Dynamics & Control
of a
Pressure Swing Adsorption Process
by
Ian Robert Zone

A dissertation submitted for the degree of
Doctor of Philosophy
in the
University of Surrey
Guildford, Surrey, England, GU2 5XH

September 1998
Abstract

Pressure Swing Adsorption (PSA) is an important and widely used industrial method for the separation and purification of gases. It is used in air separation, hydrogen recovery, and many other applications where preferential retention of one or more of the gases in the mixture is possible.

Despite the widespread use of PSA there seems to be a noticeable lack of published literature, with regard to the process performance, for either the effect of such aspects as dead volumes, or the dynamics & control of the process in order to enhance performance and reject disturbances.

A simulation robust enough to be used as part of a real-time control system was written, based on the instantaneous local equilibrium (ILE) model. The simulation includes the process equipment in addition to the adsorbent beds, such as dead space, valves, and vessels. Each bed was modelled independently, allowing the possibility of asymmetric operation. Data from a pilot plant, containing a zeolite 13X provided the experimental data to tune the theoretical model.

The effect of variations in key process parameters on the results demonstrated that the process behaviour could be significantly different to that predicted by modelling a single bed and assuming symmetry for the other beds in the plant, and could have an effect on the cyclic steady states and hence the dynamic responses to disturbances.

The model was proven to be robust enough to allow continuous manipulation of the adsorption model parameters within the beds to match the predicted output to that of the plant. Product purity control of the model, based on the predicted bed internal concentration fronts, was demonstrated to be capable of rejecting disturbances successfully. Non-linear responses were often shown in the modelling experiments.
Preface

All work described in this dissertation was carried out in the Department of Chemical and Process Engineering, at the University of Surrey, Guildford, between October 1992 and September 1995. It is the original and independent work of the author, except where otherwise stated in the text. No part of this thesis has been submitted for a degree at any other university.

I wish to express my thanks to Dr. N.F. Kirkby, my supervisor for this work. His advice, guidance, and enthusiasm throughout the period of this project proved invaluable, on many occasion showing me the light where I could only see darkness!

I would also like to thank both Ian Wilkinson and Simon Pool who helped build the PSA pilot plant originally and helped me on so many occasions with the modifications and repairs that I required to complete this project.

Finally, I would like to express my gratitude to my parents and to Sarah Sheppard, who have unfailingly supported me for so long, both during the project and throughout the preparation of this dissertation.

Ian Robert Zone
September 1998
# Contents

## Abstract

### Preface

### Contents

### Nomenclature

### Abbreviations

## Introduction

1.1 Introduction 1
1.2 PSA 1
1.3 Adsorbents 2
1.4 Cyclic PSA 3
    1.4.1 Performance Criteria 4
    1.4.2 PSA Process Design 5
    1.4.3 Steps And Cycles 6
    1.4.4 Important Operating Variables 9
1.5 The Project 9
    1.5.1 Introduction 9
    1.5.2 Theoretical Work 10
    1.5.3 Experimental Work 12
    1.5.4 Structure 12
1.6 Further Comments 14
Literature Review

2.1 Introduction
2.2 Background to PSA
2.3 Adsorbents
    2.3.1 Zeolite Molecular Sieves
    2.3.2 Carbon Molecular Sieves
2.4 The PSA Process
    2.4.1 Common PSA Steps
    2.4.2 Advanced PSA Steps
    2.4.3 Plant Design
    2.4.4 Operating Parameters
    2.4.5 PSA Process Optimisation
2.5 Theoretical Work
    2.5.1 Assumptions
    2.5.2 Modelling Work
    2.5.3 Model Solution Methods
2.7 Applications of PSA
    2.7.1 Air Separation
    2.7.2 Argon Recovery
    2.7.3 Hydrogen Recovery
    2.7.4 Paraffin Separation
    2.7.5 Methane Recovery
    2.7.6 Carbon Monoxide Recovery
    2.7.7 Ozone Enrichment
    2.7.8 Air Drying
    2.7.9 Other Applications of PSA
2.8 Economic Aspects
    2.8.1 Background
    2.8.2 General PSA Processes
    2.8.3 Air Separation
4.4.5 Type V - Instrument
4.4.6 Type VI - Sequencer/Event Manager
4.5 Model Structure
  4.5.1 Terminology
  4.5.2 Internal Model Overview
  4.5.3 Model Initialisation
4.6 Model Solution
  4.6.1 Solution Method
4.7 Plant Control
  4.7.1 Background
  4.7.2 Model/Plant Interface
  4.7.3 Plant Operating Programs
4.8 Concluding Comments

Experimental Apparatus
  5.1 Introduction
  5.2 Plant Layout
    5.2.1 Adsorbent Beds
    5.2.2 Air Pretreatment
    5.2.3 Instrumentation
    5.2.4 Computer
  5.3 Modifications
  5.4 Concluding Comments

Computer Modelling
  6.1 Introduction
    6.1.1 Experimental & Theoretical Aims
    6.1.2 Conditions & Disturbances
  6.2 Comparison With Previous ILE Model
6.2.1 Product Concentration

6.3 Model Sensitivity Studies
6.3.1 Feed End Dead Volume
6.3.2 Product End Dead Volume
6.3.3 Unsymmetrical Purging
6.3.5 Unsymmetrical Beds
6.3.6 Adsorption Parameters

6.4 Conclusions

Control Experiments

7.1 Introduction

7.2 Model Parameter Matching
7.2.1 Method
7.2.2 Proportional Only Control
7.2.3 Proportional-Integral Control
7.2.4 Proportional-Integral Control, Different Conditions

7.3 Checking Process-Model Mismatch
7.3.1 Method
7.3.2 Incorrect Purge Flow
7.3.3 Incorrect Product Pipe Length
7.3.4 Incorrect Product Receiver Volume
7.3.5 Incorrect Product Flow Rate
7.3.6 Conclusions

7.4 Process Control
7.4.1 Controlling Product Concentration
7.4.2 Discussion
7.4.3 Ramp Change Disturbance Rejection
7.4.4 Step Change Disturbance Rejection
Conclusions And Recommendations

8.1 Conclusions
  8.1.1 Program Enhancement
  8.1.2 Experimental
  8.1.3 Comparison Between Theory and Experiment
  8.1.4 Modelling Experiments
  8.1.5 Process-Model Matching
  8.1.6 Dynamics & Control Experiments

8.2 Recommendations
  8.2.1 Model and Simulation Design
  8.2.2 Plant Design
  8.2.3 Future Model and Simulation Work
  8.2.4 Future Experimental Work

8.2 Concluding Comments

References

Object-Oriented Programming Review

A1.1 Introduction
  A1.1.1 Terminology
A1.2 Brief History of Computer Languages
A1.3 Programming Techniques
  A1.3.1 Procedural Programming
  A1.3.2 Object-Oriented Programming
A1.4 Use of OO Techniques With FORTRAN 77
  A1.4.1 Procedural Basis
  A1.4.2 Object-Oriented Interface
A1.5 Conceptual Issues
A1.6 Bibliography
The Structure of the ILE Program

A2 Introduction
A2.1 Program Overview
A2.2 Object Independent Layer
  A2.2.1 Object Type Database
  A2.2.2 Object Connection Database
A2.3 Object Hierarchy
  A2.3.1 Object Methods
  A2.3.2 Object Properties
A2.4 Object Descriptions
  A2.4.2 ObjInitialise
  A2.4.3 ObjCalculate
  A2.4.4 ObjCalcAboutEnd
  A2.4.5 ObjFlowError
  A2.4.6 ObjCompError
  A2.4.7 ObjCalcEnds
  A2.4.8 ObjCycleEnds
  A2.4.9 Extra Methods for the Adsorbent Bed
  A2.4.10 Extra Methods for Valves and Sequencers
A2.5 Time Manager
A2.6 Model Solver
A2.7 Object Code Sizes
A2.8 Example of a Model Used in this Project
A2.9 Run-time Performance
A2.10 Concluding Comments

3D Characteristic Plots

A3.1 Introduction
A3.2 Notation 265
A3.3 Three-Dimensional Characteristic Plots 266
A3.4 Geometrical Calculations 267
A3.5 Problems 268
A3.6 Example Cycle Profiles 271
A3.6 Concluding Comments 277

Non-Linear Behaviour 278
A4.1 Introduction 278
A4.2 Final Concentration 279
A4.3 Bed Concentration Profiles 281
  A4.3.1 Bed Concentrations at Product End 283
A4.5 Transition Between States 287
A4.6 Chapter 6 Results 287
A4.7 Discussion & Conclusions 291

Model-Process Matching 293
A5.1 Introduction 293
A5.2 Unmatched Model 293
A5.3 Matched Model 300
A5.4 Concluding Comments 305
Nomenclature

Roman

A  bed cross section area  \( m^2 \)
C  concentration  \( \text{mol. frac.} \)
\( C_{v_{\text{max}}} \)  valve discharge coefficient
f  total flow rate  \( \text{dm}^3.s^{-1} \)
F  flow rate  \( \text{mol}.s^{-1} \)
\( \bar{F} \)  average flow rate reading  \( \text{dm}^3.s^{-1} \)
g  acceleration due to gravity  \( \text{m}.s^{-2} \)
G  density  \( \text{kg}.m^{-3} \)
I  number counter  -
\( k_N \)  equilibrium model nitrogen adsorption parameter  -
\( k_O \)  equilibrium model oxygen adsorption parameter  -
\( k_c \)  controller gain  -
K  isotherm gradient  \( \text{mol}.kg^{-1}.\text{Pa}^{-1} \)
L  overall bed length  \( m \)
n  amount of gas  \( \text{mol} \)
N  number  -
p  partial pressure  \( \text{Pa} \)
P  total pressure  \( \text{Pa} \)
\( p' \)  equilibrium model pressure  \( (1/P.dP/dt) \)  \( s^{-1} \)
q  amount adsorbed on solid  \( \text{mol}.kg^{-1} \)
R  universal gas constant (8.314)  \( J.\text{mol}^{-1}.K^{-1} \)
\( \bar{R} \)  average instrument reading  -
t  time  \( s \)
T  temperature  \( K \)
v  gas superficial velocity  \( \text{m}.s^{-1} \)
V  volume  \( m^3 \)
y  mole fraction  -
Y  augmented mole fraction  -
\( X_v \)  
valve position \( (0 \ - \ 1) \)  

\( \bar{X}_v \)  
normal output position  

\( z \)  
distance along the adsorbent bed \( \text{m} \)  

**Greek**  
\( \alpha \)  
separation factor  

valve characteristic  

\( \delta \)  
small increment  

\( \Delta \)  
finite increment  

\( \epsilon \)  
total voidage within the adsorbent bed  

\( T \)  
integral action \( \text{s} \)  

\( \omega \)  
adsorbent bulk density \( \text{kg.m}^{-3} \)  

**Subscripts**  
\( \text{dt} \)  
time interval  

\( e \)  
conditions at end of time interval  

\( 0 \)  
position, \( z = 0 \)  

\( L \)  
position, \( z = L \)  

\( N \)  
nitrogen  

\( O \)  
oxygen and argon  

\( s \)  
conditions at start of time interval  

\( \text{sp} \)  
set point
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>Backfill step</td>
</tr>
<tr>
<td>BPE</td>
<td>Bed Pressure Equalisation</td>
</tr>
<tr>
<td>CMS</td>
<td>Carbon Molecular Sieve</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>ILE</td>
<td>Instantaneous Local Equilibrium</td>
</tr>
<tr>
<td>LOP</td>
<td>Lower Operating Pressure</td>
</tr>
<tr>
<td>DP</td>
<td>Depressurisation step</td>
</tr>
<tr>
<td>NU</td>
<td>Null step</td>
</tr>
<tr>
<td>OO</td>
<td>Object Oriented</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>PD</td>
<td>Product release step</td>
</tr>
<tr>
<td>PPC</td>
<td>Per Part Cycle</td>
</tr>
<tr>
<td>PR</td>
<td>Pressurisation</td>
</tr>
<tr>
<td>PSA</td>
<td>Pressure Swing Adsorption</td>
</tr>
<tr>
<td>PU</td>
<td>Purge step</td>
</tr>
<tr>
<td>STP</td>
<td>Standard Temperature and Pressure</td>
</tr>
<tr>
<td>TSA</td>
<td>Thermal Swing Adsorption</td>
</tr>
<tr>
<td>UOP</td>
<td>Upper Operating Pressure</td>
</tr>
<tr>
<td>VOA</td>
<td>Volume Occupied by Adsorbent</td>
</tr>
<tr>
<td>VSA</td>
<td>Vacuum Swing Adsorption</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

1.1 Introduction

This chapter contains a brief introduction to the Pressure Swing Adsorption process (PSA). There is a short section on adsorbents, plant layout and design, the basic steps involved in a cyclic PSA process, and what are normally considered the most important process operating variables. Finally there is a section on the background and aims of this project.

1.2 PSA

PSA is a gas separation process, where mixtures of gases are either separated into their components, one component is removed, or a component is enriched. This is achieved by using a solid phase that either preferentially adsorbs one component or uses a molecular sieving process in which one component diffuses into the adsorbent at a higher rate. The overall driving force comes from pressure changes, and the main operating cost is pressurising the feed gas for the process or vacuum to remove waste gas in the case of vacuum swing adsorption (VSA).

The PSA process is a cyclic operation, involving a sequence of steps which are combined into a cycle which is repeated continuously. There are several common steps that are used in most cycles, along with other specific steps which are used to improve the plant performance depending on the separation duty required.

PSA can be compared with the three other main industrial processes for gas separation, which are:

1) the distillation process, performed at various temperatures and pressures such that there is a liquid and a gaseous phase of the mixture
present within a column containing some packing, for example, the cryogenic distillation of air or the separation of petroleum products.

2) absorption of a gaseous component from a mixture by contact with a solvent, often within a column, for example the case of mercaptan removal from hydrocarbons by the Merox process.

3) membrane separation of a liquid or gaseous phase using a membrane barrier which allows one or more of the components to pass through, whilst preventing others, for example hydrogen recovery from a mixed stream of hydrogen and hydrocarbons.

1.3 Adsorbents

An adsorbent is a solid which has surface properties such that fluid molecules can attach themselves to it by virtue of physical bonding, (the fluid bonding with the surface through Van de Waal forces alone is a process known as physisorption). An important point of the PSA process is that the adhesion process should be easily reversible, otherwise it would necessitate replacing the adsorbent continuously. This reversibility normally excludes solids that undergo a chemical reaction with the fluid molecules (the fluid chemically bonding with the surface is a process known as chemisorption), to form a new compound. In some cases a physical bond can become almost as strong as a chemical one, one example is the very strong adsorption of water on zeolite.

Another type of solid is one that differentiates between the various fluid species present by their size or shape, for example carbon molecular sieves (CMS), which normally act by allowing one component to diffuse much faster than another to the adsorbent surface. This separation works more by the sieving effect of the pores than the final adsorption contribution to the driving force for the diffusion process.

There are many factors that affect the amount of a component that will be adsorbed: the structure, the composition of the solid surface, the
temperature of the system, the size of the surface, and the relative concentrations of the species present. Since the 1960's it has been possible to create an adsorbent specifically for some of the species that may be involved in the separation. This is done chemically as the adsorbent is made, for example in zeolites it is possible to replace the metal ion within the cage with another that endows the zeolite with more appropriate properties for a particular duty.

There are two ways to regenerate the adsorbents used. The first is to depressurise the system such that the components desorb from the surface, known as pressure swing adsorption (PSA). This is the most commonly used technique, for example, air separation or hydrogen recovery. The second method is to apply heat, a process known as thermal swing adsorption (TSA), for example air drying. TSA is often only economical where a free, cheap, or waste heat supply is readily available. However, this process is often limited by the large heat capacity of the adsorbent, which can necessitate several hours of heating and cooling for the required regeneration.

This project is concerned only with gas phase adsorption using zeolites for the production of oxygen from air, a pressure reversible separation process, relying on selective adsorption of nitrogen. Temperature variations do occur within the zeolite beds but this is due to the heats of adsorption and not to external heating.

1.4 Cyclic PSA

Most PSA processes operate a fixed sequence of operations that are cycled repeatedly, and eventually a cyclic steady state is achieved. The cycle will be built up from a combination of steps, which will normally contain at least pressurisation, depressurisation, and product release steps.

There is another method of operating a PSA process, by high speed cycling of the process steps with a constant product draw-off. This is known
as ‘Rapid PSA’ or RPSA. This is not considered in this project, although it is mentioned in the literature search.

Normally more than one bed is used in a PSA process in order that there is a continuous product flow. The use of multiple beds can also help reduce the cost of operation, as pressure energy can be reused rather than lost, by gas depressurising from one bed being used to partially repressurise another bed.

1.4.1 Performance Criteria

All processes are categorised in some way to allow the easy comparison of performance for different plant arrangements for a method, and against other competing methods. In the case of PSA it is normal to define, as a basis, the following properties:

1) Product Yield - the amount of the desired gas component that leaves via the product stream divided by the amount of the same desired gas entering the process in the feed. The higher the value the more the desired gas is recovered by the process.

2) Product Purity - the molar concentration of desired gas in the product stream. This is normally as high as possible, although some processes may require low purity gas, such as oxygen enriched air for medical use.

3) Separation Factor - defined as $\alpha = \frac{y_{prod}}{y_{feed}}$, where $y$ is the mole fraction of the desired component, and is a measure of the concentration gain of the process.

4) Specific Energy Requirement - the amount of energy required to effect the separation. Normally calculated as energy input to the process divided by the number of moles of desired gas in the product stream. The lower this figure the more energy efficient the process is. Units are normally kJ.kmol$^{-1}$.

5) Adsorbent Productivity - a measure of how much separation is being achieved per mean amount of mass or volume of adsorbent. Calculated as molar flow of desired gas in the product stream divided by the total number
of units of adsorbent contained within the beds. The higher the value the
more desired gas will be produced for a given amount of adsorbent. In mass
terms the units are normally kmol.s\(^{-1}\).kg\(^{-1}\), for volume they are kmol.s\(^{-1}\).m\(^{-3}\).

6) Capital Cost - the amount of capital expenditure required to build the unit.

7) Operating Cost - the cost of running the unit, a major proportion of which is the energy costs, but includes maintaining it at a fully operational state.

8) Size and weight - there are often specific limitations on the size and weight of units, i.e. for off-shore use, in aircraft, and in domiciliary units.

Unfortunately the above values cannot normally be considered in isolation as they can and do interact with each other. For example, energy efficiency will go up if a reduced upper operating pressure is used, as less feed gas will need to be pressurised and to a lower level. However, this normally results in a drop in the product purity. Optimum design of a PSA process must balance these values to achieve the desired product purity at the best combination of the other three factors, for the minimum capital outlay to build the unit.

For a small unit, size or weight is often the most important factor and so a high adsorbent productivity is desired, even if it causes higher operating or capital costs, or lower yields. For larger units the energy requirement is usually more important, as higher capital costs can be offset by even a small decrease in energy requirement.

1.4.2 PSA Process Design

There are several options to choose from when configuring a PSA process to achieve a desired separation. The choice of number of beds, adsorbent type, operating conditions, plant layout, materials, and instrumentation is almost infinite. The choice is made by the designer who
will optimise the weight, size, and automatic operation of the unit to effect the desired separation and produce the correct products at the lowest cost.

For example, a portable oxygen generator for enriched air needs to be small, light, and easily transportable. It also needs little control as long as it produces oxygen at approximately the right concentration. A large high-purity nitrogen plant needs more sophisticated equipment to ensure a constant product flow and quality under all conditions.

Large scale air separation units which produce nitrogen are potentially smaller than those for oxygen production. This is because there is a higher proportion of nitrogen in the atmosphere and so, to produce the same volume of oxygen as nitrogen, three and a half times as much air must be processed.

1.4.3 Steps And Cycles

Although the physical layout for the process can be arranged in many ways there are only a limited number of basic step types that can be performed in a PSA process. All the other 'more advanced' steps are combinations of these basic steps. These steps will be described in terms of oxygen production from air using a zeolite adsorbent.

1.4.3.1 Basic Steps

All PSA processes have at least three steps in common, although the pressures may vary. These basic steps are described below for a process making oxygen from air:

Pressurisation

The product end is closed and feed gas enters the bed and raises the pressure until the upper operating pressure (UOP) is reached. During this process a greater proportion of the nitrogen (the more adsorbable component)
is removed from the gas phase, leaving a zone of gas at the product end concentrated in oxygen.

**Product Release**

Feed air is used to maintain the UOP pressure in the bed whilst gas enriched in oxygen is taken from the product end of the bed.

**Depressurisation**

The product end is closed and the waste valve at the feed end is opened. This step partially regenerates the bed, as the drop in pressure causes nitrogen to be desorbed from the bed, as it depressurises to the lower operating pressure (LOP). This gas is normally vented to atmosphere and is considered as waste gas.

A PSA cycle that consists of these three steps alone is known as the ‘simple cycle’ and normally gives a relatively poor separation. In addition, it gives a poor yield and is very wasteful of energy. Therefore this cycle is not normally used commercially.

**1.4.3.2 Advanced Steps**

These are variations of the above steps as well as new steps. They are all designed to provide a better separation of the gases by altering the state of the bed to more favourable conditions or reusing gas or pressure energy that would normally be wasted. The effects of these steps was discussed by Kirkby & Kenney, 1986, who found that the Backfill step was most efficient, as its use shortened the cycle time. All these steps increase the yield and separation factor of the PSA process. The main advanced steps are described in the following sections.
**Purge**

This step uses some of the product gas rich in oxygen to lower the partial pressure of nitrogen within the bed. Product gas is introduced at the product end of the bed, in the reverse direction. This forces more nitrogen to be desorbed and this exits via the waste line, which is kept open to maintain the bed at the LOP. This step is normally placed after the depressurisation step in the simple cycle and the cycle known as the 'purge cycle'. The addition of the purge step into the simple cycle usually improves performance considerably.

**Backfill**

This is essentially the same step as purge, but the waste valve is kept shut and so the pressure in the adsorbent bed rises above the LOP. It is really a partial repressurisation step, only using product gas entering at the product end. The nitrogen rich gas collects at the feed end, but, unlike the purge step, no gas is lost into the waste line. Additionally, as the bed is partially repressurised the pressurisation step can be shortened.

**Co-current Depressurisation**

Rather than depressurise the bed through the waste line, depressurisation is done through the product valve. This allows any oxygen rich gas remaining at the product end of the bed to be re-used, often in a backfill step. Alternatively this step could completely replace the product release step for the cycle.

There are many other steps that can be performed: second-cut, where the last of the high pressure product release gas is fed into the feed of a high pressure bed; Bed Pressure Equalisation (BPE), where a bed is connected to another and the pressure equalises between them. The latter step is often used in large commercial PSA units in order to reuse pressure energy, hence
reducing the operating cost of the process. For a multi-bed process this step can be done several times in a cascade to recover as much energy as possible.

The relative merits of all these steps and cycle types are discussed in the literature, see Chapter 2.

1.4.4 Important Operating Variables

Probably the most important operating variables under direct control, apart from the cycle steps, are the UOP, the LOP, and the rate of product removal.

The temperature of the process cannot normally be controlled and the rate of pressurisation and depressurisation have only a minor influence (within certain limits, see Section 2.4.4) on the equilibrium dominated separation of air on a zeolite.

It is possible to use purge gas either directly from a bed in a product release step or gas can be taken from a product receiver, in which case it is a well-mixed stream of average product composition. However, the source of the purge gas appears to have a relatively minor effect on the product purity, except with cycles that are over purged, as demonstrated by Espitalier-Noel 1988.

1.5 The Project

1.5.1 Introduction

Separation of mixtures is the major cost in the chemical, petrochemical and related industries, Jasra et al, 1991. Absorption, distillation, and adsorption are the three most widely used methods. Even a small increase in the performance of PSA processes could save large amounts of energy and therefore reduce the cost of operation significantly.

Normally a PSA unit is operated using a fixed sequence of steps in a cycle known to give the desired product concentration at the desired rate.
There may be little room for improvement in PSA technology in terms of new step types, but there is wide scope for improvement of adsorbents, plant layouts, and process optimisation under varying conditions.

In addition, little work has been published on controlling the PSA process in order to reject disturbances or follow set-point changes, or to operate the process across a wide range of process conditions and product requirements, for example maximum product purity at high and low product rates.

This work continues that of Kirkby (1983) who looked at the basic control of an oxygen producing PSA process in order to maintain the product purity during changes to the product flow rate. He concluded that the most promising method of control for the process was to modify the cycle and step times. His overall conclusions were that the PSA process has simple first, second or third order lag dynamics and dead times which depend on the reservoirs within the system. Finally, he showed that some of the control problems could be overcome, provided the most severely non-linear operating regions were avoided, as, in these areas, the non-linear behaviour of the process made controlling it a non-trivial matter.

### 1.5.2 Theoretical Work

Modelling is an abstraction of reality with the purpose of understanding reality better. This is often done to improve the understanding of an existing item, visualise it when created, test it before building it, or to reduce the complexity of it to where it can be understood. As a model can omit any details considered non-essential to the fundamental operation it is usually easier or faster to manipulate than the physical thing it represents. No model is truly correct or incorrect, only adequate or inadequate for the intended use.

The first major aim was to take an ILE type model and add a flexible process simulator to allow quick and easy simulation of various plant
arrangements. It could then be used to provide information that could not easily be obtained from a PSA plant. In addition the simulator was designed to run at faster than real-time speeds for on-line simulation of an operating PSA plant. Information from the simulator could then be used to control the plant over a wide range of operating conditions without the problems encountered by using a simple control scheme, such as a PID controller.

The ILE model was chosen as it could be solved extremely rapidly, due its simplifying assumptions, and so could be computed in real-time, i.e. as the process was running. By using data from Espitalier-Noel (1988) the model could be compared to previous experimental data and the existing ILE model. The main problem of the ILE model is that it tends to over predict the product concentration, however, by on-line feedback of the product purity from the plant, the model parameters could be adjusted in order that the model gives the same output as the real process.

Many of the practical difficulties of the existing ILE model were to be overcome by creating a more flexible model. This would allow the process layout to be decided by the model user and not be hard-coded into the model as in many previous programs. The intention was to create a model that could simulate any existing process step as well as any new ones invented at a later date.

The second aim was to add the ability to sequence the PSA plant to the real-time model, in order to collect experimental data and to allow the model to interact with a real process. By using this data the model could be made a reasonable representation of the pilot plant. Once this had been done a parametric study of the performance of a PSA plant with respect to process configuration details could be made.

The third aim was to have the model adapt to match data stored from previous runs of the model, but that had been made with different parameters. In this way the effects of model-process mis-match could be studied relatively quickly.
The final aim was to test a product purity control method system based on information provided by the model that would not normally be measured on a real plant, leading to better understanding and process control of the system.

The development of a new, stand-alone PSA model, with the intention of a significant improvement in accuracy over those already published, was not an aim of this project.

1.5.3 Experimental Work

An existing PSA plant for the production of oxygen from air had already been built by Espitalier-Noel (1988). Investigations had been performed into the effect of recycling waste gas into the feed, as well as optimising the steps, to improve the process yield and productivity.

Some modifications had to be made to the pilot plant in order that disturbances could be introduced into the product line, to simulate the effect of a sudden change in product demand, an event which is common in industrial applications.

The experimental nature of the project was to implement a model-based observer control scheme on a PSA unit, with the ability to measure and record the response of the process. Basic experimental work was done to demonstrate the model running in real-time whilst sequencing the pilot plant. Data from this experimental work would be used to make the model used in the modelling experiments a reasonable approximation of the pilot plant.

1.5.4 Structure

The dissertation presented falls into four broad areas:

1) General introductory chapters:
Chapter 1 is a general introduction to the work done in this project; Chapter 2 is a review of PSA literature;
Chapter 3 describes the ILE theory behind the adsorbent bed model in the simulation program.

2) A description of the computer simulation and the pilot plant used:
Chapter 4 describes the computer program used to model the pilot plant and how the sequencing of the pilot plant is achieved;
Chapter 5 details the pilot plant used.

3) Theoretical and practical results:
Chapter 6 details the way the model compares to the pilot plant for a range of conditions and theoretical results from the model on the effect of changing various process parameters, including the effects of non-symmetrical processes.
Chapter 7 describes the ability of the model to match to a process with different conditions, the effect of model-process mis-match, and experimental results from the model with and without product purity control.

4) Discussion, conclusions, and recommendations:
Chapter 8 discusses the comparisons between theory and experiment, draws conclusions, and gives some recommendations for future work.

5) Appendices
Appendix 1 looks at the use of object-oriented programming.
Appendix 2 describes the internal structure of the computer model used throughout the work.
Appendix 3 explains the generation and interpretation of three-dimensional characteristic plots.
Appendix 4 describes some non-linear behaviour found during this project.
Appendix 5 discusses the matching of the model to pilot plant data.
1.6 Further Comments

PSA is still a relatively young separation technology and there are many theoretical and practical aspects of it that are still not well understood. Although a large amount of work has been done on modelling the process, there are some notable areas of omission, for example, the effect of dead volumes within the system or non-symmetrical operation. The two main areas of improvement for the PSA process are now likely to be: the increasingly sophisticated tailoring of adsorbents for specific uses to increase the number of applications where the PSA process can be used and to further improve the economic range for current areas; better understanding of the dynamics and control of the process in order that PSA plants can be used in demanding applications whilst guaranteeing such things as product purity, power requirements, and rangeability.

This project looked at the dynamics and control of a PSA unit producing oxygen from air, however the work was kept reasonably generic in order that many of the findings will be applicable to PSA processes used in other areas.
Chapter 2

Literature Review

2.1 Introduction

Since the start of the 1960's the amount of literature published on the subject of PSA processes has risen at a remarkable rate, showing the importance of this area in the separation industry. There are a wide range of adsorbents to choose from, which are normally tailored to the specific process, and there are many areas of application where PSA is the preferred separation method.

This chapter is a review of the main work in PSA, focusing on air separation as this is the subject of this project. Other work is included where it provides some insight into how the process works or as information to be considered for a wider viewpoint. In addition a section on control is included as a control scheme is one of the main aims of the work.

This chapter also looks at the general background of the PSA process: the adsorbents that are used, the steps, cycles, and plant designs commonly used; and the aspects of control theory that should be considered.

2.2 Background to PSA

Finlayson & Sharp (1932) working for British Celanese Limited (now part of Courtaulds Ltd) in Great Britain, were granted a patent for separating or enriching a mixture of gases. The process used an adsorbent solid to adsorb the gas at high pressure, followed by fractional desorption of the adsorbed gases using a series of pressure changing steps. The adsorbents used were charcoal and silica gel, which are still in use today. There is no known commercial application or use of their discovery, but they describe all the main uses of PSA in operation today.
In 1942 Kahle, working for Linde A.G., created a two-bed water and carbon dioxide removal system for the pre-treatment of air for a cryogenic air separation unit. Nitrogen from the cryogenic unit was used to purge the beds clean.

Almost twenty years later Skarstrom (1960), working for the Esso Research and Engineering Company (now Exxon Research and Engineering) in the US, was granted a patent for a ‘Method and Apparatus for Fractionating Gaseous Mixtures by Adsorption’. He described a two-bed process which underwent a cyclic operation involving pressurisation, product release, purge, and depressurisation steps. The adsorbents described were varied and included many still used today, including molecular sieves and zeolites, as well as others such as bone char and soft tissue paper. The separations described were the drying and separation of air. His work is often considered to be the basis of all the following work on the PSA process in the thirty five years since.

Broughton et al (1961), working for UOP in the US, patented a method for the adsorptive separation of mixtures of fluids using a fixed adsorbent bed and a series of moving inlets and outlets. They described uses including the separation of hydrocarbons and water softening.

Coinciding with the creation of the modern PSA process, Milton (1959), working for Union Carbide in the US, patented various methods for preparing a number of artificial molecular sieves specifically for use as adsorbents. He named the synthetic sieves as ‘zeolite X’ and ‘zeolite A’ to distinguish them from natural zeolites. The differences between the two are in the ratios of the various molecules within the crystals and the shape and size of the pores.
2.3 Adsorbents

A suitable adsorbent for a PSA process is usually determined by several main criteria:

1) Adsorption capacity, i.e. the total amount of gas that can be adsorbed onto the surface of the solid per unit volume, this requires a high surface area, therefore a pore-like structure is most suitable.

2) High selectivity of the solid surface to retain one component in preference to another is required.

3) Mechanical & physical properties, such as weight, thermal stability, and resistance to crushing, attrition etc. when placed in a vessel.

4) The relative cost of the adsorbent.

It is partly because of the development of artificial adsorbents that the PSA process is so widely used. By tailoring the adsorbent specifically to the desired separation required the above factors can be enhanced to give the desired duty.

The selectivity of an adsorbent for one component over another can be achieved in three ways:

1) By shape or size difference between the components leading to kinetic effects, where the components have different adsorption rates onto the surface or diffusion rates into the pores of the solid, e.g. in the separation of hydrogen from hydrocarbons the hydrogen molecules are much smaller. An adsorbent with small pores will only allow hydrogen to enter the pores (the adsorbent acting as a molecular sieve) and so the hydrocarbon stream is depleted in hydrogen.

2) By equilibrium loading, where there is a significant difference between the adsorption isotherms of the components, e.g. for the case of air separation using a zeolite.

3) A combination of the two above, e.g. when using a CMS for air separation results in oxygen diffusing faster into the pore structure, which
then undergoes adsorption onto the surface, leaving nitrogen as the product. However, if the system is allowed to come to equilibrium then nitrogen will also diffuse into the adsorbent and adsorb equally.

2.3.1 Zeolite Molecular Sieves

Natural zeolites are crystalline structures of aluminosilicate, based on AlO$_4$ and SiO$_4$ tetrahedrons. Artificial zeolites have a similar composition and arrangement of molecules. Milton (1959) described type A and type X synthetic zeolites, the difference being in the ratio of Si to Al molecules and in the resulting shape of the pore apertures, type A being a square ring, type X having a hexagonal ring. This method of describing zeolites is still in use today.

There are several features of zeolites that make them particularly useful and have contributed to their success and widespread use:
1) High thermal stability, they can withstand wide changes in temperature without rapid physical degradation.
2) Uniform pore size, giving good, repeatable performance between batches.
3) Easily modifiable, by changing the cation within the crystal structure.
4) High adsorption capacity, even at low concentrations or raised temperatures.

In the common synthetic zeolites, the pore size ranges between 4.5Å for a type 5A, to 8.5Å for a type X. A comprehensive discussion of the properties of zeolites is given by Breck (1974).

2.3.1.1 Oxygen Production

Zeolites are used in air separation for the production of oxygen as they exhibit a higher equilibrium loading for nitrogen, thus they remove a higher proportion of nitrogen from the gas phase leaving it rich in oxygen. It is important to note that most zeolites show very little preference for oxygen.
over argon and hence oxygen and argon normally act as a single component, Boniface & Ruthven (1993), and it is very rare for an adsorbent to preferentially adsorb argon over oxygen.

There are many references to data for adsorption equilibria for oxygen and nitrogen on molecular sieves. Many are presented as pure component isotherms or at temperatures and pressures that are not normally of use in a PSA process, for example Valenzuela & Myers (1989). Others have presented their data specifically for use in PSA applications, for example Miller et al (1987).

Sorial et al (1983) looked at data for the pure gases and for various mixtures of the two gases. They compared their binary experimental measurements against predictions based on the pure gas isotherms and concluded that best representation of the data was given by the ideal adsorbed solution theory of Myers & Prausnitz (1965), the Cook & Basmadjian model (1965), and the statistical thermodynamic model of Ruthven et al (1973, 1976). Significantly they found that a change in temperature or pressure made less difference to the separation factor than changes in the amount of gas that was adsorbed onto the surface.

Liow & Kenney (1990) found that the adsorption rate of oxygen within the micropores of the zeolite was twice that for the nitrogen phase. They concluded that the smaller the particle size the closer the results would tend to those predicted by the instantaneous local equilibrium model, however higher pressure drops resulted across the bed.

Dry air contains 0.93% argon which adsorbs very similarly to oxygen on most adsorbents used for this application. For low grade oxygen, the argon is not important, but for uses such as welding and cutting the oxygen must be of high purity to achieve the hottest flame temperature possible. See Section 2.3.1.3 for argon recovery.
2.3.1.2 Nitrogen Production

In conventional oxygen producing PSA processes, using a zeolite, the waste stream is not pure nitrogen because the oxygen recovery is not 100%. Designing a PSA cycle using zeolites to produce pure nitrogen as a product is difficult and is not practised commercially. Recent attempts have been made to create a zeolite with a higher diffusivity for oxygen, using molecular engineering, for example, zeolite 4A can be modified by replacing the sodium cations with potassium and by 'dissolving' divalent iron. The iron causes a partial closure of the pore, reducing the rate of diffusion of nitrogen.

Shin and Knaebel (1987 & 1988) studied nitrogen production using the zeolite RS 10. Their model predicted the trends of product purity and recovery. They studied only four basic steps and assumed linear pressure changing steps together with constant pressure purge and product steps.

2.3.1.3 Argon Recovery

Producing pure argon using the PSA process would provide an alternative source of argon to that of cryogenic distillation from air, hydrogen depleted ammonia gas, or natural gas.

Boniface & Ruthven (1993) presented results for nine adsorbents, of which only one, clinoptilolite, was able to separate argon from oxygen with oxygen as the preferentially adsorbing component and so be suitable for the production of pure argon. However, they could offer no explanation as to why this adsorbent preferred oxygen to argon.

Only one adsorbent that they tested showed even a small preference for argon, Ag-mordenite and, although the preference was too low for a normal PSA unit (with a separation factor of 0.83), it is pointed out that the separation factor would increase at lower temperatures and a sub-ambient temperature unit may be possible. Interestingly, on this material the separation factor of nitrogen over oxygen is over 20, normal PSA units operate with a factor of less than 10. Unfortunately, they made no mention
of the relative costs of the adsorbents, an important factor in the viability of a PSA process.

2.3.2 Carbon Molecular Sieves

The other common adsorbent used in air separation, for the production of nitrogen, is the carbon molecular sieve (CMS). Activated carbon is chemically similar, but differs as shown in Table 2-1.

<table>
<thead>
<tr>
<th></th>
<th>CMS</th>
<th>Activated Carbon</th>
</tr>
</thead>
<tbody>
<tr>
<td>pore-size distribution</td>
<td>5 - 10 Å</td>
<td>10 - 1000 Å</td>
</tr>
<tr>
<td>pore volume</td>
<td>0.25cm$^3$.g$^{-1}$</td>
<td>0.95cm$^3$.g$^{-1}$</td>
</tr>
</tbody>
</table>

Table 2.1 - Data For CMS And Activated Carbon

Common methods of manufacture involve thermal treatment of coals, hydrocarbons or the burning of polymers, followed by activation in some way, often using steam. Many companies have favoured sources of CMS, for example, BOC use carbonised organic material (coconut shells). However, the exact methods used in the manufacture of a particular CMS are closely guarded industrial secrets.

The main problem with some CMS is the variation in pore size distribution between batches. This results in the effective (time dependent) selectivity of a CMS being below that of a zeolite. As the CMS has a very low equilibrium selectivity between nitrogen and oxygen a small change in pore size can cause a large change in performance of the sieve.
2.3.2.1 Nitrogen Production

The use of a CMS to separate nitrogen from oxygen and argon relies on the differences in diffusivity within the adsorbent. Oxygen has a smaller critical diameter than nitrogen and so diffuses faster. As the total amount adsorbed for each component at equilibrium is similar this process is time dependent. Many studies have been performed on CMS to find the effect of changing the pressurisation rate and time on the nitrogen production.

Argon normally follows the nitrogen in a CMS PSA process, however from their relative molecular diameters it is difficult to see why, as argon has a very similar diameter to that of oxygen, see Moore (1972).

Ng et al (1993) showed that the oxygen mass transfer zone within the CMS bed is very wide and changes shape with time. They concluded from their work that the shape of the zone was mainly dependent on time and slightly dependent on feed gas flow rate or temperature.

Shirley & LaCava (1993) experimented with nitrogen separation from air using a CMS. They studied the pressurisation rates to see if they could be controlled in order to improve the yield or product purity. Their main conclusion was that a slug of oxygen was often left at the product end of the bed after pressurisation. By slowing the pressurisation rate more time was available for this oxygen to diffuse into the CMS. They recommended a slow initial pressurisation to an intermediate pressure, followed by a more rapid pressurisation to the upper operating pressure. This indicates that knowing the internal compositions within the adsorbent beds can enable the PSA cycle to be made more efficient than at present.

2.4 The PSA Process

A PSA unit usually consists of two or more beds, although if a continuous product flow is not required a single bed may be used, connected such that high pressure gas can be applied from one end and product gas
taken from the other. Waste and purge lines are used to transfer gas to and from the beds as part of the cleaning stage.

The PSA process is normally cyclic, i.e. the process cycles through a set of predefined steps in order to achieve a particular cyclic steady state performance.

2.4.1 Common PSA Steps

<table>
<thead>
<tr>
<th>Name</th>
<th>Feed Valve</th>
<th>Waste Valve</th>
<th>Product Valve</th>
<th>Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressurisation</td>
<td>open</td>
<td>closed</td>
<td>closed</td>
<td>increasing</td>
</tr>
<tr>
<td>Product Release</td>
<td>open</td>
<td>closed</td>
<td>open</td>
<td>constant, increasing, or decreasing</td>
</tr>
<tr>
<td>Co-current</td>
<td>closed</td>
<td>closed</td>
<td>open</td>
<td>decreasing</td>
</tr>
<tr>
<td>Depressurisation</td>
<td>closed</td>
<td>closed</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>Backfill</td>
<td>closed</td>
<td>closed</td>
<td>open</td>
<td>increasing</td>
</tr>
<tr>
<td>Purge</td>
<td>closed</td>
<td>open</td>
<td>open</td>
<td>constant, increasing, or decreasing</td>
</tr>
<tr>
<td>Depressurisation</td>
<td>closed</td>
<td>open</td>
<td>closed</td>
<td>decreasing</td>
</tr>
<tr>
<td>Null</td>
<td>closed</td>
<td>closed</td>
<td>closed</td>
<td>constant</td>
</tr>
</tbody>
</table>

Table 2.2 - Common PSA Step Types

There are seven common step types that can be defined for an adsorbent bed, assuming that high pressure gas is to be fed in at the feed end and the product gas is removed from the other. These seven are shown in Table 2.2. A short description of each is given below:
Pressurisation

Feed gas is used to pressurise the bed whilst the product end remains closed. This is the basic pressurisation step used in many of the PSA processes, however it can be replaced by the Backfill step.

Product Release

Both ends of the bed are open. Depending on the relative flows in and out of the bed the pressure may be increasing, constant or decreasing. Normally product release is done at a constant pressure, however to save time (and hence shorten the length of the cycle) it is sometimes done with a pressure change. For example, by opening the product valve before the upper operating pressure is reached, product can be removed whilst the bed is still being pressurised from the feed end.

Co-current Depressurisation

The feed end of the bed is closed. The product end is opened and the bed depressurises into the product line. This depressurisation may be continued until the lower operating pressure is reached or, more normally, until some intermediate pressure is reached. Suh & Wankat (1989) decided that this step can increase the product delivered in some situations. However they recommended that caution should be exercised in the decision to use the step as its effectiveness depended on the separation factor. In the case of air separation they recommended the ordinary depressurisation step.

Backfill

Product gas is used to pressurise the bed whilst the feed end remains closed, i.e. the bed is filled in a direction backwards to that normally used. This is basically the same step as the pressurisation step but performed from the opposite end of the bed. The gas can either come from a product reservoir or direct from a bed on another step.
There have been few studies published on this step type. Kirkby (1983) produced several interesting experimental results for the backfill step. As well as higher product yields it was noted that the backfill cycles took longer to achieve steady state, in some cases up to twice as long as a purge cycle. Better results were obtained using both a purge and a backfill step. Kirkby & Kenney (1987) performed both theoretical modelling and experimental work on this step. It was shown that increasing the highest pressure reached in the backfill step above around half that of the mean feed pressure made no further difference to the product concentration or yield. The backfill cycles produced almost the same product concentrations as for a purge cycle, but with higher yields.

Liow & Kenney (1990) showed, for oxygen separation from air, that by using an appropriate backfill step, with no purge step, the separate pressurisation step could be eliminated by combining it with the product release step.

**Purge**

Both ends of the bed are open and product gas from a product reservoir or direct from a bed on a product release step is passed through the bed. Depending on the relative flows in and out of the bed the pressure may be increasing, constant or decreasing. Normally purge is done at a constant low pressure, however to save time (and hence shorten the length of the cycle) it is often performed simultaneously with a pressure change. For example, by keeping the waste valve only partially open whilst gas is flowing in from the product end the pressure in the bed can be gradually increased.

The function of the purge step is to reduce the partial pressure of the more strongly retained components. It is usually carried out at low pressure to minimise the amount of gas required.
Countercurrent Depressurisation

The product end of the bed remains closed and no gas is added via the product end. The feed end is opened (normally through a separate waste valve rather than the feed valve) and the bed depressurises through the waste line. This depressurisation is normally continued until the lower operating pressure is reached or possibly down to some intermediate pressure. In the case of nitrogen production from air using a CMS, careful control of the depressurisation rate can create a cycle that requires no separate cleaning step. This occurs as the oxygen diffuses out of the CMS more rapidly than the nitrogen, leaving the bed rich in nitrogen at the end of the step.

Null

During a null step the ends of the bed are sealed and the pressure remains constant, although some pressure equalisation may take place within the bed itself. A null step is placed in a cycle to allow individual step times to be changed without affecting the overall cycle time. A null step is usually used for flexibility by researchers, it is uncommon to see it used in an industrial cycle.

These common basic steps are combined to create a cycle. The simplest cycle consists of pressurisation, product release, and depressurisation, and is known as the 'simple cycle' For a two-bed process the cycle can be represented diagrammatically as shown in Figure 2.1.

<table>
<thead>
<tr>
<th>Bed A</th>
<th>PD</th>
<th></th>
<th>DP</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bed B</td>
<td>DP</td>
<td>PR</td>
<td>PD</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.1 - Simple Cycle for Two Beds**
The more common 'purge cycle' consists of pressurisation, product release, depressurisation, and purge. For a two-bed process a purge cycle would be set out as shown in Figure 2.2.

<table>
<thead>
<tr>
<th>Bed A</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bed B</td>
<td>DP</td>
<td>PU</td>
<td>PR</td>
<td>PD</td>
</tr>
</tbody>
</table>

*Figure 2.2 - Purge Cycle for Two Beds*

These cycles provide a continuous release of product, when the product release step finishes on one bed, the other bed starts to deliver product.

### 2.4.2 Advanced PSA Steps

Although several common step types have been described there are two other additional steps that are often performed:

1) Depressurisation from both ends simultaneously. This is the concurrent and countercurrent depressurisation steps performed together. This could be used to shorten the cycle length by allowing some product release to occur whilst depressurising the bed to waste.

2) Pressurisation from both ends simultaneously. This is the pressurisation step combined with the backfill step. This would allow rapid pressurisation of the bed, however the distance that the backfill gas would pass into the bed would be reduced, thus affecting the product concentration for a given flow rate.

There are many variations as to where the gas comes from or goes to, the final pressure to which the steps are taken, or the pressure level in constant pressure steps. This gives an almost limitless combination of step conditions. One of the more interesting of the advanced steps is the bed pressure equalisation step, as follows.
Bed Pressure Equalisation

This step is often used in hydrogen purification and in large multi-bed units for air separation. It is done by connecting two-beds together, either product end to product end, feed end to feed end, or both ends to both ends. It is often used to perform the backfill step whilst transferring pressure energy from a high pressure bed to a lower pressure bed. On a large multi-bed system this may be done in several stages, involving a cascade between several beds.

Hassan et al (1987) modelled the process, for nitrogen production from air using a CMS. It was shown that when using a BPE step the inclusion of a backfill step into the cycle reduced the product yield (for the same product concentration) and was therefore of no use. No significant comments were made on the relative performance benefits of including the BPE step in a cycle, as the work concentrated more on the accuracy of the model.

2.4.3 Plant Design

There are many ways of constructing a PSA unit, from a single bed oxygen enhancer for breathing applications to multi-bed units producing 30t/day of product gas. Most units designed for oxygen production use either a separate guard bed or part of the bed to remove moisture and carbon dioxide from the feed air. This is necessary as water will adsorb onto the zeolite almost irreversibly (at the conditions normally used within the process) and gradually reduce the adsorptive capacity of the bed.

2.4.4 Operating Parameters

When the plant duty, configuration and layout have been specified, apart from the actual cycle to be used, there are many parameters that have a significant effect on the performance of the plant. For the case of the purge cycle these are shown in Table 2.3.
| Independent Design Variables | Feed temperature  
|                             | Adsorbent type  
|                             | Particle dimensions  
|                             | Bed length  
|                             | Bed cross-sectional area  
|                             | Number of beds  
| Pressurisation              | Upper operating pressure  
|                             | Pressurisation time  
| Product Release             | Product flow rate  
| Depressurisation            | Lower operating pressure  
|                             | Depressurisation time  
| Purge                       | Purge flow rate  
|                             | Purge source  
|                             | Purge time  

**Table 2.3 - Purge Cycle Operating Parameters**

In the case of the pressurisation and depressurisation times, it is normal that the rate of pressurisation and depressurisation are dependent on the length of the step, for example if the pressurisation time is short then the flow rate must be high, assuming all else is constant. This will affect the performance of the unit, especially in the case of nitrogen production from air using a CMS which is rate dependent.

For the actual adsorbent used in the PSA process there are several parameters that can be adjusted or tuned to improve performance. Kumar (1994) looked at separating methane and hydrogen to produce high purity hydrogen. He found that it was necessary to look at the mass transfer and selectivity properties of the adsorbent in combination in order to produce the best adsorbent. For example, a highly selective adsorbent may have a high mass transfer resistance and so perform badly when used in practice. He also noted that there are other factors to be considered, such as a low bulk density of an adsorbent resulting in a reduction in capacity, as less surface is
available, and a loss of physical strength, as there are more macro-pores in the structure.

2.4.5 PSA Process Optimisation

The optimum design for a PSA process, as opposed to a particular step, is difficult to define and is normally dependent on the desired duty. For example, in oxygen production from air there are several possible measures of performance: product concentration, product yield, and specific energy requirement could all be used. Optimising on one factor may be to the detriment of one or more of the others, the overall capital cost, or the operating cost of the unit.

Work has been done on this area by many researchers. Smith & Westerberg (1991) looked at the annual cost of a PSA unit and produced a set of equations for the adsorption performance, the capital cost, and operational cost of each step. The model for the adsorption process was greatly simplified to reduce the computational time required and to avoid needing the detailed physical constants necessary for such models. By solving the model for various pressure equalisation steps they produced an estimate of the optimum design, based on size, operating conditions, and plant configuration. They suggested that the results obtained should be further refined by use of more detailed simulations or by experiment to ensure that the design chosen is capable of meeting the desired duty.

2.5 Theoretical Work

Over the last twenty-five years there have been many models published using fewer and fewer simplifying assumptions to improve the accuracy of the predictions. Original models looked at trace components in a non-adsorbable carrier gas whereas later models describe bulk gas separations. All the assumptions in the models are normally made to reduce
the computation time required to solve the model or to allow for a lack of measured physical data on the adsorbents.

2.5.1 Assumptions

The various assumptions used in the model in this project are given and discussed, together with appropriate references from the literature. The actual equations developed are summarised in Chapter 3.

1) The feed gas is comprised of 20.95% oxygen, 0.93% argon and 78.12% nitrogen (on a volumetric basis), of which oxygen and argon behave as a single component, hence it is assumed that \( y_{\text{O}_2\text{in}} = 21.88\% \) (by volume). Many researchers have found that argon behaves similarly to oxygen. In fact it is this similarity that results in the PSA process being unable to produce 100% oxygen as there is currently no commercial adsorbent that will separate argon from oxygen to give a pure oxygen product, Boniface & Ruthven (1993).

2) The ideal gas laws apply throughout the system, i.e. \( PV=nRT \) at all times.

3) There are no temperature changes within the system, which operates at 293K. The effects of assuming a non-isothermal bed were first studied by Chihara & Suzuki (1983). They simulated a PSA system with mass and heat transfer within the beds for air drying with activated alumina, a system that can be considered as trace component removal from a carrier gas. They concluded that for a bed radius of less than 0.05m their system could be treated as being isothermal.

4) Instantaneous Local Equilibrium (ILE) is established at all times within the adsorbent bed. This assumes that there are no mass transfer resistances within the bed. This was experimentally verified by Kayser & Knaebel (1986) who built a PSA system specifically to separate oxygen from air under conditions such that the ILE model would be valid. These experiments showed that the ILE model was valid as long as some operating limits were not violated.
5) No axial or radial dispersion occurs within the bed, i.e. the gas flow through the bed is in plug flow and there are no wall effects.

6) There is no significant axial or radial pressure drop within the bed. Assuming no axial pressure drop is common to most simulations. Much research has been done into the effect of pressure drop, as in Alpay (1992). However the pressure within the beds in the model used is assumed to be a function of time only. Espitalier-Noel (1988) showed that the beds in the pilot plant had relatively minor pressure drops, under the normal operating conditions of the pilot plant.

7) The beds are uniformly filled with identical adsorbent particles, leading to an even distribution of adsorbent capacity within the bed.

8) There is no pressure difference across any object in the system, except for the valves.

9) Linear, independent isotherms can be used to represent the adsorption isotherms of the two components. The linearity assumption was validated by Kayser & Knaebel (1986), (as in the work described in 4 above) where the experiments were conducted in a region where the isotherms of nitrogen and oxygen are reasonably linear. Finally, Miller et al (1986) showed that the isotherms of nitrogen and oxygen on zeolite 5A can be considered reasonable independent at pressures at least up to 3bar.

### 2.5.2 Modelling Work

Modelling work can be separated into two broad categories, separated by the availability of computing performance. The early work was limited by the lack of readily available computer power to solve the more complicated models, i.e. more complicated models simply could not be solved in a reasonable period of time.

As computer performance increased more complicated models became solvable within a realistic time scale. It is common that a model will take as
long to run today as it did twenty years ago, however it will be more accurate, more flexible, and include far fewer assumptions.

2.5.2.1 Early Work

Possibly the first numerical model of a PSA process was that by Shendalman & Mitchell (1972) in which they looked at 1% carbon dioxide in helium adsorbing on silica gel. This is a one component adsorption process, as helium is inert in this system, and is an example of trace gas removal. The concentration was chosen to create an almost linear adsorption isotherm. Their model showed reasonable qualitative agreement with the experimental results, but was less accurate quantitatively. Their assumptions included isothermal operation, instantaneous local equilibrium, and ignored any effect from the pressure changing steps.

2.5.2.2 Later Modelling

The later models remove most of the simplifying assumptions made in the early models. This makes them more accurate, however they are more complicated and can actually take longer to solve despite the massive increase in computer performance. Several models are notable for the advances they made, these are described in the following sections.

2.5.3 Model Solution Methods

There are many different methods of solving the models, each having some advantage over the others in certain circumstances. There is as yet no general method of solution that is optimal in all situations. The optimal solution of a model is an accurate, fast, and robust solution and many of the solution methods currently used fail on one of these criteria at some point.

For example: a solution method for the bulk separation of gases is not normally optimal for a trace component system. The assumptions within the models will be different and so the solution methods must change to suit the
final equations generated to give the fastest solution for each. Using the same method of solving the equations may be possible for both cases, but for one may be highly inefficient.

All models are systems of partial differential equations, since the gas phase concentrations are a function of both position and time. There are several solution methods: method of characteristics (if the equations are hyperbolic), finite differences, finite elements, orthogonal collocation, and orthogonal collocation of finite elements (Alpay, 1992).

### 2.6 General Solutions - PSA Simulators

A PSA simulator can be considered as a generic solution to the problem of modelling the PSA process, allowing for the wide variations that are possible in adsorbents, plant layout etc. This can be compared with the general purpose steady state process simulators that are widely used in the process industries today. In contrast, most normal PSA models published in the literature are specific to the process being modelled, i.e. a single column, or a two-bed process.

The first reference to a PSA simulator was by Ishida & Takeuchi (1987). They developed a simulator that could model non-isothermal, multi-component, and multi-adsorbent columns. A language extension to TIT (The Intelligent T), a FORTH like language (which was popular 10-15 years ago) was created to allow a user to specify a layout of their own choosing with minimum effort. The language extensions added commands to perform various step types, such as ADSorption, PURGE, REPRessurisation etc, as well as LINEARISOTHERM to override the default Langmuir isotherm. This was a fairly basic process simulator but was effective and capable of simulating a system containing up to six components, eight columns, and four adsorbents.

This program ran on an 8 MHZ 8086 personal computer. However, no mention is made of the computational time it required to solve the systems.
considered. They demonstrated its use for the enrichment of oxygen from air for various cycles and it showed good agreement with experimental data that they had been given.

The second reference to a simulator is by Chiang (1992) who describes a general purpose PSA simulator for use in the early design stage of a process. The simulator was capable of handling any process design, whilst remaining independent of the adsorption model and the solution method employed. The program used a series of simple commands in the user’s input file to sequence the valves around the columns.

The simulator could handle storage tanks for purge or blowdown gas, multiple adsorbents within a bed, as well as modelling the flow through the valves. The adsorption model used was that of isothermal Linear Driving Force (LDF), although it was noted that any current model for an adsorbent bed could be successfully incorporated into the program. This was due to the flexible solution method used, data for pressure, temperature, and concentration profiles would be loaded into an array, along with any other data for tanks etc., then integrated by a standard routine. The final data would then be used to update the system and the process repeated until cyclic steady state was achieved.

The most recent simulator to be described was that of Kumar et al (1994), who observed that all previous models in the literature were unsatisfactory for commercial purposes. However, their comments with that all academic multi-bed models simulate each bed individually seems incorrect. Their new model used a flowsheet input system, allowing complex cycles to be created relatively easily. The general purpose nature of it allowed it to have multiple multi-component feeds, parallel beds (with layers of different adsorbents), tanks, and steps. They included radial mixing, but assumed thermal equilibrium between the gas and solid phases, uniform packing, and that axial pressure changes were represented by the Ergun
equation. Additionally, they assumed there was negligible dispersion of heat and mass, and that the ideal gas laws applied throughout.

To reduce computation time, all beds were considered to behave identically, requiring only one to be calculated. They noted that this may not accurately model the transient behaviour of the process, however the speed of calculating the final cyclic steady state was considered to be the most important criterion. Even so the computation time for one cycle using a DEC 5000 workstation was estimated to be 20s for N₂ PSA, 70s for O₂ VSA, and 100s for H₂ PSA. In the worst case it took the simulation around 200 cycles to come to cyclic steady state (for the O₂ process), requiring a run time of over 38hrs for one result. Unfortunately, they did not report exactly which assumptions were used, i.e. whether radial mixing etc. were included for all cases, and so there was no indication of how much the different assumptions affected the computation time.

The only commercially available adsorption simulation package seems to be ADSIM™, by AspenTech, which also uses a flowsheet type interface and includes models for valves, tanks, pipes etc. This is capable of modelling a wide range of adsorption processes, covering both liquid phase (ion-exchange, chromatography etc.) and gas phase (pressure swing, temperature swing, vacuum swing, reaction with adsorption etc.) adsorption processes.

It is interesting that with the rapid increase in computer performance in the last ten years that there are not more references to PSA simulators in the literature. A flexible, general purpose simulator could be of great use to the academic community, allowing new adsorption models to be tested by their incorporation within an existing and proven framework for the surrounding equipment.
2.7 Applications of PSA

This section gives a brief overview of the current major applications where PSA is being used commercially.

2.7.1 Air Separation

The separation of air into nitrogen or oxygen is covered in Section 2.3.1, whilst considering the adsorbents used.

The main advantage of the PSA process over that of cryogenic distillation comes from the size and operational requirements of the units. A cryogenic unit is normally large and complicated, a PSA unit is relatively simple and can be made at virtually any size. One advantage often overlooked for the PSA process is that simple modifications to the operating cycle can allow product and gas at different purities to be produced, depending on the current demand. This could also have implications in power consumption, as a more efficient cycle would require less wasted energy for pressurising unnecessary gas.

As well as large air separation units there exists a market for smaller scale production of nitrogen or oxygen, or oxygen enriched air. See Section 2.8.3 for economic comparisons of PSA vs. cryogenic distillation.

2.7.2 Argon Recovery

Argon has increased in importance recently as its use in the electronics and metal industries has grown. It is usually produced from cryogenic air separation units, but the higher demand has led to other sources being investigated, such as from natural gas supplies (mainly from the central US, and North Africa).

Argon recovery, as opposed to argon production from air (which is covered in Section 2.3.1.3), is often performed on the purge gas from the ammonia synthesis process. The composition of this purge gas is hydrogen, nitrogen, ammonia, and 6-12% argon. ranging in pressure from 50-1900 psia.
The exact ratio of the gases and pressure depend on the design of the ammonia plant and the hydrogen recovery unit.

This recovery process is normally performed by an all cryogenic distillation unit, however Krishnamurthy et al (1987 & 1988) introduced a part cryogenic system known as HARP (Hybrid Argon Recovery Process) which used PSA as part of the separation process. The PSA unit removes the methane and most of the nitrogen from the inlet gas, producing two streams, one containing nitrogen, argon & hydrogen and one of methane & nitrogen. A 5A zeolite is used, which adsorbs methane and nitrogen in preference to the hydrogen and argon, thus all the hydrogen leaves with the argon and the nitrogen is split between the methane and the argon & hydrogen stream.

2.7.3 Hydrogen Recovery

Hydrogen recovery was an early application of PSA technology and has been used successfully in industry for almost thirty years. Hydrogen is only very weakly adsorbed onto commercial adsorbents and the process relies entirely on the adsorption of the impurities within the hydrogen stream. Therefore, the adsorbents chosen are specific to the impurities to be removed, often hydrocarbons.

The major advance in the process was made in the late 1970's by Union Carbide, (Heck 1978), who produced the first multi-bed units (known as polybed units), rather than the four-bed systems used previously. This improved recovery by over 10% and increased the throughput of a single unit. The process produced 99.999% purity hydrogen, as opposed to the old shift, scrubbing, and methanation process which produced 95-97% hydrogen. In 1980 Heck published the results of using the polybed process to produce hydrogen commercially using a steam reformer to provide the feed gas. The process saved an estimated 500,000$/year over the parallel use of two older type four-bed systems by recovering more of the hydrogen, thus requiring a smaller steam reformer, and hence less fuel.
Heck also commented on the reliability of the process, noting that during fourteen months of use only seven hours had been lost to unplanned downtime. In fact the problems were normally due to the programmable controller failing rather than the process itself. Union Carbide have since described systems that have only failed due to the malfunction of valves, rather than a process fault such as a loss in bed capacity through contamination.

2.7.4 Paraffin Separation

The separation of n-paraffins was also an early application of the PSA process and is interesting at it is one of the rare cases of the more strongly retained component being the desired product.

Union Carbide introduced the Isosiv process to separate the n-alkanes (C3 -C5) from branched and cyclic paraffins. Avery & Lee (1962), working for Linde in the US, published work on a two-bed unit. The process works by allowing normal paraffins to adsorb and the larger cyclic/branched components exit in the product stream. A desorption step is then used to collect the normals which are the main product. The recovery and quality was over 95% for both desired products. It was suggested that the normal paraffin could be used for fuel, raw materials, cracking stocks, or in many other applications. The cyclic & branched paraffins would be put into petrol or as higher quality feed for a reformer.

Cusher et al, 1990, showed how the Isosiv process, used after an C5-C6 isomerisation reactor, could be used to recover the normal paraffins for recycle into the feed stream, thus improving both the yield and the quality of the product. With less normal paraffins in the product it would have a higher octane rating and require less octane improving additives for use in petrol.
2.7.5 Methane Recovery

Methane is known as natural gas to most consumers, and in this form is a relatively pure fuel of over 90% methane, the rest being mainly carbon dioxide and ethane, together with some hydrogen and water. However, the raw gas is often obtained from natural gas reservoirs, oil wells, and landfill sites which contain between 20-80% carbon dioxide. By separating out the methane product quality gas can be produced, or by enrichment lower quality fuel gas can be created. The separation can be effected by either kinetic or equilibrium based processes.

Air Products (Kumar & van Sloun, 1989) developed a PSA process that will remove all impurities to below 1ppm from a methane & carbon dioxide stream, with 96% recovery at 90% methane product purity. A thermal swing and a pressure swing are combined within the process, the thermal swing step removing any impurities present in the feed gas (the most strongly adsorbed components requiring the extra heat to clean the beds) and the pressure swing step produces the enriched methane product.

2.7.6 Carbon Monoxide Recovery

Carbon monoxide is a common by-product produced by many industrial processes, notably in steel production. Recovery of the carbon monoxide is a cheap method of reusing a waste product as it can then be used as a raw material or burnt as a fuel. In other circumstances, carbon monoxide is a toxic by-product that must be removed from a waste stream, e.g. from a flue gas stream. Originally wet processes were used for the recovery, now PSA systems have been introduced, often in a two stage process, Suzuki, (1988). The first stage removes the carbon dioxide, using PSA with an activated carbon adsorbent, in the second stage carbon monoxide is separated from the nitrogen, using an sodium-mordenite adsorbent.
2.7.7 Ozone Enrichment

Many processes that require ozone, such as sterilisation processes and water treatment, normally do so by creating ozone from an oxygen rich feed. The conversion process only converts around 10-15% of the feed oxygen and the output gas is normally only 1-2% ozone, and at most 6%, so valuable oxygen is lost with the ozone. By separating the ozone from the product stream unconverted oxygen can be returned to the ozone generator.

The enrichment process usually uses a silica gel or zeolite as the adsorbent, ozone being preferentially adsorbed over oxygen, nitrogen, or some other carrier gas. Kiffer (1959) describes a three bed PSA process for oxygen recovery when it is contained in a carrier gas. When a pure oxygen feed has been used the process is simpler, requiring only a two-bed system.

2.7.8 Air Drying

A large application of PSA is in air drying. Often the process is a TSA system, requiring additional external heating. For the case where no external heat is applied the process is often called a ‘heatless drier’. As well as the removal of water from an air stream, the process will often remove other trace contaminants that are normally present in air, such as carbon dioxide and hydrocarbons.

The use of PSA air driers ranges from large units on the front of cryogenic air separation units, often using a TSA system although some true PSA systems have been installed more recently, to small commercial units providing a small flow of clean, dry air for instrument air. Details of a typical TSA air drier are given by Ruthven (1984) and for a PSA air drier in Ruthven et al (1994).
2.7.9 Other Applications of PSA

2.7.9.1 Hydrogen Isotope Enrichment

A minor use for PSA is in the enrichment of hydrogen isotopes, mainly deuterium enrichment from a hydrogen stream. Separating isotopes is one of the most demanding separation duties in use today. Weaver & Hamrin (1974) used a two-bed system, containing palladium deposited on alumina as the adsorbent, with a direct purge cycle. They enriched a 5.5% deuterium in hydrogen stream to give 17.5% deuterium in the product stream.

In addition, they modelled the system, getting reasonable results when compared against the experimental data. The match was better when the process was operated at longer cycle times, which may have been due to the ILE assumption they used in the model.

2.7.9.2 Other Uses

Other areas where the PSA process has been used are saturated-unsaturated hydrocarbon separation (propane-propene etc) and acetylene-ethene separation. It has also been used to remove nitrous oxide compounds from waste gases.

2.8 Economic Aspects

2.8.1 Background

There is very little literature on the economics of the PSA process, as exact costs are normally a closely guarded industrial secret. Allowing competitors to know the true cost of a process can be very revealing as to the relative advantages of their own processes. Prices for contracts can often be misleading as some may be for supplying the plant only, others will include the costs of running the plant for the customer, and some will include the
price of the plant within the gas price. Some customers prefer simply to buy the product gas and have no interest in the actual process used to provide it.

2.8.2 General PSA Processes

For a PSA process the main energy cost is from the requirement to increase the pressure of the feed gas to the upper operating pressure. The depressurisation energy is normally lost, although depressurising one bed into another to raise its pressure can recover some of this energy. Industrial PSA cycles normally include at least one of these pressure equalisation steps, and larger multi-bed plants will have several, often at several different pressures, to minimise losses.

2.8.3 Air Separation

The largest air separation units are cryogenic, operating at low temperatures and elevated pressures. They normally consist of three distillation columns (one each for nitrogen, oxygen, and argon), depending on the customer requirements. The main operating cost is the energy required to compress the feed air.

The air must be cleaned of water and carbon dioxide which would freeze out at the cryogenic temperatures used. Other impurities in the air must also be removed as these will often collect in the products, some can be ignored but others can prove dangerous e.g. hydrocarbons will collect in the oxygen reboiler and are a safety hazard. It is interesting that, although the separation of the air is done by distillation, the cleaning of the air is performed by a PSA or TSA system, and is the current technology used by all the major air separation companies.

If liquid gas products are required as a product then, depending on the relative amounts required, these can either be removed from the cryogenic plant directly as a liquid or taken as a gas which is then passed through a liquefaction unit. Because of the high capital cost of building cryogenic plants
the PSA process becomes more viable at lower tonnages and purities, however, as a PSA unit cannot produce liquid gas directly then it becomes less viable economically as a liquefaction unit must always be added.

2.9 Control

Control is a commonly used idea and has been applied to almost all machinery. There are two main types of control, manual and automatic, where manual control requires human input and automatic requires no human input (beyond that of setting up the control system).

There are also two main distinctions that can be made in the type of control being used on a process, namely,

i) feedback - where the desired output variable is measured and used to adjust the value of the manipulated variable

ii) feedforward - where an input variable is measured and used to adjust the value of the manipulated variable.

This literature review will deal mainly with automatic control in the above two cases.

2.9.1 History of Control

The earliest examples of control were the flow rate of water to regulate a water clock and to keep the level in a vessel constant in ancient times, as noted by Mayr (1970). The method developed was to use a float mechanism to adjust the flow rate, the same method is commonly used today in a cistern.

The most famous problem in control history was the search for a method of controlling the rotation speed of a shaft, for example to control the speed of the grinding stone in a mill. The fly-ball governor was the most promising candidate, however it was James Watt and the steam engine that made the fly-ball governor famous. When this system is analysed it is found that it is an example of offset control - to maintain a constant speed the position of the governor must be different to that under normal conditions,
i.e. the offset is needed to keep the controlled variable at its different position.

The main problem with feedback control is the possibility of instability occurring. As the errors are reduced by increasing the feedback (or gain) of the loop then the higher the feedback the better the performance. At some point feedback becomes so high that it becomes unstable, in an electronic amplifier this is often characterised by a squealing noise. Nyquist (1932) showed that the stability of a system could be analysed from a graphical plot of loop-frequency response. Bode (1945) extended this further into a method which is still used today.

Possibly the most important control development was that of proportional-integral-derivative (PID) control, first published by Callender et al. (1936), based on experimental work and linearised system dynamics.

Work on control developed much faster from the Second World War onwards, for example the development of the root-locus method for design or stability analysis by Evans (1948). Later the use of the ordinary differential equation (ODE) to model control systems became more common, leading to the modern day method of working directly with the ODE in normal or state forms, a method which requires the use of a computer to give a solution, with the exception of the linear cases.

2.9.2 General Methods of Control

Since the development of the PID control system this has been the mainstay of the control world. It is simple and effective, and, as long as a stability analysis has been performed on the parameters, it should be possible to keep the system from going unstable. This section will look at the PID control, in parts and as a whole.
2.9.2.1 Development of PID Control

The earliest publication of the proportional-integral-derivative (PID) controller was by Callender et al. (1936) whose work relied on experimental work and simplified linearised approximations of system dynamics.

This work was followed by the development of standard, easily performed experimental methods that would give an acceptable tuning of the PID controller. Ziegler & Nichols (1942,1943) gave two methods for tuning the controller, one based on a decay ratio and the other on stability limits. The settings given by the rules provide a good starting point for the tuning constants of a controller which are then normally finely adjusted in use.

PID control is a widely researched and studied area and there are many publications and books that deal in depth with PID control, for example Luyben (1989), Franklin et al (1994), Smith & Corripio (1985).

2.9.3 Multiple Variable Control

PID controllers are simple feedback controls commonly used in industry. When they were invented there were no computers or digital controllers as we know them today and all the control was done using pneumatic systems. In addition, these were single loop controllers, one controller looked after one variable alone and no account was taken of the effects on any other part of the process.

A more recent alternative is to use a multiple variable control scheme. A digital computer is normally used to perform the more advanced calculations required. Often a vector of variables is used, combined with a control function algorithm, to produce a vector output that gives the best overall set point for the controllers. If properly implemented this can produce a very powerful and robust control system.
2.9.4 Model-Based Control

A model-based control scheme incorporates a model of the process into the control algorithms. This allows non-linear process responses to be predicted and allowed for.

Model-based control schemes normally rely on digital computers to perform the extra computation that is required. Depending on the complexity of the model used this computation load may be small or very large. The complexity of the model is limited by the economic cost of a computer powerful enough to run it. A model that requires a supercomputer is unlikely to be used on a real plant, however accurate it may be.

The way to overcome this is to reduce the complexity of the model to a manageable size. However, this results in other sources of error due to the approximations and the result is two extra sources of error that can be considered important:

1) The model of the process may not be accurate enough, e.g. they are of a different nature such as one being linear, the other non-linear. This is known as 'structural error'.

2) The basic parameters used in the model are not the same as the real ones, possibly due to the difficulty of measurement. This is known as 'parametrical error'.

Either or both of these types of error may be present and can cause the model to predict differently from the real process. In addition, other factors come into effect with the model-based control schemes that stem from the above two problems (Flathouse & Riggs, 1996):

3) The model parameter adjustments must become smaller as the nature of the model and process start to become larger, i.e. the larger the model the more unstable it becomes, thus limiting the parameter adjustment to only small changes.
4) The measurement dead time within the system becomes important to keeping the model stable and following the plant, even with slow response times from the plant, e.g. a slow responding plant needs a slow responding model otherwise the model parameters can become unstable.

5) An inverse response or higher order dynamics can upset the stability of the model controller, limiting the parameter adjustment to only small changes.

2.9.4.1 Generic Model Control

The idea of generic model control was first published by Lee & Sullivan in 1988. They proposed a method for incorporating a model into a generic structure that, when reduced, would give the basic control schemes, such as PI etc. This would allow easy addition of a model to a controller if required. It also included model parameter adjustment from steady state values.

This work was followed by the incorporation of a method for continuously updating the model parameters to fine tune the model whilst it was not at steady state, Signal & Lee, 1992. This helped reduce errors as a result of either the parametrical or structural errors in the model, and as it did not require a steady state to make the corrections it was a more robust method.

Others have considered this problem, Flathouse & Riggs, 1996, for example have proposed an alternative method for correcting the model parameters, using disturbances to measure the response of the process. The response is then used to correct the parameters in the model.
2.9.4.2 Model-based Observer Control

Model-based observer control requires a true model of the process, as opposed to a simplified model, that can predict the behaviour of the plant and provide valuable extra information. The information is considered valuable when it is difficult or expensive to measure it directly on the real plant but is readily available as an output from a model. Examples include concentration profiles or rates of reaction throughout a vessel. The availability of this information can influence the structure of the control system, enabling it to base its actions on the most appropriate parameters rather than just those values that can be measured directly.

Gawthrop & Ponton (1996) used a model to improve process control, demonstrating its effectiveness with a simple non-linear example. They commented that the method of tuning the controller and the model were not, in principle, that difficult.

By using the information the model predicted to help control the process a more stable operation was achieved. However, they also noted that if the model predictions should be incorrect then the control could be more unstable than without the model. In addition to structural and parametrical errors they also stated two further causes of error:

1) initial state may not be the same as the plant,
2) disturbances may enter the process and so cause the model to give an incorrect prediction,

2.9.5 Control of PSA Processes

There are very few references in the literature on the subject of control and dynamics of the PSA process. The performance of PSA processes are known to suffer when adsorbent capacity is reduced, or when operating or ambient conditions change. In these cases, by measuring the composition of the product, it may be possible to alter the cycle steps in order to maintain
the desired separation. However, this method requires more expensive equipment than may be justified and will not reject transient disturbances.

One paper that deals with a method of control is that of Matz & Knaebel (1987), who suggest using temperature as the measured variable, thermocouples being reliable, accurate and inexpensive. The method relied on the heat of adsorption, as the nitrogen rich phase adsorbs in the bed the heat of adsorption released is higher than that of the oxygen rich phase. If the temperature and concentration fronts coincide, or move with a fixed offset, it should be possible to measure the temperature profile within the bed and predict the position of the concentration front.

It was shown that the temperature and concentration fronts were coincident and so the method could be used to give the position of the concentration front. However, several practical difficulties were recognised, such as the noise in the temperature data obscuring the effect of the concentration front. This mainly occurred in the purge step, as the temperature change due to the front was only a fraction of one degree Celsius. The conclusion from this work is that the use of temperature to control the PSA process is possible if noise can be reduced and the thermal and concentration fronts are coincident. No reference could be found to a commercial use of this type of control, although such control schemes often remain commercially secret for many years.

2.10 Concluding Comments

The simplicity of the PSA process has contributed to its rapid success in many areas of gas separation. Often the long operating life of the adsorbents makes it very attractive to high capacity separations, failure usually occurs due to physical damage rather than loss of adsorption capacity, resulting in a reasonably safe process.

There is only one reference to the control of a PSA process within the literature, that of Matz & Knaebel (1987) who proposed the use of
temperature front measurement to control the process. However, for this to work there must be a stable temperature front that moves in parallel to the concentration front.

Various model-based control schemes are proposed in the literature and some are now used in the process industries as more powerful computers allow the solution in real time of more complicated process models.

The use of generic model control is limited by the fact that a slowly changing process must have a slowly changing model. The generic model control scheme, Lee & Sullivan (1988), acts on more rapidly changing processes, but does not appear to have been applied to slow changing ones, and it relies on a rapid response, much as a simple PID controller. The generic model controller replaces the basic model in a PID controller with a more complicated one.

The most promising general method for control general PSA processes, i.e. from bulk separation to trace gas removal, is that of model-based observer control, Gawthrop & Ponton (1996). This relies on a process model to give critical control information which, under normal circumstances, would be impractical to measure. This information, such as concentration front positions, may then be used to control the PSA process in a more stable way.
Chapter 3
Theory

3.1 Introduction

This chapter includes a summary of the instantaneous local equilibrium (ILE) model, originally from Flores-Fernandez (1978), added to by Kirkby (1984) and Espitalier-Noel (1988). The theoretical basis for the model remains the same as before, as does the method used for solving the equations. However, the actual solution of the PSA system has been changed significantly, now including the surrounding equipment rather than just the adsorbent beds. The method of solution of the system can be found in Chapter 5, the theoretical aspects of the model are discussed in more detail within this chapter.

3.2 Assumptions

The assumptions used in the model are stated below. They are more fully described and discussed in Chapter 2, Section 2.5.1, with references to the literature.

1) The feed gas is comprised of 20.95% oxygen, 0.93% argon and 78.12% nitrogen (on a volumetric basis), of which oxygen and argon behave as a single component, hence it is assumed that $y_{O_2\text{in}} = 21.88\%$ (by volume).
2) The ideal gas laws apply throughout the system, $PV=nRT$ at all times.
3) There are no temperature changes within the system, which operates at 293K (although this temperature is an input to the program).
4) Instantaneous Local Equilibrium (ILE) is established at all times within the adsorbent bed.
5) No axial or radial dispersion occurs within the bed, i.e. the bed is plug flow and there are no wall effects.
6) There is no significant axial or radial pressure drop within the bed, the pressure within the beds is assumed to be a function of time only.
7) The beds are uniformly filled with identical adsorbent particles, leading to an even distribution of adsorbent capacity within the bed.
8) There is no pressure difference across any object in the system, except for the valves.
9) Linear, independent isotherms of the form:

\[ q_o = q_{o_0} + K_o p_o = q_{o_0} + K_o p y_o \]  \hspace{1cm} (3-1)

\[ q_N = q_{N_0} + K_N p_N = q_{N_0} + K_N p y_N \]  \hspace{1cm} (3-2)

can be used to represent the adsorption isotherms of the two components. Note that the form of these equations are such that the isotherm does not necessarily pass through the origin. This limits the applicability of the model to the area where the isotherms have been linearised.

3.3 Adsorbent Bed Equations

3.3.1 Introduction

This section deals with the equations used to solve the adsorbent bed. The most common method of analysis for a piece of equipment is to write the component and total mass balance equations.

3.3.2 Mass Balance Equations

In a gas phase system where adsorption occurs the accumulation term can be split into two components, for the gas phase and the adsorbed phase, giving:

\[ \text{INPUT} = \text{OUTPUT} + \text{GAS ACCUMULATION} + \text{ADSORBED ACCUMULATION} \]
For a differential element within the bed it can be written:

\[
\frac{AP}{RT} \frac{\partial (uv_y)}{\partial z} = \frac{AP}{RT} \frac{\partial (uv_y)}{\partial z} + \frac{AP}{RT} \frac{\partial (uv_y)}{\partial z} \delta z + \frac{\epsilon A}{RT} \frac{\partial (Py_o)}{\partial t} + \omega A \delta z \frac{\partial (q_o)}{\partial t}
\]

This can be rearranged to give:

\[
\frac{-\partial (uv_y)}{\partial z} = \frac{1}{P} (e + \omega RTK_o) \frac{\partial (Py_o)}{\partial t}
\] (3-4)

By defining an adsorption parameter \( k_o = e + \omega RTK_o \) Equation 3.4 can be simplified to

\[
\frac{-\partial (uv_y)}{\partial z} = \frac{k_o}{P} \frac{\partial (Py_o)}{\partial t}
\] (3-5)

This equation gives the velocity and composition of the gas phase as a function of both time and position in the bed. Similarly for the nitrogen, component N:

\[
\frac{-\partial (uv_N)}{\partial z} = \frac{k_N}{P} \frac{\partial (Py_N)}{\partial t}
\] (3-6)

If a rearrangement is made to Equation 3.6, using \( y_o = 1 - y_N \), it is possible to eliminate the derivative of \( Py_o \) with time. Combining 3.5 and 3.6 in this way gives:

\[
(k_N - k_o) \frac{\partial (uv_y)}{\partial z} = \frac{k_o k_N}{P} \frac{\partial P}{\partial t} + \frac{\partial v}{\partial z} k_o
\] (3-7)

By introducing two simplifying equations:

\[
p' = \frac{1}{P} \frac{\partial P}{\partial t}
\] (3-8)
\[ Y = (k_N - k_O) Y_0 + k_O \]  \hspace{1cm} (3-9)

where \( Y \) is known as the 'augmented mole fraction' it is possible to write Equation 3.7 as:
\[
\frac{\partial (vY)}{\partial z} = -k_O k_N p' \]  \hspace{1cm} (3-10)

which can be integrated with respect to distance, \( z \), as \( p' \) is a function of time only, which yields:
\[
vY = -k_O k_N p'/z + f(t) \]  \hspace{1cm} (3-11)

where \( f(t) \) is a function of time only and can be determined from the boundary conditions at either end of the bed. This gives two possibilities: at \( z = 0 \)
\[ f(t) = v_0 Y_0 \]  \hspace{1cm} (3-12)

and at \( z = L \)
\[ f(t) = v_L Y_L + k_O k_N p'/L \]  \hspace{1cm} (3-13)

This leaves two solutions to Equation 3.11, depending on which end of the bed has known boundary conditions for the current time: for \( z = 0 \),
\[
vY = v_0 Y_0 - k_O k_N p'/z \]  \hspace{1cm} (3-14)

and for \( z = L \),
\[
vY = v_L Y_L - k_O k_N p'(L-z) \]  \hspace{1cm} (3-15)
Equations 3.14 and 3.15 relate the volumetric flow rate to the local composition and the flow rate and composition at some known position along the bed. To calculate the composition the material balances are decomposed using the Method of Characteristics, as follows.

### 3.3.3 Solution by the Method of Characteristics

Equations 3.14 and 3.15 are the overall mass balance equations for the bed and relate the composition of the bed at a point to the gas velocity at that position, relative to some end condition. To solve for the velocity and composition of the gas changes with respect to position the quasi-linear partial differential Equations 3.5 and 3.6 are used. By performing the operation

Equation 3.5 * \( y_N \) - Equation 3.6 * \( y_o \)

and substituting \( \partial y_N \) with \( -\partial y_o \), (obtained by the differentiation of \( y_N=1-y_o \)), hence:

\[
\frac{\partial Y}{\partial t} + \frac{u}{Y} \frac{\partial Y}{\partial z} = -(Y-k_O)(Y-k_N)\frac{D'}{Y} \quad (3-16)
\]

An equation for the total derivative can be written as:

\[
\frac{\partial Y}{\partial t} + V \frac{\partial Y}{\partial z} = \frac{DY}{Dt} \quad (3-17)
\]

By comparison of Equation 3.16 with Equation 3.17, \( V \) can be defined as:

\[
V = \frac{u}{Y} = \frac{dz}{dt} \quad (3-18)
\]

and is termed the characteristic ground velocity, whereas the change in augmented mole fraction along the trajectory can be written as:
\[
\frac{DY}{Dt} = -(Y - k_O)(Y - k_N) \frac{P'}{Y}
\]  

(3-19)

By including Equations 3.14 and 3.15 in Equation 3.19, then the following alternative equations for the characteristic velocity are the result:

\[
\frac{dz}{dt} = \frac{(v_0 Y_0 + k_O k_N p' z)}{Y^2}
\]  

(3-20)

and for the second condition:

\[
\frac{dz}{dt} = \frac{(v_L Y_L + k_O k_N p'(L - z))}{Y^2}
\]  

(3-21)

By using Equation 3.19, in combination with either Equation 3.20 or 3.21, then the change of both position and composition, with respect to time, can be calculated.

Equations 3.20 or 3.21 can often be simplified, depending on the step type, to give a specific equation for the step, e.g. in the case of pressurisation, where \(v_L\) can be taken as being zero, i.e. there is no dead space at the product end of the bed, then Equation 3.16 can be simplified to:

\[
\frac{dz}{dt} = \frac{k_O k_N p'(L - z)}{Y^2}
\]  

(3-22)

which requires slightly less computation. However a test now needs to be applied to see if this condition is valid. This test can consume the same or more time than evaluating the removed part of Equation 3.20.
3.3.4 Interpretation of Characteristics

The Method of Characteristics was described in detail by Kirkby (1984) and so this section reviews the terms that will be used later, when 3D characteristic plots are presented.

Characteristics must be interpreted carefully, in general they can either diverge, converge, intersect, or remain parallel. In the case of an intersection both the characteristics terminate, at which point there becomes two boundary compositions at the same position within the bed, giving an infinite rate of change of composition with respect to distance. This case is known as a shock wave. Other further characteristics that collide with this shock wave terminate on impact. Kirkby (1984) showed that the speed of the shock wave in this model was the geometric mean of the velocities of the characteristics converging at that point.

In the case of air separation to produce oxygen, shock waves are a result of the oxygen phase (less adsorbable gas) being displaced by the nitrogen (more adsorbable gas). As the amount of nitrogen adsorbed is greater than that of oxygen, the oxygen concentration is enriched in the gas phase, however, oxygen enriched gas travels more slowly than nitrogen rich gas. If it is assumed that the equilibrium is instantaneous, and there is no axial diffusion, then a region of rapidly changing concentration can sharpen into a discontinuity, at which point the two zones meet and a shock wave is formed.

The opposite effect is where the characteristics are diverging, here a steep concentration profile will tend to spread out as the nitrogen rich region moves away from the oxygen rich region, creating what is known as a simple wave, with a range of concentrations between the two regions.

It is important to note that in the model of Kirkby (1984) the shock waves would normally be formed in pressurisation and product release steps, and simple waves would be produced in the depressurisation and purge steps. In the new simulation, described in Chapter 4, it is possible for shock
and simple waves to occur in any pressure changing step, depending on the
gas concentrations being fed into the beds.

3.3.5 Summary

The derivation of the equations that describe an adsorbent bed, with
the assumptions of it being isothermal, isobaric, and ILE, has been described.
The use of the Method of Characteristics was described in detail, along with
the interpretation of the characteristics. Chapter 4 will show how this
solution method can be incorporated into a computer model.
Chapter 4
Process Simulation and Sequencing Control

4.1 Introduction

This chapter describes the computer program used to operate the PSA pilot plant and the specific model which is used in the simulations.

The program to control a process plant for a single set of operating conditions is relatively simple to code. Creating a program that gives flexibility in the type of operation to be performed involves much more work. Adding the need to save data at regular intervals creates further complications.

The important point of the ILE model is that, although not the most accurate due to its assumptions, it can be solved very rapidly, fast enough to be used in a real-time control application and its parameters can be tuned such that its output matches that of the plant. A slower model may be more accurate but is of little use if it cannot keep up with the process. Again, the model must be adequate for the use, in this case a model that cannot be solved in real-time on a reasonably cheap personal computer would be considered inadequate, however accurate the final result may be.

4.2 The Original ILE Model

Flores-Fernandez (1978), Kirkby (1984), and Espitalier-Noel (1988) describe the structure of the ILE model on which this project is based. Block diagrams were included by the latter two to show how the program operated and the function of the various procedures used. In the case of Espitalier-Noel, further operational modes were added, based on using direct purge gas and waste gas recycle in various ways.

A description is given in Chapter 3, Section 3 of how the ILE equations are developed and how the Method of Characteristics is used to solve them.
4.2.1 Limitations

The model referred to above are limited by several fundamental factors, although, like all models, they have been improved over time. The main ones were:

1) Each step type had a separate set of boundary conditions, new step types required new subroutines to be written, leading to the possibility of new errors being introduced.
2) No account was taken of the effects from any equipment within the system, apart from the adsorbent beds.
3) There was only one bed, the other bed was assumed to behave identically but operate half a cycle out of phase. There was no possibility of asymmetric operation.
4) There were no time delays built in during the purge step when gas is being transferred from one bed to the other, or any other step where gas was being passed from one unit to another.

4.2.2 Features

Although there were limitations to the model there were several features that are worthy of note:

1) Purge gas could be supplied directly from one bed to the other or via the product receiver, simulating either direct purge or well-mixed purge gas. Direct purge gas had not been modelled before, although it was in common use in PSA plants.
2) The recycle of waste gas into the feed stream could be simulated, although it required iteration to find the amount of feed air to be supplied. The waste gas from the depressurisation and purge steps was mixed, removing any concentration change with time.
3) Purge snatching could be simulated, where gas leaving the bed in the waste line during the purge step could be recycled into the feed line, in this case the gas was not mixed and the concentration profiles were kept.
4.3 The New ILE model

In the new version of the model, which is still based on the same underlying assumptions for the adsorbent beds, a number of important changes have been made.

1) The model simulates, using simplified equations, the surrounding process equipment. This is intended to make the model both more general purpose, as proposed by Chiang (1993), and more realistic.

2) By using a valve equation at either end of the bed, relating flow rate to the pressure drop across the valves, the need for separate boundary conditions for each step is removed and all steps can be solved using the same numerical procedure, based on Equation 3.19 combined with either Equation 3.20 or 3.21. If a new step is created then there is no need to change the program code to simulate it (as shown in the case of equation 3.22 for the depressurisation step), it being simply a matter of adjusting the valve timings as on a real plant. However, the major drawback is that this results in the flow in and out of the beds being known and the change in pressure being the unknown, the opposite case to the solution algorithm used in the original ILE model. The change in pressure must now be solved by an iterative process, which results in more computation time being required.

3) The final major feature of the new model is that it was programmed in a style known as an 'object-oriented' approach, which is described in Appendix 1. All the equipment objects are combined into 'groups' of objects (separated by the valves), which are then combined into a system for simulation. This provides a more robust program and a simpler environment for adding more varied types of equipment.
4.4 Categorisation of Equipment

The equipment within a PSA plant can be classified, in a simplistic manner, into six basic types. These types have many common properties, but in addition have extra, specific, properties that make each of them a unique type. The six types are:

I - Valve
An object that can have a different pressure on either side of it.

II - Empty
An object that contains no adsorbing material and acts as a well-mixed vessel or a plug-flow device. Normally used as a receiver, a pipe join, or a pipe itself.

III - Bed
An object that is an adsorbent bed. It has uniform axial and radial pressure and allows concentration changes to occur both across and within it. Uses the ILE model and the Method of Characteristics as its solution method.

IV - In/out
An object that is either an inlet (source) or an outlet (sink) from the modelled system. It adjusts its flow to keep the pressure in its group constant, i.e. any group with a Type IV object keeps its pressure at the initial value.

V - Instrument
An object that takes a reading of another object's properties and performs various actions with the data.

VI - Sequencer/Event Manager
Objects that directly manipulate other object's properties as a function of time. The sequencer operates on a cyclic basis, the event manager deals with discrete events, such as the introduction and removal of a disturbance.

The major properties of these basic types are summarised in Table 4.1. Some properties have no meaning for different types of objects, e.g. there is
no volume for an instrument. Object properties are typically identical to the real-life properties of the item.

<table>
<thead>
<tr>
<th>Volume</th>
<th>ΔP</th>
<th>ΔC</th>
<th>Adsorbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve</td>
<td>no</td>
<td>yes</td>
<td>yes*</td>
</tr>
<tr>
<td>Empty</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Bed</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>In/out</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Instrument</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Sequencer / Event Manager</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

* when shut.

where \( \Delta P \) = pressure difference can exist across the object,
and \( \Delta C \) = concentration difference can exist across the object

Table 4.1 - Basic Properties of Objects

There is no limit to the number of each type of object that may be incorporated into a system for simulation, except that imposed by necessity due to having to size the arrays when compiling the program.

4.4.1 Type I Objects - Valves

The final control element in most control systems is a valve, which is normally used to manipulate flow and hence indirectly a process variable such as pressure. In the simulation all valves are represented by type I objects. Currently only a valve and a compressor fall into this grouping.

Type I objects exist with a pressure drop across them, i.e. the pressures at the two outlets may not necessarily be the same. As in the case of real valves a pressure drop normally exists across them and this is used
to calculate the flow rate passing through the valve (if it has a set point
greater than zero). A Type I object never has a volume and simply passes
concentration profiles through without modification, except when closed,
when a different concentration can be maintained on both sides.

There are four basic types of valve behaviour included in the model.
These behave identically, except for the way that flow rate varies with set
point.

Note: the units of $F$ in Equations 4-1 to 4-4 are mol.s$^{-1}$, rather than m$^3$.s$^{-1}$ as
is the norm, this was done for speed of calculation. This causes the valve
equation to predict slightly different flows to that from the normal valve
equations. The types are:

i) Linear

$$F = C_{v_{\text{max}}} X_u \sqrt{\frac{\Delta P}{G}} \quad (4-1)$$

If the $\Delta P$ across this valve stays constant then the flow rate will vary linearly
with the valve position.

ii) Square Root

$$F = C_{v_{\text{max}}} \sqrt{X_u} \sqrt{\frac{\Delta P}{G}} \quad (4-2)$$

If the $\Delta P$ across this valve stays constant then the flow rate will vary
with the square root of the valve position.

iii) Equal Percent

$$F = (C_{v_{\text{max}}}) \left(\frac{x_{v_{\text{max}}}}{X_u}\right) \sqrt{\frac{\Delta P}{G}} \quad (4-3)$$

If the $\Delta P$ across this valve stays constant then the increase in the flow
rate will become higher as the valve position gets closer to 1. True equal
percent valves will not shut and so are normally manufactured as a hybrid, having a linear region in the lowest 5% of their range. This has also been programmed into the model as valve positions can often become very small.

iv) Constant Flow

\[ F = X_v K_{\text{max}} \]  

The flow rate through this valve is independent of \( \Delta P \) across the valve and it varies only by changes in the set point. This is used to provide a constant flow rate through the valve regardless of pressure difference and can be considered as a molar flow controller (analogous to a mass flow controller).

No assumptions are made as to how a valve characteristic is modified by the surrounding resistances, i.e. the installed characteristics of all the valves are assumed to be the same as their isolated characteristics. As there is no pressure drop in any other types of equipment the installed valve characteristic will be the same as when it is in isolation. Normally an equal percent valve would be chosen with an \( \alpha \) such that the installed characteristic is approximately linear.

The other type of object included in this category is a compressor. This is a special case of a valve object that allows the pressure to be higher on one side of the object than the other, but may be located within a group. Adding a compressor to a group partially removes the zero \( dP/dz \) assumption, the group is split into a higher and a lower pressure sub-group, each of which still maintains a zero \( dP/dz \).

4.4.2 Type II Objects - Empty

The type II objects are effectively empty volumes, i.e. they contain no adsorbent material but have a fixed volume. There are four different pieces of Type II equipment programmed into the model. Each has at least two
connections, the exact number depends on what type of object it is. Flow can occur in either direction at any connection, this depends purely on the flow conditions at the time. No internal diffusion process is assumed for any of these objects.

i) Pipe - can be selected as being either a plug-flow or a well-mixed device (useful for very short pipes), it has only two connections.

ii) Receiver - behaves as a well-mixed vessel. It is used to mix the gas in a large volume, it has only two connections.

iii) Tee - effectively this is identical to the Receiver, but has three connections rather than two.

iv) Fourway - effectively this is identical to the Receiver, but has four connections rather than two.

A Type II object has a fixed volume and, for the tee, receiver, and fourway, always adjusts its concentration according to the simple component mass balance:

\[ C_{\text{out}} n_{\text{out}} = C_{\text{in}} n_{\text{in}} - \frac{d}{dt} C_{\text{out}} \frac{PV}{RT} \]  \hspace{1cm} (4-5)

In the case of the Pipe, it can be modelled as above or as a plug flow device, in which case concentration gradients can exist within it. For the plug flow case the composition profile is always retained as the gas passes through the pipe, and is compressed or expanded as the pressure changes.

4.4.3 Type III - Bed

The type III objects are the adsorbent beds. There is currently only one implementation of a type III object and this is an ILE assumption model, as described by Kirkby (1983) and Espitalier-Noel (1988). The major benefit of the ILE model is that it can be solved very quickly due to its simplifying assumptions. This solution time is important if the model is to be used for on-
line control work. See Section 3 for the theory and Section 4.6 for the solution method of the ILE model.

4.4.4 Type IV - In/out

These objects perform two separate but closely related tasks. They control the flow in or out of the defined PSA process to the outside of the system and are the only place that gas can enter or leave the whole system.

Any group with a type IV object always has dP/dt = 0, thus keeping the pressure within that group constant at the starting value. They can operate in one of two modes:

i) A Feed object is given the opportunity to match the flow leaving through the valves from a process group by adjusting its flow rate, i.e. it acts as a source of gas into the process. The gas concentration is kept constant at the value at the start of the program, however there is no reason why a change in feed gas concentration could not be programmed in if desired.

ii) A Product object is similar to a Feed but is given the opportunity to match the flow out of the object attached to it, by adjusting its flow to be exactly that leaving the object, i.e. it behaves as a sink for gas leaving the process. The Product takes the concentration of the gas leaving the adjacent object as being its own value.

4.4.5 Type V - Instrument

A type V object takes data from another object of any type and performs some action with it. Currently an instrument serves one of two purposes:

i) Recorder

A recorder instrument simply records a property, such as flow rate, and saves it to a file or displays it on-screen at specified time intervals. The
data may be recorded with the default units for that property or any other unit that has been pre-defined. The units available are shown in Table 4-2.

<table>
<thead>
<tr>
<th>Default Unit</th>
<th>Other Units Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composition</td>
<td>mole frac (component A)</td>
</tr>
<tr>
<td></td>
<td>mole frac (component B)</td>
</tr>
<tr>
<td>Pressure</td>
<td>Pa</td>
</tr>
<tr>
<td></td>
<td>bar, atm, psig</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>kg.mol.s⁻¹</td>
</tr>
<tr>
<td></td>
<td>¹³.s⁻¹ @ STP, m³s⁻¹ @ STP</td>
</tr>
</tbody>
</table>

Table 4.2 - Recorder Units

A flow recorder can also act in one of three modes:

a) flow / second - the flow recorder saves the current flow rate at the point, as a flow per second, with the sign indicating the direction of flow.

b) total flowed - the flow recorder integrates the amount of gas that has flowed past at that point and displays it as a total. If the flow direction is reversed then the total flowed will be reduced.

c) absolute total flowed - the flow recorder behaves in the same manner as the b) mode, however it takes no account of direction, just the total amount of gas that passed by it.

ii) Controller

A controller acts by taking a reading from an object and passing a new set point to another object, usually a valve. The controller can currently act in one of two modes:

a) Proportional

The controller acts as a simple proportional controller with a constant gain. Output is defined as being, in the case of a pressure controller:
The source of \( P \) is defined for each controller in the main input file, by giving the name of the object and connection point to read it from.

b) Proportional & Integral

The controller acts as a simple proportional controller but with the addition of an integral action in order to help eliminate offset error. Gain is still a constant. Output is defined, in the case of a pressure controller, as:

\[
X_v = X_v^0 + k_c (P_{sp} - P) + \frac{1}{T} \int (P_{sp} - P) dt
\]  

(4-7)

Although no facility was included in the current program to reset the integral term to zero, this would be simple to implement if required.

4.4.6 Type VI - Sequencer/Event Manager

A type VI object acts by passing information to other objects depending on the current process time. It can adjust the position of valves and the set points of controllers. For convenience it also has the ability to change the operating mode of a valve, i.e. from mode I to mode iv. The two types are:

i) Sequencer - this works on a cycle basis. It uses a set of defined actions, built up to form a sequence of operations, which are then repeated in a cyclic manner.

ii) Event Manager - this creates specific individual events, such as introducing disturbances into the process. Each event is described explicitly and discretely, i.e. there is no way to remove a disturbance except by using another event.
4.5 Model Structure

This section deals specifically with the internal structure of the model. It is possible for the plant layout to be easily changed, e.g. a pipe to be changed to a receiver by altering the input description file (a plain text file that provides the plant layout and starting conditions). To enable this flexibility, without any of the model code needing to be rewritten, requires a specific structure to exist within the model. Most previous models had the process layout hard coded within, whereas this model is similar to that described by Chiang (1992), who created a model that was more of a process simulator in that it was able to cope with any process design, adsorption model, and numerical method.

4.5.1 Terminology

As the internal structure of the model is being described it is necessary to introduce a structured way of describing the various aspects of the program. Some of the concepts have been borrowed from the C++ language and modified for use with FORTRAN77. The main problem is that FORTRAN77 is not an object-oriented language, but is very good for solving mathematical problems.

4.5.1.1 Type

This is the specific type of an object that is being referred to, e.g. a pipe or a bed. This is more a specification of the object rather than anything else. For example, all the program code for a bed is in the model, but until the initial data for a bed is loaded no usable beds exist within the program. A type can be compared to a plan, until you actually build the item you have nothing to use.

As FORTRAN77 was used a type is actually a collection of modules and blank data storage that can be filled with initial data from the input file.
how many of each type are actually used depends on the layout specified within the input description file.

4.5.1.2 Instance

As data is loaded into an object of a specific type it now has some substance and can be used. This is known as an ‘instance’, for example once the initial data for a bed has been loaded the bed can be used and referred to. With the analogy as for a type being a plan, an instance is a usable item built from the plan.

4.5.1.3 Group

Valves are used to divide the simulated plant into groups, i.e. a group is connected to other groups via valves. Feeds and products define the external connections, and must be used instead of a valve at that connection point. Within each group, at any given time, the pressure is equal in all the objects, except for a group containing a compressor, which is a special case. A group containing a compressor can have two pressures within it, however there is the restriction that the two pressures must be always be constant. These groups form the basis of the solution for the model. There is no restriction to the number of groups, except that defined when sizing the arrays on compilation of the program.

4.5.2 Internal Model Overview

A model that is intended to be a general-purpose process simulator must be able to cope with two types of changes:
i) in the process layout or equipment without affecting the basic equations used to solve the system,
ii) a change to the model equations or solution method without affecting the process layout or equipment.
To achieve this it is necessary to remove any code dependencies between the objects within the model, i.e. all objects appear the same to the other objects within the model. This prevents code being introduced to cater for a specific case and is best illustrated by an example:

![Figure 4.1 - Normal Program Message Path](image)

If a bed is connected to a tee at either end then it is possible to display this as shown in Figure 4.1. It is now possible for the bed module to have specific code built in for when it is connected to a tee. This introduces complexities into the model that then have to be checked for errors. If the code for the tee changes then the bed code may need to be changed as well.

![Figure 4.2 - Object-oriented Message Path](image)

By moving to an object-oriented design much of this can be eliminated by the 'object independent layer' which separates the objects, as shown in
Figure 4.2. This layer is an ‘abstraction layer’ and actually appears to consist of all the real objects within the model, however it is just an object independent message routing system.

The drawback of this approach is that all calls to a module must go through an extra test to decide which type of object is being referred to and an indexing procedure to find which particular instance is being requested. Using a language such as FORTRAN77 the program is slowed due to the large number of tests that must be performed. If an object-oriented language, such as C++, had been used then the overhead of this would have been virtually eliminated and the need to actually program this layer would have been removed (as this is an inherent part of any object-oriented language).

See Appendix 1 for a description of object-oriented programming and Appendix 2 for a more detailed description of the object structure of the new ILE program.

4.5.3 Model Initialisation

An important part of the model is how the input description file is translated into the plant layout that will be simulated. This translation is done in two distinct stages, after which the model is fully functional, having determined the connections, the groupings, and the calculation order for all the objects.

4.5.3.1 Instance Allocation

When the input description file is read, the first line of any object description is:

```
DECLARE name AS TYPE sometype
```

The DECLARE keyword tells the simulator that a new object is being created. The AS TYPE keyword allows the simulator to identify the last word as being the type of the object and therefore which object type is to be created. The second word is the name of the specific instance of that object,
and which will be referred to by all other objects wishing to make a connection to it.

Once this line has been loaded, the simulator knows the name and type of the object to be created. The object is then allocated a unique ID number which will be used to identify it throughout the program.

The load data code for that particular type of object is then called which then takes care of loading all the details for that object, including dealing with any specific properties for that type. The simulator then has a new object within it, containing all the information necessary to use it, including its connections.

4.5.3.2 Instance Initialisation

Distinct from the loading of the properties of an object is that of initialising it. This is done only after all the objects have been successfully created and loaded with their starting data.

Each object is called in turn to allow it to perform any actions to enable it to be fully functional. For example, only after all the objects have been created can the process connections be sorted out - it is not possible for an object to obtain the ID number of the object it is connected to without the other object having already been loaded and allocated an ID number of its own.

This stage is also used to confirm that all the adjacent objects have consistent properties i.e. they all have the same pressure. Objects that save data will open files and create headings for the data.

4.6 Model Solution

This section will deal only with the solution of the entire system, not with the solution method used for the adsorbent beds, which is detailed by Kirkby (1983) and Espitalier-Noel (1988). see Section 3.3.3.
The solution of the model system is relatively easy, however, due to its iterative nature, the method is computationally intensive. For any given time interval, \( dt \), the starting conditions at \( t=0 \) are known, the values at \( t=dt \) are not, and so must be found.

As each group is linked to at least one other, the solution is that of a set of simultaneous equations. There are many ways of solving this equation set, Gaussian elimination (if linearised), and iterative methods etc. The aim of the method is to solve the system such that \( dP/dt \), flow rates, and concentrations at \( t=dt \) are the same between successive iterations (within some tolerance).

4.6.1 Solution Method

The solution method chosen was that of successive substitution to ensure convergence for each time interval. This method was favoured as the groups that contain adsorbent beds do not always show a linear relationship between flow in and out of the beds at constant pressure, and can include discontinuities. The flow out of the beds depends not only on the inlet velocity and pressure change, but on the relative concentrations of the gas within the bed. In addition, the value of \( dP/dt \) is not constant for an adsorbent bed (as in the case of, for example, a pipe) as it depends upon the concentrations profile within the beds. By using a successive substitution method it is possible to gradually converge on a solution.

The solution method for the system relies on the groups of objects being separated by the valves, which are the only objects with pressure drops in the model (except for the special case of a compressor). Every object within a group must therefore be at the same pressure at some point in time. The model solver estimates the change in pressure for each group, based on the pressure changes for the last time interval. The objects within the system then calculate a solution set, this process is repeated once more to give two approximate solutions. The model solver then uses the flow errors for these
two calculation sets to provide a more accurate value for the pressure change within each group. This process is repeated until the flow and concentration errors have been reduced below the tolerances set in the program, at which point the solution is moved on to the next time interval. Further details on the solver object can be found in Appendix 2.

4.7 Plant Control

The sequencing of a PSA unit is normally done as a fixed cycle of steps, with the valve output being activated at the required time by some type of sequencer device. This can either be a fixed sequencing device, or a computer. The advantage of using a computer is that it can also be used to record and analyse data from the plant, however, this is achieved at extra cost and complexity.

4.7.1 Background

At the start of this project the PSA pilot plant was designed to be controlled and monitored by computer. An IBM PC computer was used, fitted with two interface boards:

1) a thermocouple board (via an external thermocouple connection board) used to read the temperatures.
2) a digital input/output and analogue/digital input board used to control the valves and compressor safety interlock, and to read the pressures, concentrations, and flow rates from the plant instruments.

The computer was controlled with a QuickBasic program that allowed various operating cycles to be chosen. The plant could then be left unattended as it sequenced the plant and recorded the data automatically. In addition it contained safety monitoring routines which could shut the plant down if certain parameters were exceeded.
4.7.2 Model/Plant Interface

For the more advanced control methods to be implemented in this project there were four distinct options that were considered for sequencing the plant and interfacing to the model:

i) use the sequencing computer as a pure sequencing device and the model runs on-line in real-time on a separate computer,
ii) use the sequencing computer as a pure sequencing device and the model runs on-line in non real-time on a separate computer,
iii) both functions are combined into one program and run on a single computer, the model being in a real-time basis,
iv) both functions are combined into one program and run on a single computer, the model being in a non real-time basis.

The relative benefits of the methods are briefly described in the following sections, although the financial cost of each method was not included or considered as a factor. A summary table of the four options is shown in Table 4.3.

<table>
<thead>
<tr>
<th></th>
<th>Option 1</th>
<th>Option 2</th>
<th>Option 3</th>
<th>Option 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Computer</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plant Computer</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Computer</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Model in Real-time</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3 - Plant and Model Computer Options
4.7.2.1 Separate Computers

The use of two separate computers (Options 1 & 2) appears to have many advantages over using one computer. Programmable Logic Controllers (PLCs) are commonly used in industry to sequence the plant, attaching a new computer would require only a few changes. By adding a new computer to the original controller or computer then if one fails the other could report the failure. If the model computer fails then the plant can simply carry on, knowing that it has lost model control and just keep sequencing as normal. If the plant computer fails then the process will halt, possibly in an unsafe condition, however the model computer could then sound an alarm as a warning.

The main drawback of using two separate computers is that of timing. If on-line modelling (Option 1) is being used then the two computer programs must be kept in perfect synchronisation. The model cannot calculate without receiving feedback from the plant about how accurate its prediction is and if conditions change on the plant the model must change its parameters immediately.

The model provides predictions about how the plant computer should operate the plant. If a disturbance occurs that must be rejected, new cycle timing may be needed and the plant must be informed immediately. If the model computer predicts that the plant needs an immediate change, it must be able to inform the plant computer as soon as possible. The solution to these problems is discussed further in Section 4.7.3.4.

If the model is used to provide operating data, but not in real-time (Option 2), i.e. the model is run to steady state and the resulting data used to decide whether the plant cycle is at the optimum for that set of conditions, then the problem of synchronisation does not occur.
4.7.2.2 Single Computer

By using a single computer (Options 3 & 4) the problem of synchronising the two computers across a communications link is eliminated entirely. However new problems are introduced: as there are effectively two programs running on the same machine, one to run the model and one to sequence the plan, each must be given a share of the processor time available and they must be able to synchronise themselves. This leads to other complications, which are discussed further in Section 4.7.3.

As in Option 2, running the model to steady state (Option 3), i.e. on-line, non real-time simplifies the synchronisation between the two programs, as the model can run as fast as possible and report when it has reached steady state.

4.7.3 Plant Operating Programs

Whether the PSA pilot plant is sequenced by a computer separate to that running the model or not, the main factors to be considered when writing the program are briefly reviewed in the sections below, followed by a longer discussion of each item.

4.7.3.1 Speed

The speed of the operating program must be the primary consideration when writing the code. The necessity to deal with the operation of the valves, read in data, process data, display data on screen, and save data to file results in a large amount of code being required. The computer operating the rig (in options 1 & 2) had only 640kB of memory, part of which is taken up by the operating system and the network drivers. This left approximately 500kB of memory for the program.

As the original computer had no maths co-processor (a computer chip designed for purely mathematical functions on floating point numbers) all real numbers had to be processed by a set of mathematical routines.
contained in a code library. This is much slower and limits the amount of mathematical manipulation that could be performed by the computer in real-time.

The amount of code required can be reduced by writing the operating program to deal only with the specific requirements of the purge cycle, as this is the only cycle to be studied, and limiting the mathematical manipulation to the minimum to save the data. By doing so flexibility is lost, however the amount of effort to rewrite the program is significantly reduced and the speed is maintained.

Problems can also occur with the amount of data written to files. If data is recorded as fast as possible from the pilot plant there is the possibility of more data being collected than can be saved (data saving to disk is a relatively slow process due to the mechanical operation of the hard disk when compared to the speed of the processor) and the program may then fail as it cannot keep up with processing the data in real-time. The only way to solve this problem is to avoid it, by restricting the amount of data to be saved and the frequency of saving it.

4.7.3.2 Integration Time Interval

The integration routines should work on a constant time base, i.e. the value of $dt$ should be fixed and constant throughout the experiment, to ensure that readings are taken regularly. If plant data readings occur at the start of each time interval, by using the trapezoidal method the average reading during a time interval can be written as:

$$\bar{R} = \frac{R_s + R_e}{2}$$ (4-8)
To simplify the integration routine the value of $dt$ is made a constant and so for $N$ samples it can be written:

$$ R = dt \sum_{i=1}^{N} \overline{R}_i $$

(4-9)

where $R =$ the total reading.

### 4.7.3.2.1 Interrupt Method

The original operating program, described by Espitalier-Noel (1988), simply scanned the data inputs from the pilot plant as rapidly as possible, multiplying by the time difference measured on the computer's internal clock. This leads to time intervals being of different length or even zero (if the more than one scan is performed within the smallest time interval measurable by the internal clock).

A conceptually simpler method is to use the computer's internal clock interrupt (used to update the internal clock counter within the operating system, which occurs at approximately 1/18 second intervals) to activate the code to scan the data inputs from the pilot plant, thus keeping a regular time base. However, this is more complicated to program and is impossible in QuickBasic (which the original program was written in). By rewriting the operating program in C this method was possible, an additional benefit was a reduction in the size of the program and improved performance (by optimizing the code for a 80286 based PC rather than for an older 8086 based PC, as produced by QuickBasic).
4.7.3.3.2 Interrupt Problems

There are several problems that can occur when using an interrupt method within the operating program, many of which result in failure of the computer program. A short description of these are given, along with the solution used.

Following from the integration method described in Section 4.6.3.2 is the problem of 'interrupt latency'. This refers to the problem of failing to deal with interrupts either fast or often enough, resulting in interrupts either being lost or occurring (and being queued) faster than the program can deal with them. This can be shown by reference to the Figure 4.3, where a viable interrupt handler is shown, enough time is available for the 'Interrupt Handler' and the 'Normal Program' to run and complete before the next interrupt occurs (the time each interrupt is generated is represented by a dashed line).

<table>
<thead>
<tr>
<th>Interrupt Number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupt Handler</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Normal Program</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>

**Figure 4.3 - Viable Interrupt Handler**

In Figure 4.4 a non-viable interrupt system is shown, where the interrupt handler and normal program both take twice as long to run as in Figure 4.3. As the interrupt handler gets priority there is no longer enough time for the normal program to run before the next interrupt occurs. The portion of the normal program after the dashed line will have to wait until
the next interrupt handler is finished. Under these conditions a backlog will begin to occur rapidly in the program (this is show in Figure 4.4 by the 'Actual Program' line which falls rapidly behind), thus causing its real-time nature to be lost.

This is a serious problem and the only solution is that the interrupt routines and normal program be as efficient and hence as short as possible. It is also necessary to avoid sections of code that disable interrupts for any prolonged period of time as this can cause interrupt handling problems. Writing data to the screen and disk can cause interrupts to be disabled on a PC, rapid continuous writing of data to the screen or disk can cause the program to miss interrupts or create a backlog.

<table>
<thead>
<tr>
<th>Interrupt Number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupt Handler</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td>Normal Program</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td>Actual Program</td>
<td>1a</td>
<td>1b</td>
<td>2a</td>
<td>2b</td>
</tr>
</tbody>
</table>

Figure 4.4 - Non-viable Interrupt Handler

The normal outcome of failing to deal with interrupts properly is that the program will stop working and the computer will fail, requiring a complete re-start of the system. However, the most insidious problem occurs
when the backlog accumulates very slowly, it may take many hours before the program will fail and then, apparently, for no reason.

The final problem that occurs is that of performing some action within the interrupt routine that conflicts with the normal operation of the program. For example, saving data to file from within the interrupt routine could result in a corrupt data file, as shown in the following sequence of events:

1) the main program code starts to write cycle heading data to the cycle output file as a new cycle has started.
2) a timer interrupt occurs and the interrupt routine is activated.
3) the interrupt routine decides to write the previous cycle data to the cycle output file for storage.
4) the interrupt routine ends.
5) the main program code resumes, continuing with writing data to the cycle output file, not knowing that the interrupt routine has written data to the file in the meantime.

This situation is normally avoided by simply not allowing the interrupt routine to perform any operation other than reading the data in from the interface boards and setting the valve positions, i.e. only operations that are solely performed by the interrupt routine.

If data are available for saving or display on-screen then a flag is set and the main program deals with the results. This requires flags that can be set to indicate any of the conditions encountered by the interrupt routine. Additionally flags can be set by the main program to communicate to the interrupt routine, e.g. to indicate the plant should be shut down.

The time base that proved optimal for the 75MHz Pentium computer used in the modelling and control of the pilot plant was a dt of 0.2s within the model calculations and a rate of 10Hz for the interrupt routine. If a 20Hz sampling frequency was used for the interrupt routine (which worked on a 90MHz Pentium computer used for program development) then the plant
computer could not keep up and random data loss would occur, eventually resulting in the model and plant controller failing.

The interrupt frequency and time base values are set from the input file, which is read at the start of the program's execution. It is important that checks are made when first running the program to ensure that data are not being lost and that the computer is not failing to keep up with the workload required.

4.7.3.4 Synchronisation of Two Computers

As stated in Section 4.7.2.1, if two computers are being used (one to operate the plant, the other to calculate the model) they must be perfectly synchronised. Using shared files is one solution, passing data via some other means, such as a serial communication line, is another option.

This shared files method was initially tried, using the departmental computer network to transfer the data between the two computers. If data was to be transferred then a signal code would be placed within a shared file, the other computer could then read the new data file and reset the signal code to zero. The problem occurred in the time delay to transmit the data as the network operating software (Lantastic, version 4) would store data in its internal buffers before passing it onto the network. This meant that the shortest guaranteed transmission time was around five seconds from one computer to another, resulting in a total delay of ten seconds before a disturbance could be reported by the plant computer and new instructions returned from the simulation computer.

In the work by Ponton et al (1989), who studied simulation methods using multiple networked computers, it was found that the major problem was that of the communication lag between the computers. The large amount of data to be transferred could also adversely affect the network performance as more than one computer would attempt to transmit information at the same time.
Although the higher speed network used in the departmental network and the fact that only two computers were in use removed the network performance problem, it was not possible to overcome the inherent timing problem. Additionally, for an unidentified reason, problems would occur if both computers tried to access the file at the same time, even with the network file sharing system in use.

Delays of another type also affected the computer communication. Other users on the network could cause large amounts of network traffic, albeit briefly, causing critical data transfers to be delayed and the synchronisation to then fail. By isolating the two computers, i.e. connecting their network cards directly together, these interruptions could be completely removed.

However, the inherent time delay could not, as this was a function of the network driver software, being designed to increase efficiency of network transfer by waiting for enough data to fill an entire network packet of an optimum size. By adjusting the network driver parameters the packet size could be slightly reduced but it did not seem possible to force the network driver to send the data packet instantly, regardless of size. Although it would have been possible to write a simple network software interface to talk directly to the network cards, and thus force instant data transmission, this would have taken too long to be incorporated into this project.

4.8 Concluding Comments

There are many factors to be considered when writing plant control and model software: it is important that the model section of the program is able to run in 'real-time'; that the integration time is a constant; that the data collection is regular and reliable; and that the plant sequencing is done on a regular basis, regardless of what the program is doing.

The final method chosen for the project was to use a single computer to run both the plant operating and the model simulator programs. This was
done by combining them into a single program, written in FORTRAN77 and compiled using Salford Software FTN77.

The plant control and model software that was written was designed specifically to meet the requirements above, and provided a stable and efficient basis for performing the experimental work.
Chapter 5
Experimental Apparatus

5.1 Introduction

The PSA pilot plant within the department was built by Espitalier-Noel (1988) in order to study the effect of waste gas recycle, which had been predicted to improve the yield and oxygen concentration of the process by reducing the oxygen content of the waste. This process involved recycling oxygen enriched gas normally lost in the waste stream back into the compressor feed line for reuse as feed gas. These experiments successfully demonstrated that it was possible to simultaneously improve the oxygen yield and increase the product concentration from the plant. The plant was also designed to allow study of the temperatures, pressures, flow rates, and compositions of the process to provide experimental data for model validation.

The pilot plant is a two bed process, originally using a Bayer 5A zeolite, with a guard section of alumina at the feed end of each bed to help prevent water contamination of the adsorbent. Each bed vessel is 1m long, with an internal diameter of 0.115m, (volume 10.400dm³) and contains approximately 5.75kg of zeolite and 0.7kg of alumina. During the experimental work for this project the beds were filled with a Bayer 5A zeolite and the guard section filled with standard alumina.

It was sequenced automatically by an IBM PC computer, which also recorded all the data from the plant and saved them to file. The computer program allowed several unattended experiments to be performed and could be left running for several days without intervention.
Figure 5.1 - Pilot Plant P&I Diagram
One objective of this project was to replace the original computer with a generic IBM PC clone computer (with a Pentium processor) running a new program, written in FORTRAN77, which would include a robust real-time model of the process as well as the ability to sequence the plant.

5.2 Plant Layout

The PSA pilot plant is a simple two bed system, with product and waste gas receivers. There are several options to modify the plant so that different step types can be tested, such as waste gas recycle, well mixed or direct purge, purge snatching, bed pressure equalisation, and backfill. A detailed diagram is shown in Figure 5.1. Full mechanical details of the PSA pilot plant are given by Espitalier-Noel (1988), however the following sections summarise the main details.

During all the experimental work the plant was set up for a purge cycle, using purge gas supplied directly from the other bed.

5.2.1 Adsorbent Beds

Halfway along the length of each bed a fitting is present for pressure transducers. At the side of each bed there are five fittings, spaced equally along the length, that can be used for thermocouples, in order to measure the temperature within the adsorbent beds, as well as possible sampling points for the gas within the beds.

At both ends of the bed are screens which consist of fine mesh gauze on a brass ring with a sealing ring on the outer rim, these hold the zeolite and alumina sections firmly in place, to help avoid attrition and loss of the particles. An additional screen separates the two adsorbents, preventing them from mixing.

The amount of dead space at the top and bottom of the beds was calculated to be $3.88 \times 10^{-4}$ m$^3$ and $1.114 \times 10^{-3}$ m$^3$ respectively. This was done by measuring the individual pieces and then summing the volumes. The
dead space volume at the feed end is higher as the inlet and waste pipes are of a larger diameter than the product and purge piping, required due to the large volumetric flows seen at the feed end, and because the voidage within the alumina is included.

5.2.1.1 Guard Section

A guard section at the feed end of each bed was used to prevent water contamination of the zeolite adsorbent. Water vapour entering with the feed gas will adsorb onto a zeolite so strongly that it cannot be removed except by heating to drive the water off. Alumina adsorbs water from the high pressure feed gas as it enters the bed, the water is desorbed when the depressurisation step occurs, thus preventing the water from reaching the zeolite.

The guard section is 0.10m in length (volume 1.040dm³) containing 700±1g of alumina (2 to 5mm beads, bulk density 800kg.m⁻³). It is held in place by a mesh screen both above and below. The top screen of the alumina acts as the bottom screen for the zeolite section, there is no dead space between the zeolite and the alumina sections.

The alumina takes no part in the adsorption process for oxygen and nitrogen, it simply removes water from the air. Therefore the voidage from the space occupied by the alumina was taken to be dead volume and so included in the dead volume for the feed end of the bed.

5.2.1.2 Zeolite Section

The main oxygen-nitrogen separation occurs in the zeolite bed, where nitrogen is adsorbed preferentially to oxygen, thus concentrating oxygen in the gas phase. Each bed was filled with 5.750kg of Bayer 5A zeolite (1 to 3mm beads, bulk density 700kg.m⁻³), giving a bed depth of 0.835m (volume 8.684dm³). As, nominally, the same adsorbent was used to that previously installed the adsorption parameters were taken to be as given by Espitalier-Noel (1988).
5.2.2 Air Pretreatment

The air supplied to the plant comes from an oil-free, air cooled, reciprocating compressor, with a maximum delivery pressure of 10 barg. The feed to this compressor comes from the ambient air within the laboratory. As this feed air contains moisture and other contaminants, a refrigerant drier and an oil & water filter were placed on the compressor discharge (within the compressor recycle loop). This delivered air to the plant saturated in water vapour at 275K, with less than 10 ppm levels of oil and hydrocarbon.

The removal of as much moisture as possible from the feed air is to prevent water adsorbing irreversibly (at the operating conditions of the plant) onto the zeolite, thus reducing its capacity for adsorption of the gases. Originally the plant had been operated without the alumina sections at the feed end of the beds, however early experimental work showed a gradual loss of performance, necessitating a refill of the beds. The alumina sections were added to act as a guard bed, effectively acting as a PSA air drier. Examination of the adsorbent showed a weight gain of 3-10% due to water contamination, Espitalier-Noel (1988). This alumina section was maintained in all experimental work performed during this project to prevent a similar problem from re-occurring.

During the later stages of the experimental work, when the project was at the stage where the model could be connected to the plant, the refrigerant drier failed. All the refrigerant escaped from the sealed system into the air stream to the plant. This was caused by corrosion failure of the air-refrigerant concentric-pipe heat exchanger. The damage proved to be irreparable and the drier had to be removed, however the oil & water filter was left in place.

It was noticed that the performance of the plant had dropped significantly and it was assumed that both beds had been equally contaminated. When the plant had been modified to allow the product receiver to be taken out of the product system it was found that only one of
the beds had a significant reduction in its adsorbent capacity whilst the other appeared to have suffered no visible contamination. At this point the adsorbent was replaced by fresh standard zeolite 5A obtained from Bayer.

5.2.3 Instrumentation

As the plant is sequenced by a computer all the instrumentation is monitored by the custom written computer program and all data is recorded in files for later examination and analysis.

An Amplicon A/D interface card in the computer is attached to a thermocouple board where the physical connections to the type K thermocouples are made. This board provides automatic cold-end compensation of the thermocouples, allowing the temperatures to be read directly without further adjustment being required. Unfortunately this board would not fit the new Pentium computer and, although the program contains the code necessary to read the temperatures, these could not be used.

The twelve thermocouples are located in various gas streams, to allow the gas temperature entering and leaving the beds to be measured, as well as the compressor feed and the ambient air temperature. In one of the beds there are five thermocouples located at equal distances axially. This allows the bed temperature profile within the bed to be monitored during the cycle.

A second interface card, a DAS-08, is used to control the valves and to provide the analogue to digital conversion of signals from the flow meters, pressure transducers, and oxygen analysers. This is a 12-bit device, ranged 0-10V.

The pressure transducers measure from 0 to 10 bara, giving an analogue output of 0-10V dc. Resolution is 0.00244 bar/bit at the computer. A correction factor is used in order that they read identically when placed at the same pressure.
The flow meters give an analogue output of 0-5V dc. They are used to measure the total air inlet flow, total waste gas flow, product gas flow, and purge gas flow. See Table 5.1 for the resolution of each flowmeter.

<table>
<thead>
<tr>
<th></th>
<th>Feed</th>
<th>Product</th>
<th>Purge</th>
<th>Waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>0.004015</td>
<td>0.000036</td>
<td>0.000626</td>
<td>0.002669</td>
</tr>
<tr>
<td>(dm³@STP /s/bit)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Flow</td>
<td>16.45</td>
<td>0.15</td>
<td>2.56</td>
<td>10.93</td>
</tr>
<tr>
<td>(dm³@STP/s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 - Flowmeter Calibration

Flowmeter calibration was carried out using a bubble column and a stopwatch, for a steady flow through each flowmeter. The computer recorded the total number of bits from the interface card during the period and the time taken to fill a known volume of the bubble column was recorded on a stopwatch. The calibration factors could then be calculated.

The on-line oxygen analysis is performed using two Servomex oxygen analysers, which use the paramagnetic principle, which are connected to the waste and product streams. These output a range of 0-100% O₂, giving an analogue output of 0-1V dc, which are then amplified to a signal of 0-10V dc with the amplified resolution being 0.0244%/bit.

5.2.4 Computer

The original computer responsible for the sequencing of the plant was an IBM PC 80286 running at 8Mhz, with 640Kb of ram, with a VGA colour monitor. It has a 20Mb hard disk for data storage. A network card was installed to allow data to be easily transferred to a more powerful computer for analysis. Figure 5.2 shows how the computer operation was integrated with the pilot plant.
5.3 Modifications

The original design of the plant was for the observation of the steady state operating performance of a PSA unit. One aim of this project was to observe, and attempt the control of, the dynamic response of the unit to disturbances. This required several modifications to be made to the unit.

In order to introduce a flow rate disturbance into the product stream it was necessary to automatically change the flow rate by a set amount. If a mass flow controller had been used on the product line then a simple adjustment to its set point would have had the desired effect. Unfortunately this use of the plant had not been envisaged and so only mass flow meters had been used on all the streams.
To replace the existing section (as shown in Figure 5.3), which had suffered from relatively poor flow control, and to provide the required disturbances a new product line was built, see Figure 5.4.

The new product line contained a main needle valve (NV-2) to set the normal product flow rate. Connected in parallel to this valve was a second needle valve (NV-3) in series with a solenoid valve (9b), see Figure 5.4. This second needle valve could be set to create a specific increase in the flow rate when the solenoid valve was opened, thus acting as a step change disturbance to the product flow rate. The solenoid valve was controlled by the computer to allow automatic introduction and removal of the disturbance.

The piping leaving the product receiver was replaced with new 1/4" copper piping, using Swagelock fittings. The specifications for the three new needle valves are given in Table 5.2.

This new layout allows Valve 9b to be controlled by the computer, enabling the automatic introduction of a step change in product flow rate. By
adjusting the flow rate manually using NV-3 whilst 9b is open allows a specific step change in flow rate to be set. When the program runs 9b is normally shut, at the required time the valve is opened and the disturbance introduced. After a specified period the 9b is shut again. Alternatively, 9b could be normally open, it then being closed to introduce the disturbance of a decrease in product flow rate, and reopened to remove the disturbance.

<table>
<thead>
<tr>
<th></th>
<th>Type</th>
<th>Cv</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-1</td>
<td>Needle</td>
<td>0.004</td>
<td>Brass</td>
</tr>
<tr>
<td>NV-2</td>
<td>Needle</td>
<td>0.004</td>
<td>Brass</td>
</tr>
<tr>
<td>NV-3</td>
<td>Needle</td>
<td>0.03</td>
<td>Brass</td>
</tr>
</tbody>
</table>

Table 5.2 - Needle Valve Specifications
5.4 Concluding Comments

The PSA pilot plant essentially remained as Espitalier-Noel had built it, except for the modification to the product pipework to allow more accurate adjustment of the product flows and a disturbance to be introduced. The adsorbent beds and alumina had been replaced and the whole system was cleaned and recalibrated.

Finally, with the replacement of the original computer with a new one, giving the system the computational power required to run a real-time model, but at the unfortunate loss of the temperature information, the pilot plant was ready for the experimental programme.
Chapter 6
Computer Modelling

6.1 Introduction

There are a number of degrees of freedom for an existing PSA plant. On a real PSA plant the adsorbent type, feed composition, ambient temperature, vessel dimensions and so forth are effectively fixed. Most previous work has concentrated on the remaining independent operating variables, such as process step times and product amount per part cycle, that can be manipulated on a real plant as well as in a model. These have been extensively studied in order to show the effect of manipulating these on the overall performance of the process.

Therefore the modelling work was done for two main reasons. The first was to compare the model against the previous ILE model developed by Kirkby (1984) and later modified by Espitalier-Noel (1988). The second reason was to look at the sensitivity of the model to parameters that had to be specified, such as dead volumes and adsorption parameters. This work was generally aimed at giving a good fundamental understanding of how the process performance is affected by changes to the equipment surrounding the adsorbent beds.

6.1.1 Experimental & Theoretical Aims

The first aim was to compare the performance of the model with the previous model by Kirkby (1983) to see how the performance changed with modelling the surrounding process plant equipment and the beds individually. In the research literature it is common practice to model only the adsorbent beds, rather than the process as a whole, and to model only one bed whilst assuming symmetrical operation for all others. Some work has

The second aim was to perform a parametric study on many of the operating variables within the simulation in order to study the effect on the model's performance and stability. This also gives an indication of the sensitivity of the model to the various parameters and suggests which may be used to fine tune the model to the data from the plant (Chapter 7). This is important because if an unsuitable parameter is chosen it may have an unpredictable effect on the model's output.

The third aim was to show that varying the adsorption parameters for the beds would have a predictable, and reasonably linear, effect on the predicted product concentration. A main overall aim of this project was to show if it was possible to manipulate these parameters in real-time to match the predicted product concentration with that of another model running with different initial conditions, see Chapter 7.

6.1.2 Conditions & Disturbances

All the simulations and pilot plant work were done using a direct purge cycle with a fixed purge flow rate, the total purge amount per cycle was altered by adjusting the lengths of the purge step and the null step, not the purge gas flow rate. In all cases the values for the upper and lower operating pressures were kept constant. As in the work of Espitalier-Noel (1988), all flow rates will be presented in terms of dm³ @ STP conditions.

The following changes in operating parameters were investigated for the model:

1) changing the adsorption parameters, by +20/-15%,
2) changing the dead volume at the product end of the adsorbent bed, from 0.02 to 7.8 dm³, compared to the bed volume of 7.4 dm³.
changing the dead volume at the feed end of the adsorbent bed, from 0.06 to 22.3 dm³,
unsymmetrical purge amounts, and
unsymmetrical bed volumes.

The PSA pilot plant was used to provide data to help tune the model to be a reasonable representation of the pilot plant. The data was collected for the following conditions:
1) start up of the process, and
2) cycle performance i.e. pressure & flow profiles within a cycle.
As cyclic steady state has not been reached during startup the trends during this period provide a good early indication of whether or not the model is a reasonable representation of the dynamic behaviour of the plant.

![Figure 6.1 - Standard Cycle](http://example.com)

All experiments in this chapter were performed using the standard cycle shown in Figure 6.1. To adjust the purge amount per part cycle a range of purge step lengths, from 0 to 30s in 5s divisions, was used. The combined length of the purge and null step was kept constant for all experiments at a total of 30s.
6.2 Comparison With Previous ILE Model

The previous ILE model tended to over predict the product oxygen concentration, but gave reasonably good agreement for oxygen yields, Espitalier-Noel (1988). It was noted that the agreement varied with the cycle parameters and it was suggested that models should be validated over a range of operating conditions.

It was also reported that when the rate of pressurisation exceeded 2psi.s\(^{-1}\) mass transfer effects started to have an effect on the performance of the plant. This indicates that below this rate of pressurisation the PSA process in this plant is equilibrium dominated. Therefore, all pressurisation rates in the experiments were performed below this value. It was also noted that the greater the rate of depressurisation the better the performance of the plant.

The new model was run with its parameters set as closely as possible to those given by Espitalier-Noel (1988), i.e. with a bed length of 0.85m and a purge amount ranging from 0 to 35 dm\(^3\)/part cycle. Dead volumes were taken to be the same as those measured from the pilot plant for this project.

6.2.1 Product Concentration

Figure 6.2 shows a plot of product oxygen concentration versus purge amount for the new model superimposed on pilot plant data from Espitalier-Noel (1988). It is similar to most plots given in the literature in that it has an optimum purge amount, on either side of which the product concentration is reduced. It can be seen that the new model predicts a higher value than the plant produces around the optimum point and lower elsewhere.

Comparing the new results against the model results obtained by Espitalier-Noel, Figure 6.3, it can be seen that both models predict similarly shaped trends, yet there are significant differences: the optimum purge amount for the new model is higher than for the old and the absolute values of the two models are quite different. The new model tends to predict lower
product concentrations than the old model for low purge amounts and higher product concentrations at high purge amounts.

There are many different assumptions between the two models although both use the same fundamental solution method for the adsorbent beds. The old model used the well-mixed purge gas assumption, whereas the new model is using direct-purge gas but in the work done by Espitalier-Noel he showed that using direct purge gas had very little effect on the product concentration. The other main difference is that the new model takes into account the effect of the surrounding equipment and that both beds are modelled separately. These differences appear to cause the new model to predict significantly different product concentrations to the old model.

6.2.2 Discussion

The prediction of the maximum product concentration and the optimum purge amount are important criteria for the assessment of a model. This gives an indication as to the suitability of the process at the conditions specified. The normal aim of modelling is to make this prediction as accurate as possible, regardless of computational load.

The underlying solution method used in the previous model by Espitalier-Noel is similar to that used in the new model to predict. However, there are significant differences between the predictions of the two models. They both exhibit the same general shape of trends and both tend to over-predict the product concentration. The removal of the symmetry assumption and the modelling of the surrounding equipment seems to play a significant role in the product purity produced by the models. Although not proven for the results presented in this section, a possible supplemental explanation for some of the differences is given in Appendix 4. There it is shown that multiple steady states are possible for many conditions, and it is possible that, as only one result is presented here, an alternative solution may give closer agreement to the experimental data.
In general, the predictions of the new program appears to be better than the old program when compared to the pilot plant, at higher product amounts it does not over-predict the product concentration as much as the old version. When compared to the old model, the new model predicts better performance at high purge amounts, but this was not a result found on the previous plant work.
Figure 6.2 - New Model vs. Plant Data from Espitalier-Noel (1988)
Figure 6.3 - New Model vs. Model of Espitalier-Noel (1988)
6.3 Model Sensitivity Studies

One main aim of these experiments on the model was to assess how adjusting various process parameters affected the predicted performance of the plant. Normally a model is validated against specific experimental data and all the conditions are matched as exactly as possible to that used to produce the experimental data.

Within the literature there does not seem to be much published work of the effect, theoretically or experimentally, of the equipment surrounding the adsorbent beds on the performance of the plant. To ascertain how the performance of the process depended on the surrounding equipment a parametric study was performed on major equipment properties, such as product receiver volume, dead space of adsorber vessels etc.

6.3.1 Feed End Dead Volume

The dead volume at the feed end of an adsorbent bed is due to the natural shape of the pressure vessel containing the bed (including space required for the retaining grid and flow distributors) and from the pipe work required for the feed and waste valves. The effect of altering this volume does not appear to have been reported in the literature, possibly because most models do not include the effect of this volume, and it would be a time-consuming and expensive item to investigate experimentally.

By increasing and decreasing the feed end dead volume within the model the effect on the predicted process performance has been demonstrated.

6.3.1.1 Investigation Technique

To investigate how the process product concentration changes with respect to the feed end dead volume, the product rate was kept constant whilst the feed end dead volume was altered over the range 0.06 to 22.3dm³.
or 0.8 to 300% of the adsorbent bed volume. This was repeated for a range of purge amounts and product rates. Purge amounts are defined as follows:

- very low - 5.5 dm³ purge ppc
- low - 11.3 dm³ purge ppc
- medium - 17.2 & 23.1 dm³ purge ppc
- high - 29.1 dm³ purge ppc
- very high - 35.0 dm³ purge ppc

### 6.3.1.2 Results for Low Product Rate

The results are shown in Figure 6.4 for a range of purge amounts. For a very low purge amount the size of dead volume appears to have little effect on the product concentration. However, the peak value of the product concentration is only 45% and the lowest value is 34%, giving a relative drop at about 30% in the purity. For a low purge amount the dead volume appears to have no effect on the product concentration until a critical value is reached. The product concentration then falls rapidly as the volume is increased further.

For medium and high purge amounts the size of the dead volume appears to have no effect even at high values of dead volume. This is most likely because the beds are over-purged and so there is enough high concentration gas available in the cycle to purge the feed end dead space.

In the case of very high purge amounts the amount of dead volume appears to have no effect on the product concentration until a critical value is reached. The product concentration then rapidly falls off as the volume increases further. This is probably because the beds are over-purged and so there is still enough high concentration gas available in the cycle to purge the feed end dead volume, until the critical limit is reached. At this point so much product gas has been removed from the on-line bed that poorer quality gas is being discharged into both the purging bed and the product stream.
6.3.1.3 Results for Medium Product Rate

The results are shown in Figure 6.5 for a range of purge flow amounts and they show the same general trends as in Figure 6.4. Again, for a very low purge amount, the amount of dead volume appears to have little effect on the product concentration. However in this case, with a low purge amount the plant is no longer capable of producing the highest quality product, and as the amount of dead volume is increased the product concentration steadily falls. At medium purge amounts the trends are the same as before, except that in one case (17.2dm³) the process gradually fails to maintain the highest product purity, and at an even higher volume so does the high purge amount.

With a very high purge amount, the size of the dead volume appears to have an enhancing effect at lower values, raising the product concentration up from 65% to 95% (its maximum possible value). As the purge rate has been kept constant the effect of the extra dead volume must be to keep some of the purge gas that normally goes out into the waste stream within the system. This gas then enters the bed during the pressurisation step and so raises the product purity.

After this process peaks there appears to be no effect on the product concentration until a second critical value is reached. The product concentration then rapidly falls off as the volume is increased beyond this point, just as for the lower product rates described previously.

6.3.1.4 Results for High Product Rate

The results are shown in Figure 6.6 for a range of purge amounts. At very low and low purge amounts the trends are similar to those seen for lower product rates. However, many more curves are of the type that as the dead volume increases the product concentration steadily falls.

For high and very high purge amounts an increase in the dead volume initially has an enhancing effect from that of the lowest volumes, pushing the product concentration up from 58% to 72% and up from 82% to 95% (its
maximum possible value). This is the same effect as will be seen in Section 6.3.2.3, where the extra dead volume keeps some of the purge gas that normally goes out into the waste stream within the system. This gas then enters the bed during the pressurisation step and so raises the product purity.

### 6.3.1.5 Discussion

Most of the data presented in Figures 6.4 and 6.5 (2 and 4 dm³ ppc of product flow respectively) show cycles with the maximum product concentration over a wide range of feed end dead volumes. However, in Figure 6.6 only one of the lines ever reaches this maximum, and then only over a narrow range of feed end dead volumes.

For the purge amounts that reach maximum product concentration some maintain this across the range of feed dead volume simulated, whilst others show a sudden drop in product purity at a specific size of dead volume, which is then followed by a plateau at a lower value, and finally by a steady decline in concentration.

For most of the cases the size of the dead volume at the feed end of the bed does not appear to be a problem, as long as it is kept below a critical amount. For the PSA pilot plant used in this project the volume is fixed and falls in the lower end of the range as a percentage of the adsorbent bed volume (0.4% VOA). Unfortunately, it falls into the sensitive region where a small increase in dead volume can have a large effect up or down on the product oxygen concentration under some conditions.

Finally, the enhancement seen by increasing the feed end dead volume at some conditions can be compared with the work by Espitalier-Noel (1988). There it was found that the optimum purge amount increased with the introduction of a waste gas recycle system into the plant, which was allowing total breakthrough of the purge gas from the bed into the waste stream. The use of increased feed end dead volumes appears to act in a similar way, but
storing gas within the empty volume as opposed to recycling it around the system. The advantage of this is that no extra pipework or valves are required, however no investigation has been made on the relative power cost of this approach over a more conventional system.
Figure 6.4 - Effect of Feed End Dead Volume (Product=2dm³/cycle)
Figure 6.6 - Effect of Feed End Dead Volume (Product=8dm³/cycle)
6.3.2 Product End Dead Volume

The dead volume at the product end of an adsorbent bed is also due to the natural shape of the pressure vessel containing the bed (including space required for the retaining grid and flow distributors), the vessel not being filled to the top (often due to natural settling of the adsorbent after time) and from the pipe work required for the product and purge valves. The effect of altering this volume does not appear to have been reported in the literature, although it is normally considered good practice to minimise the volume. Similarly to the feed end dead volume, most models do not include its effects and it would be time-consuming and expensive item to investigate experimentally.

The base case had a product end dead space of 0.4dm³, as measured on the plant, equivalent to approximately 4.5% of the adsorbent volume. By increasing and decreasing the product end dead volume within the model the effect on the predicted process performance has been demonstrated.

6.3.2.1 Investigation Technique

To investigate how the process product concentration changes with respect to the volume of the product end dead space the product rate was kept constant whilst the volume of the product end dead space was altered over the range 0.02 to 7.8dm³, equivalent to the range 0.2% to 83% of the volume occupied by the adsorbent. This was repeated for a range of purge and product amounts.

6.3.2.2 Results for a Low Product Amount

The results are shown in Figure 6.7, for a range of purge amounts. With a very low purge amount the size of the dead volume appears to have little effect on the product concentration. However, the peak value of the
product concentration is only 52% and the lowest value is 38%, a relative drop of about 35% in the purity.

For a low purge amount, the dead volume appears to have no effect on the product concentration until a critical value equivalent to about 53% VOA is reached. The product concentration then falls away more and more rapidly as the dead volume increases further, until it slows down at about 72% of VOA. For medium purge amounts the amount of dead volume appears to have no effect on the product concentration until a critical value equivalent to 48% VOA is reached. The product concentration then falls away more and more rapidly as the size of the dead volume increases further, until again it slows down at about 69% of VOA.

This pattern is repeated for higher purge amounts, the amount of dead space volume appears to have no effect on the product concentration until a critical value is reached, this critical value being dependent on the purge amount. As the purge amount increases the slope of the transition range between the higher purity region and the lower purity region becomes less steep. Interestingly, the final concentrations in the lower purity region are very similar in value, regardless of the purge amount, all converging on the concentration generated by the very low purge amount process.

6.3.2.3 Results for a Medium Product Amount

The results are shown in Figure 6.8, for a range of purge amounts. With a very low purge amount the size of the dead space appears to have little effect on the product concentration. However, the peak value of the product concentration is only 52% and the lowest value is 36%, very similar to the results for a low product amount.

For a low purge amount the dead space volume appears to have very little effect on the product concentration until a critical value equivalent to about 53% VOA is reached. However, as compared to the equivalent result for a low product amount, the process can no longer manage to achieve a
product concentration of 95%, reaching a maximum purity of 76%. After 64% VOA the product concentration falls away more and more rapidly as the dead space increases further, until it again slows down after reaching 69% of VOA.

For a medium purge amounts the amount of dead space volume appears to have no effect on the product concentration until a critical value equivalent to 40% VOA is reached. The product concentration then falls away more and more rapidly as the dead space increases further, finally slowing after reaching 67% of VOA.

Yet again, this pattern is repeated for higher purge amounts, the amount of dead space volume appears to have no effect on the product concentration until a critical value is reached, this critical value being dependent on the purge amount. As the purge amount increases the slope of the transition range between the higher purity region and the lower purity region becomes less steep. Again, the final concentrations in the lower purity region are very similar in value, regardless of the purge amount, and are all heading asymptotically to the feed concentration as the volume increases.

6.3.2.4 Results for a High Product Amount

The results are shown in Figure 6.9, for a range of purge amounts. For a very low purge amount the peak concentration is only 43% and the lowest value is 32%, which is lower than for the previous two results.

Compared to the equivalent results for a medium product amount, the process can no longer manage to achieve even a moderately high product concentration with the low or medium purge amounts. Whereas previously the process could manage to achieve a product concentration of 95%, now it reaches a maximum of 68%. At the “optimum” purge amount the critical value reappears, being at only 4% VOA.
Figure 6.8 - Effect of Product End Dead Volume (Product=4dm³/cycle)
Figure 6.9 - Effect of Product End Dead Volume (Product=8dm³/cycle)
At the high purge amounts the pattern is reversed, with an increase in purge amount causing a drop in product concentration. Again, the final concentrations in the lower purity region are very similar in value, regardless of the purge amount, all converging on the concentration generated by the very low purge amount process.

6.3.2.5 Discussion

For the cycles in Figures 6.7-6.8 there is some scatter at lower product end dead volumes. However, all the curves generally display the expected smooth transition from one product purity range to another.

At lower dead volumes there appears to be a pattern of increasing concentration, then a sudden drop, then another increase and so on. Figure 6.10 (which is an expanded view from Figure 6.9 for the range of 0 to 1.5 dm³ product end dead volumes) shows a clear example of this phenomenon. One explanation for these ridges is that very small errors accumulate, from cycle to cycle, in the positions of the shock waves. The results of the model appear to be sensitive to the exact location of some key shock waves. Slight errors in the calculations could therefore have large effects on the product concentration, especially when a key shock wave is near to the end of a bed at the end of a product release step. A supplementary explanation is offered in Appendix 4, although no proof is offered that this explanation is valid for all cases, in that it is the history of the process causing slight differences in the shock wave positions, rather than accumulating errors in their location.

For most of the cases the size of dead volume at the product end of the bed does not appear to be a problem, as long as it is kept below a critical amount. For the PSA pilot plant used in this project the volume is fixed and falls in the lower end of the range as a percentage of the adsorbent bed volume (0.4% VOA). Unfortunately, it falls into the region where a small
change in dead space volume can have a large effect on predicted product oxygen concentration under some conditions.

By plotting the volume at which the product concentration firsts drops from the maximum amount it is possible to define a critical volume. This is shown in Figure 6.11. For each product flow rate there is a range of critical volumes, depending on the amount of purge used. The higher the product flow rate the lower the critical volume, until at the highest flow rate no cycle could produce maximum concentration except for one case (with a very low volume). At lower product flow rates the curves appear to be asymptotically approaching a maximum allowable volume regardless of how low the flow goes. This graph shows that for all cases minimising the product end dead space is vital to achieving the highest possible product flow from a PSA plant.

Finally, it is clear that the importance of minimising the product end dead volume is greater than that for the feed end dead volume. The process appears to be capable of maintaining product purity with a relative increase in feed end dead volume three to four times the increase in the product end dead space.
Figure 6.11 - Critical Product End Dead Volume vs Purge Amount per Part Cycle
6.3.3 Unsymmetrical Purging

The purge flow rate and total purge amount in most models are assumed to be identical for both beds, however in practise this is not normally true due to slight variations in process layout, pressure drops etc. By using a flow controller to provide a virtually constant flow, the differences could be avoided, however no flow controllers were fitted to the pilot plant and they are rarely fitted to industrial plants. Therefore, the purge flow rate and total amount can vary slightly from bed to bed and from cycle to cycle.

6.3.4.1 Investigation Technique

To investigate how the process product concentration changes with respect to the symmetry of the purge rate, the product rate was kept constant whilst the purge flow rate to one of the beds was altered by up to +/-40%. This was repeated for a range of purge amounts and for a range of product rates.

6.3.4.2 Results & Discussion

A selection of results is shown in Figures 6.12 to 6.14. Each line represents bed A receiving the normal purge flow rate ($\approx 0.053\text{mol/s}$) whilst bed B receives the flow rate as shown on the x-axis. Most of the results appear to be approximately mirrored around the normal purge flow rate and when the process is not symmetrically purged the process rapidly fails to give the highest product purity it is capable of.

For the low product flow rate, Figure 6.12, as soon as purge symmetry is broken the process is no longer capable of producing maximum concentration product gas, the drop in product concentration is between 12 and 25%. As the purges are made more uneven the product concentration rapidly drops off, until it appears to be approaching a minimum level. In some cases though, making the purge unsymmetrical causes the process to respond better.
Figure 6.12 - Change in Purge Symmetry (Product=2dm³/cycle)
Figure 6.13 - Change in Purge Symmetry (Product=4dm³/cycle)
At the very low purge rate of 5.5dm³ ppc, with symmetrical purge rates the process can only produce product of 44%, with a highly unsymmetrical purge this rises to almost 60%. With the 23.1dm³ ppc normal purge, the process remains at maximum product concentration even when purge symmetry is broken by 10%, i.e. Bed B is receiving 10% more purge gas than bed A. This combination of purge amounts and product rate seem to be the only case where this phenomenon occurs. It probably indicates that it is better to have one bed properly purged than none.

For the medium and high product rates, Figures 6.13 & 6.14, a similar trend occurs to that at low product rates. However, the lowest normal purge rate now produces an almost constant product concentration, regardless of how unbalanced the purge rates are.

There are also cases where slightly asymmetrical purge produces a higher product concentration than with a symmetrical purge rate. In all these cases, when bed B is receiving more purge than bed A the product concentration rises, except for when the beds are over-purged.

Within the model, the two beds are simulated individually rather than being assumed to be identical but out of phase. This allows different concentration profiles to form within each bed, although they are often only slight differences. However on a real PSA plant it is possible that the effect of diffusion within the concentration fronts, and any minor differences between the two beds, would result in the beds having the same profiles at steady state.

The ability of a PSA plant to tolerate minor variations in purge rates between the beds appears to be relatively poor. To maintain the maximum product purity, normally required from industrial units, the deviations in the purge amounts must be small, and unfortunately the more tolerant cycles are probably not industrially useful, although there could be uses for the supply of enriched air.
It is unlikely that any real advantage could be taken of these effects in an industrial setting as they are minor. However, where one bed is for some reason not performing as well as the other (perhaps a different batch of adsorbent) then by unsymmetrical purging of the beds the product concentration could be maintained or at least the effect reduced.

6.3.5 Unsymmetrical Beds

The adsorbent beds in a PSA process are normally assumed to be identical, most models go as far as to assume that only one bed need be modelled as they will all behave identically, just out of phase with each other. As the model used in these experiments calculates each bed individually, it is possible to see the effect of having bed lengths that are not identical.

6.3.5.1 Investigation Technique

To investigate how the process product concentration changes with respect to the symmetry of the bed lengths, the product rate was kept constant whilst the length of one of the beds was altered by +/-40%, from the pilot plant size of 0.835m. This was repeated for a range of purge flow rates and for a range of product rates.

6.3.5.2 Results & Discussion

The results are presented as contour plots in Figure 6.15 to 6.17. The graphs are plots of the length of one bed versus purge amount and contours of product concentration are plotted for this plane.

In Figure 6.15 the results are shown for a low product rate. With one bed much shorter than the other, it is relatively easy to over-purge the shorter bed. Conversely, with one bed much longer than the other it is relatively easy to under purge the longer bed. In general the process performs well despite asymmetry because the product amount is low.
For the medium product amount, Figure 6.16, the situation is very similar to that in Figure 6.15, but the 90%+ region is smaller.

The effect of increasing the product amount again is shown in Figure 6.17. The 90%+ region is now extremely small, and it does not lie across the line that represents the two beds having equal length. This indicates that the optimum performance for this product flow rate is with one bed larger than the other. At high product rates it appears to be better to have one bed big enough than both being too small. However, if the bigger bed is made too large, then the product purity decreases because it cannot be properly purged by the smaller bed.

6.3.6 Adsorption Parameters

The initial adsorbent parameters used in the model were taken from those given by Espitalier-Noel (1988). By changing the value of these adsorption parameters it should be possible to manipulate the final steady state product concentration of the process.

The adsorption parameters can usually be measured experimentally for fresh zeolite used in a PSA plant, using a sample taken when the beds are filled. However, for many existing plants the true value of these adsorption parameters were never measured, the parameters often reduce with time (e.g. from contamination), and they can change with the process conditions (e.g. ambient temperature changes).
Figure 6.15 - Variation in the Length of One Bed (Product=2dm$^3$/cycle)
Figure 6.16 - Variation in the Length of One Bed (Product=4dm$^3$/cycle)
Figure 6.17 - Variation in the Length of One Bed (Product=8dm³/cycle)
6.3.6.1 Investigation Technique

To investigate how the process product concentration changes with respect to changing the adsorption parameters the values of $k_0$ and $k_N$ were adjusted over the range $+20/-15\%$. This range was chosen such that the values of $k_0$ and $k_N$ could not become reversed, as the adsorptivity of nitrogen must always be higher than that of oxygen.

Note: this limitation is imposed by Equation 3.9, where the definition of the augmented mole fraction ‘$Y$’ was defined as $(k_N-k_O)y_O + k_O$ and $y$ as the normal mole fraction. When converting from $Y$ values back to $y$ values, the equation is rearranged to become $y = \frac{y_O - k_O}{(k_N-k_O)}$, and if the denominator is zero the result is undefined. If the values of $k_0$ and $k_N$ become reversed the process will start to deliver nitrogen as the product, however, this separation reversal was considered to be unlikely on a real plant and so was not investigated.

The major conditions for the plant were a purge amount of 17.2dm$^3$ ppc, a product amount of 8dm$^3$ per cycle, and both beds of length 0.835m.

6.3.6.2 Results & Discussion

In Figure 6.18 the results are shown for the effect of varying both $k_0$ and $k_N$. The results are presented as a contour plot of product concentration for the $k_0$ and $k_N$ plane.

It can clearly be seen that adjusting $k_0$ and $k_N$ has a significant effect on the product concentration produced. Increasing the value of $k_0$ whilst keeping $k_N$ constant causes a drop in product concentration, whilst keeping $k_0$ the same and increasing $k_N$ causes an increase in the product concentration. This is expected as raising the ratio of $k_N/k_0$ translates physically to increasing the selectivity of the zeolite in favour of nitrogen, thus enhancing the separation.
As the selectivity of the adsorbent approaches unity, i.e. \( k_o = k_N \), then there is no preference for adsorption of nitrogen over oxygen and hence no separation. As expected under these conditions, the process simply produces a product with the composition of air.

Keeping the ratio of \( k_N/k_o \) constant keeps the product concentration approximately constant. It was expected that adjusting the total adsorption of the zeolite would have some effect on the performance of the process. That it does not seem to is interesting as it says that, at least for this model of the PSA process, product purity will be the same regardless of total adsorption as long as the selectivity is kept the same. However, the amount of gas required to pressurise the beds increases as the total adsorption goes up, thus raising the amount of power required by the process. It should be remembered that a reasonable amount of adsorption is required for the process to work on a practical scale.

The most important feature of these results is that the gradient of the product concentration, with respect to changing either \( k_N \) or \( k_o \) individually, is reasonably linear, although it does curve slightly as the ratio of \( k_N/k_o \) is reduced. This implies that the value of either \( k_N \) or \( k_o \) would be a suitable parameter to adjust the product concentration of the model to match that of a real process.

6.4 Conclusions

The first part of this chapter compared the new ILE model, including surrounding equipment with that of previous work done with a previous ILE model (Espitalier-Noel, 1988) to see if adding the equipment would affect the final product concentrations. This was proven to be the case, as comparing the new model with the previous model showed that the new model predicted lower product concentrations at lower purge amounts and higher product concentrations at higher purge rates. However, it was still not possible to match the ILE model with the product concentrations that were obtained...
from pilot plant work by Espitalier-Noel, the new model under-predicted product concentrations at low purge amounts and over-predicted it for higher purge amounts, although it did slightly better than the previous model.

Throughout the bulk of this chapter the aim has been to see the effect on the process of adding variation into what could be considered key parameters of the PSA process. There has been a large amount of literature published since the PSA process became industrially important, however there is limited information available on what would be the effect of having an unsymmetrical process.

The main conclusions for the variations studied are:

1) Feed End Dead Volume

The process can normally withstand large changes in the feed end dead volume, until a critical volume is reached, after which the product concentration shows a gradual decline in value. In a limited number of cases the inclusion of extra dead volume improved the performance, this is most likely due to the extra volume capturing gas that would otherwise be lost during the depressurisation and purge steps. However, more gas is required by the process in order to pressurise this extra volume and so the yield of the cycle goes down. As no studies were made of the capital and operating costs of the various cycles it is not possible to say whether there would be any economic benefit from using this effect.

2) Product End Deadspace

The general trends were similar to that seen for the feed end dead volume, however the process was less tolerant of increasing this dead volume. Additionally, the results would often cycle up and down from point to point in a ragged manner. This is because the results of the model are extremely sensitive to the exact location of some key shock waves. Slight errors in the calculations can have large effects on the product concentration
when a shock wave is near to the end of a bed at the end of a product release step. A supplementary explanation is offered in Appendix 4, although no proof is offered that it is valid for all the cases, in that multiple steady states are possible for the PSA process, depending on the history of the process.

The drop in product concentration was more rapid than that for the feed end dead volumes, all of the cycles produced product gas of less than 40% oxygen for the same percentage increase in the product dead volume compared to the increase in feed end dead volume.

3) Purge Symmetry

The process is very intolerant of unsymmetrical purging. Deviations in the symmetry of the purge flow amounts by 10% caused the product concentration to immediately drop from the maximum value, often the oxygen concentration dropped by as much as 15% for a difference in purge amount between the two beds of only 10%. This occurred even for the cycles withdrawing very little product which would have been expected to be the most tolerant of variations in the purge amounts.

In the odd case, it was possible to improve the product concentration slightly by unsymmetrical purging, however these were cycles that produced only low concentration oxygen and are unlikely to be used commercially.

4) Bed Symmetry

The effect of a small change in the size of one bed length had very little effect on the product concentrations. There was an effect for cycles with large differences between the sizes of the two beds, or for very low or very high purge amounts. Most cycles showed a gradual degradation in the performance as the bed lengths became more and more unsymmetrical.

At very high product amounts it was found that the optimum conditions were for beds with slightly different lengths. The most likely explanation is that it is better to have one bed of the right length than none.
However, increasing the size of the bigger bed further had a detrimental effect on the product concentration, as the smaller bed was no longer capable of purging the large one completely.

4) Adsorption Parameters

Changing the adsorption parameters gives the expected results that as the selectivity of the adsorbent for nitrogen over oxygen is increased the product concentration increases. Adjusting one of the adsorption parameters affects the product concentration fairly linearly across the entire range from maximum product purity down to that of the feed purity.

However, keeping the selectivity of the process the same whilst increasing the total adsorption seems to have little or no effect on the product concentration. As the total adsorption goes up the amount of gas required to pressurise the bed is increased, thus raising the amount of power the cycle requires.

This chapter has shown the effects of changing various parameters that are thought to have a major impact on the performance of a PSA process. The ability of the program to model each adsorbent bed separately has given the opportunity to study the effect of changing parameters, such as different bed lengths, never previously investigated.

Finally, it was shown that the product concentration has a reasonably linear response with respect to changing a single adsorption parameters. This effect will be used as the basis of the work presented in Chapter 7.
Chapter 7
Control Experiments

7.1 Introduction

This chapter investigates the dynamics and control of a non-linear process model to improve the control of the PSA pilot plant. The aims of this work are to show that the model is stable with respect to altering fundamental process parameters whilst running, that a basic control method can be applied, and that the basic control method applied is capable of rejecting disturbances. The experiments were carried out using profiles generated by the model itself. This was done to save repeatedly running the PSA pilot plant for two reasons: it speeded the experiments up (as the model runs many times faster than real-time); it removed the possibility of introducing subtle differences between the experiments when comparing the effect of tuning parameters etc. Any stored profile from a previous model run could be re-entered into the model as though it had come from the pilot plant in real-time, albeit at a faster rate.

For a number of cases, where it would be impractical to continuously alter parameters of interest, experiments such as these could only be done using results created by another simulation run or another model running simultaneously. For example, it would be extremely difficult to repeatedly alter the adsorption parameter of the pilot plant adsorbent in order to see if the controller constants were general or specific to the adsorption parameter.

All product flow rates are stated in terms of product amount per standard cycle and will be written as 8.0 dm³/cycle, where cycle denotes the standard cycle of 180s duration.
7.2 Model Parameter Matching

The ability of the model to successfully simulate the process is highly dependent on the choice of the parameters. If these are 'natural' parameters then they can be found by experimental measurement, but 'synthetic' parameters must be inferred from experimental data, i.e. they cannot be measured directly. However, on occasion a natural parameter can effectively become a synthetic parameter, as in the case of an adsorption parameter changing over time due to irreversible adsorption or gradual contamination. The value may have been measured initially, but without continuous measurement will gradually become less accurate as time passes.

If any of the model parameters are in error then it is likely that the model will predict incorrectly, but this can be overcome by modifying the model parameters to move the model's output to be the same as that of the plant. This is easily done at steady state, but is more difficult when the process is changing continuously. As shown in Chapter 6, it is possible to successfully manipulate one of the adsorption parameters in order to adjust the product concentration, with the important benefit that the steady state response will be reasonably linear.

7.2.1 Method

The first step in the experiments was to show that feedback control could be used to adapt the adsorption parameter from its initial value to one that results in a product concentration the same as a ‘real’ process. A diagrammatic view of the controller is shown in Figure 7.1.

Note that the adsorption parameters are the only adjustable parameters in the ILE model due to its simplifying assumptions. However, adjustment of either or both of these parameters, whilst the model is running, causes a violation of the mass balance and so no mass balance data will be presented in this work. If mass balance data was required a model should be chosen that has an adjustable parameter which can safely be
varied without violating the mass balance, e.g. diffusivity. See also Appendix A2.4.9.1.

Several different experiments were run to find the optimal tuning constants for this controller:

1) The control algorithm used was purely proportional, i.e. the proportional action was used and the integral action was turned off. This was used to find the ultimate gain and the ultimate period.

\[ k_O = \bar{k}_O + K_c \cdot e(t) \]

where \( e = y_{O_2, base} - y_{O_2, experiment} \)

and \( y \) is a mole fraction, and \( K_c \) is dimensionless.

2) The control algorithm was proportional and integral, developed from initial constants determined from the above experiment. This gave the optimum tuning constants.

\[ k_O = \bar{k}_O + K_c \left( e(t) + \frac{1}{T_i} \int e(t) \, dt \right) \]

3) The control was made by a proportional and integral controller, using the optimum constants determined from experiment 2), but using a second
computer generated profile (using different process conditions). This was designed to test whether the controller constants were general or specific to the original conditions.

The model was run once in order to generate a baseline profile for the cycle being considered. This profile was then re-entered into the model as though happening in real-time, allowing model settings to be changed each time whilst ensuring 100% repeatability of the inputted “experimental” data.

The conditions for this base case were: total cycle time 180s, a product amount of 8 dm$^3$/cycle, a purge amount of 17.2dm$^3$ ppc (equivalent to a 15.0s purge step and a 15.0s null step), and a value for $k_o$ of 3.27. For all the simulations, for the first 1440s (8 cycles) the product valves at the end of the beds remain closed. During this period the only product released from a bed is purge gas donated to the other bed.

The main adjustable parameter was chosen to be the adsorption parameter for oxygen, $k_o$. As the process is designed to generate oxygen as the product it seemed sensible to choose the parameter that most directly affected how much of the product gas would be adsorbed. However, this was an arbitrary choice and, as shown in Chapter 6, it could also have been the nitrogen parameter, $k_N$, so this facility was also included in the model. It was hoped that the model would have a reasonably linear and quick response to the manipulation of this parameter. This controller is also started at 1440s.

The intention was to use a Ziegler-Nichols (1942,1943) type tuning method to find the controller constants (the method of ultimate gain & period was chosen rather than the decay ratio method).

### 7.2.2 Proportional Only Control

A proportional only controller was used. Experiments were carried out to find the ultimate gain and the ultimate period. The conditions for the simulation runs are given in Table 7.1.
<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Initial $k_0$</th>
<th>Final $k_0$</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>1.0</td>
<td>2.943</td>
<td>3.09</td>
<td>7.2</td>
</tr>
<tr>
<td>2002</td>
<td>2.0</td>
<td>2.943</td>
<td>3.18</td>
<td>7.2</td>
</tr>
<tr>
<td>2003</td>
<td>4.0</td>
<td>2.943</td>
<td>3.17</td>
<td>7.2</td>
</tr>
<tr>
<td>2004</td>
<td>8.0</td>
<td>2.943</td>
<td>3.40</td>
<td>7.3</td>
</tr>
</tbody>
</table>

*Table 7.1 - Conditions for Proportional Only Controller*

### 7.2.2.1 Results & Discussion

It was discovered early in the experiments that by allowing the value of the oxygen adsorption parameter to be manipulated without any bounds resulted in the model breaking down. Initially this was thought to be purely due to the oxygen adsorption parameter being made zero or negative, however model breakdown would also occur if the oxygen adsorption parameter became very close to, or greater than that for nitrogen. Therefore, the value of the oxygen adsorption parameter was restricted to being less than 90% of the nitrogen value and greater than 50% of the initial oxygen adsorption parameter, i.e. $0.9k_N > k_o > 0.5k_{oi}$

In Figure 7.2 the initial higher peaks of the results above the base line are due to the model being started with a lower adsorption parameter than that of the base line. This results in the model building a higher product concentration within the beds before product release is started.

It can be seen that as the gain is increased the offset is reduced. Conversely, as the gain is increased the large scale oscillations get worse. This is a normal characteristic of proportional only control. Although only proportional gain is being used the model is still being adapted by feedback control towards the stored profile. An important feature is that the model seems to be tolerant of having the oxygen adsorption parameter changed dynamically, but using proportional control alone results in an unacceptable offset between the model and the stored profile.
Figure 7.2 - Model Matching Using Proportional Controller

Figure 7.3 - Model Matching Using PI Controller (Ziegler-Nichols Constants)
The gain was increased until stable oscillations occurred. From Figure 7.3 the ultimate gain was determined to be 8.0 and the ultimate period 1800s. This gave the tuning parameters, as calculated by the Ziegler-Nichols ultimate gain method, to be as shown in Table 7.2. The values for a PI controller were then used and the result is also shown on Figure 7.3. Having required a total of six simulations the result was reasonable, requiring only ≈8000s (≈45 cycles) to match up with the stored profile.

<table>
<thead>
<tr>
<th>Controller Type</th>
<th>Gain</th>
<th>Integral</th>
<th>Derivative</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>8.0</td>
<td>-</td>
<td>-</td>
<td>7.3</td>
</tr>
<tr>
<td>P</td>
<td>0.50</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PI</td>
<td>0.45</td>
<td>1500</td>
<td>-</td>
<td>7.3</td>
</tr>
<tr>
<td>PID</td>
<td>0.6</td>
<td>900</td>
<td>225</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.2 - Ziegler-Nichols Tuning constants

Using the Ziegler-Nichols ultimate gain method to estimate the tuning constants gives a reasonable set of values that allows the model to match up with the stored profile in ≈45 cycles. As the stored profile takes ≈25 cycles to come to steady state this is a good initial result. The Ziegler-Nichols ultimate gain method normally only gives a reasonable initial estimate of the tuning parameters (except in the case of a pure first order process) and further tuning is usually required to improve the initial values.

There is a typical linear response from the model shown in Figure 7.3 as the oscillations are of equal size (around the value of the set point plus the offset). However, in Figure 7.2 a value of $K_c$ of 1.0 gives a more oscillatory response than a $K_c$ of 2.0, a sign of a non-linear process. This implies that changing the oxygen adsorption parameter may not always have a linear effect on the final process concentration.
7.2.3 Proportional-Integral Control

A proportional plus integral action controller was used. The conditions for the simulation runs were as shown in Table 7.3. The stored profile was the same as used in Section 7.2.2.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Integral</th>
<th>Initial $k_o$</th>
<th>Final $k_o$</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>1.0</td>
<td>2000</td>
<td>2.943</td>
<td>3.30</td>
<td>7.4</td>
</tr>
<tr>
<td>2012</td>
<td>1.0</td>
<td>1000</td>
<td>2.943</td>
<td>3.31</td>
<td>7.4</td>
</tr>
<tr>
<td>2013</td>
<td>1.0</td>
<td>500</td>
<td>2.943</td>
<td>3.38</td>
<td>7.4</td>
</tr>
<tr>
<td>2021</td>
<td>2.0</td>
<td>2000</td>
<td>2.943</td>
<td>3.26</td>
<td>7.5</td>
</tr>
<tr>
<td>2022</td>
<td>2.0</td>
<td>1000</td>
<td>2.943</td>
<td>3.28</td>
<td>7.5</td>
</tr>
<tr>
<td>2023</td>
<td>2.0</td>
<td>500</td>
<td>2.943</td>
<td>3.40</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Table 7.3 - Conditions for PI-type Control Runs

It is expected that the addition of the integral term into the controller will improve the response, as shown in Figure 7.2 for the Ziegler-Nichols results. However, this may come at the expense of making the process more oscillatory.

7.2.3.1 Results & Discussion

Looking at Figures 7.4 & 7.5 the model has adapted much better to the stored profile, and has managed to match up correctly with the stored profile within only ≈6000s (≈33 cycles). The offset has been removed in all cases. There are two sets of tuning parameters that appear to give very good results, runs 2012 & 2021.
Figure 7.4 - Model Matching Using PI-type Control (gain=1.0)

Figure 7.5 - Model Matching Using PI-type Control (gain=2.0)
Figure 7.6 - Model Matching Using PI-type Control
With proportional gain and integral action being used the model adapts very rapidly to follow the stored profile. However, adding the integral term to the controller has created more instability in the product concentration, i.e. as more integral action is added the more the product concentration is likely to oscillate.

Looking at Figure 7.6, where the Ziegler-Nichols tuning constants are compared with those determined by experiment shows that although the Ziegler-Nichols values are a reasonable initial estimate it is possible to reduce the time taken to match to the stored profile by 30% by a careful parametric search on the tuning parameters. Obviously by using a stored profile, the repeated testing of tuning constants is speeded up enormously and in this case has provided a significant performance benefit at minimal cost.

7.2.4 Proportional-Integral Control, Different Conditions

To determine if the tuning constants can be considered generic over a range of conditions the model was re-run with different operating conditions. Base case A had 15s purge time (the base case as described in Section 7.2.1) and for base case B the purge time was reduced to 10s of purge time and the null time increased to 20s. For this cycle, as the amount of time allowed for purging is decreased, and hence the purge amount per part cycle, the lower the product concentration will be.

The tuning constants were those determined in Section 7.2.3. The base case conditions of set A were used for run 2000 & 2021 and conditions of set B were used in runs 2030 & 2031. Apart from the length of the purge and null steps all other parameters were unchanged.
## 7.2.4.1 Results & Discussion

The results are compared in Figure 7.7a (A conditions) and Figure 7.7b (B conditions). The profile for B conditions took longer to match up with the stored profile by approximately 900s (5 cycles).

The final value of the oxygen adsorption parameter was slightly different for the run with B conditions than used in the base case. Whereas the run with A conditions gave a parameter almost identical to that used in the original base simulation, the run for B conditions gave a parameter that was low by 1%.

The tuning constants determined in Section 7.2.3 behave reasonably well over a range of operating conditions. However, it seems likely that they will be optimal only for the one particular set.

The fact that a different value of the oxygen adsorption parameter was obtained for B conditions yet the same product concentration was achieved may suggest that the PSA process has a memory of what has happened before, i.e. its history affects its current operation. This is important because it means that more than one set of conditions can give the same steady state and conversely that more than one steady state is possible for one set of conditions, depending on the history of the process. Further investigation of this effect was outside the scope of this project, however some more examples of this effect can be found in Appendix 4.

### Table 7.4 - Conditions for PI-type Control Runs

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Purge</th>
<th>Gain</th>
<th>Integral</th>
<th>Initial $k_0$</th>
<th>Final $k_0$</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>15s</td>
<td>-</td>
<td>-</td>
<td>3.270</td>
<td>-</td>
<td>7.7a</td>
</tr>
<tr>
<td>2021</td>
<td>15s</td>
<td>2.0</td>
<td>2000</td>
<td>2.943</td>
<td>3.26</td>
<td>7.7a</td>
</tr>
<tr>
<td>2030</td>
<td>10s</td>
<td>-</td>
<td>-</td>
<td>3.270</td>
<td>-</td>
<td>7.7b</td>
</tr>
<tr>
<td>2031</td>
<td>10s</td>
<td>2.0</td>
<td>2000</td>
<td>2.943</td>
<td>3.24</td>
<td>7.7b</td>
</tr>
</tbody>
</table>
The fact that it takes time for a change in product gas leaving the process to pass through the product receiver (a first order process) and down the product line (a pure time delay) means there is a considerable time delay before a change in its output is seen at the analyser used to control the oxygen adsorption parameter. However, without the product receiver to mix the product gas, the product concentration swings from close to its maximum purity of 95% oxygen down to 36% twice per cycle.

In Figure 7.7c the input to the product receiver (which was averaged on a 180s basis to remove the large oscillations in product concentration that leaves the beds on this cycle) together with the output from the product receiver are shown. This indicates that the process has actually responded much faster than it appears from the output, initially dropping its product purity down from the maximum 95% to below that finally required in order to dilute the contents of the product receiver.

This implies that the product receiver should be no larger than required to damp the large concentration changes coming out of the beds to an acceptable oscillation in final product concentration. This will improve the response of the process to changes in product concentration being released from the beds as it is manipulated. On the PSA pilot plant, the volume of 20dm³, as used in all the simulations, is probably too large for the flow rate of product gas used in these cases. However, the pilot plant was not designed to have product purity control, in which case having a larger product receiver acts as an effective buffer against minor product flow rate and composition changes.
Figure 7.7a - Model (condition A) Matched Using a PI Controller (gain=2.0, integral=2000)

Figure 7.7b - Model (condition B) Matched Using a PI Controller (gain=2.0, integral=2000)
Figure 7.7c - Model (condition A) Matched Using a PI Controller, Showing Direct Process Output
7.3 Checking Process-Model Mismatch

It is important that small errors between parameters in the model and the physical process do not significantly impact the ability of the model to match the process. If the adaptation of the model cannot cope with the introduction of highly controlled errors from another simulation it is very unlikely that it will be able to cope with the inevitable error from a real plant.

In order to see the effect of introducing various errors into the model a set of simulations was run, in each case the model was given incorrect information about the physical 'plant'. Again, the plant data was generated from a separate run of the model. Four different experiments were run in order to see these effects:

1) The purge flow rate in the model was wrong. As the purge flow rate is not normally measured on a PSA unit this would be an unmeasured error.
2) The product pipe length in the model was wrong. As the product pipe length is easily determined on a PSA unit this would be an unlikely error. However, it tests the model’s tolerance of any time delays being incorrect.
3) The product receiver volume in the model was wrong. As the product receiver volume is easily determined on a PSA unit this would be an unlikely error. However, it tests the model’s tolerance of a major time constant being incorrect.
4) The product flow rate in the model was wrong. As the product flow rate is usually measured on a PSA unit this would be a measurement error. In all four experiments the model was only allowed to adapt the oxygen adsorption parameter.

The conditions for this base case were: total cycle time 180s, a product amount of 8dm$^3$/cycle, a purge amount of 17.2dm$^3$ppc (equivalent to a 15.0s purge step and a 15.0s null step), and a value for $k_o$ of 3.27. Any other details of the experiments are given in the sections below.
7.3.1 Method

The model was run once for each case in order to generate a baseline profile for the mis-match being considered. This profile was then re-entered into the model. The same tuning constants were used throughout the experiments.

7.3.2 Incorrect Purge Flow

The purge flow rate was set at its base value ±20%. This gave the conditions as in Table 7.4. On a normal PSA plant it is unusual to have a flow meter in the purge line so there is potential error between the purge flow rate being used by the plant and that used by the model.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Initial $k_0$</th>
<th>Final $k_0$</th>
<th>Final Product (mf)</th>
<th>Purge rate (mol/s)</th>
<th>Purge Error (%)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>3.270</td>
<td>-</td>
<td>0.664</td>
<td>0.0529</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2041</td>
<td>3.270</td>
<td>3.30</td>
<td>0.668</td>
<td>0.0503</td>
<td>-5</td>
<td>7.8a</td>
</tr>
<tr>
<td>2042</td>
<td>3.270</td>
<td>3.36</td>
<td>0.673</td>
<td>0.0556</td>
<td>+5</td>
<td>7.8a</td>
</tr>
<tr>
<td>2043</td>
<td>3.270</td>
<td>3.27</td>
<td>0.675</td>
<td>0.0476</td>
<td>-10</td>
<td>7.8b</td>
</tr>
<tr>
<td>2044</td>
<td>3.270</td>
<td>3.35</td>
<td>0.654</td>
<td>0.0582</td>
<td>+10</td>
<td>7.8b</td>
</tr>
<tr>
<td>2045</td>
<td>3.270</td>
<td>3.17</td>
<td>0.664</td>
<td>0.0450</td>
<td>-15</td>
<td>7.8c</td>
</tr>
<tr>
<td>2046</td>
<td>3.270</td>
<td>3.41</td>
<td>0.661</td>
<td>0.0609</td>
<td>+15</td>
<td>7.8c</td>
</tr>
<tr>
<td>2047</td>
<td>3.270</td>
<td>3.17</td>
<td>0.664</td>
<td>0.0423</td>
<td>-20</td>
<td>7.8d</td>
</tr>
<tr>
<td>2048</td>
<td>3.270</td>
<td>3.41</td>
<td>0.661</td>
<td>0.0635</td>
<td>+20</td>
<td>7.8d</td>
</tr>
</tbody>
</table>

Table 7.4 - Incorrect Purge Flow Rates

7.3.2.1 Results & Discussion

The results for this experiment are shown in Figures 7.8a-d. In almost all cases the model struggled to reach a true steady state, however on average it did match the stored profile closely for all cases. The model seems
to oscillate around the desired product concentration, but was never more than 4% in error.

The occasional oscillations appear to be the result of small changes accumulating in the position of concentration fronts. As the product concentration drifts away from the set point it is then corrected by the oxygen adsorption parameter being adjusted, thus adjusting the positions of the fronts on subsequent cycles. This process then appears to repeat itself endlessly.

Finally, the process is again showing non-linear behaviour. At the start of the simulation, making the error in purge flow rate to one bed positive or negative increases the product concentration. As the error in the purge flow rate is increased the changes in the product concentration become larger, until Figure 7.8d when the changes start to lessen. Changing the purge flow rate whilst keeping the purge time constant is equivalent to altering the purge amount and, as shown in Section 6.3.3, increasing the error in the purge amount can have the effect of increasing the product concentration in some circumstances. Comparing the conditions for the graphs shown in Figures 7.8a-d with Figure 6.14 (which has the same flow rate), the results for the 17.2dm³ of purge per part cycle show that we would expect an increase in product concentration up to at least ±10% purge flow rate error. Despite this the model has “survived” substantial errors in purge flow rate.

7.3.3 Incorrect Product Pipe Length

The product pipe length was set to various lengths. This gave the conditions as in Table 7.5. The pipe introduces a time delay into the response of the model to a change in the product concentration from the beds, assuming plug flow as in the model. Each 5m of pipe length adds ≈95s to the time delay. The initial pipe length of 10m gave approximately one whole cycle time (180s) to the delay.
Figure 7.8a - Model Matching With 5% Purge Rate Error

Figure 7.8b - Model Matching With 10% Purge Rate Error
Figure 7.8c - Model Matching With 15% Purge Rate Error

Figure 7.8d - Model Matching With 20% Purge Rate Error
Various lengths of pipe were chosen to give time delays that would force the product concentration to be out of phase with the base case profile, either by half, three quarters, or a whole cycle. These are given in Table 7.5.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Initial $k_o$</th>
<th>Final $k_o$</th>
<th>Final Product (mf)</th>
<th>Pipe Length (m)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>3.270</td>
<td>-</td>
<td>0.664</td>
<td>10</td>
<td>7.9</td>
</tr>
<tr>
<td>2051</td>
<td>3.270</td>
<td>3.28</td>
<td>0.663</td>
<td>15</td>
<td>7.9</td>
</tr>
<tr>
<td>2052</td>
<td>3.270</td>
<td>3.32</td>
<td>0.663</td>
<td>17.5</td>
<td>7.9</td>
</tr>
<tr>
<td>2053</td>
<td>3.270</td>
<td>3.31</td>
<td>0.662</td>
<td>25</td>
<td>7.9</td>
</tr>
<tr>
<td>2054</td>
<td>3.270</td>
<td>3.32</td>
<td>0.661</td>
<td>30</td>
<td>7.9</td>
</tr>
<tr>
<td>2055</td>
<td>3.270</td>
<td>3.28</td>
<td>0.663</td>
<td>5</td>
<td>7.9</td>
</tr>
</tbody>
</table>

Table 7.5 - Incorrect Product Pipe Length

7.3.3.1 Results & Discussion

With the introduction of a time delay into the system the model was still stable. In Figure 7.9 it can be seen that the longer the length of pipe the greater the model overshoot and the time taken to settle at the target concentration. The overshoot is caused from the large initial error between the target composition and the model. This causes a large controller response and so the value of $k_o$ falls rapidly at this point. The integral term then builds up, keeping the value of $k_o$ low, although the product concentration has now started to respond. The longer the time delay the longer the response in the product concentration and the more the controller over-corrects.

The response for a shorter pipe length is completely different as the concentration from the model arrives before that of the plant. Therefore the model composition is very close to the target value as soon as the controller
is started, thus preventing the controller from over-correcting. This causes the model to match up with the profile very rapidly. Although it seems attractive to set the product pipe on the model to be shorter than the plant this could lead to problems in the case of disturbances, as the response from the model will arrive before that of the plant.

As expected, when the product peaks were in phase with the base case the model was stable. Delaying the product peaks by a half or entire cycle appeared to make no difference to the stability of the model. Interestingly, even when the pipe length was such that the product concentration peaks arrived out of phase (with the waves from the plant and model in almost perfect opposition) the model was still stable and the average product concentration was still correct.

At extremely long pipe lengths, and hence long time delays, the model was still able to cope with the time delay. The change from the base case of 10m to 30m of pipe still gave a stable model, although the time to match up to the base profile was now about 1½ hours.

Despite the problems introduced by long time delays the model was still stable. However, with these long delays any transients entering the process will not be seen until much later and the response of the model will be much slower. This could leave the model adapter responding to a disturbance which may have already gone. Conversely, artificially reducing the length of the pipe in the model can lead to disturbances in the model arriving before those of the plant. To reduce the effects of this on a real plant the product purity analyser should be located as close as possible to the outlet of the product receiver.
Figure 7.9 - Model Matching With Product Pipe Length Error
7.3.4 Incorrect Product Receiver Volume

In Section 7.3.2.1 it was noted that the product receiver volume should be kept as small as possible to improve the response of the model. To investigate the effect of the product receiver volume on the response of the model the volume was varied from 50% to 125% of the base case value (20dm³, as on the pilot plant).

The volumes used are shown in Table 7.6. However, these are extreme values, it is unlikely that on a real plant the error in the size of this vessel would be more than 6% (based on a 2% tolerance in the vessels dimensions).

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Initial k₀</th>
<th>Final k₀</th>
<th>Final Product (mf)</th>
<th>Receiver Volume (dm³)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>3.270</td>
<td>-</td>
<td>0.664</td>
<td>20</td>
<td>7.10</td>
</tr>
<tr>
<td>2061</td>
<td>3.270</td>
<td>3.32</td>
<td>0.661</td>
<td>10</td>
<td>7.10</td>
</tr>
<tr>
<td>2062</td>
<td>3.270</td>
<td>3.29</td>
<td>0.663</td>
<td>15</td>
<td>7.10</td>
</tr>
<tr>
<td>2063</td>
<td>3.270</td>
<td>3.29</td>
<td>0.663</td>
<td>25</td>
<td>7.10</td>
</tr>
</tbody>
</table>

Table 7.6 - Incorrect Product Receiver Volume

7.3.4.1 Results & Discussion

In Figure 7.10 it can be seen that as the product receiver volume is changed the dynamics of the process are affected. The main effect of reducing the receiver volume was to decrease the time constant of the process, however this also increased the height of the initial peaks. The overshoot for the process with a 10dm³ receiver was the largest, but the response was still good, although it is now oscillatory. The steady state response for all the volumes was good, this is to be expected as the product receiver has almost no effect once the product concentration has reached steady state.
The initial peak is built by the start up procedure for the process. Because there is no product release from the beds during the first 5 cycles (1440s) the oxygen concentration rapidly builds up in the beds. When the product gas is first released it must pressurise the product receiver, the oxygen rich gas from the bed is mixed with the lower quality gas in the receiver. The smaller the volume of the product receiver, the higher the initial peak will be. As noted in Section 7.3.2.1 this volume should therefore be kept as small as possible to improve the response of the model.

Finally, adjusting the size of the product receiver affected only the height of the concentration peaks leaving the receiver and had no effect on the timing of the peaks or the average product concentration. Regardless of the receiver volume the concentration peaks were coincident with the base case profile. The changing size of the concentration peaks caused the model to oscillate the oxygen adsorption parameter slightly, however it was still stable.

**7.3.5 Incorrect Product Flow Rate**

To test the effect of having the wrong product flow rate on the ability of the model to match the base profile, the flow rate in the model was set to be both too high and too low, as shown in Table 7.7.
<table>
<thead>
<tr>
<th>Run ID</th>
<th>Initial $k_o$</th>
<th>Final $k_o$</th>
<th>Final Product (mf)</th>
<th>Product Flow Rate (dm$^3$/cycle)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>3.270</td>
<td>-</td>
<td>0.664</td>
<td>8.0</td>
<td>7.11</td>
</tr>
<tr>
<td>2071</td>
<td>3.270</td>
<td>3.45</td>
<td>0.663</td>
<td>7.0</td>
<td>7.11</td>
</tr>
<tr>
<td>2072</td>
<td>3.270</td>
<td>3.45</td>
<td>0.663</td>
<td>6.0</td>
<td>7.11</td>
</tr>
<tr>
<td>2073</td>
<td>3.270</td>
<td>3.25</td>
<td>0.663</td>
<td>9.0</td>
<td>7.11</td>
</tr>
</tbody>
</table>

Table 7.7 - Incorrect Product Flow Rate

7.3.5.1 Results & Discussion

As shown in Figure 7.11, using the wrong product flow rate in the model appeared to make no difference to the ability of the model to match the final value of the base profile. The time to reach steady state is about half that of the results in Figure 7.10, and gets progressively slower as the product rate decreases.

Reducing the product flow rate had the effect of increasing the residence time of the product receiver and the time delay from the product pipe, thus increasing the total time delay of the process. However, the model seemed stable even when the product flow rate was reduced to only 50% of the base profile value.

Increasing the product flow rate had virtually no effect. As seen in Section 7.3.3, decreasing the pipe length improved the response of the model. Increasing the product flow is effectively the same as reducing the length of the product pipe as both reduce the time delay of the system. As there is virtually no difference between the model concentration and the base profile only a small control action is required to match the two.
Figure 7.11 - Model Matching With Product Product Flow Rate Error
7.3.6 Conclusions

The ability to adapt the model using feedback control to match a stored profile is very good. For a model started with an incorrect adsorption parameter matching normally occurs within 7200s (2hrs or 40 cycles) of the process start time. The final adsorption parameter is usually very close to that used to generate the stored profile. When a model is started with the correct value for the adsorption parameter but the wrong value for another parameter, such as product receiver volume, the model again adjusts itself successfully to match the stored profile.

In some cases, such as for an incorrect product flow rate, the adsorption parameter in the model settles to the same value used to generate the stored profile, however the model is now operating with a different product flow rate, yet producing the same product concentration.

The model seems stable when dynamically changing the oxygen adsorption parameter to match the product concentration to the stored profile. The program has shown itself to be robust when dynamically adjusting an adsorption parameter, over the ranges investigated, and the feedback tuning constants appear to be reasonably independent of the cycle conditions.

The pilot plant was originally designed to be capable of supplying purge gas from the product receiver in order to investigate the effects of using well-mixed purge gas as opposed to gas delivered directly from another bed. As the work in this project was done exclusively with directly purged cycles there is some evidence that the volume of the product receiver on the pilot plant (20 dm³) is too large in these circumstances. This has two major effects:

i) the time for the product concentration to reach its steady state value will be increased, and
ii) the response of any experiment that involves injecting disturbances or manipulating the process will suffer from a larger time constant than necessary.

However, for an uncontrolled plant the use of a larger product receiver will help to smooth out minor disturbances in product flow rate and concentration.

7.4 Process Control

One aim of this project was to introduce control using an internal value of the model that would be difficult or expensive to measure in a commercial plant to better control the disturbance response of a PSA plant. This section shows how manipulating one parameter, the duration of the null step, based on the internal concentration fronts within the beds can lead to good disturbance rejection. The control scheme is illustrated in Figure 7.12.

![Figure 7.12 - Schematic of Model and Process Controller](image)

Reducing the null step time reduces the product release on the other bed by the same amount, thus reducing the overall cycle time. An alternative
approach would be to adjust the purge step time to compensate, however this would also require an adjustment of the purge flow rate to maintain the same purge amount per cycle. As the ability to dynamically adjust the purge flow rate is not normally available on a commercial PSA plant this option was not chosen.

Note that a standard cycle was defined in Section 7.1 as being in 180s in duration and is denoted by cycle. All cycles so far have been of this standard length. For comparative purposes, all flow rates within this section will be expressed in the same form, however it is important to remember the true cycle time will now be variable.

7.4.1 Controlling Product Concentration

The base case chosen was similar to that used in the previous work, however at a slightly different product concentration. The cycle had 15s of purge time and 15s of null time, as illustrated in Figure 7.13.

<table>
<thead>
<tr>
<th>Bed A</th>
<th>PD</th>
<th>DP</th>
<th>PU</th>
<th>NU</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bed B</td>
<td>DP</td>
<td>PU</td>
<td>NU</td>
<td>PR</td>
<td></td>
</tr>
<tr>
<td>0s</td>
<td>30s</td>
<td>45s</td>
<td>60s</td>
<td>90s</td>
<td>120s</td>
</tr>
</tbody>
</table>

Figure 7.13 - Base Case Cycle Steps

The model had a 20dm³ product receiver and the product concentration was measured at the end of the product pipe (10m). A ‘marker’ is defined as being the point within the model beds that the concentration fronts are measured at. In this work the marker was specified as being at the outlet of the product end of the beds, i.e. at z = L (0.835m). All dimensions in the model were set to match those on the pilot plant.
The controller acts during the product release step of one bed by adjusting the duration of the null step on the other bed, based on the time when the main concentration front passed the marker. The controller algorithm can thus be expressed as:

\[ t_{NU} = \tilde{t} + K_c \left( e(t) + \frac{1}{T_i} \int e(t) dt \right) \]

where \( e = y_{O_2sp} - y_{O_2prod} \), \( y \) is a mole fraction, and \( \tilde{t} \) is the initial estimate of the null step time required for the desired product purity. In this case \( K_c \) is no longer dimensionless, having units of s.

Cycle types can be divided into two distinct types, based on the product purity:
1) those cycles which can achieve the maximum product purity, and,
2) cycles that have breakthrough of low quality product gas during the product release step and so cannot achieve the maximum purity.

By basing the duration of the null step on the time the front passes the marker, rather than on, for example, the start of the cycle for the bed, it was hoped that the controller would respond more rapidly to changes in the internal concentration fronts that occur when a disturbance is introduced into the system.

The final part of the controller was a 'cut-off' switch. This was added with the intention that it would allow the controller to rapidly shorten the null time if the concentration front had already passed the marker before the start of the null step and the product concentration was too low.

Finally, it is important to note that the controller is only active whilst a bed is within a null step, thus limiting control action to twice per cycle, and whereas in the previous work the cycles were all of fixed length, in the following sections the total cycle time is variable.
7.4.1.1 Proportional Only Control

The conditions were as shown in Table 7.8. Each run was allowed to steady out first and then the product concentration controller was started. The set point of the product concentration controller was always such that by adjusting the null step duration it was possible for the controller to find a solution.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Initial Null time (s)</th>
<th>Final Null Time (s)</th>
<th>Set Point (mf)</th>
<th>Final Value (mf)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3001</td>
<td>500</td>
<td>15s</td>
<td>8.6</td>
<td>0.766</td>
<td>0.754</td>
<td>7.14</td>
</tr>
<tr>
<td>3002</td>
<td>1000</td>
<td>15s</td>
<td>6.0</td>
<td>0.766</td>
<td>0.757</td>
<td>7.14</td>
</tr>
<tr>
<td>3003</td>
<td>500</td>
<td>15s</td>
<td>33.4</td>
<td>0.622</td>
<td>0.658</td>
<td>7.14</td>
</tr>
</tbody>
</table>

Table 7.8 - Conditions for P-type Control of Product Concentration

7.4.1.2 Results & Discussion

In Figure 7.14 the set-points are indicated by dotted lines. It can be seen that there was always an offset between the set point and the final value. This is expected for a controller that has no integral action. As the gain was increased oscillations in the product concentration appeared, as would be expected. The offsets for the two set points were not identical, the result for the lower set point had an offset around three times larger than the same controller settings gave with the higher setting.

To help understand the action of the controller, the requested null step times were plotted, as shown Figure 7.15, for the two controllers with the higher set point. When the controllers first act they immediately attempt to reduce the null step time from 15s to a negative value (the controller is obviously limited to a minimum null step time of zero seconds), as the current composition is below the set point. As the composition nears the set
point the null step time is gradually increased until a stable value is found, around 10s.

When the integral action was doubled it initially caused the system to behave similarly to the previous control settings, although the control action was more pronounced. After the composition neared the set point the system started to oscillate. Originally it was thought that the null step duration would either be different for each bed or would oscillate between two different values from cycle to cycle. However, it was found that the duration of the null steps oscillated from cycle to cycle.
Figure 7.14 - Product Controlled Using Proportional Only Control

- base case
  - $K_c=500$, set point = 0.766
  - $K_c=1000$, set point = 0.766
  - $K_c=500$, set point = 0.622
Figure 7.15 - Controller Output for Proportional Only Control
7.4.1.3 Proportional Plus Integral Control

The conditions were as shown in Table 7.9. The procedure followed was the same as in Section 7.4.1.1.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Integral</th>
<th>Initial Null Time (s)</th>
<th>Final Null Time (s)</th>
<th>Set Point (mf)</th>
<th>Final Value (mf)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3010</td>
<td>500</td>
<td>20000</td>
<td>15.0</td>
<td>7.6</td>
<td>0.766</td>
<td>0.757</td>
<td>7.16</td>
</tr>
<tr>
<td>3011</td>
<td>500</td>
<td>10000</td>
<td>15.0</td>
<td>6.9</td>
<td>0.766</td>
<td>0.760</td>
<td>7.16</td>
</tr>
<tr>
<td>3012</td>
<td>500</td>
<td>5000</td>
<td>15.0</td>
<td>6.1</td>
<td>0.766</td>
<td>0.763</td>
<td>7.16</td>
</tr>
<tr>
<td>3013</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>6.0</td>
<td>0.766</td>
<td>0.766</td>
<td>7.16</td>
</tr>
</tbody>
</table>

Table 7.9 - Conditions for Proportional Plus Integral Control of Product Concentration

7.4.1.4 Results & Discussion

The results are shown in Figure 7.16a where it is observed that addition of the integral action term into the controller does not completely eliminate offset, as in Figure 7.13. Comparing the ratio of gain to integral used for matching the oxygen parameter with that required for product concentration control shows that the integral term is much more important in the second case. This is because the response of the model to a change in the duration of the null step is much slower than that for a change in oxygen adsorption parameter.

An expanded view is shown in Figure 7.16b. The response is still stable with an integral action as low as 1000, although it shows overshoot, and so this was selected as the optimum value. With these gain and integral action values the oscillatory behaviour, which had been found in Section 7.4.1.1, was not seen. If this oscillating behaviour is seen in later experiments then an integral value of 5000 could be used, as the response for this value was also good, although it had a small offset.
7.4.2 Discussion

The results presented in Figures 7.14 and 7.16 were exactly as hoped for. Without integral action the control was stable but had an offset and with the addition of the integral term this was almost completely eliminated. The control of the PSA system seems to be stable for all conditions tried.

There were some unexpected features of the control which were shown in Figure 7.14. The behaviour of the system when the controller forced the null step times to oscillate was unexpected but still stable overall. The asymmetry in the oscillations are probably partly due to the non-linearity of the process and partly due to the fact if the front passed the required point before the controller was operating then the controller would immediately end the null step if the product concentration was too low.

The optimum values for the controller were selected as being a gain of 500 with an integral action of 1000. However, in the event of oscillatory behaviour being seen the alternative integral value of 5000 would be a reasonable alternative.
Figure 7.16a - Product Controlled Using P+I Control (gain=500)

Figure 7.16b - Product Controlled Using P+I Control (gain=500)
7.4.3 Ramp Change Disturbance Rejection

In the previous section it was shown that a controller using an internal bed concentration prediction could be tuned to give almost exactly the desired product composition. However, one vital part of any controller is its ability to reject disturbances, whether externally or internally generated. In order to test the response of the product concentration controller to a disturbance, the product flow rate was selected since it is one of the most likely cause of a disturbance. This type of disturbance is common in industry, where a customer wishes to change the throughput of their plant at will.

Product flow rates were selected such that for the uncontrolled case when the product flow rate was changed a significant change in product concentration would occur. The controller would then be expected to maintain the product concentration at the desired set point.

A ramped change in product flow rate is common in industrial applications. To test the response of the controller to this effect the product rate was ramped at a rate of $\pm 0.5\text{dm}^3/\text{cycle}$ until the required rate was reached, as illustrated in Figure 7.17.

![Figure 7.17 - Ramp Change Disturbance](image)

Figure 7.17 - Ramp Change Disturbance
7.4.3.1 Conditions

The conditions were as shown in Table 7.10. The product concentration set point was the same as the final value for the base case (3000), i.e. a mole fraction of oxygen of 0.71.

For each test the proportional gain was set at 500 and the integral action time to a value of 1000, as determined in Section 7.4.1. The start of the disturbance was at 5400s, and the constant ramp rate was maintained until the desired product flow rate was achieved.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Integral</th>
<th>Initial Null Time (s)</th>
<th>Final Null Time (s)</th>
<th>Product Amount (dm$^3$/cycle)</th>
<th>Final Value (mf)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>8.0</td>
<td>0.709</td>
<td>-</td>
</tr>
<tr>
<td>3140</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>6.0</td>
<td>0.835</td>
<td>7.18a</td>
</tr>
<tr>
<td>3142</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>49.0</td>
<td>6.0</td>
<td>0.708</td>
<td>7.18a</td>
</tr>
<tr>
<td>3150</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>4.0</td>
<td>0.946</td>
<td>7.18b</td>
</tr>
<tr>
<td>3152</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>95.8</td>
<td>4.0</td>
<td>0.702</td>
<td>7.18b</td>
</tr>
<tr>
<td>3160</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>10.0</td>
<td>0.657</td>
<td>7.19a</td>
</tr>
<tr>
<td>3162</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>2.0</td>
<td>10.0</td>
<td>0.709</td>
<td>7.19a</td>
</tr>
<tr>
<td>3170</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>12.0</td>
<td>0.598</td>
<td>7.19b</td>
</tr>
<tr>
<td>3172</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>0.0</td>
<td>12.0</td>
<td>0.653</td>
<td>7.19b</td>
</tr>
</tbody>
</table>

Table 7.10 - Conditions for Product Rate Ramp Disturbance

7.4.3.2 Results & Discussion

The results for two different types of response for this experiment are shown in Figures 7.18 and 7.19. In Figures 7.18a & b the results are shown for a process that had its product flow rate reduced by 25% and 50% (to 6.0dm$^3$/cycle and 4.0dm$^3$/cycle respectively). Without the controller in
operation the product concentration rises rapidly. With the controller active the response to the disturbance in product flow rate is good, taking around 2700s to bring the product concentration back to its set point, with a maximum peak of only 0.05 to 0.07 above the set point.

In Figure 7.19a it can be seen that with an increase in product rate of 25% the controller has managed to keep the product concentration within 0.01 of the set point at all times. This can be compared with Figure 7.18a, where an equal flow rate change in the opposite direction produced a peak around 0.04 mole fraction higher than the set point.

This shows the non-linear behaviour of the process; for a linear process, irrespective of the sign, a disturbance always produces a response of equal magnitude. However, the control algorithm has managed to maintain control in both circumstances.

A flow rate increase of 25% to 10.0dm$^3$/cycle is the largest increase for which the controller can maintain the product at the set point. With a 50% disturbance the flow rate rises to 12.0dm$^3$/cycle and the controller reduces the null step time to zero. Although there was a final offset the loss in product purity was minimised and the response was rapid. The increase in product flow rate immediately caused the ‘cut-off’ switch to reduce the null step to zero seconds, giving a much more rapid response than the PI controller. Adjusting the size of the integral action had no effect on the response as the ‘cut-off’ switch was now controlling the null step time, as the concentration front was always passing out of the system before the null step started.
Figure 7.18a - -25% Ramp Flow Disturbance With and Without Product Control (Ti=1000)

Figure 7.18b - -50% Ramp Flow Disturbance With and Without Product Control (Ti=1000)
Figure 7.19a - +25% Ramp Flow Disturbance With and Without Product Control (Ti=1000)

Figure 7.19b - +50% Ramp Flow Disturbance With and Without Product Control (Ti=1000)
7.4.4 Step Change Disturbance Rejection

In Section 7.4.3 it was shown that a controller using an internal bed measurement could be used to maintain the desired product composition when a ramped disturbance in product flow rate was introduced.

Another common disturbance is a step change in product flow rate, as illustrated in Figure 7.20, where a very rapid change in product flow rate is required. This disturbance differs from that of a ramped disturbance in that the product flow changes almost instantaneously from the base case rate to the new rate in a single integration time interval.

![Figure 7.20 - Step Change Disturbance](image)

**7.4.4.1 Conditions**

The conditions were as shown in Table 7.11. The product concentration set point was the same as the final value for the base case (3000), i.e. a mole fraction of oxygen of 0.71, the same as in Section 7.4.3.
Table 7.11 - Conditions for Product Rate Step Disturbance

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Gain</th>
<th>Integral</th>
<th>Initial Null Time (s)</th>
<th>Final Null Time (s)</th>
<th>Product Amount (dm³/cycle)</th>
<th>Final Value (mf)</th>
<th>Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>8.0</td>
<td>0.709</td>
<td>-</td>
</tr>
<tr>
<td>3040</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>6.0</td>
<td>0.835</td>
<td>7.21</td>
</tr>
<tr>
<td>3042</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>-</td>
<td>6.0</td>
<td>0.709</td>
<td>7.21a</td>
</tr>
<tr>
<td>3043</td>
<td>500</td>
<td>5000</td>
<td>15.0</td>
<td>-</td>
<td>6.0</td>
<td>0.729</td>
<td>7.21b</td>
</tr>
<tr>
<td>3050</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>4.0</td>
<td>0.957</td>
<td>7.22</td>
</tr>
<tr>
<td>3052</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>94.6</td>
<td>4.0</td>
<td>0.702</td>
<td>7.22</td>
</tr>
<tr>
<td>3060</td>
<td>-</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>12.0</td>
<td>0.598</td>
<td>7.23</td>
</tr>
<tr>
<td>3062</td>
<td>500</td>
<td>1000</td>
<td>15.0</td>
<td>-</td>
<td>12.0</td>
<td>0.652</td>
<td>7.23</td>
</tr>
</tbody>
</table>

The disturbance was injected at 5400s, ensuring that in all cases the model had sufficient time to come to steady state at the target product concentration. For the uncontrolled process when the disturbance occurs the plant immediately moves from the base case towards a new product concentration.

7.4.4.2 Results & Discussion

In Figure 7.21a the result is shown for a process that had its product flow rate reduced by 25% to 6.0dm³/cycle. The response of the model to a step change is very similar to that of a ramped change, however the overshoot is slightly smaller and the disturbance in product concentration has been fully rejected in just under one hour.

In order to show that the response to an integral action value of 1000 was better than the alternative choice of integral action of 5000, the results with integral action of 5000 were also recorded. The result is shown in Figure 7.21b. It can be seen that although the initial peak is very similar to that
obtained with an integral action of 1000, the final response is much slower than before and there is a final offset of 0.02 mole fraction. The conclusion from this is that 1000 was the optimum integral term.

The test was repeated for a process that had its product flow rate cut to 4.0dm³/cycle, i.e. 50% of the base case rate. The result is shown in Figure 7.22, and again is very similar to the response for the ramped disturbance. The disturbance in product concentration has been fully rejected in just over one hour.

In each of the previous two tests the controller increased the duration of the null step in order to maximise the product delivered from a cycle of the beds. It is obvious that the null step can be lengthened as much as required in order to meet the product purity requirement. In Section 7.4.3, with a 50% increase in product flow rate, the controller was specifically saturated, i.e. the product flow rate was such that even when the null step duration was reduced to zero seconds the set point could not be met. In Figure 7.23 it can be seen that the response was quicker than for the ramped change, shown in Figure 7.19 and again there was a final offset.
Figure 7.21a - -25% Step Flow Disturbance With and Without Product Control (Ti=1000)

Figure 7.21b - -25% Step Flow Disturbance With and Without Product Control (Ti=5000)
7.5 Conclusions

The method of matching the predicted product concentration of the process by manipulating the oxygen adsorption parameter of the model worked well. The algorithm seems robust under all the conditions tried. With a well-tuned controller the model had adapted to match the process in about three hours from initial start-up of the process, or about two hours from the time of the first release of product gas from the process.

When the model was given incorrect information about the process, i.e. the wrong product receiver volume the method still coped. The introduction of extra time lag into the system, for example due to the wrong product pipe length, proved to be the most difficult for the controller to deal with, increasing the response time of the model. Although the increased response time was, in some cases, doubled over the base case, the model was still stable and adapted to match the product concentration within three hours.

When a control system, based on a predicted concentration within a bed, was applied to the process in order to maintain the product purity at a given set point, the control algorithm chosen was stable and controlled well over a range of disturbances. However, no attempt was made to optimise the position of the 'pseudo concentration measurement' within the beds and it is possible that by careful positioning of this point the response could be improved further.

The control response was much better for disturbances that increased the product flow rate, these required a reduction in the null step duration and this action was mainly handled by the 'cut-off' switch. The inclusion of this 'cut-off' switch into the controller gave very rapid responses, most notably under conditions where the set point for the product purity could not be achieved. Under these conditions, once the null step time had been reduced to zero, another control scheme is required to maintain the product purity. The most logical approach would be to include a hierarchical control system, where as each controller failed the next higher one took over. In this
way the null step could first be reduced and then the purge time decreased (with an appropriate increase in the purge flow rate to maintain the total purge amount per cycle. Although in an ILE model purge velocity has no effect on the ability of the purge gas to clean the beds, in reality there would be a limiting value of purge velocity due to mass transfer resistances within the bed. This type of control may not be commercially possible as very few PSA plants have the ability to control the purge gas rate, the rate being set at the design stage of the plant and becoming implicit once constructed.

Once this point has been reached more 'exotic' steps could be introduced into the cycle in an attempt to maintain the product purity, such as releasing product from a bed before it had finished fully re-pressurising on the pressurisation step, thus allowing the product flow from the other bed to cease earlier in the cycle. At some point the control system would have used all available possibilities to maintain the product purity and the final cycle would be considered optimum for the process. Normally the optimum cycle is determined when the plant was designed, however as performance of the adsorbent dropped over time the controller would always be capable of finding an optimum for the process. Additionally, as the controller would have to move further up the cascade to maintain the product purity as the performance of the adsorbent deteriorated, it could provide feedback as to the current internal state of the adsorbent, thus providing a valuable indicator of the expected life of the adsorbent.

Although the control algorithm works well on the model, there is a risk that it will not work on a real plant, so the next stage of this work would be to repeat the experiments on the pilot plant, i.e. sections 7.2, 7.3, and 7.4 should be repeated with the model connected to the pilot plant.

The process again showed non-linear behaviour in many of the experiments, for example, a change in the product flow rate has effects of different magnitudes depending on its direction. The effect of more than one possible output from the same set of cycle conditions is not surprising as the
PSA process has been shown to have non-linear dynamics in many of the experiments in this work. Most control theory is based on linear responses and uniqueness, in that for a given set of input conditions there is only one possible output. Given all this, it is remarkable that such a simple control scheme can provide good control of the product concentration, especially when even very large disturbances are introduced.
Chapter 8
Conclusions And Recommendations

8.1 Conclusions

This dissertation describes theoretical modelling and experimental work for the separation of oxygen from air using a zeolite adsorbent in a two bed PSA process.

The majority of the work was put into the development of a flexible process simulation program capable of solving the model in real-time. This was based on Kirkby's (1983) original ILE model, as modified by Espitalier-Noel (1988). It used the same assumptions, but included the ability to simplistically model the surrounding process equipment, rather than just the adsorbent beds. The new simulator was a general purpose PSA simulator, allowing the use of multiple beds and the modelling of any cycle consistent with an ILE assumption for the adsorption process. This simulator was not only capable of predicting the behaviour of the process at steady state but during transients as well.

This program was further developed to generate three-dimensional graphical output of the bed concentration profiles against time. The ability to sequence a two bed PSA pilot plant whilst running the model in real-time was also included.

The second largest effort was put into using the simulator to model the effects of changing operating parameters that would be difficult or time consuming to do experimentally. These included the effect of changing the dead space volume at either end of the bed, the adsorbent bed length, and the adsorption parameters, as well as the purge flow rate (which can easily be altered experimentally by suitable use of a control valve). These show that the design of the surrounding process equipment can have a major influence
on the performance of the PSA process and that the system performance does not depend on the adsorption beds alone.

In particular, it was shown that the performance of the PSA process is more sensitive to the 'dead' volume at the product end of the beds than the feed end. A critical volume for product end dead volume was identified, only above which is the product purity reduced. In some circumstances it was shown that a performance enhancement could be obtained by adjusting the feed end dead volume, however the conditions are such that they are probably not of commercial interest.

The importance of having identical lengths for the two adsorbent beds within the system was also demonstrated. With beds of different length it was shown that it was possible to enhance the performance of the plant under some conditions.

Work was done to see the effect of asymmetric purging on the process, by introducing an imbalance in the amount of purge gas used for each of the two beds. The model showed that a PSA process is more intolerant to non-symmetrical purging of the two beds than any other parameter tested in this project.

Experimental work was concerned with the practical difficulties involved in running and controlling a two bed PSA process whilst running a simulation of the process in real-time. Matching of the model parameters in real-time to allow the process performance to be predicted more accurately was demonstrated against previous runs of the model, including the effect of both parametric and structural errors. Finally, a simple model based system was demonstrated for product purity control.

8.1.1 Program Enhancement

Modelling of the PSA process over the last twenty years has led to more complex models being conceived and implemented. These include assumptions such as non-isothermal operation, non-linear isotherms, non-
equilibrium adsorption processes, and non-isobaric beds. Some models work with multiple adsorbents, multi-component gas mixtures, and many varied process steps.

The models have gradually improved the fit between theory and experiment, to where one of the latest published models, Kumar et al (1994), reports only a 0.1% error in prediction of product concentration from an O₂ VSA plant. However, these models require an enormous amount of computer time, Kumar et al (1994) report 20s per cycle for an O₂ VSA process, running on a DEC 5000 workstation, and this model assumed symmetrical beds and so effectively only had the work load of one bed to simulate.

With the latest models requiring such large amounts of computer power to solve, even with simplifying assumptions such as symmetrical beds, it is impractical to expect to be able to have a complex model running in real-time on a readily available personal computer. The simplifying assumptions normally made for a more simplistic model were looked at with the intention of seeing how their variation would affect the predicted process performance. The method of solution was based on Kirkby's (1983) ILE model, as modified by Espitalier-Noel (1988), which allowed easy comparison with previous work for checking purposes. As the solution is so fast compared to other models the simulator could run over fifteen times faster than real-time on the 75MHz Pentium PC used in this work.

Additionally, some of the assumptions were removed, e.g. that of symmetrical beds (to allow transient behaviour to be studied), the fact that only the adsorbent beds were modelled, and that process steps were always well behaved (i.e. product release being at a constant pressure throughout).

With the amount of time required to write a general purpose PSA simulator being enormous, and well beyond the scope of this project, a simple simulator capable of predicting transient behaviour was written, allowing the process layout and operation of a PSA process to be easily modified. This program was written in FORTRAN 77, for historical reasons, but with as
many object-oriented principles so that in future it could be modified or converted to another programming language more easily.

The model is capable of dealing with any normal process step due to the flexible nature of the solution, provided that the gas velocities and rates of change of concentration are not so high that the ILE assumption is not valid. It no longer depends on specific subroutines for specific step types, using the state of the valves in the system to calculate the flow of gas in or out of the beds. Due to the fact that it solves more than one bed simultaneously, the simulator can handle direct bed to bed gas transfers, something which has often been avoided in other models due to its perceived complexity.

One major enhancement that was added was the ability to generate three-dimensional plots when the program is running, whilst still maintaining the ability to run the model faster than real-time and to sequence the pilot plant. The 3-D plots could also be saved for later viewing. These 3-D plots gave great insight into how the concentration profiles develop within the cycles.

The numerical routines used in the simulator proved to be stable in operation and were easily able to cope with varying the process parameters, such as volumes and adsorption parameters, even dynamically whilst the program was running.

8.1.2 Experimental

The limited amount of experimental work concentrated on the use of the simulation program to sequence the plant in real-time and collect data. This data was used to initially tune the model to be a reasonable representation of the pilot plant. Work was also done to ensure that the program could solve the model in real-time, whilst sequencing the plant and recording data, for future work.
8.1.3 Comparison Between Theory and Experiment

Prediction of the process product concentration was only marginally improved over that of the model developed by Kirkby and extended by Espitalier-Noel for the same conditions, as compared to experimental data from Espitalier-Noel's work. However, operating the process with slight differences between the beds or flow rates, and hence removing the assumption of symmetry, created asymmetric steady states. This produced the remarkable effect of significantly altering the prediction of performance and improving the shape of the predicted performance curves to be a much better match with previous experimental data.

Although the ILE model predicts the breakthrough effects well for an equilibrium based adsorption process it has never been that successful in predicting the quantitative behaviour of an equilibrium PSA process. It is likely that the reason that the ILE model does not predict the behaviour of equilibrium PSA processes is due to the sensitivity of the product concentration to the location of key shock waves within the beds. Adding axial diffusion to the ILE model should significantly improve the predictive ability of this type of PSA model. The addition of intra-particle mass transfer would further advance the performance of the model, especially at high flow rates. It is estimated that the addition of either of these effects would increase the run-time of the model approximately ten-fold, although with the massive increase in affordable computers no available this should present no problem to the model still running in real-time.

8.1.4 Modelling Experiments

In Chapter 6 a large number of experiments on the model were presented, showing the effect on the product purity when various equipment parameters, e.g. product end dead volume, were adjusted. For most of the parameters studied it would be difficult or time consuming to adjust them in a practical experiment. The following points were concluded from this work:
1) Feed End Dead Volume

The process is remarkably tolerant to the size of the feed end dead volume. In many cases an effect on the product concentration was only seen after the volume was increased above that of the adsorbent beds, a situation unlikely to be found in any real plant.

Under some conditions it was shown to be possible to enhance the product purity from the plant by adjusting the dead volume, although it is felt that the nature of the conditions are such that they would have little or no commercial interest.

2) Product End Dead Volume

Generally the effect on the process was the same as for the feed end dead volume. However, the process was far less tolerant of changes to the dead volume at the product end of the bed than the feed. A dead volume equal to only around 10% of the adsorbent volume was shown to have a significant impact on product purity under some conditions.

The conditions that proved most tolerant withstood a product end dead volume of, at most, around two-thirds the volume of the adsorbent. In contrast, the feed end volume was raised to three times the adsorbent volume without any apparent effect for some systems.

3) Purge Symmetry

The process proved to be least tolerant of non-symmetrical purging than any other parameter manipulated. A deviation in the amount of purge between the beds of as little as 10% could cause a 15% drop in product concentration, regardless of whether the bed was now over or under purged. Although they were initially expected to be more tolerant, even systems which withdrew very small product amounts proved equally susceptible to differences in the purge amounts.
In some circumstances it was shown to be possible to enhance the performance of the plant by slightly non-symmetrical purge amounts, however these were for cycles that produced only low concentration product and again are felt to be of little commercial interest.

4) Bed Symmetry

A small variation in the length of a bed had little effect on the overall performance of the plant. An effect was noted for systems that had very large differences between the lengths of the two beds and for some systems that had either very low or very high purge amounts, neither of which are likely to be used commercially.

At very high product amounts it was found to be possible to enhance the performance of the plant by having beds of different lengths. The most likely explanation for this scenario is that it is better to have one bed of the right length than neither containing enough adsorbent.

8.1.5 Process-Model Matching

The first section of Chapter 7 showed that a model could be matched to a previous run of the model, which had been run with different parameters, in real-time. When matching the predicted product concentration against the previous runs of the model it proved possible to use a simple feedback PI controller to remove the error, as suggested by Gawthrop & Ponton (1996). It was shown to be possible to match the predicted product concentration to that of the process by adjusting the adsorption parameter for one component alone, in this case the combined parameter for oxygen + argon. It is important to remember that the adjustment of an adsorption parameter, whilst the model is running, causes a mass balance violation, see Section 7.2.1.

It was demonstrated that the model was robust enough to survive structural and parametrical mismatches between the model and the process.
It was also noted that care had to be taken in order to limit the range of the adjusted parameters to that allowed by the equations contained within the model. For use in a control scheme a model must be robust and failing to put any bounds on a critical parameter was initially a common cause of program failure when the experiments were first started.

8.1.6 Dynamics & Control Experiments

The second half of Chapter 7 dealt with the effect on the product purity of simple disturbances in the product flow rate, with and without purity control. The controller was based on a predicted concentration within the adsorbent beds, a measurement not normally made on a real PSA plant. It was shown that it is possible to effectively control a PSA process in this way, even allowing for the non-linear dynamics displayed during much of the previous work.

Additionally it was shown that multiple steady states could exist within the model, something that had previously suggested as not being possible. Reversible transition between steady states was demonstrated, along with an explanation for why they may form within this particular model.

8.2 Recommendations

Throughout this dissertation various problems have been highlighted. This section briefly collates the problems felt to be most important, then summarises the main recommendations for improvements. Finally, many areas that are felt to be of interest and worthy of further study are commented on.

8.2.1 Model and Simulation Design

A simple general purpose process simulator for PSA systems based on the ILE assumption has been programmed and it performed well. It has some
problems and limitations and these are discussed below (in no particular order of importance):

1) The purge cycle was used throughout this work. However, due to the flexible nature of the program it can, in principle, simulate any type of step, many of which are not often not included in a model due to their complexity. By careful use of the program it should be possible to model many other steps and cycles that are in common use in industry, such as bed pressure equalisation steps with beds connected top-to-top, bottom-to-bottom, or both simultaneously.

2) The program is written using an object-oriented methodology but in a procedural language (FORTRAN 77). By re-writing the program in a true object-oriented language, such as C++, it would be possible to significantly simplify the structure of the program and allow an entire layer of complexity to be removed. In addition, it would allow unlimited numbers of objects to be simulated, at present the number of each type of object is limited by the requirement to pre-define the size of the arrays used to store them.

3) Only ILE type beds are simulated. Other types of bed model could be added, allowing a choice to be made as to which is most appropriate for the process to be modelled. These could include multi-component adsorption, pressure drop, axial diffusion, intra-particle mass transfer, and non-isothermal operation. Removing the isothermal assumption would require all the objects in the system to be reprogrammed to include energy terms.

4) The plant layout description must be entered manually. A separate program with a graphical interface could be added, this would then be used to generate the text input file required by the simulation program and review the results.

5) The simulation could be modified to include more than two components. This change would be reasonably easy for most of the objects, however a new type of adsorbent bed would then definitely be required, see item 2 above.
6) The isobaric assumption within a group of objects could be removed. This would require a major re-write of the model solver routines as the solution method currently assumes that the pressure is equal for all objects within a group.

Due to the object-oriented way that the program has been intended to be structured, implementing item 2 first should be relatively easy. The other problems and limitations could then be removed without requiring the whole structure of the program to be re-written.

8.2.2 Plant Design

The pilot plant performed very well, being originally constructed around eight years before the project started and was around eleven years old at the end. It did have some important limitations which had to be worked around:

1) The use of mass flow meters on the pilot plant proved to be one of the biggest obstacles to overcome. None of the flow rates around the plant were capable of being controlled automatically, any flow control relied on the setting of manual valves which had to be positioned at the start of the experiment. This meant that the amount of gas being let in or out of the beds could not be kept symmetrical at all times.

2) The original thermocouple board in the pilot plant computer would not fit the new Pentium motherboard's expansion slots. The use of a new thermocouple board would allow temperatures to be measured within the bed, which could then be compared against the model's prediction of where the concentration fronts are. This could provide an indirect check that the predicted concentration fronts are in the correct position.

An alternative to this might be to take very small sample flows from the beds through the existing thermocouple fittings, thus allowing direct measurement of the concentration fronts. The main drawback with this
procedure would be that the concentration fronts move very rapidly and so very careful design of the sampling and analysis system would be needed.

3) The feed air dryer needs replacing as it failed during the experiments. This is especially important if prolonged high flows are required for calibration of the flow meters.

4) The problem of leaking pipe work was commented on by Espitalier-Noel (1988), who suggested that soldered fittings should be used wherever possible. In even the brief amount of experimental work done for this project a significant amount of time was wasted ensuring that the plant was essentially leak tight. Soldering many of the fittings in place removed most of the leaks, as did ensuring that the remaining fittings were tightened correctly and that the pipe work was properly aligned into the fittings and so the use of compression fittings in future experimental apparatus is to be discouraged.

5) Additionally, the actual experimental apparatus used for PSA research should be checked to ensure it is operating truly symmetrically at all times, as non-symmetrical purging was shown to have a large influence on the product concentration predicted by the model.

8.2.3 Future Model and Simulation Work

The future areas of theoretical work in PSA modelling lie far beyond the theory used in this project. One of the most important is the accurate description of heats of adsorption and multi-component equilibrium. This continues to be the area that restricts the accuracy of the latest models, causing them to predict the wrong temperature profiles within the adsorbent beds as even a relatively small change in temperature affects the adsorption equilibrium. This is especially noticeable on large industrial PSA units where much larger temperature changes occur than in the small laboratory units used for experimental work, due to the significant change in the ratio of bed volume to wall area.
In the area of simulator design, with the computing power of personal computers, on average, doubling every eighteen months, it should be possible to create a general purpose PSA process simulator, along the same lines as many commercial simulators for the process industries. It should model all the equipment and beds to remove the symmetry assumption, making it suitable for experiments on the dynamics of the process rather than just the steady state performance. This project alone would probably require a good computer programmer to spend their entire Ph.D. writing it, even if the adsorbent bed models had been written in advance.

It would be interesting to measure the effects of changing many other parameters on the performance of a PSA process. There are many parameters that could be adjusted that could significantly affect the performance, some of which are:

1) The effect of pipe lengths and hence delays in concentration profiles reaching their destinations, especially during steps where gas is transferred directly from one bed to another.

2) The effect of bed shape could be investigated in more detail. There is the possibility of changing the bed area along its length to improve the movement of concentration fronts within the process. Many industrial units are now of a cross flow or radial design, resulting in beds that do not have a constant cross sectional area with respect to the direction of gas flow.

8.2.4 Future Experimental Work

Experimental work provides the absolute test for all the theoretical modelling work that is done. The data provided from experiments must be exactly defined, including the actual profiles for flow rates, pressures etc., rather than just the total or average amount used. For example, given the pressure profile vs. time for the process it would be possible to match the exact shape of the bed pressurise profile during a cycle, important, for example, to a mass-transfer dominated model.
More experimental work could be done to provide the shape of the profiles for flows, temperatures, concentrations to allow models to work with more realistic data.

There appears to be no published literature on the effects of disturbances on the performance of PSA processes, yet it is almost impossible to operate a plant without a disturbance of one kind or another. For example, the effect of changing the product flow rate on the product concentration produced by the PSA plant. Experimental work in this area would prove invaluable as modelling the dynamic behaviour of a process often requires more accurate models than for steady state.

8.2 Concluding Comments

The applicability of the PSA process is vast, as witnessed by the current wide range of commercial systems installed on many different types of plants throughout the world. From air separation to hydrogen recovery the PSA process has been found to be both efficient and economically viable. Given this background, and the continuing efforts devoted to its study and improvement, the future of the PSA process seems assured.
References


Conclusions And Recommendations


Finlayson, D & Sharp, AJ, Improvements in or Relating to the Treatment of Gaseous Mixtures for the Purpose of Separating them into their Components or Enriching them with Respect to One or More of their Components, British Patent, 365,092, 1932


**Conclusions And Recommendations**


Conclusions And Recommendations


Shirley, Al & LaCava, Al. Novel Pressurization Methods in Pressure Swing Adsorption Systems for the Generation of High-Purity Gases,


Conclusions And Recommendations 222


Appendix 1
Object-Oriented Programming Review

A1.1 Introduction

One of the more recent programming paradigms to become widely available has been that of Object-Oriented Programming (OOP). Object-Orientation (OO) is a method of program design coupled with specific programming language facilities to create more robust and maintainable programs.

This project took many techniques and ideas from object-oriented programming languages and applied them, as far as practical, to a program written in FORTRAN 77. This appendix gives a basic explanation of these concepts and how they were used to create a flexible, robust object type program using only FORTRAN 77, a language which was not designed for OOP. In fact, FORTRAN 77 has no inherent features that could be truly described as object-oriented and no assistance for an object-oriented programming style.

However, FORTRAN 77 was used for this project because previous models were written in this language and it was a familiar language to those involved in the project.

A1.1.1 Terminology

The terminology of computer languages is often considered both terse and complex. Some of this terminology is given below to explain terms used subsequently in this both this appendix and Appendix 2.

Class - A definition for an object. This is the preferred term for use with OOP development. A class does not exist but can be considered as the plan for an object.
Data - Any information contained, used, or modified within the program.

Method - An alternative word for a procedure and is the preferred term when dealing with OOP development.

Module - A collection of data and procedures that are packaged together, a term usually used in non-OOP development.

Object - A class that has been created ready for use, as opposed to the class which is simply the definition of it.

Private Variable - A private variable is a piece of data that can only be used within the class where it was defined, i.e. bed length in a bed module.

Procedure - A specific subroutine or function that performs some operation on or with the data.

Properties - An alternative word for data. This term is normally used when dealing with OOP languages for data that is visible external to the object, i.e. public variables.

Public Variable - A public variable is data that is directly accessible outside of a class (alternatively a known as a property), or has been defined explicitly for general use, e.g. elapsed program time.

A1.2 Brief History of Computer Languages

It is interesting to look briefly at the history of FORTRAN because it has changed rapidly in the 40 years since the first version was invented and because it can be considered as the 'grandparent' of many languages developed since.

The development of real computer languages, as recognised today, began in the 1950's with the invention of FORTRAN (FORmula TRANslator), by IBM, which was intended to allow the easy conversion of scientific or algebraic formulas into the machine code instructions required by a computer.
<table>
<thead>
<tr>
<th>Year</th>
<th>Name</th>
<th>Usage</th>
<th>Original Platform</th>
<th>Current Platform</th>
<th>Original Company</th>
<th>Original Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1954-1957</td>
<td>FORTRAN</td>
<td>Scientific &amp; engineering</td>
<td>IBM 704</td>
<td>IBM</td>
<td>IBM</td>
<td>John Backus</td>
</tr>
<tr>
<td>1958</td>
<td>FORTRAN II</td>
<td>Scientific &amp; engineering</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>1962</td>
<td>FORTRAN IV</td>
<td>Scientific &amp; engineering</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>1960-1962</td>
<td>COBOL</td>
<td>Database</td>
<td>any</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1965</td>
<td>BASIC</td>
<td>Interactive</td>
<td>any</td>
<td></td>
<td>Dartmouth College, USA</td>
<td>J. Kemeny &amp; T. Kurtz</td>
</tr>
<tr>
<td>1966</td>
<td>FORTRAN 66</td>
<td>Scientific &amp; engineering</td>
<td>any</td>
<td></td>
<td>ANSI (X3J3 committee)</td>
<td></td>
</tr>
<tr>
<td>1968</td>
<td>Simula67</td>
<td>Simulation</td>
<td>any</td>
<td></td>
<td>K. Nygaard &amp; O. Dahl</td>
<td></td>
</tr>
<tr>
<td>1970's</td>
<td>C</td>
<td>General</td>
<td>DEC PDP-11</td>
<td>any</td>
<td></td>
<td>Dennis. Ritchie</td>
</tr>
<tr>
<td>1972</td>
<td>Smalltalk</td>
<td>General</td>
<td>any</td>
<td></td>
<td>Xerox</td>
<td>Dan Ingalls</td>
</tr>
<tr>
<td>1978</td>
<td>FORTRAN 77</td>
<td>Scientific &amp; engineering</td>
<td>any</td>
<td></td>
<td>ANSI (X3J3 committee)</td>
<td></td>
</tr>
<tr>
<td>1974-1978</td>
<td>ADA</td>
<td>Large scale, real-time</td>
<td>any</td>
<td></td>
<td>US Defence Department</td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td>C++</td>
<td>General</td>
<td>any</td>
<td></td>
<td>AT&amp;T Bell Labs</td>
<td>Bjarne Stroustrup</td>
</tr>
<tr>
<td>1985</td>
<td>QuickBasic</td>
<td>General, compiled or</td>
<td>IBM PC compatible</td>
<td></td>
<td>Microsoft</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>interactive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>Fortran 90</td>
<td>Scientific &amp; engineering</td>
<td>any</td>
<td></td>
<td>ANSI (X3J3 committee)</td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>Fortran 95</td>
<td>Scientific &amp; engineering</td>
<td>any</td>
<td></td>
<td>ANSI (X3J3 committee)</td>
<td></td>
</tr>
</tbody>
</table>

"**" indicates that it is considered an object-oriented language

Table A1.1 - Brief Development History of Programming Languages

Appendix 1 - Object-Oriented Programming Review
The primary aim of FORTRAN was to allow non-programmers to use the computer at a more conceptual level and therefore increase the demand for IBM computers. Previously all computers had been programmed directly with machine code language, or at a marginally higher level. With FORTRAN, scientists could concentrate on the algorithms of solution, not the conversion of algorithms into machine code.

Table A1.1 shows a brief summary of many of the main programming languages developed since the 1950's, although there are many others that are not included. It is interesting that many programming languages are descendants of earlier ones, but with additional features to solve problems that were associated with the original language. Currently there are estimated to be at least 80 languages with object-oriented features (Booch 1994) and countless numbers of procedural languages for various purposes.

FORTRAN is a language designed for science and engineering. It has evolved steadily over the years, until it reached the latest version, Fortran 95, and is still in wide spread use as the language of choice for these types of applications. A more detailed review of the history of the Fortran language can be found in Edgar (1992).

A1.3 Programming Techniques

Many of the basic techniques used today when writing a program have been used since the earliest versions of FORTRAN were developed. One example is the use of procedures to implement operations that are repeatedly performed by different parts of a program, thus avoiding mistakes by having multiple copies of the same calculation.

An example of a later development is the 'user defined data type', which was added to many languages, allowing the creation of new types of data from the basic ones built into the language, i.e. a custom data type can be made from a collection of the default data types provided. This feature was not added to the Fortran language until Fortran 90.
As the power of computers grew it became obvious that the simple techniques used to maintain the quality of the software had become insufficient. Faster computers meant more functionality could be implemented and programs rapidly grew in size and complexity. It soon became almost impossible to prevent software faults, known as “bugs”, appearing in any large program, commercial or otherwise.

To aid in the prevention of software faults, language developers created new concepts which were specifically designed to make life easier for programmers dealing with large programs. These were languages such as Pascal, Ada, Smalltalk, and C++, each following a separate route in the search for a solution. For example, Ada focused on having many specific commands in the language to allow more data manipulation to be implemented by the compiler rather than the programmer. C++ concentrated on improving the management of the programs and the interfaces between modules within the program.

Overall, the general trend over the last thirty years has been to compartmentalise methods with their data into self-contained modules, thus preventing errors being introduced by procedures which manipulated the data in some unexpected way.

**A1.3.1 Procedural Programming**

Early languages e.g. FORTRAN concentrated on procedural methods of programming. This can be explained best by the example:

1) Create a variable.
2) The data are passed to a procedure.
3) The procedure performs some set of operations on the variable.
4) The data are passed back.
5) Repeat from step 2.

In this type of program the algorithm is the most important part, i.e. the instructions that operate on the data. The programmer has to decide
exactly what will happen to the data during the entire operation of the program (unless using an external pre-written subroutine). If the data format is changed then all parts of the program that deal with it must be updated to work correctly.

Additionally, the data are passed from one procedure to another and each acts on it as it sees fit. If another programmer modifies a procedure then it can introduce errors in other places within the program if the operation they implemented is not correct.

A1.3.2 Object-Oriented Programming

Object-oriented programming looks at the program design from the opposite point of view to procedural programming. In OOP the data are the centre of concern and the actions that happen can be considered to be the point of the data themselves.

In an object-oriented program the programmer creates an object and the object implicitly "knows" how to manipulate itself. The programmer does not need to know what manipulation was actually used, only that it produced the desired transformation of the data within the object. This allows the manipulation procedure or the specifics of the data to be altered within the object without affecting any other part of the program.

In this style of programming, the data and the procedure have been linked together as a single object. Normally, the user of an object has no direct access to the data within and so it becomes more difficult to introduce errors into the data.

The term 'procedure' is rarely used in OOP development, it is replaced by 'method'. The term 'data' has also been dropped for externally visible data, being replaced by 'properties'. This is the logical way of looking at the objects within an OO program and is best demonstrated by an example: a person has certain external properties that make them unique, i.e. red hair, and green eyes, and certain methods, e.g. a method of singing a song. You would not
normally say someone has the data of red hair and green eyes and the procedure for singing a song. Internal data are still commonly referred to as simply data.

There are several distinctive aspects that are considered to define an OO program, a brief explanation of each follows.

**A1.3.2.1 Encapsulation**

Encapsulation is the concept that the data and the methods are combined together to become an object, as described in Section 1.3.2. A fundamental principle of this is that the inner workings of the object and data are hidden from public view and only the public properties and methods can be accessed or used.

**Benefits**

The programmer creates an object and can then use only the external methods to manipulate the data. If the storage of data or method of manipulation is changed within the object then the external effect is unseen. This allows programmers to prevent data from being corrupted by parts of the program that are badly written or have not been changed when the data storage format was altered.

Data can normally be declared “public” or “private”, private data cannot be changed from outside of the object, public data are often referred to as the properties of the object. It is often possible to define public properties as being read only to prevent them being changed by code outside of the object.

**Drawbacks**

Instead of having direct access to the data they remain private to the object. However, if a suitable manipulation has not been supplied with a pre-written object then there is no way to perform that operation on the data. In
addition, the program can be slower to execute because all changes to the data must be made through function calls. If some or all of the data are made public as properties, but not made read only, then it can be manipulated freely, however all the benefits of control of the data are then lost.

A1.3.2.2 Inheritance

A class is the template for an object. It contains all the definitions for the methods, properties, and data required by the object. Inheritance is the ability of a class to be used as a building block for other classes, the original class is then a 'base' class and the new class is termed a 'derived' class. Derived classes retain all the features of the original class but usually have additional methods, data, and properties.

More importantly, a class derived from another class can still be referred to as if it were still of the original class type. If an array of base objects is defined then it can contain a mixture of objects from both the base class and any derived classes. The appropriate methods are used for each type of object as required.

Example

An adsorbent bed is a base class defined with the model. It has certain assumptions built in, e.g. no axial dispersion. If a bed is required where dispersion is significant then this could be created from scratch or by being based on the existing type of bed. Extra properties and data would be added for the dispersion coefficients and a new calculation method written to replace the original version. The other methods would remain the same as the original, thus minimising the work required and the potential for mistakes. The new bed object can still be stored in the same array as the old types.
**Benefits**

Solid, tested code can be reused by many different objects and if this code is changed then all the objects dependent on it will be updated automatically.

**Drawbacks**

A mistake in one basic piece of code is replicated throughout the program, however it is also easy to correct. It can also be confusing as to exactly which version of a method is being used by the object at any particular time.

**A1.3.2.3 Polymorphism**

Polymorphism is the use of a generic name for multiple methods on multiple objects, distinguished only by the object and possibly the parameters passed to the method. There are many examples of this in the real world, where a verb can be used with many different nouns, i.e. 'turn on the light', 'turn on the computer' and the operation that is performed depends upon the type of object being manipulated.

In the same way within the program, calculating a 'tee' object results in different code being used to that when an 'adsorbent bed' object is calculated.

This is one of the most useful parts of OO techniques as it eliminates many of the 'IF' statements traditionally required when dealing with many different types of object within a program. The usefulness of this is enhanced by inheritance, see Section A1.3.2.2, as multiple types can be stored within an array of the base class and the appropriate method for each is called.
Example

In mathematics the use of the '+' sign implies that the items to the left and right should be added together. It is used to add integers, real, and complex numbers, as well as other items such as matrices. All types of data can be combined using the same symbol, although how the combination occurs depends on the objects involved. For ease of use, the symbols used in everyday life are commonly found to mean the same in most programming languages.

Benefits

The programmer does not need to uniquely name the procedures within the program. The same name can be used for more than one procedure and an array of objects can be manipulated with a single named command.

Drawbacks

The same name can have a different effect when used on a different object. It can be confusing as to exactly which version of a method is being used by the object at any particular time.

A1.4 Use of OO Techniques With FORTRAN 77

A1.4.1 Procedural Basis

FORTRAN 77 was never intended to be an OO programming language, having its basis before OO techniques became common, and was intended to be a purely procedural way of solving mathematical problems.

This project required a more flexible model that was required to cope with multiple objects (i.e. the pieces of the PSA plant) that could be created or removed from the system by the user at run-time, rather than a fixed layout being written into the model. The need to deal with objects that are determined only after the program has started, and objects that need to be
dynamically connected to other objects, causes difficulties with the interface between the objects unless it is clearly defined. There is also the requirement to use the correct procedure for each object.

The usual solution results in a large number of IF statements within the program to deal with each case of object being used. If a new object is added then all the IF statements must be updated.

A1.4.2 Object-Oriented Interface

The OO approach allows a consistent, predefined interface to be maintained for all the objects within the code. The only way to make use of a particular object is to use the objects’ methods, it is impossible to manipulate its data directly, except for its public properties.

The object independent layer is an extra section of the program added between the higher routines (that decide calculation order etc) and the lower routines that implement the desired action (i.e. calculate the values for the current time interval). The higher routines deal only in terms of objects that require manipulation and they have no need to know the actual type of an object in order to manipulate it. The object independent layer uses an object number (objUID) as a pointer into a table of objects that identifies the true object type and which particular number, known as the (typUID), within the collection for that object type.

The typUID is passed to the specific code for that object, as identified by the object type, and is then used as a pointer within the code to identify which part of the data arrays to use. This is best illustrated by example:

the main calculation routine calls for an object to be calculated:

\[
\text{ObjCalculate(objUID)}
\]

the object layer identifies the true object type:

\[
\text{Type = object\_type(objUID)}
\]

and the sub-object number:
typUID = object_sub-number(objUID)

and then calls the correct object type:

Type_Calculate(typUID)

doing the correct object type:

Type_Calculate(typUID)

the object then calculates itself, using the typUID as an index to its
data arrays (as there is normally more than one object for each piece
of code):

some_data = some_data_array(typUID)

The main piece of code in this system is the object independent layer,
which must keep track of all the objects from the time they are created from
the information in the data input file to the end of program execution. It can
then translates the function calls to the correct object-type. If the object does
not exist then it returns an error code so the program does not fail
immediately and for no apparent reason. This makes debugging the program
and the input file much easier. In addition, if a new object type is added then
the object layer can be updated and the rest of the program will remain
unchanged.

As a contrast to this, in an OO language, such as C++, this object
independent layer is unnecessary as the compiler builds the program in such
a way that this layer becomes implicit within the program itself.

This object independent layer is more fully described in Appendix 2.

A1.5 Conceptual Issues

When making the change from a procedural language to an object-
oriented language the question is normally asked is ‘Where is the program
logic?’. In an object-oriented language the logic is still there, however the
program can now be looked at as two distinct parts: what is to be done and
how it is actually done.

A good object-oriented program can be viewed as a series of higher
level instructions or messages that are ‘sent’ to each object in turn (by calling
the methods) to perform some action. Neither the object type nor the specifics of how the action is carried out are relevant. All the object type checking that a procedural language must go through to determine which particular procedure to call is therefore eliminated.

Details of the structure of the object-oriented ILE program are more fully described in Appendix 2.

A1.6 Bibliography

This short list details books that have proved useful during this project:

Appendix 2

The Structure of the ILE Program

A2 Introduction

This appendix summarises the structure and design of the new ILE program used in this work. This new model marks a new direction for the ILE model which was originally developed from work by Flores-Fernandez (1978), added to by Kirkby (1984), and finally Espitalier-Noel (1988).

Although each of the previous versions added more features, including the ability to model more complex steps they were all written in a procedural way, i.e. all the possible routes of program execution were determined when the program was first written. In order to add features such as simulating the surrounding equipment and allowing the model of the plant to be modified without requiring reprogramming, the new program used 'object-oriented' techniques, albeit using a language not designed with those in mind, FORTRAN 77. See Appendix 1 for a review of object-oriented programming.

The traditional way of writing a simulation program is to write it for a specific plant layout and to have the solution method built into its logic. Therefore it is normally straightforward to show block diagrams of the order the calculations are performed etc. However, in an object-oriented model, where the calculation is not predetermined, these kind of diagrams do not usually prove so useful. As the model layout is determined at run-time it must be possible for the solution route to adapt. Every model therefore will have its own solution route and so a diagram for the program would only truly refer to an specific model. Rather than detailing the execution path for an example model, this appendix details the object model of the program and how the objects appear identical from the model solving section of the program.
This appendix will detail the object independent layer from the top down to a description of how the individual objects work, then look at how the model solver uses the objects to perform the simulation. As FORTRAN 77 provides no facilities for object-oriented programs, all the code and facilities detailed in this appendix were written for this project.

Finally, this appendix will not describe in any way the chemical engineering aspects of the computations, which are all to be found in Chapter 4.

A2.1 Program Overview

The ILE program can be divided into several main areas:

1) preprocessor for input file analysis,
2) convergence routines,
3) equipment to be calculated,
4) PSA pilot plant controller (sequencing & data collection),
5) object independent layer, and
6) helper routines.

Each area is summarised in the following sections. To get an overview of the program, Figure A2.1 shows the general program structure in block diagram form. It is relatively simple, consisting of four main areas:

i) An input file processor, which reads the input file containing the description of the model to be simulated, and sets up the required objects and connections.

ii) A model solver object that converges the model.

iii) A time keeping object that keeps track of all time related information for the system, i.e. current time, current time interval size etc.

iv) An object independent layer that allows objects to be freely used, and without requiring any significant re-programming to add new types.
iv) The objects for each different piece of equipment, i.e. pipes, beds, valves etc, are accessed through a common interface.

The fact that no objects can access the actual type or call object functions directly means that additional object types can be added simply by updating this layer.

**A2.2 Object Independent Layer**

Because all access to the objects must go through the object independent layer, every object appears identical when viewed from any other object. Access from an object to another object below this layer is also not permitted, each object must go back up into the object independent layer and then back down. The appearance of the object independent layer is shown in Figure A2.2.
The object independent layer contains approximately 500 lines of code, plus comments. This is mainly functions containing IF statements to divert object independent calls to the right subroutine for the right object.

The convenience and flexibility provided by the inclusion of this layer is such that new plant layouts can be specified in a matter of minutes by changing the contents of the program input file.

A2.2.1 Object Type Database

To keep track of the type of each piece of equipment contained within the model an object type database is required. Each object is allocated a unique object identifier (objUID) and a unique type identifier (typUID) when first read from the input file. This objUID must then be used by any other object to refer to that object, the object independent layer passes the typUID to the appropriate subroutine.

An example of a section from an input file is given below:

```
DECLARE FEED_LINE AS TYPE FEED
DECLARE PRODUCT_LINE AS TYPE PRODUCT
DECLARE BED_TWO AS TYPE BED
DECLARE BED_ONE AS TYPE BED
```
Here four objects have been defined, so at the end of loading the input file the object type database would be filled as in Table A2.1.

<table>
<thead>
<tr>
<th>objUID</th>
<th>Type</th>
<th>typUID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feed</td>
<td>1</td>
<td>FEED_LINE</td>
</tr>
<tr>
<td>2</td>
<td>Product</td>
<td>1</td>
<td>PRODUCT_LINE</td>
</tr>
<tr>
<td>3</td>
<td>Bed</td>
<td>1</td>
<td>BED_TWO</td>
</tr>
<tr>
<td>4</td>
<td>Bed</td>
<td>2</td>
<td>BED_ONE</td>
</tr>
</tbody>
</table>

Table A2.1 - Object Type Database

Now that this table has been completed the object independent layer can translate calls to a particular objUID into the appropriate subroutine for the type of object being called, together with the typUID to identify the specific object of that type to be used.

The object database contains approximately 400 lines of code, plus comments, the majority of which is dealing with loading and initialising the database.

A2.2.2 Object Connection Database

For ease of setting up the plant layout file for a model, it was decided to allow the user to refer to each piece of equipment and each connection by name rather than numbers. This requires that as each object is loaded its connection point names are stored within the database as well. When all objects have been loaded the equipment and connection point names can be replaced by objects UID’s and connection numbers.

Although not technically required to be part of the object independent layer, the fact that the layer must store all the object numbers and types made this the ideal place to put this database.

This is illustrated simplistically in the following example.
Four objects are defined:

DECLARE FEED_LINE AS TYPE FEED
  CONNECTION TO BED_ONE
END DECLARE

DECLARE PRODUCT_LINE AS TYPE PRODUCT
  CONNECTION TO BED_TWO
END DECLARE

DECLARE BED_TWO AS TYPE BED
  CONNECTION TO BED_ONE
  CONNECTION TO PRODUCT_LINE
END DECLARE

DECLARE BED_ONE AS TYPE BED
  CONNECTION TO FEED_LINE
  CONNECTION TO BED_TWO
END DECLARE

In this case the object that is used in the connections from FEED_LINE and PRODUCT_LINE is not defined until the BED_ONE is declared at the end of the file. However, once all the objects have been loaded from file it is a simple matter to match the connections up, as in Table A2.2.

<table>
<thead>
<tr>
<th>object</th>
<th>1st connection</th>
<th>2nd connection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>object</td>
<td>end</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A2.2 - Object Connection Database

Once this table has been completed the object independent layer can connect the objects together and check that all connections are consistent, i.e. that a connection is only cross-referenced by one other connection. The
exception to this is that instruments and controllers are exempt from this restriction, as it is possible to have more than one instrument taking data from an object, e.g. both a composition and a flow recorder can be on the product line.

The object connection database contains approximately 400 lines of code, plus comments, the majority of which is dealing with loading, initialising and searching the database. About 30% deals with checking that the database is consistent, i.e. only one other object is connected to a single outlet of an object.

A2.3 Object Hierarchy

The object hierarchy determines how the objects are related to one another, whether they are based upon other objects, and whether they have a common interface.

---

**Figure A2.3 - Object Hierarchy**
The object hierarchy for the model is shown in Figure A1.3. As the program was written in FORTRAN 77, and not an object-oriented language, the object hierarchy is more for showing how the objects relate to each other and their common interfaces.

A2.3.1 Object Methods

Only the methods available through the object independent layer are described here. There are often many other helper functions used internally within a module that makes up an object. If a method for a particular object make no sense, e.g. changing the set point of a pipe, then there are two choices available to the programmer:

![Diagram of Object Methods](image)

**Figure A2.4 - Object Methods**
i) implement the function, but just put a message in the output file that this function was called but it does not make sense to do so, and then possibly stop the program execution.

ii) ignore the function completely, and rely on the error trapping of the object independent layer to put a message in the output file that this function was called but it does not make sense to do so, and the program will continue to execute normally.

By using option ii) in this way, the programmer can save time by not needing to write many functions that should never actually be called.

If a function does not exist then the programmer can add the function to the object independent layer relatively easily. For pre-existing objects a call to the new method will simply give the default response from ii) above. In Figure A2.4, there are extra functions added specifically for valves, adsorbent beds, and sequencers.

All the methods return a code to indicate their status, although often this is simply the default value of zero (indicating success). The only information currently returned via a method is the set point of the object, see Section A2.4.10, and the concentration at a certain position within an adsorbent bed, see Section A2.4.9. All other information is obtained through the properties of the object.

**A2.3.2 Object Properties**

As well as providing a way of making all objects appear identical, the object independent layer provides a way of making most of the data appear identical and publicly available to all routines and objects. The most commonly passed data between objects are flows, pressures, and compositions. These are held in a common storage area, indexed by the objUID plus the desired connection number. This significantly violates the rule on encapsulation; normally objects have a limited number of public
variables. Every object potentially has access to the data of every other object, rather than only the objects that it connects to, because of the use of the common storage area for all the flow data, as shown in Figure A2.5.

This breaking of the rules was required for two main reasons:

i) the use of methods to transfer return the data is time-consuming due to the need to go through the object independent layer each time.

ii) providing the object requiring the data a specific pointer limited to the public data for the other object is not possible in FORTRAN 77.

### A2.4 Object Descriptions

The actual objects, such as beds, receivers, valves, pipes etc, all implement the same interface through the object independent layer and so a brief description only will be given for the functionality that these general methods must provide. Details on extra methods provided by the adsorbent beds, valves, and sequencer will also be given.

#### A2.4.1 ObjLoadData

The object must load all its data, starting at the current point in the input file. It must finish processing and return at the end of its own section. At the end of the data loading routine, the object must have performed the following actions:

i) Checked that there is space to hold the new object.
ii) Loaded all connection details into the connection database.

iii) Loaded all initial conditions for the connection points.

iv) Loaded all initial conditions for any internal values required.

A2.4.2 ObjInitialise

Loading of the data and initialisation of the objects are separate methods. The reason for these two stages being separate methods was detailed in Section A2.2.2.

After all the object definitions, and hence the model description, have been loaded from file the object must be initialised. The first stage of this is to identify the matching object and connection points for each of its connections. Once this has been done the object must use external and internal conditions to calculate its initial state. Finally, the object must fill in any public properties, such as flows etc. at each of its connection points.

A2.4.3 ObjCalculate

This is the main method called once the model is running. It gives the object the opportunity to update its internal and external properties for the current time interval. The way that the object stores and calculates its internal properties is entirely hidden from the rest of the objects, however the external properties must be of the correct format.

Even if there is no change in external conditions around some objects in the current time interval every object is always given the opportunity to calculate. This allows objects that have an internal state that depends upon time as well as changing external conditions to update themselves.

A2.4.4 ObjCalcAboutEnd

To improve the performance of the program certain optimisations have been provided for. When the model solver decides that the current time
interval has a converged solution, each object has its ObjCalcAboutEnd method called, in the normal calculation order for the objects.

When an object has this method called it has the opportunity to do a rigorous final calculation of its internal and external state. This allows it to use a faster, but less accurate, calculation routine for the main body of the calculations and only perform the slowest, most accurate, calculations at the very end when the solution is almost converged. Additionally, if having done the rigorous calculations the object calculates new external conditions for the time interval then the object should use the more rigorous method until the model has truly converged for that time interval.

Although for many objects, e.g. a well-mixed receiver, there is no performance gain using this technique, for the objects such as adsorbent beds that require a large amounts of calculations, the overall performance of the object, and hence the program, can be increased significantly.

A2.4.5 ObjFlowError

This method is called every iteration of the model, within the time interval, to see if the error in flows between the last object in a group and the adjacent object are converged and returns the flow error between the two objects.

A2.4.6 ObjCompError

This method is called immediately after the ObjFlowError method to see if the error in composition between the last object in a group and the adjacent object are converged and returns the composition error between the two objects.
A2.4.7 ObjCalcEnds

Once the model has truly converged for the current time interval, every object has its ObjCalcEnds method called. This allows each object to save or update any values it will need to be ready for the start of the next time interval.

Objects such as instruments, controller etc. usually use this method to save their data to file, or update control parameters ready for the next time interval, knowing that the model has finished calculating the current time interval.

A2.4.8 ObjCycleEnds

This is similar in nature to the ObjCalcEnds method. It is called when a sequencer object decides that an entire cycle of an adsorbent bed has been completed. A composition recorder that is calculating the average product composition per cycle of an adsorbent bed can use this method to perform its final calculations and save the data to file.

By calling this method for every object in the model, objects do not have to keep track of the current position within the PSA cycle individually, thus allowing the PSA cycle times to be varied during the simulation, yet events that are required only at the end of a cycle can still be performed at the correct time.

A2.4.9 Extra Methods for the Adsorbent Bed

The adsorbent bed provides all of the default methods required by the object independent layer. In addition it provides three extra methods specific for its extra requirements. These extra methods can be considered as functions for changing or inspecting private data in the bed object. It would be possible to simply make these data public properties and these functions would not be required. However, this would violate the principle of data encapsulation, to find the concentration at any position within the bed
requires knowing how the data is held within the bed module. Changing the internal structure of the bed module would thus require updating of every piece of code that required this kind of information.

Therefore the three extra functions provide a standard interface for inquiring about or manipulation of data within the adsorbent bed module, without the calling code requiring any need to understand the inner workings of the adsorbent bed module.

A2.4.9.1 ObjChangeKa and ObjChangeKb

In order to adapt the output of the model to match that of the plant it was necessary to be able to change the adsorption parameters of the bed itself. This must be done whilst the simulation is running and without affecting the compositions at that point in time.

The adsorbent bed module supports two methods for changing the adsorption parameters of the bed. They must only be called between time intervals, i.e. after all objects have received a call to the ObjCalcEnds method. The program was not specifically written for an oxygen - nitrogen system, but in terms of two components, 'a' and 'b', thus allowing the possibility of using the program for other two component mixtures. The adsorption parameter for oxygen was specified as Ka and the adsorption parameter for nitrogen as Kb throughout this work.

Adjusting an adsorption parameter of a bed whilst the model is running is performed by taking the bed concentration profile in terms of 'Y' (the augmented mole fraction, see Section 3.3.2), converting it to true mole fractions 'y' using the current adsorption parameters, and then converting it back into terms of 'Y' using the new adsorption parameters, as illustrated in Figure A2.6.
Note: this routine worked well and the model proved robust to having the adsorption parameters changed dynamically, see Chapter 7. However, the use of this facility has a significant side effect in that the mass balance for the system is violated when the value of the adsorption parameter is adjusted. The amount adsorbed in the bed is spontaneously changed when this method is used, yet no material appears to have entered or exited the bed. This creates a mass balance error for the cycle in which the method was used, but if the adsorption parameter is not subsequently changed then the cycle mass balances will be consistent from the start of the next cycle.

A2.4.9.2 ObjConcAtPos

To control the product concentration using the internal position of a concentration front, it is necessary to know the location of the front at the end of a time interval. This method is provided to allow the internal concentration of an adsorbent bed to be measured. The method returns the current composition at the location specified in the call. The location is specified as a dimensionless number, range 0.0 to 1.0, relative to the connection point of the bed specified.

Any part of the program calling this method does not need to understand the internal structure of the adsorbent bed, so different calculation methods could be used for the beds without affecting the rest of the program.
A2.4.10 Extra Methods for Valves and Sequencers

Extra methods are provided by the valve and the sequencer objects. These special methods are used for control of the process, for example, if the sequencer determines that a valve should be shut it will call the ObjAdjustPosition method for that particular valve with a value of zero, thus closing the valve. The use of a method to change a setting for an object allows the object to check the validity of a setting before applying it, for example if the position of a valve was a public property it could be set to a negative value, but if it can only be accessed through a method the sign of the setting can be checked, and possibly corrected, before it is applied.

For example, for a valve object, the valve may be enhanced in that a limited speed of movement may be added. The controller does not need to know about the addition of this rate limit or how fast this rate is in order to still be able to position the valve. For a sequencer, it may be that the sequence can only be safely changed at certain times, by changing the sequence through a method the request can be held until the most appropriate time is available, for example, when it is appropriate for both beds.

A short description of each of these methods is given below:

**ObjAdjustPosition** - change the current set point or position of the object to the new one given. Returns the value of the old position. Used mainly for valve and controller objects.

**ObjAdjustFlowrate** - change the current flow rate of the object to the new one given. Returns the value of the old flow rate setting. Used mainly for “perfect” mass flow controllers.

**ObjChangeSubType** - change the current sub-type of the object to another. Normally used by valves, allowing them to be changed from “perfect” mass flow controllers to ordinary valves controlled by a pressure controller.
**ObjSetPoint** - returns the value of the current set point of the object. This was added to find out information about the current states of controllers, valves etc, in order to allow for the possibility of an advanced control object. **ObjAlterSequence** - modifies the current sequence being used by a sequencer object. Returns the previous value of the altered value of the sequence. **ObjChangeSequence** - changes the current sequence to another previously stored sequence. Returns the number of the previous sequence.

### A2.5 Time Manager

An important part of the program is the time manager object. This keeps track of the current time of the model and the size of the time interval.

The time manager object is an example of how data encapsulation allows the internal properties of an object to be different from the external properties. Internally the time manager uses integers to keep track of the current time, thus achieving much higher accuracy and speed than is possible with real numbers. The internal integer value of time is then converted to a real number for the rest of the objects in the model. This allows objects such as the eventer to be notified at exactly the right time, without rounding errors accumulating in the real numbers during the execution of the program.

### A2.6 Model Solver

The most important part of the model is the model solver object as this converges the solution of all the objects for each time interval. The equipment objects are divided into groups, separated by the valves (see Chapter 4). The instrument and controller objects are kept in the “instrument” group, the sequencer and eventer objects are kept in the “logic” group. For convenience, each equipment group is individually converged, then the whole model is
checked for convergence, and finally the instrument and logic group objects are called.

The model solver calls the ObjCalculate methods for all the objects within a group in the correct order. It then uses the ObjFlowError and ObjCompError methods to determine the errors for each iteration of the group within the time interval. Once the errors for all groups fall below a certain tolerance the model solver calls the ObjCalcAboutEnd method for every object. The ObjFlowError and ObjCompError methods are called and if the model is converged then the ObjCalcEnds methods are called, if not the calculate process is repeated until convergence is obtained.

The absolute tolerances in the program were set as:

Compositions: $1.0 \times 10^{-10}$ (mole fraction)

Flow Rates: $1.0 \times 10^{-10}$ (mol.s$^{-1}$)

and convergence of each time interval would require between 10 and 20 iterations, the determining factor in each case was the highest rate of pressure change within any of the groups.

When this process has been completed the ObjCalculate methods are called for all the instruments and controllers, to allow them to save data and make adjustments ready for the next time interval. Finally the sequencer and timer ObjCalculate methods are called to allow any time based events to be made.

Since the model solver is the centre of the model solution, it would be expected to be a large part of the program, however it consists of only around 600 lines of FORTRAN 77. Most of this is involved in looping through the object groups, calling ObjCalculate and other methods in turn. There is little more to do other than to keep track of the errors between objects and call the appropriate methods when the solution has been found because each object in the model is totally self contained.
A2.7 Object Code Sizes

Table A2.3 details the approximate code size of all the objects and the main helper routines used in the program.

To show the relative amount of time spent coding each part of the model, the adsorbent bed object contains 50% more code than all the other process equipment objects combined.

The object independent layer appears to contain twice as much code as the model solver and time manager objects combined. However, much of this code is simply diverting function calls to the right object, and consists mainly of large numbers of IF statements.

<table>
<thead>
<tr>
<th>name</th>
<th>lines of code</th>
<th>code size (kB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>feed</td>
<td>150</td>
<td>24</td>
</tr>
<tr>
<td>product</td>
<td>150</td>
<td>24</td>
</tr>
<tr>
<td>compressor</td>
<td>160</td>
<td>28</td>
</tr>
<tr>
<td>tee</td>
<td>150</td>
<td>28</td>
</tr>
<tr>
<td>fourway</td>
<td>150</td>
<td>29</td>
</tr>
<tr>
<td>valve</td>
<td>400</td>
<td>324</td>
</tr>
<tr>
<td>bed</td>
<td>2200</td>
<td>117</td>
</tr>
<tr>
<td>pipe</td>
<td>400</td>
<td>30</td>
</tr>
<tr>
<td>receiver</td>
<td>130</td>
<td>27</td>
</tr>
<tr>
<td>sequencer</td>
<td>415</td>
<td>93</td>
</tr>
<tr>
<td>eventer</td>
<td>215</td>
<td>18</td>
</tr>
<tr>
<td>controllers</td>
<td>550</td>
<td>54</td>
</tr>
<tr>
<td>recorders</td>
<td>550</td>
<td>68</td>
</tr>
<tr>
<td>object independent layer</td>
<td>1300</td>
<td>34</td>
</tr>
<tr>
<td>model solver</td>
<td>450</td>
<td>12</td>
</tr>
<tr>
<td>time manager</td>
<td>225</td>
<td>6</td>
</tr>
<tr>
<td>graphic routines</td>
<td>750</td>
<td>45</td>
</tr>
<tr>
<td>other functions</td>
<td>750</td>
<td>45</td>
</tr>
</tbody>
</table>

Table A2.3 - Model Code Size
A2.8 Example of a Model Used in this Project

As an example of how one of the models in this project is specified, Figure A2.7 shows a graphical representation of the process equipment for the system being simulated.

As well as the process equipment there will be a controller on many of the valves and instruments to record flows, pressures, and compositions.

The model contains 22 objects, as detailed in Table 2.4. The other two objects shown are the sequencer and the eventer objects. These are defined as objects, but the objUID’s are held in a separate array to the equipment objects, as they are called by the time manager object. Because they are allocated objUID’s it is possible to have more than one of them.
The sequencer object is defined to adjust the valves and set points of the controllers to create the desired PSA cycle. The eventer object will cause any individual events that are required, such as product flow rate disturbances.

<table>
<thead>
<tr>
<th>objUID</th>
<th>name</th>
<th>type</th>
<th>typUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FD_IN</td>
<td>feed</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>FD_T</td>
<td>tee</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>BED_1_FD_V</td>
<td>valve</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>BED_2_FD_V</td>
<td>valve</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>BED_1_FD_T</td>
<td>tee</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>BED_2_FD_T</td>
<td>tee</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>BED_1_WST_V</td>
<td>valve</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>BED_2_WST_V</td>
<td>valve</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>WST_T</td>
<td>tee</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>WST_OUT</td>
<td>product</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>BED_1</td>
<td>bed</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>BED_2</td>
<td>bed</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>BED_1_PD_T</td>
<td>tee</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>BED_2_PD_T</td>
<td>tee</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>PURGE_V</td>
<td>valve</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>BED_1_PD_V</td>
<td>valve</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td>BED_2_PD_V</td>
<td>valve</td>
<td>7</td>
</tr>
<tr>
<td>18</td>
<td>PD_T</td>
<td>tee</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>PD_REC</td>
<td>receiver</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>PD_V</td>
<td>valve</td>
<td>8</td>
</tr>
<tr>
<td>21</td>
<td>PD_PIPE</td>
<td>pipe</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>PD_OUT</td>
<td>product</td>
<td>2</td>
</tr>
<tr>
<td>23</td>
<td>SEQUENCE_1</td>
<td>sequencer</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>EVENT_1</td>
<td>evener</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A2.4 - Example Model Objects
A2.9 Run-time Performance

For the simulations used in this project the typical run-time for the model would be as shown in Table A2.5. The 90MHz Pentium system had 16MB of RAM memory and the 133MHz Pentium had 24MB of RAM. The model was as shown in Section A2.8 and the convergence tolerances as in Section A2.6. The total simulated time was 9000s and real-time performance is defined as:

\[
\text{real-time performance} = \frac{\text{simulated time}}{\text{elapsed run-time}}
\]

<table>
<thead>
<tr>
<th>Integration Step Length (s)</th>
<th>Elapsed Run-time</th>
<th>Real-time Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90MHz (s)</td>
<td>133MHz (s)</td>
</tr>
<tr>
<td>1.0</td>
<td>303</td>
<td>226</td>
</tr>
<tr>
<td>0.5</td>
<td>557</td>
<td>417</td>
</tr>
<tr>
<td>0.25</td>
<td>1030</td>
<td>775</td>
</tr>
<tr>
<td>0.125</td>
<td>1635</td>
<td>1238</td>
</tr>
</tbody>
</table>

Table A2.5 - Model Run-time Performance

The standard integration step length used in this project was 0.5s for all runs except for those involving process control, which had a step length of 0.2s. The model would therefore average between 20 and 8 times real-time, thus the model was more than capable of keeping up with pilot plant in real-time when required.

As can be seen in Table 2.5, and presented graphically in Figure A2.8, the performance scales reasonably with respect to computational load versus the integration step length. The approximate ratio is every time the integration step length is halved the run time increases by 80%. Performance scales approximately linearly with processor speed and is not significantly affected by RAM size, within some restrictions detailed below.
The approximate RAM requirement of the model when running is about 2MB but this increases if more objects are required to be simulated. When sequencing the pilot plant, the RAM requirements could be significantly raised if rapid storage of process data was required. This facility allowed rapid capture of data from the pilot plant at time intervals as small 1/20s, allowing process data from within a cycle to be captured for later study. At 180s/cycle, 1/20s intervals, and 8 pieces of information plus the time to be recorded, the storage requirements are about 800kB/cycle. The data capture was limited to specified cycles only, the total number of which was limited by the amount of RAM present in the computer used. Finally, this data had to be written at the end of the program run as the required data transfer rate to save it in real-time far exceeded the performance of the hard-disk within the computer.

The speed of the 133MHz Pentium processor is 48% faster than the 90MHz Pentium, but the model performance only increases by 34%. This is due to speed limitations in the disk and display sub-systems slightly restricting the processor from achieving its maximum possible speed.

In Table A2.6 a representative sample of the computational load for the main objects is shown. The total run-time was 557s yet the total processing time recorded for the objects is only 446s. The missing time is used by the system itself, i.e. the time manager, the event manager, the sequencer object, the model solver routines, and various general help functions.

By far the biggest user of processing time is the adsorbent bed, using 57% of the time, yet only having 2 objects, compared to the 8 valves which used only 13% of the time. Each valve used 8s of processor time for the entire simulation, whereas each bed required 134s. This shows the relative complexity of the adsorbent bed object compared to a valve object.
Figure A2.8a - Model Run Time vs. Integration Step Size

Figure A2.8b - Real Time Performance vs Integration Step Size
<table>
<thead>
<tr>
<th>Object</th>
<th>Method</th>
<th>AdjPosition</th>
<th>AdjFlow</th>
<th>Flow rate</th>
<th>Pressure</th>
<th>Calculate</th>
<th>CalcEnds</th>
<th>FlowError</th>
<th>CompError</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valves</td>
<td></td>
<td>0.11</td>
<td></td>
<td></td>
<td></td>
<td>60.49</td>
<td>1.59</td>
<td></td>
<td></td>
<td>79.78</td>
</tr>
<tr>
<td>Pipes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.33</td>
<td>0.38</td>
<td>1.70</td>
<td>1.21</td>
<td>8.63</td>
</tr>
<tr>
<td>Beds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>249.84</td>
<td>17.80</td>
<td></td>
<td></td>
<td>267.64</td>
</tr>
<tr>
<td>Tees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49.51</td>
<td>2.25</td>
<td>3.74</td>
<td>4.56</td>
<td>60.05</td>
</tr>
<tr>
<td>Feeds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.93</td>
<td>0.96</td>
<td>0.96</td>
<td></td>
<td>2.25</td>
</tr>
<tr>
<td>Products</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.65</td>
<td>1.76</td>
<td>1.76</td>
<td></td>
<td>4.68</td>
</tr>
<tr>
<td>Receivers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.98</td>
<td>0.44</td>
<td>1.81</td>
<td>1.65</td>
<td>10.88</td>
</tr>
<tr>
<td>Flow recorders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.36</td>
<td></td>
<td></td>
<td></td>
<td>12.36</td>
</tr>
<tr>
<td>Pressure controllers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td>1.65</td>
</tr>
<tr>
<td>Pressure recorders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.12</td>
<td></td>
<td></td>
<td></td>
<td>4.12</td>
</tr>
<tr>
<td>Composition recorders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.36</td>
<td></td>
<td></td>
<td></td>
<td>7.36</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>0.11</td>
<td></td>
<td>2.58</td>
<td>11.51</td>
<td>11.51</td>
<td></td>
<td>402.20</td>
<td>23.46</td>
<td>7.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.02%</td>
<td></td>
<td>0.55%</td>
<td>2.47%</td>
<td>2.47%</td>
<td></td>
<td>86.30%</td>
<td>5.03%</td>
<td>1.56%</td>
</tr>
</tbody>
</table>

Notes: Many methods required very small amounts of processor time and these have been removed from the table for clarity.

The simulation had an integration step length of 0.5s and total elapsed run-time was 557s.

**Table A2.6 - Model Processing Requirements**
The use of the valves as group separators results in a high number of requests for the flow rate and pressure at a valve, as shown by the fact it required 3.8% of the processor time to handle these.

A2.10 Concluding Comments

This appendix has detailed the main objects, methods, and properties for the PSA model used in this project. The input file defines a particular process containing as many beds, vessels etc as computer memory and computational time will allow. In this project only a rather small amount of possible ways of using this flexible model have been exploited.

As every object has a objUID, which is held in an array, there can be more than one of every type, for process equipment this is obviously necessary, however it is also possible to have more than one of every type of object in the model. This leads to the possibility of having more than one PSA plant defined in the same file and hence being simulated at the same time.

The performance of the model has been shown to be almost linear with respect to processor speed, however the run-time rises 80% with every halving of the integration step length. The optimum step length was selected as being between 0.2s and 0.5s, resulting in reasonable run-times for modelling work and yet allowing the model enough spare capacity to sequence the pilot plant in real-time when required.
Appendix 3
3D Characteristic Plots

A3.1 Introduction

Traditionally the representation of the characteristics within an adsorbent bed have been as pairs of two-dimensional plots and separate ones for each step. One graph would plot the characteristic position against time and the other plot the concentration profile against position for one particular time, as shown in Figure A3.1.

With the increase in computer processing power and graphics capability it is now possible to view the bed profiles as three-dimensional (3-D) images. These images can be displayed whilst the model is running and optionally stored as files for later inspection. The use of these 3-D images proved invaluable during this project for two reasons:

1) During development of the model the graphical output provided an real-time representation of what the program was doing within the beds. Some mistakes in the program could be spotted instantly and corrections made to the code.

2) When running simulations the ability to be able to see the bed profiles was a great aid to understanding the behaviour of the process.
Figure A3.1 - Traditional Characteristic Plots
A3.2 Notation

The notation used through this work for all 3-D characteristic plots is shown in Figure A3.2. Time is represented along the x axis (from left to right), concentration on the y-axis (from bottom to top), and position on the z-axis (from the feed end of the bed at the front, z=0, to the product end at the back, z=L).

![Figure A3.2 - 3D Characteristic Notation](image)

To aid the visualisation process the colour of a line at any point represents its concentration, as a mole fraction in terms of the product component. The colour scheme is illustrated in Figure A3.3. All plots are displayed in 256 colours.

![Figure A3.3 - 3D Characteristic Colour Scheme](image)
It is important to note that the product concentration in the 3D characteristic plots is displayed in terms of oxygen and argon, rather than just oxygen as plotted in all other graphs in this work. It would be possible to scale the concentrations to remove the argon content, however because the model lumps oxygen and argon together, as the equations for the theoretical model work for a two component system only, the plots were intended to show the raw internal concentration profiles.

In a real PSA plant base on a zeolite the argon behaves very similarly to the oxygen, this is the main reason why it is very difficult to build a PSA plant which will deliver pure oxygen, or to build a PSA plant which is capable of separating oxygen and argon.

Finally, to make the identification of shock waves easy all shock waves were coloured black. Shock waves represent a discontinuity in the concentration within the beds, as two different concentrations effectively occupy the same location in the bed, see Chapter 3. Extra lines were drawn across the shock wave, i.e. from the low to the high concentration at the same value of z and t, thus giving a much clearer illusion of a vertical front.

A3.3 Three-Dimensional Characteristic Plots

An example of a 3-D plot is shown in Figure A3.4. After many trials with different angles and view-points this layout was decided to be the most appropriate arrangement for visualisation.
A3.4 Geometrical Calculations

There are several calculations that are required to translate the data about a characteristic within the bed in order to create suitable data for the display. The procedure to translate a characteristic is as follows:

i) The concentration at both ends of the characteristic must be converted from the augmented mole fractions, \( Y \), back into real mole fractions, \( y \), (ranging from 0.0 to 1.0).

ii) The value of \( y \) must then be converted into a colour value, by scaling it into the range 45 to 255 (210 out of a possible 256 output colours were reserved for characteristic lines). This is only done once, for the concentration at the starting point of the characteristic as it was not possible to have a line change its colour from one end to another. As the time intervals were small, and hence the characteristics were short, the effect of this was not normally visible.

iii) The values of position, \( z \), concentration, \( y \), and time, \( t \), at both ends of the characteristic, are used to calculate the screen position of the start and end points of the characteristic. The bed position is scaled from a physical position (\( z=0.0 \) to \( z=L \), in units of metres) into a dimensionless position, \( zz \), (ranged 0.0 to 1.0) relative to the start of the bed. The value of \( t \) is translated from being relative to
the start of the model, \( t=0 \), to being relative to the start time for the current cycle, \( t_t \), for example at \( t=3720 \) and \( t_{cycle} = 3600 \), then \( t_t = 120 \), therefore the range of \( t_t \) is from 0 to \( t_{cycle} \).

iv) the screen positions of the start and end point of the characteristic, together with its colour value, are passed to a line drawing subroutine (which was available as a language extension in the FORTRAN77 compiler) which displayed the output on the screen.

This process is repeated for every characteristic within the bed, starting with the characteristic closest to \( z=L \). Extra tests were done to highlight shock waves in black, and to draw extra lines across the shock wave in order to highlight them. The total amount of calculation done in generating the 3-D graphics is significant, see Section A3.5, due to the large amount of geometrical calculations required to translate three-dimensional co-ordinates into two dimensional screen co-ordinates.

<table>
<thead>
<tr>
<th>From</th>
<th>Range</th>
<th>Units</th>
<th>To</th>
<th>Range</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>( k_0 \text{ to } k_N )</td>
<td>-</td>
<td>( y )</td>
<td>0 to 1.0</td>
<td>mf</td>
</tr>
<tr>
<td>( z )</td>
<td>0 to ( L )</td>
<td>( m )</td>
<td>( zz )</td>
<td>0 to 1.0</td>
<td>-</td>
</tr>
<tr>
<td>( t )</td>
<td>0 to ( t_{end} )</td>
<td>( s )</td>
<td>( tt )</td>
<td>0 to ( t_{cycle} )</td>
<td>( s )</td>
</tr>
</tbody>
</table>

Table A3.1 - Notation for Geometry Transformation

A3.5 Problems

There are several drawbacks to the use of this graphic visualisation but all of them would be overcome with the use of a more powerful computer. The first is that the extra computational load required to display the plots is significant. The performance drop is shown in Figure A3.5. A fairly coarse graphical display of 640 by 480 pixels increases program run time by 6%, rising to 18% for a resolution of 1280 by 1024 pixels. The end point for each line must
be translated from bed co-ordinates to display co-ordinates, a procedure which requires many floating point calculations, and the actual display pixels that make the line must then be calculated, a process requiring a number of integer calculations.

Secondly, the time taken to save the data, if required, is also significant. For a display of 640 by 480 pixels run time rises by 5%, rising to 6% at 1280 by 1024 pixels. Interestingly, the time required to save the data is not linear with respect to image size, having one value at or below 800 by 600 pixels (25s) and another above this resolution (35s). This can only be explained by a significant change in the method of memory organisation and data access for the graphics adapter, which occurs at different display resolutions, and may therefore be adapter dependent (in this case the adapter was a Diamond Stealth with 2MB of RAM).

Hidden line removal could be used in order to make the visualisation of the profiles easier, however the computational requirements for this kind of procedure are high. Therefore an alternative method was selected, that of always drawing the profile from the back (z=L) to the front (z=0). With this method lines closer to the front of the bed overwrite those further to the back, which had already been plotted. One unexpected advantage of this method was that lines drawn 'behind' were often left with parts still visible at the end of the drawing process, thus making the profiles appear 'transparent'. This allowed features that would disappear from view with a hidden line removal procedure to still be partly in view.

The final problem was related to that of interrupt routines taking too long, as discussed in Section 4.7.3.3.2. Use of the graphical display caused interrupts to be disabled for a significant amount of time as the display was updated. If the plant was being sequenced by the model, the control of the plant would be disrupted because the plant code was located in a time-based interrupt routine the programming of which assumed it would be called at specific times.
Figure A3.5a - Model Run Time vs. Graphics Load

Figure A3.5b - Real-time Performance vs. Graphics Load
A3.6 Example Cycle Profiles

The standard cycle, as shown in Figure A3.6, was used to generate the plots in this section. Pressurisation of the product receiver started at 1440s. After this time, the product flow rate from the plant increased until it reached its set point of 8.0 dm³/cycle.

To illustrate the usefulness of the graphical plots to understand the start-up of a PSA process a brief description of the important features of each plot is given.

Figure A3.7 shows the first cycle for Bed A, which starts with a pressurisation step. However, the pressurisation step was not the first step in the sequence, it came about naturally because the planned step was a product release step, and so the feed end of the bed was open to the feed supply. This is an example of how flexible the program is, relying on the pressure drop across, and position of, valves to determine flows around the system.
The cycle continues with a product release step which also donates gas to Bed B, followed by a further period of normal product release, although at this point no product is being released from the beds. After depressurisation the bed is purged with gas from Bed A and is finally repressurised ready for product release.

In Figure A3.8 the first cycle for Bed B is shown. This sequence was started with a depressurisation step, which had no effect as the bed was already
at atmospheric pressure. The bed is then purged with product gas from Bed A, followed by a null step where the profiles are stationary. Pressurisation of the beds then occurs, immediately followed by a product release step and a product release step which is donating gas to Bed A for purging.

It is immediately obvious that there are different concentration profiles in the beds during the start up phase of this process. This is because the two beds are modelled separately. The differences in the two beds during startup is due to the way in which the sequences for the two beds were started. If the start of the step sequences had been restricted to only starting with a pressurisation step, with no action before that time, the concentration profiles for the two beds would be more symmetrical. In Appendix 4 it will be shown that final performance of the process can depend on the way the process is started.

![Figure A3.9 - Bed A, Cycle 4](image)

The 4th process cycle is shown in Figure A3.9 where Bed A has built up the internal concentration to a maximum value of around 0.85. Product gas being transferred as purge gas for Bed B is of much higher quality, as is the purge gas being received back from Bed B.
In Figure A3.10 the corresponding cycle is shown for Bed B. The continuous swapping of high quality purge gas between the two beds has resulted in the process rapidly building up its product concentration to near its maximum in very few cycles. The interesting feature of this cycle is that the product concentration has built up sufficiently that the velocity of the key shock wave, formed during the pressurisation step, is slow enough that during the donation of purge gas to Bed A not all of the high purity gas is lost. This is important because keeping some high purity gas within the bed at the end of product release will greatly enhance the cleaning of the bed during the subsequent depressurisation step.
By cycle 10 the product concentration within the beds has built up to maximum, as shown in Figure A3.11. Product release started to the product receiver at 1440s (at the start of cycle 9) and so additional product is being withdrawn from each bed, the amount of purge gas required per cycle is 17.2 dm³/bed and the product amount delivered on each cycle is now 4.0 dm³/bed. Each bed is now having to deliver nearly 25% more product gas than before.
The product concentration this system is capable of delivering is only a mole fraction of 0.64 oxygen. Currently the process is delivering a maximum purity product (0.9575 oxygen plus 0.0425 argon), however this is the last cycle for which the high concentration gas is kept within the bed at the end of the product release step. The average purity of the gas exiting the beds will now be significantly reduced as breakthrough of low purity gas from the feed end of the bed occurs.

In Figure A3.13 a cycle is shown for the final steady state of the process. It can be seen that breakthrough of the feed gas occurs in the product release step for Bed A before the end of the purge step on Bed 2. Therefore, low purity gas is being used to purge Bed B at the end of the purge step, although because it has been mixed with high purity product gas in the dead volume at the end of the bed, the purge gas concentration does not drop to that of the breakthrough gas.

Figure A3.13 - Bed A, Cycle 20

At the start of the depressurisation step (at t=3510) there is a small step up in concentration at the product end of the bed. This is gas from the dead volume at the product end of the bed re-entering the bed. The equivalent cycle for Bed B is shown in Figure A3.14. The concentration profiles within the two
beds are now virtually identical, any differences due to the asymmetry during the start up phase have gone.

![Figure A3.14 - Bed B, Cycle 20](image)

### A3.6 Concluding Comments

The inclusion of 3D characteristic plots as an optional feature for the program was of great benefit during this project. During development of the program it clearly showed mistakes in the code which caused the characteristics to be mis-handled.

The ability to see the concentration profiles allowed a much clearer understanding of the way that the fronts build up within the beds and it was relatively easy to see the effects of changing various parameters on the location of the concentration fronts. Together these gave a much clearer insight as to why the process behaves as it does.

In particular, in Appendix 4, some multiple steady state behaviour will be explored using these visualisation techniques.
Appendix 4
Non-Linear Behaviour

A4.1 Introduction

An effect that has been alluded to within this dissertation as being observed at various stages in this project are cycles that can produce different outputs for the same inputs, depending on the history of the process.

This type of behaviour is interesting for two reasons:
i) There are some references within the literature to multiple steady states for PSA processes, however there does not seem to be any mention of this effect for isothermal processes with linear isotherms, Farooq et al (1988) and Ritter & Yang (1991).

ii) most control theories are based on the assumption that for a single set of input conditions there can only be one output.

It is believed that only by modelling the adsorbents beds individually within a PSA system, as the program written in this project was designed to do, can this effect be seen. As previous models have concentrated on modelling a single bed, and assuming symmetry for all others, this important effect may have remained hidden.

In Appendix 3, when 3-D characteristic plots were introduced, it was mentioned that the way the process is started has a significant effect on the concentration profiles within the beds and it is possible for different profiles to develop within each bed. In this appendix it will be shown that, depending only on the history of the process, two different product concentrations can be produced by the same cycle conditions.
A4.2 Final Concentration

The most obvious way to spot the effect is by the final product concentration. Many plots containing the results from two or more sets of conditions have been shown previously.

With the conditions as shown in Figure A4.1, and with a product amount of 8.0 dm³/cycle, a set of experiments were run on the model. Each run of the model differed only in the time that product was first withdrawn from the system. Product release from the beds occurs 180s before the product valve opens, in order to allow time for the product receiver to pressurise.

![Figure A4.1 - Standard Cycle](image)

In Figure A4.2 the results for several different product release start times are shown, for the first 5040s (28 cycles) of the model run. The product concentration was measured at the outlet of the product valve, thus eliminating any time delay due to the length of the product pipe. The result for the product release time of 1440s (8 cycles), as used throughout the project, is highlighted by a thicker line.

It can clearly be seen that the results fall into two sets of results. One set of results produces a product concentration averaging 0.6358, the other produces product at 0.6864, a jump of 0.0506 or 8% of the lower value. The results within each group are the same to four decimal places.
Figure A4.2 - Product Concentration For Various Product Release Start Times
A4.3 Bed Concentration Profiles

With the ability to view 3-D characteristic plots of the concentration fronts within the beds it is possible to see the differences between the two results at the steady state conditions. The results chosen were for product release start times of 1440s and 1720s, i.e. the two adjacent results that spanned the two product concentrations.

The steady state concentration profiles are shown in Figure A4.3 for 1440s and Figure A4.4 for 1720s. The final concentration profiles appear almost
identical for each set of conditions. Both appear to have symmetrical profiles, equal purge depth of the bed and breakthrough of the feed gas at the same time. However, there appears to be a small difference in the end of the purge step at approximately 5175s.

Figure A4.4a - 1720s Start Time, Bed 1, Cycle 29

Figure A4.4b - 1720s Start Time, Bed 2, Cycle 29

By expanding the area of interest, as in Figure A4.5, it can now be clearly seen that the purge appears to be of slightly higher quality at the end of the step for the 1720s condition. This can be traced back to the duration of high quality
gas release during the product release step being slightly higher, as expanded in Figure A4.6.

In addition, as the product concentration is slightly higher for the 1720s process, there is a slightly higher purity gas accumulated in the dead space at the product end of the bed during the product release step. When the bed is depressurised this gas finishes slightly closer to the feed end of the bed than for the 1440s conditions, thus cleaning the bed slightly more.

**A4.3.1 Bed Concentrations at Product End**

In Figure A4.7 the concentration of the gas leaving Bed 1 has been plotted for both sets of conditions. The main differences have already been accounted for in the preceding paragraphs, however there are also some slight differences in the time at which the shock wave leaves the bed. The shock wave for the 1720s conditions leaves the bed slightly later than for the 1440s conditions. By careful inspection of the Figure A4.7 it was found to leave 0.5s later, however as this is in the period when purge gas is being supplied the flow rate is higher than during the rest of the product release period. The extra product gas being released is equivalent to 0.42 dm$^3$ of extra product. This is almost exactly equal to the extra oxygen required to account for the higher amount of oxygen in the product for the 1720s process.
Figure A4.5a - Purge Detail for 1440s Start Time, Bed 1

Figure A4.5b - Purge Detail for 1720s Start Time, Bed 1
Figure A4.6a - Product Release Detail for 1440s Start Time, Bed 1

Figure A4.6b - Product Release Detail for 1720s Start Time, Bed 1

Appendix 4 - Non-Linear Behaviour
A4.5 Transition Between States

As the only difference between the two conditions at steady state is the quality of the gas used to clean the beds, it may be possible to switch the process between the two states.

The most obvious way to do this is to temporarily reduce the product flow rate for the poorer performing conditions (1440s). By doing this it should allow more high quality product gas to remain within the beds, improving the clean up for that period. When the flow rate is returned to normal the higher quality gas should stay within the system, allowing the process to perform better. By reversing the actions for the better performing conditions (1720s) it should be possible to remove some purge gas and hence drop the performance.

The results are plotted in Figure A4.8. The product flow rate was reduced to 6 dm$^3$ for the poorer performing conditions (1440s) and raised to 10dm$^3$ for the better performing conditions (1720s), for a period of 180s (1 cycle) in order to ensure the disturbance affects both beds. It can be seen that the 1440s line has been moved up to the purity where the 1720s line was, and vice versa.

It did not prove possible to find the existence of any other steady states with these cycle conditions. All other attempts to find a third stationary state failed.

A4.6 Chapter 6 Results

In Sections 6.3.1 and 6.3.2 many graphs were presented of performance versus dead volumes, either at the feed or product end. These graphs contained many ‘ragged’ sets of results, where steady state product concentration suddenly dropped after only a small increase in a dead volume. This was explained as the process being very sensitive to the final location of the shock waves within the bed and that slight errors in the positions could accumulate with time. However, a supplementary explanation is presented here, which has its basis on the findings above.
Two results were chosen from Section 6.3.2, taken from Figure 6.10 as being on either side of a sudden drop in performance, with the following conditions:

A) product end dead volume 0.388 dm$^3$,
B) product end dead volume 0.369 dm$^3$.

The purge rate was 17.2 dm$^3$/part cycle, the product rate at 8.0 dm$^3$/cycle, and the process first delivers product gas at 900s (5 cycles). Each process was then subjected to a disturbance in product flow rate, as above, at 3600s to attempt to move either of the two results to a new steady state.

The results are shown in Figure A4.9. For conditions set A it proved possible to switch the process to a new steady state, much closer to that for set B. For set B conditions, disturbing the process with a decrease in the product flow rate had no effect on the final steady state of the process. Many different size disturbances were tried, but all returned to the original steady state.
Figure A4.8 - Effect of Product Flow Rate Disturbance
Figure A4.9 - Alternative States for Other Conditions
A4.7 Discussion & Conclusions

Two initial conditions, differing only in the time at which product is first withdrawn from the plant, have been shown to give different product concentrations at steady state. Most process control theory is for linear processes, i.e. ones that do not exhibit this type of behaviour.

After close inspection of the final steady state data available, in 3-D characteristic plots and concentration versus time plots, there is obviously a slight difference in the two final operating states, where one state is producing slightly more high quality product gas due to slightly better cleaning of the bed during the depressurisation and purge steps. The reason for this is attributable solely to the time of product release from the beds. It appears that there is an optimum time for the first release of product from the beds, before this time the process can never achieve the maximum concentration the system is true capable of, however delaying the release for longer than this time gives no further performance gain.

When comparing the results of the new program with those obtained by Espitalier-Noel (1988), it was seen that the different product concentrations were obtained for what appeared to be the same cycle conditions. It is possible that by having a different time for the first product release that the product concentration would have been affected. The differences were all attributed to the effect of adding the surrounding process equipment into the model.

Results were also presented in this appendix of the effect of multiple steady states at other conditions, in an attempt to explain some of the 'ragged' results found in Chapter 6. It was again shown possible to switch the process between two states for each condition. This multiplicity of steady states is therefore offered as an alternative explanation to errors in the shock wave positions, although the effect is the same. It is argued that the shock wave positions are not in error but are simply in a slightly different position due to the history of the process.
Multiple steady states have been reported in the literature for some processes, however these were obtained by either starting with completely clean or completely saturated beds, in which case a different steady state could be achieved with the same cycle conditions, Farooq et al (1988), Ritter & Yang (1991), and in thermal swing adsorption processes, LeVan (1990). It is also suggested, Farooq et al (1988), that the only case where multiple steady states would not be found is for isothermal systems with linear isotherms.

It is likely that multiple steady states do not exist for all cycles, and that multiple steady states require breakthrough of the feed gas during the period of the product release step when purge gas is being directly supplied to another bed.

The model used in this work assumes the process is isothermal and has a linear isotherm, yet two steady states have been found, both starting from clean beds, a result not previously seen. As all previous PSA models in the literature appear to model only one bed, and assume that the conditions within it are it is representative of all others differing only in time, it is suggested that only by modelling the adsorbent beds separately can this effect be observed.
A5.1 Introduction

In order to provide a representative model of the process it is necessary to match the model to the process being simulated. Data from the PSA pilot plant was used to ‘tune’ the model, although some experiments were done at other conditions. The cycle used on the plant and in the model is shown in Figure A5.1.

```
<table>
<thead>
<tr>
<th>Bed A</th>
<th>PD</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bed B</td>
<td>PU</td>
<td>NU</td>
</tr>
<tr>
<td>0s</td>
<td>90s</td>
<td>40s</td>
</tr>
<tr>
<td>90s</td>
<td>120s</td>
<td>150s</td>
</tr>
<tr>
<td>60s</td>
<td>130s</td>
<td>180s</td>
</tr>
<tr>
<td>150s</td>
<td>150s</td>
<td>180s</td>
</tr>
</tbody>
</table>
```

**Figure A5.1 - Purge Cycle Used**

A5.2 Unmatched Model

To match the pilot plant data the model was initially given a set of default parameters for all the controllers. These were chosen simply by looking at the data from the pilot plant and making the set points of the controllers equal to the value required. The tuning constants for the controllers were simply guessed from previous runs of the model.

Even with such rough initial inputs the model simulates the pilot reasonably well. In Figure A5.2 the pressure profile has been plotted for one bed for one cycle of the process. The model pressure is too high at the end of the pressurisation step (150 to 180s) and it has pressurised at too high a rate. During the donate purge gas period (30 to 60s) it has not depressurised enough, and during the final stage of product release (40 to 90s) it is still over-predicting.
the bed pressure. During the depressurisation step (90 to 120s) it has depressurised too slowly initially, but reaches too low a final pressure. In the purge step (120 to 130s) it has the pressure about right.

In Figure A5.3 the profile for the feed flow rate to the plant is plotted and it can be seen that the feed flow rate is reasonable for most of the cycle. However, in the pressurisation steps (60 to 90s and 150 to 180s) the model is completely different, predicting almost twice the flow at the start of the step, with a more rapid decline of flow towards the end. This can be partly explained by the bed pressure rising too fast during the pressurisation step, see Figure A3.2, and partly as the gas flow rate in the model is not being restricted at sonic velocities in the feed valve.

In Figure A5.4 the purge flow rate profile is plotted. The model predicts a much more regular flow rate, and fails to predict the extra flow after the purge step has supposed to finish. This extra flow is probably due to the purge valve failing to seat at the end of the purge time.

The waste gas flow rate from the plant is plotted in Figure A5.5. The flow rate is too high initially during the depressurisation steps (0 to 30s and 90 to 120s), and is too low at the end. The waste flow rate during purge is also wrong, being of too short duration, but almost the right height.

The more important flows and pressures are given in Table A5.1, together with the final product purity produced by the model.
<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
<th>Units</th>
<th>Plant</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed</td>
<td>pressure</td>
<td>Pa</td>
<td>543000</td>
<td>554000</td>
</tr>
<tr>
<td></td>
<td>feed flow</td>
<td>dm³</td>
<td>121.5</td>
<td>121.6</td>
</tr>
<tr>
<td>Product Release</td>
<td>pressure</td>
<td>Pa</td>
<td>537000</td>
<td>554000</td>
</tr>
<tr>
<td>Depressurisation</td>
<td>pressure</td>
<td>Pa</td>
<td>126000</td>
<td>105000</td>
</tr>
<tr>
<td></td>
<td>waste flow</td>
<td>dm³</td>
<td>127.8</td>
<td>135.5</td>
</tr>
<tr>
<td>Purge</td>
<td>pressure</td>
<td>Pa</td>
<td>153000</td>
<td>148000</td>
</tr>
<tr>
<td></td>
<td>purge flow</td>
<td>dm³</td>
<td>12.8</td>
<td>11.9</td>
</tr>
<tr>
<td>Overall</td>
<td>product</td>
<td>mf</td>
<td>0.673</td>
<td>0.576</td>
</tr>
</tbody>
</table>

Table A5.1 - Plant and Unmatched Model Values
Figure A5.3 - Feed Flow Rate Profile for Unmatched Model
Figure A5.5: Waste Flow Rate Profile for Unmatched Model
A5.3 Matched Model

The model was then “tuned” to match the pilot plant profiles. Controller values and valve sizes were adjusted until a good match for the pressurisation rates was found. The flow rate through the valves was restricted when the flow was sonic.

In Figure A5.6 the pressure profile has been plotted for the matched mode for a cycle of the process. The model now pressurises at the right rate (150 to 180s) to the correct final value. During the donate purge gas period (30 to 60s) it has still not depressurised quite enough, but it is much closer than before. In the final stage of product release (40 to 90s) it now predicts the bed pressure correctly. During the depressurisation step (90 to 120s) it still depressurises too slowly initially, but now reaches the correct final pressure. In the purge step (120 to 130s) it has the pressure about right, although it depressurises slightly towards the end of the step.

In Figure A5.7 the profile for the feed flow rate to the plant is plotted and it can be seen that the feed flow rate is very good for most of the cycle. In the pressurisation steps (60 to 90s and 150 to 180s) the model is now very closely matched to the plant profile.

In Figure A5.8 the purge flow rate profile is plotted. The extra flow at the end of the purge step has now been included, although the model always predicts a much more regular flow rate.

The waste gas flow rate from the plant is plotted in Figure A5.9. The flow rate is still slightly too high during the start of the depressurisation steps (0 to 30s and 90 to 120s) and too low at the end. The waste flow rate during purge is still wrong, being of the right shape but the rises too and falls too slowly.

The more important flows and pressures for the model matched to the plant are given in Table A5.2, together with the final product purity produced by the model which is now closer to that of the plant, but still around 5% out.
Figure A5.6 - Pressure Profile for Matched Model
## Table A5.2 - Plant and Matched Model Values

<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
<th>Units</th>
<th>Plant</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed</td>
<td>pressure</td>
<td>Pa</td>
<td>543000</td>
<td>543000</td>
</tr>
<tr>
<td></td>
<td>feed flow</td>
<td>dm³</td>
<td>121.5</td>
<td>121.3</td>
</tr>
<tr>
<td>Product Release</td>
<td>pressure</td>
<td>Pa</td>
<td>537000</td>
<td>542000</td>
</tr>
<tr>
<td>Depressurisation</td>
<td>pressure</td>
<td>Pa</td>
<td>126000</td>
<td>126000</td>
</tr>
<tr>
<td></td>
<td>waste flow</td>
<td>dm³</td>
<td>127.8</td>
<td>122.3</td>
</tr>
<tr>
<td>Purge</td>
<td>pressure</td>
<td>Pa</td>
<td>153000</td>
<td>154000</td>
</tr>
<tr>
<td></td>
<td>purge flow</td>
<td>dm³</td>
<td>12.8</td>
<td>12.6</td>
</tr>
<tr>
<td>Overall</td>
<td>product</td>
<td>mf</td>
<td>0.673</td>
<td>0.625</td>
</tr>
</tbody>
</table>

### A5.4 Concluding Comments

The model currently requires ‘tuning’ to the pilot plant data with regard to pressurisation rates. Once this has been done the flow profiles follow those on the plant very well. The main problem areas are:

1) the flow meters in the plant often saturate. This means that the model loses the ability to precisely match the actual flows around the plant.
2) the flow meters do not appear to always respond fast enough to the rapid changes in flow rate that occur as valves open.
3) the purge valve does not seem to close properly at the end of the purge period. This allows extra purge gas to be donated to the purging bed.

Most of these problems can be partially overcome, but the facility to fully do so was not included in the model. For example, a leakage rate could be added for the leaking purge valve, with the ability to reduce the leak rate against time for purge flow rate could be matched much more accurately.