too small to be removed by the centrifuge.

Such a bar chart could be produced by a simple stoichiometric model. However, the advantage of the simulation developed in this work is that it can be used as base from which to evaluate different operating points and determine the impact of changing control variables. This is demonstrated in figures 3.3a and 3.3b, which look at the impact of reducing the pressure in the homogeniser to 300bar. This is half the pressure assumed in the initial study. As can be seen from Figure 3.3b the resulting process is able to remove the debris. However, figure 3.3a shows that there is much less product in the supernatant stream. This is because fewer cells are getting ruptured and therefore a significant quantity of product remains within unruptured cells which are collected by the centrifuge. Under these operating conditions, less than 30% of the product is recovered in the centrifuge supernatant.

Comparing figures 3.2 and 3.3 it can be seen that increasing the pressure results in better cell rupture but more cell debris. Plotting multiple bar graphs would enable an engineer to find an operating point that meets the criteria. However, this would require analysis of many bar charts and would require the examination of the interactions between each of the control variables. Additionally, once an operating point is found further analysis will be required to determine whether it will be robust given likely variations in the control variables.

A simpler alternative is to use surface response plots. Figure 3.4 shows a surface response plot defined by homogenisation pressure and debris removal centrifuge flowrate. The graph shows the amount of alcohol dehydrogenase produced with different combinations of pressure and centrifugation. Higher pressure results in more product released. However, low centrifuge flowrates result in the loss of product due to proteolysis.

Figure 3.4 shows a series of contours representing increasing levels of alcohol dehydrogenase. For the process to function effectively, cell debris must be removed,

as it is a contaminant that fouls chromatography columns. In this process it has been assumed that the debris concentration needs to be below 0.5g/L. The purple region in Figure 3.4 represents the operating conditions where this constraint is not met. The region occurs when the pressure and the centrifuge flowrate are both high, which produces a significant amount of micronised debris that cannot be picked up by the centrifuge.

The alcohol dehydrogenase process has many similar interactions. For example Figure 3.5 shows the interaction between homogenisation pressure and homogeniser number of passes. However the disadvantage of this approach is that only two dimensions can be visualised at any time.

3.8 Conclusions

This work looks at how bioprocess simulations can be generated and the results that can be obtained. The process used as the basis of this investigation was the alcohol dehydrogenase process. This process has a number characteristics typically associated with labile intracellular enzymes and a number of models have been developed to predict the performance of unit operations used in the process.

The first part of this work looked at different platforms that can be used for developing a bioprocess simulation. The simplest method for developing a simulation would be to use an existing bioprocess simulation package such as SuperPro designer. However, the limitation of this package is that the models are restricted in their predictive capabilities. Equally the package cannot be extended to incorporate new models. As the objective of this work was to look at interactions between control variables this was deemed unsuitable.

The other solution is to use a simulation package or programming language. Previous work had used Labview. However, the graphical programming language, whilst making simple tasks easy, actually made more difficult task very complex and resulted in code that was difficult to adapt and maintain.

The languages looked at were Matlab and C++. Matlab has the advantage that it is relatively easy to program simulations. Additionally Matlab has the advantage that the results of any simulation can be easily visualised using a series of built-in visualisation functions. However, the main drawback of Matlab is that code that requires loops can run slowly. Often however, this limitation can be overcome by writing the code using vectors, which Matlab can handle very efficiently. By contrast C++ can run faster. However, to get the performance from C++ requires a higher degree of expertise. This means that Matlab is probably more suitable for most simulations rather than C++. The only exceptions are for simulations where performance is critical and time and resource for development are less restricted.

When a simulation is developed one critical aspect is the architecture of the solution. The solution should ideally be structured in such a way that it can be easily understood and adapted. This ensures that effort spent developing a model will not be wasted and can be further developed at a later date. In this work a data structure was developed that could hold details of the streams that connect each of the unit operations. The unit operations themselves were modelled using functions in Matlab and Labview and classes in C++. This architecture means that the problem is broken down in a logical manner and is therefore easier to understand.

Ultimately, the utility of the simulation depends on the level of detail within the models. At the simplest level stoichiometric models can be used to generate mass balance models. Such models can be used to track the levels of different components through the system. This can be used to determine steps that lead to a loss of product or are ineffective at removing contaminants so that they can be reviewed or optimised. However, such models are usually only valid for a small range of operating conditions. Critically the sensitivity of the models to different operating conditions cannot be tested. As a result they have no real utility for analysing operating strategies.

The models used in the simulation were able to predict performance of different control settings and take account of the properties of the input streams. The models for centrifugation and homogenisation were based on work by Bulmer *et al* (1996),

Clarkson (1994), Clarkson *et al* (1996), Siddiqi (1996) and Hetherington (1971) and had previously been incorporated into a simulation built by Zhou and Titchener-Hooker (1999). However, this work has been extended by building a new fermentation model using data from Gregory *et al* (1994), Gregory *et al* (1996) and Siddiqi (1996)

The results of the simulation show that it can be used to analyse the trade-offs between control variables. This was demonstrated by using four bar charts to analyse the performance of the process when operated at different conditions. Surface response plots were also used to show that trade off between control variables can be analysed simultaneously. However as mentioned in Chapter 2, one limitation of this approach is that only two control variables can be analysed. The next chapter will look at how the feasible region (defined by a set of constraints) can be analysed to find potential operating points.

Table 3.1: A summary of the components that define the streams in the ADH simulation

Stream Fields	Contents
Volume	Volume of Stream
Component Mass	Protein (g) ADH (units) Nucleic Acid (g) Debris/Cell Wall (g ww) Whole Cells (g dw)
Intracellular Fractions	Intracellular Protein (g) Intracellular ADH (units) Intracellular Nucleic Acid (g) Cell Wall (g ww) Whole Cells (g dw)
Precipitant Fractions	Precipitant Protein (g) Precipitant ADH (units) Precipitant Nucleic Acid (g) Precipitant Debris/Cell Wall (g ww) Precipitant Whole Cells (g dw)
Component Concentrations	Protein (g L^{-1}) ADH (units L^{-1}) Nucleic Acid (g L^{-1}) Debris (g ww L^{-1}) Whole Cells (g dw L^{-1})
Liquid Properties	Viscosity (Ns/m^2) Density (g/L)
Solid Properties	Average Particle Size (μm) Standard Deviation (μm) Density (kg/m^3) Concentration Solids (kg/m^3)
Cell Properties	Hetherington breakage constant (K_p) Pressure Exponent (a) Cell Fractionation (K_d) Protein degradation (k)
Operations Time	Time of completion of operation (h)

Figure 3.1: The alcohol dehydrogenase process.

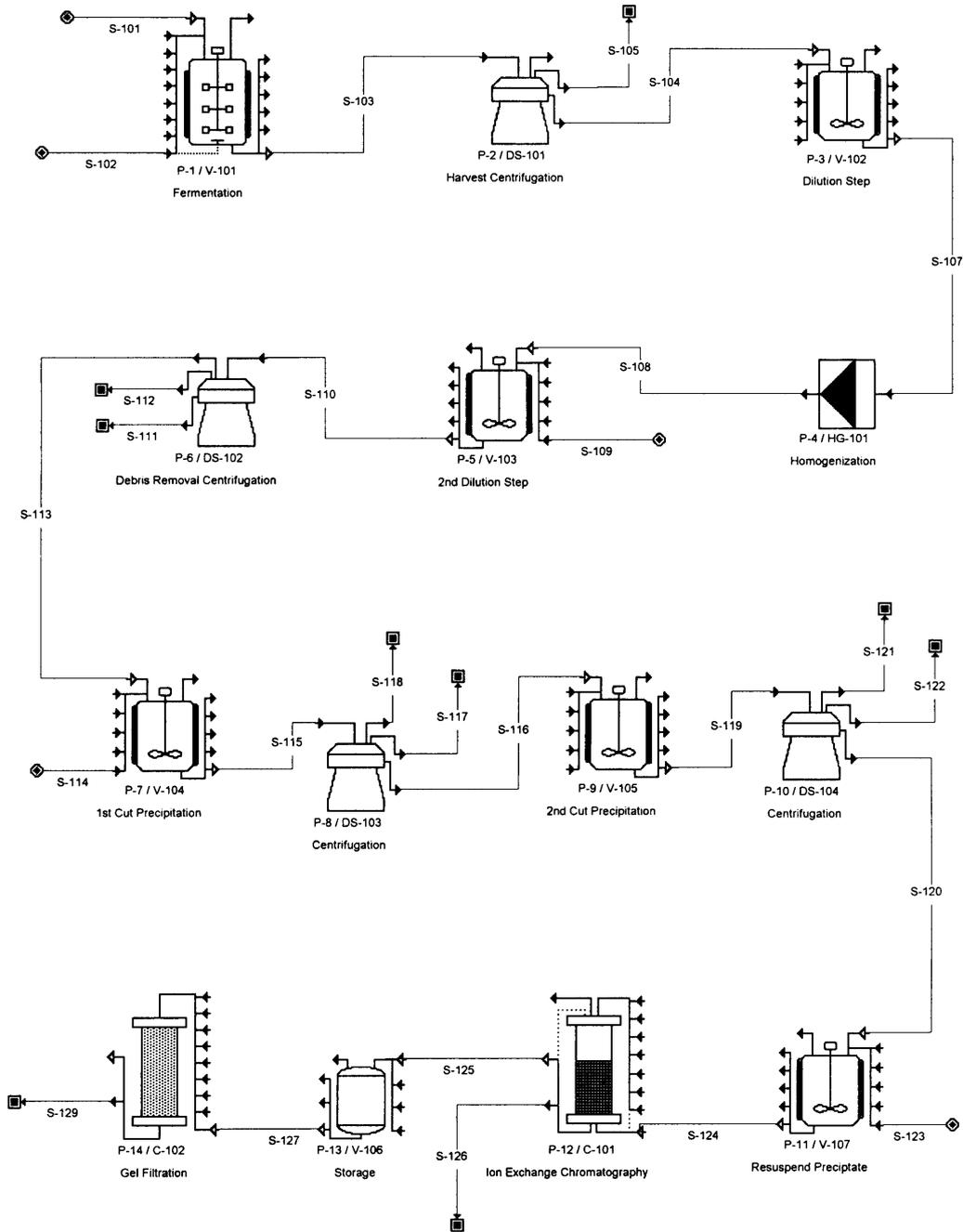


Figure 3.2: Two bar charts showing the levels of alcohol dehydrogenase and cell wall in the different streams in the alcohol dehydrogenase process.

The operating point is defined by: -

- growth rate - 0.18h^{-1}
- harvest centrifuge flowrate - $400\text{L}\cdot\text{h}^{-1}$
- homogenisation pressure - 600bar
- homogeniser passes - 4passes
- dilution ratio - 1:1
- debris removal centrifuge flowrate - $160\text{L}\cdot\text{h}^{-1}$

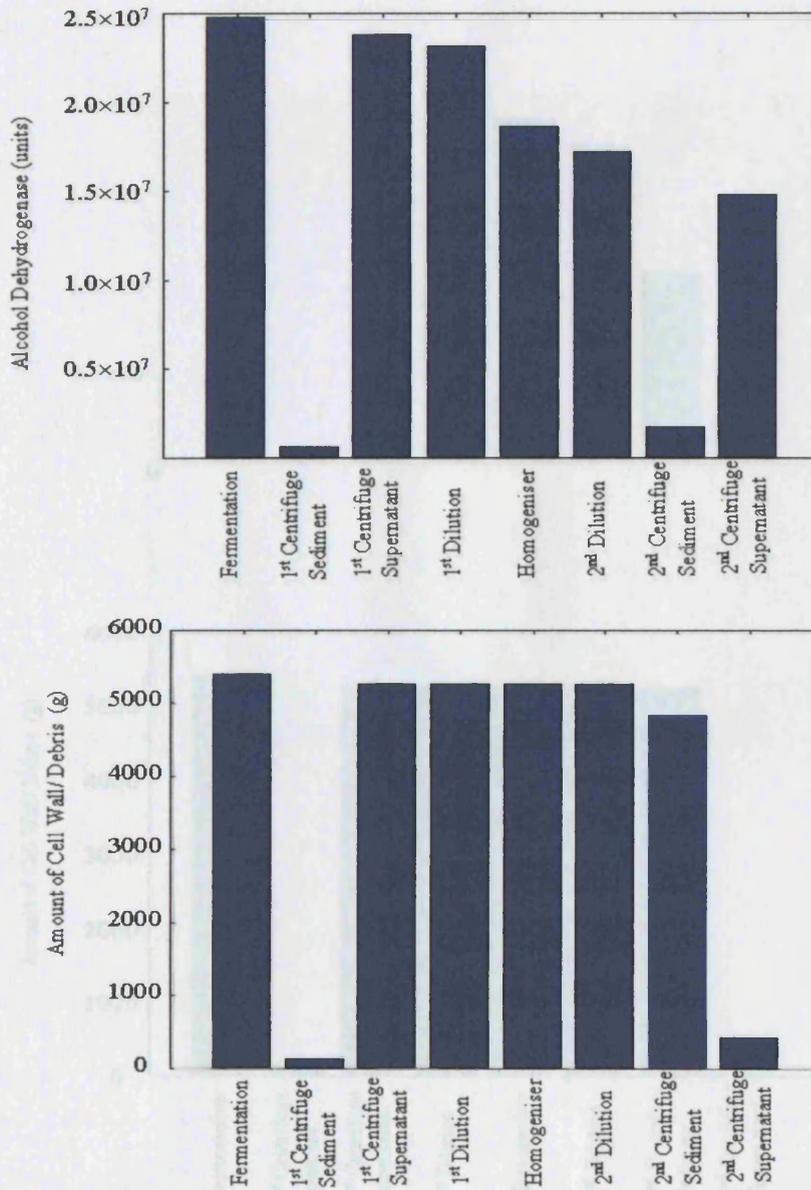


Figure 3.3: Two bar charts showing showing the levels of alcohol dehydrogenase and cell wall in the different stream in the alcohol dehydrogenase process

The operating point is defined by: -

- growth rate - 0.18h^{-1}
- harvest centrifuge flowrate - $400\text{L}\cdot\text{h}^{-1}$
- homogenisation pressure - 300bar
- homogeniser passes - 4passes
- dilution ratio - 1:1
- debris removal centrifuge flowrate - $160\text{L}\cdot\text{h}^{-1}$

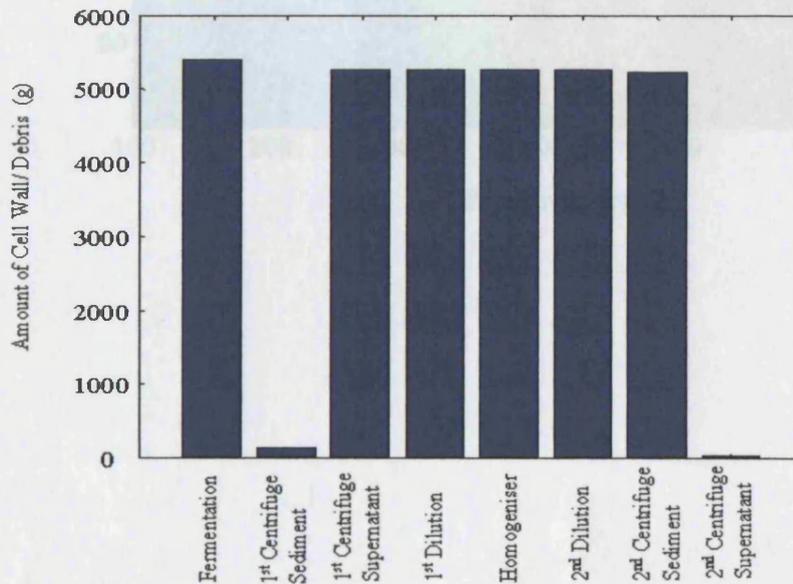
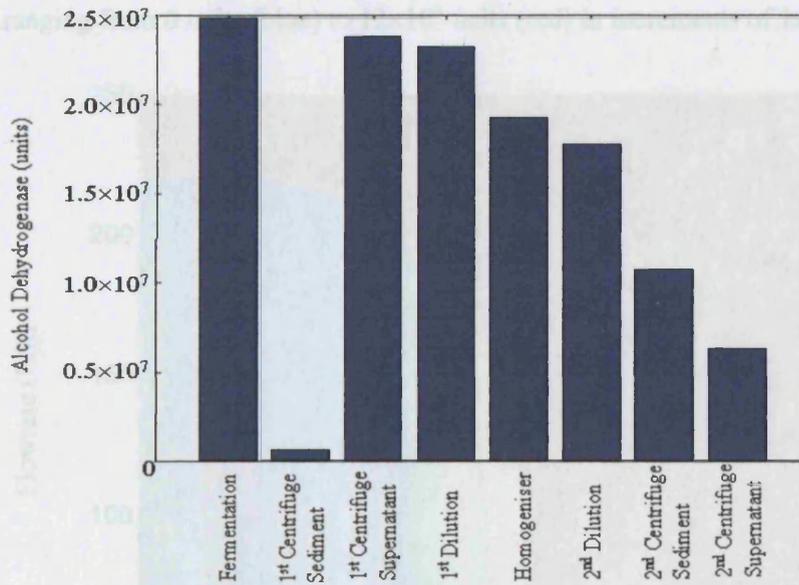


Figure 3.4: The 'Window of Operation' showing the interactions between pressure and debris removal flowrate.

The sub-region was defined by: -

- growth rate - 0.18hr^{-1}
- harvest flowrate - $400\text{L}\cdot\text{h}^{-1}$
- dilution ratio - 1:1
- homogeniser passes - 4

The purple region represents the area where debris concentration is greater than $0.5\text{g}\cdot\text{L}$. The different coloured contour levels represent a series of ADH constraints ranging from 0 units (blue) to 12×10^6 units (red) in increments of 1×10^6 units.

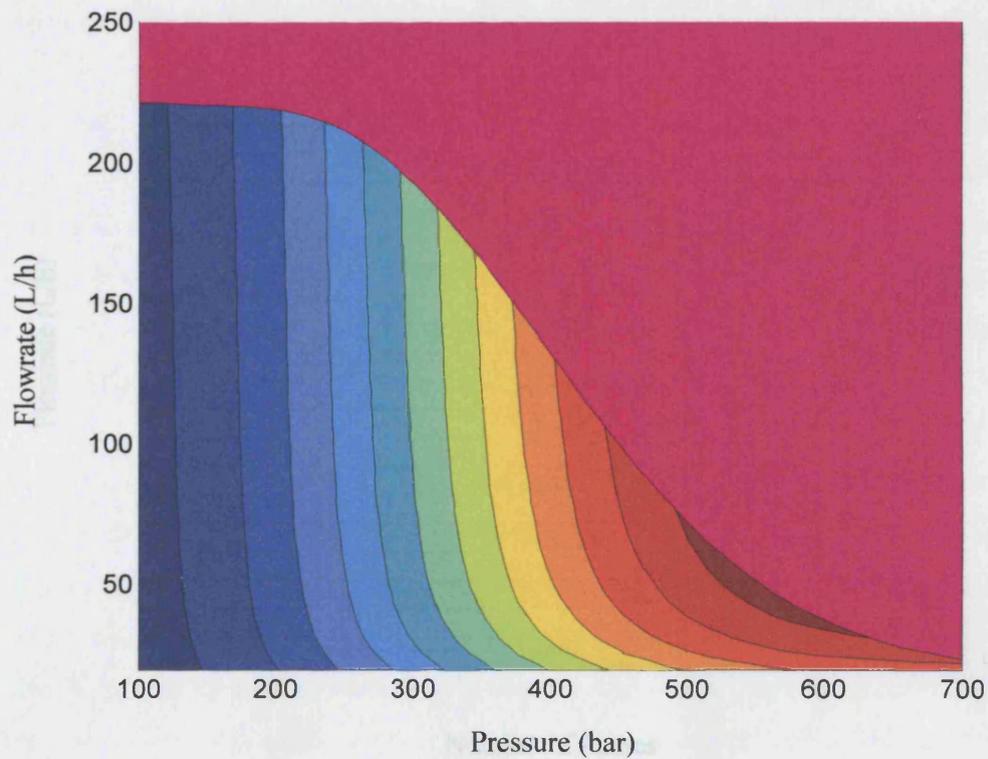
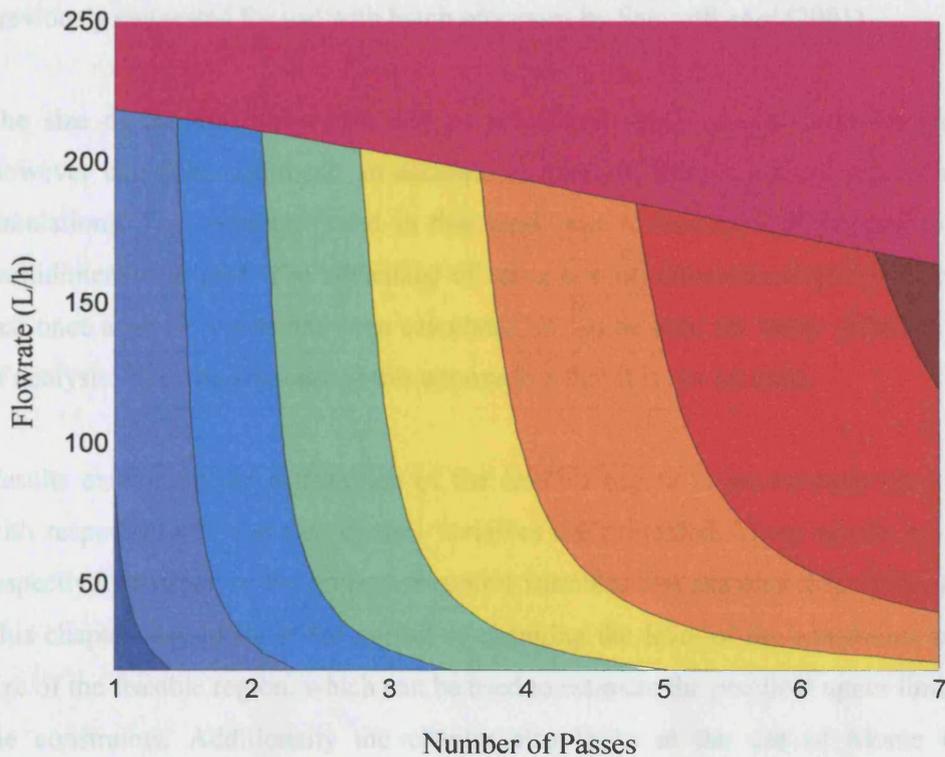


Figure 3.5: The 'Window of Operation' showing the interactions between debris removal flowrate and the number of passes.

The sub-region was defined by: -

- growth rate - 0.18hr^{-1}
- harvest flowrate - $400\text{L}\cdot\text{h}^{-1}$
- dilution ratio - 1
- homogeniser pressure - 300bar

The purple region represents the area where debris concentration is greater than $0.5\text{g}\cdot\text{L}$. The different coloured contour levels represent a series of ADH constraints ranging from 0 units (blue) to 8×10^6 units (red) in increments of 1×10^6 units.



4 Analysing Bioprocess Operating Space

4.1 Abstract

Previous analysis of the feasible region of a bioprocess was restricted to two control variables. The work in this chapter looks at two techniques that can be used to analyse bioprocesses with multiple control variables. The main approach is to look at the size and distribution of the feasible region. The other technique is to locate the maximum ranges for each of the control variables, such that an operating strategy using any combination of points within these ranges will be feasible. This latter technique was previously suggested for use with batch processes by Samsatli *et al* (2001).

The size of the feasible region can be calculated using Monte Carlo integration. However using this approach an accurate estimate of the size would require many simulations. The technique used in this work was to calculate all the points on a multidimensional grid. The advantage of using a multi-dimensional grid of points is that once a set of results has been calculated, it can be used for many different types of analysis. The disadvantage of this approach is that it is not scalable.

Results examining the distribution of the feasible region in an example bioprocess with respect to one and two control variables are presented. These results are used respectively to analyse the impact of control variables and examine their interactions. This chapter also looks at the impact of changing the level of the constraints on the size of the feasible region, which can be used to estimate the practical upper limits for the constraints. Additionally the chapter also looks at the use of Monte Carlo integration and the accuracy of the results when the number of sample points is changed.

Finally the chapter applies a method for finding the largest feasible ranges for each of the control variables. However, the work here demonstrates that the proportion of the feasible region captured within the maximum feasible ranges may be very small. This suggests that this approach may not give an accurate indication of all the potential operating points.

4.2 Introduction

The effectiveness of a bioprocess design can be determined by the ability of the process to meet consistently quality criteria whilst generating the required amounts of product. There are typically many complex interactions between the unit operations in bioprocesses. Investigation into these interactions has led to sophisticated models that are able to calculate the performance of a step based upon the quality of the feed stream and the operating conditions. Such models can be used to develop whole process simulations that capture the interactions between unit operations. However, simulations can only provide value to an engineer or process scientist, when there are effective methods for presenting and analysing simulation outputs.

Previous work has focused on examining the interactions between two control variables by plotting a two-dimensional subset of the feasible region (Woodley and Titchener-Hooker, 1996). This gives the user the ability to evaluate the trade-offs in processes in an intuitive manner. However, this approach can only effectively examine two control variables simultaneously. In reality, the output of most bioprocesses is determined by more than two control variables. Therefore, multiple figures need to be plotted in order to understand the interactions in a bioprocess.

As a result new methods are required for analysing the operating space of processes with multiple control variables. One approach suggested by Samsatli *et al* (1999, 2000) is to locate a set of ranges inside the feasible region such that an operating point in these ranges will be feasible. Another approach is to evaluate the size of the feasible region. This latter approach is the primary focus of this chapter.

The approach developed in this chapter can be used to evaluate the effect of imposing a variety of different constraints on a process. Calculating the feasible region for different constraints gives both an indication of the scope for optimising particular constraints and the corresponding changes in the size of the feasible region. The size in turn gives an indication of the number of viable processing options available and therefore the process robustness.

This chapter looks at how analysing the size and distribution of the feasible region can give an insight into a bioprocess. It also compares the results of such an analysis with results from an approach similar to that of Samsatli *et al.* (1999, 2001). The next section explains the theory behind this approach. Section 4.4 looks at how the size of such a region can be calculated. The results section uses the simulation, described in the previous chapter, to demonstrate this approach.

4.3 Theory

In any bioprocess an operating point can be defined by an n-dimensional vector (z), where n is the number of control variables. This vector can lie at any point in the operating space (Z), an N-dimensional space with a range determined by equipment constraints.

However most bioprocesses are subject to performance constraints such as the maximum allowable amount of a contaminant or the minimum acceptable level of productivity. These constraints on the performance of the process define a feasible region, which is a subset of the operating space. The feasible region, R_{FR} , for a given process design, d , with a set of performance constraints, c , can be expressed as: -

$$R_{FR}(d, c) = \{z | g(d, z, c) \leq 0\} \quad \text{Equation 4.1}$$

Previous work by Woodley and Titchener-Hooker (1996) has shown how a two-dimensional sub-region of the space can be visualised, however, as most processes have more than two control variables other approaches are required in order to make of the results of simulations useful. The approach demonstrated in this chapter is to determine both the size of the feasible region and how the region is distributed in the operating space. The size of the feasible region, I_{FR} , is given by: -

$$I_{FR}(d, c) = \iiint_{R_{FR}(d, c)} \dots \int dz_1 dz_2 dz_3 \dots dz_n \quad \text{Equation 4.2}$$

The value that is generated will have units and a size dependent on the range and units used to define the control variables. Therefore a more meaningful measurement is to determine the fraction of the operating space occupied by the feasible region, I'_{FR} : -

$$I'_{FR}(d, c) = \frac{\iiint \dots \int dz_1 dz_2 dz_3 \dots dz_n}{\prod_{v=1}^n (z_{v, \max} - z_{v, \min})} \quad \text{Equation 4.3}$$

When trying to examine control variables an engineer will want to know how the feasible region is distributed with respect to one or more control variables. This will give an indication of where the region lies and therefore the location of potential operating points. This distribution is equivalent to finding the size of a series of “slices” of the feasible region along one control variable.

This is demonstrated in Figure 4.1, which shows a plot of a two dimensional feasible region and two graphs showing the size distribution of the feasible region along the control variables. In this case the distribution of the feasible region at each point for one control variable can be determined by finding the range of feasible values in the other control variable. This approach can also be used to analyse how the feasible region is distributed across multiple control variables by measuring the size of the feasible region along two control variables. The next section describes the methodologies used to calculate both the size of the multi-dimensional feasible regions as well as a mechanism for determining the largest feasible ranges.

4.4 Computational Methods

4.4.1 Calculating Feasibility

In order to calculate the size of the feasible region a method is needed to calculate the points that are feasible. Using the simulation the performance (y) of the constrained variables can be calculated: -

$$y_i = g_i(d, z) \quad \text{Equation 4.4}$$

These values, generated by the simulation, can then be normalised so that all points below zero represent points where the constraint is not met and points above zero represent points where the constraint is met. It follows that for a point to be feasible

all the normalised constraints must be greater than zero. Therefore the feasibility, Ψ , of a point can be measured by calculating the minimum (or most limiting constraint):-

$$\Psi(d, z) = \min_{i \in I} (g_{norm,i}(d, z)) \quad \text{Equation 4.5}$$

4.4.2 Size of the Feasible Region

Two approaches were examined for calculating the size of the feasible region. The first approach was to break each of the control variables into a series of discrete points, then evaluate the simulation at each point of this multi-dimensional grid. The second approach was using Monte Carlo integration. Here points are sampled at random and the size of the feasible region can be estimated by determining the proportion that is feasible.

This multi-dimensional grid approach was carried out using a series of nested loops to increment each control variable. Each point was assessed to determine whether it was feasible. If the point was feasible the size of the surrounding hyper-rectangle was calculated and added to the size of the feasible region. The algorithm finishes when all of the combinations of control variables have been evaluated, giving the total size of the feasible region.

The main limitation of this technique is that it is not scaleable to higher dimensions. This is because each additional dimension leads to an exponential increase in the number of points that need to be evaluated. Also, doubling the number of intervals along each control variable would raise the computation requirement by a factor of 2^n , where n is the number of dimensions or control variables. This means that the accuracy of the results is likely to be limited, particularly for problems with multiple dimensions.

By contrast, the Monte Carlo integration scales approximately linearly with increasing dimensions. The Monte Carlo integration works by running a series of simulations at random points. These points can be calculated using a random number generator in

conjunction with a series linear function to generate points that lie in the ranges of each control variable.

In this work the objective was to analyse the size distribution with respect to many variables. Monte Carlo integration would require the generation of multiple sets of random points for each control variable in order to determine the size. The advantage of the grid of points is that the same set of results can be used for multiple integration calculations reducing the overall amount of simulation required. Nevertheless, a brief study of Monte Carlo integration was undertaken to compare its accuracy with the results generated by the grid approach.

4.4.3 Determining Operating Ranges

As mentioned in the introduction another approach for analysing the feasible region is to find a set of ranges for each control variable such that an operating point with control variables in these ranges will be feasible. The method proposed by Samsatli *et al* (2001) for achieving this is to find a hyper-rectangle that fits inside the feasible region. The geometry of the hyper-rectangle means that independent ranges can be defined for each control variable so that any operating point with these ranges will result in the performance constraints being met.

In this work, a simple algorithm was developed that calculates the size of this largest hyper-rectangle in the feasible region, so that the results of the integration approach could be compared with it. The algorithm used the multi-dimensional grid of data described previously. It cycles through all the possible combinations of lower and upper vertices. If the upper and lower vertices were feasible it checks the internal points to make sure that they were all feasible. Finally it retains the ranges of the hyper-rectangles with the highest internal volumes. The algorithm also screens solutions to remove instances where any of the operating ranges are below a set of minimum ranges. This stops the algorithm from returning solutions that are inappropriate.

This approach is it is not at all scalable and hence could not be used on a problem with many more control variables. Additionally the algorithm only looks at discrete

points meaning that it is likely to under predict the size of the maximum hyper-rectangle. By contrast the algorithm described by Samsatli *et al* (2001) is more complex and is not constrained to discrete points. However, their algorithm requires that all the vertices in the problem are calculated at each iteration. For a problem with n control variables, this will correspond to 2^n runs of the simulation, meaning that their algorithm is also inappropriate for very large problems.

Nevertheless, the solution proposed here should be able to generate a sensible set of results that could be used for comparison with the results from the volume analysis. The next section looks at the results obtained from both the volume analysis and this work

4.5 Results

4.5.1 Introduction

The approaches described in the previous section were applied to the alcohol dehydrogenase simulation, described in the previous chapter. The alcohol dehydrogenase process has six control variables. These are: -

- Growth Rate in the Fermenter
- Flowrate through the harvest centrifuge
- Number of Passes through the homogeniser
- Pressure in the homogeniser
- Dilution Ratio after homogenisation
- Flowrate through the second centrifuge

The objective of this work was to determine how effectively this process could be analysed using the techniques described in this chapter. The results section is divided into a number of sub-sections each looking at different approaches that can be applied to bioprocesses. The first sub-section shows how the volume analysis can be used to look at the distribution of the feasible region with respect to one control variable. The second sub-section extends this analysis to consider two control variables. Section 3.5.4 looks at how the size of the feasible region can be used to analyse the capability of the process. Section 3.5.5 looks at an analysis that can be carried by determining

operating ranges. Finally section 3.5.6 examines the application of Monte Carlo integration for examining the size of the feasible region.

4.5.2 Distribution of the Feasible Region

Figures 4.2-4.6 show the distribution of the feasible region. In this work debris concentration is treated as a hard constraint that must be met, whereas alcohol dehydrogenase production is treated as a soft constraint that should be as high as possible without breaking the hard constraints. The figures look at the impact of optimising the alcohol dehydrogenase production on the size and distribution of the feasible region as a function of different control variables.

Figure 4.2 shows how the growth rate in the fermenter significantly affects the performance of the alcohol dehydrogenase process. The growth rate in the fermentation step affects a number of cell properties that determine the effectiveness of the downstream process. Such properties include the quantity of alcohol dehydrogenase in the cells and the cell wall strength, which in turn determines how much product is released and the size of the cell debris. Figure 4.2 shows the distribution of the feasible region with respect to the growth rate. The plot shows that there are two peaks. Although the peak at 0.18h^{-1} is higher, the size of the feasible region decreases rapidly, indicating that any feasible point with this growth rate may not be robust. Above a growth rate of 0.22h^{-1} there is no feasible region. This is due to the drop in alcohol dehydrogenase expression at higher growth rates and production of weaker cell walls, which results in more cell debris in the end product. Figure 4.2 also shows the impact of increasing the alcohol dehydrogenase production constraints. The distributions of the feasible regions, produced under the different alcohol dehydrogenase production constraints, follow a similar pattern. This shows that the feasible region is largest when the level of alcohol dehydrogenase production is greatest.

Figure 4.2 implies that lower growth rates could be effectively used in production. However, the fermentation process is time consuming and this ultimately affects the throughput and process economics. Figure 4.3 shows the size of the feasible region defined by productivity. Upon applying a productivity constraint low growth rates

become infeasible as the long operation times reduce the overall productivity. In addition, the size of the feasible regions defined by different productivity constraints change rapidly. At growth rates between 0.1h^{-1} - 0.15h^{-1} , the feasible region for the lowest productivity constraint can be met. However, feasible regions for the other productivity constraints cannot be met. Again this is due to longer processing times and the process would need to produce a large quantity of alcohol dehydrogenase to achieve the higher productivity specifications. [NB, productivity is only an indication of the economic performance of a process: economic performance is also critically dependent on process scheduling, and this is beyond the scope of this work.]

The next step in the process is the harvest centrifuge. Figure 4.4 shows how the feasible region varies with the harvest centrifuge flowrate. This graph shows that the size of the feasible region varies very little over a large range of operating conditions, although it is slightly smaller at very high and low flowrates. At low flowrates the processing time is longer and consequently more alcohol dehydrogenase is lost through degradation. Equally high flowrates result in less efficient cell separation and consequently a loss of some product in the supernatant.

The effect of homogeniser pressure on the size of the feasible region can be seen in Figure 4.5. More product is released at higher pressures meaning that a greater proportion of the feasible region lies here. However, increasing the pressure beyond a certain point results in a drop in the size of the feasible region, as micronised debris associated with higher pressures serve to decrease the size of the feasible region. When the constraint associated with the ADH production is increased the pressure associated with the largest proportion of the feasible region is increased. This is because more of the product is released at higher pressures and therefore the tighter constraints can be met.

The final step in the process is that of the debris removal centrifuge. The debris removal centrifuge primarily determines the amount of debris carried over from the homogeniser to subsequent high resolution steps such as chromatography. The debris can be very damaging and protective filtration may be needed, which is both expensive and time consuming. As can be seen from Figure 4.6, the distribution of the

feasible region is generally skewed to lower flowrates, as higher flowrates are associated with less efficient removal of the cell debris. However, very low flowrates result in long processing times and a loss of product through degradation. The graph shows that the location of the largest feasible region varies with the alcohol dehydrogenase constraints used. For higher alcohol dehydrogenase constraints, higher flowrates are more suitable as they decrease the processing time and reduce the level of product degradation.

4.5.3 Interactions in the Feasible Region

The concept was also applied to analysing how the feasible region is distributed with respect to two control variables. This enables the user to identify key interactions between two control variables and therefore determine desirable operating points. Figure 4.7 shows how the size of the (n-2) feasible region changes with the debris removal centrifuge flowrate and homogeniser pressure. Previous work has shown that there are significant interactions between these control variables. Comparing Figure 4.7 with Figures 4.5 and 4.6 suggests they roughly correspond to each other; in particular the largest feasible volumes for each of the two individual plots corresponds to the largest plot for the two control variables examined simultaneously (500bar and 40L.h⁻¹).

However, this example also shows how there is danger of misreading the data from the one-dimensional plots. According to Figure 4.6, there should be only a very little change when the debris removal flowrate is reduced from 40L.h⁻¹ to 20L.h⁻¹. A relatively large feasible volume would be expected when the debris removal centrifuge flowrate is 20L.h⁻¹ and the homogenisation pressure is 400-500 bar. However Figure 4.7 indicates that there is only a small feasible region in this part of the operating space, thus highlighting the importance of interactions.

4.5.4 Size in Process Analysis

So far it has been assumed that the process must meet some predetermined constraints. However, in certain situations it may be more useful to analyse a process to see what the process is capable of when the constraints are changed. In such

situations the size of the feasible region can be used to determine the ability of the process to meet these constraints. Figure 4.8 shows the size of the feasible region when the alcohol dehydrogenase production constraint is varied and the debris concentration is set at $<0.5\text{g.L}^{-1}$. As can be seen from the graph, when in the minimum level alcohol dehydrogenase is set at 1×10^7 units, approximately 30% of the operating space is feasible. Increasing the alcohol dehydrogenase production constraint causes the size of the feasible region to drop rapidly so that when the constraint is greater than 5×10^6 it is less than 10% of the operating space.

Figure 4.9 extends this concept to examine the impact of changing both minimum alcohol dehydrogenase production and debris concentration constraints. This graph shows a two-way trade-off between the debris concentration and the alcohol dehydrogenase production against the size of the feasible region generated. Decreasing the maximum acceptable debris concentration or increasing the alcohol dehydrogenase constraints results in a smaller feasible region.

The size of the feasible region decreases rapidly when the maximum debris constraint is below 0.5g.L^{-1} . This suggests that there is not much capacity in the process to attain debris concentration of less than 0.5g.L^{-1} , as at this level the process is operating close to its maximum realistic performance. An engineer could therefore deduce that the process would need to be modified if higher levels of performance were required. Equally the rapid change in the size of the feasible region could indicate that the process is very sensitive to small changes in conditions when operated at this specification. Hence if the underlying model were inaccurate then the process may not be operable with this constraint at all.

Figure 4.9 can be used to analyse the trade-off between achieving a low debris concentration and maximising the amount of alcohol dehydrogenase production. An increase in the minimum concentration of alcohol dehydrogenase or decrease in the maximum amount of debris constraints reduces the size of the feasible region. The values of the debris and alcohol dehydrogenase production constraints, where the feasible region approaches zero, represent the Pareto optimum. The points along this curve represent the best combinations of debris concentration and alcohol

dehydrogenase production possible with the process. However selecting process constraints from the Pareto curve would result in a very small feasible region and would make the process difficult to operate robustly. This graph can be used to determine realistic performance objectives that correspond to a feasible region with a reasonable size.

4.5.5 Monte Carlo Integration

All the work shown so far in this chapter is based on volumes calculated using a sequence of points in a multi-dimensional grid. This means that as the number of control variables increases the number of points that require analysis will also increase exponentially. An alternative approach is to use Monte-Carlo integration where random numbers are used to calculate the size of the feasible region.

To investigate the applicability of Monte Carlo integration a series of tests were performed. The objective of these tests was to find out whether Monte Carlo integration could be used as an alternative to calculating all the points in a grid. Figure 4.10 shows the results obtained by using a series of Monte Carlo integrations. This shows that, as the number of points (m) used in the integration increases, the variation in the sizes of the feasible regions predicted decrease. Figure 4.11 shows the root mean error for all these runs. When the number of runs is low (<100) the errors are unacceptably large, however this decreases as the number of simulations increase. The results shown here seem to correspond with the data given by Press *et al* (1999) as the error seems to decrease at approximately $m^{-0.5}$. By the time the number of simulations reaches 10,000 the error is less than 3%.

The work shows that a relatively simple Monte Carlo integration can be used to investigate the size of the feasible region. The results also show that relatively good results can be obtained with a 10,000 simulation run. Such a simulation using the C++ version of the code only takes seconds to run indicating that this approach could be applied.

One other interesting feature is that the size calculated by this analysis is slightly higher than that generated by multidimensional grid. This is possibly because the grid

evaluates points on the boundary, giving it a bias towards points that are less likely to be feasible. This may underestimate the size of the feasible region. However, the main advantage of this approach for future work is that it will scale well for larger problems.

4.5.6 Operating Ranges

As discussed in the introduction, an alternative strategy for analysing the shape of the feasible region was demonstrated by Samsatli *et al* (1999, 2001). In their work they used a hyper-rectangle in the feasible region to find ranges for each control variable. The advantage of this approach is that all points within the ranges are guaranteed to be feasible. To test the effectiveness of this approach a simple algorithm was used to find a suitable set of ranges. In the algorithm, the following minimum ranges were specified for each of the control variables: -

Growth Rate	0.015h ⁻¹
Harvest Flowrate	50L.h ⁻¹
No of Passes	0 passes
Pressure	50bar
Dilution Ratio	0.25L.L ⁻¹
Debris Removal Flowrate	40L.h ⁻¹

The five largest hyper-rectangles obtained for the first set of conditions are shown in Table 4.1.

Table 4.1: The five largest hyper-rectangles that will fit inside a feasible region defined by an alcohol dehydrogenase constraint of 4×10^6 units and debris concentration of 0.5g.L^{-1} .

	Growth Rate		1 st Centrifuge Flowrate		Homogenise Pressure		No of Passes		Dilution Ratio		2 nd Centrifuge Flowrate		Proportion of the Operating Space (%)	Proportion of Feasible Region (%)
1	0.075	0.180	150	500	400	450	5	7	1.75	2.50	60	100	0.0840	0.9935
2	0.090	0.180	100	500	400	450	5	7	1.75	2.50	60	100	0.0823	0.9732
3	0.075	0.180	150	500	400	450	5	6	1.50	2.50	60	100	0.0747	0.8831
4	0.090	0.180	100	500	400	450	5	6	1.50	2.50	60	100	0.0732	0.8651
5	0.075	0.180	150	450	400	450	5	7	1.75	2.50	60	100	0.0720	0.8516

The operating ranges shown in the table above seem to correspond directly with the information given by the volume analysis carried out previously. For example, in Figure 4.4 the feasible region does not vary much in size for different harvest centrifuge flowrates. Consequently the harvest centrifuge has a relatively large operating range, which will ensure that the constraints are met.

Figure 4.2 also shows that the size of the feasible region has large variations for small changes in growth rate. However there is a region in the middle of the range where the size of the feasible region is relatively constant for growth rate. As a result this range of values (between 0.0075h^{-1} and 0.18h^{-1}) is selected as the operating range for the growth rate.

In contrast with growth rate and harvest centrifuge flowrate, the homogeniser pressure has a relatively small range. This can be correlated with the results shown in Figure 4.5, which shows that there is a very small range of homogeniser pressures, which correspond to a large proportion of the feasible region. The small range in the pressure means that this algorithm is maximising the size of hyper-rectangle by keeping within this limited range.

Equally the second centrifuge flowrate also has a very small range. Interestingly the range lies above that region with the maximum volume as predicted by Figure 4.6. This again may be due to interactions in the process. Looking at Figure 4.7, it can be seen that there is an interaction between homogeniser pressure and flowrate. That means low flowrates are less likely to be feasible when the pressure is around 400bar.

The advantage of this approach is that an operating point can be selected that will lie in the middle of the operating ranges. Such a point should be robust to variations in control variables caused by inaccuracies in the process control. However, the results indicate two problems with this approach. The first problem is that the algorithm cannot take account of how difficult a particular control variable is to control. Hence the results shown here have a large range for the harvest centrifuge but the minimum range for both the homogeniser pressure and debris removal centrifuge, which in practice are likely to be much harder to control.

However, the most significant problem is that it does not capture that much of the feasible region. In fact the largest hyper-rectangle, calculated using this procedure, captures is less than 1% of the feasible region. This can in part be explained by the use of the data grid, which restricts the solution to certain set ranges. However the problem is largely due to a limitation in the technique, as hyper-rectangles will only ever capture a limited proportion of any irregular shape. Analysis suggests that this problem is likely to be more acute as the number of dimensions increases (see examining hyper-rectangles in hyper-spheres in Appendix B). Overall this seems to indicate that this approach will not give a very good indication of the full extent of the feasible region of a bioprocess.

4.6 Conclusions

In this chapter the objective was to find a method to analyse the feasible region of bioprocesses with multiple control variables. The main body of the work looked at analysing the size and distribution of the feasible region using different control strategies. The work shows that size of the feasible region can be used to analyse a process and to determine the location of good operating points. In order to determine the impact of control variables the work looked at how feasible regions were

distributed with respect to one or more control variables. The distribution of the feasible region was calculated using a simple integration technique and the results were plotted to show where the feasible region lay. In this work both one and two dimensional distribution graphs were plotted and the latter were able to show interactions between the control variables and highlight regions where the one-dimensional plots can lead to misleading conclusions.

This work has also looked at the impact of adjusting the constraints on the size of the feasible region. This gives an indication of the likely performance of a process. An engineer can look at such a plot and determine that, if the size of the region is small when a constraint is met, then the process may not be robust. Equally, if a small increase in a constraint results in a rapid decrease in the size of the feasible region, then this could also indicate that it will be difficult to operate the process robustly.

One limitation of the work is that the current implementation is not scalable. An alternative approach is to use Monte-Carlo integration. The advantage of Monte-Carlo integration is that it should be scalable to higher dimensions, as it does not need to examine every point in a grid. The work here shows that this approach can give very accurate results and hence has a lot of potential for solving such problems.

Finally results from the analysis of the size of the feasible region were compared to results generated by the trying to find the largest hyper-rectangle in the feasible region. The work in the thesis shows that this approach does not give any real indication of the extent of the feasible region. This can in part be attributed to the limitations of the implementation used here and in particular in the inability of the algorithm used to consider vertices that do not lie at discrete intervals. However, it is mainly due to the highly irregular shape of the feasible region. This seems to indicate a key limitation of the approach. The main advantage of the operating ranges approach is that an operating point can be selected so that it lies a fixed distance from each boundary meaning that it is more likely to be robust. A point selected using the feasible volumes would need to be validated by another approach to confirm that it is robust (and in extreme situations that it is feasible).

Neither the integration nor the operating ranges methodology can guarantee a robust point because they do not account for the impact of variation in the bioprocess performance from that predicated by the model. Such variation could mean that some set points are not robust despite lying comfortably inside the feasible region. The next two chapters look at methods for analysing a process to allow for deviations and uncertainty in the control variables in the process.

Figure 4.1: An example where analysing the size distribution of the feasible region can lead to the selection of a robust operating point.

(See section 4.3)

This figure shows an example feasible region defined by two control variables (z_1 and z_2). The two graphs underneath show the distribution of the size of the feasible region with respect to each control variable.

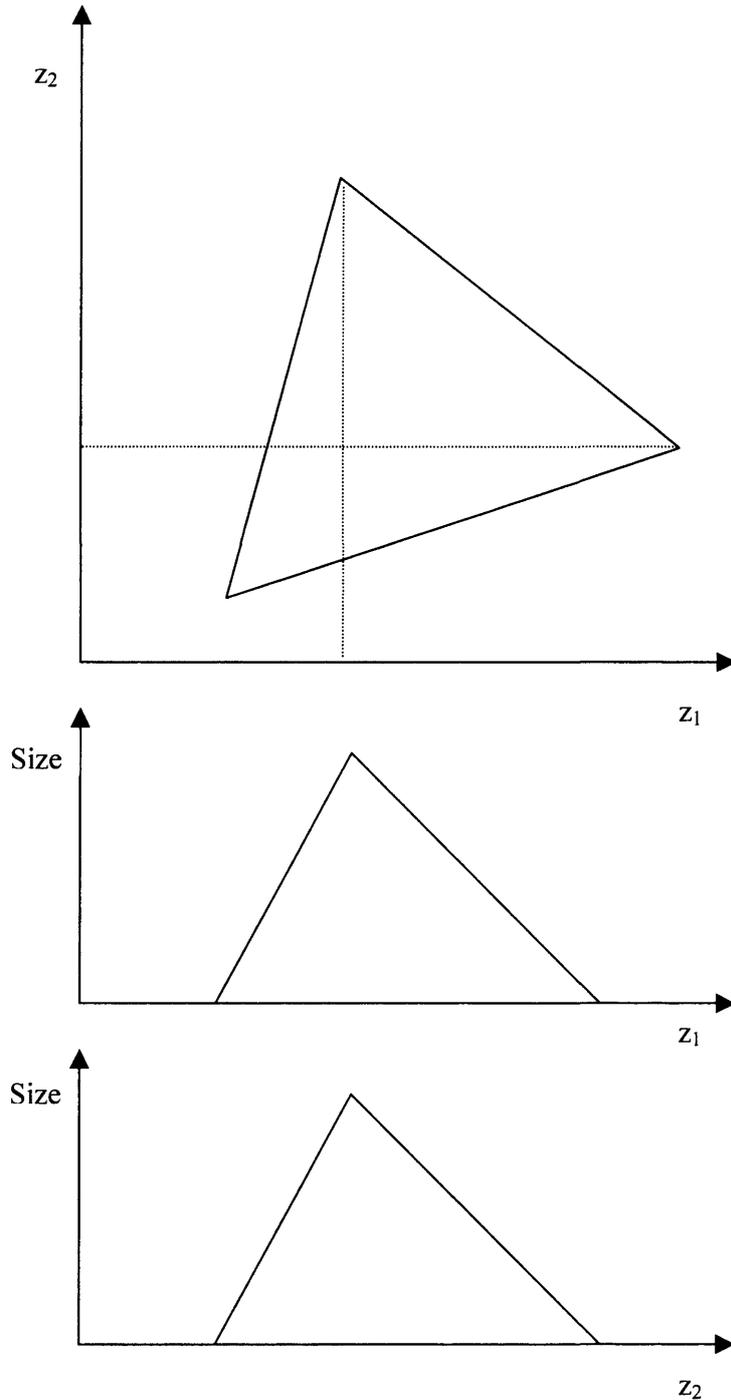
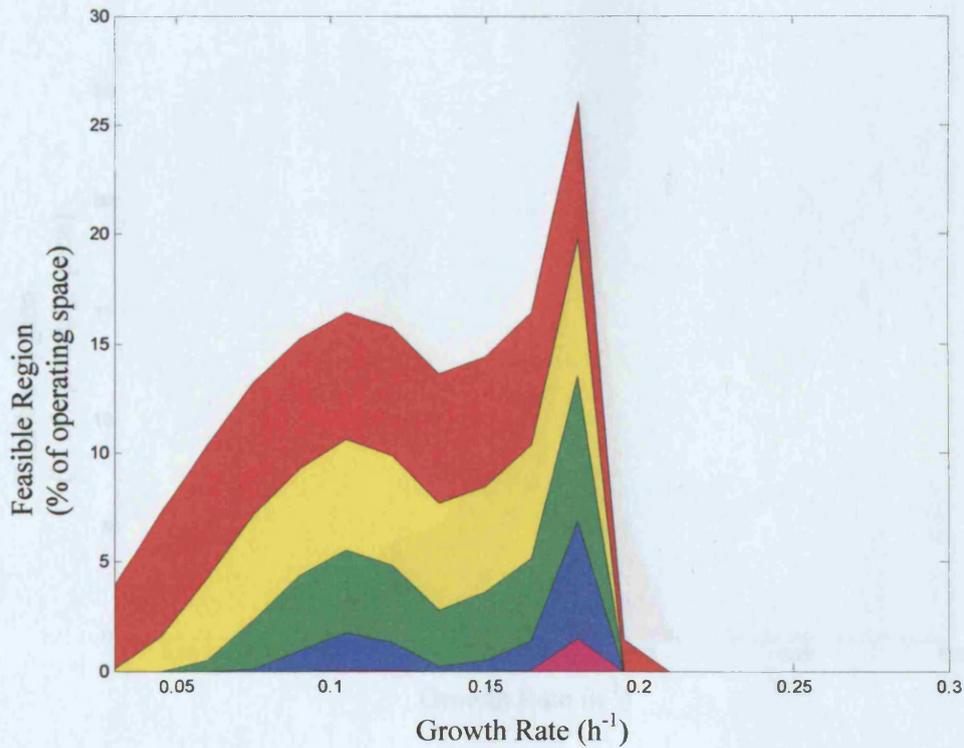


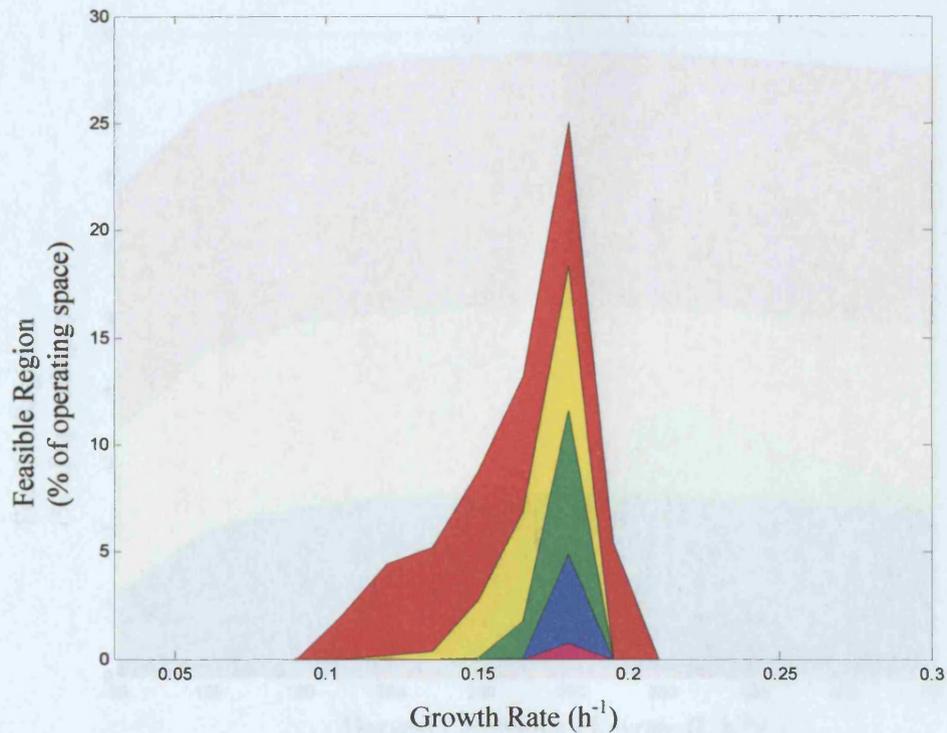
Figure 4.2: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different growth rates.



The minimum levels of ADH shown below are: -

- 4×10^6 units – red
- 5×10^6 units – yellow
- 6×10^6 units – green
- 7×10^6 units – blue,
- 8×10^6 units – magenta

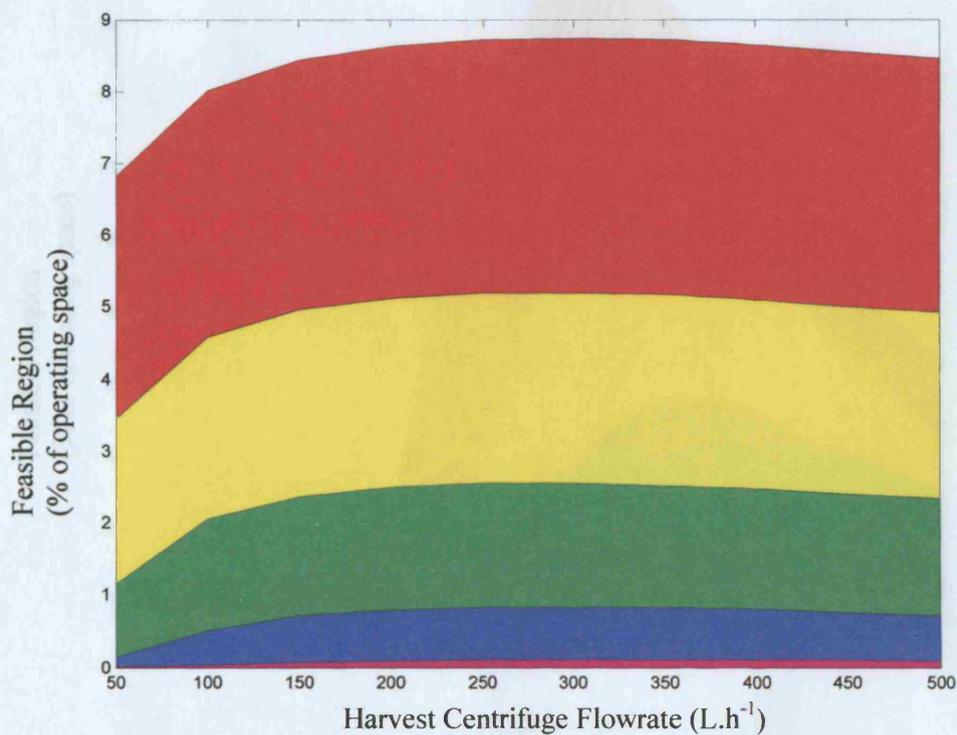
Figure 4.3: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of productivity, with different growth rates.



The minimum levels of ADH used were: -

- 1×10^5 units/h (fermentation time) – red
- 1.25×10^5 units/h (fermentation time)– yellow
- 1.5×10^5 units/h (fermentation time)– green
- 1.75×10^5 units/h (fermentation time)– blue
- 2×10^5 units/h (fermentation time)– magenta

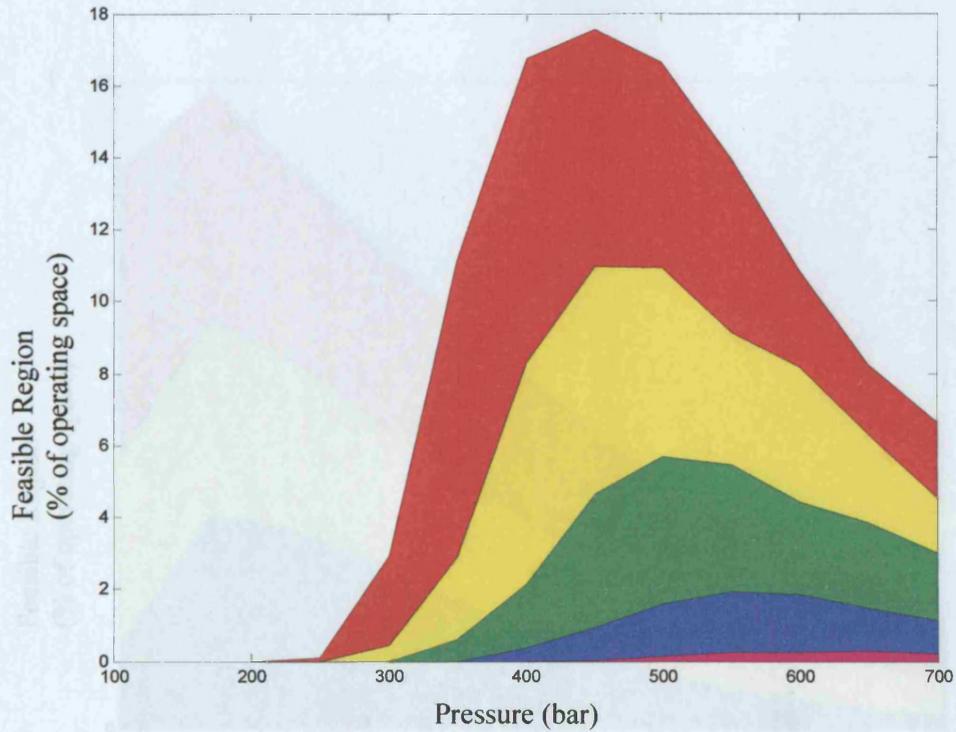
Figure 4.4: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different harvest centrifuge flowrates.



The minimum levels of ADH shown below are: -

- 4×10^6 units – red
- 5×10^6 units – yellow
- 6×10^6 units – green
- 7×10^6 units – blue
- 8×10^6 units – magenta

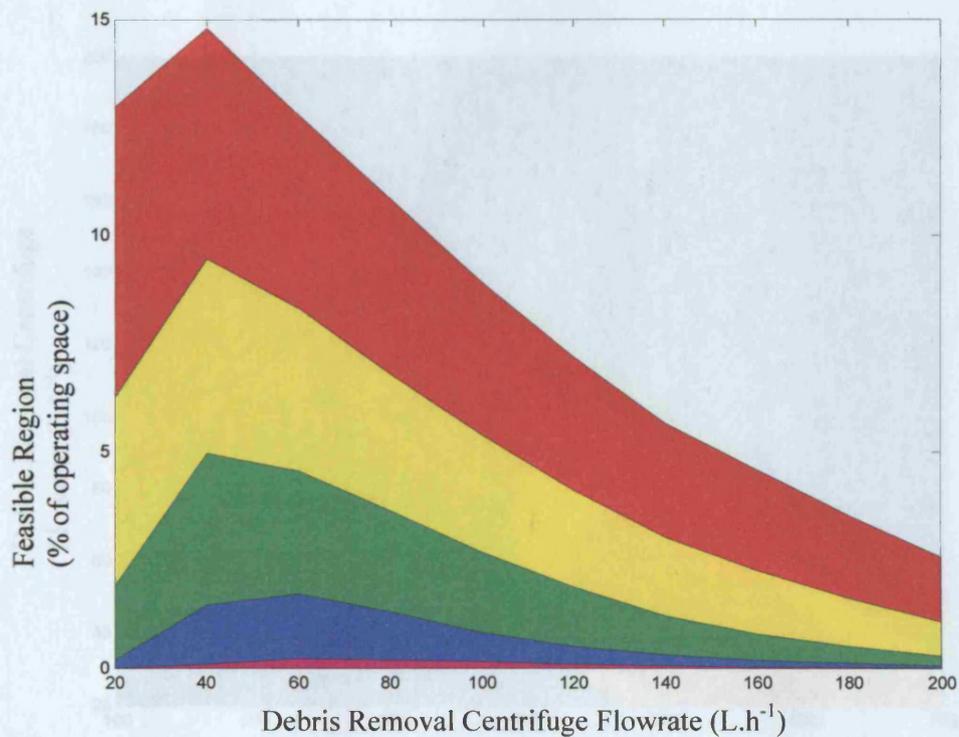
Figure 4.5: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different homogeniser pressures.



The minimum levels of ADH shown below are: -

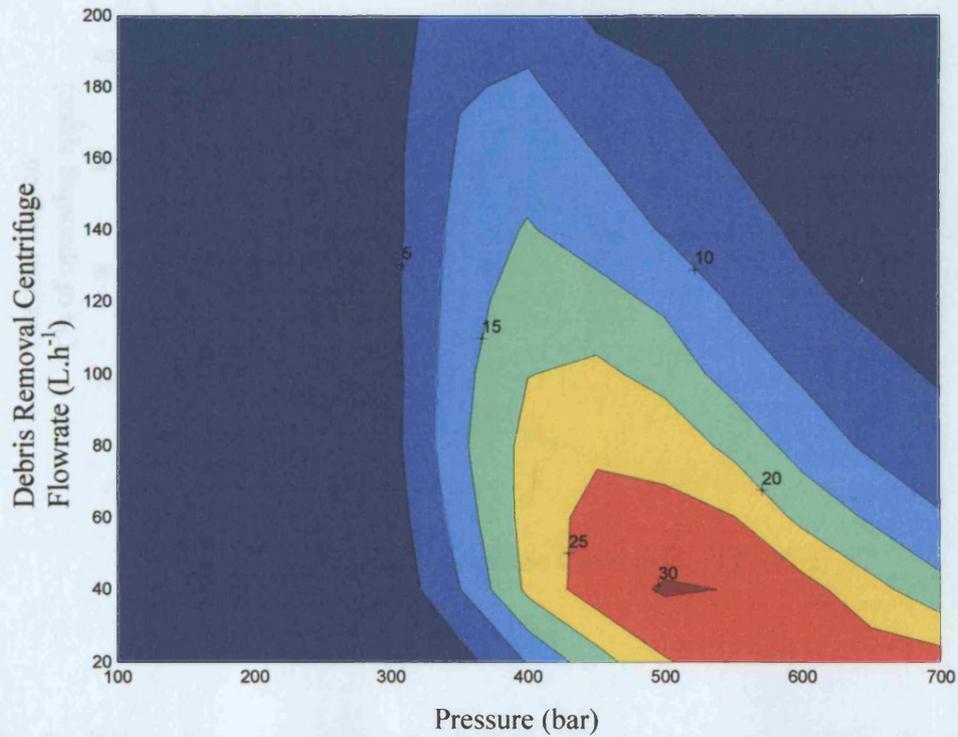
- 4×10^6 units – red
- 5×10^6 units – yellow
- 6×10^6 units – green
- 7×10^6 units – blue
- 8×10^6 units – magenta

Figure 4.6: Distribution of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a series of minimum levels of ADH production, with different debris removal centrifuge flowrates.



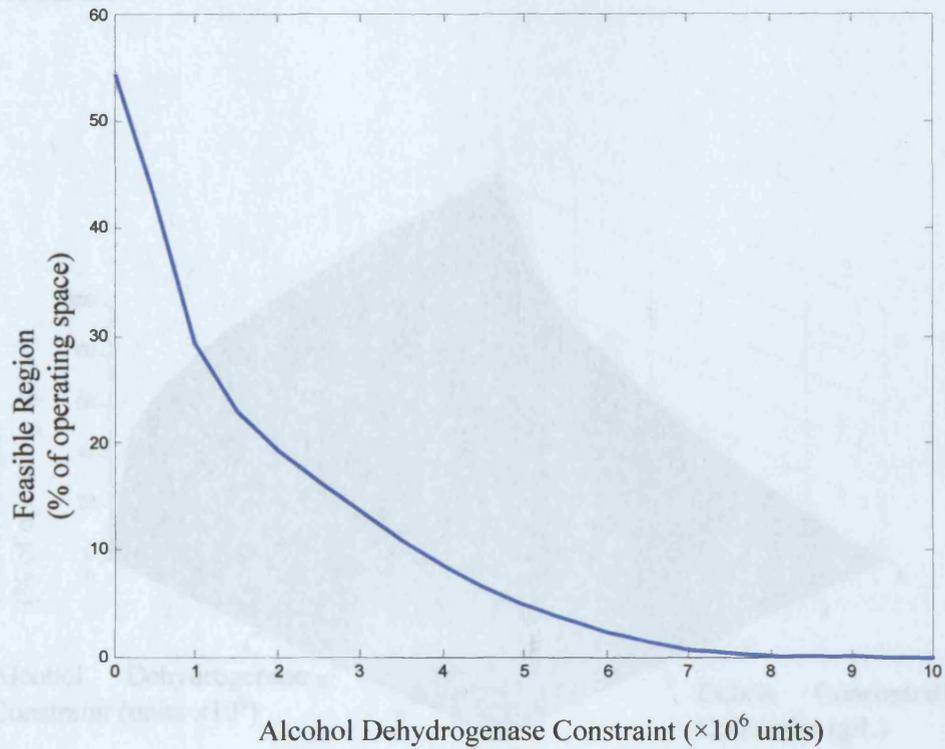
The minimum levels of ADH shown below are: -

- 4×10^6 units – red
- 5×10^6 units – yellow
- 6×10^6 units – green
- 7×10^6 units – blue
- 8×10^6 units – magenta

Figure 4.7: Variation in the size of the feasible regions, defined by a maximum debris concentration of 0.5g/L and a minimum levels of ADH productions of 4×10^6 units, for different homogeniser pressures and flowrates in the debris removal centrifuge.

Proportion of the Operating Space occupied by the Feasible Region

0%-5%	dark blue
5%-10%	light blue
10%-15%	turquoise
15%-20%	green
20%-25%	yellow
25%-30%	red
over 30%	dark red

Figure 4.8: Variation in the size of the six-dimensional feasible regions.

The feasible region encompasses the growth rate, the harvest centrifuge flowrate, number of passes through the homogeniser, homogeniser pressure, homogenate dilution and debris removal centrifuge under different minimum levels of ADH of constraints.

Figure 4.9: Variation in the size of the six-dimensional feasible regions, under different maximum debris removal constraints and minimum levels of ADH of constraints.

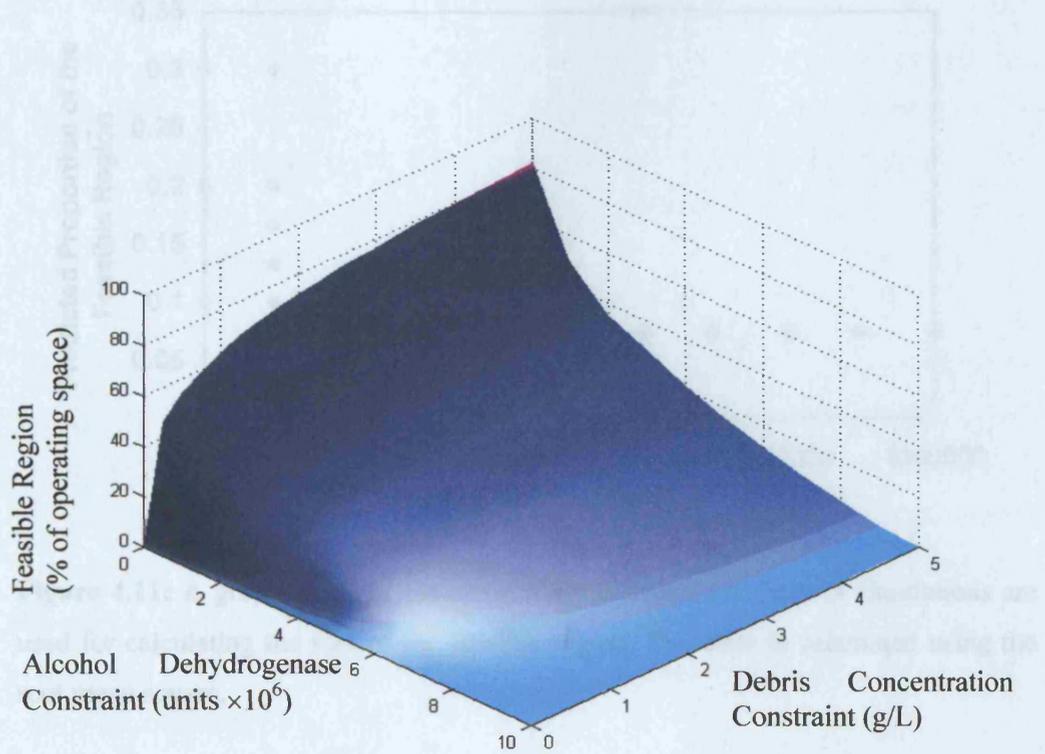


Figure 4.10: A graph showing the results of a series of Monte Carlo integrations showing predicted size against the number of simulations.

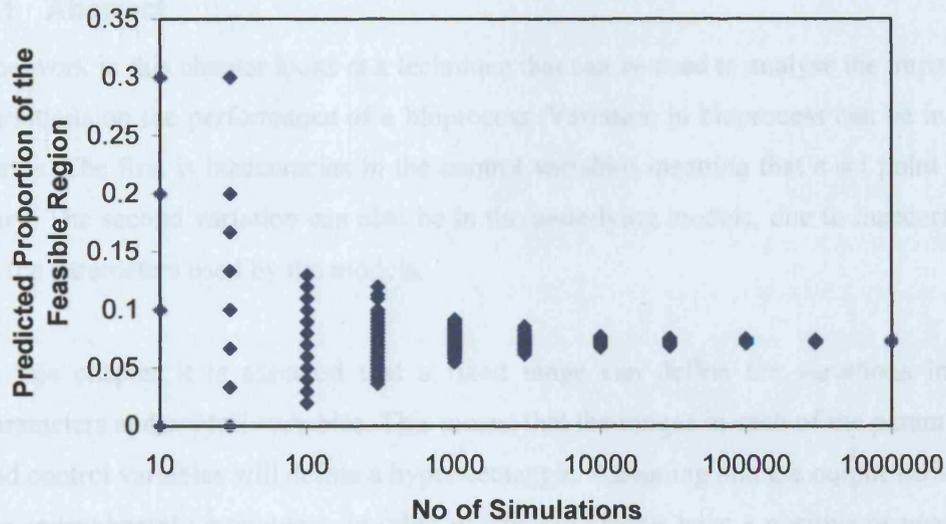
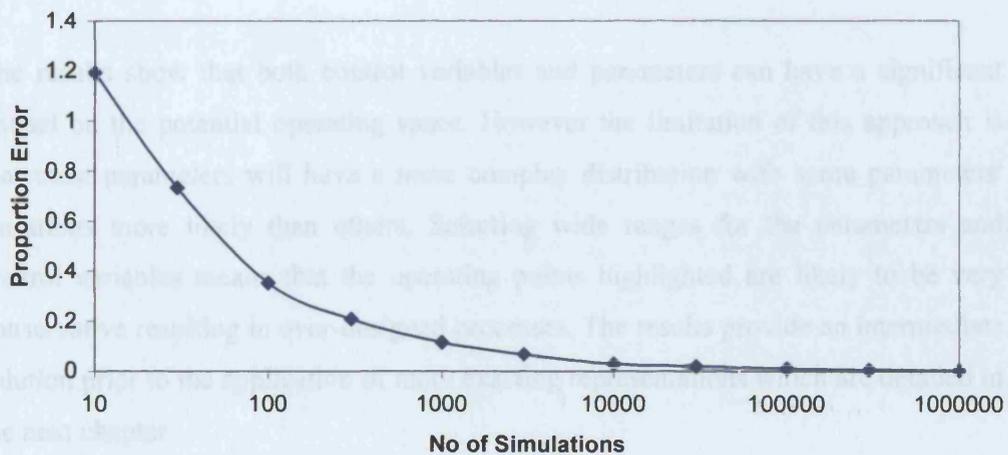


Figure 4.11: A graph showing the error when different numbers of simulations are used for calculating the size of the feasible region. The error is calculated using the root mean square.



5.2 Introduction

In the last chapter, novel analysis were used to visualise the feasible region and determine its size and analyse its distribution with respect to different control variables. This was based on the assumption that an operating point located in the middle of the feasible region would be the most robust. One problem with this

5 Scenario-Based Approaches

5.1 Abstract

The work in this chapter looks at a technique that can be used to analyse the impact of variations on the performance of a bioprocess. Variation in bioprocess can be in two forms. The first is inaccuracies in the control variables meaning that a set point may vary. The second variation can also be in the underlying models, due to inaccuracies in the parameters used by the models.

In this chapter it is assumed that a fixed range can define the variations in the parameters and control variables. This means that the ranges in each of the parameters and control variables will define a hyper-rectangle. Assuming that the output variables are approximately monotonic, in other words will always have a positive or negative gradient, the worst (or best) performance will occur at one of the vertices. Therefore the method used to analyse the variation was to consider a series of scenarios, which correspond to the vertices of a hyper-rectangle defined by variations in the control variables and parameters.

The results show that both control variables and parameters can have a significant impact on the potential operating space. However the limitation of this approach is that most parameters will have a more complex distribution with some parameters' variations more likely than others. Selecting wide ranges for the parameters and control variables means that the operating points highlighted are likely to be very conservative resulting in over-designed processes. The results provide an intermediate solution prior to the application of more exacting representations which are detailed in the next chapter.

5.2 Introduction

In the last chapter, novel methods were used to visualise the feasible region and determine its size and analyse its distribution with respect to different control variables. This was based on the assumption that an operating point located in the middle of the feasible region would be the more robust. One problem with this

approach is that it does not allow for the effects of likely variations in process performance. Such variations could mean that a feasible point is not robust despite lying in the middle of a feasible region. This chapter looks at techniques that consider robustness as part of the evaluation process so that robust points will be highlighted when determining a suitable operating point.

The two major factors that affect process reliability are accuracy of process control and uncertainty in the underlying models. The accuracy in bioprocesses control is limited because such processes are often operated manually and based on readings taken from measuring devices, which are themselves often subject to error. Imprecision in process control can mean that feasible operating points are not always suitable for full-scale operation as they may not be robust given such variations.

Bioprocess simulations are subject to uncertainty because of imprecision in the underlying models. Typically bioprocess models are imprecise because they are subject to simplifying assumptions, as capturing the full complexity of a bioprocess would be extremely difficult. Two sources of potential uncertainty are: -

- **Process Repeatability**

Typically in a bioprocess there will be a number of factors that will affect the performance of the process that may not be considered in any models. An example might be handling of the inoculum and the environmental conditions in the processing facility each of which can impact the performance of the process. In practice, strict adherence to the protocols should minimise the impact of these factors. However, there will always be some variation between batches and this will have an impact on the process performance.

- **Measurement Inaccuracy**

Models are based upon experimental measurements, which will be subject to error. Some assays have particularly low accuracies. For Example the ELISA assay for measuring monoclonal antibodies as an accuracy of $\pm 20\%$. Short development times limit the number of times these experiments can be

repeated. Consequently inaccuracies in the measurements will lead to inaccurate models.

For a point to be considered robust, it must remain feasible given the likely variation in both control variables and parameters. This chapter uses scenario-based techniques to determine whether a point is robust given possible variations in these variables. These results are used in conjunction with methods to visualise the feasible region to determine the robust subset of the feasible region.

5.3 Theory

A number of methods have been applied to analysis of chemical process simulations to obtain robust process designs. One approach that is often referred to as ‘Scenario based’ analysis, assumes that parameters will vary within fixed ranges and therefore a design will only be suitable if it can deal with all possible parameter variations (Grossmann and Sargent, 1978, Halemann and Grossmann, 1983). A variation of this approach (Swaney and Grossmann, 1985), seeks to determine maximum ranges for the uncertain parameters. These ranges were then used to calculate a flexibility index, which gives an indication of how effectively the process can cope with changes in parameters.

However, these approaches implicitly assume that control variables can be adjusted to compensate for changes in parameter values. Such approaches are unsuitable for analysing bioprocesses, since these are subject to strict regulatory requirements and this restricts the level of control variable changes permissible. The focus in this work is finding a set of robust operating points that will be feasible over a range of parameter values.

In a bioprocess, an operating set point (z_s) is defined as being feasible if the performance of the process (y) satisfies the pre-determined specifications c , i.e. $y_i > c_i$, where $i \in I$, where $I = \{1, \dots, i_{\max}\}$ and i_{\max} is the total number of constraints. However such a point may not be robust if there are likely to be variations in either control variables or parameters.

In a bioprocess subject to variation in control variables, the control variables will deviate from their set points independently and simultaneously. If each control variable can deviate by an amount (Δz), then a point can only be considered robust if all the points in a hyper-rectangle defined as the set $\{z_{var} | z_s - \Delta z \leq z_{var} \leq z_s + \Delta z\}$, will result in the constraints being met.

This is demonstrated in Figure 5.1 for a two-dimensional example. The diagram shows two points both of which lie inside the feasible region. Around each point is a rectangle defined by the variation in the two control variables. While both points are feasible only point A can be considered robust. This is because the variation could result in such a point moving outside its operating space. Consequently a robust point will be one where: -

$$g(d, z_{var}, c) \leq 0 \quad \forall z_{var} \in \Omega_O \quad \text{Equation 5.1}$$

$$\text{given that } \Omega_O = \{z_{var} | z_s - \Delta z \leq z_{var} \leq z_s + \Delta z\}$$

where Ω_O is the set of all the possible control variables around the set point, z_{var} is the possible values of the control variables within the region Ω , z_s is the operating set point and Δz is the set of possible variation in control variables

Consequently the robust region, where all the points meet the robustness criteria, can be defined as: -

$$R_{OFR}(d, c, \Delta z) = \{z | g(d, z_{var}, c) \leq 0 \forall z_{var} \in \Omega_O\} \quad \text{Equation 5.2}$$

where R_{OFR} is the robust region given variation in control variables.

This approach can also be extended and applied to bioprocesses subject to variation in the underlying models. The impact of variability in a bioprocess can be significant as

it is hard to ensure both that the model is based on accurate data and that the process will perform in a reproducible manner.

In chemical processes, variations can occur in parameters such as reaction rates and heat exchanger coefficients, and external factors such as cooling water temperature and the quality of feeds. The impact of variation of factors can be evaluated using models, which include parameters for these factors.

By contrast, in bioprocesses the underlying phenomena are less well understood. For example, it is often difficult to determine the exact impact of changing a factor such as temperature or media composition will have on fermentation. Nevertheless a number of empirical models have been developed that estimate how much certain factors will affect process performance. An example of this might be a model to predict cell wall strength as a function of the fermentation growth rate. In such situations the approach could be to assume that the model accurately predicts the relationship but that the parameter will be subject to variability. Hence: -

$$\theta = \theta_N(z)(1 + \Delta_\theta) \quad \text{Equation 5.3}$$

where Δ_θ is the possible variation in the parameter value and $\theta_N(z)$ are the nominal parameter values as a function of the control variables.

A robust point, given variation in parameter values, can be defined using a similar approach to that used for defining robust points given variations in control variables. Here a point will be considered robust if all the variable parameters within a defined hyper-rectangle are feasible: -

$$g(d, z, c, \theta_{\text{var}}) \leq 0 \forall \theta_{\text{var}} \in \Omega_R \quad \text{Equation 5.4}$$

Given that $\Omega_R = \{\theta_{\text{var}} \mid \theta_s - \Delta\theta \leq \theta_{\text{var}} \leq \theta_s + \Delta\theta\}$

where Ω_R is the set of all the possible parameter values, θ_s is the normal parameter value, $\Delta\theta$ is the variation in the parameter values and θ_{var} is the range of potential parameter values

Therefore a robust feasible region, given uncertainty in the parameter values, can be written as: -

$$R_{RFR}(d, c) = \{z \mid g(d, z, c, \theta_{var}) \leq 0 \forall \theta_{var} \in \Omega_R\} \quad \text{Equation 5.5}$$

where R_{RFR} is the robust feasible region given uncertainty in the parameters.

The approach taken in this work was to look at the impact of different types of uncertainty on the size of the robust region relative to the feasible region. This was done by plotting two-dimensional subsets of both the feasible region and the robust region on the same graphs. The next section looks at the methods used to generate such graphs.

5.4 Computational Methods

A series of graphs was generated showing the robust feasible region defined by different levels of uncertainty in parameters and control variables. The approach that was taken was to calculate the feasibility and the robustness at each point on a grid. A contour plot was then used to visualise the regions where selected feasibility and robustness criteria were met.

A point is defined as feasible when all the constraints are met at that point. Each constraint can be written as: -

$$y_i(d, z) > c_i \quad \text{Equation 5.6}$$

Each constraint can be normalised so that any value greater than zero is feasible (e.g. the constraint is met) as described in the previous chapter. A measure of the feasibility (ψ) can be calculated by using the lowest normalised constraint: -

$$\Psi(d, z) = \min_{i \in I} (g_{norm,i}(d, z)) \quad \text{Equation 5.7}$$

The feasibility measure can then be stored in an array and used to visualise a feasible region (e.g. the combination of points that are greater than zero). A similar approach can also be used to define robust regions, given variations in the level of control. For an operating point to be robust, given variation in control variables, all the points in the surrounding hyper-rectangle need to be feasible. Hence the robustness of a point can be measured by selecting the most infeasible of the normalised constraints in the surrounding hyper-rectangle. This can be written as: -

$$\Psi_O(d, z) = \min_{z_{var} \in \Omega} \min_{i \in I} (g_{norm,i}(d, z_{var})) \quad \text{Equation 5.8}$$

where Ψ_O is a measure of the robustness of an operating point (z) given variation in control variables.

Equally for a point (z_s) to be robustly feasible given uncertain parameters, it must be feasible for a range of parameter values. The robustness of each point can be written as: -

$$\Psi_R(d, z) = \min_{\theta_{var}} \min_{i \in I} (g_{norm,i}(d, z, \theta_{var})) \quad \text{Equation 5.9}$$

where Ψ_R is a measure of the robustness given uncertain parameters.

However, there is an infinite number of points in a hyper-rectangle and therefore testing every point is not viable. The approach adopted in this thesis was to look at a limited number of scenarios. In this work, the simulation was run at each of the points furthest away from the set point given the variation in the control variables or parameters. These points provide the vertices of the hyper-rectangle. The number of vertices in a hyper-rectangle is given by 2^{N_v} , where N_v is the number of parameters or control variables subject to variation.

The logic behind evaluating the vertices is that if the most extreme points are feasible then the intermediate points within the hyper-rectangle should also be feasible. This implicitly assumes that the constraints are monotonic (the gradient with respect to each control variable is either consistently positive or negative). However bioprocesses are often non-linear and such assumptions are not necessarily valid. Nevertheless, provided the variability of the control variables is relatively small (i.e. <10% of the parameter or control value), there are unlikely to be set points where the vertices are feasible but large parts of the hyper-rectangle are infeasible. This is because over such small ranges the control variables are generally monotonic.

A problem with the approach is scalability because for every additional uncertain variable the number of vertices that needs to be evaluated doubles. In other words the number of calculation increases exponentially with the number of uncertain variables. This means that if there were ten uncertain parameters over one thousand points would need to be evaluated.

One possible solution to this would be to determine the limiting vertex for each constrained value by looking at the linear change with respect to each parameter or control variable. For example we may determine that increasing control x_1 and decreasing control variable x_2 will both result in a decrease in the constrained variable y_1 . This information could then be used to find the vertex with the highest value of y_1 without calculating all four vertices.

The advantage of this approach is that calculating the derivatives for each control variable will require two runs of the simulation. A further maximum of i_{\max} runs for each of the constraints is then required. This will mean that the number of calculations will be $2n_v + i_{\max}$, (where i_{\max} is the number of constraints.) and hence will scale better. For example if there are five uncertain variables and two constraints then this approach requires twelve simulation runs, whereas evaluating each vertex would require 32 simulation runs.

However in this work the numbers of parameters and control variables investigated were relatively small (the examples in this section have 2-4 uncertain variables).

When there are few uncertain variables, this approach will actually increase the computation effort. For example if there are two control variables and two constraints then this approach would require six points to be evaluated, whereas only four vertices would need evaluating. The next section shows the graphs generated using this approach and give a brief description of the results they show.

5.5 Results

5.5.1 Introduction

The methods described in the previous section were applied to the alcohol dehydrogenase simulation, described in chapter 3. The results shown here look at a two-dimensional subset of the feasible region. Unlike similar analysis by Zhou and Titchener-Hooker (1999), the graphs also show the impact of variation in control variables and parameters. Consequently, they indicate how processes could be operated given such limitations.

The results section is divided into a two sub-sections looking at the impact of variation in the control variables and the impact of variation in the model parameters. These are demonstrated with a series of case studies. The cases studied are all based on the constraints that there must be at least 5×10^6 units ADH and that the debris concentration in the supernatant is less than 0.5g.L^{-1} , unless otherwise stated.

5.5.2 Uncertainty in Control Variables

Figures 5.2-5.7 show the robust feasible regions generated when there is imprecision in the level of control. These figures show a series of regions laid on top of one another. The outermost region is a feasible region and was generated when there was no anticipated imprecision in the control variables. Two robust regions defined by two levels of imprecision in the control variables are then super-imposed.

Figure 5.2 shows a robust region defined by debris removal centrifuge flowrate and homogeniser pressure. The smaller blue region, corresponding to a greater level of imprecision in the control variables, is defined by homogeniser pressure with a

variability of 50bar and flowrate variable of 20Lh^{-1} . The larger yellow region corresponds to lower imprecision in the control variables, where the variability in the homogeniser pressure and debris removal centrifuge flowrate are 25bar and 10L.h^{-1} respectively. This shows that reducing the variability in the control variables increases the proportion of the feasible region that can be used to operate the process robustly.

Smaller operable windows mean that there is less scope for process optimisation, as increasing the constraint specification may result in there being no feasible region. Figure 5.3 shows the same subspace as Figure 5.2, however the feasible and robust regions are defined here by an alcohol dehydrogenase constraint increased by 20% to 6×10^6 units ADH. This reduces the size of the feasible region and the robust regions defined by the more precise level of process control. Critically there is no robust region for operation with the most imprecise process control settings. However, Figure 5.3 only shows a subset of the operating space and there may be robust set points for the less precise controls elsewhere in the operating space.

In both Figures 5.2 and 5.3, the impact of imprecise control could have been determined from examining the feasible region because the controls that are subject to imprecision are those visualised. However, most bioprocess will have multiple control variables, many of which will be subject to variability. Often an engineer will wish to examine the trade-off between two control variables whilst considering variability caused by other control variables in the system. Such a situation is shown in Figures 5.4 and 5.5.

Figure 5.4 shows the trade-off between the number of passes through the homogeniser and pressure in the homogeniser, whereas Figure 5.5 shows the trade-off between the number of passes through the homogeniser and debris removal centrifuge flowrate. Both figures are subject to variability in the debris removal centrifuge flowrate and pressure. In both figures, one of the imprecise control variables is not being visualised. In Figure 5.4 the centrifuge flowrate is not visualised on one of the axes. In this graph, both of the operable regions are skewed to lower pressures. This is in marked contrast to the earlier examples where the operationally robust regions lay concentrically inside the feasible region. This skew is caused by the imprecise control

in debris removal flowrate and means that the debris removal centrifuge flowrate may be higher and hence unable remove small debris produced at higher homogeniser pressures.

In Figure 5.5 the pressure is not visualised on one of the axes. Here, it is clear that the impact of variability is very pronounced as both the robust regions are significantly smaller than the feasible region. At low numbers of passes the variability in the pressure could result in a lower pressure and a significantly reduced amount of product released. However, at high number of passes and flowrates, a higher than expected pressure due to the imprecise control would result in greater micronisation and therefore additional debris carry-over.

So far all the figures have examined the impact of imprecision in the second centrifuge flowrate and homogeniser pressure. However, other control variables in the alcohol dehydrogenase process will also be subject to imprecise control. Logically if there is imprecise control in the second centrifuge flowrate, there will also be variability in that of the first centrifuge. However, the impact is unlikely to be significant because the process is insensitive to changes in this operation, as proven in the previous chapter (Figure 4.4). By contrast the fermentation growth rate will have a significant impact on the process, as it determines the properties of the cells. In this study, the growth rate is controlled by an exponential feed. Control may be inaccurate due to poor calibration of the pump or actuator making the actual growth rate higher or lower than expected. The dilution step, which is used to reduce the debris concentration and solution viscosity prior to the debris removal centrifuge, may be imprecise due to poor measurement of the buffer solution. This could affect the centrifuge performance as well as the debris concentration in the output stream.

The feasible region shown in Figure 5.6 is defined by the same set of operating conditions and constraints as for Figure 5.2. However the robust region shown in Figure 5.6 considers variability in the dilution rate and the fermentation growth rate in addition to the debris removal centrifuge flowrate and the homogeniser pressure. The region in this figure is very similar to the region shown in Figure 5.2 because neither dilution nor growth rate impact on the size and shape of this part of the feasible

region. The robust region is primarily defined by the variability in the pressure and flowrate.

Figure 5.7 shows a feasible region and robust region generated when the growth rate is increased to 0.18h^{-1} . The feasible region is considerably larger than that in figure 5.6, due to the higher amount of alcohol dehydrogenase expressed in the cell. However, the robust region generated is smaller because at this critical growth rate a small variation in growth rate would mean that the cells could enter oxido-reductive growth, where productivity is lower. [The impact of oxido-reductive growth was explored in Chapter 4, Figure 4.2, which shows that the size of the feasible region drops considerably when the growth rate is above 0.19h^{-1} .]

5.5.3 Uncertainty in the Parameters

The case studies so far have looked at the impact of variation on the control variables. Here the focus was on finding a set of robust points given variation from a set point in the control variables. This methodology can be applied to look at variations in the parameters. Equation 5.9 shows a method for calculating the robustness for an operating point given parameters that are subject to uncertainty. Figures 5.8 to 5.10 look at robust regions defined by uncertainty in the parameters. These figures examine the impact of two parameters (debris density and cell breakage coefficient), which are critical for defining the level of debris removal achieved.

Figure 5.8 shows a robust region defined by variations in debris density. Determining an accurate value for the wet debris density is difficult as it is experimentally hard to remove the excess water without changing the debris properties. In this work the robust region is defined by a 2% variation in debris density. Debris density has a critical impact on the efficiency of the centrifuge, as it is a key driving force in the separation of the solid and liquid phases. Figure 5.8 shows that this small amount of variation in debris density results in higher pressures and flowrates no longer being feasible.

Figure 5.9 shows the robust window defined by a 10% variation in the cell breakage coefficient. The cell breakage coefficient is a parameter that can only be calculated

using results from homogenisation experiments. It is determined from the change in particle size in multiple homogenisation experiments run with different pressures and numbers of passes. Consequently there are likely to be errors when calculating this value as it is dependent on both accurately measuring the particle size of the debris and interpolating these values.

As can be seen in Figure 5.9, variation in the cell breakage coefficient can have a significant impact on the process performance. This is because a lower cell breakage coefficient results in more small debris for a given pressure and number of passes in the homogeniser. This smaller debris is more difficult to collect at high centrifuge throughputs resulting in the debris constraint not being met at higher pressures and centrifuge flowrates.

In reality both debris density and the debris breakage coefficient will vary simultaneously, and a more realistic scenario is shown in Figure 5.10, which shows a robust region defined by both a 10% variation in the cell breakage coefficient and a 2% variation in the debris density. The combined effect of the two variable parameters is, as expected, to produce a much smaller robust region. The analysis indicates the significant impact that variations in parameters and control variables can have on the performance of a process and emphasises the need for techniques to determine robust operating strategies for process subject to these kinds of variations.

5.6 Conclusions

The objective in this work was to develop techniques to evaluate the robustness of a particular operating point. The idea here was to determine a sub-region within the feasible region where theoretically the process can be operated robustly. In this work the robustness of a process flowsheet was calculated when both parameters and control variables are subject to uncertainty.

In this work a scenario-based approach was used where the robustness was calculated by calculating the feasibility at each of the vertices of a hyper-rectangle. The advantage of this approach is that it is relatively simple to implement. However with each additional uncertain control variable or parameter, the number of points that

needs to be evaluated doubles. Hence this approach is not scalable when there are many uncertain control variables and parameters. This problem can be overcome by determining which vertex will be lowest by examining the change in each constraint with each parameter and control variable. This approach assumes that output variables will vary monotonically with control variables and parameters. Nevertheless this approach might be justified when variations in parameter and control variables are small and hence the output variations are likely to be linear.

In this work, the robust regions were then plotted on top of the feasible regions to show the impact uncertain variables can have on available operating space. The first set of results looked at the robust regions defined by variation in control variables. The second set of results looked at the impact of variation in model parameters. A limitation with this approach is that a robust region may be missed if it was not in the subspace that is visualised. In theory this approach could be applied in conjunction with the volume analysis technique shown in the previous chapter. However this was not done because significant computation time would be required to calculate such a large number of robust points.

Arguably the biggest limitation of this work is that it only partially solves the problem of determining an operating point. This is because it still relies on an engineer selecting appropriate ranges for each parameter. The parameters are often distributed values, with a large range of potential values but a much narrower range of values that frequently occur in practice. In such situations selecting a wide parameter range that covers all possible parameter values may result in an over-designed process whereas selecting a narrow range may result in a process that is not robust. In most of the examples shown here large ranges were selected to show points that should be robust. Ideally a method looking at robustness should give a user an indication of the probability that the constraint will be met. The next chapter looks at how probabilistic techniques can be applied to bioprocesses to determine the probability a constraint being broken given variations in multiple parameters and control variables.

Figure 5.1: A schematic showing a two-dimensional feasible region (R_{FR}) and the subset of the feasible region that is robust (R_{OFR}) given a pre-determined level of uncertainty in the process control. The schematic also shows two points that are feasible, however one is not robust.

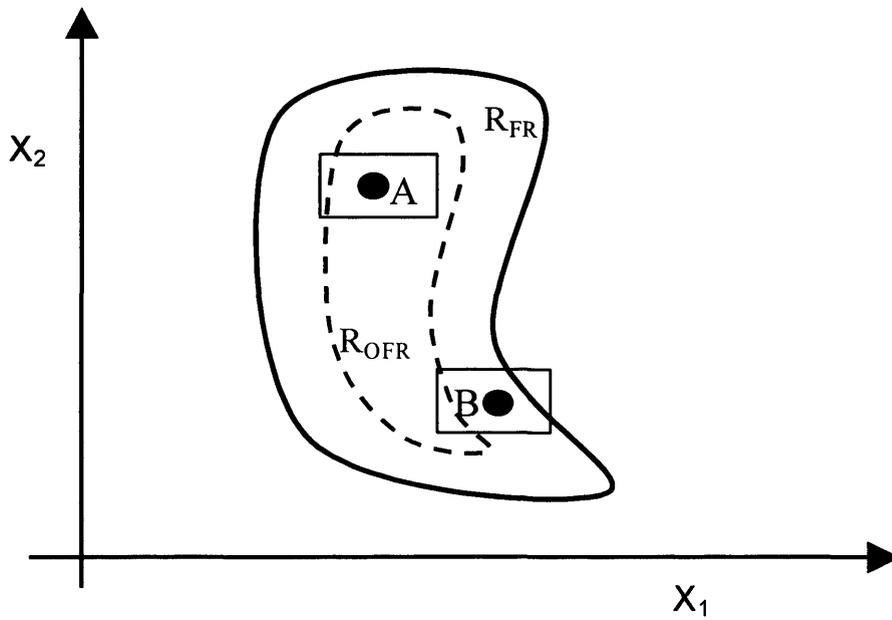
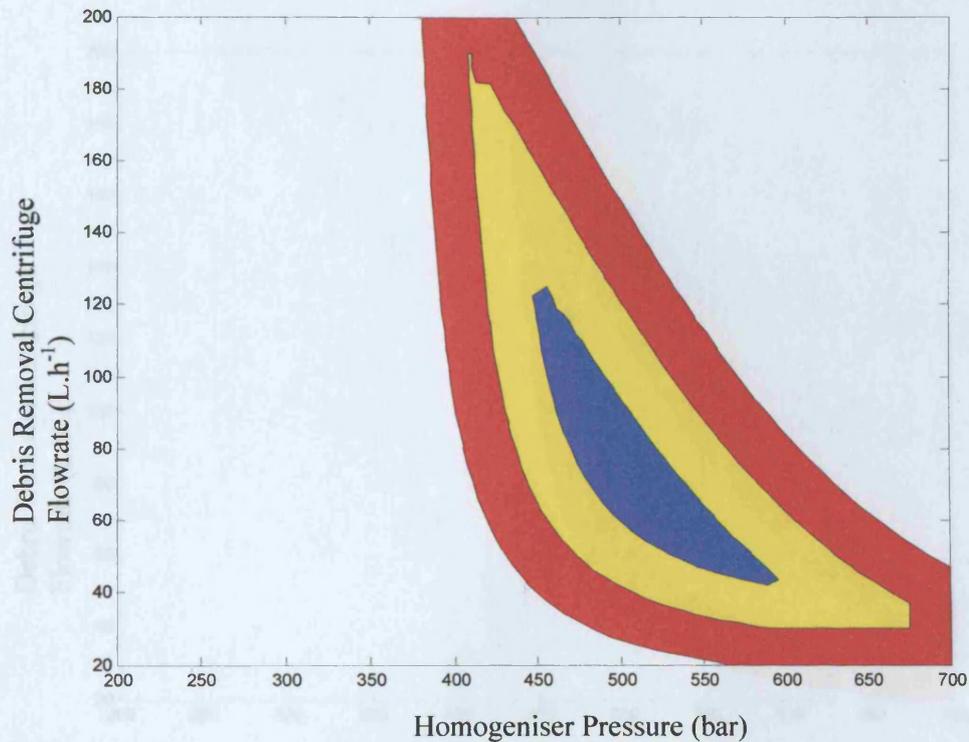


Figure 5.2: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate and homogeniser pressure



The sub-region is defined by: -

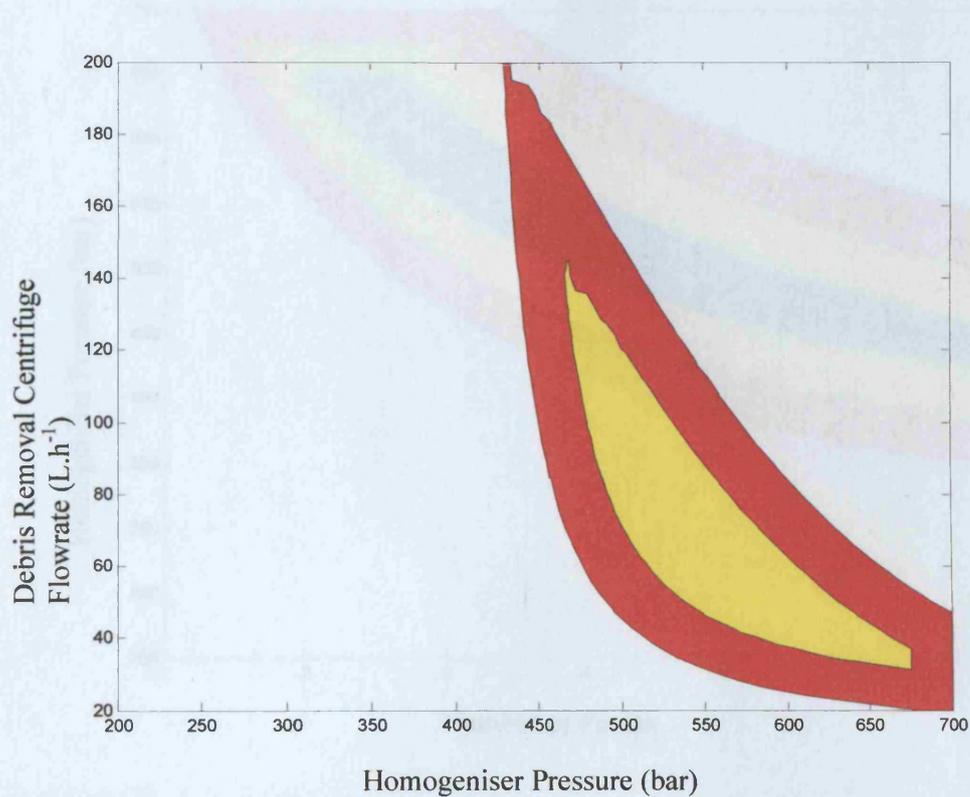
- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} .

The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the centrifuge flowrate of $\pm 10\text{L.h}^{-1}$ and homogeniser pressure of $\pm 25\text{bar}$.
- Blue region - robust region given imprecision in the centrifuge flowrate of $\pm 20\text{L.h}^{-1}$ and homogeniser pressure of $\pm 50\text{bar}$.

Figure 5.3: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate and homogeniser pressure and given a higher ADH production constraint.



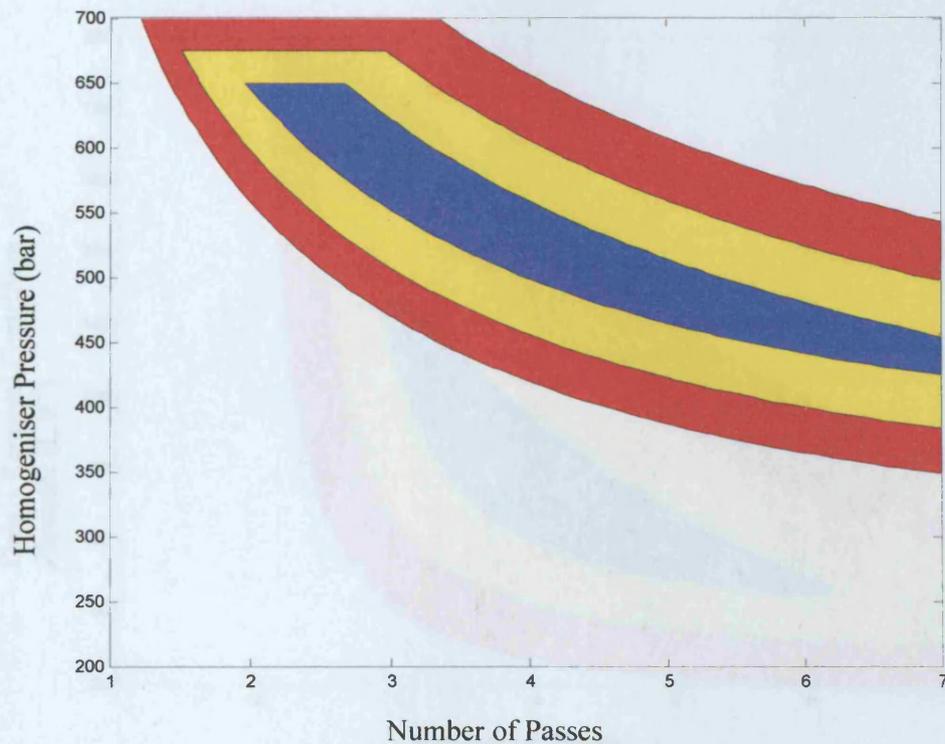
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 6×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the centrifuge flowrate of $\pm 10\text{L.h}^{-1}$ and homogeniser pressure of $\pm 25\text{bar}$.
- [N.B. The robust region given imprecision in the centrifuge flowrate of $\pm 20\text{L.h}^{-1}$ and homogeniser pressure of $\pm 50\text{bar}$ does not exist.]

Figure 5.4: The robust regions defined by homogeniser pressure and homogeniser passes given imprecision in the centrifuge flowrate and homogeniser pressure



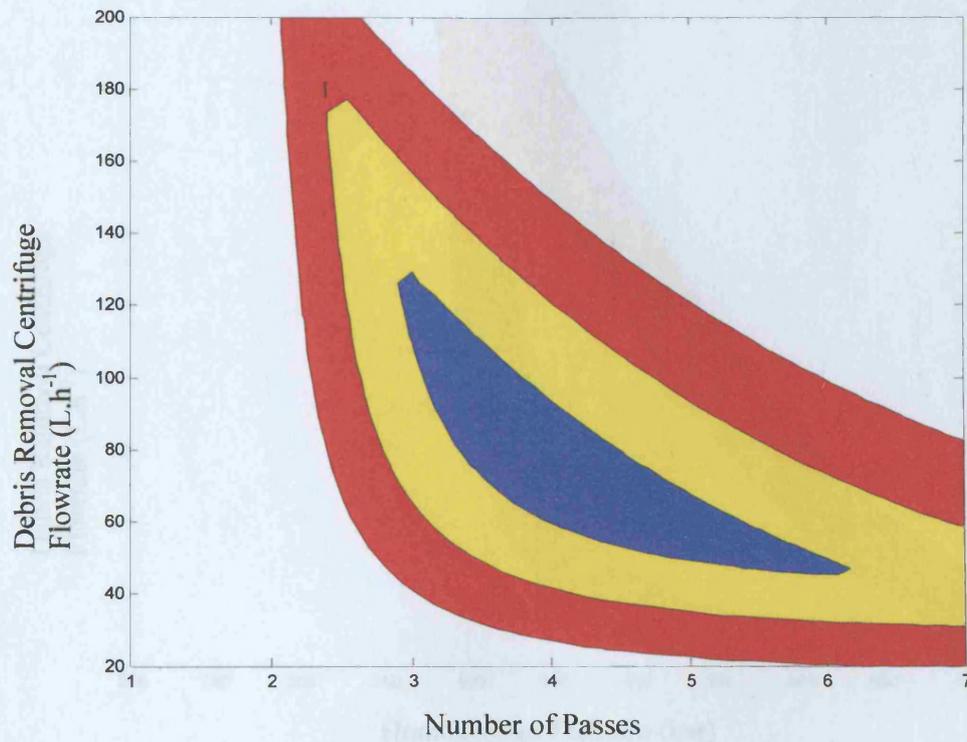
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- debris removal centrifuge flowrate - 60Lh^{-1}

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the centrifuge flowrate of $\pm 10\text{L.h}^{-1}$ and homogeniser pressure of $\pm 25\text{bar}$.
- Blue region - robust region given imprecision in the centrifuge flowrate of $\pm 20\text{L.h}^{-1}$ and homogeniser pressure of $\pm 50\text{bar}$.

Figure 5.5: The robust regions defined by debris removal centrifuge flowrate and number of passes given imprecision in the centrifuge flowrate and homogeniser pressure



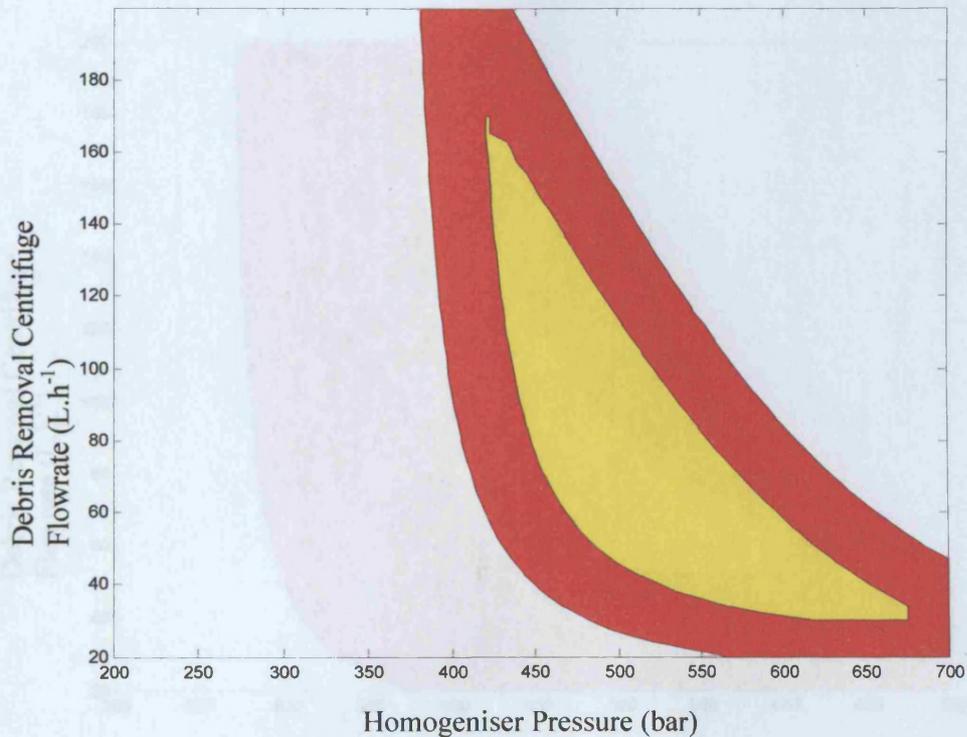
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser pressure - 500bar

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the centrifuge flowrate of $\pm 10\text{L.h}^{-1}$ and homogeniser pressure of $\pm 25\text{bar}$.
- Blue region - robust region given imprecision in the centrifuge flowrate of $\pm 20\text{L.h}^{-1}$ and homogeniser pressure of $\pm 50\text{bar}$.

Figure 5.6: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate, homogeniser pressure, dilution ratio and growth rate.



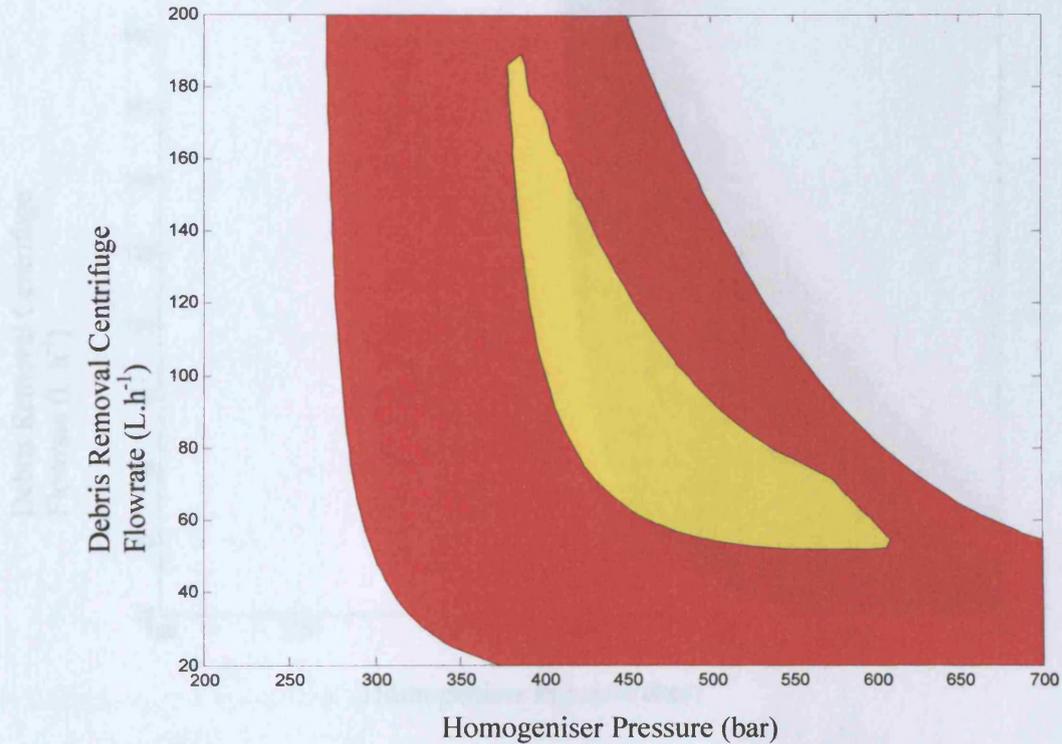
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the debris removal centrifuge flowrate of $\pm 10\text{L.h}^{-1}$, homogeniser pressure of $\pm 25\text{bar}$, dilution 1L^{-1} and growth rate 0.1h^{-1}

Figure 5.7: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the centrifuge flowrate, homogeniser pressure, dilution ratio and growth rate.



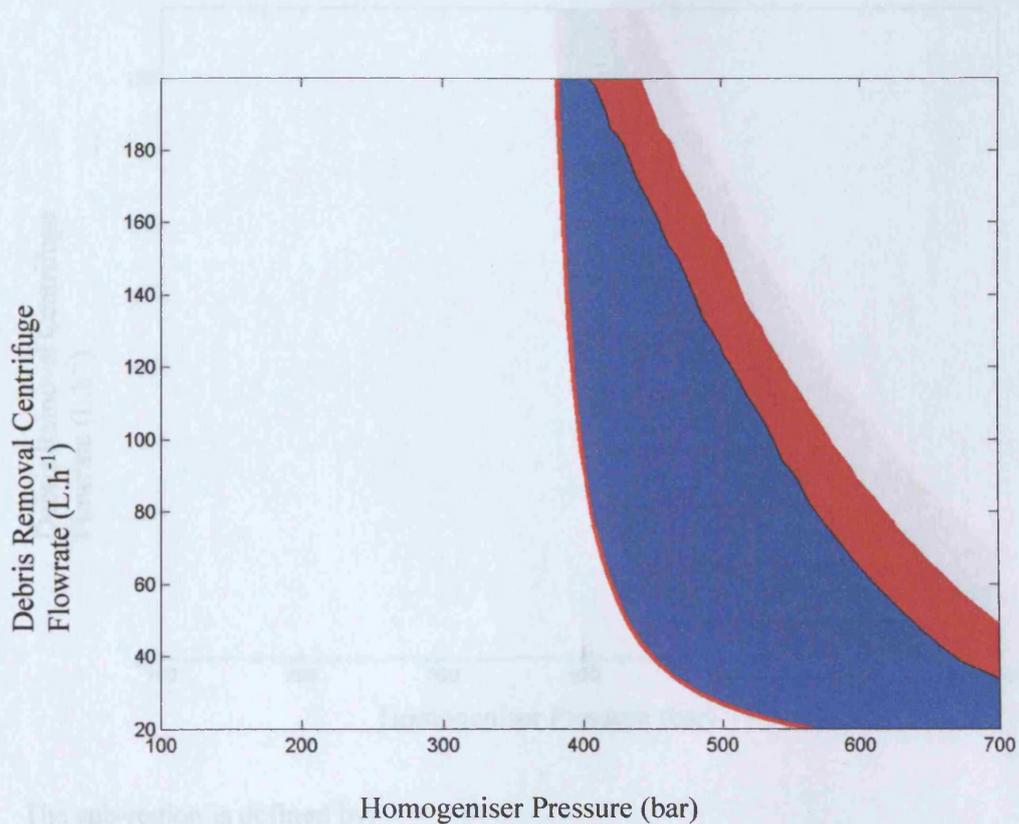
The sub-region is defined by: -

- growth rate - 0.18hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Yellow region - robust region given imprecision in the debris removal centrifuge flowrate of $\pm 10\text{L.h}^{-1}$, homogeniser pressure of $\pm 25\text{bar}$, dilution 1L^{-1} and growth rate 0.1h^{-1}

Figure 5.8: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in debris density parameter.



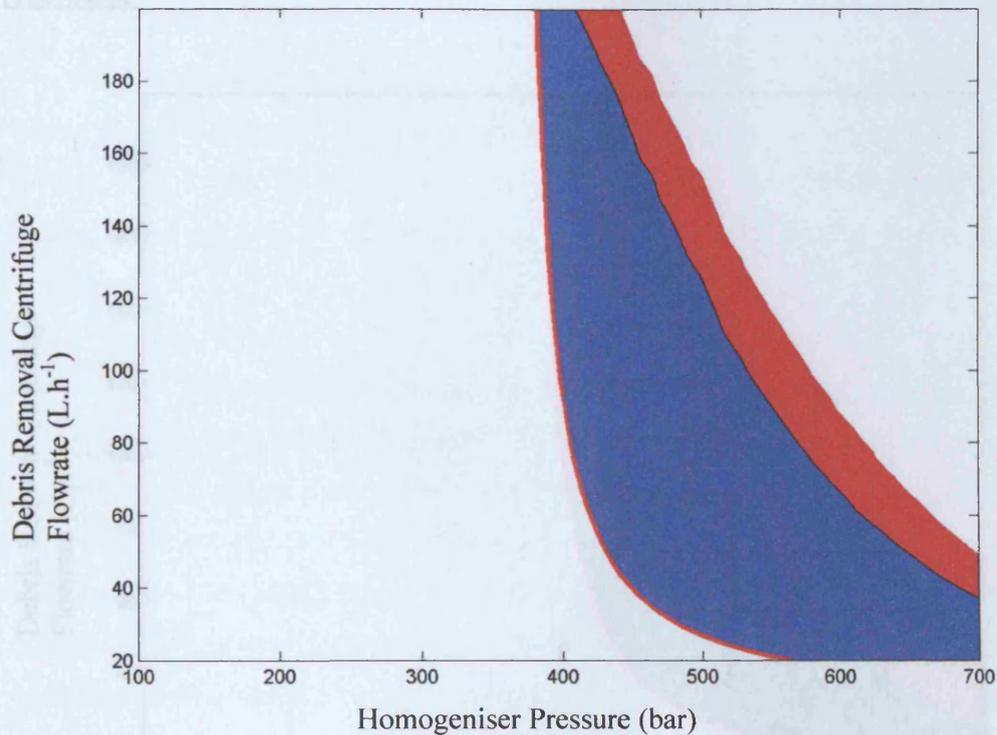
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Blue region - robust region given a 2% variation in debris density.

Figure 5.9: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the cell breakage coefficient.



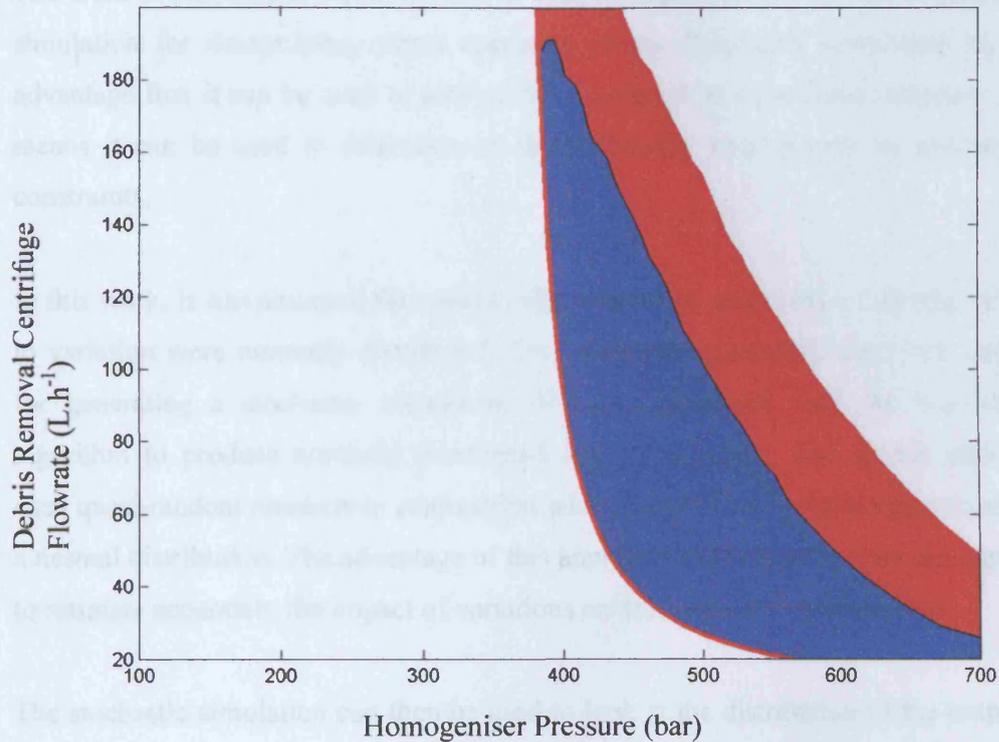
The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Blue region - robust region given a 10% variation in the cell breakage coefficient.

Figure 5.10: The robust regions defined by homogeniser pressure and debris removal centrifuge flowrate given imprecision in the debris density and the cell breakage coefficient.



The sub-region is defined by: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The feasible region is defined by defined by a minimum ADH production of 5×10^6 units and a maximum debris concentration 0.5gL^{-1} . The regions on the graph are: -

- Red region - the feasible region
- Blue region - robust region given a 2% variation in the debris density and a 10% variation in the cell breakage coefficient

6 Stochastic Simulation and Analysis

6.1 Abstract

The work in this chapter builds on that in Chapter 5 and looks at the use of stochastic simulation for determining robust operating points. Stochastic simulation has the advantage that it can be used to look at the likelihood of a particular outcome. This means it can be used to determine the probability of a bioprocess meeting its constraints.

In this work, it was assumed that each of the parameters and control variables subject to variation were normally distributed. Two different approaches were then applied for generating a stochastic simulation. The first approach used the Box-Muller algorithm to produce normally distributed random numbers. The second approach used quasi-random numbers in conjunction with an algorithm for calculating points on a normal distribution. The advantage of this approach is that fewer points are required to estimate accurately the impact of variations on the bioprocess performance.

The stochastic simulation can then be used to look at the distribution of the output or performance variables at different operating points. The approach taken in this work was to assume that the output variables are approximately normally distributed. Using this assumption two approaches were then used to analyse the simulation. The first approach was to visualise a two-dimensional subset of the feasible region.

The second section of the chapter uses optimisation to examine the trade-off between achieving robustness and maximising process performance. A non-greedy algorithm was used for the optimisation routine. Two versions of the algorithm were applied to the case study; the first algorithm is a standard algorithm that treats all the variables as continuous, the second algorithm was developed to handle discrete variables.

The work in this chapter demonstrates the potential insight that can be gained through using stochastic simulation. Equally it shows how the combination of stochastic

simulation and optimisation has the potential to find robust operating points in multi-variable bioprocesses.

6.2 Introduction

The previous chapter demonstrated that variability in both the control variables and model parameters will have a significant impact on the performance of a bioprocess. However the limitation of the scenario-based approach is that it does not consider the likely variability of each control variable or model parameter. An engineer can only assign ranges for each control variable and parameter. The temptation for an engineer will be to select large parameter ranges; however, this is likely to lead to a sub-optimal design.

An alternative approach is to treat the parameters as having a distribution and this can be achieved by generating a stochastic simulation. This chapter looks at both the generation of a stochastic simulation and how such a simulation can be analysed to find robust operating points for a given flowsheet.

A stochastic simulation can be generated by running multiple simulations using different control and parameter values within a given distribution. The results can then be analysed to calculate the most likely outcome, the variation, and even risk of an undesirable outcome. When there are multiple parameters, each subject to uncertainty, it is not possible to look at all combinations of parameter settings. In this work the approach that is used to address the problem is that of Monte Carlo integration. Monte Carlo Integration was demonstrated in Chapter 4 when analysing the integral of the feasible region. In this chapter it is used to analyse the uncertainty space defined by uncertain parameters and by variations in control variables.

Monte Carlo simulation has been applied in many situations including analysis of scheduling and resource allocation (Farid *et al*, 2000). However, there is very little work looking at its application in accurate modelling of bioprocess unit operations. One notable exception is the work by Uesbeck *et al*. (1998). However, this work only looked at a single fermentation step and not a bioprocess sequence. In their conclusion

they acknowledged that the work needs to be extended to incorporate the impact of downstream steps.

This chapter examines how the approach of Monte Carlo Integration can be used in the analysis of the interactions between unit operations. It also shows how it can be used to evaluate suitable operating regions for different levels of risk. Finally the work demonstrates how Monte Carlo Integration can be applied to evaluate the general trade-off between maximising economic performance whilst ensuring process robustness.

6.3 Theory

6.3.1 Stochastic Modelling

A bioprocess operating at a specific set point will be subject to variation in both control variables and parameters. Therefore an engineer will need to consider the impact of such variations when selecting an operating point. When evaluating such a problem the engineer will have to make a trade-off -- normally between process robustness and process performance. In the previous chapter, the problem was dealt with by assuming that the parameters and control variables could be defined by a fixed range. However, an alternative approach is to treat the parameters and control variable variations as being distributed variables.

Typically the output of a process will be dependent on a series of parameters and control variables e.g. $(y_i(x,\theta))$. When both the control variable and the parameters are subject to variability the expected value of the output can be calculated by integrating the output function multiplied by the probability function: -

$$\bar{y} = E_{\theta}(f_u) = \int_{\theta} f_u(\theta) \cdot j(\theta) d\theta \quad \text{Equation 6.1}$$

where \bar{y} is the average value for an output variable, $E_{\theta}(f_u)$ is the expected value for the function (f_u) , $f_u(\theta)$ is the function of f_u for a given parameter value (θ) and $j(\theta)$ is the probability of the parameter value (θ) .

Such a problem can be evaluated by using Monte Carlo Integration to estimate the integral by calculating both the value of the function ($f_u(\theta)$) and distribution function ($j(\theta)$) at a series of random points. However, often in such problems the probability distribution function will already be known. Hence the sampling set can be skewed to select from values that are more likely to occur. The advantage of this is that less simulation time is spent evaluating points that are unlikely to contribute significantly to the integral.

There are a number of techniques that can be used to calculate random numbers that fall into such distributions. One of the simplest is to use an inverse normal distribution function, which converts raw probabilities into the corresponding numbers of standard deviations from a mean (Acklam, 1999). This technique can be used to transform a set of random numbers into a normally distributed set of values. A second, and more efficient approach, is the Box-Muller technique (Press *et al*, 2002).

As well as being able to calculate the average, Monte Carlo simulations can also be used to calculate the standard deviation. If it is assumed that the output variable can be approximated by a normal distribution then it is possible to determine the probability of a constraint being broken in the course of a simulation using the equation below: -

$$\bar{y} + \Phi^{-1}(\gamma)\sigma_y \leq 0 \qquad \text{Equation 6.2}$$

where γ is the desired probability for a constraint, $\Phi^{-1}(\gamma)$ is the inverse normal distribution algorithm and σ_y is the standard deviation in the output variable (y).

The work in Chapter 4 showed that in a potential limitation of a Monte Carlo Integration was the accuracy of the results. In order to obtain increased accuracy a larger number of simulations are required. It was shown that increasing the number of samples (m) will decrease the error by $m^{-1/2}$. However, with large problems with more variables the error is likely to increase because of increased variance.

One solution to this problem is to use quasi-random numbers in Monte Carlo simulations. Quasi-random numbers are generated to fill uniformly a multi-dimension cube and reduce the risk of bias in a particular sample. The advantage of this approach is that the error of the integral is proportional to $\log(m)/m$. This results in better estimates of the integral for less computation effort. An example of such an algorithm is the Sobol generator and an overview of this is given by Bratley and Fox (1988).

6.3.2 Analysis of Uncertainty

A Monte Carlo simulation enables the prediction of both the expected outcome and the likely risk. As discussed in the literature survey, a significant body of research has been carried out investigating the impact of uncertainty. A review of optimisation under uncertainty for chemical engineering problems is given by Sahinidis (2004). However that paper primarily focuses on integer and linear programming. It only gives a passing mention to non-linear and non-convex problems typically found in bioprocesses.

In this thesis, stochastic simulation was used to determine a trade-off between the performance and reliability of an example process. The problem can be thought of as one of a multi-objective optimisation. Typically a bioprocess will be subject to a series of constraints. However, variation in the underlying process means that no operating point completely removes the risk of a constraint not being met.

The approach developed in this thesis was first to calculate the stochastic simulation. The results from the stochastic simulation were then used to create an objective function. The results of the objective function are then in turn used by an optimisation routine which searches for the optimum set of control variables. This approach is similar to that used by Bernardo and Saraiva (1998), which uses an optimisation routine with a stochastic simulation sub-routine. The advantage with this approach is that the two problems are dealt with separately, therefore reducing the complexity of the overall problem. However, one drawback is that a full stochastic simulation must be run at each point in the optimisation. Therefore overall time required by the algorithm will be the product of the time required by the stochastic simulation and the time required by the optimisation.

In this chapter an objective function was created that combines the performance and robustness criteria. The optimisation problem was run several times using different parameters in order to analyse the trade-off between robustness and performance. The example used in this work has one constraint that needs to be met robustly though the approach could easily be extended for multiple constraints. More details on the computation methods are given in the next section.

Most bioprocesses are likely to be non-linear. Bioprocesses are also likely to have non-convex feasible regions, meaning that their constraints will form a region where a chord between two feasible points may need to pass outside the feasible region. This means that any problem is likely to have several local optima. Such problems are often difficult for traditional optimisation techniques, such as Newtonian and Quasi-Newtonian methods. This is because these routines will look for the best downhill move at each point. However the best downhill move is often likely to lead to the routine finding a local optimum. The routines are said to be “greedy”. Often the only way such routines can solve such problems is by using multiple re-starts at different points which is computationally intensive.

The optimisation approach examined in this work was simulated annealing, which is non-greedy. This is because the simulated annealing algorithm occasionally allows uphill moves as determined by the ‘Temperature’ of the optimisation routine. This enables the routine to avoid being trapped in local optima and means that simulated annealing can find a global optimum even when there are many local optima. The next section gives more detail of the underlying computation methods used to solve these problems. Details on how the architecture of the problem was resolved are also included.

6.4 Computation Methods

6.4.1 Introduction

The work in this section can be divided into two parts. The first looks at the stochastic simulation and the changes required to convert the C++ version of the alcohol

dehydrogenase simulation into a stochastic simulation. The second looks at methods for analysing the trade-off between robustness and performance.

The first section (6.4.2) analyses the methods used for generating the stochastic simulation and investigates the use of object-orientated design. The work also looks at different approaches for generating the Monte Carlo simulation and in particular the application of the Box-Muller algorithm and the Sobol sequence, techniques used respectively for generating normally distributed random numbers and Quasi-random numbers.

The second section (6.4.3) looks at methodologies used to analyse the trade-off between performance of the process and the robustness of the process. This work builds on the previous stochastic simulation work and uses the results of the simulation to evaluate the trade-offs.

6.4.2 Stochastic Simulation

The objective of this work was to develop a simulation of the case study process (described in Chapter 3 and Appendix A) that was able to incorporate the effects of uncertainty in process models. This was done by running multiple trials using normally distributed random numbers for several variables. One of the main challenges identified at an early stage was the ability to generate a sample of random numbers with a normal distribution. Two different approaches were used: -

- Normally distributed random numbers using the Box-Muller algorithm.
- Normally distributed numbers from the Sobol Sequence.

Both of these approaches require the development of their own simulations and object-orientated programming was used to model the case study process in both cases. This enabled the problem to be broken down into logical sub-units. At the top level was the simulation class which contains the entire simulation. Other classes created for use in the simulation were: -

- Classes to represent the unit operations.
- A Stream Class to represent the streams contents.

- A “Statistical” class designed for calculating the average and standard deviation.
- Random number generating class.

This class structure is illustrated in Figures 6.1 and 6.2, which show the relationship between the different classes used. Details on each of the classes are given in the sections below.

6.4.2.1 The Simulation Class

The simulation class was created so that any analysis work using the simulation could be performed using an instance of this class. The class had a series of methods for: -

- Setting each of the control variables in the process.
- Setting the number of runs in the stochastic simulation.
- Running the simulation.
- Returning values calculated by the previous simulation run.

The “simulation” class was designed such that all the other objects required for the simulation would be initialised when the class was created. The initialisation therefore creates a series of stream objects and unit operations. These objects are created on the free-store with pointers to them being held internally in the class.

The main objective of the Simulation Class was to perform a Monte Carlo simulation of the process. This Monte Carlo simulation was carried out by running multiple trials of the simulation, each using different values for the parameters and control variables subject to uncertainty. In each run the mass balance and general property calculations are performed for each of the unit operations. At the end of each trial, each of the output variables is held in a corresponding instance of the “statistical” class. After the final trial, the “statistical” objects are used to calculate the average and standard deviation for each output variable.

In this work, the primary interest was to obtain the highest level of performance whilst ensuring that there was minimal risk of the constraints being broken. This was calculated using the formulae below: -

$$f_{P(i)} = \Phi\left(\frac{c_i - \bar{y}_i}{\sigma_i}\right) \quad \text{Equation 6.3}$$

where $f_{P(i)}$ is the probability of a constraint being met, Φ is the function for evaluating the normal distribution, c_i is the value of the constraint, \bar{y}_i is the expected (or average) value of the constrained variable and σ_{yi} is the expected standard deviation in the constrained variable.

The function for evaluating the normal distribution was taken from work by Marsaglia (2004).

6.4.2.2 The Unit Operations and Streams

The design of the unit operations and streams is specified in chapter 3. The streams were designed to hold the values that described the material flowing between each of the unit operations. The unit operations themselves were designed with a series of methods for: -

- Setting the control variables.
- Setting the input and output streams.
- Calculating material balance and changes in stream properties.

In the previous simulation the parameters and variables were treated as normal variables. In this work the parameters and control variables had to vary between runs. To achieve this, two new classes were built that generated normally distributed random numbers and quasi-random numbers. These classes are described in greater detail in the next section.

The unit operation classes were changed, so that the variables that were subject to variability were defined as instances of this class. This approach meant that internal

changes to the unit operations were minimised. Nevertheless new methods were added to each of the classes affected so that the distribution and average values of each of these uncertain parameters and control variables could be specified.

6.4.2.3 Random Variable Classes

In this work two classes (“Random” and “Quasi-Random”) were developed to calculate the normally distributed random number and quasi-random numbers. Both classes were designed with the following methods: -

- Setting the average and standard deviation of each variable.
- Setting minimum and maximum ranges on the variables.
- Returning a number when the object is used for calculations.

The methods for setting the average and standard deviation were incorporated so that a distribution profile could be defined for each instance of the class. The minimum and maximum values that were used ensured that random numbers did not lie outside set ranges. This was, for example, to stop variables from having negative values. Finally a method was built so that the object would return its current random value so that it could be used in the simulation.

Both the classes also had methods for calculating the normally distributed random numbers or quasi-random numbers for a particular run of the simulation. However, here there were substantial differences between the two classes. The simplest approach was that used by the “Random” class. This class had a method called “Reset” that ran the Box-Muller algorithm (Press, 1999) to calculate a normally distributed set of random numbers. This value was then converted into a value for the particular parameter using the average value and standard deviation. This was stored internally and returned every time the object was used in an equation.

The “Quasi-Random” class was calculated using the Sobol sequence and required a more complex approach. Unlike random numbers, quasi-random numbers follow a sequence meaning the class needs to keep track of how many times it has been called. Equally the numbers returned will be dependent on how many other random variables there are in the simulation.

To handle this, the “Quasi-Random” class had three static variables and two static functions. Static variables and function are not applied to a specific instance of the class (like normal variables and functions) instead they are shared by all instances of the class. The first static variable was used for storing both the number of instances of the class and therefore was incremented each time a new instance of the “Quasi-Random” class was initialised. This static variable was also used to set an index number for each instance of the class. The second static variable was used to track the current step in the Sobol sequence. Finally a static array was used to store all the values of the Sobol sequence for a particular step.

The class also had two static functions called “Reset” and “Next” for calculating the random variables. The “Next” method is run to calculate a new set of “Quasi-Random” numbers. This function increases the static variable used to track the current step in the Sobol sequence. The function then puts the calculated values for the current step of the Sobol sequence into the static array. The “Reset” function was used to restart the Sobol sequence.

When the object is used in an equation it uses index number to retrieve the appropriate value from the static array. It then uses an inverse normal distribution algorithm in conjunction with its average and the standard deviation values to calculate a value. To improve performance this value is cached in the object after it is calculated to ensure that the same calculation is not run multiple times.

6.4.2.4 Statistics Class

The “Statistics” class is used to hold a list of values and then calculate the average and standard deviation.

After each run of the simulation, the “statistics” object stores each of the values that are given to it in a linked list. The advantage of using the linked list is that, unlike an array, it can be dynamically resized giving considerably more flexibility. When the simulation has finished the object interrogates its list and calculates both the average

and standard deviations using the values stored in the list. At the end of each stochastic simulation the statistics object has its lists purged thus freeing up memory.

6.4.3 Analysis of the Stochastic Simulation

The second part of this work was evaluating the trade-off between the robustness and the performance of the system. The main problem with the process used for the case study is that the problem is non-linear and the feasible region defined by the debris constraint is non-convex. This makes it unsuitable for most traditional optimisation techniques as they are greedy and are likely to get trapped in a local optimum. Therefore the approach used in this work was to create a single objective function containing the constraints and optimise it using a simulated annealing algorithm. The objective function that was created included the following terms: -

- The performance term to be maximised.
- Penalty function based on the risk of breaking a constraint.
- Penalty function to ensure the control variables were inside their ranges.

By varying the penalty function based on the risk of breaking a constraint and re-running the optimisation routine, it should be possible to find a series of points corresponding to different levels of performance and robustness. This basic approach could be applied to any bioprocess.

Simulated annealing is based on an analogy with the thermodynamics of a liquid metal solidifying. Traditional greedy optimisation routines are analogous to quenching, where the metal is cooled quickly resulting in a suboptimal end product. To overcome this limitation, simulated annealing allows occasional uphill moves proportional to a 'temperature' property meaning it is analogous to a controlled cooling. The optimisation routine uses a high temperature initially to ensure that it does not fall into a local optimum. However, this temperature is decreased as the routine moves toward the optimum.

The version of the simulated annealing algorithm used was that given by Press *et al* (2002) based on the Nelder-Mead algorithm. The algorithm works by adding a random number onto each vertex of the simplex proportional to the 'temperature'

property. After a set number of iterations the temperature is reduced by a cooling factor meaning that uphill moves become less likely. This procedure is repeated until either the algorithm meets the convergence criteria or exceeds the maximum number of iterations. This algorithm has four properties that need to be set: -

- The initial temperature – This is the initial temperature of the system and determines the initial probability of an “uphill” move.
- The number of iterations per cycle - This is the number of iterations that will be at one temperature before the temperature is reduced.
- The cooling factor – This is determined by the factor at which the temperatures is reduced after the number of iterations has been exceeded.
- The maximum number of iterations – This is the maximum number of iterations before the algorithm will terminate (regardless of whether it has converged on an optimum point or not).

One limitation of this algorithm is that it treats each of the control variables as continuous. However in the example process, one of the control variables (the number of passes through the homogeniser) is actually integer. This necessitates that a second algorithm be developed.

This algorithm made a slight modification to the algorithm described by Press *et al* (2002) so that it could deal with integer variables. This was done by effectively treating the continuous variables as a subset of the control variables. The algorithm works by using the method proposed by Press *et al* (2002). However, after the cycle at a particular temperature is run an additional step is used to change the integer variable.

This step first uses a random binary to decide whether the integer variable should be allowed to increase or decrease in this iteration. Depending on this value the algorithm will evaluate the objective function at the lower or higher value. It will then determine whether moving the integer variable in this direction will improve the objective function. However, the decision on whether to move to the new value is based on a combination of this value and the random variable generated using the temperature property. This approach in theory could be expanded to allow for

multiple integer variables. In many respects this new algorithm is similar to the algorithm described by Cardosa *et al* (1997), although their approach was limited to binary integer variables.

The next section looks at the results obtained using the methods described in this chapter.

6.5 Results

6.5.1 Introduction

The results look at the application of the stochastic model described in the previous section and are divided into two sub-sections. The first section looks at visualisation of the data generated by the stochastic model whereas the second section looks at the trade-off between robustness and maximising production.

The visualisation was used to determine whether the data generated by the stochastic simulation appears to be broadly accurate. The visualisation was carried out by writing two simple programs. The first program calls the stochastic simulation object and writes out the returned values to a series of text files. The text files were then imported into Matlab and visualised using a second program. Two examples were looked at: -

- A simple example where only homogeniser pressure and debris removal centrifuge flowrate were subject to variation.
- A complex example where a range of control variables and model parameters were subject to variation.

6.5.2 Visualisation

Figure 6.3 looks at the simple case where only homogeniser pressure and debris removal centrifuge flowrate are subject to variation. The graph shows a plot of the homogeniser pressure against centrifuge flowrate and a series of different levels of certainty in the debris constraint being met. The most confident region (coloured deep blue) represents a region where there is a less than 1% chance of the debris constraint

being broken. The two red lines represent the points where there is a 50% chance of either the debris constraint or the alcohol dehydrogenase constraint being broken.

The first thing that can be seen from the graph is that the lines are not smooth because the number of points used to plot this contour is relatively low (20×20). As can be seen from the graph, the problem is particularly acute when plotting high and low probability contours. This is because probabilities range between zero and one meaning that the contour algorithm has less gradient information and therefore it cannot accurately interpolate at such values.

Figure 6.4 overcomes this problem by using more points (100×100) but at the cost of a significantly increased computational time required to generate a plot. An alternative strategy for solving this problem would be to use another utility for generating these plots that could use the results of the simulation directly. Figure 6.4 shows that there is still some jaggedness in the lines that define low and high probabilities of the constraints being met. This is caused by the simulation itself since when low numbers of trials are used in the Monte-Carlo Integration there will be a relatively high error.

Figure 6.5 shows that increasing the number of simulations solves this problem though again at the expense of increasing the number of trials in a simulation and therefore the time to generate the graph is also increased. An alternative strategy is to use quasi-random numbers. Figure 6.6 shows the same graph generated using 100 trials using quasi-random numbers. The main advantage of using quasi-random numbers is that it requires a lower number of trials to generate an accurate integration with little bias.

In figure 6.6 the jaggedness is removed by ensuring that each time the stochastic simulation was run it used the Sobol sequence with the same starting point. This means that each of the points on this graph is based on the same random number sequence. The danger of this approach is that if small numbers of runs are used in each stochastic simulation then the results will have a constant bias and therefore be

misleading. However, here the number of trials is large (100) for two variables subject to uncertainty meaning that any bias should be small.

So far each of the graphs has only looked at the situation where there are two control variables subject to uncertainty. However this approach scales well to problems with larger numbers of uncertain parameters and control variables. This is demonstrated in figure 6.7, which looks at the variation in a number of control variables and parameters. In this work it was assumed that the control variables could be controlled relatively accurately. For this simulation the following control variables were all assumed to have a standard deviation of 2%: -

- growth rate
- harvest centrifuge flowrate
- homogeniser pressure
- dilution ratio (after the homogeniser)
- debris removal centrifuge flowrate

Figure 6.7 also looks at the variation in a number of parameters. For this work it was assumed that they all had a standard deviation of 5%: -

- alcohol dehydrogenase in the cell
- protein in the cell
- alcohol dehydrogenase degradation
- protein release coefficient
- cell breakage coefficient
- pressure exponent for cell breakage
- debris size from homogeniser
- viscosity of the homogenate

Additionally, debris density was also assumed to be subject to variation. However, this was assumed to have a standard deviation of 5g/L.

One limitation of the work here is that each of the variations of the uncertain parameters and control variables is based on simple assumptions. For such analysis to

have real application the estimates for the variation in the parameters should be generated whilst the models are being built and the experiments carried out.

In Figure 6.7 there are several interesting features; in particular there is much greater variation than Figure 6.6. This is because there are additional factors that will affect the amount of debris and the ability of the process to collect the debris. However, the variability is not as large as might be expected and this is probably because many of the uncertain variables in this simulation do not directly affect the debris collection. This can also be attributed to constraints being calculated by looking at the distribution of output variables rather than assuming fixed ranges for control variables and parameters subject to uncertainty, as was done in the previous chapter. Using simple ranges for each of the parameters and control variables subject to uncertainty can lead to a very over-constrained problem.

Another advantage of this approach is that it can be used to look at the trade-off between robustness and maximum productivity. This can allow an engineer to determine whether the process will be able to produce enough of the product with the desired quality, consistently.

6.5.3 The Trade-off between Robustness and Performance

The optimisation tried to find the best operating points for various acceptable levels of uncertainty in the process. The optimisation routine was run to determine the best operating strategy within the following ranges: -

- growth rate $0.03 - 0.25\text{h}^{-1}$
- harvest centrifuge flowrate $100 - 700\text{Lh}^{-1}$
- homogeniser pressure $100 - 600\text{bar}$
- homogeniser passes $1 - 7\text{passes}$
- dilution ratio $0 - 2.5\text{L.L}^{-1}$
- debris removal centrifuge flowrate $20 - 200\text{Lh}^{-1}$

The objective function for this case study used: -

- alcohol dehydrogenase as the performance term.

- A penalty term for the probability of the debris constraint being broken.
- A penalty term to stop the search moving outside the operating space.

In this work a number of different objective functions were looked at. The first approach looked at a combination the three terms in a simple linear arrangement e.g: -

$$f_{obj} = f_{ADH} + w \cdot f_{P(Debris)} + f_{Penalty} \quad \text{Equation 6.4}$$

where f_{obj} is the objective function being optimised, f_{ADH} is a function for the amount of ADH produced, $f_{P(Debris)}$ is the function for calculating the probability of the debris constraint being broken, $f_{Penalty}$ is the penalty function for moving outside the operating space and w is the weight variable.

The idea here was that the weight value could be varied in order to locate a series of points, each with a decreasing amount of alcohol dehydrogenase but an increased probability of the constraint being met. However, this approach seemed to be only able to produce results at the two extremes regardless of what values the weight was set at. As a result, a second approach was tried. This approach used a penalty function when the probability of the debris constraint not being met was above a critical threshold. The penalty function used was linearly proportional to the violation of this threshold. E.g.: -

$$f_{obj} = f_{ADH} + \max(w \cdot (f_{p(Debris)} - c_{p(Debris)}), 0) + f_{Penalty} \quad \text{Equation 6.5}$$

where $c_{p(Debris)}$ is the debris probability threshold.

In this approach the weight (w_f) was not varied. Instead different values of debris probability constraint (c_p) were used. The problem with this approach is that if a high value of w_f is used then potentially it breaks up the feasible region forming steep valleys that the simulated annealing algorithm cannot climb out of. Therefore the penalty function was modified to: -

$$f_{obj} = f_{ADH} + \max\left(\frac{w \cdot (f_{P(Debris)} - c_{P(Debris)})}{(1 + (f_{P(Debris)} - c_{P(Debris)}))}, 0\right) + f_{Penalty} \quad \text{Equation 6.6}$$

The advantage of this penalty function is that it sets a maximum limit on the penalty size (w_f). This means that it should, in theory, avoid creating barriers in the search space that the simulated annealing algorithm cannot escape. Ultimately Equation 6.6 was the objective function used in the work, with the units of alcohol dehydrogenase normalised by dividing the value by 10^7 . The debris penalty function was also set up so that it would have a minimum value of zero and a maximum value of 5. The stochastic simulation used had the variable parameters with the same standard deviations as those used to generate figure 6.7.

In this work the final objective function was used in conjunction with the two optimisation techniques defined in section 6.4.3. The simulated annealing profile is controlled by a number of parameters. The optimisation techniques were run with the parameters defined in Table 6.1 below. Each of the combinations of parameters was then run ten times for each of the probability constraints on the debris concentration. The three different conditions were used in case one set of parameters was not able to locate an optimum.

Table 6.1: The parameters used during optimisation

	Optimisation Parameters 1	Optimisation Parameters 2	Optimisation Parameters 3
Initial temperature	5	5	5
Number of iterations per cycle	20	20	50
Cooling factor	0.98	0.985	0.985
maximum number of iterations	5000	5000	6500

The initial temperature of 5 was selected as this corresponded approximately to the largest uphill move using the objective function. A relatively low number of iterations was used per cycle because the second optimisation routine needs to evaluate the integer variables between cycles. To compensate for the low numbers of iterations in each cycle a high cooling factor was selected giving a gentle cooling profile.

These two techniques were then run using a series of different values of acceptable risk for the debris constraint being broken. In both cases good results were obtained using this approach, when the probability of the debris constraint being broken was between 0.002 and 0.5. The results for the two approaches are shown in Figures 6.8 and 6.9. As can be seen from the graphs that the majority of the points are close to the optimum value. However, a number of points are a long way from this point. This is because sometimes the routine appears to stop at points below the optimum.

In theory lower cooling and a higher number of iterations per cycle should result in better convergence at the optimum point. However, there is no obvious trend that can be seen from Figures 6.8 and 6.9. This may reflect that ten runs for each set of parameters at each of the points along the curve was not enough to find a trend. It is possible that trends could be seen if a greater number of runs were carried out. Equally it is possible that there is not enough variation between the parameters and therefore none of the sets of parameters are significantly better. If the latter is true then it would seem sensible to use the first parameter set as this is likely to require the fewest iterations.

A comparison between Figure 6.8 and Figure 6.9 shows that the optimisation routine, which is designed to handle the integer variables, is more likely to converge sub-optimally. This seems to indicate that the new algorithm is not particularly efficient. One possible limitation of this algorithm is that the integer variable is examined less often and is restricted to either making an upward or downward move depending on the random binary variable. Consequently the algorithm probably could be designed to be more efficient.

Figure 6.10 is a comparison of the results obtained from the two processes. The results obtained when the number of passes is allowed to have any value are slightly better than the results obtained when the number of passes is limited to integer values. This is most probably due to the higher degree of freedom when selecting an operating point. In both cases however, a relatively small decrease in productivity will

yield a more robust process indicating that the process can be operated in a robust manner without a significant drop in productivity.

Using this approach we can also look at the position of the optimum point and how this changes when better performance is required. Table 6.2 shows the best results for each of the different acceptable levels of risk of the debris constraint not being met.

Table 6.2: The best results for each of the different levels of risk of the debris constraint being broken.

Probability of Failing to Met the Debris Concentration	Growth Rate (h ⁻¹)	Harvest centrifuge Flowrate (L.h ⁻¹)	Homogeniser Pressure (Bar)	Homogeniser Number of Passes	Dilution Ratio (L.L ⁻¹)	Debris Removal Centrifuge Flowrate (L.hr ⁻¹)	Amount of ADH (units) × 10 ⁶
1	0.1796	283.2	600	4.15	0	200	15.34
0.5	0.1803	304.7	600	2.90	1.27	72.1	12.08
0.1	0.1803	289.3	600	2.75	1.44	67.8	11.66
0.05	0.1804	286.5	600	2.71	1.48	66.7	11.56
0.02	0.1804	283.9	600	2.68	1.51	65.6	11.45
0.01	0.1804	282.1	600	2.65	1.53	65.1	11.38
0.05	0.1804	280.2	600	2.62	1.55	64.4	11.32
0.002	0.1805	278.6	600	2.59	1.57	63.8	11.25

The results in this table show a number of interesting features. If we ignore the impact of the debris constraint (e.g. the probability of failure is one) then we can see that the optimum point lies at the maximum pressure, a relatively high number of passes, no dilution and a fast debris removal centrifuge step. This fits with our existing understanding of the process. Higher pressure and a larger number of passes results in more product being released. However, a large number of passes may result in the greater degradation due to longer processing times. Equally, high dilution and centrifuge flowrates will also increase the processing time and therefore the loss of product.

When the debris concentration constraint is set then the optimum point is shifted. The new optimum point has a lower number of passes, a lower debris removal centrifuge flowrate and a higher dilution ratio. This is because a lower number of passes will

reduce the amount of micronised debris. Equally, increasing the dilution ratio and decreasing the centrifuge flowrate and number of passes will ensure that more debris can be removed. In this work it can be seen that changing these settings will ensure greater robustness.

One unexpected trend is that when the robustness criterion is increased the harvest debris flowrate should be decreased. Lower harvest centrifuge flowrates are associated with higher recovery of cells but greater loss of product through longer processing. It is possible that lower harvest centrifuge flowrates become more desirable when the debris concentration constraint is increased to compensate for the longer processing times in later steps. However, previous work has shown that this control variable will have a relatively small impact on the performance of the process.

As discussed earlier these results would only be realistic as they assume that the homogeniser is in a continuous mode. However often the number of passes is limited to an integer amount. The results in Table 6.3 show the performance using the second optimisation routine which limits the number of passes to integer values.

Table 6.3: The best results for each of the different levels of risk of the debris constraint being broken when the number of homogeniser passes is treated as an integer variable.

Probability of Failing to Met the Debris Concentration	Growth Rate (h ⁻¹)	Harvest centrifuge Flowrate (L.h ⁻¹)	Homogeniser Pressure (bar)	Homogeniser Number of Passes	Dilution Ratio (L.L ⁻¹)	Debris Removal Centrifuge Flowrate (L.h ⁻¹)	Amount of ADH (units) ×10 ⁶
1	0.1798	283.5	600	4	0.00	200.00	15.34
0.5	0.1802	304.8	600	3	1.31	71.65	12.08
0.1	0.1802	287.7	600	3	1.51	64.96	11.63
0.05	0.1802	284.2	600	3	1.56	63.58	11.52
0.02	0.1802	279.7	600	3	1.61	62.10	11.40
0.01	0.1802	277.7	600	3	1.65	61.44	11.32
0.05	0.1802	274.9	600	3	1.68	60.53	11.26
0.002	0.1802	272.9	600	3	1.69	60.17	11.18

This work shows similar trends to those shown previously. The number of passes through the homogeniser is rounded up to three when the debris constraint is enforced. Equally the dilution rate is higher and the debris dilution rate is lower than when the number of passes through the homogeniser was not constrained to integer variables. This is probably because they need to compensate for the number of passes being higher than they would be if the number of passes was not an integer variable.

This work shows that stochastic simulation can be used in conjunction with optimisation to look at the trade-off required to operate a process robustly. This can give some insight into how effectively the bioprocess is capable of performing as well as a potential set of operating conditions that will ensure both a higher productivity and a certain level of robustness.

6.6 Conclusions

In the previous chapter, it was shown that the impact of variation in parameters and control variables could be examined by using a series of worst-case scenarios for the control variable and parameters. However, this approach could lead to sub-optimal processes if large ranges are selected for each of the control variables or parameters. With this in mind this chapter looked at the use of stochastic simulation and how this can be applied to analysing a bioprocess. The advantage of stochastic simulation is that it is scalable meaning that potentially very large numbers of uncertain variables and parameters can be looked at simultaneously. Additionally this work can be used to make assumptions about how likely a constraint is to be broken with a particular operating strategy.

The first part of the work looked at visualising the results of such a simulation. Two examples were taken. The first looked at the impact of two control variables. As the impact of these two control variables are well understood it was used to gauge the effectiveness of this approach. The second example looked at a situation that included multiple control variables and parameters subject to uncertainty. The visualisation work shows that stochastic simulation can be used to find a subset of the feasible region that will actually be robust.

The work also shows that Monte Carlo simulations will be subject to bias unless a large number of random points are used (as demonstrated in Figure 6.4 and 6.5). This can be avoided by using a higher number of trials or alternatively using quasi-random numbers. The advantage of quasi-random numbers is that the points are selected in such a way that they are distributed evenly across the sample space and hence reduce bias.

The second part of this work looked at the trade-off between robustly achieving a constraint and maximising process productivity. The approach used here was to generate an optimisation function that contained a penalty term whenever the probability of the constraint being broken was deemed unacceptably high. The objective function was then optimised using a simulated annealing algorithm selected for its ability to deal with problems with local optima.

In this work two version of the optimisation routine were run. The first uses a simulated annealing algorithm, which treats all the variables as continuous variables. However, the case study selected contained one control variable which could only have an integer value. Therefore a modified optimisation routine was developed that was able to deal with integer variables as well as continuous variables. The latter approach tended to converge on sub-optimal points which suggest further refinements may be necessary to improve performance of the routine.

In this work both optimisation routines were run using different optimisation parameters, which control the rate of cooling in the algorithm. However, with the limited number of runs using each combination of parameters, no conclusions could be drawn about the best parameter setting for the algorithm.

The main limitation of the stochastic simulation is the assumptions that had to be made about the distributions of the control variables and parameters. Future work should look at techniques for estimating the uncertainty from experimental data. Another limitation is the assumption that the output variables will be approximately normally distributed. Algorithms need to be investigated for when the distributions of the output variables exhibit skew.

Overall the work in this chapter has demonstrated that such analysis can be used to show a trade-off between robustness and performance and give an indication of the capabilities of the process. In the case study looked at here the analysis indicates that the bioprocess being studied could be operated in a much more robust manner at the expense of a relatively modest drop in productivity. The approach was able to highlight a series of operating strategies that maximise process performance for various levels of risk. Ultimately if this approach could be used in conjunction with better model development strategies, it would have great potential for bioprocess design.

Figure 6.1: A collaboration diagram showing how each of the objects in the stochastic simulation, using random variables, interacts.

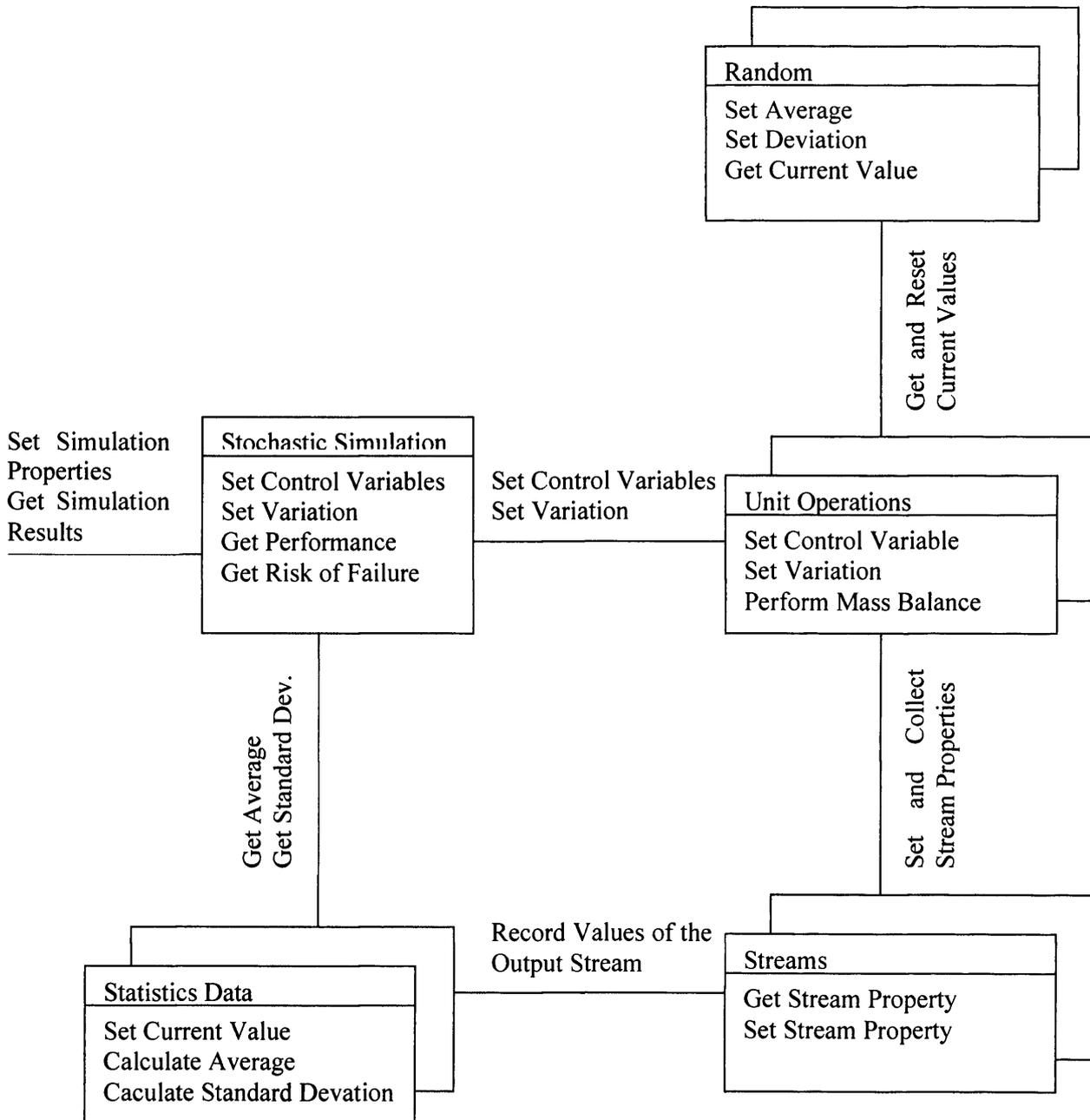


Figure 6.2: A collaboration diagram showing how each of the objects in the stochastic simulation, using quasi random variables, interacts.

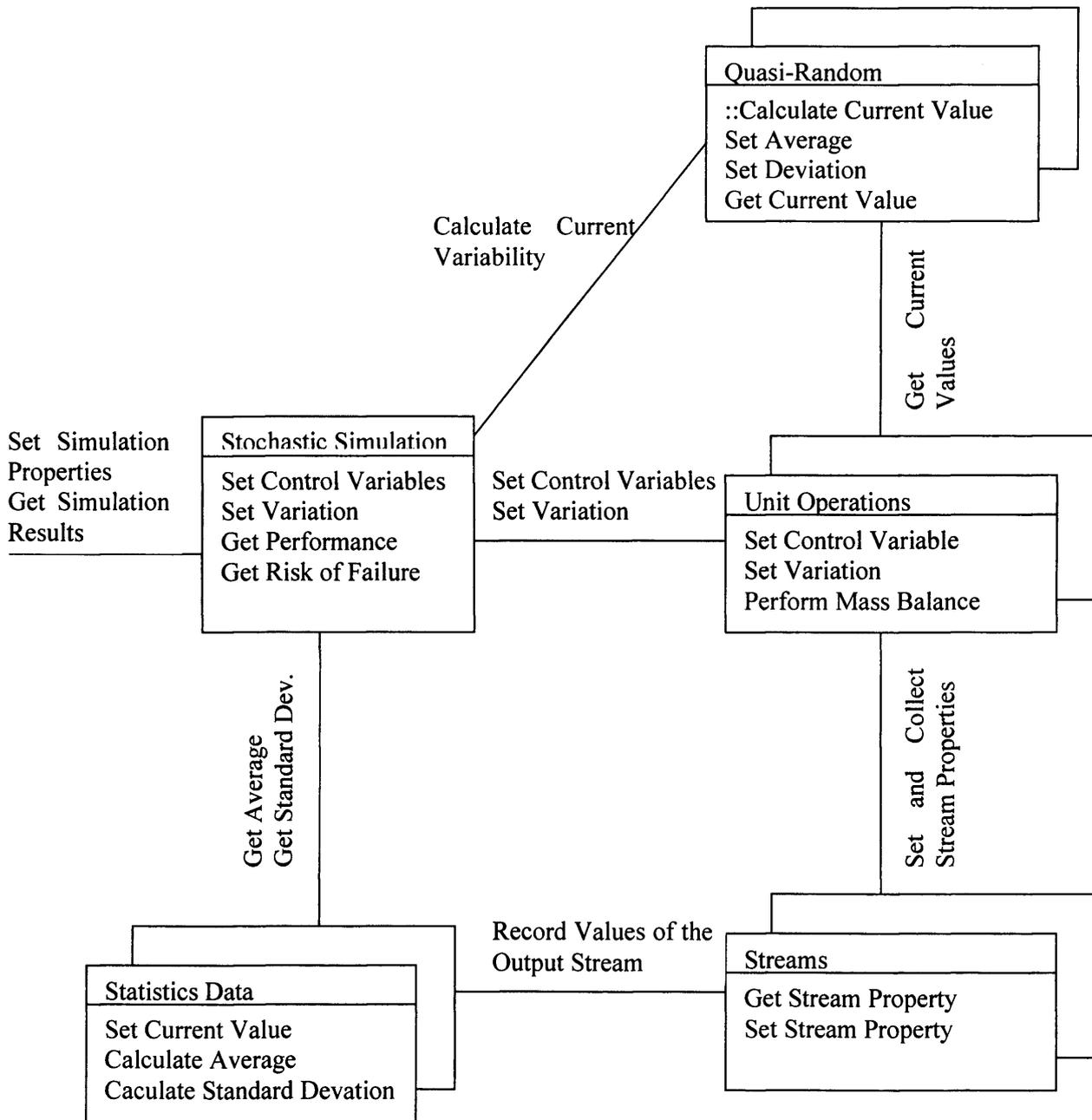
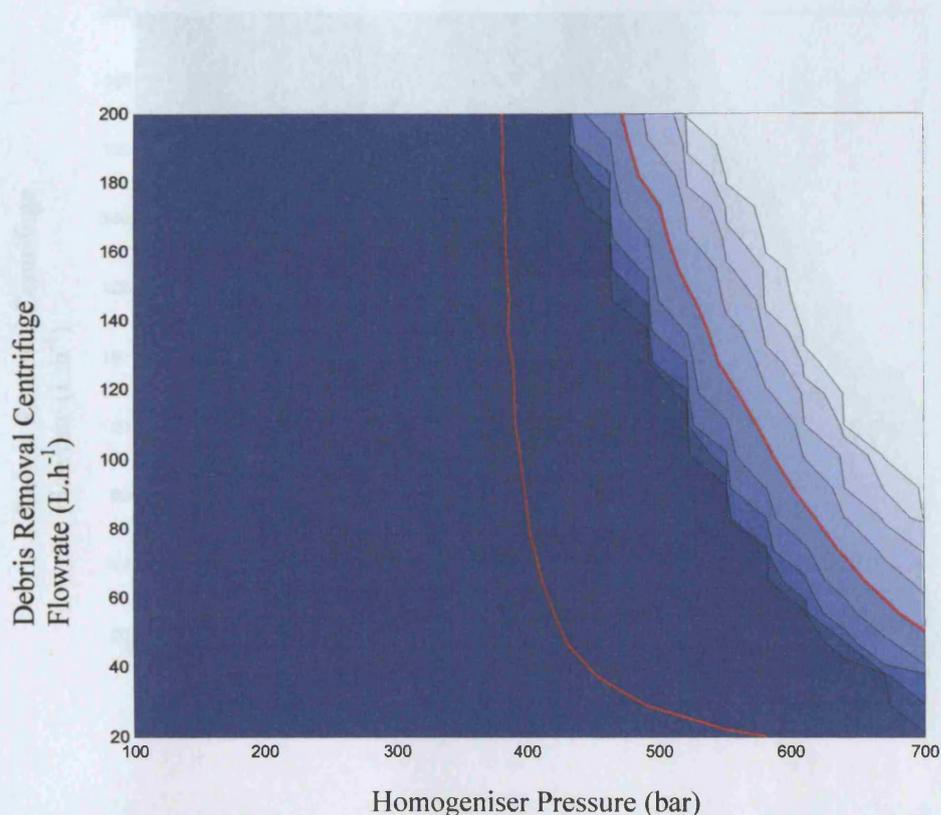


Figure 6.3: The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure with a graph generated using 20x20 points, each calculated using 100 trials with random numbers generated using the Box-Muller algorithm.

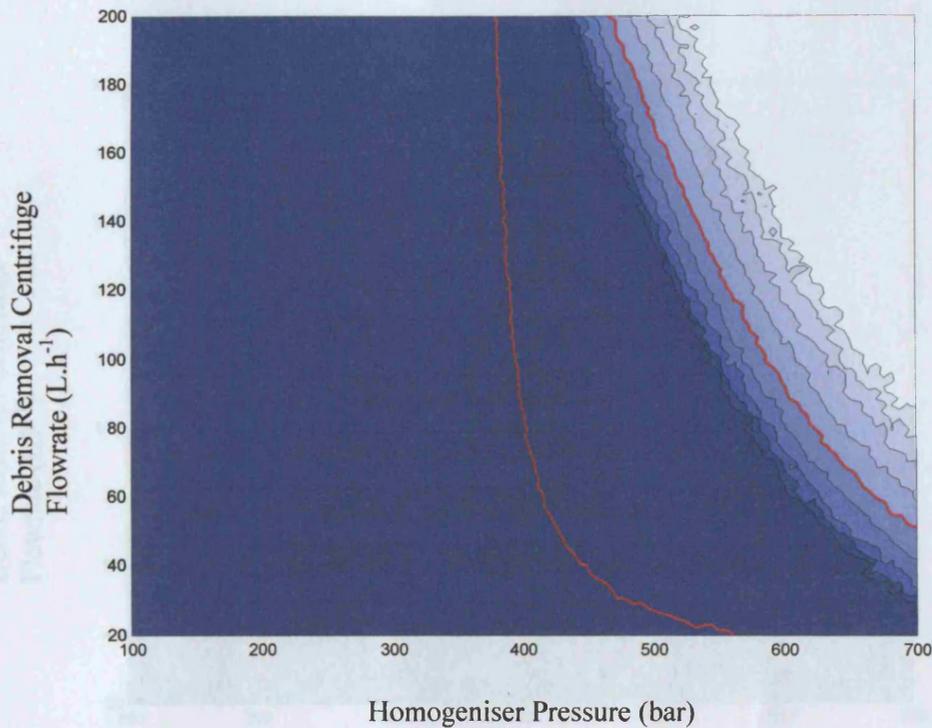


The sub-region is defined by homogeniser pressure and debris removal centrifuge flowrate where: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The constraints represented by the red lines are $>5 \times 10^6$ units ADH and a debris concentration 0.5gL^{-1} . The shades of blue represent the probability of the debris constraint being broken with the probabilities of 0.1% (darkest), 1%, 10%, 90%, 99% and 99.9% (lightest).

Figure 6.4: The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure. This graph was generated using 100x100 points, each calculated using 100 trials with random numbers generated using the Box-Muller algorithm.

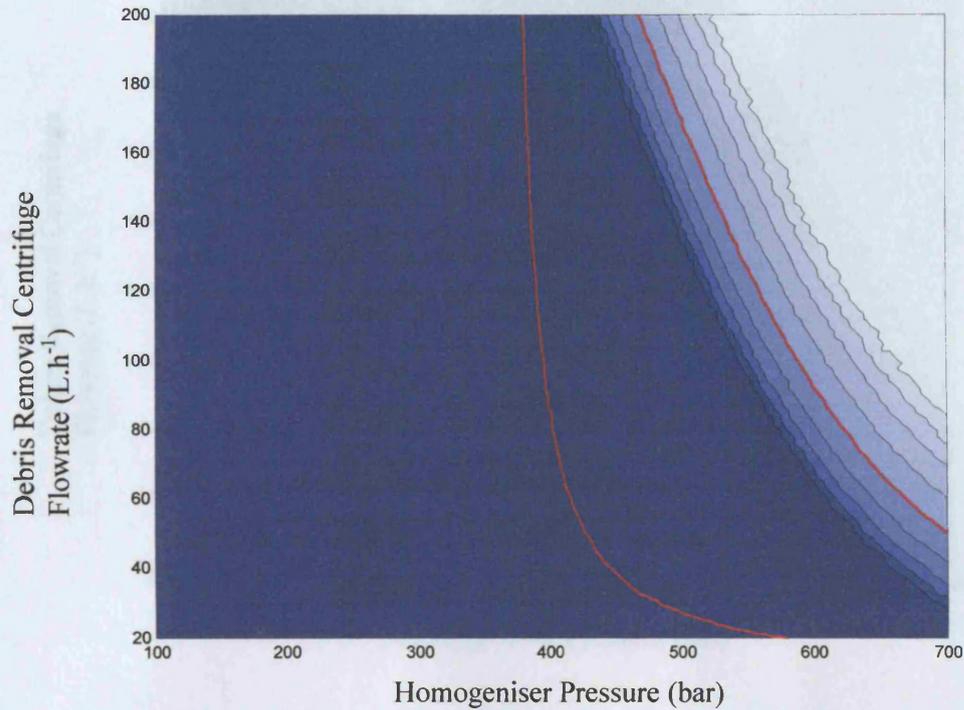


The sub-region is defined by homogeniser pressure and debris removal centrifuge flowrate where: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The constraints represented by the red lines are $>5 \times 10^6$ units ADH and a debris concentration 0.5gL^{-1} . The shades of blue represent the probability of the debris constraint being broken with the probabilities of 0.1% (darkest), 1%, 10%, 90%, 99% and 99.9% (lightest).

Figure 6.5: The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation. This graph was generated using 100x100 points, each calculated using 1000 trials with random numbers generated using the Box-Muller algorithm.

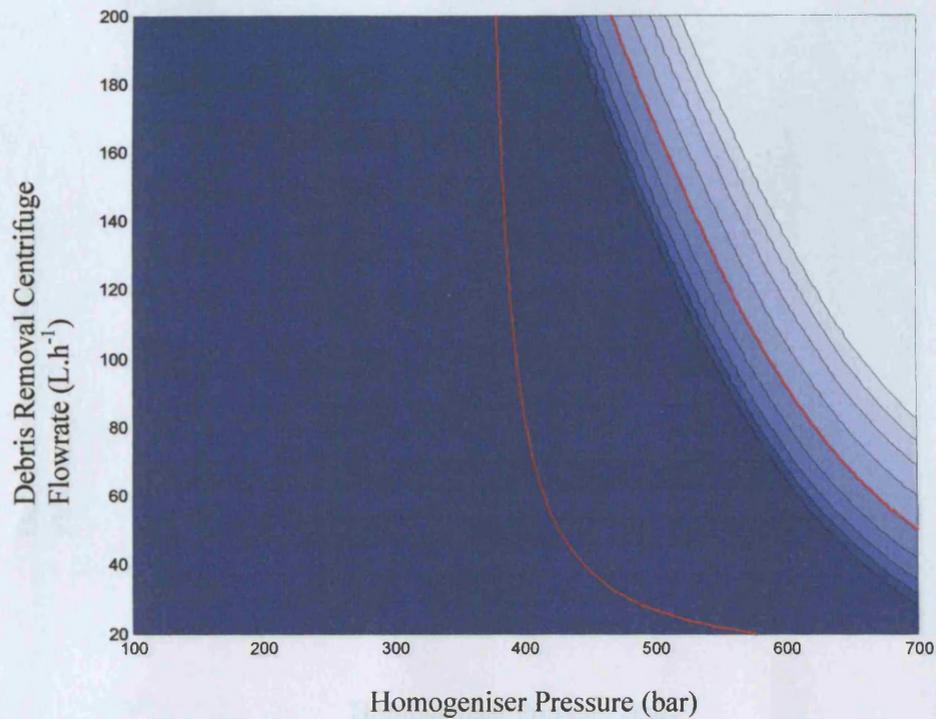


The sub-region is defined by homogeniser pressure and debris removal centrifuge flowrate where: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The constraints represented by the red lines are $>5 \times 10^6$ units ADH and a debris concentration 0.5gL^{-1} . The shades of blue represent the probability of the debris constraint being broken with the probabilities of 0.1% (darkest), 1%, 10%, 90%, 99% and 99.9% (lightest).

Figure 6.6: The robust region defined by 2% variation in debris removal centrifuge flowrate and homogenisation pressure. This graph was generated using 100x100 points, each calculated using 100 trials with quasi random numbers generated using the Sobol sequence and an inverse normal distribution algorithm.

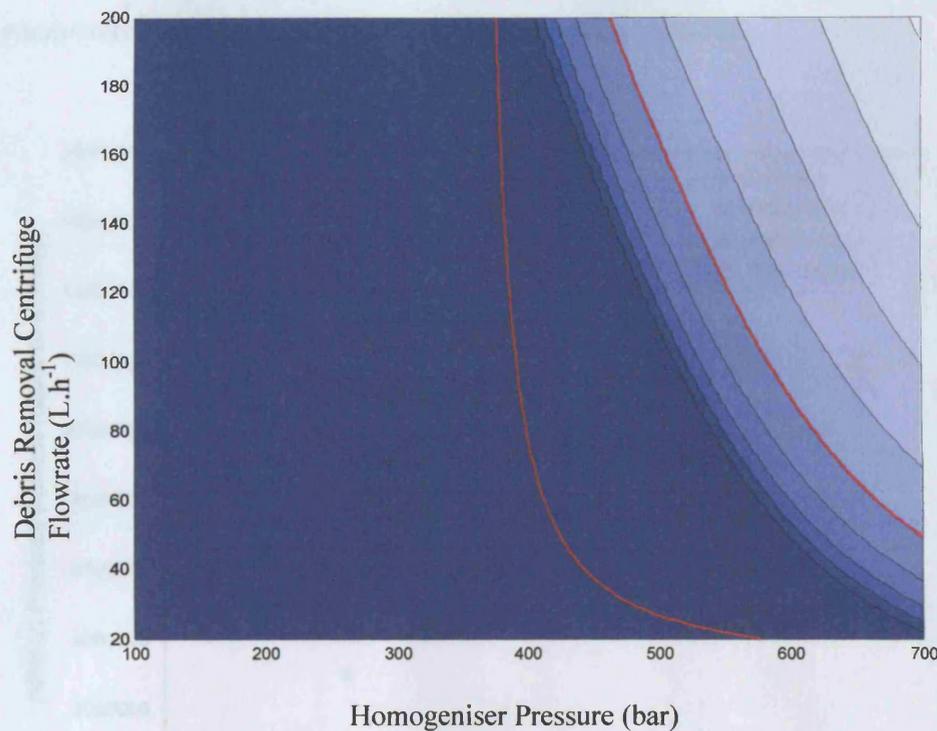


The sub-region is defined by homogeniser pressure and debris removal centrifuge flowrate where: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The constraints represented by the red lines are $>5 \times 10^6$ units ADH and a debris concentration 0.5gL^{-1} . The shades of blue represent the probability of the debris constraint being broken with the probabilities of 0.1% (darkest), 1%, 10%, 90%, 99% and 99.9% (lightest).

Figure 6.7: The robust region defined by variation in number of parameters and control variables. This graph was generated using 100x100 points, each calculated using 10,000 trials with quasi random numbers generated using the Sobol sequence and an inverse normal distribution algorithm.



The sub-region is defined by homogeniser pressure and debris removal centrifuge flowrate where: -

- growth Rate - 0.15hr^{-1}
- harvest flowrate - 300L.h^{-1}
- dilution ratio - 2
- homogeniser passes - 4

The constraints represented by the red lines are $>5 \times 10^6$ units ADH and a debris concentration 0.5gL^{-1} . The shades of blue represent the probability of the debris constraint being broken with the probabilities of 0.1% (darkest), 1%, 10%, 90%, 99% and 99.9% (lightest).

Figure 6.8: The results showing the maximum amount of alcohol dehydrogenase (units of activity) in the product stream against predicted probability of failure to meet debris constraint where the Number of Passes is treated as a continuous variable.

The graph shows the results obtained using three sets of simulated annealing parameters and the best overall results from all of the methods.

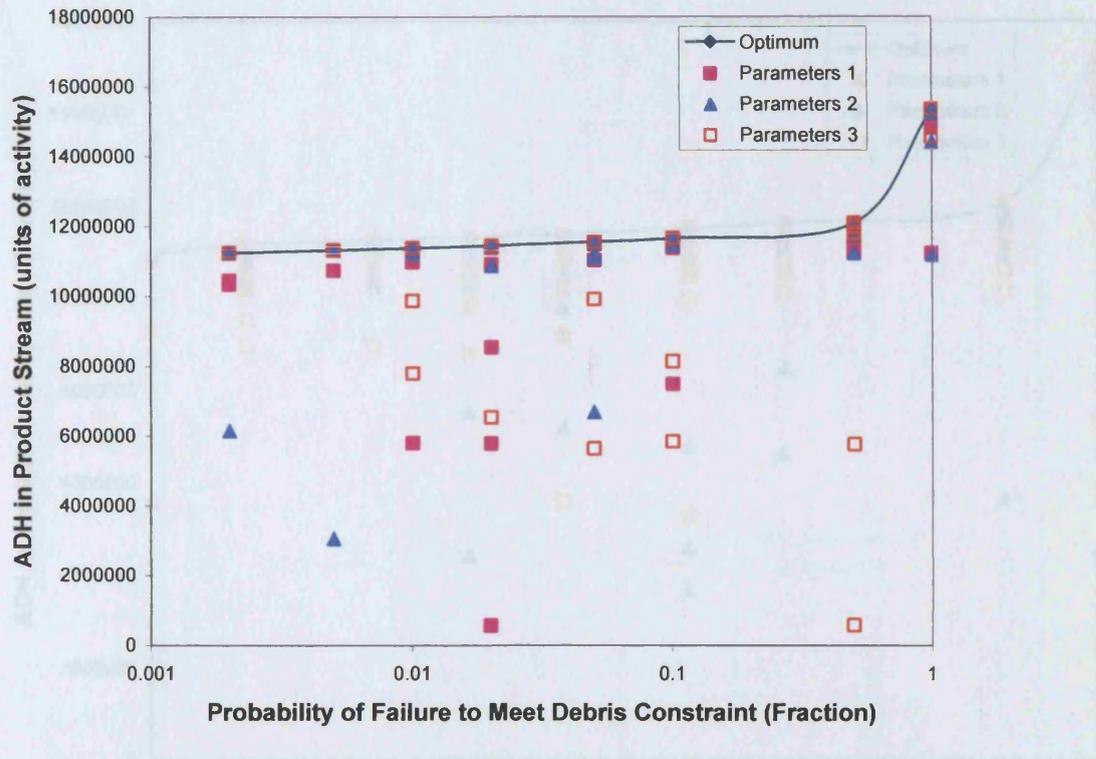


Figure 6.9: The results showing the maximum amount of alcohol dehydrogenase (units of activity) in the product stream against predicted probability of failure to meet debris constraint where the Number of Passes is treated as an integer variable.

The graph shows the results obtained using three sets of simulated annealing parameters and the best overall best results from all of methods.

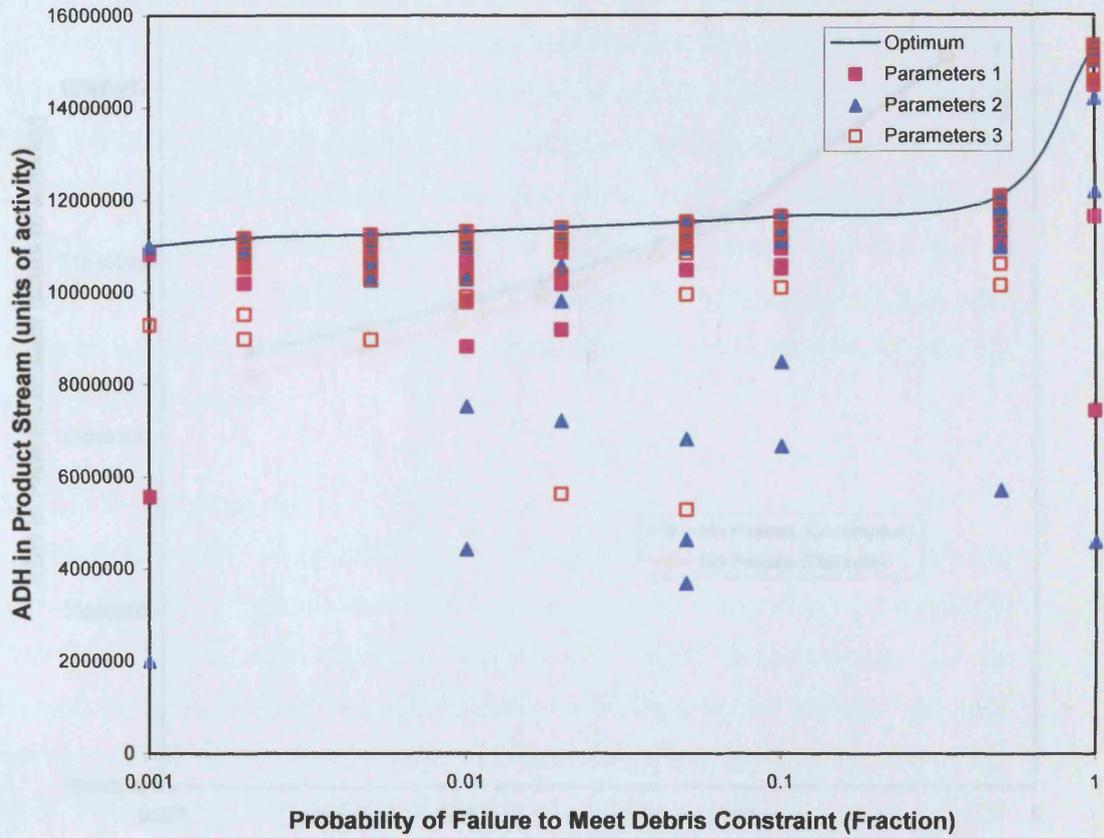
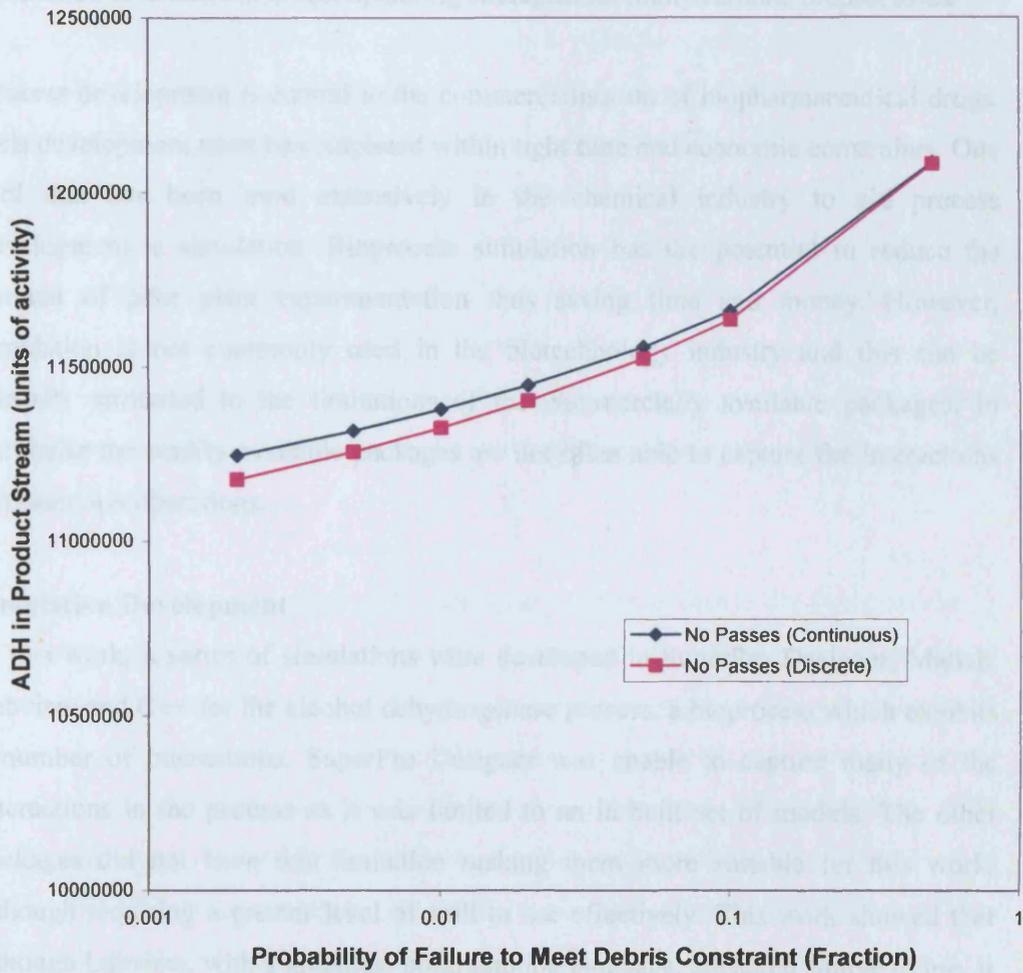


Figure 6.10: A Comparison between the results obtained looking at the trade-off between maximising alcohol dehydrogenase and probability of failure to meet the debris constraint, using the technique treating number of passes as a continuous variable (blue line) and as an integer variable (purple line)



Conclusions

This thesis has examined techniques that can be applied for analysing bioprocess flowsheets to determine operating strategies. The objective of the work was to investigate possible techniques that can be used in conjunction with bioprocess simulation to determine robust operating strategies for multi-variable bioprocesses.

Process development is central to the commercialisation of biopharmaceutical drugs. This development must be completed within tight time and economic constraints. One tool that has been used extensively in the chemical industry to aid process development is simulation. Bioprocess simulation has the potential to reduce the amount of pilot plant experimentation thus saving time and money. However, simulation is not commonly used in the biotechnology industry and this can be partially attributed to the limitations of the commercially available packages. In particular the readily available packages are not often able to capture the interactions between unit operations.

Simulation Development

In this work, a series of simulations were developed in SuperPro Designer, Matlab, Labview and C++ for the alcohol dehydrogenase process, a bioprocess which exhibits a number of interactions. SuperPro Designer was unable to capture many of the interactions in the process as it was limited to an in-built set of models. The other packages did not have this limitation making them more suitable for this work, although requiring a greater level of skill to use effectively. This work showed that although Labview, with a graphical programming language, appeared simple to use, it actually resulted in code that was harder to maintain. Matlab and C++ with their text-based languages resulted in models that were much easier to maintain. Matlab was the simpler of the two languages and probably the more suited for developing small simulations. However, C++ could offer performance advantages. Additionally C++ can potentially be used for developing a commercial package as the resulting code can be compiled to work on most computers.

Analysing Multidimensional Feasible Regions

The remainder of the thesis focused on analysing the results generated by the simulation. One approach that has been used previously by the bioprocessing industry is “Windows of Operation”. This is a two dimensional region defined by two control variables where series of constraints are met. The advantage of this is that it enables the user to evaluate the impact of these control variables on the process meeting its specifications. However, this approach is limited to two control variables whereas most bioprocesses have multiple control variables. Therefore this work looked at alternative methodologies, which can examine processes with multiple control variables. One approach was to investigate the size of the feasible region. The impact of control variable settings was examined by evaluating the distribution of the feasible region with respect to different control variables. The methodology was also used to evaluate the impact of constraints on the size of the feasible region. This demonstrated that higher performance resulted in a smaller feasible region which would be harder to operate robustly.

This work was then compared with techniques suggested previously for determining feasible ranges for the control variables. Examining the feasible ranges had been previously suggested by researchers in this area as a method for analysing batch processes. However, the work in this thesis suggests that a short-coming of calculating the feasible ranges is that it only captures a very small proportion of the total feasible space. Nevertheless both the feasible ranges and the size of the feasible region can give an insight into multi-variable processes and some indication of how feasible the process is together with an indirect indication of process robustness.

Scenario Based Approaches

In a bioprocess, variability can be divided into two types, variation in the settings of control variables and variation in process performance caused by uncertainty in the models. The first approach used looked at a series of scenarios that corresponded to the largest expected variation in each of the process parameters or control variables. The maximum variations of the parameters and controls were looked at simultaneously creating a hyper-rectangle of points, where each scenario

corresponded to a vertice. In this work it was assumed that if each of these “extreme” scenarios were feasible, then the corresponding operating point would be robust. The results of this approach were demonstrated by plotting a two dimensional robust region where all the points were robust, on top of a two-dimensional feasible region.

This technique has the advantage that it is relatively simple to implement and can highlight some regions where the process is guaranteed to be robust. In this work the scenario-based approach was only applied for visualising two-dimensional regions. However it could potentially be used in conjunction with analysis of the size of the feasible region to look for robust regions in a multivariable process.

Stochastic Modelling

In this work stochastic modelling was also investigated to see if this could overcome some of the limitations in the scenario based approach. The parameters and control variables that are subject to uncertainty were approximated by normal distributions. This was more realistic than treating each of the parameters as having a fixed range. Ideally such stochastic simulations should be used in conjunction with model building strategies that enable the calculation of the parameter variability. However any technique for calculating this variability would need to ensure that it did not increase the quantity of experimentation required or slow down process development.

In this work two approaches were used for generating the stochastic simulation. The first was to use random numbers and the second approach was to use quasi-random numbers. The latter approach has the advantage that the values selected will be well distributed throughout the region being evaluated. Both these techniques are applied to generate plots of a sub-section of the feasible region. This demonstrates how the two approaches can highlight regions where robust operating points are possible. However, this approach is limited to looking at a sub region of the operating space.

To overcome this limitation, the stochastic simulation was used in conjunction with an optimisation routine. In this work an objective function was built that contained the level of production in the process combined with a penalty function for when the probability of a constraint being broken was deemed to large. This was used to

analyse the trade-off between obtaining a robust process and maximising process performance. This shows how the location of the feasible region changes when greater levels of process robustness are required.

Ultimately this approach has the advantage that it can locate an operating point that maximises the process productivity whilst ensuring the process reaches a predefined level of robustness, given accurate estimates of the distributions of control variable and parameter variations. This means that this approach has the potential to analyse multi-dimensional processes whilst at the same time providing a robust operating point meaning that it meets both the original objectives of the research.

Future Work

The current work has shown that models can be developed for modelling and simulating bioprocesses. It has also show that complex strategies can be used to determine robust operating points given uncertainty in model parameters and process control. However, currently the application of advanced modelling strategies is likely to be consuming in terms of both labour and resources.

One key limitation of the advanced modelling work in this thesis was measuring the uncertainty in the models. The approach taken was to estimate using engineering judgement. However, this in part negates one of the key benefits of any systematic approach. Ideally estimating the errors in the models needs to be an integral part of the model and simulation building process.

Consequently future work will need to look at methods for reducing both the effort required to build such simulations and extracting data as effectively as possible. The list below highlights a number of areas that require further investigation to speed up the simulation development.

- **Simulation Building Methodologies**

In the current work, simulations have been developed on a case-by-case basis. This means when new simulations are developed, the existing unit operation models need to be re-coded. Work should be carried out to investigate developing a standard architecture.

This would involve developing a common stream definition to carry data between modules. This would enable to models to be built that would not be specific for one process simulation. In turn this would enable greater reuse of code and therefore faster simulation development. A common stream definition would require analysis of a number of different processes to determine what properties such a stream should contain.

- **Model Development**

Although models have been developed for many unit operations, others have not yet been thoroughly examined. These unit operations should be studied so that models can be developed. Existing models also need to be further developed, to make them more comprehensive and better able to capture interactions in the process.

Ideally the unit operation models should be developed so they can be easily adapted for different processes, using data from scale-down experiments. This could be done using object-orientated programming. For example, a well-designed class architecture would enable the development of new models that inherit properties and methods from previous unit operation models.

- **Measuring Uncertainty**

In this work the approach taken was to assume a certain level of uncertainty, as there was limited information on which to base these estimates. More work is required to develop techniques that are able to determine accurately the level of uncertainty in a model (whilst requiring little experimental effort).

This will in part be achieved through the use of scale-down to attain more experimental data, however, will probably also require strategies that are able to estimate the uncertainty in the model that is generated. Such work lay outside the scope of this thesis. However, it will be essential if better model building strategies are to be used in the future.

- **Simulation of Uncertainty**

The work in Chapter 6 treated each uncertain control variable and parameter as being normally distributed. In reality it is quite likely that the distributions for some parameters may not be normal. Ideally the approach used to measure the uncertainty in the model should be able to suggest how the model parameters are likely to be distributed. This information should be incorporated into the

simulation thus avoiding simplistic assumptions and providing a more accurate picture of the likely distribution of the robustness of the process.

Equally in this work only two approaches were looked at for simulating the distribution of the parameters. Future work will need to look at other techniques that could offer potentially better computational performance such as quadrature and advanced Monte Carlo algorithms that try to sample in areas that exhibit greater variations.

- **Calculating Robustness**

In the current work it is assumed that all the output variables were also normally distributed. Consequently the probability that a constraint will be met is determined from the number of standard deviations the expected value is from the constraint value and an inverse normal distribution function.

However, this is a simplification of the situation. In reality the output variables are unlikely to have a perfect normal distribution. Future work should look toward calculating the skew and kurtosis of the output variable in order to use these properties to calculate a more accurate estimate of the likelihood of a constraint being met.

- **Optimisation**

In this work an optimisation routine was used in conjunction with stochastic simulation. The optimisation technique used was simulated annealing. This technique is not greedy and hence may find a global optima rather than getting trapped in a local optima. The work also extended a simulated annealing algorithm to look at a mixed integer non-linear programming problem.

This work did not investigate setting the search parameters for the optimisation nor did it specifically investigate other potential optimisation techniques. These aspects could potentially be investigated in future work to see if better strategies could be used to attain the optimum point.

Another limitation was that the optimisation routine and the stochastic simulation took a long time to run. This could possibly be avoided by reducing the number of repeat simulations at the earlier stages of the optimisation routine and increasing this as the optimisation progresses.

Appendix A

A.1 Introduction

The simulated process at the heart of this thesis is shown in Figure 3.2. Processing starts with a fed-batch fermentation. The cells are then separated from the fermentation broth by centrifugation and ruptured using high-pressure homogenisation. The debris is removed from the homogenate again by centrifugation. This section gives an overview of the models used and the source of these models.

A.2 Fermentation

A.2.1 Overview

The first step in the simulated process was a 100 litre fed batch *Saccharomyces cerevisiae* fermentation. The model of the step was designed to examine cell growth and product formation as well as the impact of growth rate on downstream processing. Previous experimental work has shown that the growth rate of *Saccharomyces cerevisiae* has an effect on the physical properties of the cells (Siddiqi *et al.*, 1995; Siddiqi, 1996) and the expression levels of different intracellular proteins (Gregory *et al.*, 1996). Growth rate also affects the fermentation time, which has a critical impact on the production rate and therefore the process economics.

The fed-batch fermentation model used in the simulation is based on the experimental protocol described by Gregory *et al.* (1994). Based on this text the fermentation was assumed to have the following parameters: -

- Initial cell concentration – 0.16g L^{-1}
- Concentration of glucose in the feed – 500g L^{-1}
- Initial fermentation volume – 45L
- Final working fermenter volume – 70L
- The respiratory quotient was 1.1

The feeding strategy in this protocol is based on the Wang-Cooney model (Wang *et al.*, 1979), in which the fed rate is increased exponentially to obtain a constant growth

rate. The fermentation is assumed to proceed until the fermenter has reached its final working volume.

A.2.2 Wang Cooney Fed-Batch Model

The Wang Cooney model predicts the addition of substrate into the fermentation: -

$$Q_{fed} = \frac{\mu C_X V}{C_N Y_{x/s}} \varepsilon = \frac{\mu C_{X0} V_0 e^{\mu t}}{C_N Y_{x/s}} \varepsilon \quad \text{Equation A.1}$$

where Q_{fed} is the Fed Rate (L/hr), ε is the Respiratory Quotient, C_N is the concentration of the substrate (g/L), C_X is the concentration of the biomass (g/L), C_{X0} is the initial concentration of the biomass (g/L), $Y_{x/s}$ is the yield of biomass on substrate (g/g), V is the volume in the fermenter (L), V_0 is the initial Volume (L) and μ is the growth rate (h^{-1}).

This can be rearranged to give the volume at any given point in time: -

$$V_f = V_0 + \int_0^t Q_{fed} dt \quad \text{Equation A.2}$$

$$\Rightarrow V_f = V_0 + \int_0^t \frac{\mu C_{X0} V_0 e^{\mu t}}{C_N Y_{x/s}} \varepsilon \cdot dt$$

In order to calculate the amount of feed that is added, the yield for that specific growth rate must be known. In this work, the yield was calculated using a correlation estimated from data in Gregory *et al* (1996). This work showed that the yield drops off above a critical growth rate reflecting the transition to oxido-reductive growth. To capture this in the model an equation was generated that included a hyperbolic tangent term and the Nelder-Mead algorithm was used to calculate the parameters.

$$Y_{x/s} = 0.3608 - 0.0742\mu - 0.4471\mu^2 - 0.144 \tanh(107.65(\mu - 0.2005)) \quad \text{Equation A.3}$$

The Wang-Cooney model can be re-arranged to determine the time take for the fermentation (given the like maximum working volume). This in turn can be used to determine the cell concentration (Equation A.5 and Equation A.6).

$$t = \frac{1}{\mu} \ln \left(\left(\frac{V_f}{V_0} - 1 \right) \cdot \frac{Y_{X/S} C_N}{C_{X0} \varepsilon} + 1 \right) \quad \text{Equation A.4}$$

$$X_f = \frac{V_0 X_0 e^{\mu t}}{V_f} \quad \text{Equation A.5}$$

$$X_f = \frac{Y_{X/S} C_N}{\varepsilon} \left(1 - \frac{V_0}{V_f} \right) + \frac{V_0 X_0}{V_f} \quad \text{Equation A.6}$$

A.2.3 Intracellular Stream Properties

In this work, five stream properties were tracked through the system. These were DNA (g), protein (g), dry cell weight (g), cell wall/debris (g) and alcohol dehydrogenase (units). In this work, Equations A.7 and A.8 were derived using experimental work by Gregory *et al* (1996).

$$F_{\text{Prot/Biomass}} = 11.4 + 273.0\mu - 302.0\mu^2 - 10.72 \tanh(682.25(\mu - 0.185)) \quad \text{Equation A.7}$$

$$F_{\text{ADH/Prot}} = 5.49 - 8.374\mu + 10.02\mu^2 + 3.19 \tanh(-360.2(\mu - 0.1906)) \quad \text{Equation A.8}$$

where $F_{\text{Prot/Biomass}}$ is the fraction of the dry cell weight that is protein (g/g) and $F_{\text{ADH/Prot}}$ is the fraction of the protein that is alcohol dehydrogenase (g/g).

The remaining components were assumed to be a constant proportion of cell weight. The wet weight of the cell wall was assumed to be equal to the dry weigh of cells, as the wet cell weight is roughly three times the dry cell weight and the wall makes up approximately one third of this mass. Also the nucleic acids were assumed to make up a constant 10% of the dry cell weight.

A.2.4 Solid Properties

Work by Siddiqi (1996) on characterising cell and size distribution, showed that the average cell diameter (d_m) varied with growth rate with smaller cells produced under the oxido-reductive conditions. Equation 3.10 is a correlation giving the cell size at different growth rates and was generated using from data from Siddiqi (1996) containing parameters calculated by applying the Nelder-Meld algorithm.

$$\bar{\lambda}_{cell} = 5.3 - \tanh(-92(\mu - 0.1822)) \quad \text{Equation A.9}$$

where $\bar{\lambda}_{cell}$ is the average Cell diameter (μm).

The standard deviation in the cell size was a constant at $0.75\mu\text{m}$. In this simulation, the solids concentration was calculated from the wet cell weight, which is assumed to be three times the dry cell weight (Engelking, 2002). The cells are assumed to have a density of 1110g L^{-1} , based on previous experimental measurements.

A.2.5 Liquid Properties

The liquid properties were found to be unaffected by the fermentation conditions with the viscosity of the fermentation set at 0.00169Ns m^{-2} . The density of the liquid was set at 1010g/L .

A.2.6 Other Properties

One key factor that is affected by fermentation is the cell wall strength. The cell wall strength will determine how the cells behave in the homogenisation step. In particular it will affect how much intracellular product is released and the size of the cell debris produced.

The amount of product released is given by the Hetherington Equation (Hetherington *et al*, 1971). Siddiqi (1996) also looked at the impact of growth rate in the fermentation on both breakage constant and the pressure exponent. Using the results the following equations were generated: -

$$K_p = \exp(1.5 \cdot \tanh(-92 \cdot (\mu - 0.1822)) - 15.5) \quad \text{Equation A.10}$$

$$\alpha = 2.4 - (0.4 \cdot \tanh(-92 \cdot (\mu - 0.1822)))$$
 Equation A.11

where K_p is the protein release coefficient and α is the pressure exponent in the Hetherington Equation.

Siddiqi *et al* (1996) also developed an equation for predicting the size of the homogenised debris with contained a breakage coefficient (k_d). Siddiqi (1996) then ran a number of experiments looking at the impact of growth rate of how the debris fragments. From this the following equation was developed.

$$k_d = 1030 + 330 \tanh(-92(\mu - 0.1822))$$
 Equation A.12

where k_d is the cell breakage coefficient.

A.3 Centrifugation

A.3.1 Introduction

In alcohol dehydrogenase process simulation, centrifugation is used twice, to separate cells from the fermentation broth and later to remove the contaminant cell debris from the homogenate. The basic premise of the centrifuge is that it uses centrifugal force to separate an incoming feed stream into a solids heavy sediment stream and supernatant stream depleted in solids. This section looks at models for determining the separation efficiency of a disk stack centrifuge, so that the mass balance over the centrifuge can be calculated.

The simulation was based upon a disk-stack centrifuge with an equivalent settling area of 1465m² (CSA1, Westfalia). The centrifuge model allows the flowrate through the centrifuge to be varied, so that different centrifuge operating strategies can be examined.

A.3.2 Separation Performance

In this work, the separation of the solids is simulated by using the grade-efficiency model. In the grade-efficiency model, particles in the centrifuge are assumed to be subject to centrifugal forces pulling them out of solution and resistant drag forces determined by Stokes Law. At a critical size the centrifugal and drag forces will be balanced, the so-called critical diameter. Particles larger than a critical diameter should be 'theoretically' collected in the sediment, as their centrifugal forces will be greater than the drag forces. Equation A.13 can be used to calculate the critical diameter. The equation also accounts for hindered settling, where the concentration of particles reduces their ability to settle (Zaki and Richardson, 1954).

$$\lambda_c = \left[\frac{18Q'\eta}{(\rho_s - \rho_L) \cdot \Sigma \cdot (1 - C_v)^\beta A_g} \right]^{1/2} \quad \text{Equation A.13}$$

where λ_c is the critical diameter (m), η is the viscosity (Ns/m), Q' is the flowrate through the centrifuge (m^3/sec), ρ_L is the density of liquid phase (g/L), ρ_s is the density of solids phase (g/L), Σ is the centrifuge equivalent settling area (m^2), A_g is the acceleration due to Gravity (m/s^2), C_v is the volume concentration of solids in the suspension (L/L) and β is the particle geometric factor.

By knowing the critical diameter, the separation efficiency of particles of different sizes can be calculated using the grade efficiency curve ($\tau(d)$). Under ideal Stokes Law conditions, particles greater than the critical diameter should all be collected. However experiment work carried out with latex particles in a disk stage centrifuge showed that the efficiency curve varied from the predictions of the 'theoretical' model (Mannweiler, 1990). This data was used to generate a modified grade-efficiency curve: -

$$T(\lambda_p) = 1 - \exp \left[- \left(\frac{k_{GE} \cdot \lambda_p}{\lambda_c} \right)^w \right] \quad \text{Equation A.14}$$

where λ_p is the size of the solid particles, $\tau(\lambda_p)$ is the proportion of solids of that size collected in the sediment, k_{GE} is the coefficient in the grade efficiency model (0.865) and ϖ is the exponent in the grade efficiency model (2).

In most situations the solid particles will have a range of sizes, which can be modelled by using a normal distribution. The quantity of particles of a given size can be calculated from the mean particle size and the standard deviation: -

$$f(\lambda_p) = \frac{1}{\sigma_d \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{\lambda_p - \bar{\lambda}_p}{\sigma_d} \right)^2 \right] \quad \text{Equation A.15}$$

where λ_p is the particle diameter (μm), $\phi(\lambda_p)$ is the fraction of solids with a particular size in feed, $\bar{\lambda}_p$ is the average particle diameter (μm) and σ_λ is the standard deviations in particle size (μm).

The fraction of the solids that are collected by the centrifuge from the feed can be calculated using the equation below: -

$$F_{ss} = \int_0^{\infty} \phi(\lambda_p) \cdot \tau(\lambda_p) d\lambda_p \quad \text{Equation A.16}$$

where F_{ss} is the fraction of solids to collected in the sediment.

In a disk stack centrifuge system, the collected solid needs to be periodically discharged from the centrifuge. This discharge step leads to some of the liquid phase leaving the centrifuge in the sediment stream. In this model it is assumed that the entire contents of the centrifuge are discharged when the solid collected in the centrifuge bowl reaches a critical amount. Based on this, the number of discharges can be calculated: -

$$N_{Dis} = \left(\frac{C_s V_{feed} F_{ss}}{\rho_s} \right) / V_{sc} \quad \text{Equation A.17}$$

where N_{Dis} is the number of Discharges, C_S is the concentration of solids (g/L), ρ_S is the density of solids (g/L) and V_{SC} is the maximum volume of solids in the centrifuge (L).

The quantity of solid in the sediment predicted by equation A.16, however, this overlooks any additional solids that may be released from the bowl during discharge. In this work Equations A.18 and A.19, were derived to account for this phenomenon.

$$F_{LD} = \left(1 - \frac{C_s}{\rho_s}\right) \cdot \frac{N(V_C - V_{CS})}{V_{feed}} \quad \text{Equation A.18}$$

$$F_{SD} = F_{ss} + \frac{C_s \cdot N(V_C - V_{CS})(1 - F_{ss})}{C_s \cdot V_{feed}} \quad \text{Equation A.19}$$

where F_{LD} is the fraction of liquid discharged, F_{SD} is the fraction of solids discharge and V_C is the total volume of centrifuge bowl (L).

A.3.3 Mass Balance

As described previously, the stream is defined with the components distributed across three phases; the aqueous phase, the intracellular/biological phase, and the precipitant phase. The total flow of components is the sum of all the components in all the phases in a stream.

$$M_{T.sup} = (M_{T.feed} - M_{I.feed} - M_{P.feed}) \cdot (1 - F_{LD}) + M_{I.feed}(1 - F_{SD}) + M_{P.feed}(1 - F_{SD}) \quad \text{Equation A.20}$$

$$M_{I.sup} = M_{I.feed}(1 - F_{SD}) \quad \text{Equation A.21}$$

$$M_{P.sup} = M_{P.feed}(1 - F_{SD}) \quad \text{Equation A.22}$$

$$M_{T.dis} = (M_{T.feed} - M_{I.feed} - M_{P.feed}) \cdot F_{LD} + M_{I.feed} \cdot F_{SD} + M_{P.feed} \cdot F_{SD}$$

Equation A.23

$$M_{I.dis} = M_{I.feed} F_{SD}$$

Equation A.24

$$M_{P.dis} = M_{P.feed} F_{SD}$$

Equation A.25

where $M_{T,stream}$ is a vector of total mass of each component in the specified stream (g), $M_{I,stream}$ is a vector of intracellular components in the specified stream (g) and $M_{P,stream}$ is the vector of precipitant components in the specified stream (g). The subscripts *Dis*, *Feed* and *Sup* refer to the discharge stream, the centrifuge feed and supernatant stream respectively).

A.3.4 Volume

Volume is conserved over the centrifuge and therefore the volumes exiting each stream are calculated using equation A.26 & A.27.

$$V_{Dis} = \frac{M_{Solids.Feed} \cdot F_{ss}}{\rho} + \left(1 - \frac{C_s}{\rho}\right) \cdot N(V_C - V_{CS})$$

Equation A.26

$$V_{Sup} = V_{feed} - V_{Dis}$$

Equation A.27

where $M_{Solids.Feed}$ is the total mass in the solid phase in the feed stream (g), V_{Dis} is the volume leaving the discharge stream (L) and V_{Sup} is the volume leaving via the supernatant stream (L).

A.3.5 Concentration

The concentration of solid particles will change and hence the new concentrations can be calculated using Equation A.28 & A.29, for the supernatant and sediment streams respectively.

$$C_{s.Dis} = \frac{M_{Solids.Feed} \cdot F_{SD}}{V_{Dis}}$$

Equation A28

$$C_{s,Sup} = \frac{C_s V_{feed} - M_{Solids.Feed} \cdot F_{SD}}{V_{Super}} \quad \text{Equation A.29}$$

where $C_{s,Dis}$ is the concentration of solids in the discharge (g/L) and $C_{s,Sup}$ is the concentration of solids in the supernatant (g/L).

A.3.6 Time Taken

Higher centrifuge flowrates lead to shorter processing time (Equation A.30), resulting in less degradation of the product.

$$t = \frac{V_{feed}}{Q} \quad \text{Equation A.30}$$

where t is the Time (hr), V_{feed} is the volume of feed (L) and Q is the flowrate (L/hr).

A.4 Homogenisation

A.4.1 Introduction

In this unit operation the cells are forced at high pressure through a small valve causing the cells to rupture and release intracellular product. The performance of this unit operation is determined by two control variables, the homogenisation pressure and number of passes that the cells have through the valve. Higher pressure or an increased number of passes results in greater release of the intracellular components. However, this also causes the cell wall to break into smaller fragments, which are harder to remove via centrifugation. Consequently lower flowrates need to be employed in subsequent centrifugal stages to remove this debris which results in longer processing times. These trade-offs need to be considered carefully when assessing an operating strategy.

Homogenisation of yeast cells is dependent on upstream conditions, in particular fermentation growth rate and the concentration of the cells fed into the homogenate (Siddiqi, 1996). The impact of fermentation conditions has been examined and

consequently the simulation is able to predict changes in the cell properties that affect homogenisation. By contrast, the impact of cell concentration, although significant, has not been fully characterised. Hence the simulation assumes that all cells entering the homogenisation are diluted to a concentration of 450g/L.

A.4.2 Mass Balance

The amount of product released in this step is a function of the pressure drop and number of passes. The pressure drop over the valve determines the shear forces that act upon the cells and the number of passes determines the length of time that the cells are subjected to these forces. The effect of the step on the cells is to rupture them releasing intracellular product. Equation A.31, which describes the relationship between the operation conditions and the amount of product released, was developed by Hetherington *et al* (1971): -

$$r = 1 - e^{-K_p P^\alpha N_{Pass}} \quad \text{Equation A.31}$$

where P is the pressure in the homogeniser (bar) and N_{Pass} is the number of passes.

The fraction released can be used to calculate the mass balance on each component over the homogenisation step based on conservation laws.

$$M_{T.hom.(protein,ADH,Nucleic,Debris)} = M_{T.sus.(protein,ADH,Nucleic,Debris)} \quad \text{Equation A.32}$$

$$M_{T.hom.(DCW)} = (1 - r) \cdot M_{T.sus.(DCW)} \quad \text{Equation A.33}$$

$$M_{I.hom.(DCW,protein,ADH,Nucleic,Debris)} = (1 - r) \cdot M_{I.sus.(DCW,protein,ADH,Nucleic,Debris)} \quad \text{Equation A.34}$$

$$M_{I.hom.(Debris)} = M_{I.sus.(Debris)} \quad \text{Equation A.35}$$

where $M_{T.stream.(components)}$ is the total amount of the named components in the stream (g), $M_{I.stream.(components)}$ is the amount of the named intracellular components in the

stream (g). The subscripts *hom* and *sus* refer to the homogenate stream and the cell suspension stream respectively.

A.4.3 Solid Properties

In addition to the changes in the mass balance a number of changes occur in the physical properties of the streams. Probably the most critical ones are the changes in the solids properties. As the cell wall breaks open small debris fragments are created. Using experimental data the following model was created to correlate the average particle size and particle size standard deviation to the number of passes and homogeniser pressure respectively (Siddiqi *et al*, 1996).

$$\bar{\lambda}_p = \left(1 - e^{-\frac{K_d}{N_{Pass}^{0.4} P}} \right) \cdot \bar{\lambda}_{p,N=0} \quad \text{Equation A.36}$$

$$\sigma_\lambda = \begin{cases} \left(1 + 2.3e^{-\frac{K_d}{N_{Pass}^{0.4} P}} \right) \sigma_{\lambda,N=0} & \text{if } e^{-\frac{K_d}{N^{0.4} P}} < 0.33 \\ \left(3.4 - 5.5e^{-\frac{K_d}{N_{Pass}^{0.4} P}} \right) \sigma_{\lambda,N=0} & \text{if } e^{-\frac{K_d}{N^{0.4} P}} \geq 0.33 \end{cases} \quad \text{Equation A.37}$$

The cell rupture also results in changes to the density of the solid phase. The debris density was assumed to be 1050kg/m³ based on laboratory measurements. The average solids density was calculated by interpolating the density of the cells and debris based on the amount of cells ruptured (Equation A.38). Equally the concentration of solids is affected by the amount of cell rupture. The concentration of solids after homogenisation is the wet cell weight of the remaining cells plus the total free cell debris per unit volume (Equation A.39).

$$\rho_s = (1 - r) \cdot \rho_{Cells} + r \cdot \rho_{debris} \quad \text{Equation A.38}$$

$$C_S = (3 \cdot M_{T, hom. (DCW)} + r \cdot M_{T, hom. (Debris)}) / V_{hom} \quad \text{Equation A.39}$$

where ρ_s is the average density of the solid phase (g/L), ρ_{cells} is the density of cells (g/L) and ρ_{debris} is the density of cell debris (g/L).

A.4.4 Liquid Properties

The release of intracellular products increases both the viscosity and the density of the liquid. Equation A.41 relates the release of intracellular components to the viscosity of homogenate supernatant based on data generated in earlier experimental work (Mosquiera *et al.*, 1981). This work was carried out using a cell concentration of $450\text{g}\cdot\text{L}^{-1}$ and indicated that there is a linear relationship between viscosity and the level of cell rupture.

$$\eta_{hom} = \eta_{sus} + (4.911 \times 10^{-7}) \cdot X \cdot r \quad \text{Equation A.40}$$

where η_{stream} is the viscosity of the specified stream (Ns/m) and X is the concentration of cells (g/L).

The rupture process also serves to increase the density of the surrounding solution by virtue of the released intracellular material having a higher density. The change in density of the homogenate is calculated from the volume of ruptured cells and the density difference between the cells and the surrounding liquor (Equation A.41).

$$\rho_{L,hom} = \rho_{L,susp} + (\rho_{cells} - \rho_{L,sus}) \left(\frac{X \cdot r}{\rho_{L,sus}} \right) \quad \text{Equation A.41}$$

where $\rho_{L,stream}$ is the density of specified stream (g/L) and ρ_{cells} is the density of cells (g/L).

A.4.5 Time Taken

The time taken by the homogenisation step is also dependent on the pressure and the number of passes. Reducing the size of the valve orifice increases the pressure drop across the orifice but decreases the flowrate through the orifice. For the homogeniser

used in these studies there is the following experimentally determined relationship between flowrate and pressure.

$$Q = 94.4 - 0.0384P \quad \text{Equation A.42}$$

This can in turn be used to calculate the time taken in the homogenisation and in turn the amount of product degradation.

$$t = \frac{N_{Pass} \cdot V_{feed}}{Q} \quad \text{Equation A.43}$$

A.5 Dilution

A.5.1 Introduction

The process also has two dilution steps, before and after homogenisation. Before homogenisation a buffer is added to the cells collected by the centrifuge to obtain a cell concentration of 450g/L. The amount of buffer added in this dilution step is determined by the amount needed to achieve this concentration, as the homogenisation model is based on data obtained at this concentration. The second dilution step is after the homogenisation step. Adding buffer reduces the viscosity, the liquid density and the concentration of solids, which improves the performance of the subsequent centrifuge process. However use of too much buffer solution results in longer processing times due to the higher batch volumes and therefore loss of product via degradation.

A.5.2 Liquid Properties

The main change is in the liquid properties. Equations A.44 & A.45 describe the changes in viscosity and liquid density. The viscosity is assumed to approach asymptotically the viscosity of the buffer. The density is given by linear interpolation and the change in the volume of liquid.

$$\eta_{Dil} = \eta_{Buf} \left(\frac{\eta_{Conc}}{\eta_{Buf}} \right)^{V_{Conc} / (V_{Conc} + V_{Buf})} \quad \text{Equation A.44}$$

$$\rho_{L,Dil} = \frac{\rho_{L,Conc} V_{Conc} + \rho_{L,Buf} V_{Buf}}{V_{Conc} + V_{Buf}}$$

Equation A.45

where V_{Conc} is the volume into dilution step (L), V_{Buf} is the volume of buffer added (L), η_{Stream} is the viscosity of the specified stream (Ns/m) and $\rho_{L,stream}$ is the density of the specified stream (g/L). The subscripts *Conc*, *Dil* and *Buf* refer to the Concentrated Feed, Diluted Output and the Buffer Solution.

Appendix B: Hyper-Rectangles inside a Hyper-sphere

This section looks at the largest n-dimensions hyper-rectangle that will fit inside an n-dimensional hyper-sphere. The equation that defines an n-dimensional hyper-sphere is: -

$$R^2 = \sum_i^n s_i^2 \quad \text{Equation B.1}$$

The volume of an n-dimensional hyper-sphere is given by: -

$$V_S(n) = \frac{\pi^{n/2} R^n}{\Gamma(n/2 + 1)} \quad \text{Equation B.2}$$

where $V_S(n)$ is the volume of an n-dimensional sphere, R is the radius of the Sphere and $\Gamma(x)$ is the Gamma function.

The gamma function can be calculated using the following equation: -

$$\Gamma(x) = \int_0^\infty p^{x-1} e^{-p} dp \quad \text{Equation B.3}$$

Assuming the largest hyper-rectangle sits on the same origin as the hyper-sphere, which would seem logical due to the symmetry of the hyper-sphere, the largest hyper-rectangle inside an N-dimensional hyper-sphere will be given by: -

$$V_R(n) = 2^n \prod_{i=1}^n s_i \quad \text{Equation B.4}$$

where $V_R(n)$ is the volume of an n-dimensional hyper-rectangle and s_i is a vector of the distance from the axis which must lie inside Equation B.1.

In a two-dimensional case this problem becomes the largest rectangle inside a circle.

Therefore Equation B.4 can be re-written as: -

$$V_R(2) = 4s_1s_2 \quad \text{Equation B.5}$$

Inserting Equation B1, this becomes: -

$$V_R(2) = 4s_1\sqrt{(R^2 - s_1^2)} \quad \text{Equation B.6}$$

This can be solved to show that the largest rectangle is in fact a square with sides $R/\sqrt{2}$. It would seem sensible that given the largest rectangle in a circle is a square that the largest hyper-rectangle inside a sphere would be a hypercube. Solving this from first principles for higher dimensions is not a simple problem. Therefore to investigate this, a small Matlab program was written that tried to find the largest hyper-rectangle inside a hyper-sphere. The program assumed that such a hyper-rectangle would be centred on the same origin as the hyper-sphere therefore only looked for one vertex (as the opposite vertex assumed to be a reflection through the origin). The program then used a Nelder-Mead optimisation routine to find the largest internal hyper-rectangle as calculated by Equation B.4. This sought to optimise (N-1) of the variables that defined the vertex in N-dimensional space, with the remaining variable calculated using Equation B1.

The results of this program seemed to confirm that the hypercube would be the largest hyperrectangle inside the sphere. Therefore using Equation B.1 it can be shown that for a n-dimensional sphere the sides of the largest hyper-rectangle will equal $2R/\sqrt{n}$. Therefore the volume of the hyper-rectangle will be: -

$$V_R(n) = \left(\frac{2R}{\sqrt{n}}\right)^n \quad \text{Equation B.7}$$

In this work we were interested in the volume of the hyper-rectangle to the hypersphere. The volume of a hyper-sphere relative to its radius to the power n increases as the number of dimensions increase until the 5 dimensions. After this number of dimensions the ratio of the hyper-sphere to its radius to the power n diminishes with increasing dimensions. By contrast the size of the hyper-cube inside the hyper-sphere drops consistently as the number of dimensions increases. Both of these characteristics can be seen in figure B.1.

More interestingly figure B.2 shows the ratio the hyper-rectangle inside the hyper-sphere to the hyper-sphere itself. It can be seen that this ratio drops rapidly as the number of dimensions increases. This indicates that a hyper-rectangle inside a hyper-sphere will capture an increasingly small proportion of the volume as the number of dimensions increases.

Figure B.1: The volume of a Hyper-Sphere with a radius of 1 and the largest Hyper-Rectangle that can fit inside it for different numbers of dimensions.

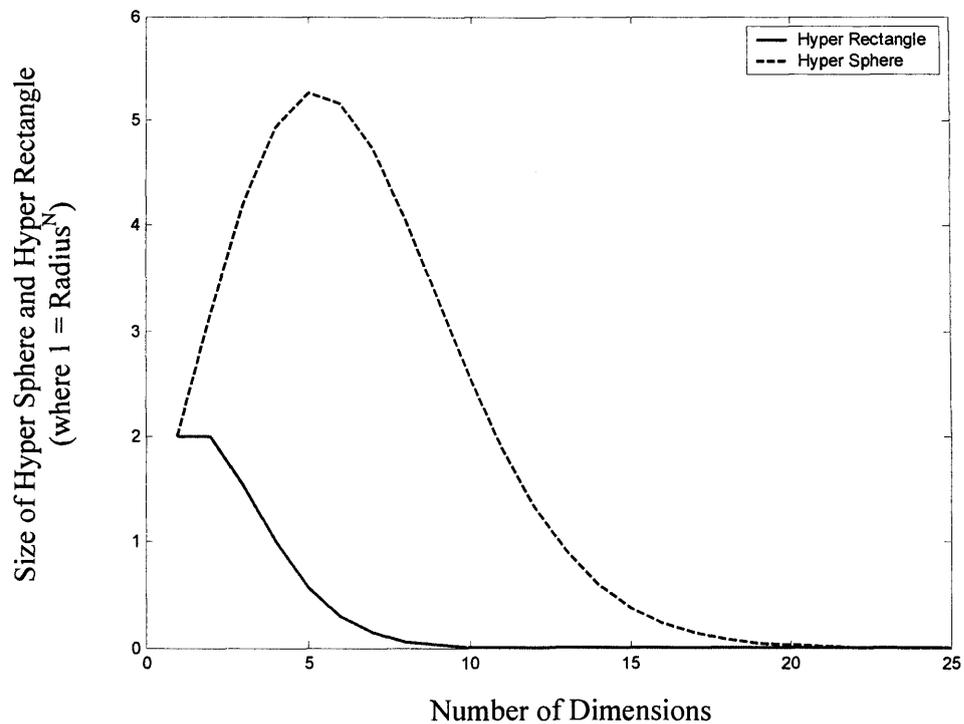
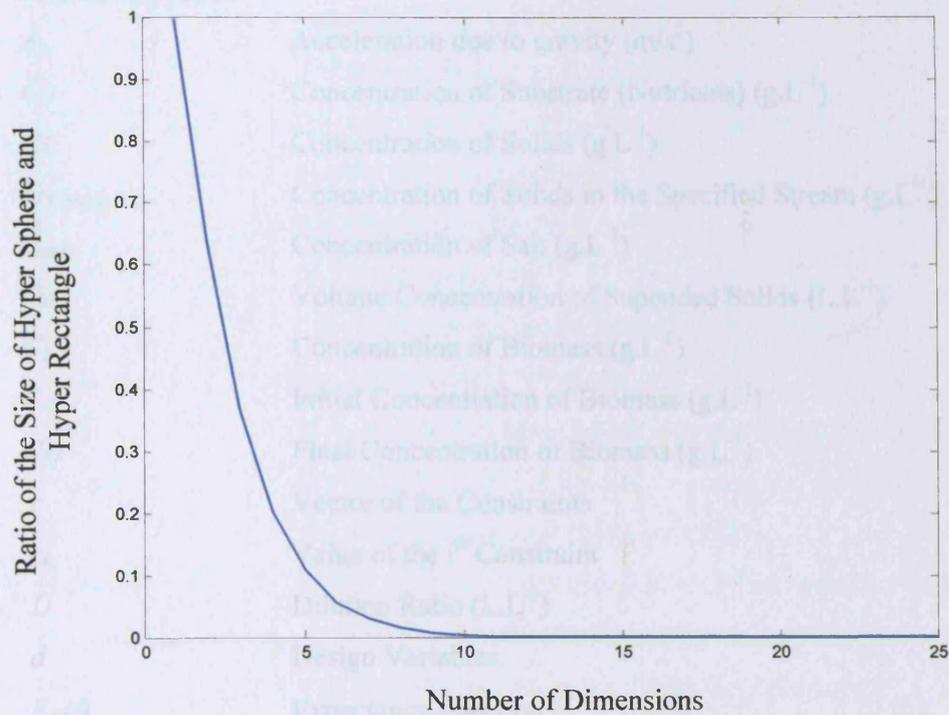


Figure B.2: The ratio of the size of a Hyper-Sphere with a radius of 1 to largest Hyper-Rectangle that can fit inside it for different numbers of dimensions.



Nomenclature

Roman Symbols

A_g	Acceleration due to gravity (m/s^2)
C_N	Concentration of Substrate (Nutrients) ($g.L^{-1}$)
C_S	Concentration of Solids ($g.L^{-1}$)
$C_{S,Stream}$	Concentration of Solids in the Specified Stream ($g.L^{-1}$)
C_{salt}	Concentration of Salt ($g.L^{-1}$)
C_V	Volume Concentration of Suspended Solids ($L.L^{-1}$)
C_X	Concentration of Biomass ($g.L^{-1}$)
C_{X0}	Initial Concentration of Biomass ($g.L^{-1}$)
C_{Xf}	Final Concentration of Biomass ($g.L^{-1}$)
c	Vector of the Constraints
c_i	Value of the i^{th} Constraint
D	Dilution Ratio ($L.L^{-1}$)
d	Design Variables
$E_{\phi(f)}$	Expectancy Function
$F_{ADH/Prot}$	Ratio Alcohol Dehydrogenase to Protein ($g.g^{-1}$)
F_{LD}	Fraction of Liquid to Discharge ($g.g^{-1}$)
F_{Precip}	Fraction of Component in Precipitant Phase ($g.g^{-1}$)
$F_{Prot/Biomass}$	Ratio of Dry Cell Weight to Protein ($g.g^{-1}$)
F_{SD}	Fraction of Solids to Discharge
F_{SS}	Fraction of Solids to Sediment
f_{ADH}	Function for calculating the ADH
f_{obj}	The Objective Function
$f_{Penalty}$	Penalty Function
$f_{P(Debris)}$	Probability of the debris constraint being broken
$f_{P(i)}$	Probability of constraint i being broken
f_{flex}	Measure of Process Flexibility
f_1	Set of Differential Equations
f_2	Set of Algebraic Relationships
g	Set of Inequality Constraints
$g_{norm,i}$	Normalised Values of the i^{th} Constraint

h	Equations for calculating the State Variables
I	Set of Inequality Constraint
I_{FR}	Integral of the feasible region
I'_{FR}	Normalised Integral of the feasible region
i	Index for different Constraints
i_{max}	Total Number of Constraints
$j(\theta)$	Joint Probability Density Function for the Uncertain Parameters
k_p	Protein Release Coefficient
k_d	Cell Breakage Coefficient
k_{deg}	Degradation Rate (h^{-1})
k_{GE}	Coefficient in the Grade Efficiency Model (0.865)
M_{ADH}	Mass of Alcohol Dehydrogenase (g)
M_{ADH_deg}	Mass of Alcohol Dehydrogenase after Degradation (units)
$M_{aqueous}$	Vector of the mass of aqueous components (units)
M_I	Vector of intracellular components (g)
$M_{I.stream}$	Vector of intracellular components in the specified stream (g)
$M_{I.stream.components}$	Named intracellular components in the specified stream (g)
M_P	Vector of the mass of precipitant components (g)
$M_{P.stream}$	Vector of the precipitant components in the specified stream (g)
$M_{P.stream.components}$	Named precipitant components in the specified stream (g)
$M_{Solids.Stream}$	Total mass of solids in the specified stream (g)
M_T	Vector of the total mass of each component (g)
$M_{T.stream}$	Vector of the total components in the specified stream (g)
$M_{T.stream.components}$	Named components in the specified stream (g)
m	Number of Sample points in a Monte Carlo simulation
N_{Dis}	Number of Discharges
N_{Pass}	Number of Passes
n	Number of Control Variables or Dimensions
n_v	Number of Control Variables subject to uncertainty
P	Pressure in the Homogeniser (bar)
Q	Flowrate (L/hr)
Q'	Flowrate (m^3/sec)
$Q_{substrate}$	Feed Rate into the Fermenter (L/hr)

R	Radius of a hypersphere
R_{flex}	Flexible Region
R_{FR}	Feasible Region
R_{OFR}	The Robust Feasible Region given Control Variations
R_{RFR}	The Robust Feasible Region given Parameter Variations
r	Fraction of Protein Released
s_i	Variables used to define a hypersphere
T	Hyper-Rectangle Defining a Series of Parameter Ranges
T_{Flex}	Hyper-Rectangle in the Flexible Region
T_{FR}	Hyper-Rectangle in the Feasible Region
t	Time (hr)
u	Algebraic variables
v	The number of control variables
V	Volume (L)
V_0	Initial Volume in the fermenter (L)
V_C	Total Volume of Centrifuge Bowl (L)
V_f	Final Volume in the Fermenter (L)
V_{Stream}	Volume in the Specified Stream (L)
V_{SC}	Maximum Volume of Solids in Centrifuge (L)
w	Weight variable in the objective function
x	State Variables
\dot{x}	Derivative of the State Variables
$Y_{x/s}$	Yield of Biomass on Substrate ($\text{g}\cdot\text{g}^{-1}$)
y	Process Performance
\bar{y}	Expected or Average Performance
y_i	Performance of the i^{th} Constrained Variable
Z	The operating Space (or control variable space)
z	Control Variables
z_{var}	The range of Values of around a set point
$z_{v,min}$	Minimum value for control variable v .
$z_{v,max}$	Maximum value for control variable v .

Greek Symbols

α	Pressure Exponent in the Hetherington Equation
β	Particle Geometric Factor
Γ	The Gamma Function
γ	Probability for a particular constraint
δ	Scaled Parameter Deviations
ε	Respiratory Quotient
η	Viscosity (Ns/m)
η_{stream}	Viscosity in a Specified Stream (Ns/m)
θ	Parameter Values
θ_N	Nominal Parameter Values
θ_L	Lower Limit for Parameter Values
θ_s	Specified Parameter Value
θ_{var}	Range of Possible Parameter Values
θ_U	Upper Limit for Parameter Values
$\Delta\theta$	Vector of Parameter Deviations
λ_c	Critical Diameter (m)
$\bar{\lambda}_{cell}$	Average Cell Diameter (μm)
λ_p	Particle Diameter (μm)
$\bar{\lambda}_p$	Average Particle Diameter (μm)
μ	Growth Rate (h^{-1})
Π	Production function
ρ_{cells}	Density of cells (g.L^{-1})
ρ_{debris}	Density of Cell Debris (g.L^{-1})
ρ_L	Density of the Liquid Phase (g.L^{-1})
$\rho_{L.stream}$	Density of the Liquid Phase in the Specified Stream (g.L^{-1})
ρ_S	Density of the Solid Phase (g.L^{-1})
Σ	Centrifuge Equivalent Settling Area (m^2)
σ_{yi}	Standard Deviation in a Performance Variable
σ_λ	Standard Deviation in Particle Size (μm)
$\tau(\lambda)$	Proportion of Solids of a Size Collected in the Sediment

Φ	Standard Normal Distribution Function
Φ^{-1}	Inverse Standard Normal Distribution Function
$\varphi(\lambda)$	Proportion of particles with a particular size
Ψ	Feasibility of a Set Point
Ψ_O	Robustness of a Set Point given uncertainty in control variables
Ψ_R	Robustness of a Set Point given Uncertainty in parameters
Ω_O	Hyper-Rectangle Surrounding defined by control variables
Ω_R	Hyper-Rectangle Surrounding defined by parameters

Logical Symbols

\exists	Existential quantification
\forall	Universal quantification

References

Acklam P. J., <http://home.online.no/~pjacklam/notes/invnorm/>, (2000)

Arlington S., Hughes S., McKiernan J., Hamilton J., Palo J. and Solano P. Pharma 2005 An Industrial Revolution in R & D. PriceWaterhouse Coopers (1998).

Bailey J.E. and Ollis D.F., *Biochemical Engineering Fundamentals*, McGraw Hill, 2002.

Baird R. and De Santis P., *Validation of Biopharmaceutical Facilities*. In *Bioprocess Engineering: Systems, Equipment and Facilities*. (Ed. D'Elia N.A. and Nelson K.L.) pp. 747-781, Wiley, (1994).

Barton P.I. and Pantelides C.C., *Modeling of Combined Discrete-Continuous Processes*. *AIChE Journal* 40, 966-979 (1994).

Bernardo F.P., Pistikopoulos E.N. and Saraiva P.M., *Quality Costs and Robustness Criteria in Chemical Process Design Optimization*. *Computers and Chemical Engineering* 25, 27-40 (2001).

Bernardo F.P., Saraiva P., *Robust Optimization Framework for Process Parameter and Tolerance Design*, *AIChE Journal* 44, 2007-2017 (1998).

Biegler L.T. and Hughes R.R., *Infeasible Path Optimisation with Sequential Modular Simulators*. *AIChE Journal* 28, 994-1002 (1982).

Biegler L.T., *Chemical Process Simulation*. *Chemical Engineering Progress* 50-61 (1989).

Bonnerjea J., Ob S., Hoare M. and Dunnill P., *Protein Purification: The Right Step at the Right Time*. *Bio/Technology* 4, 954-958 (1986).

Boychyn M., Doyle W., Bulmer M., More J. and Hoare M., *Laboratory Scaledown of Protein Purification Processes Involving Fractional Precipitation and Centrifugal Recovery*. *Biotechnology and Bioengineering* 69, 1-9 (2000).

Boychyn M., Yim S.S., Ayazi-Shamlou P., Bulmer M., More J. and Hoare M., *Characterization of flow intensity in continuous centrifuges for the development of laboratory mimics*. *Chemical Engineering Science* 56, 4759-4770 (2001).

Bradley P. and Fox B.L., *Implementing Sobol Quasirandom Sequence generator*, *ACM Transactions on Mathematical Software*, 14, 88-100 (1988).

Bulmer M., Clarkson A.I., Titchener-Hooker N.J. and Dunnill P., *Computer-based simulation of the recovery of intracellular enzymes and its pilot-scale verification*. *Bioprocess Engineering* 15, 331-337 (1996).

Byrom D., *Role and Timing of Process Development for Biopharmaceutical Manufacture*. *Pharmaceutical Technology Europe* 12, 52-56 (2000).

Cardosa M.F., Salcedo R.L., DeAzevedo S.F., *The Simplex-Simulated Annealing approach to Continuous non-Linear Optimization*, *Computers and Chemical Engineering* 20, 1065-1080 (1996).

Cardosa M.F., Salcedo R.L., DeAzevedo S.F., Barbosa P, *A Simulated Annealing Approach to the Solution of a MINLP Problem*, *Computers and Chemical Engineering* 21, 1349-1364 (1997).

Chen B.H., Doig S.D., Lye G.J. and Woodley J.M., *Modelling of the Baeyer-Villiger Monooxygenase Catalysed Synthesis of Optically Pure Lactones*. *Trans IChemE Part C* 1-51 (2002).

Clarkson A.I., Lefevre P. and Titchener-Hooker N.J., *A Study of Process Interactions between Cell Disruption and Debris Clarification Stages in the Recovery of Yeast Intracellular Products*. *Biotechnology Progress* 3, 462-467 (1993).

Clarkson A.I. A Study of the Use of Process Simulation and Pilot-Scale Verification Trials for the Design of Bioprocesses. 1994. PhD Thesis, University of London, U.K. (1996)

Clarkson A.I., Bulmer M. and Titchener-Hooker N.J., *Pilot-scale verification of a computer-based simulation for fraction protein precipitation*. *Bioprocess Engineering* 14, 69-80 (1996).

Clarkson A.I., Bulmer M. and Titchener-Hooker N.J., *Pilot-scale verification of a computer-based simulation for the centrifugal recovery of biological particles*. *Bioprocess Engineering* 14, 81-89 (1996).

Datar R., *Economics of Primary Separation Steps in relation to fermentation and genetic engineering*. *Process Biochemistry* 19-26 (1986).

Datar R., Cartwright T. and Rosen C.G., *Process Economics of Animal Cell and Bacterial Fermentations: A Case Study Analysis of Tissue Plasminogen Activator*. *Bio/Technology* 11, 349-357 (1993).

Dvorin J., *Large Molecules: Too Late For Big Pharma?* *In Vivo: The Business and Medicine Report* 53-58 (2001).

Edgar T.F., Himmelblau D.M. and Lasdon L.S., *Optimisation of Chemical Processes*, McGraw Hill, (2002).

Engelking L.R. Chapter 1: Chemical Composition of Living Cells, *Textbook of Physiological Chemistry*, TetonNewMedia, (2004)

Evans L.B. and Field R.P., *Bioprocess Simulation: A new tool for Process Development*. *Bio/Technology* 6, 200-203 (1988).

Fahrner R.L., Iver H.V. and Blank G.S., *The optimal flow rate and column length for maximum production rate of protein A affinity chromatography*. *Bioprocess Engineering* 21, 287-292 (1999).

Farid S., Washbrook J., Birch J. and Titchener-Hooker N.J., *A Hierarchical Framework for Modelling Biopharmaceutical Manufacturing to Address Process and Business Needs*. Computers and Chemical Engineering 8, 673-678 (2000).

Farid S., Novais J.L., Karri S., Washbrook J. and Titchener-Hooker N.J., *A Tool for Modeling Strategic Decisions in Cell Culture Manufacturing*. Biotechnology Progress 16, 829-836 (2000).

Farid S. A Decision-Support Tool for Simulating the Process and Business perspectives of Biopharmaceutical Manufacture. PhD Thesis, University of London, UK. (2002)

Foo F., Karri S., Davis E., Titchener-Hooker N.J. and Dunnill P., *Biopharmaceutical Process Development: Part I, Information from the First Product Generation*. Biopharm Europe (2001).

Gandikota M.S., Davis J.F., Yang S.T. and Marchio J., *Bioprocess flowsheets made easy*. Chemical Technology 22, 694-699 (1992).

Gregory M., Keay P.J., Dean P., Bulmer M. and Thornhill N.F., *A Visual Programming Environment for Bioprocess Control*. Journal of Biotechnology 33, 223-241 (1994).

Gregory M., Bulmer M., Bogle I.D.L. and Titchener-Hooker N.J., *Optimising enzyme production by bakers yeast in continuous culture: physiological knowledge useful for process design and control*. Bioprocess Engineering 15, 239-245 (1996).

Gritsis D. and Titchener-Hooker N.J., *Biochemical Process Simulation*. IChemE Symposium Series 114, 69-77 (1989).

Grossmann I.E. and Sargent R.W.H., *Optimum Design of Chemical Plants with Uncertain Parameters*. AIChE Journal 24, 1021-1028 (1978).

Halemane K.P. and Grossmann I.E., *Optimal Process Design under Uncertainty*. AIChE Journal 29, 425-433 (2002).

Hearle D.C., Aguilera-Soriano G., Wiksell E. and Titchener-Hooker N.J., *Quantifying the Fouling Effects of a Biological Process Stream on Chromatographic Supports*. IChemE Symposium Series 174-176 (1994).

Hetherington P.J., Follows M., Dunnill P. and Lilly M.D., *Release of Protein from Baker's Yeast (Sacchromyces Cerevisiae) by disruption in an industrial homogeniser*. Trans IChemE 49, 142-148 (1971).

Jungbauer A. and Kaltenbrunner O., *Chromatography: Computer Aided Design and Optimisation*. In *Encyclopedia of Bioprocess Technology: Fermentation, Biocatalysis and Bioseparation*. (Ed. Flickinger M. and Drew S.) pp. 585-601, Wiley, New York (1999).

Karri S., Davis E., Titchener-Hooker N.J. and Washbrook J., *Biopharmaceutical Process Development: Part III, A Framework to Assist Decision Making*. Biopharm Europe 76-89 (2001).

Kelley B.D. and Hatton A., *The Fermentation / Downstream Processing Interface*. Bioseparation 1, 333-349 (1991).

Kleinig A.R. and Middleberg A.P.J., *The Correlations of Cell Disruption with Homogeniser Value Pressure Gradient Determined by Computational Fluid Dynamics*. Chemical Engineering Science 51, 5103-5110 (1996).

Kreiger J.H., *Process Simulation Seen As Pivotal In Corporate Information Flow*. Chemical and Engineering News (1995).

Labview Starter Guide, Texas Instruments (1999)

Mannweiler K. and Hoare M., *The Scale-down of an industrial disc stack centrifuge*. Bioprocess and Biosystems Engineering 8, 19-25 (1992).

Marsaglia G., *Evaluating the Normal Distribution*, Journal of Statistical Software 11, (2004).

Metropolis N., Rosenbluth A., Rosenbluth M., Teller A. and Teller E., *Perspective on "Equation of State calculations by fast Computing Machines"*, Journal of Chemical Physics 21, 1087-1092 (1953).

Maybury J.P., Hoare M. and Dunnill P., *The Use of Laboratory Centrifugation Studies to Predict Performance of Industrial Machines: Studies of Shear-Insensitive and Shear-Sensitive Materials*. Biotechnology and Bioengineering 67, 265-273 (2000).

Matlab User Guide, Mathworks Limited (2000)

Middelberg A.P.J., O'Neill B.K. and Bogle I.D.L., *Modelling Bioprocess Interactions for optimal design and operating strategies*. Trans IChemE Part C 70, 8-12 (1992).

Middelberg A.P.J., O'Neill B.K., Bogle I.D.L., Gully N.J., Rogers A.H. and Thomas C.R., *A new model for the disruption of Escherichia Coli by high Pressure Homogenisation*. Trans IChemE Part C 70, 213-218 (1992)

Mosqueira F.G., Higgins J.J., Dunnill P. and Lilly M.D., *Characteristics of Mechanically Disrupted Bakers' Yeast in Relation to its Separation in Industrial Centrifuges*. Biotechnology and Bioengineering 23, 335-343 (1981)

Naess L., Mjåavatten A. and Li J., *Using Dynamic Process Simulations from Conception to Normal Operation of Process Plants*. Computers and Chemical Engineering 17, 585-600 (1993)

Narodoslawsky M., *Bioprocess Simulation: A Systems Theoretical Approach to Biotechnology*. Chemical and Biochemical Engineering Quarterly 5, 183-187 (1991)

-
- Nelder J.A., Mead R., *A Simplex Method for Function Minimisation*, The Computer Journal 7, 308-313 (1965)
- Ngiam S.H., Zhou Y., Turner M.K. and Titchener-Hooker N.J., *Graphical Method for the calculation of chromatographic performance in representing the trade-off between purity and recovery*. Journal of Chromatography A 937, 1-11 (2001)
- Oolman and Liu, *Filtration Properties of Mycelial Microbial Broths*, Biotechnology Progress 7, 534-539 (1991)
- Pantelides C.C., *Speedup - Recent advances in process simulation*. Computers and Chemical Engineering 12, 745-755 (1988)
- Pantelides C.C. and Barton P.I., *Equation-Oriented Dynamic Simulation: Current Status and Future Perspectives*. Computers and Chemical Engineering 17, 263-285 (1993)
- Pascal F., Dagot C., Pingaud H., Corriou J.P., Pons M.N., Engasser J.M., *Modelling of an industrial alcohol fermentation and simulation of the plant by a process simulator*, Biotechnology and Bioengineering 43, 202-217 (1995)
- Pate M. E., Turner M.K., Thornhill N.F. and Titchener-Hooker N.J., *The Use of Principle Component Analysis for the modelling of high performance liquid chromatography*. Bioprocess Engineering 21, 261-272 (1999)
- Petrides D.P., Cooney C.L. and Evans L.B., *Bioprocess Simulation: An Integrated Approach to Process Development*. Computers and Chemical Engineering 13, 553-561 (1989)
- Petrides D.P., *BioPro Designer: An Advanced Computing Environment for Modeling and Design of Integrated Biochemical Processes*. Computers and Chemical Engineering 18, 621-625 (1994)

Petrides D.P., Sepidou E. and Calandranis J., *Computer Aided Process Analysis and Economic Evaluation of Biosynthetic Human Insulin Production - A Case Study*. *Biotechnology and Bioengineering* 48, 529-541 (1995)

Petrides D.P., Calandranis J. and Cooney C.L., *Bioprocess Optimization via CAPD and Simulation for Product Commercialization*. *Genetic Engineering News* (1996)

Petrides D.P., Nir R., Calandranis J. and Cooney C.L., *Introduction to Bioprocess Simulation*. In *Manual of Industrial Microbiology and Biotechnology*. (Ed. Demain A.L. and Davies J.E.) pp. 289-299, 1999

Petrides D.P., Koulouris A. and Siletti C.A., *Throughput Analysis and Debottlenecking of Biomanufacturing Facilities*. *Biopharm* 2-7 (2002)

Pisano G.P. and Wheelwright S.C., *The New Logic of High-Tech R & D*. *Harvard Business Review* 73, 93-105 (1995)

Press W., Teukolsky S., Vetterling W.T., Flannery B.P., *Numerical Recipes in C++*, Cambridge University Press (2002)

Reismann H.B., *Economics*. In *Manual of Industrial Microbiology and Biotechnology*. (Ed. Demain A.L. and Davies J.E.) pp. 273-288, (2002).

Richardson P. and Zaki W.N., *Sedimentation and Fluidisation: Part I*. *Trans IChemE* 32, 35-53 (1954).

Sahinidis N.V., *Optimisation under uncertainty: state of the art and opportunities*, *Computers and Chemical Engineering* 28, 971-983 (2004).

Samsatli N.J., Papageorgiou L.G. and Shah N., *Batch Process Design and Operation using Operational Envelopes*. *Computers and Chemical Engineering* 23, 887-890 (1999).

Samsatli N.J., Sharif M., Shah N. and Papageorgiou L.G., *Operational Envelopes for Batch Processes*. *AIChE Journal* 47, 2277-2288 (2001).

Sargent R.W.H., *Integrated Design and Optimisation of Processes*, *Chemical Engineering Progress* 63 (9), 71-78, (1967).

Shanklin T., Roper K., Yegneswaran P.K. and Marten M.R., *Selection of Bioprocess Simulation Software for Industrial Applications*. *Biotechnology and Bioengineering* 72, 483-489 (2001).

Siddiqi S.F., Bulmer M., Ayazi-Shamlou P. and Titchener-Hooker N.J., *The effects of fermentation conditions on yeast cell debris particle size distribution during high pressure homogenisation*. *Bioprocess Engineering* 14, 1-8 (1995).

Siddiqi S.F., Titchener-Hooker N.J. and Ayazi-Shamlou P., *Simulation of particle size distribution changes occurring during high-pressure disruption of bakers' yeast*. *Biotechnology and Bioengineering* 50, 145-150 (1996).

Siddiqi S.F. *Process Simulation and Optimisation of High Pressure Disruption for the Release of Intracellular Protein*. PhD Thesis, University of London, U.K. (1996)

Siddiqi S.F., Titchener-Hooker N.J. and Ayazi-Shamlou P., *High Pressure Disruption of Yeast Cells: The Use of Scale Down Operations for the Prediction of Protein Release and Cell Debris Size Distribution*. *Biotechnology and Bioengineering* 55, 642-649 (1997).

Siletti C.A. and Stephanopoulos G., *BioSep Designer: A Process Synthesizer for Bioseparations*. In *Artificial Intelligence in Engineering Design*. pp. 295-316, (1992).

Smith M. P., *An Evaluation of Expanded Bed Adsorption for the recovery of proteins from crude feedstocks*. PhD Thesis, University of London, U.K. (1997)

Steffans M.A., Fraga E.S. and Bogle I.D.L., *Multicriteria process synthesis for generating sustainable and economic bioprocesses*. Computers and Chemical Engineering 23, 1455-1467 (1999).

Swaney R.E. and Grossmann I.E., *An Index for Operational Flexibility in Chemical Process Design. Part 2*. Computers and Chemical Engineering 31, 631-641 (1985).

Swaney R.E. and Grossmann I.E., *An Index for Operational Flexibility in Chemical Process Design*. AIChE Journal 31, 621-630 (2002).

Titchener-Hooker N.J., Zhou Y., Hoare M. and Dunnill P., *Biopharmaceutical Process Development: Part II, Methods of Reducing Development Time*. Biopharm Europe (2001).

Research and Development - The Key to Innovation. In Pharmaceutical Industry Profile 2002. PhRMA (Ed. Unknown) pp. 12-27, (2002)

Uesbeck F., Samsatli N.J., Papageorgiou L.G., Shah N., *Robust Optimal Fermentation Operating Policies*, Computers and Chemical Engineering 22 (Suppl.), 167-173 (1998).

Varga E.G., Titchener-Hooker N.J. and Dunnill P., *Using scale-down methods to rapidly apply natural yeast homogenisation models to a recombinant strain*. Bioprocess Engineering 19, 373-380 (1997).

Varga E.G. Modelling of Downstream Processing of a Recombinant Intracellular Enzyme. PhD Thesis, University of London, U.K. (1997)

Wang H.Y., Cooney C.L. and Wang D.I.C., *Computer Control of Baker's Yeast Production*. Biotechnology and Bioengineering 21, 975-995 (1979).

Werner R.G., Langlouis-Gau H., Walz F. and Hoffmann H., *Validation of Biotechnological Production Processes*. Arzneimittel - Forsch/ Drug Res 38, 855-862 (1988).

Werner R.G. and Langlouis-Gau H., *Meeting the Regulatory Requirements for Pharmaceutical Production of Recombinant DNA Derived Products*. *Arzneim - Forsch/ Drug Res* 39, 108-111 (1989).

Werner R.G., *Validation and Quality Assurance: Risk and Potential of Process Changes*, *Chimia* 38, 460-466 (1994).

Werner R.G., *Identification and Development of New Biopharmaceuticals*. *Arzneim - Forsch/ Drug Res* 48, 523-530 (1998).

Werner R.G., *Innovative and Economic Potential of Mammalian Cell Culture*. *Arzneim - Forsch/ Drug Res* 48, 423-426 (1998).

Westerberg A.W., *Synthesis in Engineering Design*. *Computers and Chemical Engineering* 13, 365-376 (1989).

Wheelwright S.M., *Designing Downstream Processes for Large Scale Protein Purification*. *Bio/Technology* 5, 789-793 (1991).

Wong H.H., O'Neill B.K. and Middelberg A.P.J., *A mathematical model for Escherichia Coli debris size reduction during high pressure homogenisation based on grinding theory*. *Chemical Engineering Science* 52, 2883-2890 (1997).

Woodley J.M. and Titchener-Hooker N.J., *The Use of Windows of Operation as a Bioprocess Design Tool*. *Biotechnology and Bioengineering* 14, 263-268 (1996).

Zhou Y., Holwill I.L.J. and Titchener-Hooker N.J., *A Study of the use of computer simulations for the design of integrated downstream processes*. *Bioprocess Engineering* 16, 367-374 (1997).

Zhou Y. and Titchener-Hooker N.J., *Visualizing Integrated Bioprocess Designs Through "Windows of Operation"*. *Biotechnology and Bioengineering* 65, 550-557 (1999).

UNIVERSITY OF LONDON

SENATE HOUSE. MALET STREET, LONDON, WC1E 7HU



REPRODUCTION OF THESES

A thesis which is accepted by the University for the award of a Research Degree is placed in the Library of the College and in the University of London Library. The copyright of the thesis is retained by the author.

As you are about to submit a thesis for a Research Degree, you are required to sign the declaration below. This declaration is separate from any which may be made under arrangements with the College at which you have *pursued* your course (for internal candidates only). The declaration will be destroyed if your thesis is not approved by the examiners, being either rejected or referred for revision.

Academic Registrar

To be completed by the candidate

NAME IN FULL (Block Capitals) PAUL ANDREW GRIFFITHS

TITLE OF THESIS DEVELOPING METHODOLOGIES FOR
DETERMINING OPERATING STRATEGIES FOR BIOPROCESSES

DEGREE FOR WHICH THESIS IS PRESENTED PHD

DATE OF AWARD OF DEGREE (To be completed by the University) 31 DEC 2006

DECLARATION

1. I authorise that the thesis presented by me in *[] for examination for the MPhil/PhD Degree of the University of London shall, if a degree is awarded, be deposited in the library of the appropriate College and in the University of London Library and that, subject to the conditions set out below, my thesis be made available for public reference, inter-library loan and copying.
2. I authorise the College or University authorities as appropriate to supply a copy of the abstract of my thesis for inclusion in any published list of theses offered for higher degrees in British universities or in any supplement thereto, or for consultation in any central file of abstracts of such theses.
3. I authorise the College and the University of London Libraries, or their designated agents, to make a microform or digital copy of my thesis for the purposes of inter-library loan and the supply of copies.
4. I understand that before my thesis is made available for public reference, inter-library loan and copying, the following statement will have been included at the beginning of my thesis: The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.
5. I authorise the College and/or the University of London to make a microform or digital copy of my thesis in due course as the archival copy for permanent retention in substitution for the original copy.
6. I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.
7. I understand that in the event of my thesis being not approved by the examiners, this declaration would become void.

*Please state year.

DATE 116106 SIGNATURE Paul Griffiths

Note: The University's Ordinances make provision for restriction of access to an MPhil/PhD thesis and/or the abstract but only in certain specified circumstances and for a maximum period of two years. If you wish to apply for such restriction, please enquire at your College about the conditions and procedures. External Students should enquire at the Research Degree Examinations Office, Room NBQ1 (North Block), Senate House.

**THIS DECLARATION MUST BE COMPLETED AND RETURNED WITH THE
MAIN EXAMINATION ENTRY FORM**