SIMULATION AND OPTIMISATION OF THE CONTROLS OF THE STOCK PREPARATION AREA OF A PAPER MACHINE

Sébastien Lacour

Dipl.Chem.Eng. (INP, Toulouse)

Submitted in fulfilment of the academic requirements for the degree of

Master of Science in Engineering

in the

School of Chemical Engineering,

University of KwaZulu-Natal, Durban

Durban

June 2004

Preface and acknowledgments

Finishing university, one is likely to look back at what was achieved and how it contributed to the development of one's personality. The opportunity of doing a master's degree in South Africa was given to me when completing my bachelor degree in France. This was a real challenge. Those two years were a nice transition from university to the industrial world. I learnt to be responsible, independent in my work, I opened up to people, and more importantly I learnt to listen to others. I developed a motivation and an enthusiasm for this specific area of process control. Its challenging aspect resides in its nature: it is adaptable to every process. It requires one to understand the process and then to adapt his knowledge to improve it. Therefore, it has definitely a big potential for the future in the industries.

I would like to express my gratitude to the following people/organisations for their contribution towards this research.

My supervisor M. Mulholland, without whom nothing would have been possible. Thank you for your patience, your attentive ear, and your support during those two years.

A. Leske, for his advice and his insights on the papermaking process.

Eugene Muthen, Selvum Moodley and Julian Randelhoff for their collaboration and their explanations regarding Mondi's papermaking process.

Mondi for its financial support during this research.

My colleague post-graduate students, too many to be specific, who played an important role in the achievement of this work, by their help, their support and their care.

My family, close and extended, and friends for their sincere support about this experience overseas, the encouraging mails, and the love.

I, Sébastien Lacour, declare that unless indicated, this dissertation is my own work and that it has not been submitted in whole or in part for a degree at another University or Institution.

S.Lacour	Professor M. Mulholland
29/09/2004	6/5/2005
Date	Date

Abstract

At Mondi Paper Ltd, Merebank, South of Durban, Paper Machine 2 has recently been transferred onto a Distributed Control System (DCS). This was seen as a good opportunity to enhance the control of the pulp feed to the machine. A prime concern in operating a paper machine is to ensure consistent set-point paper properties in the Cross-Direction (CD: ie. across the paper width) and in the Machine-Direction (MD: ie. along the paper length).

Sophisticated adjustments are available to ensure an even feed of the stock (consistencies around 2% $^{m}/_{m}$ wood fibres in water) from the head-box across the receiving width of the paper machine. The properties of prime interest as the pulp is pumped through the head-box distributor onto the receiving belt of the machine are the basis weight (fibre mass per unit area) and moisture content (per unit area). However, the distribution system is highly dependent on the properties of the stock as it arrives at the head-box. Variations in upstream chest levels, the supplied pressure, flow-rate and fibre/water ratio, all cause MD and even CD variations. The problems of maintaining steady conditions at the head-box are well known, and are understood to arise from sub-optimal control in the preceding section involving a blend chest and machine chest, amongst other items, where several pulp streams and dilution water are combined. A number of control loops are involved, but appear to require different tuning for different paper grades. Often individual loops are taken off-line.

In this study, an understanding of the controller interactions in the stock preparation section has been developed by detailed dynamic modelling, including all of the existing control loops. The model is built up in a modular fashion using a basic element, having one input (which can collect multiple streams originating elsewhere) and four outputs, linked through a vessel of variable volume. Several basic elements are linked together to form the overall system. All of the necessary properties can be defined so that the model allows the simulation of all features of the network: vessels, pipes, junctions, valves, levels and consistencies. A set of first order differential equations is solved which includes total water balance, species mass balances, derivatives of flow controller action, and derivatives of supervisory controller action. Supervisory controllers for consistency or level cascade onto flow controllers. Flow

controllers manipulate valves which give a first-order dynamic response of actual flow. Where valves are manipulated *directly* by the supervisory level, the flow controller is effectively bypassed.

This study involves a constraint problem around the blend chest, resulting in a loss of specification at the paper machine. This was solved by the implementation of a static optimiser. Its objective function penalizes deviations from setpoint of five parameters (ratios, consistency and level) using respective weight factors. Both the model and its optimiser were included in a simulator designed with the graphical user interface (GUI) of Matlab. The simulator has then been used to explore control performance over the operating range, by means of a set of scenarios.

Table of contents

Preface and acknowledgments........ii

Abstract........iv

Table of content........vi

List of figures and tables........ix

List of symbols........xi

Chapter 1 : Introduction........**1-1**

1.1 Modelling and simulation........*1-2*

1.2 Modelling in the pulp and paper industry........*1-3*

1.3 Papermaking process........*1-4*

1.4 Problems faced by the industry........*1-5*

1.5 Objectives of this thesis........*1-5*

1.6 Dissertation layout........*1-6*

Chapter 2 : Paper industry and papermaking process........**2-1**

2.1 Paper industry........*2-1*

2.1.1 Paper origins........2-1

2.1.2 Papermaking origins........2-1

2.1.2.1 Geographic evolution........2-1

2.1.2.2 Process evolution........2-2

2.1.3 Paper industry in the digital age........2-3

2.1.4 Mondi and South Africa in the paper industry........2-3

2.2 Papermaking process........*2-4*

2.2.1 Wood pulp........2-4

2.2.2 Papermaking process........2-5

2.2.3 Quality management........2-9

2.3 Stock preparation area and approach flow........*2-11*

2.4 PM2 Paper Machine........*2-13*

Chapter 3 : Literature review........**3-1**

3.1 Process control........*3-1*

3.1.1 Interactions........3-1

3.1.2 Multivariable control........3-2

3.1.2.1 Relative gain array (RGA)........3-4

3.1.2.2 Inverse Nyquist array (INA)........3-4

3.1.2.3 Compensator........3-4

3.1.3 Adaptive control........3-5

3.1.3.1 Gain scheduling........3-7

3.1.3.2 Model reference adaptive control (MRAC)........3-8

3.1.3.3 Self-tuning adaptive control (STAC)........3-9

3.2 Modelling and simulation *3-10*
3.2.1 Modelling 3-10
3.2.1.1 Model classification 3-10
3.2.1.2 Model construction 3-11
3.2.1.3 Model refinement and validation 3-12
3.2.2 Simulation 3-13
3.2.3 Simulation environment 3-14
3.3 Optimisation *3-15*
3.3.1 Performance index 3-16
3.3.2 Optimisation methods 3-16
3.3.2.1 Direct methods 3-17
3.3.2.2 Indirect methods 3-18
3.3.2.3 Constrained optimisation 3-19
3.4 Modelling and control in the pulp and paper industry *3-20*
3.4.1 Advanced control 3-21
3.4.2 Dynamic models 3-22
3.4.2.1 Refiners 3-22
3.4.2.2 Stock preparation and wet end 3-23
3.4.2.3 Paper machines 3-23
Chapter 4 : Modelling and control 4-1
4.1 The stock preparation area model: two different approaches *4-1*
4.1.1 Equation-solving approach 4-2
4.1.1.1 Description of the approach 4-3
4.1.1.2 Discrete form 4-6
4.1.1.3 Problems 4-8
4.1.2 Modular approach 4-9
4.1.2.1 Description of the approach 4-9
4.1.2.2 Program structure 4-11
4.1.2.3 Modelling devices 4-13
4.2 Flow diagram of the program *4-17*
Chapter 5 : Optimisation 5-1
5.1 Blend Chest constraint problem *5-1*
5.2 The solution: Constrained optimisation *5-5*
5.3 Optimisation method *5-7*
5.4 Programming *5-8*
5.4.1 Simplified flow diagram 5-8
5.4.2 Identifying filter 5-11
5.4.3 Improvements 5-12
Chapter 6 : Simulation 6-1
6.1 The graphical user interface (GUI) *6-1*
6.2 GUI design *6-3*
6.3 GUI screen shots *6-4*
6.3.1 Level perturbation response 6-5
6.3.2 Flow perturbation response 6-6
6.3.3 Consistency perturbation response 6-6

Chapter 7 : Results **7-1**
7.1 Data collected from the plant *7-1*
7.2 Tuning of the controllers *7-4*
7.2.1 Tuning of cascaded PI controllers 7-4
7.2.2 Interaction between controllers 7-4
7.3 Simulator : Case study *7-5*
7.3.1 Parameters of the study 7-5
7.3.2 Unconstrained situations 7-9
7.3.2.1 Consistency ramp 7-9
7.3.2.2 Total flow ramp 7-10
7.3.2.3 Ratio ramp 7-11
7.3.3 Constrained situations 7-12
7.3.3.1 Consistency ramp 7-13
7.3.3.2 Total flow ramp 7-14
7.3.3.3 SF2 ratio ramp 7-14
7.4 Conclusion *7-15*
Chapter 8 : Conclusion and recommendations **8-1**
8.1 Conclusion *8-1*
8.2 Recommendations *8-3*

References R-1

Appendices

Appendix A: Different types of pulp A-1
Appendix B: Plant data B-1
Appendix C: Listing of the simulation program C-1
Appendix D: Matlab software D-1
Appendix E: PID controllers E-1

List of figures and tables

Figure 1.1: Papermaking process simplified diagram 1-4

Figure 2.1: Dry and wet end operations 2-6

Figure 2.2: Three variation types 2-9

Figure 2.3: Stock preparation area 2-11

Figure 2.4: Air-cushioned headbox 2-13

Figure 3.1: Multivariable process (Leiviska, 1999) 3-2

Figure 3.2: Diagonal control strategy in multivariable system (Leiviska, 1999) 3-3

*Figure 3.3: Decoupling controller for 2*2 process (Leiviska, 1999)* 3-5

Figure 3.4: Gain scheduling structure (Leiviska, 1999) 3-7

Figure 3.5: Model reference control block diagram (Leiviska, 1999) 3-8

Figure 3.6: Self-tuning controller (Leiviska, 1999) 3-9

Figure 4.1: Test example for the equation-solving approach 4-2

Figure 4.2: Matrix representation 4-6

Figure 4.3: Modular approach basic element 4-9

Figure 4.4: Program structure 4-11

Figure 4.5: Model 4-12

Figure 4.6: Overflow model 4-14

Figure 4.7: Separation process model for filters, cleaners and screens 4-15

Figure 4.8: Blend chest stock preparation 4-16

Figure 4.9: Simplified model flow sheet diagram 4-18

Figure 5.1: Mondi's blend chest control system 5-2

Figure 5.2: Short fiber refiner consistency controllers 5-3

Figure 5.3: Feedback control 5-4

Figure 5.4: Inside broke to blend chest flow controller (constraint problem) 5-5

Figure 5.5: Optimiser feedforward setpoint determination 5-6

Figure 5.6: Cartesian progression 5-7

Figure 5.7: Optimiser simplified flow sheet 5-10

Figure 5.8: Blend chest stock preparation 5-12

Figure 6.1: Simulator 6-2

Figure 6.2: Level setpoint step response 6-5

Figure 6.3: Machine chest level response 6-5
Figure 6.4: Cascaded flow controller response 6-5
Figure 7.1: Short fibre flow control 7-3
Figure 7.2: Blend chest level control 7-3
Figure 7.3: Short fiber refiner consistency control 7-4
Figure 7.4: Blend chest level 7-5
Figure 7.5: Flow controller reaction 7-5
Figure 7.6: Steady state simulation 7-8
Figure 7.7: Unconstrained simulation – consistency ramp 7-10
Figure 7.8: Unconstrained simulation – total flow ramp 7-11
Figure 7.9: Unconstrained simulation – SF2 ratio ramp 7-12
Figure 7.10: Constrained situation – consistency ramp 7-13
Figure 7.11: Constrained situation – total flow ramp 7-14
Figure 7.12: Constrained situation – SF2 ratio ramp 7-15

Table 5.1: Listing abbreviations 5-9
Table 7.1: Unit dimensions and controller settings 7-2
Table 7.2: Initial values of the parameters of the study 7-6

List of symbols

Acronyms

CC - Consistency Controller

CD - Cross Direction (across paper width)

CO - Output Signal

CV - Controlled Variable

DCS - Distributed Control System

FC - Flow Controller

GPC - Generalized Predictive Control

GUI - Graphical User Interface

GUI - Graphical User Interface

GWD - Groundwood

IB - Inside broke

INA - Inverse Nyquist Array

LC - Level Controller

LF- Long fibre

LP - Linear Programming

MD - Machine Direction (along paper length)

MRAC - Model Reference Adaptive Control

MV - Manipulated Variable

ODE - Ordinary Differential Equation

PFE - Partial Fraction Expansion

PID - Proportional, Integral and Derivative

PM - Process Measurement

PM2 - Paper Machine 2

QPI - Quadratic Performance Index

RFP - Recovered Fibre Pulp

RGA - Relative Gain Array

SF - Short Fibre

SP - Setpoint

STAC - Self Tuning Adaptive Control

STC – Self Tuning Controller

STR - Self Tuning Regulator

TMP - Thermo-Mechanical Pulp

Variables, Vectors and Matrices

Chapter 4 – Equation-solving approach (Modular approach)

m_k - volume of the tank k (V_k)

$\overline{m}$ - volume vector ($\overline{V}$)

$\overline{m}_{SP}$ - volume setpoint vector

$f_{i,k}$ - input flow i of the tank k

$f_{o,k}$ - output flow o of the tank k

$\overline{f}$ - flow vector

$\overline{f}_{SP}$ - flow setpoint vector

c_k - consistency of the tank k (y_k)

$\overline{c}$ - consistency vector ($\overline{y}$)

$\overline{c}_{SP}$ - consistency setpoint vector

k_c, k_i - proportional and integral term matrices for the PI controllers

T - time constant on the valve response

$\overline{x}$ - valve position vector

$\overline{y}$ - system input vector ($\overline{L}$)

$\overline{u}$ - system external input vector ($\overline{N}$)

Chapter 5 – Optimisation

J - objective function for the optimisation

Jopt - optimised objective function J

α_i - supply flows ratio

α_{iSP} - supply flows ratio setpoint

φ_i - weighting factor for the objective function

ev - exploration vector of the manipulated variables for the optimisation

ev_{new} - temporary exploration vector for the optimisation search

$ev_{\max}$ - maximum range vector for the exploration vector

FFMR - Fixed Fraction of Maximum Range

Chapter 1 : Introduction

The need for paper in the historic civilisations has been present for several reasons. It was used as a means of communication as well as a means of recording and passing information. Despite the advent of computers, the demand for paper is still present. Data continues to be collected and saved on paper, newspapers continue to be printed. It is the quality demands that have changed due to the expanding printing range. The process of papermaking is quite old and has been evolving over the years.

Papermaking is a long and complex process. It contains a number of different unit processes, which work through different mechanisms to produce the desired effects on the fibre suspension, and subsequently, on the fibrous web. It starts with the suspension of fibres and other raw materials in water, continues through the paper machine and finishing operations, and ends with the packaging of paper and boards. (Paulapuro, 2000)

Papermaking is a vast, multidisciplinary technology that has expanded tremendously in recent years. Significant advances have been made in all areas of papermaking, including raw materials, production technology, process control and end products. The complexity of the processes, the scale of operation and the production speeds leave little room for error or malfunction. Modern papermaking would not be possible without a proper command of a great variety of technologies, in particular, advanced process control and diagnostic methods. Not only has the technology progressed and new technology emerged, but our understanding of the fundamentals of unit processes, raw materials and product properties has also deepened considerably. The variations in the industry's heterogeneous raw materials, and the sophistication of pulping and papermaking processes require a profound understanding of the mechanisms involved. Paper and board products are complex in structure and contain many different components. The requirements placed on the way these products perform are wide, varied and often conflicting (Paulapuro, 2000). However, the paper industry must maintain its competitiveness through continuous product development to meet the ever-increasing demands on paper performance. It must be produced economically by environmentally friendly processes with a minimum use of the resources.

Gullichsen et al. (1999) note that paper mills are large capital-intensive units. The requirements for profitability and market acceptance constitute a constant business challenge for the paper industry and have been the driving force for developing the mill units. The technical solutions applied to meet requirements for mill performance have been developed through an evolutionary but on-going and dynamic process.

The use of simulators is essential, as modelling and simulation aids in the design of new facilities. Models can also be used to decide the size of equipment and how to optimise the configuration and operating conditions of various unit operations. For an existing operation, models can be used to optimise operating conditions for a particular unit of an entire flow system, to determine the cause of process upsets and to guide the engineer to the best remedial action, to evaluate the effect of changing process inputs ("what if" scenarios), and monitor an operation.

1.1 **Modelling and simulation**

Process modelling is not a new concept. Chemical processes have been analysed and modelled since the early days of chemical engineering. However, the development of the computer and its enhancements in both speed and capacity has significantly changed the complexity and the type of the process models. Early process modelling developed when computations were done by hand or with a hand-held calculator, involved relatively simple, steady state models that incorporated a large number of simplifying assumptions to make the problem tractable.

Process simulation, which is process modelling performed using a computer, uses a digital computer program to perform the necessary mass and energy balances to solve the model. The use of computer simulation for process modelling has greatly extended the size of the models that can be created and solved. Process models that relied on hand calculations were generally limited to include just a handful of unit operations. Computer simulation extends the size of the models to include hundreds if not thousands of operations. Likewise, the unit operations that make up the model can be more complex and can include dynamic components rather than simply steady state characteristics.

Process models and process simulations have several useful functions in the design and operation of manufacturing facilities. The most primitive type of software used is typically some type of programming language such as Pascal or Fortran. Technical computation software such as Matlab might also be used.

1.2 Modelling in the pulp and paper industry

A review of the last thirty years of modelling and process control literature reveals that many of the industry-specific problems have been worked on and solutions to some of the complex problems have emerged. As noticed by F.Kayihan (1996), the pulp and paper industry represents a big challenge. First, its main raw material is a natural product that retains its non-uniform characteristics and behaviour through the manufacturing processes all the way to the final consumer products. Secondly, it has to comply with the increasingly stringent environmental requirements and be responsive to customers demands for using environmentally friendly processes.

Over the years, the understanding of the wood processing behaviour through mathematical modelling has improved considerably. Due to their critical importance, major processes like digesters, recovery boilers, pulp and paper machines and evaporators continue to be modelled. At the same time, process control is improving due to the requirements of product quality. Pulp and paper processes are typically very similar to systems to which chemical engineers have been accustomed. Therefore, appropriate and successful control approaches are also similar in principle to some that have already been established through applications in petroleum and chemical industries. The challenge with the pulp and paper industries is due to the stochastic nature of the raw material, highly interactive multivariate process behaviour, long time delays and grade change transients.

In principle, process control issues of the stock preparation area are normally quite straightforward. However, in practice, they require a lot of attention, mainly due to process and loop interactions, sensor problems, changes in operating conditions and imperfect mixing tanks. From this point within the process to the end of sheet forming, total material balance control is crucial as the final product is highly sensitive to consistency and flow variations in fibre delivery.

1.3 Papermaking process

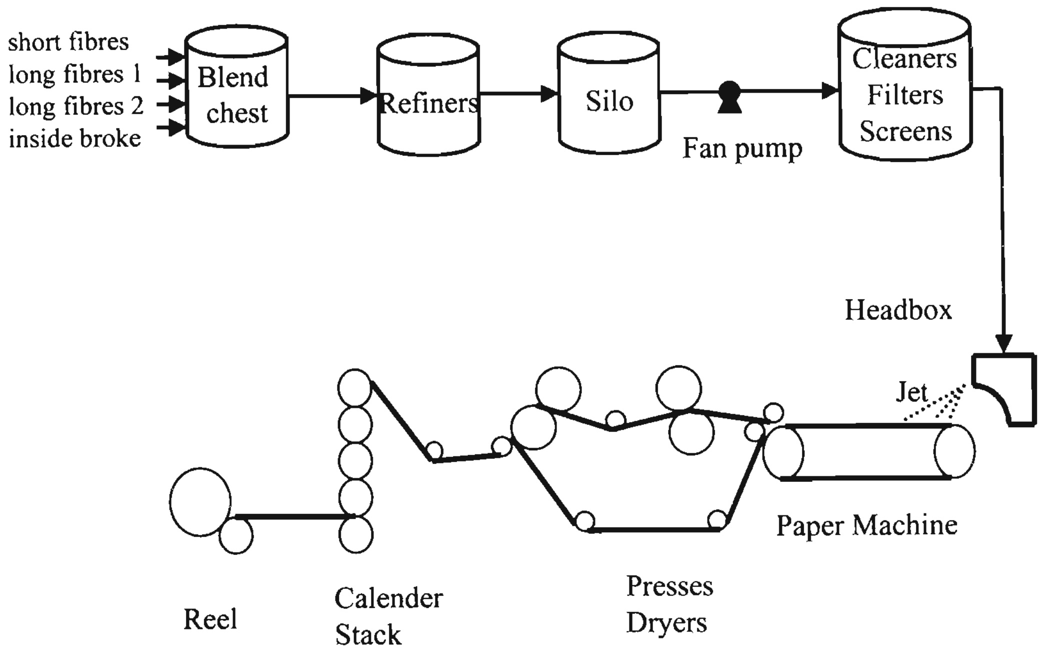

Figure 1.1: Papermaking process simplified diagram

This process starts at the blend chest (figure1.1), which stores the different feed flows of the system. The pulp is then treated (diluted), modified (refined) and finally mixed uniformly with additives to form the papermaking stock (silo). The fan pump delivers the stock to the machine over a wide range of paper machine production rates. The flow must be very steady and free from flow or pressure pulses or surges. By the time the pulp reaches the paper machine it should be relatively uniform. Therefore the pulp goes through a series of cleaners, screens and filters. The flow spreader (headbox) takes the incoming pipeline stock flow and distributes it evenly across the machine from back to front. The pressurized headbox discharges a uniform jet of papermaking stock onto the moving forming fabric. The endless, moving fourdrinier fabric (paper machine) forms the fibres into a continuous matted web whilst the fourdrinier table drains the water by suction forces. The sheet is conveyed through a series of roll presses where additional water is removed and the web structure is consolidated. Most of the remaining water is evaporated and fibre-to-fibre bonds are developed as the paper contacts a series of steam-heated cylinders in the dryer section. The sheet is calendered through a series of roll nips to reduce thickness and smooth the surface (calender stack). Finally, the dried, calendered sheet is accumulated by winding onto a reel.

1.4 Problems faced by the industry

Often in the papermaking industry, problems arise due to incorrect spreading of fibre by the headbox over the belt feeding the paper machine. Variations in the upstream chest level, in the supplied pressure, in the flow rate and fiber/water ratio delivered cause deviations at the headbox, and produce major losses of specification, which are understood to arise from sub optimal control in the preceding section. This preceding section, where flows are mixed and diluted, involves a number of control loops requiring different tuning for different paper grades.

Carrying out grade changes has to be considered carefully when designing the approach flow system to ensure stable operation over the entire product range. This applies to single-grade as well as to multi-grade machines. The multi-grade machines require quick responses upon changes, to gain a short transition period when changing to another grade. This leads to an improvement in production, time and efficiency. The approach flow also has to adapt without production disturbance. Stable operation of the system and all its components at all operation levels is required. Controllers have to be tuned accordingly.

The constant thick stock (stock made of a mix of the different pulps) feed flow to the paper machine has been a major concern since the development of continuous papermaking. Long-term thick stock concentration variations can be caused by large and sudden variations in the component flows to the blend chest, or by problems with consistency measurement and control caused by faulty measurements, poorly tuned controllers, or strong pressure variations in the dilution water header.

1.5 Objectives of this thesis

At Mondi Paper Ltd, Merebank, South of Durban, Paper Machine 2 (PM2) producing office paper has recently been transferred onto a Distributed Control System (DCS). This was seen as a good opportunity to enhance the control of the pulp feed to the machine, the main objective of this work. In order to understand and improve the controllability of this area, a detailed study of the stock preparation area dynamics and controls was carried out. This study resulted in the creation of several tools useful for a plant engineer:

- A dynamic model of the controlled processes involved in the stock preparation area.
- A static optimiser to deal with constraints problems observed.
- A simulator developed with the graphical user interface (GUI) of Matlab including both the model and its optimiser.
- A set of "what-if" scenarios to evaluate controller responses under process variation.

1.6 Dissertation layout

Chapter 2 gives a quick overview of the paper industry and details the most important units and principles necessary in the papermaking process. It highlights the process control problems encountered in the mill. Chapter 3 reviews advanced control techniques, some modelling principles, common optimisation techniques and gives a literature survey of dynamic simulations applied in the pulp and paper industry over the last decades. Chapter 4 details the approaches developed to create the dynamic model on Matlab. Chapter 5 is dedicated to the static optimiser, its parameters and its structure. Chapter 6 explains the different features to use the simulator created with the graphical user interface. Chapter 7 shows the results of this work, the improvements achieved and a discussion on the limits of the simulator. In conclusion, a summary of the work developed is presented as well as some recommendations for future work.

Chapter 2 : Paper industry and papermaking process

This chapter presents an overview of the paper origins and details papermaking process evolution over the years. It highlights the importance of the paper industry in our time and in our world.

2.1 <u>Paper industry</u>

2.1.1 **Paper origins**

In past civilizations, there has always been a need for various forms of communication, be it in oral or written form. The earliest man-made documents remaining emanate from prehistoric peoples, cave dwellers, who drew sketches on the wall of their caves. The system of communication by means of symbols and markings came from the need to find a way to convey thoughts and messages in a form not limited to time and space. It was not until the invention of paper that information could be recorded and passed efficiently.

The first substance identified as paper was the papyrus sheet, invented by the ancient Egyptians. Papyrus was a woven mat of reeds, pounded together into a hard, thin sheet. The origin of the papyrus manufacture is unknown but the oldest known roll was found in an Egyptian tomb of the first Dynasty about 3100-2900 BC. Papyrus was much used by the Greeks and later by the Romans whereas in the Western world, parchment, made from the skin of sheep and goat, had become the most important writing material.

2.1.2 **Papermaking origins**

2.1.2.1 Geographic evolution

Papermaking originated in China and was invented by Ts'ai Lun, in 105 AD. The art of papermaking first spread eastward over Korea to Japan where manufacturing started in 610 AD. By the ninth century, it was known throughout the Orient. In the battle between Chinese and Arabs in 751 AD in Samarkand, which was the western outpost of the Chinese Empire, Chinese craftsmen were taken prisoner by the Arabs, who carried the art of papermaking westward over Persia and Egypt. From there, papermaking spread along the southern shores of the Mediterranean following the conquest of Arabs. So it is not surprising to find the oldest paper mill in Europe in Spain in the town of Xativa near present day Valencia, founded in 1144. Paper was first manufactured in Europe in Spain. Then rapidly growing paper manufacture in Italy at the end of the 13th century supplied the whole Europe up until the middle of 14th century, when the first mills were set up in France.

2.1.2.2 Process evolution

The oldest method of making paper in China was with a floating mold. The principle involves pouring a fixed amount of fibre suspension onto a mold partially submerged in water. The fibre material in China was different from the materials to be used later in Europe. In the Far East, bast fibres of the mulberry tree, fibres of plants, and even old fishnets were used. This kind of raw material had to undergo tedious chemical pre-treatment before the fibres became suitable for papermaking. When the art of papermaking moved westward, the traditional raw materials were no longer available. The Arabs introduced rags in the manufacturing process and, in the early European papermaking, linen rags and hemp rope were the main raw material sources. A chemical pre-treatment was no longer necessary. Hand-papermaking was a slow and cumbersome process. When the paper demand rapidly increased in the eighteenth century papermakers in Continental Europe came up with the idea of developing a machine for forming the paper. Nicolas-Louis Robert (1761-1828) has gone down in history as the inventor of a continuously moving belt of cloth on which the fibre suspension was spread and the water was allowed to drain away, leaving an endless sheet of paper on the cloth. The developmental work was done at a mill in Essones, twenty kilometres south of Paris where Robert was employed in 1793. The first trials were performed as early as 1793, but it was 1798 before he came up with a construction on which a continuous web of paper could be made. The wooden machine was 260 cm long, and the width of the paper was 64 centimetres, which was incidentally the width of the wallpaper used in France at the time. The first machine to be ever built and successfully operated was in England in 1803. The Fourdrinier brothers took over development in 1804 and by 1807 had acquired all patent rights. The machine eventually became known as the Fourdrinier machine. The early paper machines consisted of a headbox adding paper stock to a moving wire supported between two rolls. The wet sheet was pressed once on the wire and then taken to a felt and run through another press nip before being accumulated on a roll for eventual drying in sheeted form.

2.1.3 **Paper industry in the digital age**

The question of paper's future in the digital age is best answered by the needs of consumers, but based on global demand outlooks, consumers still want paper and will still want paper well into the 21st century. World demand for paper has doubled in the past 20 years and is forecast to double again by the year 2010. In an age of notebooks computers and websites, it's easy to overlook some of the enduring qualities of paper. It is portable, disposable, recyclable, economical and versatile.

In developing countries, population increases and a rising standard of living will result in a tremendous demand for paper, packaging and fibre. In developed countries, the continuing need for 'hard copy' suggests a prominent complementary role for paper even as the information highway develops into the next century. At the same time, shifts can be expected in paper consumption and paper products. For example, the demand for varieties of printer paper and value-added papers will certainly increase while the demand for cheaper, lower grades may decrease.

The paper industry produces a variety of grades for industrial and consumer purposes. Paper and board grades can be divided into the following main groups: printing and writing paper, packaging grades and tissue grades. There are more than 5000 products made from papermaking by-products.

2.1.4 **Mondi and South Africa in the paper industry**

Mondi Ltd, an Anglo American group company, is a large South African producer of pulp, paper, packaging board, sawn-timber and related products. It owns or leases 526000 hectares of forest land and has an annual turnover of approximately US$ 2 billion. Since the first reel of newsprint rolled out of Merebank mill in 1971, Mondi has grown into a major supplier of paper products to South Africa and the world. The four members of the Paper Manufacturers Association of South Africa (PAMSA) – Sappi, Mondi, Nampak, and Kimberly-Clark contribute to almost 98% of the national pulp and paper production with a total of 17 pulp, paper and board mills across the country. The pulp and paper industry is a true global competitor, cost effectively producing paper and board products of world-class quality. It has transformed South Africa from a net importer in the early eighties to a net exporter, R 3.4 billion in 1997 alone. While world production of paper and board has grown by an average annual rate of 3.0% since 1970, the South African industry has outstripped these figures, returning an annual average growth rate of 5.2%. South Africa is now rated 12^{th} largest producer of pulp and 24^{th} largest producer of paper and board products in the world. South Africa is a net exporter of paper and board. Of the total production in 1997, 25% was exported. The requirements for profitability and market acceptance constitute a constant business challenge for the paper industry and have been the driving force for developing the mill units. The technical solutions applied to meet requirements for mill performance have been developed through an evolutionary but on going and dynamic process. Various mill concepts, processes, and equipment have been developed for various paper grades.

2.2 **Papermaking process**

2.2.1 **Wood pulp**

The wood can be divided into two categories: hardwood and softwood species. Hardwood trees such as birch, beech, poplar, oak or ash have wood with very short fibres. Paper made from these species is weaker than that made from softwoods, but the surface is smoother, and therefore better to write and print on. Softwood trees such as fir, spruce, pine or cedar have wood with long fibres, and paper made from this type of wood is much stronger, it is ideal for making products like shipping containers that requires superior strength but the finish is rougher and then not ideal for printing. Most of the paper seen today is made from hardwoods and softwoods, a special blend of those two types for the different purposes of the paper. For instance, 'newsprint' paper is made to be opaque, 'grocery bag' paper to be strong, 'tissue' to be soft, 'fine writing' paper to be smooth.

Wood is made of four major components: cellulose, hemicellulose, lignins and extractives. The major structural component is cellulose; its linear polymers are bound together with hydrogen bonds. Hemicelluloses are similar to cellulose only in that they are polymers of sugar monomers, they are short chained polymers often branched. Their function is poorly understood. Lignins are aromatic compounds, which bind wood cells together. The prime purpose of chemical pulping is to remove lignins from the wood, thereby liberating the cellulose fibres from the wood. The final groups of chemicals in wood are called extractives. These are the volatile components of wood, which are normally removed by the time the pulp is ready for papermaking. Paper derives its strength from hydrogen bonds formed when cellulose fibres or fibrils come into contact with each other.

2.2.2 **Papermaking process**

The papermaking process starts in the forests, where seedlings, specially cultivated in a nursery, are planted out to grow into trees destined for the pulp industry. Hardwoods and softwoods are the main sources of timber. Once the trees are fully-grown, they are felled in the forests, cut into logs and transported to the mill where the logs are stored in the woodyard before being sent to the debarker, a large rotating drum, where the bark is removed. Bark has little fibre value and causes dirty pulp. The first step in making paper is to turn the wood into pulp, the breaking down of the wood into fibres being called the pulping process. Most papers are made from a blend of both types hardwood (short fibres) and softwood (long fibres) to achieve the desired properties. Different processes are used to produce different types of pulp, and these different types of pulp make up the blends for the various grades of paper. The different types of pulp derive their names from the process used to produce them. These are mechanical pulps and chemical pulps. A third process is the repulping of de-inked paper (newspapers and magazines), which recovers fibre (RCF). Those processes are detailed in appendix A.

The next stage in the process is the stock preparation. This involves the blending of the different types of pulp, further refining if required, and the addition of various non-fibrous additives. Different types of pulp are used in various combinations to produce the many different kinds of paper used by printers. The different pulps are mixed with additives such as caustic soda to adjust the pH level, alum to disperse pitch and calcium carbonate (for fine paper) or clay (for gravure paper) in a vessel called a blending chest. The combination of stock and additives, which go to make up a particular type of paper, is called the furnish. The stock is then diluted to about 99% water and less than 1% wood fibres and is now ready to be made into paper. The stock is pumped to the paper machine where the watery mixture is turned into paper. A more detailed description of stock preparation is given in section 2.3.

The flow spreader takes the incoming pipeline stock flow and distributes it evenly through the headbox across the width of the paper machine. The pressurised headbox discharges a uniform jet of papermaking stock onto the moving forming fabric, an endless belt called the wire. The mixture then enters the wet end operations (figure 2.1).

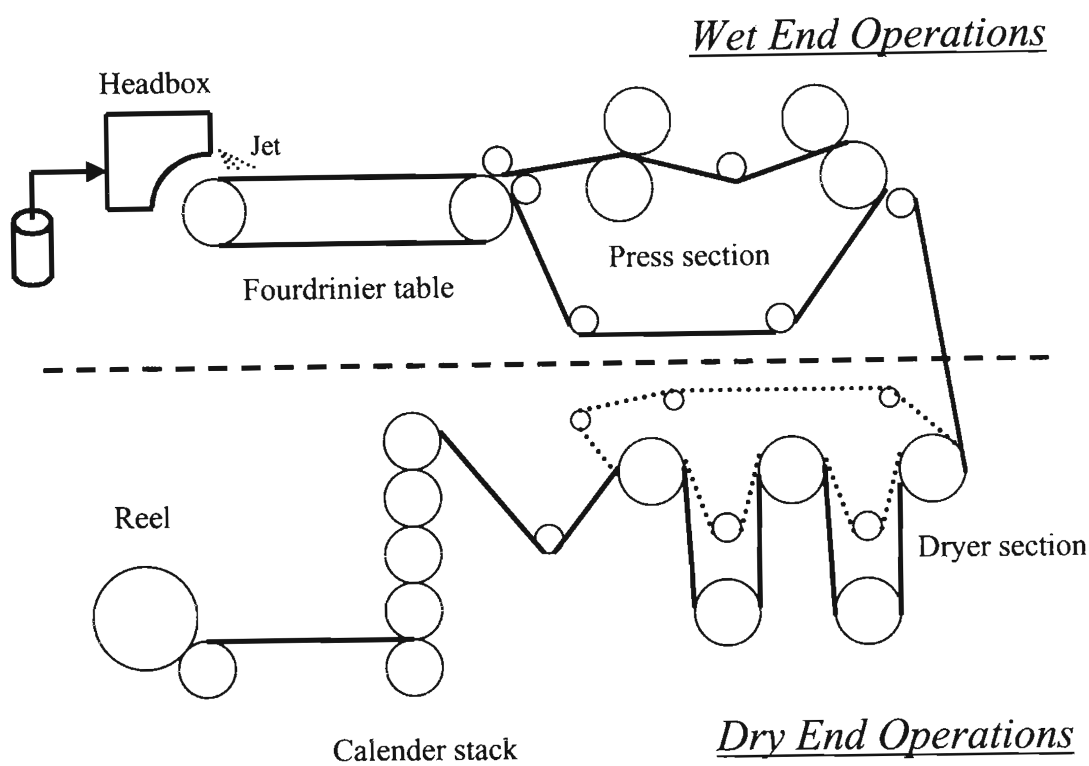

Figure 2.1: Dry and wet end operations

Originally the wire was a finely woven mesh of phosphor bronze, but nowadays, synthetic fabrics are used. The speed of movement of the wire causes a low-pressure effect, which sucks some of the water through and begins the de-watering process. As highlighted by Shinskey (1996), sheet forming can be considered as a composite of three basic hydrodynamic processes, which are drainage, oriented shear, and turbulence. The most important effect of the drainage process is the dewatering of the fibre suspensions to form the mat. When the fibres are free to move independently of one another, drainage proceeds by the mechanism of filtration, and the fibres are deposited in discrete layers. Filtration is the dominant mechanism in most fourdrinier forming applications, as shown by the layered structure and relatively uniform formation of the sheets. When the fibres in suspension are immobilised, they floc together in coherent networks; drainage then occurs by thickening, and a more felted and "floccy" sheet structure results. Care must be taken with floc formation as they introduce discontinuities in the sheet properties. Papermaking suspensions spontaneously form networks during drainage unless sufficient dilution is used or supplemental mixing energy is provided. Dilution is a powerful mechanism for dispersion; but the level required to adequately control flocking on paper machines is not economically feasible. Additional dispersion must be generated during drainage by the turbulence- inducing effects of drainage elements below the forming fabric, or by shear-inducing devices above the fabric. In each design of commercial sheet-forming machines, the three elementary forming effects of dilution, turbulence and oriented shear are applied to different degrees in an attempt to optimise sheet quality. More water is removed by suction boxes underneath the wire and a suction roll at the end. The process described is called the flat fourdrinier but most paper machines are actually twin-wire machines. These machines use two forming wires with the stock continuously squirted between the wires, the water being removed in both directions through the wires. This shortens the distance required to de-water the stock and eliminates one sidedness of the paper. (The wire side of the paper tends to be rougher). By now the stock has become a weak, wet sheet, which transfers onto a moving felt at the start of the press section.

In the press section of the machine, the wet sheet lying on a moving felt passes through pairs of heavy rollers. Here the sheet is pressed, squeezing out more of the water through the felt. After pressing, so much of the water has been removed it is no longer possible to remove the remaining water by pressure alone. Now strong enough to sustain its own weight, the fast moving sheet detaches from the felt and enters the drying section of the paper machine. Finally the paper reaches the presses, which have the main functions of removing water from the sheet, consolidating the sheet, imparting favourable properties to the sheet and promoting higher wet strength of the sheet to improve runability in the dryers. The paper from the press sections contains 55 to 60% moisture and needs to be dried.

The last section is called the dry end operations (figure 2.1). In the drying section, the paper travels through a long series of rotating, steam heated rollers, where nearly all of the remaining water is removed through evaporation and where fibre-to-fibre bonds are developed. The drying process consists of passing the paper over rotating steam heated cylinders so that the heat is transferred to the paper evaporating the water, then the evaporated water is carried away by ventilation air. The wet web is held against the drying cylinder by a synthetic permeable fabric called a dryer felt or screen. The screen not only improves the heat transfer to the paper but also reduces shrinkage of the paper and helps prevent sheet cockling. The paper is now ready for smoothing.

The calendering is done by passing the paper between heavy, polished steel rollers, called calender rolls. Its purpose is to obtain a smooth surface for printing and to improve the cross direction uniformity in certain properties such as thickness (caliper), density or smoothness. The pressure on the paper compresses it, giving it a flat smooth surface. This process of ironing of the paper is called calendering. Certain types of paper are also supercalendered to give a glossy finish such as that required for magazine grade paper; this gravure paper has clay added to the furnish which, when supercalendered, produces a gloss.

From the machine calender the paper is wound onto a rubber covered spool (reeling section). A full spool of paper is called a jumbo reel and weighs 15-20 tons depending on the diameter and type of paper. When a jumbo reel has reached the required diameter a new spool is lowered onto the running sheet of paper. The sheet is torn across its width and transferred onto the new spool. This is done without stopping or slowing down the machine.

The full jumbo reel is then moved to the rewinder where it is cut into narrower width rolls by circular slitter knives, again depending on customer requirements, and wound onto smaller cardboard cores prior to wrapping.

Various grades and types of paper and board may receive a surface coat in the coating section. Its main purpose is to either improve the characteristics of the paper, or to give it special properties, such as a barrier to water or grease. These coats may be applied during the papermaking process itself (on line) or after the sheet has been cut down to specific sized reels (off line), and could include waxes, lacquer, resins and adhesives, but most often clay. A particular type of coating is used to produce carbonless paper, used typically for credit card vouchers, where one or more layers of paper are coated with microcapsules of ink and a layer of clay; the capsules burst under pressure to give a copy of the original.

Most of the basic papermaking principles apply to making board as well. Board can comprise one or several layers of product, depending on its usage. The two most common types produced in South Africa are linerboard (with at least two layers or plies, the top layer usually being of a better quality) and cartonboard (folding box board).

The reels of paper are then wrapped for protection against damage and dirt, weighed and labelled. If the customer requires his paper in sheets rather than in reels; the reels are transferred to high-speed cutters where the sheets are cut to size, counted and inspected for quality. The sheets are wrapped, labelled and boxed in the customer's own packaging if required.

2.2.3 **Quality management**

While being formed, the paper is submitted to a series of profile controls. Scanners are used on paper machines to measure several properties such as moisture, basis weight, calliper, brightness, and dirt count. The variations observed in a sheet of paper can be of three types (figure 2.2): the Machine Direction variations (MD), the Cross Direction variations (CD) and the random variations, which are a composite of the two preceding types of variations.

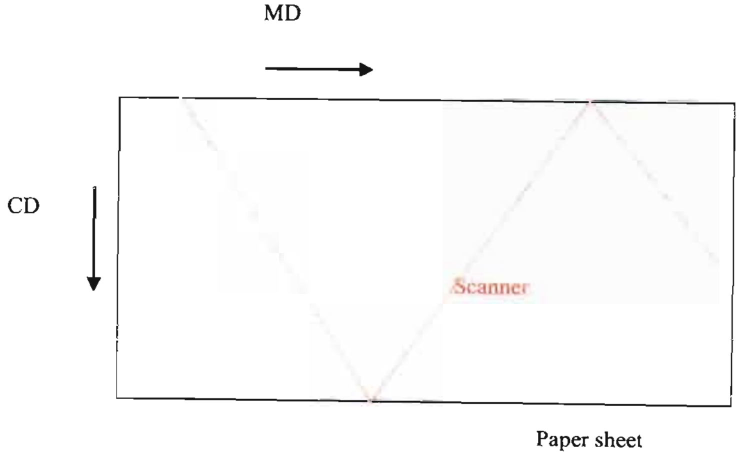

Figure 2.2: Three variation types

MD variations are due to variations in the supply of stock in the headbox. CD variations can be due to several factors such as hydrodynamics on the fourdrinier, the slice opening on the headbox, cleanliness of press and dryer felts, press roll crowning and ventilation in the dryer section. The most commonly controlled profile characteristics are caliper, moisture and basis weight. The scanner actually measures a diagonal profile due to the movement of the paper.

Quality control testing is carried out at all stages of the papermaking process to ensure that customers recieve a high quality paper. A range of tests is conducted to ensure that the paper meets specifications; some testing is carried out in laboratories adjacent to the process and other monitoring is carried out on-line by sophisticated instrumentation. These tests include brightness, shade, burst, tear, caliper and basis mass. Basis mass (also called grammage of weight) is the common unit of measurement of paper and is the mass per unit area of paper - specifically in grams per square metre (g m^{-2}). A typical photocopy paper is 80 g m^{-2}.

Different properties are important for different paper grades. They can be classified as follows.

Two-sideness: The two surfaces of a paper vary due to the formation of the paper on a forming wire. The topside of the sheet is smoother and contains more fines. This characteristic of paper affects the optical properties and the smoothness of the sheet.

Directionality: The general orientation of the fibres in a sheet of paper tends to be in the direction the paper machine is running- machine direction. This affects the properties such as tensile and tear strengths and folding endurance.

Mechanical and strength properties: The grammage of a sheet is simply the mass in grams of the paper per square meter of the paper. The caliper is the thickness of the sheet. Bulk and density can be calculated from the grammage and the caliper. Tearing strength measures the fibre strength within a sheet of paper. Stiffness is the force required to bend a strip through a given angle. Bursting strength is determined by clamping a paper sample over a rubber diaphragm through which pressure is applied at a gradually increasing rate until the paper ruptures. The pressure at the rupture point is the bursting strength. Folding endurance is the number of folds a sample of paper can endure when folded under a specific tension through a specific angle. Tensile strength is the force required to break a narrow strip of paper where both the dimensions of the paper and the loading rate are specified. The amount of stretch at rupture is determined at the same time. The edgewise compression strength test was developed for corrugating papers where the compression strength on the walls of the containers is important.

Surface properties: Smoothness (or roughness) is usually measured by the amount of air leaking underneath a metal annulus in contact with the paper surface.

Optical properties: The brightness, opacity, and colour of paper are determined by taking reflectance readings at the appropriate wavelengths of light.

2.3 Stock preparation area and approach flow

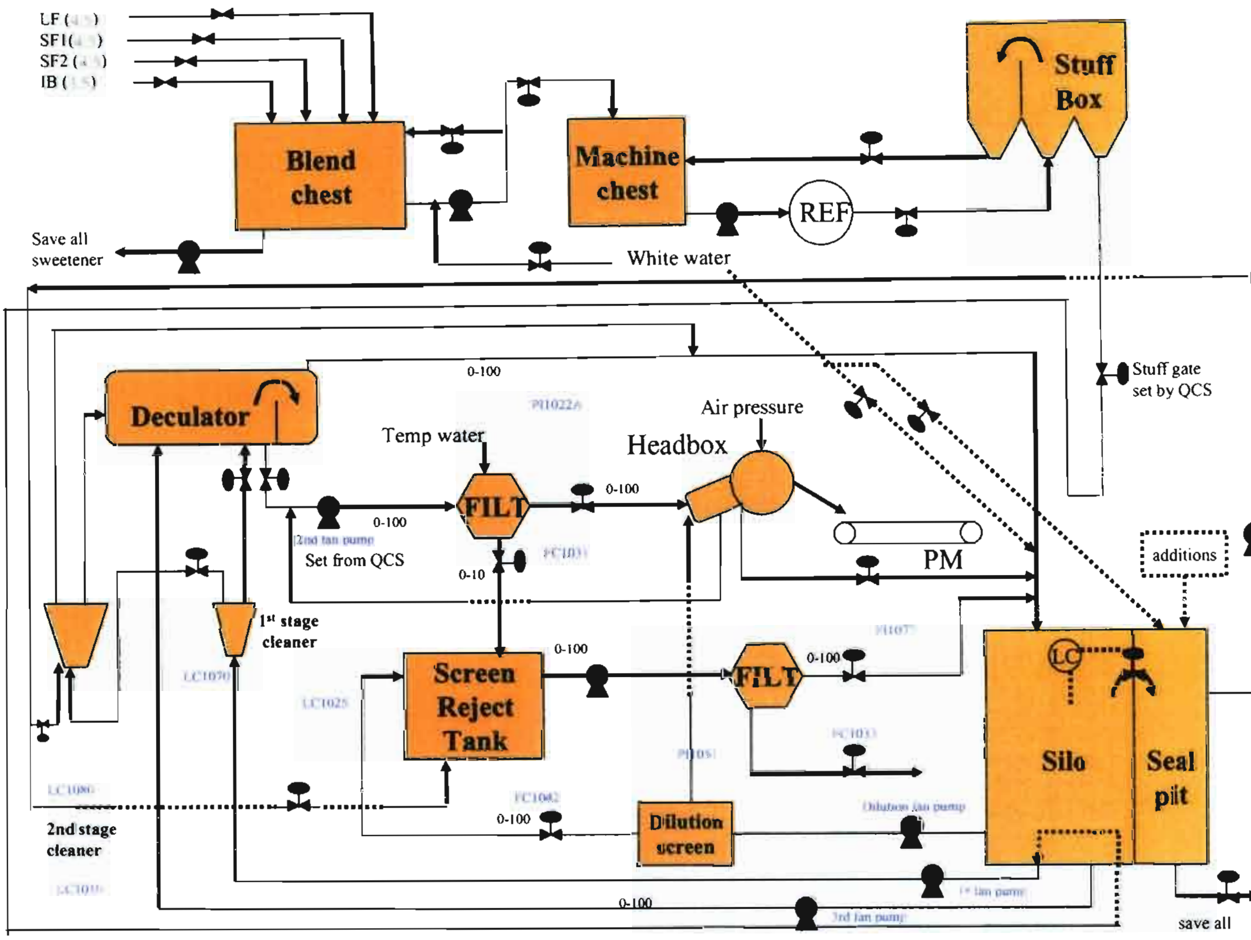

Figure 2.3: Stock preparation area

The stock preparation area (figure 2.3) starts at the blend chest and ends at the machine chest. Its objective is to take fibrous material (pulp) and non-fibrous components (additives) and treat and modify them, then, combine the ingredients continuously and uniformly into papermaking stock. This process consists of three major operations:

- *Pulp supply and dilution* where the pulp must be pumped and diluted from head storage towers.

- *Refining* is the mechanical treatment of pulp fibres to develop their optimum papermaking properties. Refining increases the strength of fibre to fibre bonds by increasing surface area for bonds and by making fibres more pliable to conform around each other (also collapsing the fibres into ribbons)

- *Stock blending and mixing*. Any change in the blend of pulps and chemicals will affect the paper produced. So this operation must ensure an accurate blend of the furnished constituents and an even consistency at the exit. Pulp from the storage chest is brought forward to the paper machine through a series of controlled dilution steps. The consistency and the flow rate of each pulp stream must be accurately measured. The function of the blend chest is to mix the components before they are sent forward to the machine chest, which ensures a good mixing uniformity.

The approach flow of the paper machine refers to the operations that transfer pulp from the machine chest to the paper machine. The central unit of the approach flow system is the fan pump – this is normally the largest pump on the machine or in the mill. The fan pump must deliver the stock to the machine over a wide range of machine production rates. The flow must be very steady and free from flow or pressure pulses or surges. By the time the pulp reaches the paper machine it should be relatively clean. The cleaners and screens provide additional insurance. To ensure a uniform feed of thick stock into the system, a constant head tank called a stuff box is sometimes used. The flow of pulp is controlled through a valve called a basis weight valve. Then the pulp enters the headbox, the main unit of the papermaking process. Figure 2.4 represents the cross section of an air-cushioned headbox.

The purpose of the headbox is to spread stock evenly across the width of the machine (cross directional uniformity), to even out cross currents and consistency variations, to even out machine direction velocity (flow) changes, to create controlled turbulence to break up fibres flocs and to discharge evenly from the slice opening and impinge on the forming fabric at the correct position and correct angle. The headbox slice is a full-width orifice or nozzle with a completely adjustable opening in small segments to give the desired rate of flow. The slice geometry and opening determine the thickness of the slice jet, while the headbox pressure determines the velocity. Its main operating variables are the stock temperature and consistency. The consistency must be low enough for good formation but not too low for poor first pass retention of fibres or to hydraulically overload the wire. The mass of the pulp fed forward is controlled using the basis weight valve. Finally the jet-to-wire ratio is usually kept near 1. Indeed, if it is lower than 1, the stock is dragged; if it is above, the stock is rushed. The jet-to-wire ratio is controlled by the pressure in the headbox and affects the fibre orientation in the paper.

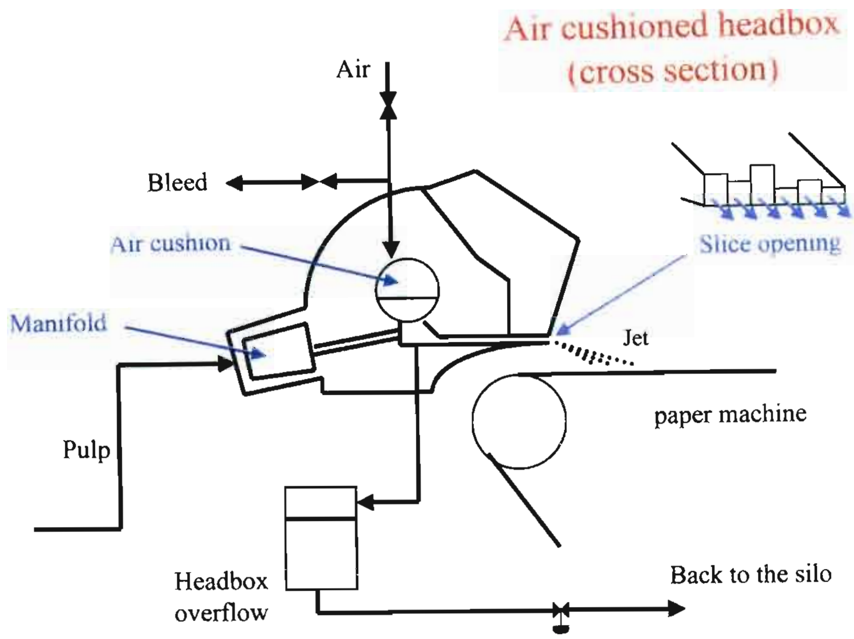

Figure 2.4: Air-cushioned headbox

2.4 PM2 Paper Machine

The PM2 (Paper Machine 2) at Mondi, Merebank, produces fine paper grades including various bond and white copier grades, tablets and white kraft. Traditionally fine papers are grades of paper which do not contain any mechanical pulp. However, these days, small amounts (10%-15%) of groundwood pulp (GWD) or thermo-mechanical pulp (TMP) are allowed in certain grades. The furnish used for the PM2 product range is normally 100% bleached chemical pulp, long and short fibre, except in the case of the white kraft grades which use a furnish of 100% long fibre bleached chemical pulp to achieve the required strength. The furnish of other grades (i.e. tablet) contains small amounts (10%-15%) of mechanical pulp. Most grades produced on PM2 contain small amounts of China clay as a filler. The average machine speed is 559 m/minute.

Paper Machine 2 has recently been transferred onto a Distributed Control System (DCS). This was seen as a good opportunity to enhance the control of the pulp feed to the machine. The problems of maintaining steady conditions at the headbox are well known, and are understood to arise from sub-optimal control in the preceding section involving the blend chest and machine chest, amongst other items, where several pulp streams and dilution water are combined. A number of control loops are involved, but appear to require different tuning for different paper grades. Often individual loops are taken off-line.

Chapter 3 : Literature review

The stock preparation area involves a number of control loops, which may or may not be independent. The losses of specifications observed on the paper machine are understood to be due to interaction between those control loops. The controllers are classic Proportional/Integral (PI) controllers and some of them interact through a cascade structure. A cascade control structure has two feedback controllers with the output of the primary (or master) controller changing the setpoint of the secondary (or slave) controller. The output of the secondary goes to the valve. There are two purposes for cascade control: to eliminate the effects of some disturbances and to improve the dynamic performance of the control loop. Background has been obtained from a search of the literature of advanced control and advanced tuning techniques. This first part of the literature review deals with the methods developed to understand interactions, and with the types of controllers available to overcome them.

3.1 Process control

3.1.1 Interactions

The vast majority of automatic controllers used by industry are of the PID type. As observed by Astrom and Hagglund (1995), the Japan Electric Measuring Instrument Manufacturer's Association conducted a survey of the state of process control systems in 1989 and it appeared that more than 90% of the control loops were of the proportional-integral-derivative (PID) type. Despite the fact that the use of PID control is well established in process industries, many control loops are still found to perform poorly.

Interaction between controllers is a common problem in industry. In the specific issue of the stock preparation area, bad tuning of the controllers can be one cause and neglected interactions between controllers another one. The study of interactions in-between controllers starts with the study of the stability of the process. Being given a process, one of the primary concerns is to find out if the entire multivariable system is closedloop stable and then to know how stable it is. Another question concerns the robustness of the controller, i.e. the tolerance of the controller to changes in parameters. A fairly useful stability analysis method is the Niederlinski index

(Niederlinski, 1971), which can be used to eliminate unworkable pairings of variables at an early stage in the design. The settings of the controllers do not have to be known, but it applies only when integral action is used in all loops. It uses only the steadystate gains of the process transfer function matrix. The method is a necessary but not sufficient condition for stability of a closedloop system with integral action. Then some processes are easier to control than others. Some choices of manipulated and controlled variables produce systems that are easier to control than others. This inherent property of ease of controllability is called resiliency (Luyben, 1990). Morari (1983) developed a very useful measure of this property. The Morari resilience index (MRI) gives an indication of the inherent controllability of a process. It depends on the steady state gains from inputs to outputs at a chosen frequency. It is a useful tool for estimation of how easily a multivariable system might be controlled.

3.1.2 **Multivariable control**

A multivariable process has several inputs and outputs (figure 3.1). The interactions between variables are often strong. This means that changes in inputs (manipulated variables) influence several output variables. The strength of these interactions determines how well conventional multiple-loop feedback control strategy controls the multivariable process. Multivariable control requires the replacement of conventional compensation technology by computerised algorithms. The aim is to compensate the effect of process interactions and make the entire process behave optimally.

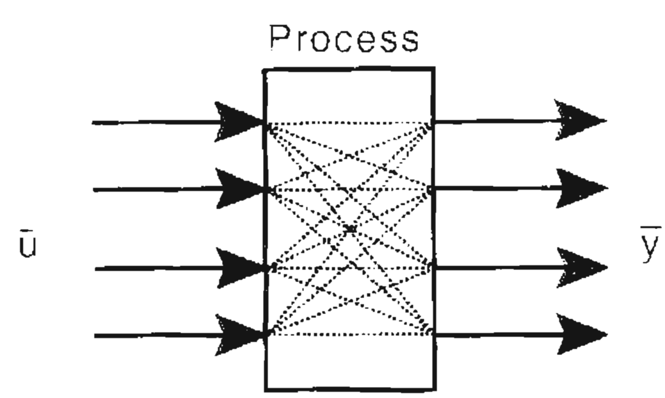

Figure 3.1: Multivariable process (Leiviska, 1999)

Many possibilities are available to handle multivariable systems. One is to use analog compensators that try to consider interactions. The computer control uses the upper level supervisory actions based on process models for the same purpose. This requires process models and the conversion of process knowledge to the actual control strategy. Multivariable control methods are common but practical applications seem rare.

The simplest way to handle multivariable processes is to ignore the interactions and design simple, separate control loops (figure 3.2).

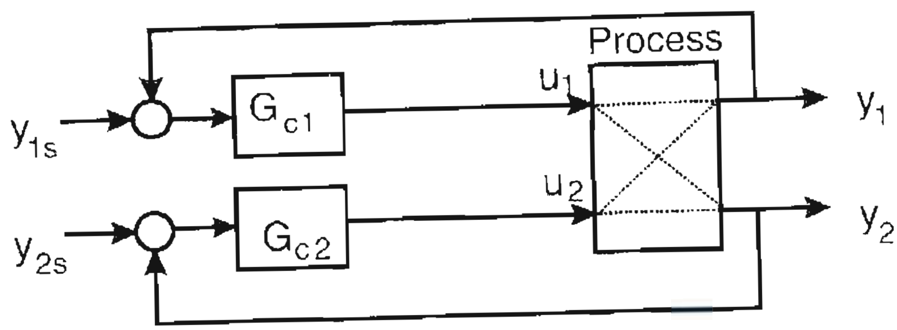

Figure 3.2: Diagonal control strategy in multivariable system (Leiviska, 1999)

This approach works sufficiently for most cases. If severe interactions exist, an oscillating or even unstable system can result. Careful selection of input and output pairs and realistic tuning of control loops improves this situation. Multivariable control is necessary in cases where separate control loops influence each other. This results in decreased control performance of the strategy when interacting control loops respond to setpoint changes or load disturbances. To assure stability, certain loops may have to be "detuned" to a slack control. This decreases their performance.

Another way of handling multivariable processes is by using decouplers. Various types of decouplers were explored to separate the loops. Rosenbrock (1974) presented the inverse Nyquist array (INA) to quantify the amount of interaction. Bristol (1966), Shinskey (1979, 1981), and McAvoy (1983) developed the relative gain array (RGA) as an index of loop interaction. As highlighted by Luyben (1990), all of this work was based on the premise that interaction was undesirable. This is true for setpoint disturbances. One would like to be able to change a setpoint in one loop without affecting the other loops. And if the loops do not interact, each individual loop can be tuned by itself and the whole system should be stable if each individual loop is stable. Unfortunately much of this interaction analysis work has clouded the issue of how to design an effective control system for a multivariable process. In most process control applications the problem is not setpoint responses but load responses. Systems must hold the process at the desired setpoints in the face of load disturbances. Interaction is therefore not necessarily bad, and in fact in some systems it helps in rejecting the effects of load disturbances.

3.1.2.1 Relative gain array (RGA)

The RGA was proposed by Bristol (1966) and has been extensively applied by many workers. It has the advantage of being easy to calculate and only requires steadystate gain information. In the RGA, columns represent possible controller output (manipulated variable MV) choices, and rows the controlled variables (CV). The elements in the RGA, each representing the associated pairing, can be numbers that vary from very large negative values to very large positive values. The closer the number is to 1, the less difference closing the loop makes on the other loops being considered. The problem with pairing on the basis of avoiding interaction is that interaction is not necessarily a bad thing. Therefore, the use of the RGA to decide how to pair variables is not always an effective tool for process control applications.

3.1.2.2 Inverse Nyquist array (INA)

Rosenbrock (1974) was one of the early workers in the area of multivariable control. He proposed the use of INA plots to indicate the amount of interaction among the loops. Usually the INA is a very conservative measure of stability. The INA method strives for the elimination of interaction among the loops and therefore has limited usefulness in process control where load rejection is the most important question.

3.1.2.3 Compensator

Another way of decoupling is to add a compensator into a diagonal control structure (figure 3.3). The design of this compensator starts by using the situation where the selection of input and output pairs is as good as possible. The compensator and the process should form a system that is as diagonal as possible. Then the control using separate controllers is possible. The compensator coordinates the variation of the actual manipulated variables. Computerised design methods for these compensators exist.

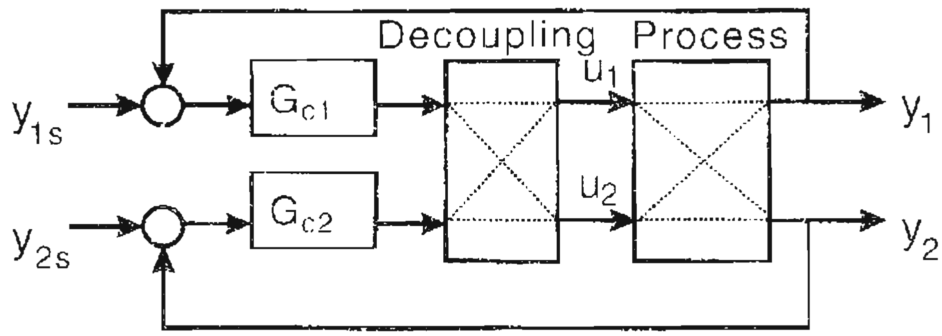

Figure 3.3: Decoupling controller for 2*2 process (Leiviska, 1999)

3.1.3 Adaptive control

Another approach in controlling multi-loop processes is by using adaptive controllers. In adaptive control, problems can be of two categories depending on the way controller parameters are evaluated. There are problems where process changes can be deduced from process measurements, and consequently parameters are adjusted in a predefined manner. This approach is referred to as gain scheduling, and has been tackled by several teams of researchers over the past years. (Andreiev, 1981; Astrom, 1983; Wong and Seborg, 1986; and Shinskey, 1996). There are problems where process changes cannot be anticipated or measured. For this particular family of problems, the most popular methods are the Model Reference Adaptive Control (MRAC), based on the global stability theory and the Self-Tuning Adaptive Control (STAC), based on the use of a quadratic cost function. (Kalman, 1958; Peterka and Astrom, 1973; Ydstie et al., 1985; Clarke et al., 1987 ; Camacho and Bordons, 1995)

Recent research showed that even though MRAC and STAC were developed from different viewpoints, the techniques are closely interrelated, and can even be analysed from a unified theoretical framework. However, whilst significant applications of MRAC are still scarce in industry (the global stability is a non trivial problem), the STAC approach has received more attention than any other adaptive control strategy over the past decades.

The first development of STAC arose with Peterka and Astrom's work (1973). They developed the predictor-based self-tuning control, by deriving it directly from Kalman's original idea (1958) of a self-optimising control system. This control system, presented for the first time at a conference on parameter estimation, minimizes the expected value of a quadratic objective function, for the most recent predicted values, over a given horizon. The original method of self-tuning control was further developed by Ydstie et al. (1985). Their method enabled the regulator to cope more easily with the features of industrial control in the chemical industry. The most significant modifications to Peterka and Astrom's work concerns the use of a variable forgetting factor for the handling of past data, the replacement of the minimum variance control objective by an extended horizon controller and the expression of the algorithm in an incremental form to eliminate the offset. The use of an extended horizon gives more time to the controller and therefore allows it to look beyond process time delays, which was the limiting factor of the previous approach. Despite all of these improvements, it appeared that this method was still not applicable to suit a majority of existing chemical processes.

Clarke et al. (1987) are the first researchers who designed a robust regulator capable of controlling either a simple plant (e.g. openloop stable), or a more complex one, such as open loop unstable and having variable dead time. Their method, known as Generalized Predictive Control (GPC), presented in two separate papers (one for the algorithm, one for the extensions and interpretations of the method) is still one of the most popular in the adaptive control family, both in academic and industrial worlds. As summarized by Camacho and Bordons (1995), the basic idea of GPC is to calculate a sequence of future control signals by minimizing a multistage cost function defined over a prediction horizon. The function to be minimized is the expectation of a quadratic function measuring the distance between the predicted system output and some predicted reference sequence over the horizon, plus a quadratic function measuring the control effort.

An adaptive control system changes according to changes in the environment and plant, i.e. adapts itself so as to maintain satisfactory control, which is usually judged by some performance index. To be of use, the controller must adapt as rapidly as possible in comparison to the parameter changes of the plant or environment so that it is always at, or nearly at, its most suitable performance. Various degrees of adaptation may fall within the rather ill-defined area of adaptive control, from feedback controllers supplemented by feedforward elements to full optimisation procedures. In the model reference control system, the control signal is generated according to the difference between the output of a model and the output of the real plant. An adaptive controller can change its tuning or structure when necessary. This facilitates re-tuning and guarantees the optimum performance of the controller when the behaviour of the process is changing. Adaptive actions can occur continuously or batch-wise. Adaptive control is necessary when performance of the conventional fixed parameter feedback controller is insufficient. This is due to a non-linear process or actuator or to the simple fact that the behaviour of the process is time dependent or unknown. In addition, dead time processes and processes with strong interactions can require adaptive features from controllers. Three main approaches of adaptive control exist: gain scheduling, model reference, and self-tuning control.

3.1.3.1 Gain scheduling

In practice, a single process variable such as the production rate usually explains most changes in process dynamics. Production rate changes also act on the time constants and dead time of processes. In such cases, tabulation of controller parameters can be a function of production rate or any corresponding variable. When changes occur, these tables are useful as a source of controller tuning parameters.

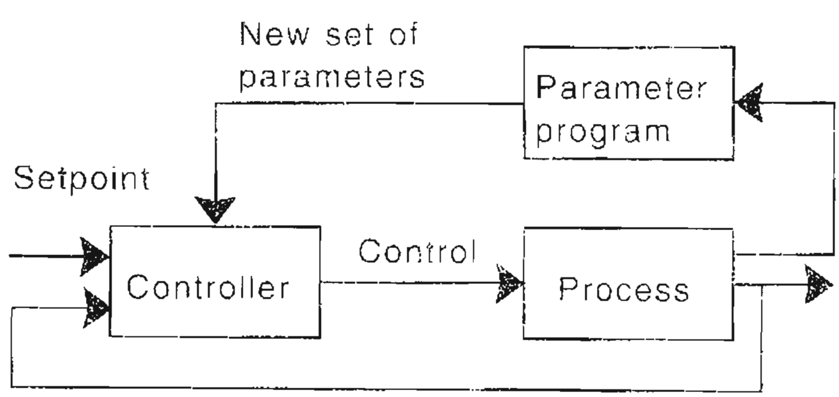

Figure 3.4: Gain scheduling structure (Leiviska, 1999)

When it is possible to find measurable variables that correlate well with changes in the process dynamics, a gain scheduling control strategy is possible. A block diagram of this scheme is presented figure 3.4. Such variables can be used to change the controller parameters in a predetermined manner, by monitoring the operating conditions. The most common approach is to store in a table several sets of process parameters (parameter program), for different plant conditions and choose amongst these sets the most adequate settings as operating conditions vary. As no parameter estimation occurs for this particular approach, the controller adapts quickly to changing process conditions, which constitutes the main advantage of this method. However, as highlighted by Astrom (1983), a good knowledge of the process is required to apply it, which constitutes one of the main drawbacks.

3.1.3.2 Model reference adaptive control (MRAC)

Historically, the first attempt to design stable adaptive control systems appeared with the development of the MRAC approach. In MRAC, the basic objective is to force asymptotically the output of a process towards a given reference. The key technical problem of this family of applications is to determine the structure of the parameter adjustment mechanism so that the overall system is globally stable. In this particular context, the stability criterion is equivalent to having process input and output remaining bounded for all time-steps, with the observed error tending to be equal to zero when time goes to infinity.

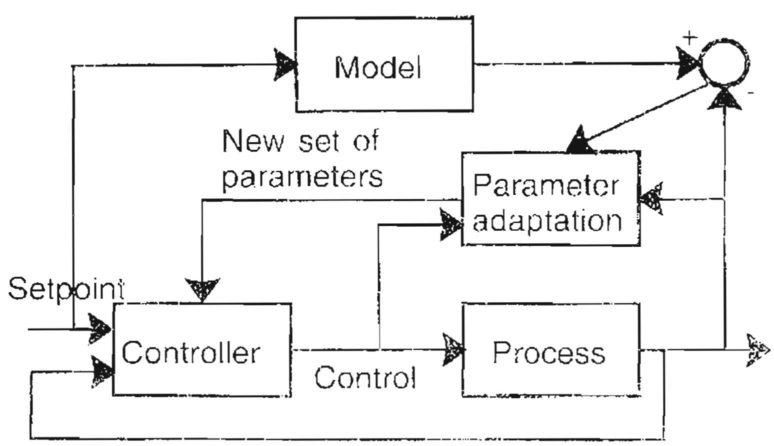

Figure 3.5: Model reference control block diagram (Leiviska, 1999)

Model reference control (figure 3.5) uses a model that describes the ideal behaviour of the system when the setpoint is changing. The tuning mechanism then forces the response of the real system to follow the dynamics of the reference model by changing the controller tuning. The control loop consists of two separate loops. One is the conventional feedback loop, and the other includes the controller tuning mechanism. Astrom (1983) says that the key problem is to determine the adjustment mechanism so that a stable system, which brings the error to zero, is obtained. The biggest problem in constructing model reference controllers is finding the tuning mechanism that guarantees the controller stability and simultaneously minimizes the control error. Also, lack in model accuracy can cause problems with controller and tuning stability.

3.1.3.3 Self-tuning adaptive control (STAC)

STAC can be defined as the strategy, which estimates model parameters and adjusts the controller settings with these parameter estimates. The self-tuning control configuration is flexible enough to accommodate a large number of parameter estimation techniques and controller design strategies. The most essential feature here is to have robust model identification, the model determining the effectiveness of the controller.

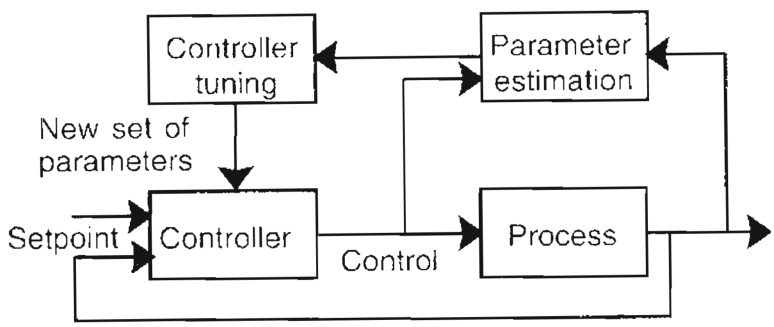

Figure 3.6: Self-tuning controller (Leiviska, 1999)

A self-tuning controller offers some advantages in situations where the controller requires continuous tuning because of the disturbance behaviour of the process. It has three parts (figure 3.6): the controller with adjustable parameters, a parameter estimator, and a tuning algorithm. Parameter estimation continuously calculates parameters in a process model using measurements of control and output variables. The tuning part calculates new tuning parameters for the controller. Several variations concerning the algorithmic realization of the functional blocks exist. The controller tuning can use required gain and phase margins, pole placement or minimum variance control. In practice, self-tuning control faces several problems. The most difficult is the drifting of controller parameters when the process is stable and no changes occur. One possibility is to use the self-tuning feature only in tuning the controller and "forget" it when no changes occur. This means loss of some advantages of the self-tuning approach.

Finally, the MRAC was obtained by considering a deterministic servo-problem and the STAC by considering a stochastic regulation problem. In spite of the differences in their origin it is clear that they are closely related. The inner loop is an ordinary feedback loop with a process and a regulator. The regulator has adjustable parameters, which are set by the outer loop. The adjustments are based on feedback from the process inputs and outputs. The methods for design of the inner loop and the techniques used to adjust the parameters in the outer loop may be different, however.

3.2 Modelling and simulation

3.2.1 Modelling

3.2.1.1 Model classification

Models may be classified according to different features. The development of a model always fulfils a certain purpose that requires defining the model structure with as much detail as possible. The purpose and application area define the model type. The actual model formulation depends on model type and available data. Different models also have use in different stages of modelling. Process and control system analysis employs modelling techniques for testing, tuning and optimisation.

Models also fall into classifications according to the modelling principle. There are analytical and experimental models (black box models). The analytical approach starts from the theoretical analytical principles or mechanisms that cover the operation of the physical system. These mechanisms define the method of description used in the modelling and the assumptions. The representation of the plant behaviour in terms of mathematical statements is open to various interpretations. The representation can be based on observed behaviour, it can be based on natural chemical and physical laws or it may be a combination of these two approaches. In addition, it may be a very detailed representation or it may be only a very simple form indicating approximate behaviour or behaviour within a limited operating region. Interaction with other plants may be important, the dynamic behaviour may be of more significance than steady state operation, a lot may be known about a process and its inputs or only very little; the plant may exist or it may be at the design stage.

Because of the many factors which need consideration, there are many ways in which modelling can be classified and formalized. The divisions may be made in terms of non-linear and linear, lumped and distributed parameters, deterministic and stochastic models.

Selection of model type depends on the problem itself. In process analysis, two separate cases exist. One is to describe the internal behaviour of the process, and the other is to describe the input-output relationships between process control variables. These cases require different types of models. In the first case, detailed models describing the physical and chemical phenomena in the process are necessary. In the second case, simpler input/output models are sufficient. Process control and control systems design use simple dynamic models that can be stochastic, adaptive or both.

3.2.1.2 Model construction

Models require restriction according to the processes or systems included in the simulation. Model construction starts from a division of the system to be modelled into two parts: the model and the environment. In the model, all variables and aspects that are important for the investigation are collected. All other characteristics of the system are part of the environment. The model and the environment interact, but a common idea in the division is that the influence from the model to the environment is small. The model will ultimately be a set of mathematical equations. Simulating the system then means a solution of the model equations. This sometimes requires a reformulation of the problem to a solution algorithm programmed into the computer. The formulation of a model equation as an algorithm makes possible the simulation of the system on a digital computer. In selecting a specific programming language, consideration of the objective of the simulation study is again important because different languages have different features.

At the start of a model development, it must be clear as to the purpose of the model, the extent (boundaries) of the model, which variables need consideration and the type of model that will be used. The purpose of the model plays a major role in determining the form of the model. To some extent, the boundaries, which determine the system size, will determine in addition both the complexity of the model and the source of the model equations.

Before the model can be developed at all, the necessary variables must be identified in order that the plant behaviour can be adequately described. These variables fall essentially into two categories, the independent variables (input variables) and the dependent (output variables) whose quality is dependent on the plant behaviour.

The computational model is comprised of mathematical relationships between the variables. The subdivision of this type is open to a considerable number of variations. The process of establishing the relationships between the variables is referred to as development. In order to obtain some measure of process performance a performance function is sought by which the criterion will include economic and quality considerations and will be subject to constraints in many cases. This function is a basis of optimisation studies. The process model equations form a very large part of many process control problems. The relationships may be established on an essentially theoretical basis but some empirical help or confirmation will generally be required.

The theoretically-based models use the basic physical and chemical rules, as well as the continuity equations. The dynamic equations will be ordinary differential equations or partial differential equations, the ODEs being reduced for steady state cases to algebraic equations. Major advantages of a fully theoretical model are its reliability and flexibility in allowing for major changes in plant and control, in predicting behaviour over a wide operating range, and in our ability to represent the behaviour of a new plant.

3.2.1.3 Model refinement and validation

Refinement concerns the taking of the model and verifying it, reducing it, and/or improving it. The process of development and refinement are thus closely related and refinement may be considered as a part of the development of the model to its final form. It may be found that the model is of a size or complexity which renders it unsuitable for use as it stands, or it may need restating in a form more suited for computation or linearizing prior to further use. If the complexity is too great then some reduction in size is necessary which may be effected by neglecting some effects, by making additional simplifications and approximations, or by direct model reduction techniques. During reduction, certain physical effects may be lost. It is important that the correct variables and a physical understanding are retained and some partitioning of the model may be required.

The validation of a simulation is the most important phase of any simulation study. In this phase, one is assuring that the model is a true representation of the system under consideration. This means comparing the model and the actual system to ensure that they give the same responses. If this is not the case, the model requires changing by tuning model parameters or by changing the model structure. To ensure reliable use of the model, all variables must have a certain area of confidence. When using simulation results, one must know the inherent limitations of their applicability. Model transferability is sometimes problematic. Re-parameterisation is the least that is necessary when using the developed model in another process.

3.2.2 **Simulation**

Simulation, while not common, is being increasingly used in the pulp and paper industry. Two main types of simulation exist: the steady state simulation (static approach) and the dynamic simulation (dynamic approach). Dynamic models are less common than steady state ones, being considerably more expensive to develop and validate. Dynamic simulation involves discrete changes or continuous changes. The time variable may be either continuous or discrete depending on whether the discrete changes in the dependent variables can occur at any point in the time (continuous) or only at the specified points (discrete).

The simulation contains three parts:

- the data set, which includes the input conditions, the processing sequence, the order of calculation, and information pertaining to the calculations in the unit computations.
- the executive program, which transmits information through the streams and stores the calculated results. It can plan the sequence in which calculations are to be done if it is requested to do so.
- the unit computation, a set of equations that predicts what occurs within an equipment unit.

Steady state flow sheet programs have had use since the 1960s in the chemical industry. Their potential applications are for mill engineering design and development of operating strategies for existing mills. Some dynamic flow sheet simulators have also been developed. These programs have use in operator training, development of control systems, analysis of start-up and shutdown situations, and operations scheduling.

The flow sheet simulators use two solution techniques:

- A modular simultaneous approach is one alternative for solving flow-sheeting problems. In this approach, the information contained by the flow sheet and modules is converted into a set of linear equations solved simultaneously. This method minimizes the number of iterations to provide a rapid solution. The first approach of our simulator in the present work was based on this method.

- The modular sequential approach is the most common technique. It computes modules singly in a certain sequence that usually follows the direction of physical flows in the system under study. This requires knowledge about the input flows of the module to be calculated. Calculation of output flows uses the input flows and module parameters. If recycle flows exist, their values require an initial guess in the beginning with iterative improvement during the simulation. The iteration continues until the balance constraints are satisfied and convergence occurs.

3.2.3 **Simulation environment**

For a variety of reasons, partly economic perhaps, the pulp and paper industry was until recently poorly served by the standard chemical process simulators. The reasons for this are partly explained by the unique characteristics of this industry that makes little use of a large physical property database of say hydrocarbons, or a comprehensive thermodynamic library, but does need data on components such as pulp fibre types, fines, clay additives, non-Newtonian flow models, and correlations of paper properties such as bending stiffness, tear strength and brightness. Given the unique characteristics and special requirements of the pulp and paper industry and taking into consideration the typical user for a simulator, the ideal modelling and simulation software should assist in a number of tasks such as the construction of the model (to avoid high index problems), subsequent simplification (model reduction), and changing the algorithm (perhaps using symbolic manipulation if necessary) if the requirements change. A block diagram approach is favoured since it is natural for chemical engineers used to a unit operations way of thinking.

Many process simulators have been developed. To a large degree the underlying software tool dictates much of what the model is capable of. Steady state simulators specifically directed to the pulp and paper industry such as WINGEMS of MassBal II tend to be used for retrofitting and design tasks, whereas dynamic simulators are more suited for control and operations. While some packages (e.g. FlowMac) now have the capacity for dynamics of flows and levels, they are essentially steady state flowsheeting mass balance models and are used as such. The most common simulator specific to the pulp and paper industry is MAPPS (Modular Analysis of Pulp and Paper Systems). MAPPS includes many of the unit operations used in the paper industry, including kraft digesters and recovery, stock preparation, and paper machine operations. Additionally, MAPPS contains a set of correlations that allow the prediction of paper properties from the composition of the pulp stream. MAPPS also includes data on wood species as well as many pulp and paper chemicals. It is capable of running whole mill simulations. GEMS also has many of the basic models necessary for modelling pulp and paper operations, although the list is not as extensive as that for MAPPS. Like MAPPS, GEMS is capable of running whole mill simulations and contains data for steam, moist air, and other chemical components. Other simulators that can be used for pulp and paper mill simulations include ASPEN PLUS, MASSBAL, and FlowCalc. For the most part, these are steady state simulators.

The dynamic simulators currently fall into two camps: those with an academic heritage (SpeedUp, gProms, Omola, Matlab/Simulink, Dymola/Modelica) which tend towards strong underlying numerical routines and often innovative thinking but are intended for general purpose modelling, and those with an industrial heritage (FlowMac, IDEAS, Entech's VISSIM, KCLWEDGE, VTT's APMS (advanced paper mill simulator) which concentrate on the effort into building a high quality library of unit operations, typically restricted to certain industries.

3.3 Optimisation

Optimisation can be classified into two different categories: the static optimisation, as opposed to the dynamic optimisation. Static optimisation concerns systems for which the variables to optimise have reached their steady state, whereas for dynamic systems, the variables to optimise are still evolving with time.

3.3.1 **Performance index**

Within the control problem, the main concerns reside in reducing any error, which exists between a desired state, which may either be constant or changing as a function of time, and the actual state of the system. The principal form of the performance indices is the quadratic performance index (QPI). In this index, the cost function is a sum of weightings of the squares of the deviation of the individual state variables and of the control variables. This leads to particularly suitable mathematical treatment as well as satisfying the qualitative consideration of penalizing both large deviations from the desired state and also increased control actions more severely. Note that the QPI is not by any means the only index that could be chosen, and a linear weighting, for example, could also be chosen, but this is not necessarily so convenient. Also, in setting up the performance index criteria one hopes to have used the correct weightings, these can be changed with experience of the results. With the prescribed objective function, the control system, although optimal in the analytical sense, may be such that its implementation cannot in fact be justified or even physically carried out. A system less than "optimal" may then in fact be used.

Optimisation problems are made up of three basic ingredients:

- an objective function to be minimized or maximized.
- a set of unknowns or variables that affect the value of the objective function.
- a set of constraints that allow the unknowns to take on certain values but exclude others.

The optimisation problem is then to find values of the variables that minimize or maximize the objective function while satisfying the constraints.

3.3.2 **Optimisation methods**

The numerical optimisation of general non-linear multivariable objective functions requires that efficient and robust techniques be employed. Most iterative procedures that are effective alternate between two phases in the optimisation: choosing a search direction and minimizing in that direction to find a new point. In addition, an algorithm must specify the initial starting vector and the convergence criteria. There are two different types of methods in optimisation: the direct ones and the indirect ones.

3.3.2.1 Direct methods

Direct methods do not require the use of derivatives in determining the search direction. They do have the advantage of being simple to understand and execute.

- A random search method simply selects a starting vector x0, evaluates f(x)a at x0, and then randomly selects another vector x1, and evaluates f(x) at x1. In effect, both a search direction and step length are chosen simultaneously. After one or more stages, the value of $f(x_k)$ is compared with the best previous value of f(x) and the decision is made to continue or not the procedure.

- On a grid search, a series of points is evaluated about a reference point selected according to some type of design (e.g. a square for an objective function of two variables).

- In a univariate search, a selection of n fixed search directions is done (usually the coordinate axes) for an objective function of n variables. Then f(x) is minimized in each search direction sequentially using a one-dimensional search.

- The method of the "sequential simplex" formulated by Spendley, Hext and Himsworth (1962) uses a regular geometric figure (simplex) to select points at the vertices of the simplex at which to evaluate f(x). In two dimensions, the figure is an equilateral triangle, in three dimensions, this figure becomes a tetradrehon. Each search direction points away from the vertex having the highest value of f(x). Thus, the direction of search changes, but the step size is fixed for a given size simplex. As the optimum is approached, the procedure cannot get closer to the optimum and repeat itself so that the simplex size must be reduced, such as halving the length of all the sides of the simplex containing the vertex where the oscillation started. Thus, the optimum position is determined to within a tolerance influenced by the size of the simplex.

- In the conjugate search direction method, the two directions of search (s^i and s^j) are conjugate with respect to each other (eq 3-1).

$$(s^i)^T Q(s^j) = 0 \qquad \textbf{(3-1)}$$

Q being the Hessian matrix of the objective function.

- Powell's method (1965) locates the minimum of a function f by sequential unidimensional searches from an initial point along a set of conjugate directions generated by the algorithm. New search directions are introduced as the search progresses.

Those direct method were the earliest methods proposed for unconstrained optimisation, they are not as efficient and robust as many of the more modern indirect methods described in the following section.

3.3.2.2 Indirect methods

As opposed to direct methods, indirect methods do make use of derivatives in determining the search direction for optimisation.

- The gradient method uses only the first derivatives of the objective function in the calculations. The search direction s^k is simply the negative gradient of the objective function f (eq 3-2).

$$s^k = -\nabla f(x^k) \tag{3-2}$$

In the steepest descent, the transition from point x^k to another point x^{k+1} is done by (eq 3-3):

$$x^{k+1} = x^k - \lambda^k \nabla f(x^k) \tag{3-3}$$

λ^k being the scalar that determines the step length in direction s^k.

At the minimum, the value of the elements of the vector gradient will each be equal to zero. A procedure of strictly steepest descent can terminate at any type of stationary point, i.e. at a point where the elements of the gradient of *f(x)* are zero. Thus, a check must be done regarding the presumed minimum to know whether its indeed a local minimum or a saddle point and in this case, it is necessary to employ a non-gradient method to move away from the point, after which the minimization may continue as before.

- The conjugate gradient method was devised by Fletcher and Reeves (1964). It combines current information about the gradient vector with that of gradient vectors from previous search directions.

- Newton's method (cubic iteration) makes use of the second-order approximation of f(x) at x^k, and thus employs second-order information about f(x), that is, information obtained from the second partial derivatives of f(x) with respect to the independent variables. Thus, it is possible to take into account the curvature of f(x) at s^k and identify better search directions.

- Another method that can be cited is the secant method. By analogy with the secant technique for functions of a single variable, the procedure minimizes f(x) using only values of f(x) and ∇f(x), the Hessian matrix of f(x) is approximated from combinations of the two.

3.3.2.3 Constrained optimisation

The presence of inequality constraints on the state, control and output variables is a distinctive feature of many practical control problems. In process control problems, constraints can occur due to physical limitations of plant equipment such as pumps or control valves. Linear programming (LP) has proved to be a powerful method for solving optimal control problems for linear systems with linear equality and inequality constraints. In Chang and Seborg (1983), a new feedback control strategy is developed for multivariable problems which have inequality constraints on the variables. At each sampling instant, a linear programming problem is solved on-line to determine the values of the control variables which minimise a linear performance index while satisfying the inequality constraints. In order to use an LP algorithm, a linear performance index must be selected. A reasonable choice is the weighted sum of absolute errors (eq 3-4). The objective function becomes:

$$J_1(k) = \sum_{i=1}^{n} w_i \left| e_i'(k+1) + M_i \sum_{l=1}^{k} e_i(l) \right| \qquad \textbf{(3-4)}$$
$$e_i'(k+1) = x_i'(k+1) - x_{id}(k)$$

w_i are the non-negative weighting factors for the ith state variable,

e'_i and x'_i are predicted values,

M_i represents the reciprocal of the integral time constant, $1/\tau_i$ in conventional PID controllers.

These weighting factors are used to penalize predicted errors in the more important state variables. The summation term, $\sum e_i(l)$, provides integral control action to eliminate offsets when sustained disturbances or modelling errors occur. Results of this study demonstrated that the LP approach compares favourably with other multi-variable control techniques, such as optimal feedback by dynamic programming.

3.4 Modelling and control in the pulp and paper industry

A review of the last thirty years of modelling and process control literature reveals that many of the industry-specific problems have been worked on and solutions to some of the complex problems have emerged. As noticed by F.Kayihan (1996), the pulp and paper industry represents a big challenge. Firstly, its main raw material is a natural product that retains its non-uniform characteristics and behaviour through the manufacturing processes all the way to the final consumer products. And secondly, it has to comply with increasingly stringent environmental requirements and be responsive to customers' demands for using environmentally friendly processes. Consequently, the understanding of the process behaviour through mathematical modelling has improved over the years. Because of their critical importance, major processes like digesters, recovery boilers, pulp and paper machines and evaporators continue to be modelled. At the same time, process control is improving due to the requirements of product quality. Pulp and paper processes are typically very similar to systems to which chemical engineers have been accustomed. Therefore, appropriate and successful control approaches are also similar in principle to what has already been established through applications in petroleum and chemical industries. The challenge with the pulp and paper industries is due to the stochastic nature of the raw material, highly interactive multivariate process behaviour, long time delays and grade change transients.

In principle, process control issues of the stock preparation area are normally quite straightforward, but in practice, they require a lot of attention, mainly due to process and loop interactions, sensor problems, changes in operating conditions and imperfect mixing tanks. From this point within the process to the end of sheet forming, total material balance control is crucial as the final product is highly sensitive to consistency and flow variations in fibre delivery.

In Haag and Wilson (2001), a large-scale dynamic model of a board machine is developed and implemented in Simons IDEA (commercial block oriented dynamic simulation environment). The model covers all unit operations from the pulp storage towers to the finished sheet on the jumbo roll, including the broke system and the long circulation of white water. The model can predict the fractions of four kinds of fibres, water, fines, filler, starch and COD, as well as paper properties such as density, E-modulus and bending stiffness. The model is based on fundamental mass and momentum balances where possible, although it resorts to semi-empirical regressions where little is known. Model inputs include material flows, properties of pulps, refining energy and tuning constants for control loops. Several simulations were carried out with this model to quantify variations in the medium to long term. A simulation of grade change was done after a long period of producing the same grade. The basis weight rapidly reached its setpoint but the bending force takes considerably longer to reach the quality limit of the grade due to the considerable volume of the pulp of previous quality. The combination of this scheme with a feedback controller stabilizes the system and the response is further improved. Other experiments on the chemical additive response and variation in broke input effects were carried out. Major problems of the model concerned the excessive simulation turnaround times and the lack of knowledge of the process. Besides, the complexity of differential and algebraic equation set rapidly became too unwieldy for some uses and it was found completely unsuitable for online model-based control purposes.

3.4.1 **Advanced control**

The late seventies marked an important turning point in control theory. The development of multivariable control methods, based on the state-space approach, along with the developments of identification and adaptive control methods, were major breakthroughs. The development of sophisticated identification methods allows the designer to empirically obtain a good model of the process. The development of applicable adaptive control methods eliminates the need for accurate models and allows control of time-varying processes.

Pioneering work in adaptive control was performed in the late sixties and early seventies, using the paper machine as a benchmark. The first application of self tuning regulators (STAC) reported by a supplier of turnkey control systems to the industry can be found in Fjeld and Wilhelm (1981), Wilhem(1982) and Kelly (1982) with an application to machine direction (MD) control of moisture on a fine paper machine. The algorithm uses the Clarke-Gawthrop (1979) self-tuning controller (STC) in "velocity form" with fixed forgetting factors.

A similar approach is taken to control the cross-machine direction (CD) weight profile (Wilhem and Fjeld 1983). A space recursive model is used to represent the action of a slice actuator on the basis-weight profile. The three parameters characterizing this action are identified on-line. The results are used to compute the slice actuator movements to obtain the desired profile, solving the decoupling problem between the numerous actuators by means of generalized-minimum variance control theory.

In many cases the headbox control problem is used to test new developments in multivariable STAC methods. Borisson (1979) used it with an extension of minimum-variance control to n*n MIMO systems with a single dead time applied to a simulated headbox.

Halme and Selkainaho (1982) describe an application of a non-linear filter coupled with a state-space controller to a pilot plant headbox. The non-linear filter estimates both parameters and states. A multivariable PI regulator is then tuned to minimize a quadratic criterion.

In D'Hulster et al. (1980), the performance of several SISO adaptive controllers is compared on a simulated headbox. Decoupling is performed by adding feedforward compensation terms in the model, then using two MISO controllers. All adaptive controllers outperform conventional PID control.

In Halme and Ahava (1984), the plant dynamics are estimated via a modified recursive least-squares scheme and the regulator tuned to minimize a non-quadratic performance index.

3.4.2 **Dynamic models**

Most of the simulations concern a specific operation in the papermaking process such as a digester, a refiner, a headbox or a press but only a few deal with the overall process.

3.4.2.1 Refiners

For a chip refiner, Di Ruscio and Balchen (1992) developed a dynamic model for the fibre size distribution resulting from refining, and showed that unknown model parameters can be recursively estimated. Qian and Tessier (1993) developed a model to relate sheet properties to refining operating conditions. Subsequently, Tessier and Qian (1994) modelled the dynamic behaviour of the system from the chip silos to the secondary refiners in order to study the impacts of process disturbances and operating conditions on performance. Partanen and Kovio (1984) directly measured pulp freeness on-line and developed a non-linear PI algorithm to control a thermo-mechanical plant. Kooi and Khorasani (1992) developed static and dynamic neural network based controllers to replace self-tuning regulators for chip refiners. They found the dynamic neural networks to be more effective.

3.4.2.2 Stock preparation and wet end

In principle, process control issues of the stock preparation area are normally quite straightforward but in practice, they require a lot of attention, mainly due to process and loop interactions, sensor problems, changes in operating conditions and imperfect mixing tanks.

In Dumdie (1988), a strategy is developed to control the consistency at the wet end of the paper machine, as good consistency control is critical. Traditionally, blend controls consist of setting fibre stock ratios for each furnish on a volumetric basis. Ideally, if each furnish can be controlled at the same consistency, the wet and dry blends will be the same. However because of process upsets and specific mill operating objectives, this control is not always possible. From this point within the process to the end of sheet forming, total material balance control is crucial as the final product is highly sensitive to consistency and flow variations in fibre delivery. During the wet-end operations (between the machine chest and the end of the former), significant amount of water called white water is required. White water systems have been modelled for dynamic behaviour and process sensitivity analysis. (Kaunonen et al.,1985, Virtanen, 1988, and Bussiere, Roche and Paris, 1992). Bialkowski (1991) implemented a decoupled control design for a multi-ply headbox system. Piirto and Koivo (1992) used a self-tuning SISO controller for retention with self-tuning MIMO for headbox control.

3.4.2.3 Paper machines

The product of the paper machine is a continuous sheet for which all of the properties like basis weight, moisture, caliper, colour, smoothness and strength are expected to be on target and uniform everywhere. Bond and Dumont (1988) used the Dahlin algorithm (1968) with gain scheduling and recursive model identification for a colour control scheme. Vincent et al. (1994) developed an adaptive approach using both linear and non-linear predictions to control colour during grade transitions. Roa and coworkers proposed basis weight and moisture control schemes using bilinear suboptmial control (Ying, Rao and Sun, 1991), MIMO adaptive (Xia, Rao, Shen and Zhu, 1993, 1994), and model algorithmic control based on impulse response model approaches (Xia,Rao and Qian, 1993). Chen and Wilhem (1986), Chen (1988) and Chen and Adler (1990) developed commercial controllers using a dual Kalman filtering approach to estimate both temporal and spatial variations including the use of quadratic penalty functions to handle constraints on the control action.

Chapter 4 : Modelling and control

A model means, in the context of a study of the dynamics and control of a plant, a representation of the plant behaviour in terms of mathematical statements. The combination of model and control theory enables the behaviour of the plant under various conditions and controls to be investigated, it helps in the choice of controls for a given plant, and possibly leads to changes at the design stage of the plant.

To be familiar with process and control operations requires data, information, and knowledge. Several areas of engineering use mathematical models. In research and development, mathematical models have use in studies of process internal phenomena such as flow, mixing, reactions, heat and mass transfer. The models provide a thorough understanding about what is actually happening inside the process. In product design for the process industries, different kinds of models define the effects of variables on product quality and amount. Simulation is the process of building a mathematical model for a system and using that model for a systematic investigation. In process design, modelling and simulation methods have use today to study alternatives for process equipment and connections, optimise process operation, and discover the best way to use raw materials and energy. In control engineering, models and simulation can determine control strategies for the process.

In the current modelling work, the program was written with Matlab, a software package available from The MathWorks,Inc.. It is designed for high performance numerical computation and visualization. Matlab incorporates a graphical interface together with matrix computation and numerical analysis functions. The interface allows models to be created and modified without the use of traditional programming languages: models are expressed in a mathematical and matrix-oriented form. More details about Matlab software are given in appendix D.

4.1 <u>The stock preparation area model: two different approaches</u>

In a first modelling attempt, the equation-solving approach was used. The mathematical model of the process is organized and handled as one large global set of equations representing the entire process.

The final approach was a sequential modular approach. The modular viewpoint involves collecting equations and constraints for each process unit into a module as well as the solution procedures for these equations. This concept is parallel to that of a unit operation. Each module calculates values pertaining to the output streams for the given input conditions and parameters for that process unit. The sequential part of the approach involves carrying out calculations from module to module, starting with the feed streams until the product is obtained.

In general the simulation requires the solving of a set of linear or non-linear simultaneous ordinary differential equations (ODE).

4.1.1 **Equation-solving approach**

For this approach, a simple example was used in order to test the modelling technique (figure 4.1). The process involves several unit operations of the stock preparation area (separation, mixing and dilution) and includes all of its controllers (level controller LC, flow controller FC, consistency controller CC and a ratio controller).

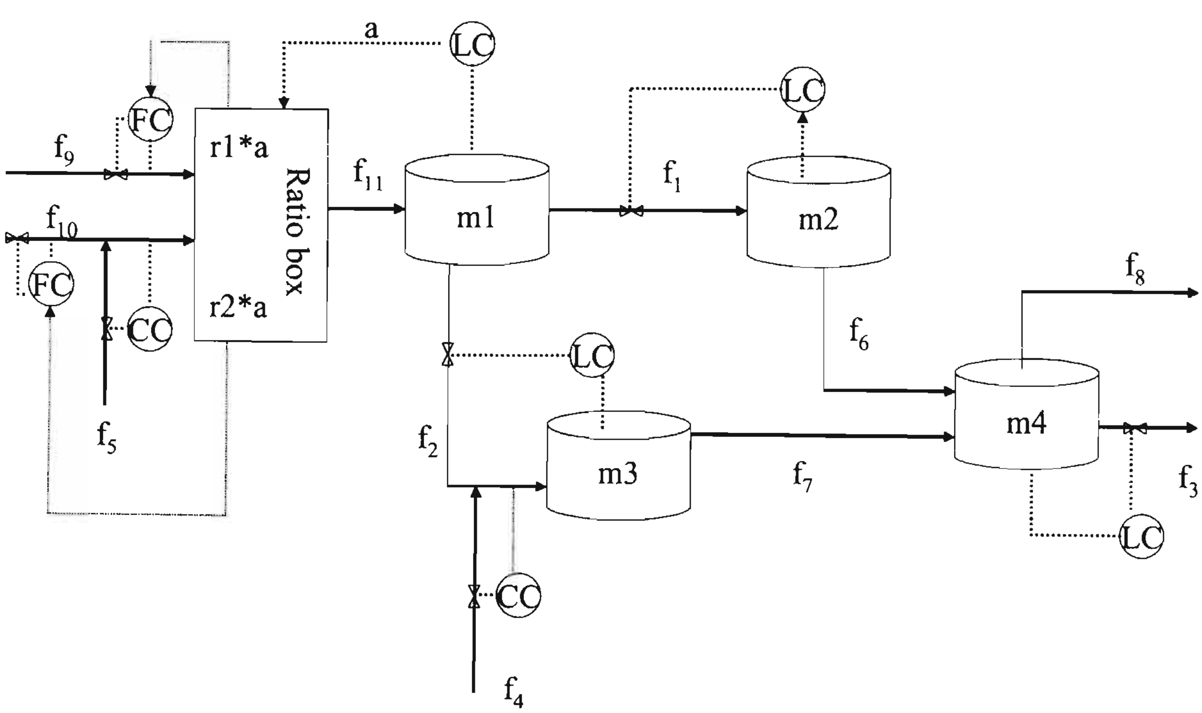

Figure 4.1: Test example for the equation-solving approach

In the structure of the program, each flow is submitted to the action of a flow controller. When not desired, this action can be switched off by adjusting the controller parameters (section 4.1.2.2).

4.1.1.1 Description of the approach

The flows were arranged depending on their functionality in the model (fig 4.1):

- flows set by a level controller f_m (f_1, f_2 *and* f_3)
- flows set by a consistency controller f_c (f_4 *and* f_5)
- flows set by a flow controller f_s (f_6, f_7 *and* f_8) (FC not displayed on the graphs)
- cascaded flows set by a flow controller f_a (f_9 *and* f_{10})
- the intermediate flow set by the level controller a (f_{11})

Concerning the controllers, only proportional and integral actions were taken into account (PI controllers) as it is on the actual plant. The derivative action is omitted. The derivative of the action thus yields a derivative for the proportional term, and the integral term yields the original setpoint deviation. PID controller actions are explained in Appendix E.

On each cell m_k of the system were applied the continuity equations (eq 4-1 and 4-2):

- total mass balance : the time rate of change of mass inside the system is the mass flow into the system minus the mass flow out.

$$\frac{dm_k}{dt} = \sum_{in} f_{i,k} - \sum_{out} f_{o,k} \qquad \textbf{(4-1)}$$

- component mass balance :

$$\frac{dm_k c_k}{dt} = m_k \frac{dc_k}{dt} + c_k \frac{dm_k}{dt} = \sum_{in} f_{i,k} c_i - \sum_{out} f_{o,k} c_k \qquad \textbf{(4-2)}$$

On each controlled flow was applied the PI action:

- level controller action on the flows f_m : (k_{cm} gain, k_{im} integral time)

$$\frac{d\overline{f_m}}{dt} = k_{cm} \frac{d\left(\overline{m} - \overline{m}_{SP}\right)}{dt} + k_{im}\left(\overline{m} - \overline{m}_{SP}\right) \qquad \textbf{(4-3)}$$

- level controller action on the intermediate flow a (f_{11}) :

(k_{cma} gain, k_{ima} integral time)

$$\frac{d\bar{a}}{dt} = k_{cma}\frac{d\left(\bar{m}-\bar{m}_{SP}\right)}{dt} + k_{ima}\left(\bar{m}-\bar{m}_{SP}\right) \tag{4-4}$$

- consistency controller action on the flows f_c : (k_{cc} gain, k_{ic} integral time)

$$\frac{d\bar{f_c}}{dt} = k_{cc}\frac{d\left(\bar{c}-\bar{c}_{SP}\right)}{dt} + k_{ic}\left(\bar{c}-\bar{c}_{SP}\right) \tag{4-5}$$

- flow controller action flows f_s : (k_{cs} gain, k_{is} integral time)

$$\frac{d\bar{x}}{dt} = k_{cs}\frac{d\left(\bar{f_s}-\bar{f}_{s\,SP}\right)}{dt} + k_{is}\left(\bar{f_s}-\bar{f}_{s\,SP}\right) \tag{4-6}$$

- flow controller action on fixed flows f_a : (k_{ca} gain, k_{ia} integral time)

$$\frac{d\bar{x}}{dt} = k_{ca}\frac{d\left(\bar{f_a}-\bar{f}_{a\,SP}\right)}{dt} + k_{ia}\left(\bar{f_a}-\bar{f}_{a\,SP}\right) \tag{4-7}$$

- valve-flow relation: (τ_i time constant)

$$\frac{d\bar{f}}{dt} = T\cdot(\bar{x}-\bar{f}) \tag{4-8}$$

$$T = \frac{1}{\tau_I}\cdot I$$

The action of the valve (x) on the flow (f) (eq 4-8) is a first order relationship. To take this action into account and to be able to combine equation **(4-8)** with equations **(4-6)** and **(4-7)**, the order of the system is increased.

Therefore, the final flow controller action becomes:

$$\frac{d^2\bar{f}}{dt^2} = T\cdot\left(\frac{d\bar{x}}{dt}-\frac{d\bar{f}}{dt}\right) \tag{4-9}$$

Hence, the final set of second order differential equations in matrix form:

$$\frac{d^2\bar{m}}{dt^2} = \alpha\frac{d\bar{f}}{dt} \quad (\alpha \text{ matrix}) \tag{4-10}$$

$$\frac{d^2\bar{c}}{dt^2} = \beta\frac{d\bar{c}}{dt} + \gamma\frac{d\bar{f}}{dt} + \beta_2\frac{d\overline{c_{ext}}}{dt} \quad (\beta,\gamma,\beta_1 \text{ matrices, } c_{ext} \text{ external consistencies}) \tag{4-11}$$

$$\frac{d^2\overline{f_m}}{dt^2} = k_{im}\frac{d\overline{m}}{dt} + k_{cm}\alpha - k_{cm}\frac{d^2\overline{m_{SP}}}{dt^2} - k_{im}\frac{d\overline{m_{SP}}}{dt} \tag{4-12}$$

$$\frac{d^2\overline{f_c}}{dt^2} = \left[k_{cc}\beta + k_{ic}\right]\frac{d\overline{c}}{dt} + k_{cc}\alpha\frac{d\overline{f}}{dt} - k_{cc}\frac{d^2\overline{c_{SP}}}{dt^2} - k_{ic}\frac{d\overline{c_{SP}}}{dt} + k_{cc}\beta_2\frac{d\overline{c_{ext}}}{dt} \tag{4-13}$$

$$\frac{d^2\overline{f_s}}{dt^2} = T\left[k_{is} - I\right]\frac{d\overline{f_s}}{dt} + Tk_{is}\overline{f_s} - Tk_{cs}\frac{d\overline{f_{s_{SP}}}}{dt} - Tk_{is}\overline{f_{s_{SP}}} \tag{4-14}$$

$$\frac{d^2\overline{f_a}}{dt^2} = T\left(k_{ca} - I\right)\frac{d\overline{f_a}}{dt} - Tk_{ca}R_a\frac{d\overline{a}}{dt} + Tk_{ia}\overline{f_a} - Tk_{ia}R_a\overline{a} \tag{4-15}$$

$$\frac{d^2\overline{a}}{dt^2} = k_{ima}\frac{d\overline{m}}{dt} + k_{cma}\alpha\frac{d\overline{f}}{dt} - k_{cma}\frac{d^2\overline{m_{SP}}}{dt^2} - k_{ima}\frac{d\overline{m_{SP}}}{dt} \tag{4-16}$$

This overall system was rearranged as a set of first order differential equations (eq 4-17) and is represented in figure 4.2:

$$\frac{d\overline{y}}{dt} = \mathbf{A}\overline{y} + \mathbf{B}\overline{u} \tag{4-17}$$

The y vector gathers all of the necessary data concerning the mass (*m*), the consistency (*c*), the flows (*f and a*) and their derivative forms ($\dot{m}, \dot{c}, \dot{f}, \dot{a}$). The u vector gathers the external input of the system such as the mass setpoints (m_{SP}), the consistency setpoints (c_{SP}) and the flow setpoints (f_{SP}).

$$
\begin{array}{c}
y \\
\begin{bmatrix} \dot{m} \\ \dot{c} \\ \dot{f}_m \\ \dot{f}_c \\ \dot{f}_s \\ \dot{f}_a \\ \dot{a} \\ m \\ c \\ f_m \\ f_c \\ f_s \\ f_a \\ a \end{bmatrix}
\end{array}
\frac{d}{dt}
=
\begin{array}{c}
\begin{array}{cccccccccccccc}
\dot{m} & \dot{c} & \dot{f}_m & \dot{f}_c & \dot{f}_s & \dot{f}_a & \dot{a} & m & c & f_m & f_c & f_s & f_a & a
\end{array} \\
\left[\begin{array}{cccccccccccccc}
 & & \multicolumn{5}{c}{[\quad \alpha \quad]} & & & & & & & \\
 & \beta & \multicolumn{5}{c}{[\quad \gamma \quad]} & & & & & & & \\
k_{im} & & \multicolumn{5}{c}{[\quad k_{cm}\alpha \quad]} & & & & & & & \\
 & (k_{cc}\beta + k_{ic}) & \multicolumn{5}{c}{[\quad k_{cc}\alpha \quad]} & & & & & & & \\
 & & & & T(k_{cs} - I) & & & & & & & Tk_{is} & & \\
 & & & & & T(k_{ca} - I) & -Tk_{ca}R_a & & & & & & Tk_{ia} & -Tk_{ia}R_a \\
k_{ima} & & \multicolumn{5}{c}{[\quad k_{cma}\alpha \quad]} & & & & & & & \\
I & & & & & & & & & & & & & \\
 & I & & & & & & & & & & & & \\
 & & I & & & & & & & & & & & \\
 & & & I & & & & & & & & & & \\
 & & & & I & & & & & & & & & \\
 & & & & & I & & & & & & & & \\
 & & & & & & I & & & & & & &
\end{array}\right]
\end{array}
\bar{y} + B.\bar{u}
$$

Figure 4.2: Matrix representation

The **A** matrix is a square matrix. α, β, γ are matrices of the system, the different k_c and k_i terms represent respectively the proportional and integral term matrices, T the integral time for the valve flow relation, R_A the cascaded controllers ratio matrix and I the identity matrix.

4.1.1.2 Discrete form

This equation is then discretized. The first step consists of using the Laplace transforms. Laplace transformation converts functions from the time domain (where *t* is the independent variable) into the Laplace domain (where *s* is the independent variable). For a non-zero initial value, the Laplace transform ($L([])$) of a derivative form is:

$$L\left[\frac{d\overline{y}}{dt}\right] = s\overline{Y}(s) - \overline{y}(0)$$

Therefore (5-17) gives:

$$s\overline{Y}(s) - \overline{y}(0) = \mathbf{A}\overline{Y}(s) + \mathbf{B}\overline{U}(s)$$

For a typical step input:

$$\overline{U}(s) = \frac{1}{s}\overline{u}(0)$$

Regrouping the terms, the equation becomes

$$[s\mathbf{I} - \mathbf{A}]\overline{Y}(s) = \overline{y}(0) + \frac{1}{s}\mathbf{B}\overline{u}(0)$$

$$\overline{Y}(s) = [s\mathbf{I} - \mathbf{A}]^{-1}\overline{y}(0) + [s\mathbf{I} - \mathbf{A}]^{-1}\frac{1}{s}\mathbf{B}\overline{u}(0)$$

Using a Partial Fraction Expansion (PFE), the second term of the right side gives:

$$\begin{aligned}
[s\mathbf{I} - \mathbf{A}]^{-1}[s\mathbf{I}]^{-1} &= [s\mathbf{I}]^{-1}[s\mathbf{I} - \mathbf{A}]^{-1} && ([sI]\text{diagonal}) \\
&= -\mathbf{A}^{-1}[s\mathbf{I}]^{-1} + [s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{A}^{-1} && (\text{PFE}) \\
&= -[s\mathbf{I}]^{-1}\mathbf{A}^{-1} + [s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{A}^{-1} && ([sI]\text{diagonal}) \\
&= \left([s\mathbf{I} - \mathbf{A}]^{-1} - [s\mathbf{I}]^{-1}\right)\mathbf{A}^{-1}
\end{aligned}$$

and finally

$$\overline{Y}(s) = [s\mathbf{I} - \mathbf{A}]^{-1}\overline{y}(0) + \left([s\mathbf{I} - \mathbf{A}]^{-1} - [s\mathbf{I}^{-1}]\right)\mathbf{A}^{-1}\mathbf{B}\overline{u}(0) \qquad \textbf{(4-18)}$$

Defining $L^{-1}\left([s\mathbf{I} - \mathbf{A}]\right)^{-1}$ as the matrix exponential symbol $e^{\mathbf{A}t}$, the Laplace transform equation is inverted. After transforming equations into the Laplace domain and solving for output variables as functions of s, it is useful to get back to the time domain, this operation is called inversion.

$$\overline{y}(t) = e^{\mathbf{A}t}\overline{y}(0) + \left[e^{\mathbf{A}t} - \mathbf{I}\right]\mathbf{A}^{-1}\mathbf{B}\overline{u}(0) \qquad \textbf{(4-19)}$$

Instead of looking at the time horizon $0 \rightarrow t$, rather consider the interval $t \rightarrow t + T$ with T sampling time of the system, the vector $\overline{u}(t)$ being constant during this interval.

$$\overline{y}(t+T)=e^{\mathbf{A}T}\overline{y}(t)+\left[e^{\mathbf{A}T}-\mathbf{I}\right]\mathbf{A}^{-1}\mathbf{B}\overline{u}(t) \qquad \textbf{(4-20)}$$

For a standard linear system of 1st order differential equations with constant coefficients $\frac{d\overline{y}}{dt}=\mathbf{A}\overline{y}+\mathbf{B}\overline{u}$, the discrete form of the equation becomes:

$$\overline{y}_{i+1}=e^{\mathbf{A}T}\overline{y}_i+\left(e^{\mathbf{A}T}-\mathbf{I}\right)\mathbf{A}^{-1}\mathbf{B}\overline{u}_i \qquad \textbf{(4-21)}$$

e^{AT} being the matrix exponential, T being the time constant of the system.

This final equation contains an inverse matrix form $\mathbf{A}^{-1}$. In many cases, the matrix A is singular but that does not prevent the equation from being solved. The A matrix coefficients can be evaluated using the Taylor expansion:

$$e^{\mathbf{A}\Delta t}\approx e^{[0]}+\frac{(\mathbf{A}\Delta t)}{1!}e^{[0]}+\frac{(\mathbf{A}\Delta t)^2}{2!}e^{[0]}+\frac{(\mathbf{A}\Delta t)^3}{3!}e^{[0]}+\ldots\text{with } e^{[0]}=\mathbf{I}$$

$$\left[e^{\mathbf{A}\Delta t}-\mathbf{I}\right]\mathbf{A}^{-1}=\Delta t\left\{\frac{\mathbf{I}}{1!}+\frac{(\mathbf{A}\Delta t)}{2!}+\frac{(\mathbf{A}\Delta t)^2}{3!}+\ldots\right\} \qquad \textbf{(4-22)}$$

In the coding of the model, the series is evaluated by adding terms until the sum of the absolute changes of the elements falls below a given tolerance.

4.1.1.3 Problems

The major problem concerning this method is the lack of readability of the solution. Moreover, the programming of the system is not generic enough. Each flow is enumerated depending on the way it is controlled so if the process is modified, the entire numbering has to change. An adaptation to the real stock preparation area would become much more complicated partly because of the recycled flows. To introduce the controllers into this open loop structure, it has been necessary to increase the order of the system (second order). The tuning of the controllers was very difficult due to interactions that changed the behaviour of the overall system.

This lack of readability and the interactions observed in the systems were convincing reasons to change the modelling approach to the sequential modular approach.

4.1.2 **Modular approach**

4.1.2.1 Description of the approach

The model is built up in a modular fashion using a basic element, having one input (which can collect multiple streams originating elsewhere) and four outputs, linked through a vessel of variable volume V and consistency C (figure 4.3).

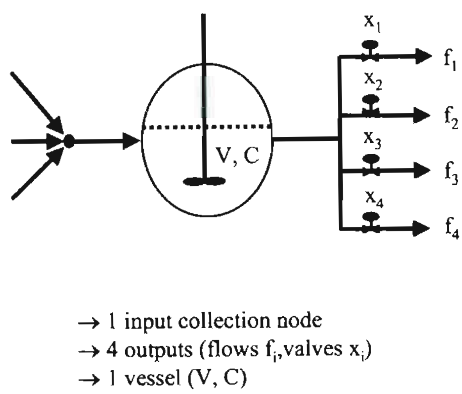

Figure 4.3: Modular approach basic element

The structure of the basic element enables the representation of two unit operations: mixing and separation. The input can collect several flows (mixing) and the tank splits them into a maximum of four streams (separation). The tank volume introduces the dynamics in the system. Therefore by setting a small volume in the tank and a fast time constant on the valve response, operations of summation and separation can be achieved with little lag, if desired. Several basic elements are linked together to form the overall system. All of the necessary properties can be defined so that the model allows the simulation of all features of the network: vessels, pipes, junctions, valves, levels and consistencies. Plant data are gathered into an "M" matrix, which contains:

- the algebraic equations (linking vessels to flows) [A B C D]

e.g. if A=2, the first output of the cell 1 (f_1) goes to the input of the cell 2.

- the fractional valve openings [W X Y Z]

e.g. if W=0.3, the valve (x_1) of the first output of the cell (f_1) is 30% open.

- the actual output flows of the cell [E F G H]

e.g. if E=150, the output flow f_1 is equal to 150 Ls^{-1}

- the time constants τ for the valves [τ_E τ_F τ_G τ_H] (indicator of the response speed of the flow to a valve change)

e.g. if a level controller has a direct action on the valve, the time constant τ is set really small to minimise the lag of the intermediate flow control loop

- the maximum cell volumes [V] (constant) and their initial consistencies [T] (variable)

- the maximum flows (or constraints) [E_{max} F_{max} G_{max} H_{max}], the minimum flow being set to zero

- the actual volume of liquid in each cell [Vinit]

A set of first order differential equations is solved, which includes **(4-23)** total mass balance, **(4-24)** species mass balances, **(4-26)** derivatives of flow controller action and derivatives of supervisory controller action, and finally a valve-flow relation **(4-25)**.

$$\frac{dV_k}{dt} = \sum_i f_{i,k} - \sum_o f_{o,k} \qquad \textbf{(4-23)}$$

$$\frac{dy_j}{dt} = \frac{1}{V_j}\left(\sum_i f_i y_i - \sum_o f_o y_j\right) - \frac{y_j}{V_j}\left(\sum_i f_i - \sum_o f_o\right) \qquad \textbf{(4-24)}$$

$$\frac{df_k}{dt} = -\frac{1}{\tau_k} f_k + \frac{f_{k_{\max}}}{\tau_k} x_k \qquad \textbf{(4-25)}$$

Differentiating the valve action of a flow controller, note that we only considered PI controllers, so no second order derivative on flow is obtained.

$$\frac{dx_k}{dt} = k_{c,k} \frac{d}{dt}\left(f_k - f_{kSP}\right) + \frac{k_{c,k}}{\tau_{i,k}}\left(f_k - f_{kSP}\right) \qquad \textbf{(4-26)}$$

Equations **(4-23)**, **(4-24)** and **(4-25)** are regrouped into a set of first order differential equations representing the open loop process.

$$\frac{d}{dt}\overline{L} = \mathbf{A}\overline{L} + \mathbf{M}\overline{N}$$

The vector L regroups the levels, the consistencies and the output flows; the vector N regroups the external data relative to this open loop process such as the valve openings, dilution flows and consistencies.

For a standard linear system of 1st order differential equations with constant coefficients, the discrete form of the equation becomes (section 4.1.1.2):

$$\overline{L}_{i+1} = e^{\mathbf{A}T}\overline{L}_i + \left(e^{\mathbf{A}T} - \mathbf{I}\right)\mathbf{A}^{-1}\mathbf{M}\overline{N}_i \qquad \textbf{(4-27)}$$

e^{AT} being the matrix exponential, T being the time constant of the system.

4.1.2.2 Program structure

Now that the equations have been established, the model structure must be detailed. The program is made of three major steps (figure 4.4) that occur in every time loop. Every single flow of the system is under the control of a FC (flow controller Gc_1). For some specific flows, this FC is itself under the control of a LC or a CC, those controllers being cascaded (see advanced controller Gc_2). The first step of the program considers those specific flows manipulated by an 'advanced controller'. As an input it takes the controlled properties setpoints and evaluates the flow setpoints to be sent to the cascaded flow controller. The second step of the program represents the action of the flow controllers onto the valves. Therefore, it determines the valve positions to adopt to satisfy the system demands. Finally, the third step is the open loop process. It contains all of the system information, and calculates from the valve positions the 'new' properties (volumes, flows and consistencies). To close the loop, some of those 'new' properties are compared again to the setpoints of the advanced controllers and the program goes on.

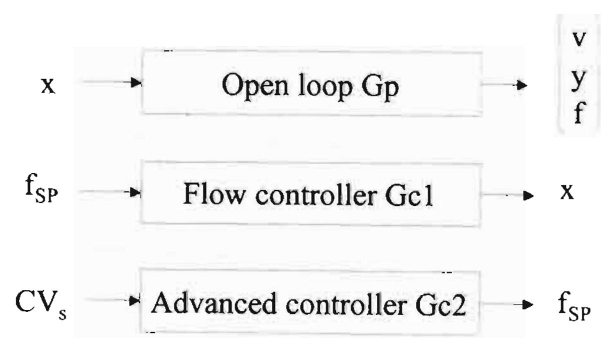

Figure 4.4: Program structure

In other words, supervisory controllers for consistency or level (CV_s) cascade onto flow controllers. Flow controllers (f_s) manipulate valves (x) that give a first-order dynamic response of actual flow. Where valves are manipulated *directly* by the supervisory level, the flow controller is effectively bypassed by making it fast and giving the valve a quick response.

The model covers all unit operations from the initial pulp storage towers to the headbox including the internal loops (filters and cleaners) as shown in figure 4.5. Model inputs include the hundred or so material flows, properties of the different pulps and tuning constants for control loops. There are likewise many parameters such as tank volumes and pipe sizes (maximum flows). In any industrial system of this magnitude, not all of the inputs or parameters have the same impact on the model outputs. However a semi-rigorous model has the advantage that physical components on the machine have a one-to-one correspondence to elements in the model (even if some elements of the model do not have a physical significance). This makes it easy to rebuild the model following machine retrofits, and presents a more natural interface to the operators. Most importantly, a certain degree of fundamentalism has been applied in the creation of the model. Enough features were provided to enable almost any realistic scenario on the machine to be simulated.

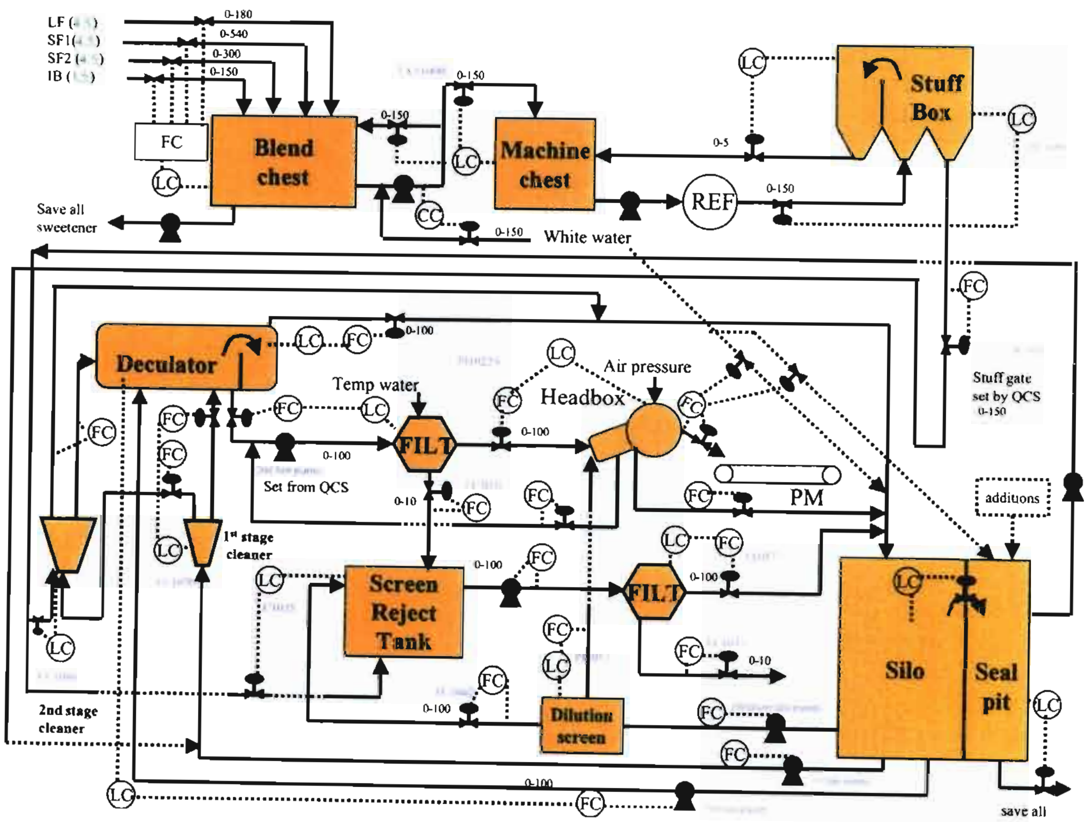

Figure 4.5: Model

4.1.2.3 Modelling devices

– Mixing, dilution and separation steps

Three unit operations are modelled in the current system: mixing, dilution, and separation. Model parameters are used to vary the action of the operations.

- Mixing is the simplest of the operations described here. The purpose of a mixing block is to combine two streams into a single stream. Computationally, this is done by simply adding a new cell in the model with two streams (or more) as an input and the combined stream will appear as one of the four outputs of the cell. The volume of the tank must be set quite small with a fast response.

- The dilution operation has two inputs and one output as well. The two inputs consist of the stream to be diluted and the dilution water. The output consists of the diluted pulp stream. Flow from the second input stream is added to the first input stream to reach the desired consistency, which is specified by a parameter (the consistency setpoint of the consistency controller). If the consistency of the pulp stream is already lower than the target consistency, no further dilution (and no thickening) occurs. On the plant, 'white water' with very low consistency, is used to lower pulp stream consistencies to the desired setpoints.

- The separation processes used include both screens and cleaners. Here is the description of a unit operation that can be used to model both of these processes. In this operation, an input stream is split into two streams: an accepted stream, which ideally contains all of the fibres and none of the contaminants, and a rejected stream, which ideally contains all of the contaminants. Since we are not interested in the different fibre consistencies but only in the total consistency of the pulp, the efficiency of the separation is determined by specifying the fraction of the input flow that will leave in the accepted stream.

– Overflows

Apart from these basic modelling concepts, the creation of the stock preparation area model required several tricks to represent the different units. On the flowsheet (figure 4.5), there are two overflow tanks: the stuff box and the deculator. On the model, those have been simulated by setting two level controllers on the same tank, one cascaded on the input flow controller (LC_1), the second one cascaded on the output flow controller recycled (LC_2) (figure 4.6).

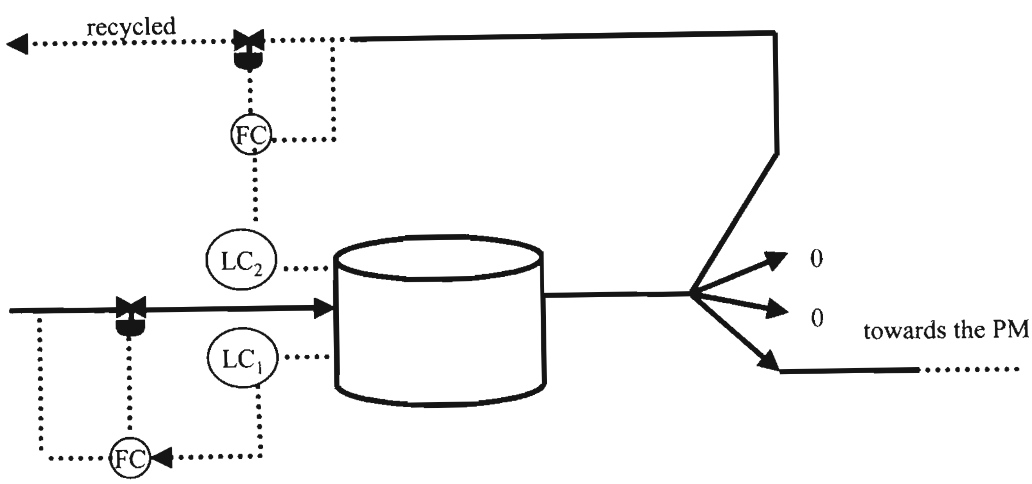

Figure 4.6: Overflow model

In order to represent the overflow situation, the level controller setpoint (LC_2) of the recycled flow must be set lower than the input level controller setpoint (LC_1) in such a way that if the level is in-between those two level setpoints, the action of the LC_2 controller will be to open the valve of the recycled flow whilst LC_1 will also be opening. A smaller recycle flow limit means that LC_1 can always keep up.

- <u>Separation processes</u>

When creating the model, several flows were neglected because of their small impact on the system. The separation processes (filters, screeners, cleaners) are modelled using a level controller cascaded onto a flow controller on the accepted flow and on the rejected flow. Each of these flow controllers use a fixed percentage of the level controller demand, the volume of the cell being set really small, i.e. the time constant fast, as there are no dynamics involved in these vessels (figure 4.7).

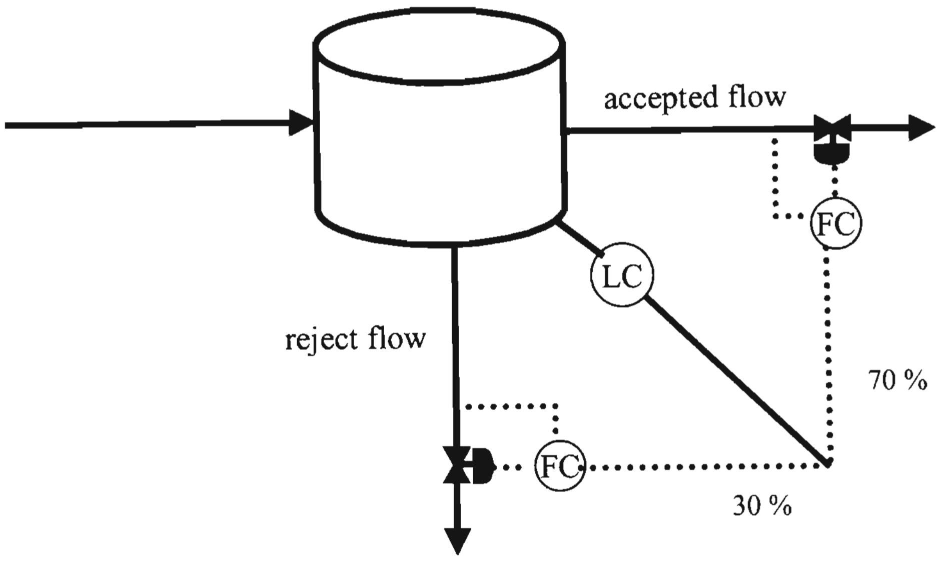

Figure 4.7: Separation process model for filters, cleaners and screens

– Headbox

The headbox level is controlled by a pressure controller, but in the model no pressure measurement is considered. Therefore, a level controller using the input flow was introduced. Besides this, the paper machine itself was not represented in the model. There is no cell modelling a paper machine, but it was necessary for the mass balance of the system to take into account the white water drawn by suction forces out of the mat on the wire of the paper machine. To achieve this, a flow controller was set on the flow from the headbox to the paper machine. This flow controller demand was then cascaded onto the setpoint of a white water flow controller to return the equivalent volume of white water to the silo (see figure 4.5). Because of these extra devices, the number of controllers increased substantially, some of them being real ones (in which case the actual controller settings from the plant were kept), some were virtual ones for the purpose of modelling the process (these were tuned really tightly to give a fast response).

– Supply stock consistency

Figure 4.8 represents a zoom on the blend chest area from figure 4.5.

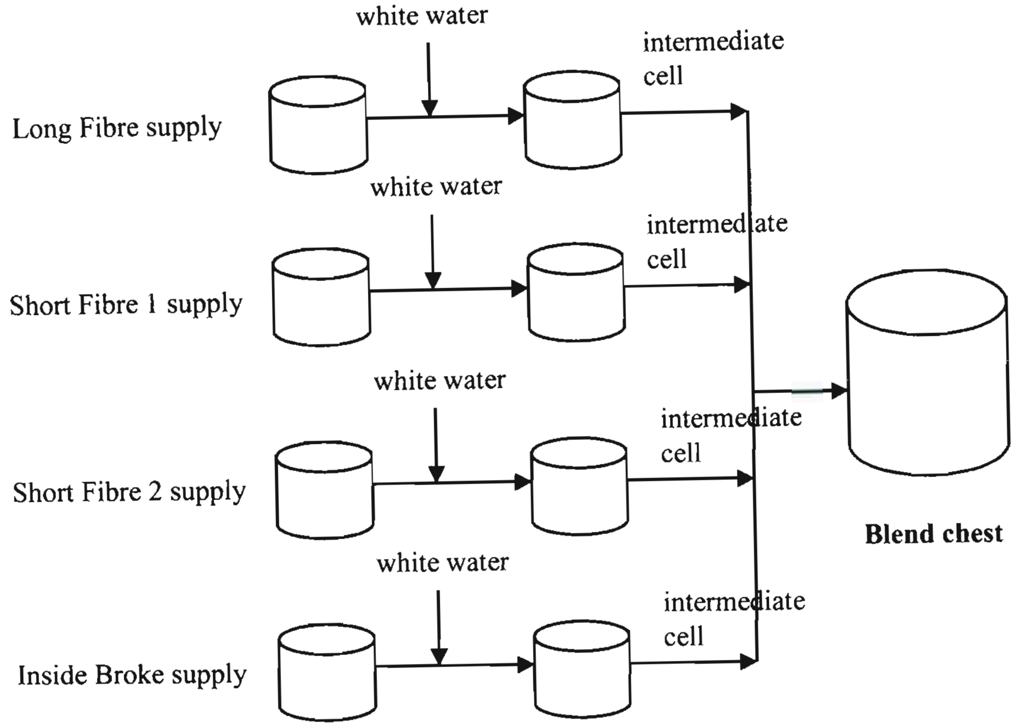

Figure 4.8: Blend chest stock preparation

With the feed arrangement to the blend chest shown in figure 4.8, the concentration of the original supply stocks is unknown; therefore a first order filter with a smoothing factor α was used to estimate the consistency backwards from the value after the consistency controller dilution. The smoothing used in the program is a single-exponential smoothing filter. If y_i is the input and v_i the output smoothed signal, the equation can be written in the discrete form:

$$v_i = \alpha y_i + (1-\alpha) v_{i-1}$$
$$0 \le \alpha \le 1$$

α being the smoothing factor (if α=0 there is no response, if α=1 there is no filtering).

In the system, every single flow is flow-controlled. In reality, some of the valves are directly manipulated by other controllers, e.g. the level controller acting directly on the valve. In that case the time constant of the flow-valve response must be set really short so that flow responds virtually as quickly as the valve moves. Finally, thanks to this structure the flows are constrained to a maximum value (in practical terms, the pipes size limit), which is set by the user.

4.2 Flow diagram of the program

To provide a better understanding of the program structure, a simplified flow diagram is given in figure 4.9. Appendix C gives a listing of the algorithm code created in Matlab.

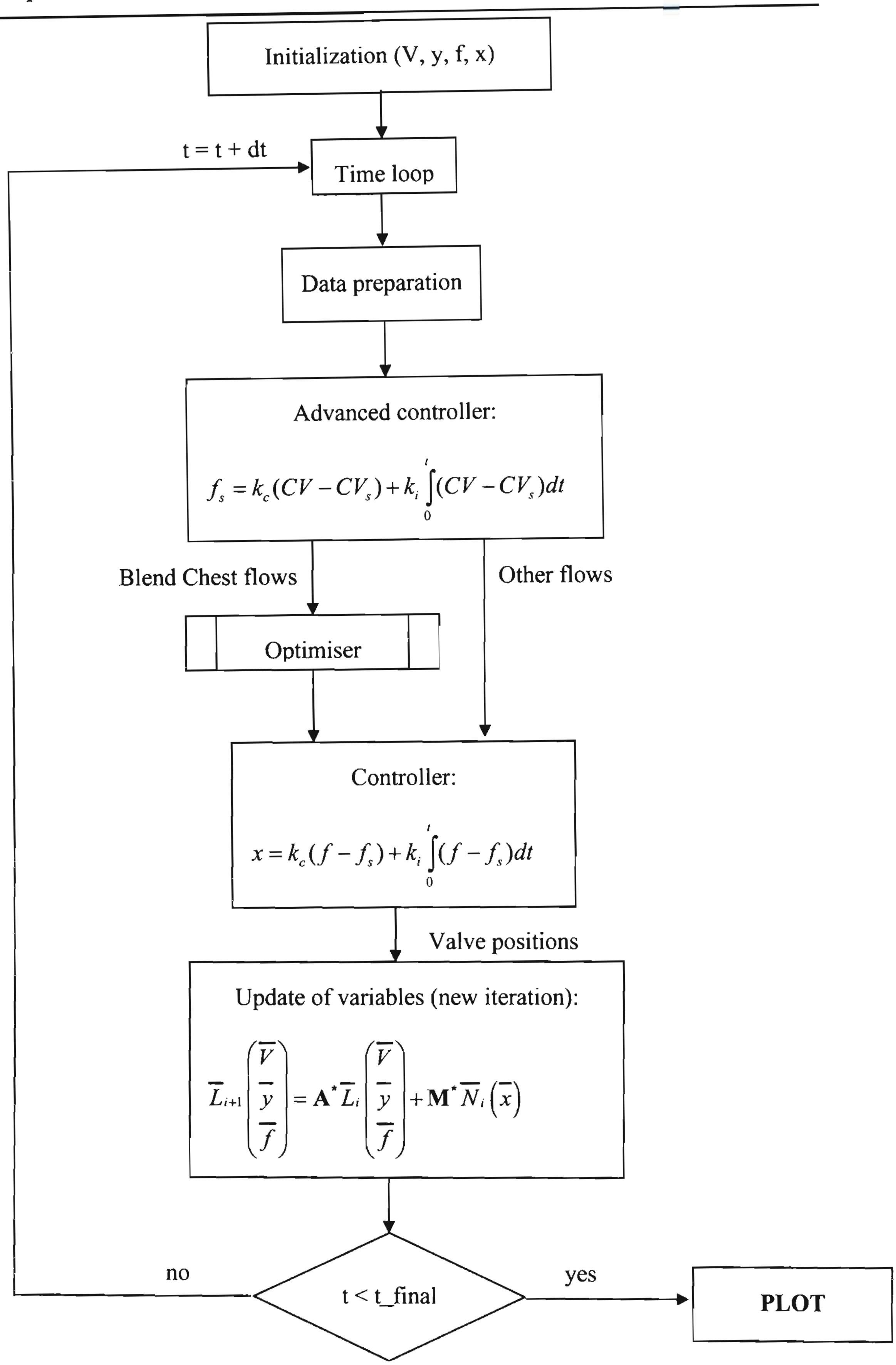

Figure 4.9: Simplified model flow sheet diagram

Chapter 5 : Optimisation

Within the study of optimal control, it is necessary to specify a performance criterion and to use various methods for optimisation. The aim of an optimal policy may be expressed in a "performance criterion" which is a scalar measure of performance but which may be a function of all the state variables and control variables of the system. The "optimal policy" will be that which causes the function to be either a maximum or a minimum, normally under conditions, such as constraints on the controlled and manipulated variables. The selection of the performance criterion, or index, is thus critical to the concept of optimal control.

A solution to an optimisation problem specifies the values of the decision variables and, therefore those of the objective function. It is referred to as feasible when it satisfies all of the constraints. It will be qualified as optimal if it is feasible and if it provides the best value for the objective function. It can be noted that several optimal solutions can exist for the same problem. A solution can also be near-optimal, which means that it is feasible and also provides a superior objective function value, but not necessarily the best.

5.1 **Blend Chest constraint problem**

Figure 5.1 zooms on the blend chest of the entire process represented in figure 4.5.

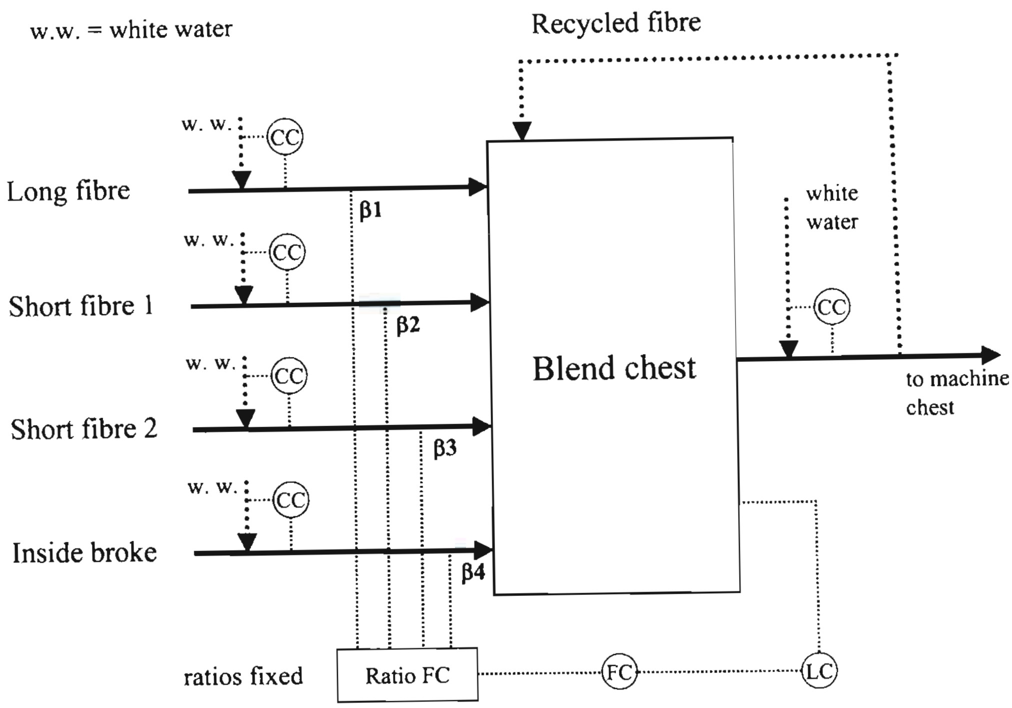

Figure 5.1: Mondi's blend chest control system

The blend chest has 5 inputs (4 feed flows + 1 recycled flow from its output flow after consistency correction) and 1 output to the machine chest.

- <u>Long fibres (LF)</u>: Wood is divided into two major categories: hardwood and softwood. Softwood is the one consisting essentially of long fibres. Its consistency is around 5 % ($^m/_m$ wood fibres in water).
- <u>Short fibres (SF)</u>: Hardwood is used for very short fibres. Its consistency is around 4.5 % ($^m/_m$ wood fibres in water).
- <u>Inside broke (IB)</u>: Inside broke is made essentially of the rejects of the paper machine and paper recycled by the public. It is highly diluted with white water and has a lower consistency of 2.5 % ($^m/_m$ wood fibres in water).
- <u>Recycled stream</u>: Stream returning from the output of the blend chest after consistency correction, to ensure a good uniformity of mixing.

'White water' (represented figure 5.1 by w.w.) is a given name to the water re-circulating in the system. It has a very low consistency (0.2 %) as it is composed of water and small quantities of pulp from the different units. It is used through the system to dilute the pulp flows.

The blend chest ensures a good uniformity mixing of the components so that the concentration corresponds to the demand. This operation is really important for the consistency, as no consistency control occurs further down in the process.

The controls involved in the blend chest consist of:

- consistency controllers (CC) on the feed streams. Their design allows control only if the supplied consistency is above setpoint. Control is achieved by addition of white water. If the value is below setpoint, no control occurs (figure 5.2).

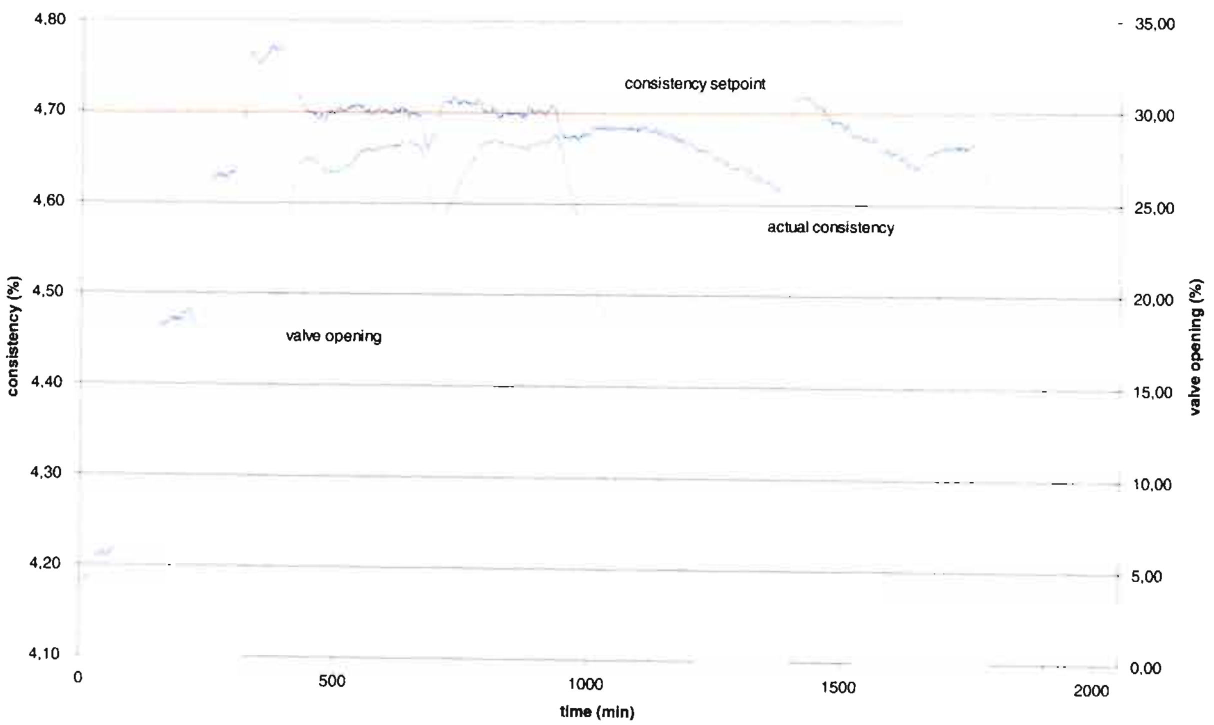

Figure 5.2: Short fiber refiner consistency controllers

Control occurs when the white water valve (valve opening) opens, i.e. when the consistency (actual consistency) goes above setpoint (consistency setpoint). Graphs of other consistency controllers on the feed flows are provided in appendix B.3 and B.4.

- level controller (LC) on the blend chest cascaded onto a flow controller (FC), which is ratioed onto the 4 feed stream flow controllers (FC) (figure 5.1). The ratios β_i of the different feed streams are fixed by the paper recipe. Flows must then be computed from:

$$f_i c_i = \beta_i \sum_{j=1}^{4} f_j c_j \qquad \textbf{(5-1)}$$

- consistency controller (CC) on the output flow of the blend chest to control the final consistency by addition of white water (appendix B.2).
- flow controllers on the feed streams (FC) (appendix B.1).

The blend chest control system has the structure of a feedback controller. It looks at the controlled variable, compares it to the setpoint, and changes consequently the manipulated variables to drive the controlled variable back to the desired value (figure 5.3).

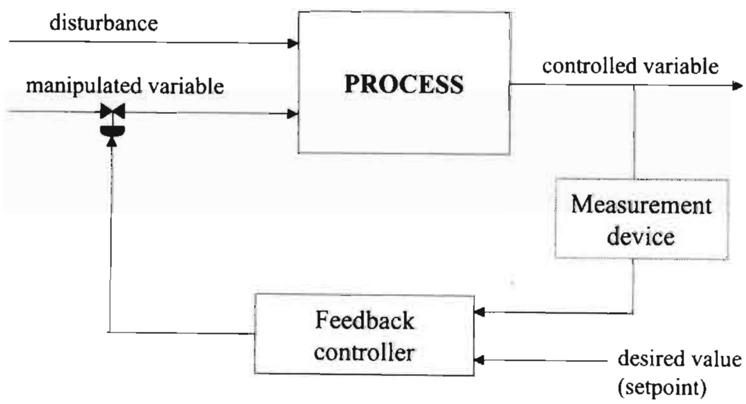

Figure 5.3: Feedback control

A study of the controllers involved in the blend chest area revealed that they were well tuned (figure 5.4). The graph represents the flow control of the inside broke flow to the blend chest. The flow follows its fast-moving setpoint so closely that the two traces overlap in almost all of figure 5.4.

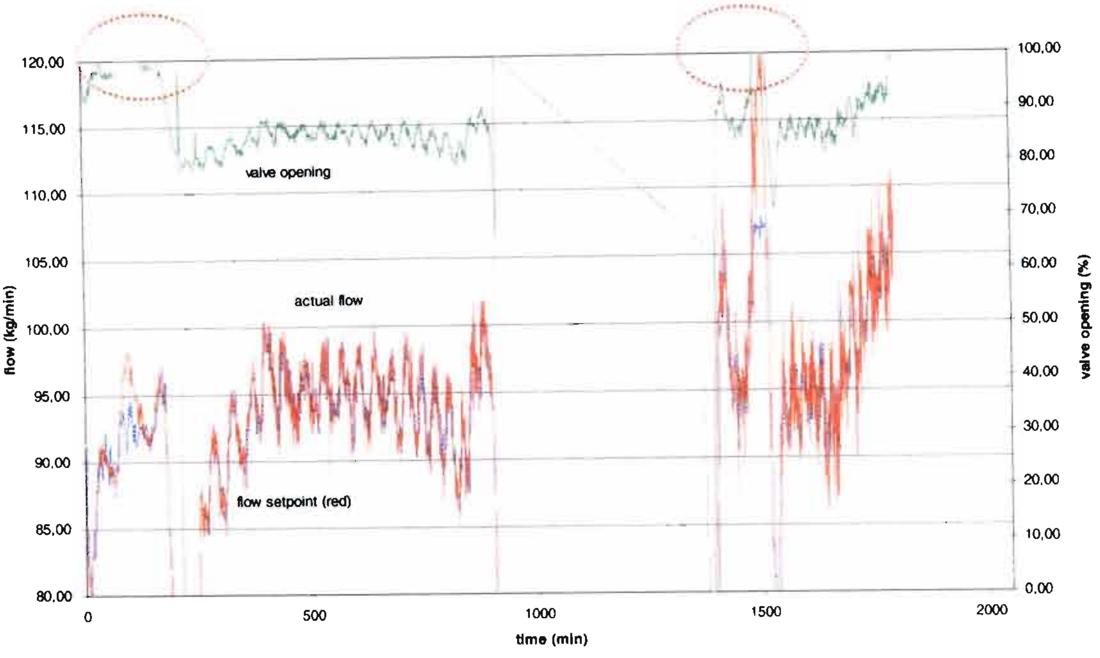

Figure 5.4: Inside broke to blend chest flow controller (constraint problem)

However, on this same graph (figure 5.4), at two different times (circled on the graph), the valve gets fully open and the flow cannot reach its setpoint. This is a good illustration of the constraint problem occuring around the blend chest. This constraint problem is at the origin of the interactions observed further on in the process. There is no automatic device implemented on the system to deal with constrained situations.

As an example of those situations, consider that at one stage, the level of the blend chest becomes too low. Consequently (figure 5.1), the LC asks the FC's for more flow. The global flow controller (FC) sets the setpoint of the different feed FC's according to the ratios β_i. If one of the feed flows (e.g. inside broke in figure 5.4) is constrained, the demand is not satisfied. The effects on the system are immediate. The LC lags in adjusting the level. The consistency in the blend chest is highly affected by this constraint. The average consistency is much higher than expected (consistency of the inside broke flow is lower than the others).

5.2 **The solution: Constrained optimisation**

The system has no automatic way of dealing with constraints. To alleviate this situation, an optimiser was designed. Its objective function compromises between the different controlled variables: ratios, level and consistency. The optimiser is considered as a static openloop feedforward controller (figure 5.5). Its inputs are the desired settings for ratios, blend chest desired consistency and level. An algorithm minimizes an objective function J (eq 5-2) by adjusting the manipulated variables (stock flows and white water flows).

$$\text{Min } J = g(\alpha_{iSP}, L_{SP}, C_{SP})$$

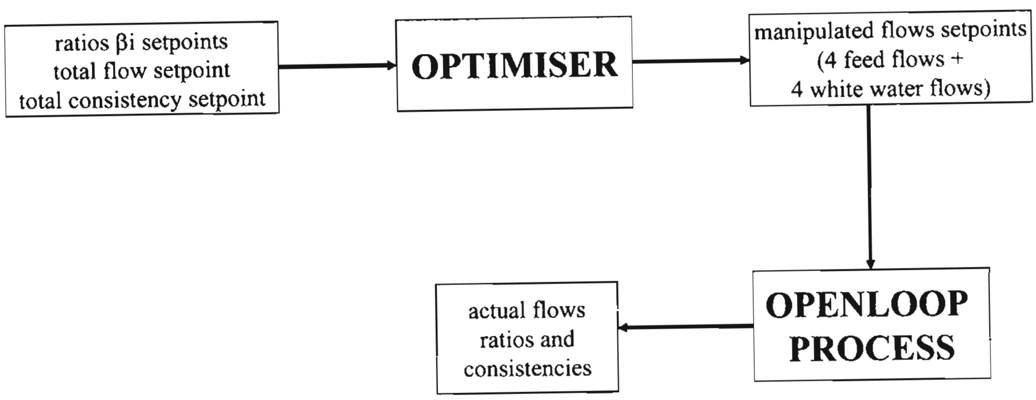

Figure 5.5: Optimiser feedforward setpoint determination

The objective function J (eq 5-2) is a summation of the weighted square of the deviations from setpoints of the optimised variables (ratios, total flow and total consistency). The weighting factors φ_i attributed to each controlled variable defines its relative importance. The algorithm provides a means to identify the demand that needs to be satisfied in priority if all the demands cannot be satisfied at the same time.

$$J = \varphi_1(\beta_1 - \beta_{1SP})^2 + \varphi_2(\beta_2 - \beta_{2SP})^2 + \varphi_3(\beta_3 - \beta_{3SP})^2 + \varphi_4(l - l_{SP})^2 + \varphi_5(c - c_{SP})^2 \qquad \textbf{(5-2)}$$

An order of priority amongst the controlled variables has to be defined. A discussion with the engineers of the plant revealed the order of priority: ratios β_i, total consistency and finally blend chest level.

The ratios must be controlled very cautiously as they play a crucial role in the formation of the paper. A particular set of ratios provides a particular retention necessary for the good formation of the paper (section 2.3). The consistency is important as well as it does not vary much after this mixing step. The level of the blend chest is therefore the variable that can vary within limits to satisfy the other demands.

5.3 Optimisation method

From the various methods described in the literature review (section 3.3), a direct method in the form of a grid search was implemented. The main reasons for this choice were the need of the optimiser to be simple enough to fit on the Distributed Control System (DCS) of the plant. The optimiser manipulates a total of 8 variables (4 feed flows and 4 white water flows (figure 5.1)). A cartesian progression (figure 5.6) was used.

Cartesian progression

For each variable,

$$\left.\begin{array}{l} ev_{new} = ev + 0.1\, ev_{max} \Rightarrow J \\ ev_{new} = ev - 0.1\, ev_{max} \Rightarrow J \end{array}\right\} \text{2*8 tests}$$

$J<Jopt$? $\Rightarrow$ best ev_{new} is kept

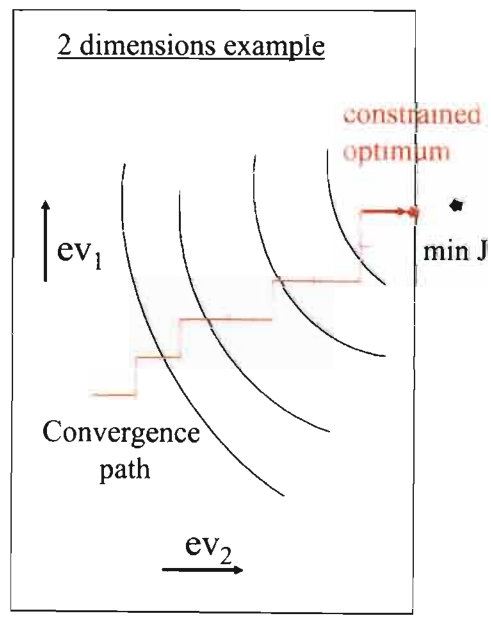

Figure 5.6: Cartesian progression

The direction of the search is orthogonal, it follows the coordinate axes. The step length of the exploration vector *ev* is a fixed fraction of the maximum range (FFMR) of the flow considered (eq 5-3).

$$ev_{new} = ev \pm 0.001 ev_{max} \qquad \textbf{(5-3)}$$

FFMR = 0.001

There are 8 manipulated variables therefore the search is in 8 dimensions. The objective function J is evaluated for a step in every dimension (2 senses per dimension $^{+}/_{-}$) and compared to its recorded minimum value ($J < J_{opt}$?). If the objective function is lower, this variable is kept at its new value and the new exploration vector ev_{new} re-enters the loop (figure 5.6).

Concerning the progression, a two-dimensional example is given in figure 5.6 which emphasises the effect of the constraints on the optimisation. In a bi-dimensional space, the convergence path can only take two directions (ev_1 and ev_2) to approach the optimum. By acting on ev_1 and ev_2, the optimiser improves the objective function until it encounters the constraint of ev_2, which prevents it from accessing the minimum of the function. The final result of the optimisation will be this constrained optimum. This represents the limits of the optimiser, the weighting factor effectively determining which variable must be 'neglected' to satisfy the others.

5.4 Programming

5.4.1 Simplified flow diagram

A simplified flow diagram representing the algorithm structure is provided in figure 5.7. Table 5.1 gives a listing of the variables used.

itermax :	Number maximum of iterations allowed
Jopt_last :	Optimised objective function at the previous iteration
Jopt :	Objective function optimised at the actual iteration
jopt :	Recorded index of the variable that optimised J for the iteration i
topttemp_j :	Recorded value of the variable that optimised J for the iteration i
topt :	Input vector with manipulated variables
topttemp :	Temporary vector for calculations
topt_max :	Input vector with maximum range of manipulated variables
f_s :	Flows setpoint vector
β :	Supply flows ratios
F :	Total flow
C:	Total consistency
FFMR :	Fixed fraction of maximum range for the optimisation search
VC :	Output vector containing controlled variables
CSP :	Controlled variable setpoint
weight :	Weighting factors for the objective function

Table 5.1: Listing abbreviations

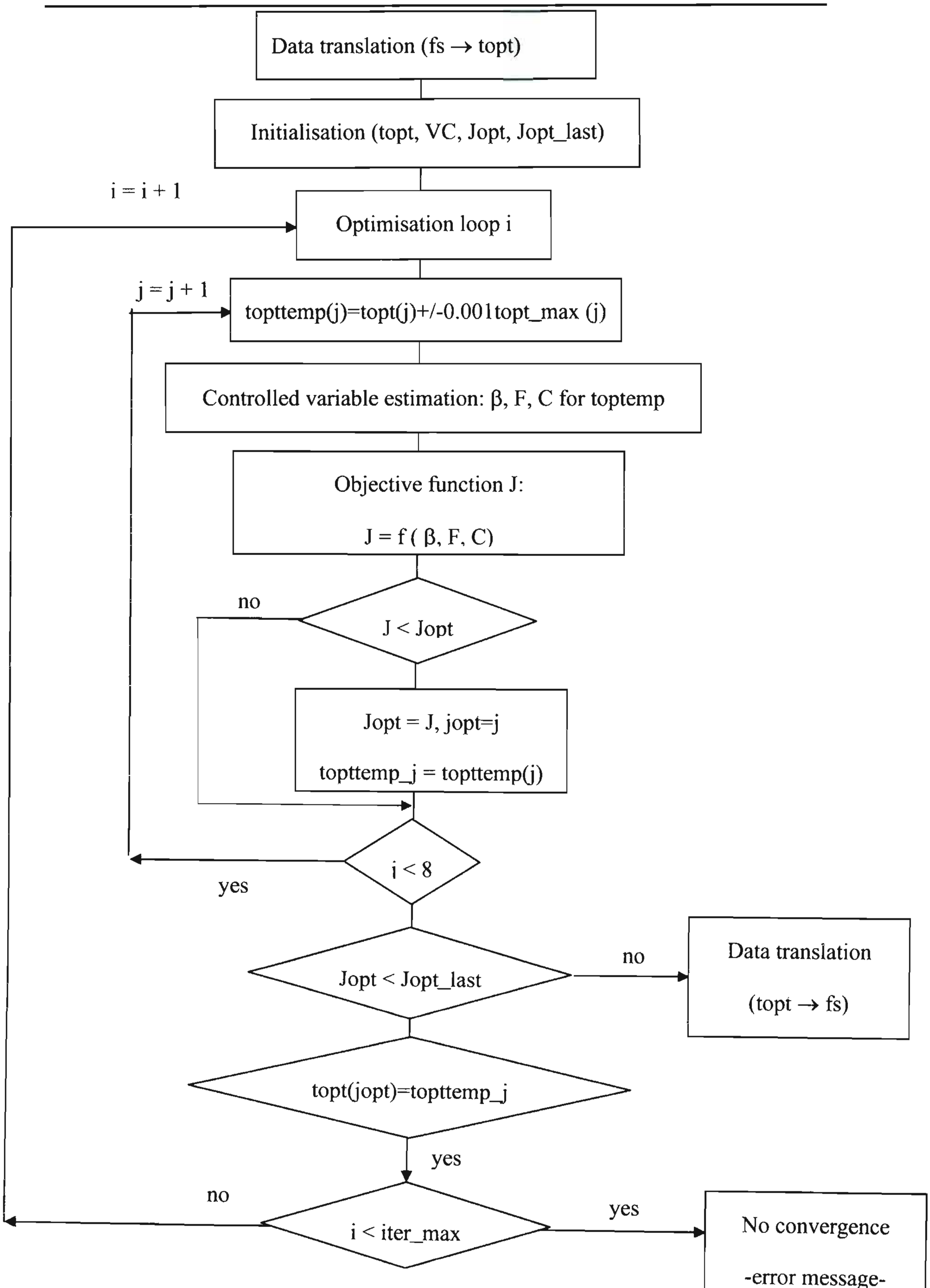

Figure 5.7: Optimiser simplified flow sheet

5.4.2 **Identifying filter**

The supply consistencies of the original feed pulp stocks are unknown. The optimiser algorithm needs to access those properties of the input of the system (pulp streams and white water streams) and to combine them to obtain the final consistency and total flow to the blend chest. The individual pulp flows are mixed with some white water to get the desired consistency (figure 5.8). The intermediate cells do not exist on the plant, they are present for programming purposes. Because the consistencies of the supply stocks are unknown, they were estimated using the measured "intermediate" consistencies following the individual consistency controllers. Using this device, the original pulp consistency can be estimated by the simple calculation:

$$c_{\text{supply}} = \frac{\left(f_{\text{supply}} + f_{w.w.}\right)\cdot c_{\text{int}.cell} - f_{w.w.}\cdot c_{w.w.}}{f_{\text{supply}}} \qquad \textbf{(5-4)}$$

Finally, a first order filter is used to smooth the effects of time lags in these measurements:

$$\begin{aligned} c_i &= \alpha c_{i\,estimation} + (1-\alpha)c_{i\,initialisation} \\ 0 &\le \alpha \le 1 \end{aligned} \qquad \textbf{(5-5)}$$

α being the smoothing factor (if $\alpha=0$ there is no response, if $\alpha=1$ there is no filtering).

The intermediate consistencies were introduced into the objective function to discourage large deviations from their desired operating values (eq 5-6). Their weighting factors were set really low in order not to interfere with the other controlled parameters.

$$\begin{aligned} J = {} & \varphi_1(\beta_1-\beta_{1SP})^2 + \varphi_2(\beta_2-\beta_{2SP})^2 + \varphi_3(\beta_3-\beta_{3SP})^2 + \varphi_4(l-l_{SP})^2 + \varphi_5(c-c_{SP})^2 \\ & +\varphi_6(c_1-c_{1SP})^2 + \varphi_6(c_2-c_{2SP})^2 + \varphi_6(c_3-c_{3SP})^2 + \varphi_6(c_4-c_{4SP})^2 \end{aligned} \qquad \textbf{(5-6)}$$

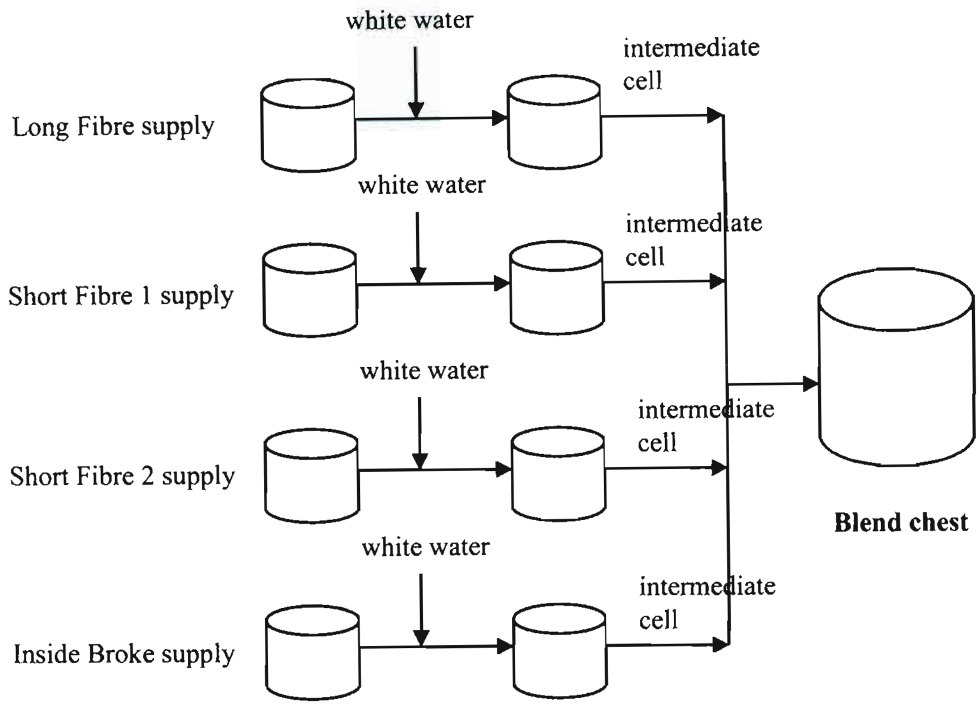

Figure 5.8: Blend chest stock preparation

5.4.3 **Improvements**

The implementation of the optimiser minimised the effects of constraints. Obviously, as successive constraints are met, the ability to meet the setpoints of fibre ratios, total flow and overall consistency becomes increasingly compromised. It is the choice of weights in the objective function (equation 5.2) that determines the order of compromise.

Chapter 6 : Simulation

The first process simulators in the 1960s were very simple and had limited use. The number of different unit operation modules available was small, data banks for physical and thermodynamic data and correlations were limited, and calculation procedures were simple. Because of the limitations of computers at that time, their use was also difficult. Flow sheet simulators of the second generation came onto the market at the beginning of the 1970s. They had large libraries for unit operations and physical properties. The numerical methods used were more efficient than in the earlier simulators, and the man to machine interface started to approach the modern level. Simulation results could also connect to cost and investment data. Third generation code development came in the 1980s. The man to machine interface uses the capabilities of workstations and modern personal computers.

6.1 The graphical user interface (GUI)

A graphical user interface is a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. The major reason for the introduction of the graphical user interface in the present work (figure 6.1) was the improvement of the readability of the model. Where it is possible to plot graphs on Matlab, the graphical user interface links those graphs with the corresponding units. The tuning of the controllers also becomes much easier owing to the scroll buttons on the screen. Finally, the setpoint of a controlled variable can be stepped and its response observed over a specified time interval.

The simulator was developed using the graphical user interface (GUI) of Matlab. Figure 6.1 shows a representation of the simulator as it is when the program starts.

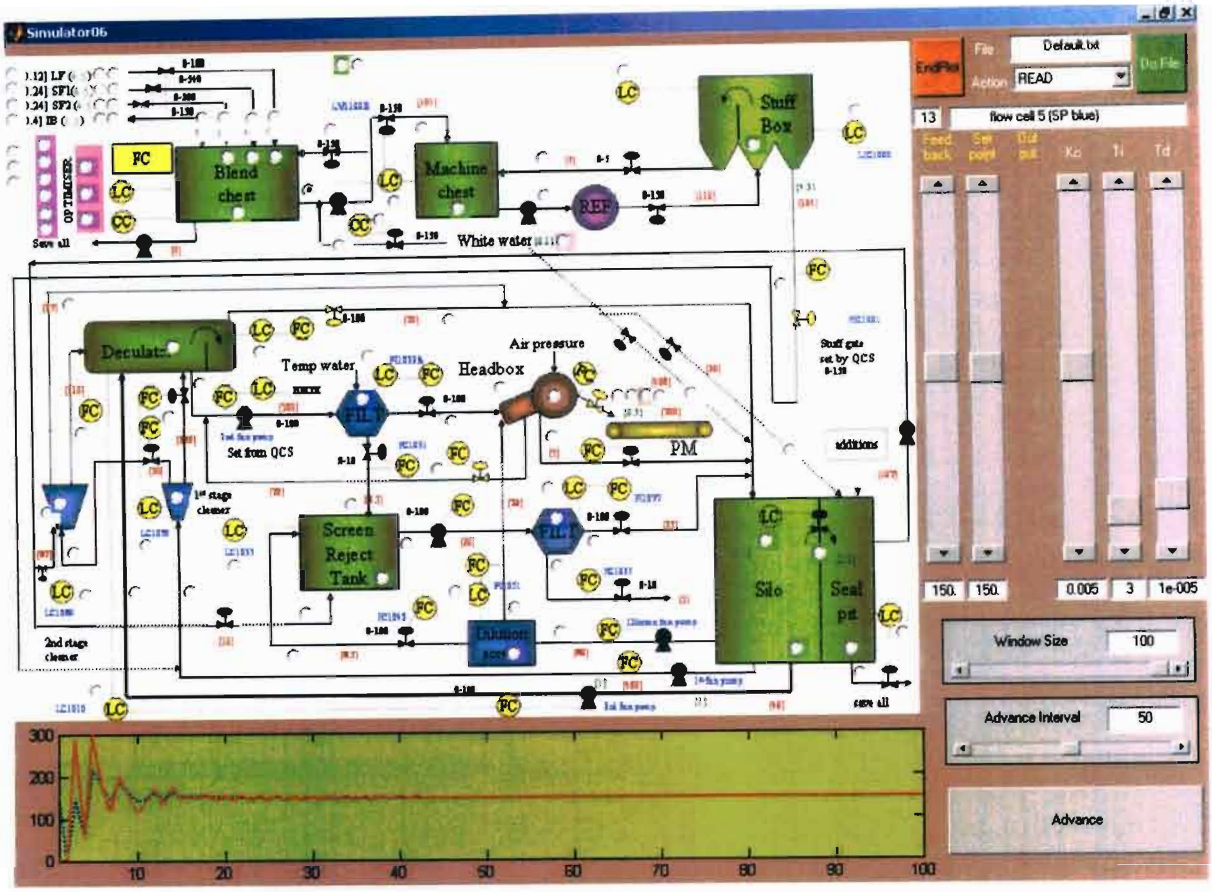

Figure 6.1: Simulator

The main part of the window represents the stock preparation area with all of its units. At the bottom, a graph of the object selected by the user (by clicking on the appropriate 'radio button') is displayed (e.g. the graph in figure 6.1 represents the blend chest output flow). On the right hand side are the value of the object selected (feedback), its setpoint (setpoint), and the tuning applied on its controller parameters (k_c, τ_i and τ_d). The following section concerns the use of the simulator:

- To select the different displays, press the appropriate "Radio Button". Flows are shown near the pipe, concentrations are show on the vessel. Levels are shown next to the level controller.

- Setpoints and controller settings can be adjusted using the slides. Even a controlled variable can be moved but it will be controlled back to its setpoint position over the next few steps.

- The solution does the first "advance" of 10 seconds. To advance again, press "advance". The advance size or window size can be changed using the sliders. To stop and print a final graph, press the red button at the top. To restart a run, the simulation needs to be redone from the beginning.

- At any time, the whole solution and controller settings can be saved by selecting "write" at the top and pressing the green button. The solution always starts with settings from the Default.txt file. So when the settings are right, save to this file and make a backup of it. Once the compilation is running one can restart with any file by giving the file name and setting "read" then pressing the green button.

- Concerning the optimiser, the objective function value is accessible as well as the number of iterations necessary to converge. The weights of the different parameters of the objective function can be changed at any time on the screen.

This simulator provides a better understanding of the phenomena happening in the system, and permits interactive setting of variables.

6.2 GUI design

To design the graphic user interface, several concepts needed to be added to the actual model. A graphical user interface (GUI) is a user interface built with graphical objects: the components of the GUI being items such as buttons, text fields, sliders, and menus. By providing an interface between the user and the application's underlying code, GUIs enable the user to operate the application without knowing the commands that would be required by a command line interface. For this reason, applications that provide GUIs are easier to learn and use than those that are run from the command line.

Matlab implements GUIs as figure windows containing various "uicontrol" objects. Each object must be programmed to perform the action one intends it to do when a user activates the component. The development of the GUI was done using GUIDE, the Matlab graphical user interface development environment.

Creating a GUI involves two basic tasks: laying out the GUI components and programming the GUI components. GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialisation and launching of the GUI. This M-file provides a framework for the implementation of the callbacks (the functions that execute when users activate components in the GUI).

When the GUI is saved or run, GUIDE automatically generates two files:

- A FIG-file, a file with a '.fig' file name extension, which contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

- An M-file, a file with a '.m' file name extension, which contains the functions that run and control the GUI and the callbacks. This file is referred to as the GUI M-file. Note that the M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following section consists of a quick review of the main function of the GUI.

- Data between callbacks can be shared by storing it in the Matlab handles structure. For example, to store data contained in a vector X in the handles structure, a name for the field of the handles structure where you want to store the data is chosen, e.g. handles.x set the field equal to x with the following command:

handles.x = x

- To save the handles structure, the guidata function is used:

guidata(hObject, handles)

Here, hObject is the handle to the object that executes the callback. Note that to save any changes that you make to the handles structure, you must add the command guidata(hObject, handles) following the code that implements the changes.

- To retrieve x in another callback, use the command:

x = handles.x

the data in the handles structure can be accessed in any callback because hObject and handles are input arguments for all of the callbacks generated by GUIDE.

6.3 GUI screen shots

The major advantage of the simulator is that it provides an easy access to data. This is appreciated especially when the response of the system to a perturbation needs to be observed. Perturbations were introduced to the system in the form of setpoint changes for the level, the consistency or the flow.

6.3.1 **Level perturbation response**

The level setpoint of the machine chest is stepped up from 70% to 90%. The reaction of the level to this change is observed over a period of 1000 seconds (figure 6 .2).

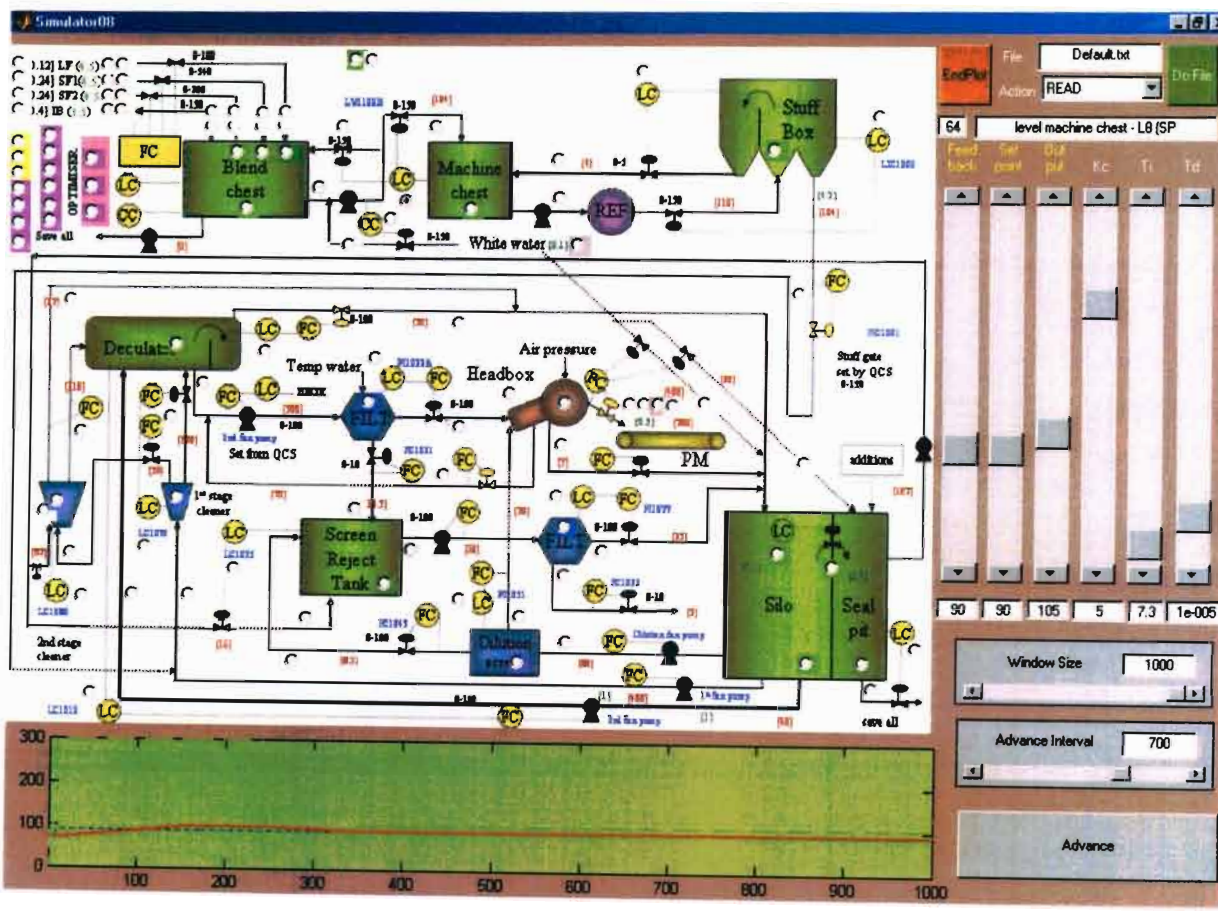

Figure 6.2: Level setpoint step response

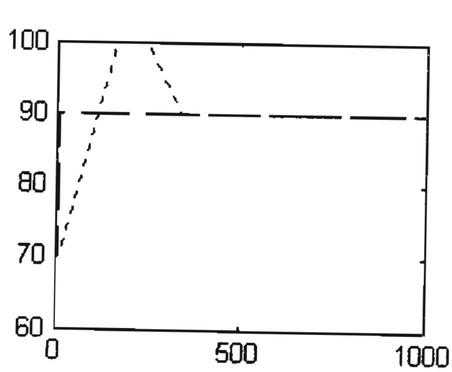

Figure 6.3: Machine chest level response

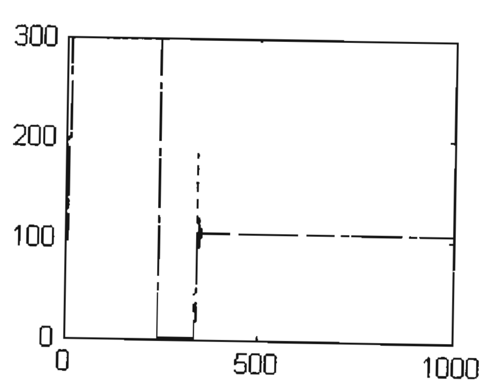

Figure 6.4: Cascaded flow controller response

A "zoom" of the graphs observed on the simulator is provided in figures 6.3 and 6.4. Figure 6.3 shows the response of the level to the perturbation. The response has an overshoot and converges to the setpoint rapidly after 300 seconds. Figure 6.4 represents the output (MV) of the level controller, in other words the flow controller setpoint reaction to the level controller output. This last controller is set really tightly, the flow following its setpoint really closely.

6.3.2 **Flow perturbation response**

The stuff box output flow demand is stepped up from 120 to 130 Ls^{-1}. The response of the stuff box flow is observed over a period of 50 seconds (figure 6.5).

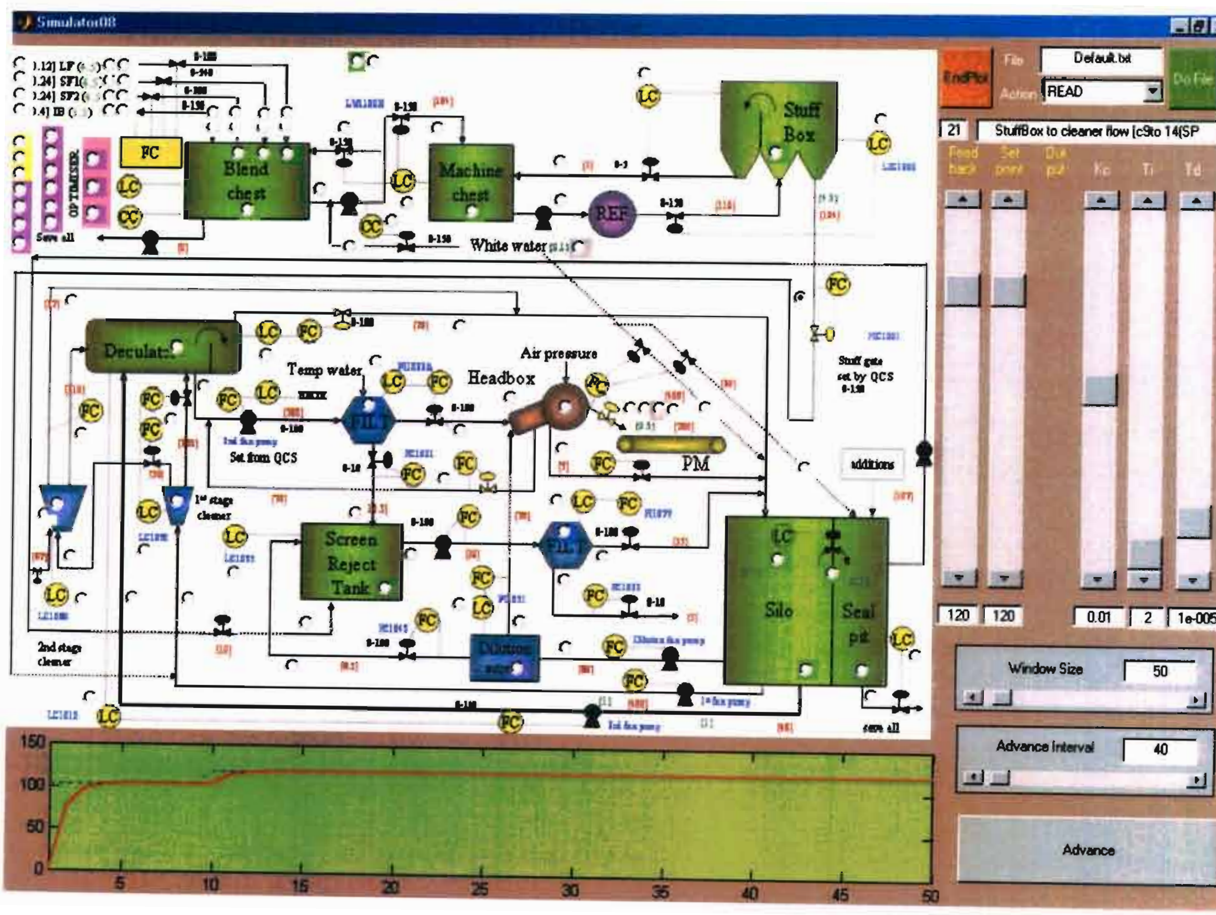

Figure 6.5: Simulator response to a flow perturbation

The response is fast, the flow reaches its setpoint after only 3 seconds.

6.3.3 **Consistency perturbation response**

The consistency setpoint of the flow to the machine chest is stepped down from 3 to 2.5 % and the time horizon is 200 seconds (figure 6.6).

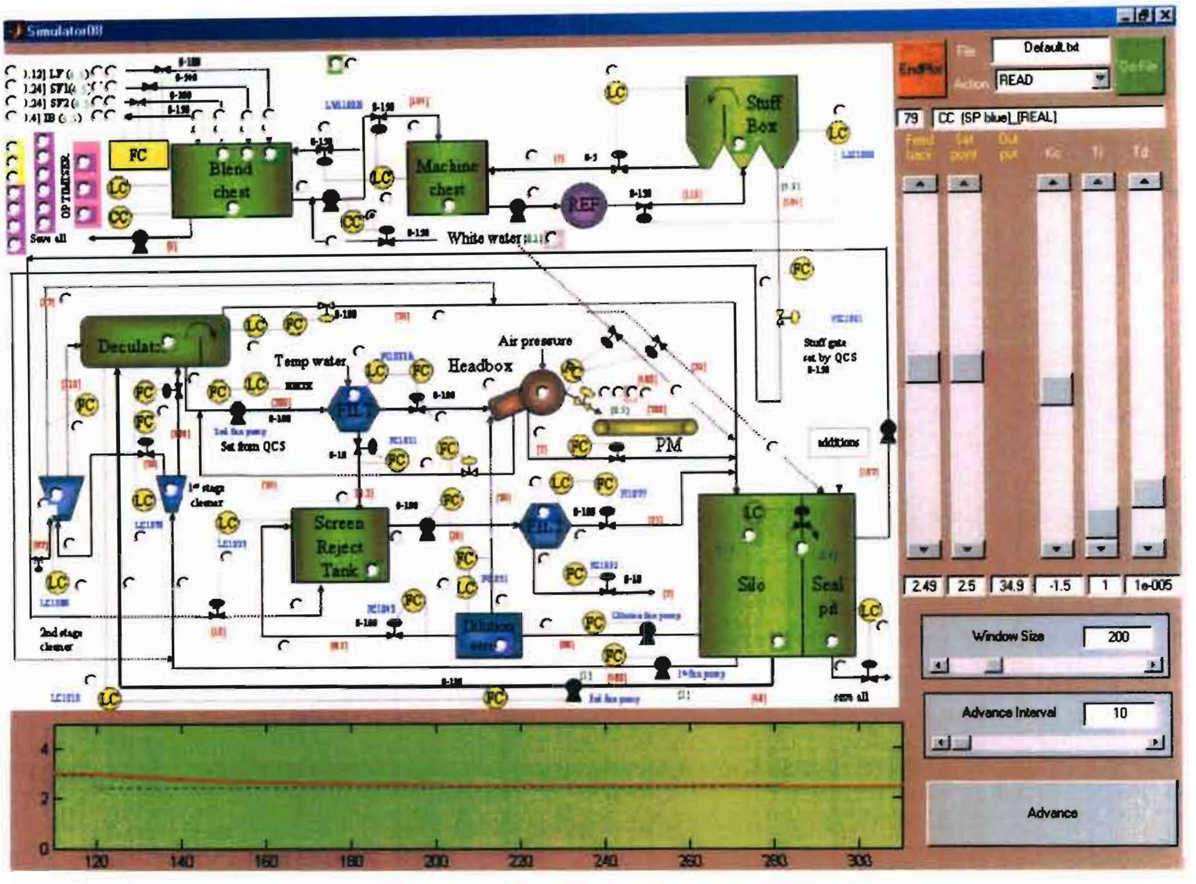

Figure 6.6: Consistency setpoint step response

A zoom of the graphs observed on the simulator is provided figure 6.7 and figure 6.8. Figure 6.7 shows the response of the consistency to the perturbation. The response is fast (200 seconds). Figure 6.8 represents the slave flow controller setpoint reaction which is fast as well. To decrease the consistency in the machine chest the white water flow is increased (figure 6.8).

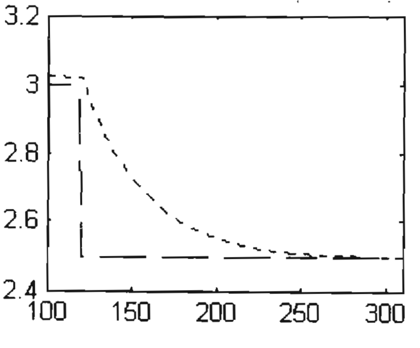

Figure 6.7: Consistency step response

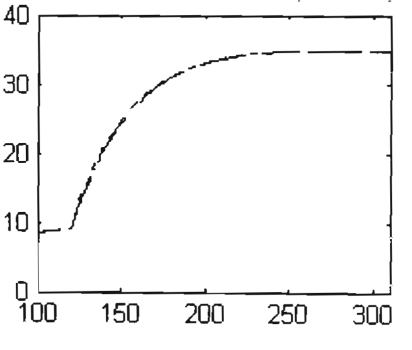

Figure 6.8: Flow controller reaction

Chapter 7 : Results

In this work, an understanding of the controller interactions in the stock preparation section has been developed by detailed dynamic modelling, including all of the existing control loops (chapter 4). Closer consideration of the plant behaviour revealed that some of the input flows were frequently constrained, and that the system had no automatic way of reacting under constrained conditions, causing excessive loss of specification at the paper machine. To deal with these constraints systematically, a static optimiser was created and implemented on the model (chapter 5). Both the model and its optimiser were represented in a simulator, designed with the graphic user interface (GUI) of Matlab (chapter 6). The simulator has then been used to explore control performance over the operating range, leading to the formulation of a set of tuning scenarios and some requirements relative to the limits of the simulator. It is intended that this will provide a basis for on-line implementation of the optimiser.

7.1 Data collected from the plant

In order to have a realistic model, data were collected on the plant. Vessel dimensions were roughly measured but the controllers settings are actual values for one paper grade. The volume of each unit was approximated and level (LC) and consistency controller (CC) tunings were collected on table 7.1. Each controller has an output range (the range of the device manipulated by the controller) and an input range (the range of the feedback parameter that enters the controller).

- For LCs, the input range is in m and the output range (the input range of the cascaded flow controller) is in kg/min.

- For CCs, the input range is in percentage % $^{m}/_{m}$ and the output range (the input range of the cascaded flow controller) is in kg/min.

- For FCs, input and output ranges are in kg/min.

- The k_c and τ_i represent the gain and integral time settings of the PI controllers. Where τ_i is in seconds, k_c represents the fraction of manipulated variable (MV) range per fraction of controlled variable (CV) range.

Unit	Output Range	Input Range	Kc MV range / CV range	Ti s	Volume m3
Machine chest					
LC	0/150 kg/min	100%=5m	5	440	100
CC	0/150 kg/min	%	0.8	9.87	
Blend chest					
LC	0/825 kg/min	100%=5m	3	240	100
Components					
Inside broke FC	0/150 kg/min	0/150 kg/min	0.33	2.35	
SF2 FC	0/300 kg/min	0/300 kg/min	0.45	1.4	
SF1 FC	0/540 kg/min	0/540 kg/min	1	1.4	
LF1 FC	0/180 kg/min	0/180 kg/min	0.2	15	
Inside broke CC		2/5 %	0.5	20	
SF1 CC		3/6 %	0.35	125	
LF1 CC		3/6 %	0.08	5	
Stuff box					
LC	0/100 kg/min	100%=2m	0.6	3	10
Stuff box to silo flow					
FC	0/150 kg/min	0/150 kg/min	0.95	9.5	2
Silo					
LC	0/100 kg/min	100%=2m	1.5	50	30
Deculator					
LC	0/100 kg/min	100%=1m	1	110	15
Headbox					
LC	0/100 kg/min	100%=0.5m	1.25	30	3
Screen Reject tank					
LC	0/100 kg/min	100%=2.5m	3.5	334	6

Table 7.1: Unit dimensions and controller settings

Plant data were collected and manipulated to get representative graphs. The following sequence of graphs (figure 7.1, 7.2, 7.3) represents the actual evolution of three controlled variables (flow, level and consistency). On each graph are represented the actual observed value (flow for a FC, level for a LC, and consistency for a CC), the setpoint and the manipulated variable. The time interval [900-1400] is not important for the study, the unusual shape of the response is due to an error in the data measurement on the plant. The different controllers are nicely tuned.

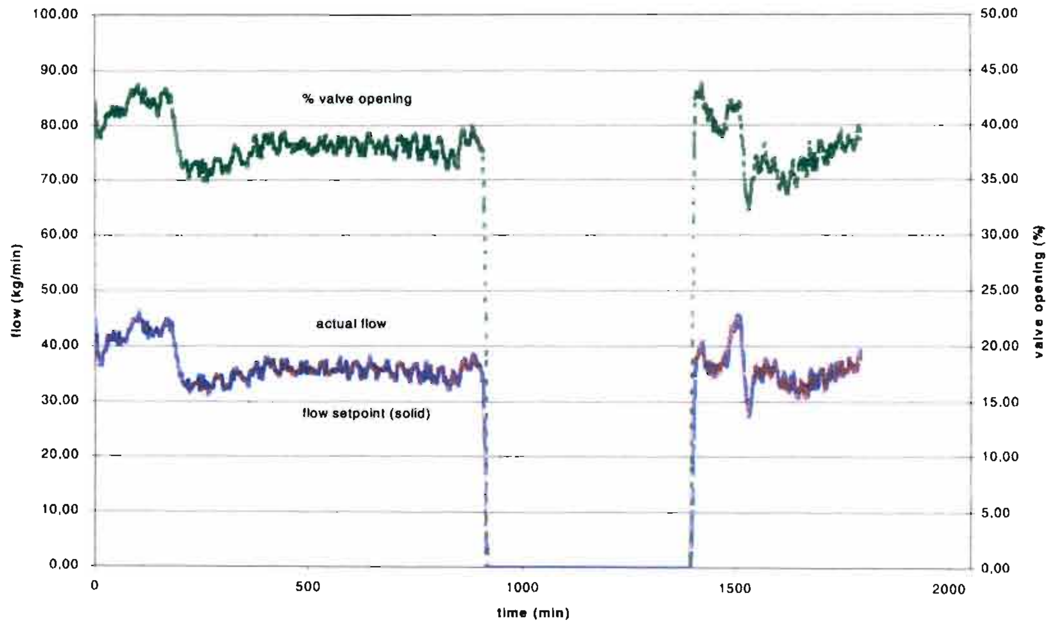

Figure 7.1: Short fibre flow control

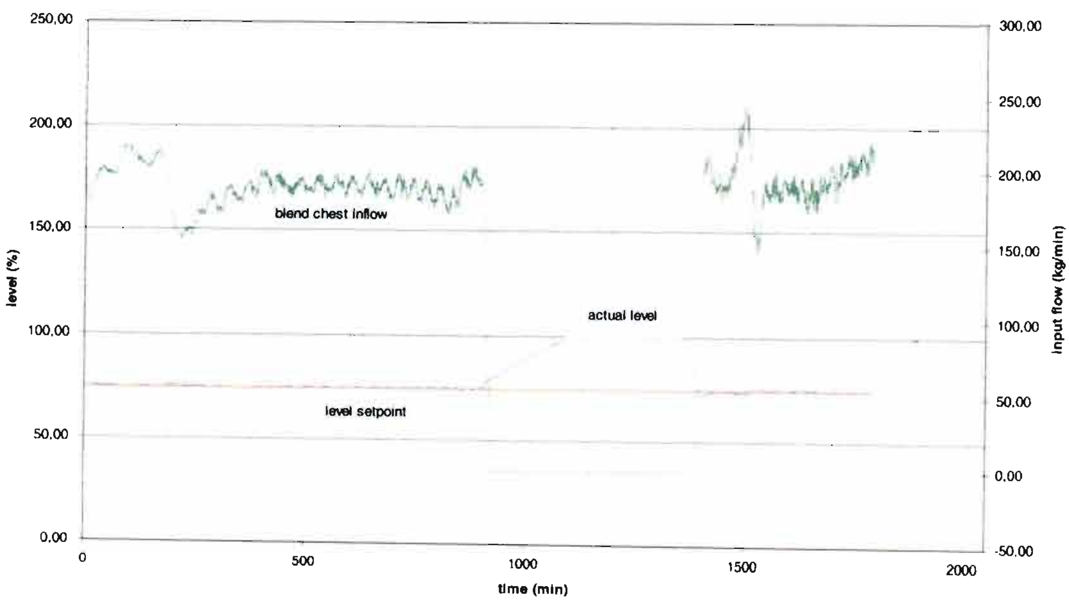

Figure 7.2: Blend chest level control

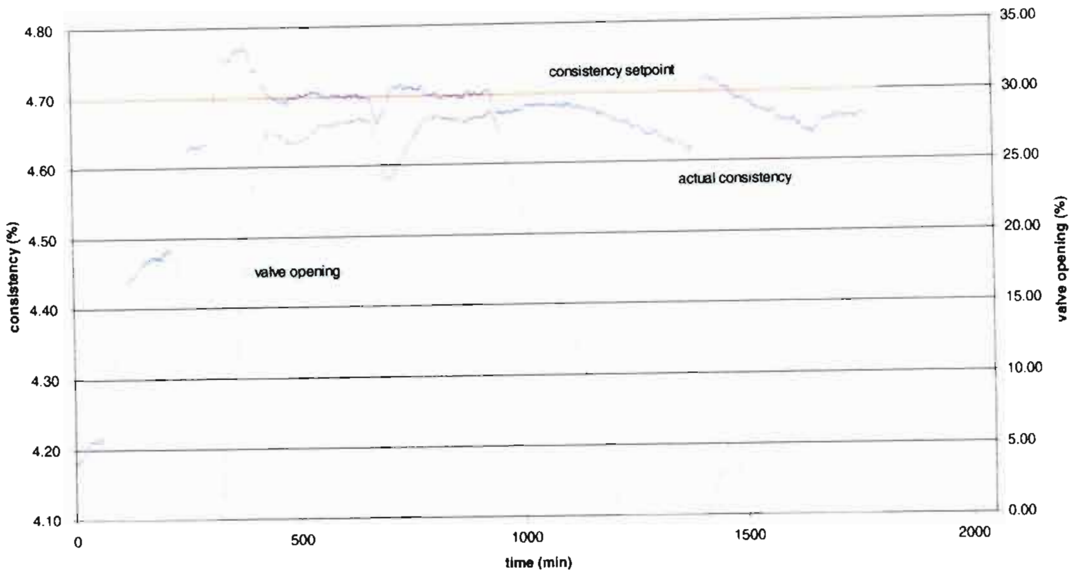

Figure 7.3: Short fiber refiner consistency control

7.2 <u>Tuning of the controllers</u>

7.2.1 **Tuning of cascaded PI controllers**

The tuning of the different controllers was done according to the trial and error tuning technique (gain is set first by a step response analysis, then integral action). Particular attention was paid however to the advanced controllers. The slave controller is tuned first and then only the master controller can be tuned.

7.2.2 **Interaction between controllers**

During the tuning stage, a number of problems were encountered due to the presence of interactions. For instance, good blend chest level control was really difficult to achieve. Most of the vessels present act as pure integrators therefore the controller can cycle without converging, or only after a really long time. Integral action degrades the dynamic response of the control loop, adding to the natural integration and making the closed loop more oscillatory. In that case, very little/zero integral action is needed in controllers so that the proportional gain will dampen oscillations. To solve this problem an increased range was allowed for the τ_i value so that a big value could be implemented, cancelling therefore the integral term of the controller. As a consequence, the level controller still oscillates but converges rapidly due to the damping effect of the proportional action. (figures 7.4 and 7.5).

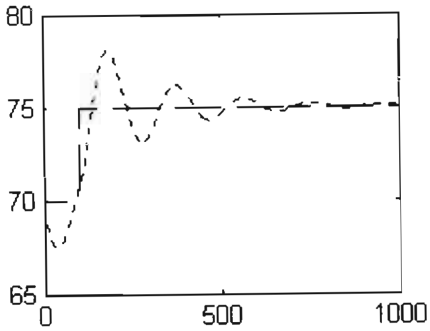

Figure 7.4: Blend chest level

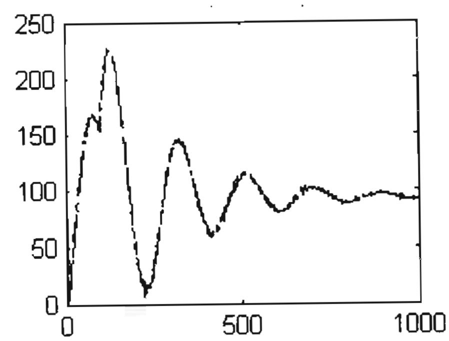

Figure 7.5: Flow controller reaction

7.3 Simulator : Case study

To determine the limits of the simulator, a series of experiments was carried out. Different types of perturbations were introduced in the system and simulator responses were analysed.

7.3.1 Parameters of the study

This section details the important parameters of the study, parameters that influence the behaviour of the simulator. As presented in section 5.2, the simulator uses a static openloop feedforward controller. This controller optimises an objective function J:

$$J = \varphi_1(\beta_1 - \beta_{1SP})^2 + \varphi_2(\beta_2 - \beta_{2SP})^2 + \varphi_3(\beta_3 - \beta_{3SP})^2 + \varphi_4(l - l_{SP})^2 + \varphi_5(c - c_{SP})^2 \quad \textbf{(5-2)}$$

In the algorithm, the form of the objective function was modified for programming purposes (section 5.4). On the plant, the blend chest level controller supervises a flow controller. In the optimiser, the total flow is kept as close as possible to the output of the level controller (F_{SP}). However, the level variations are still recorded and introduced in the objective function via the total flow weighting parameter $\varphi_4^{'}$ to strengthen tracking of F_{SP} when level deviates far from L_{SP} (equation 7-1).

$$J = \varphi_1(\beta_1 - \beta_{1SP})^2 + \varphi_2(\beta_2 - \beta_{2SP})^2 + \varphi_3(\beta_3 - \beta_{3SP})^2 + \varphi_4^{'}(F - F_{SP})^2 + \varphi_5(c - c_{SP})^2$$
$$\varphi_4^{'} = \varphi_4^{'}(1 + \varphi_6(L - L_{SP})^2) \quad \textbf{(7-1)}$$

Another addition to the algorithm was the introduction of the intermediate consistencies in the formulation of the objective function (section 5.4.3). These consistencies correspond to those of the supply flows after dilution with white water. The consistencies are observed to ensure that the optimiser is coherent and that the consistencies remain physically acceptable.

The optimisation algorithm is important for convergence. In order to find the best solution (the solution that optimises the objective function J), the algorithm searches between different combinations of the manipulated variables in the space (section 5.3). The variable that defines this search is the "fixed fraction of maximum range" (FFMR) (equation 7-2).

$$topttemp(i) = topt(i) \pm FFMR \cdot topt_{max}(i) \tag{7-2}$$

This fixed fraction is an important criterion in the convergence of the optimiser.

Another important criterion is the number maximum of iterations *itermax*.

To simplify the study, the weighting parameters were set equal for the three ratios β, their influence on the process being similar.

Table 7.1 gathers all of the data relative to this study and described in this section.

Variable	Setpoint	Weighting factors	Values
Total flow	100 $L.s^{-1}$	ω_1 (ratios)	10
Total consistency	4.1 $\%^{m}/_{m}$	ω_2 (flow)	0.01
Ratio LF	11 %	ω_3 (level)	0
Ratio SF1	17 %	ω_4 (total consistency)	600
Ratio SF2	17 %	ω_5 (intermediate consistencies)	1
Algorithm parameters		**Tuning**	
FFMR		0.0001	
Itermax		20000	

Table 7.2: Initial values of the parameters of the study

The variable setpoints were set according to plant data and the algortithm parameter values are the results of some tuning experiments. The algorithm form of the objective function is given in equation 7-3.

$$J = \omega_1 \sum_i (\beta_i - \beta_{i_{SP}})^2 + \omega_2 (F - F_{SP})^2 + \omega_4 (c - c_{SP})^2 + \omega_5 \sum_i (c_i - c_{i_{SP}})^2$$
$$\omega_2 = \omega_2 (1 + \omega_3 (L - L_{SP})^2) \qquad \textbf{(7-3)}$$

Weighting parameters ω_i are set according to the priority order defined in section 5.2. It is important to notice that the weighting parameters differ consequently from one another (table 7.1). Note that the weights in Table 7.2 apply to square deviations, eg. a weight ratio of 0.01 determines a 0.1 ratio between the corresponding deviations.

The optimiser is actually choosing nine variables: 3 ratios β_i, the total flow, the total consistency and the 4 intermediate flow consistencies. To control these variables, the optimiser supervises 8 manipulated variables (4 white water flows and 4 feed flows).

The steady state simulation results, with the settings given in table 7.1, are displayed in figure 7.6.

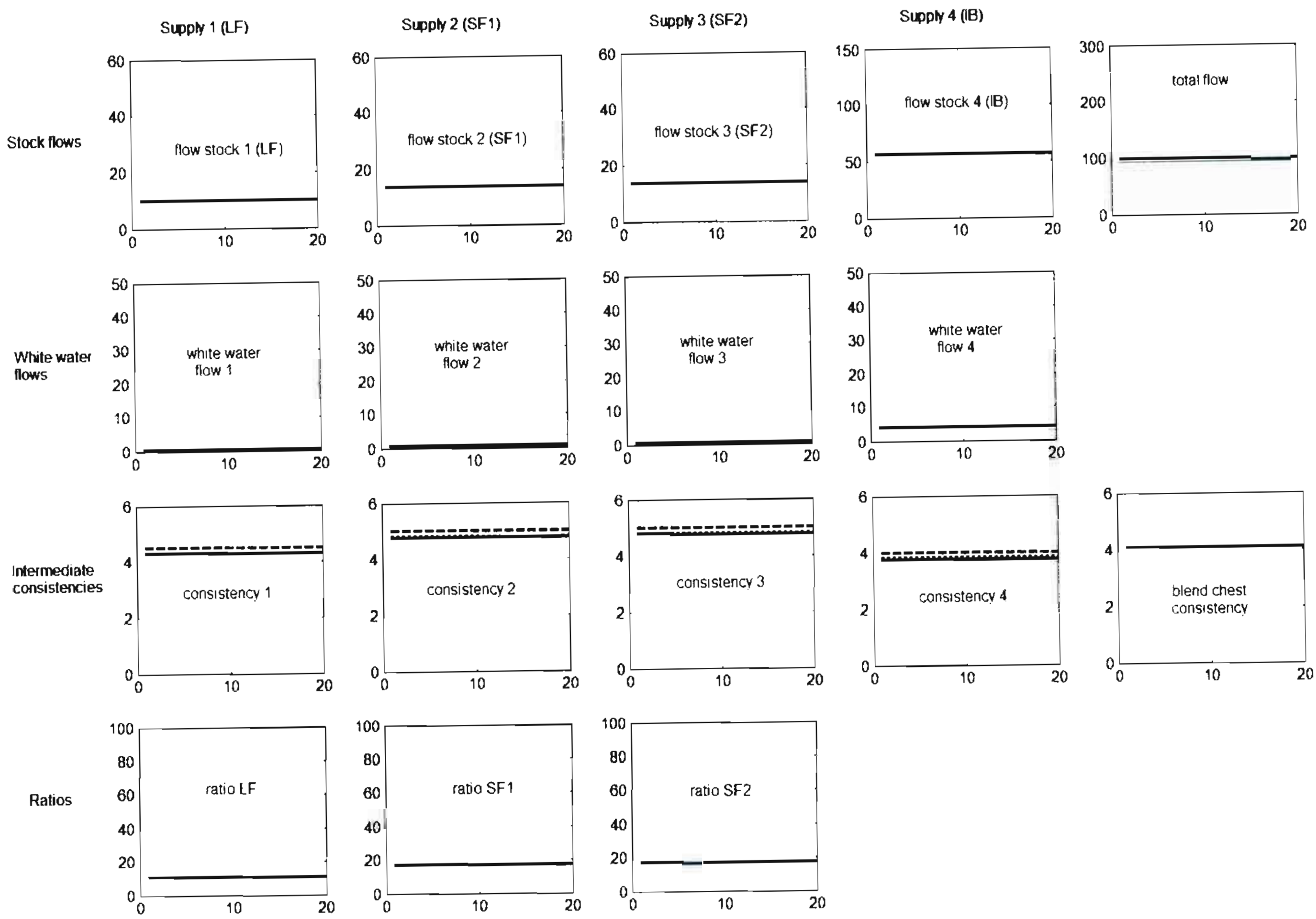

Figure 7.6: Steady state simulation

On these graphs (figure 7.6), the actual measurement is a solid line, the setpoint is a dotted line and for the intermediate consistencies, the upper limit is a dashed line.

- in the first row are the four stock flows (manipulated variables). The vertical scale represents their range of variation (Ls^{-1}) and the horizontal one the time (s). For this specific simulation (figure 7.6), none of them is constrained.

- in the second row are the four white water flows (manipulated variables). The vertical scale represents their range of variation (Ls^{-1}) and the horizontal one the time (s). For this specific simulation (figure 7.6), none of them is constrained.

- the last graph of the first row represents the total flow to the blend chest (controlled variable), summation of stock flows and white water flows, and its setpoint. The demand is satisfied as the flow line overlaps the setpoint line.

- in the third row are the four intermediate consistencies, their setpoint, and the maximum consistency achievable (the constraints). These parameters are not really controlled because their weighting factors in the objective function are low. They are observed and the optimiser tries to set them as close as possible to their demand without interfering with the major demands. The demands are satisfied as the flow lines overlap the setpoint lines.
- the last graph of the third row represents the consistency of the blend chest (controlled variable), mix of the intermediate consistencies, and its setpoint. The demand is satisfied, the actual consistency line overlapping the setpoint line.
- in the fourth row, the three sets of ratios (controlled variables) and their setpoints are shown. The demands are satisfied.

The steady state graphs represented in figure 7.6 were obtained for one typical plant situation (table 7.1) and are considered as a reference in the study.

In order to determine the limits of the simulator, some perturbations were introduced in the model and responses were observed on different graphs. The perturbations concerned the main controlled variables (total flow, blend chest consistency and stock ratios). The objective was to observe the reaction of the simulator under constrained and unconstrained situations.

7.3.2 **Unconstrained situations**

The optimiser was created because the system had no automatic way of dealing with constraints. Therefore, in an unconstrained situation, the optimiser is inactive. Different perturbations were introduced in the system: the consistency, the total flow and finally the ratios were ramped independently, one at a time.

7.3.2.1 Consistency ramp

In this experiment, the total consistency is ramped from 2.5 to 3.5%$^{m}/_{m}$ and the simulation results are represented in figure 7.7.

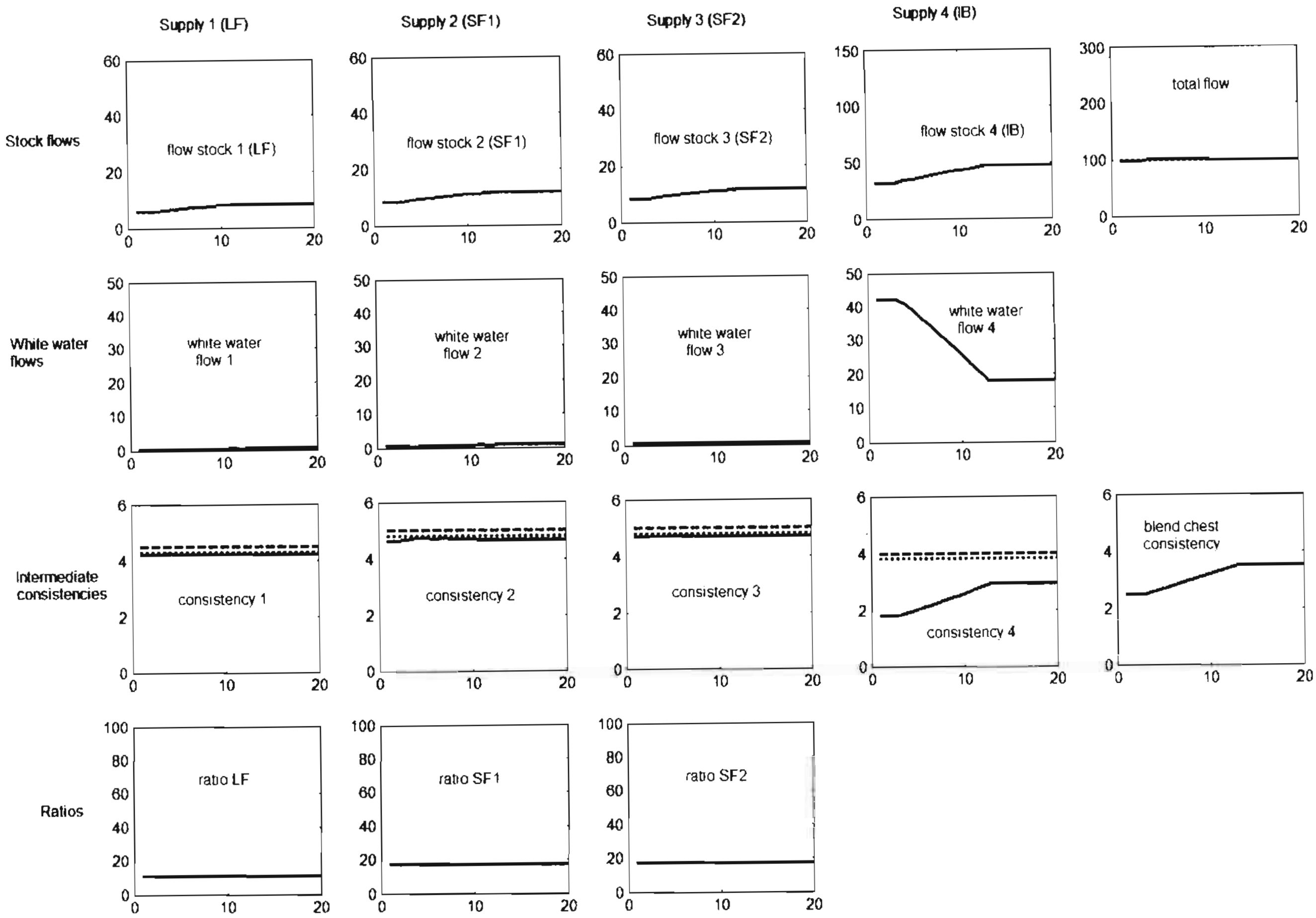

Figure 7.7: Unconstrained simulation – consistency ramp

All of the demands concerning the ratios, the blend chest consistency and total flow are satisfied. To reach the increasing consistency demand, the optimiser adjusts the stock flows and the white water flows, especially with the flow of stock 4 (IB) and the white water flow 4. The reason for this choice is that this IB flow has a really low consistency of $^{+}/_{-}$ 4%$^{m}/_{m}$ and does not affect controlled ratios, therefore it is the best candidate to satisfy a consistency demand of 2.5%$^{m}/_{m}$. The intermediate consistencies satisfy their setpoint, except for the consistency 4 which is used to lower the blend chest total consistency. This simulation is a good illustration of an unconstrained situation where all of the demands are satisfied.

7.3.2.2 Total flow ramp

In this simulation, the total flow is ramped from 50 to 150 L.s^{-1}. Simulation results are displayed in figure 7.8.

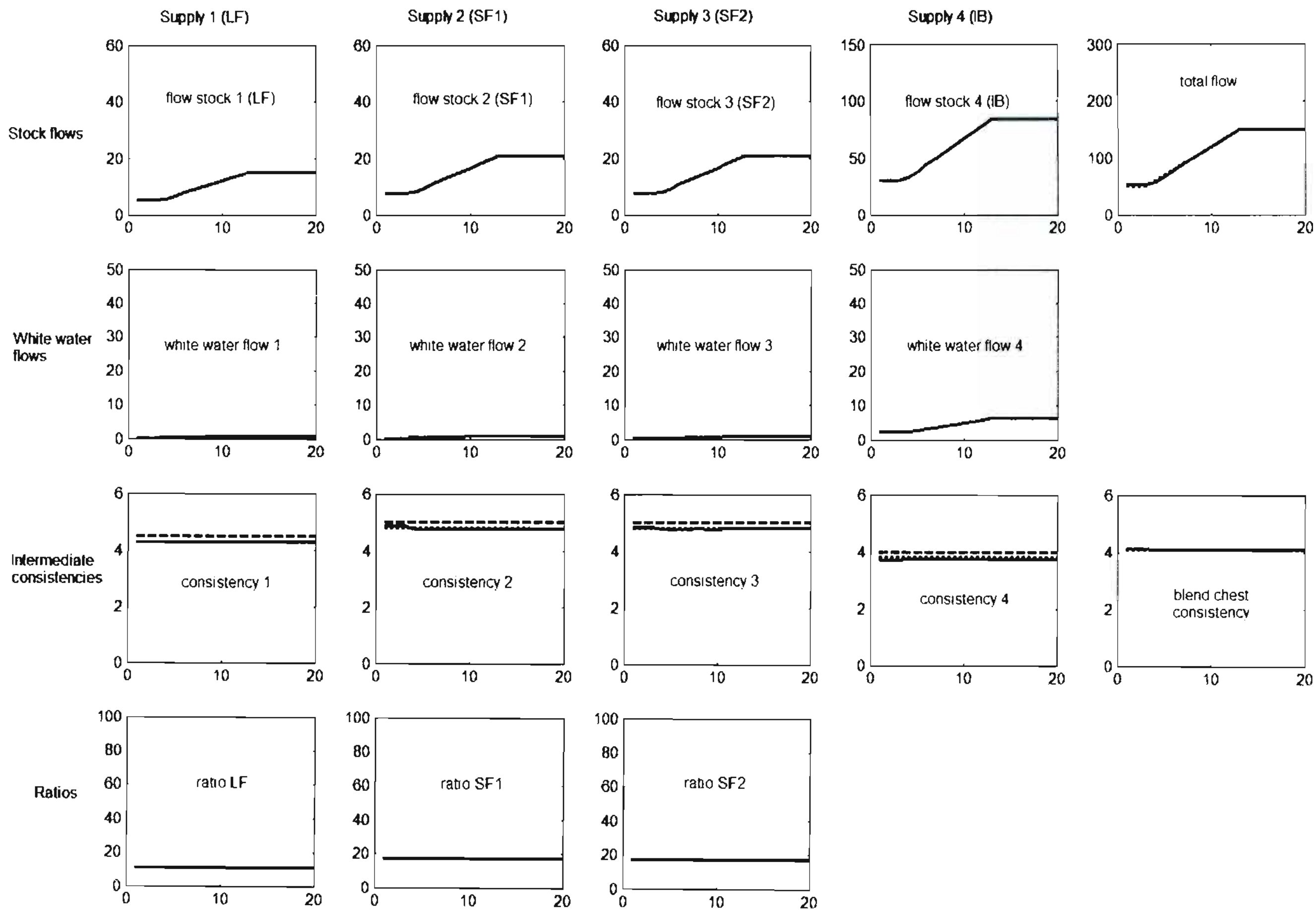

Figure 7.8: Unconstrained simulation – total flow ramp

The principal demands (consistency, flow and ratios) are satisfied. The intermediate consistencies are at their setpoints. The system handles the total flow ramp without any problems. Here again the IB stock flow appears prominent, but all flows rise in proportion.

7.3.2.3 Ratio ramp

In this scenario, the SF2 ratio β_2 is ramped from 10 to 20%. Results are represented in figure 7.9.

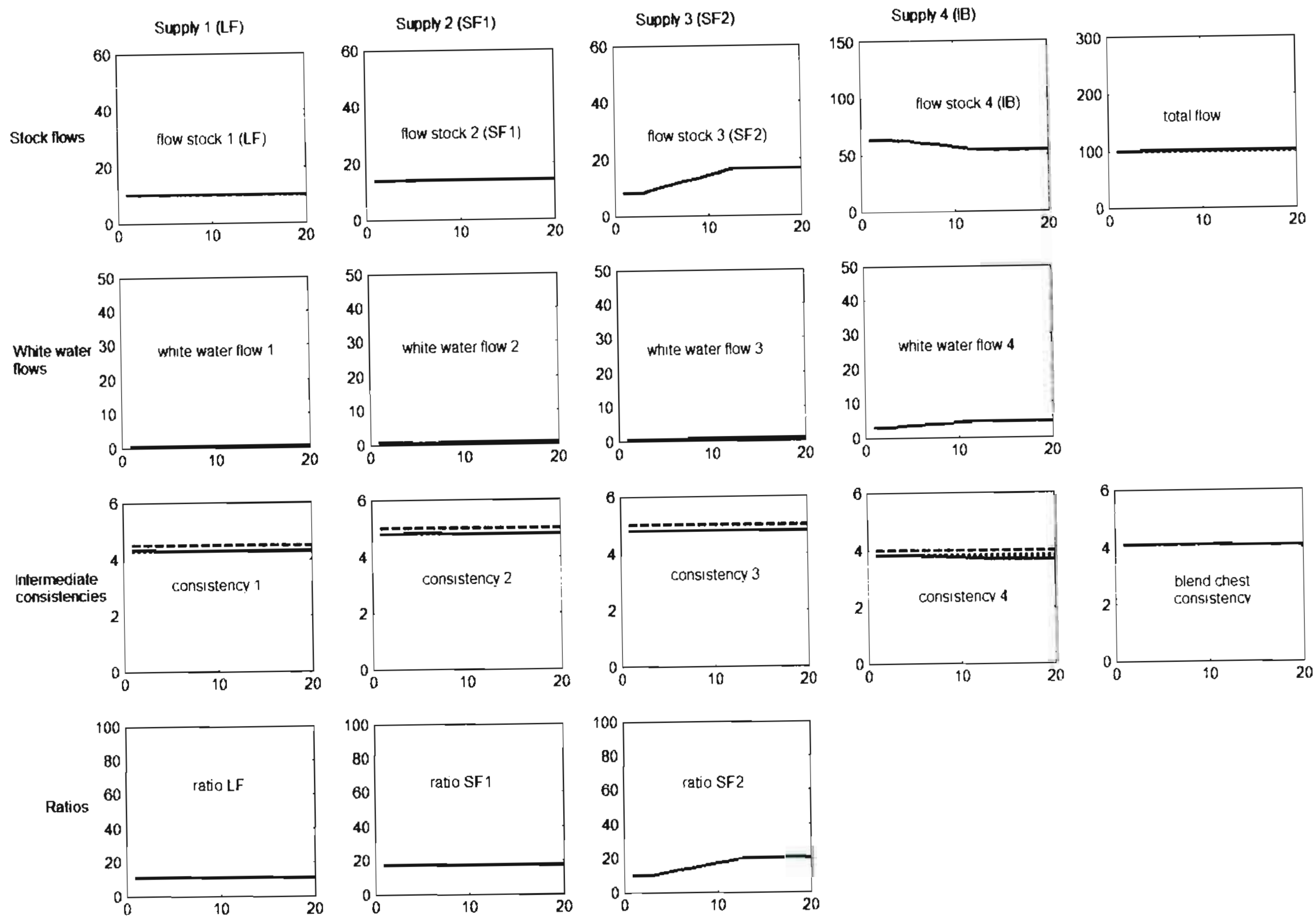

Figure 7.9: Unconstrained simulation – SF2 ratio ramp

Again in figure 7.9, all of the demands are satisfied. To reply to the ratio ramp, the system uses the IB flow stock and its corresponding white water flow which are reduced to maintain total flow and consistency.

Reflecting on these three situations, it has been proved that the simulator handles the unconstrained situation. In order to determine the limits of this simulator, some constrained scenarios are now considered.

7.3.3 **Constrained situations**

The objective of this specific part of the study is to determine the limits of the simulator. The optimiser was created to handle constrained situations by establishing a priority order amongst the controlled variables (section 5.2). This priority order states that ratios, blend chest consistency, blend chest total flow and finally intermediate consistencies must be satisfied in this specific order. To observe those constrained situations, big perturbations were introduced in the system and the simulator reaction was observed.

7.3.3.1 Consistency ramp

To be in a constrained situation, the total consistency was ramped from its minimum ($c=0.2\%^{m}/_{m}$, white water consistency) to its maximum feasible value. The results are observed in figure 7.10.

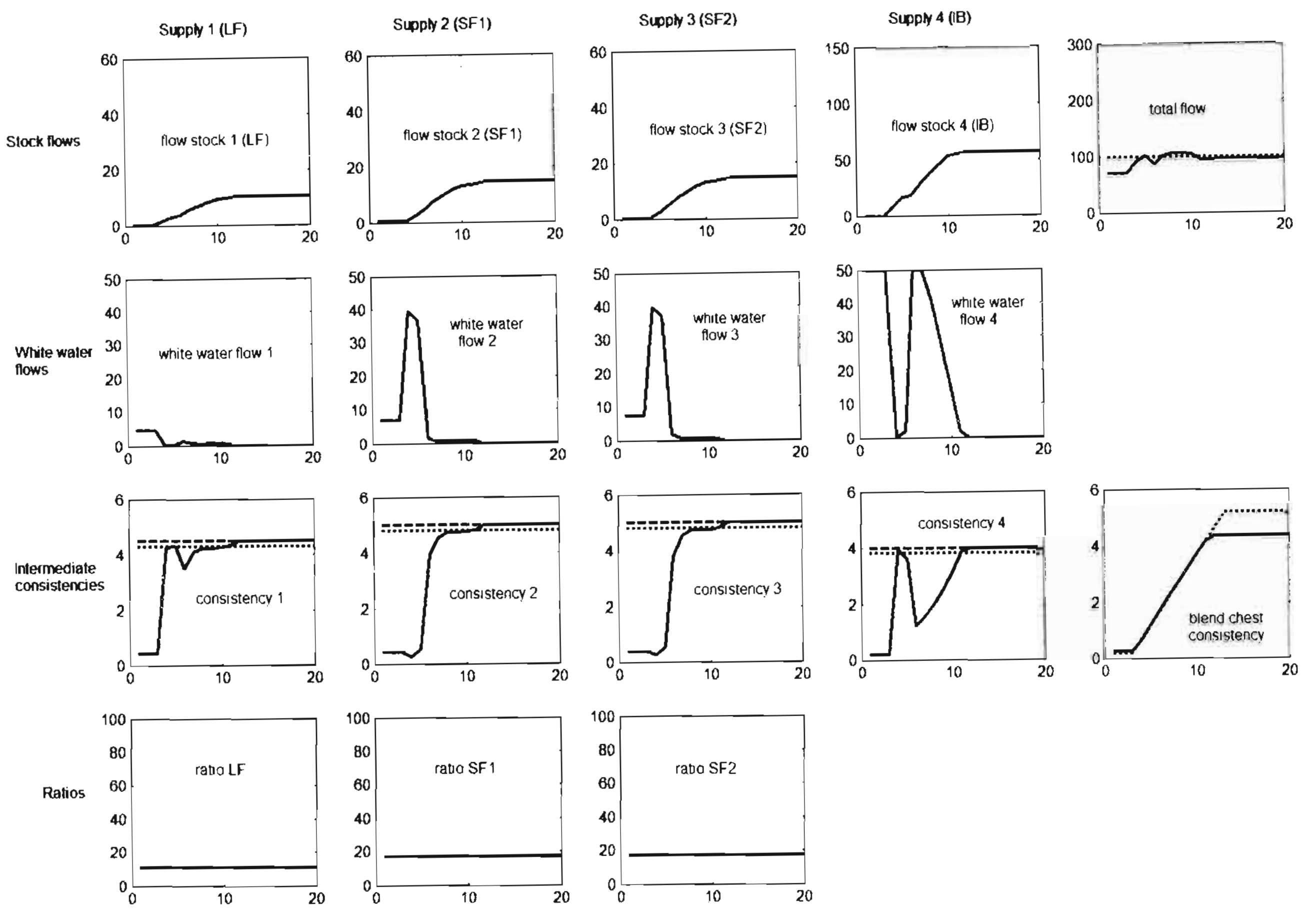

Figure 7.10: Constrained situation – consistency ramp

For this simulation, not all of the demands are satisfied: the ratios are on setpoint, the blend chest consistency is on setpoint as well, the demanded consistency at 11 sec is not physically possible. The total flow demand cannot be satisfied until the consistency is constrained (at 11 sec) but before this time, the optimiser uses its priority order to react to the system. It adjusts the total flow and the intermediate consistencies to satisfy the ratios and blend chest consistency demands.

7.3.3.2 Total flow ramp

The total flow was ramped from 50 $L.s^{-1}$ to 550 $L.s^{-1}$. The results are displayed in figure 7.11.

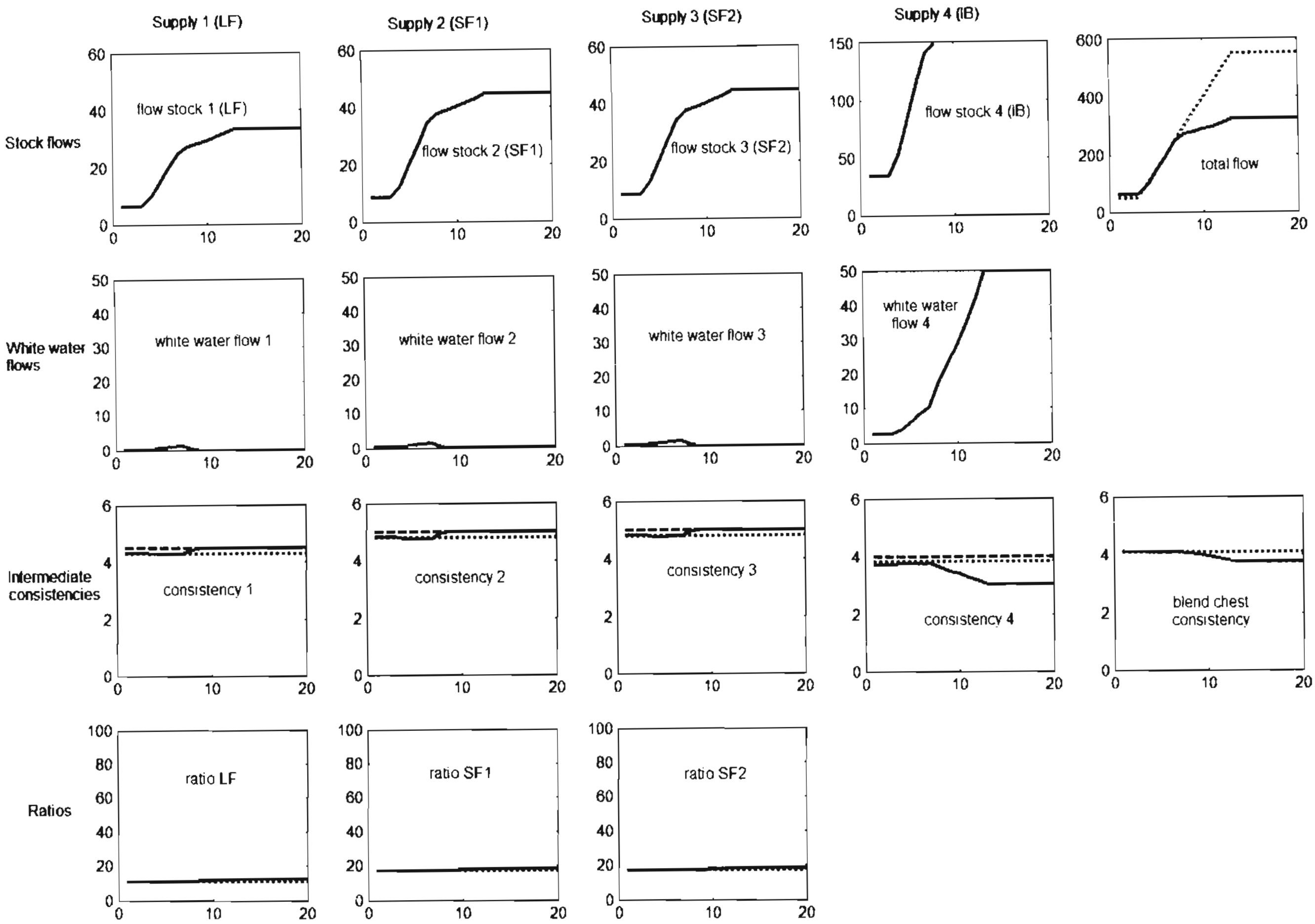

Figure 7.11: Constrained situation – total flow ramp

For this specific simulation, the introduced perturbation is drastic. The ratio demands are satisfied, the blend chest consistency is on setpoint until t = 10s and then moves from setpoint in a reasonable range. The total flow stays on setpoint until t = 8s and moves away from it. There are possibilities for the system to increase the total flow by increasing the stock flows but the priority is given to the consistency. Here again, the optimiser compromises between the controlled variables (the total flow and the total consistency) to keep the ratios on setpoint.

7.3.3.3 SF2 ratio ramp

For this experiment, the short fibre 2 ratio was ramped from 10% to 70% and the simulator response is displayed in figure 7.12.

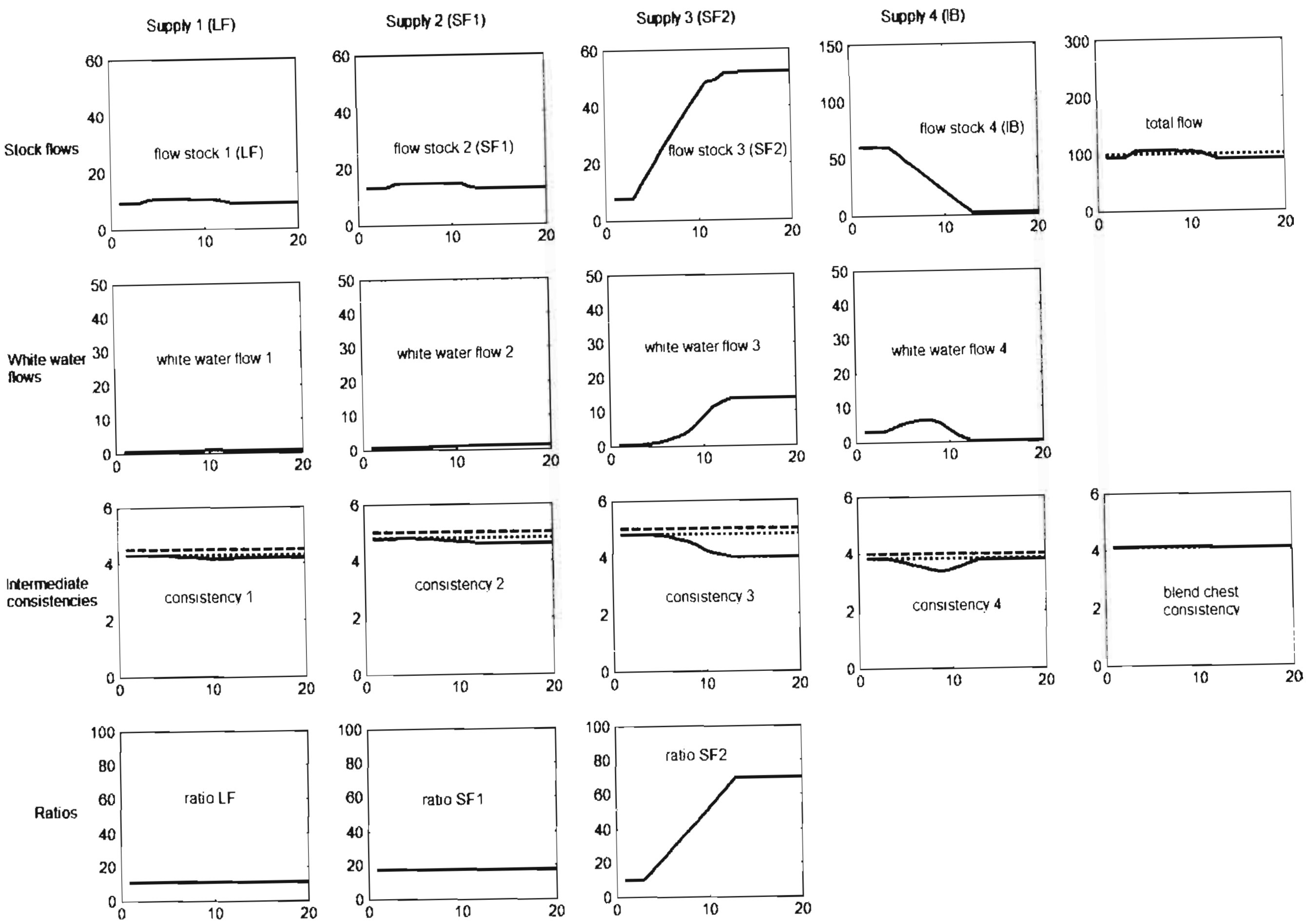

Figure 7.12: Constrained situation – SF2 ratio ramp

The different demands are respected except the total flow which has to move from setpoint. The intermediate consistencies move from setpoint to make sure the overall consistency remains constant.

7.4 Conclusion

In conclusion, a dynamic model of the stock preparation area including all of its controllers was developed in order to understand the interactions provoking loss of specification in the quality of the paper produced. It was discovered in reviewing plant data that the system had no automatic way to handle constrained situations. An optimiser "constraint pusher" was designed and tested by introduction of perturbations in the system. Responses were observed on different graphs. Optimiser algorithm parameters such as the fixed fraction of maximum range and the maximum number of iterations were tuned to guarantee the best performances and recorded in table 7.1. An order of priority in the handling of the constraints was established in agreement with Mondi. The optimiser proved to be efficient in extreme situations by satisfying the demands as much as possible with regards to this order of priority.

Chapter 8 : Conclusion and recommendations

8.1 Conclusion

The pulp and paper industry represents a big challenge. Firstly, its main raw material is a natural product that retains its non-uniform characteristics and behaviour through the manufacturing processes all of the way to the final consumer products. Secondly, it has to comply with the increasingly stringent environmental requirements and be responsive to customers demands for using environmentally friendly processes.

Over the years, the understanding of the process behaviour through mathematical modelling has improved considerably. Due to their critical importance, major processes like digesters, recovery boilers, pulp and paper machines, and evaporators continue to be modelled. At the same time, process control is improving due to the requirements of product quality. Pulp and paper processes are typically very similar to systems to which chemical engineers have been accustomed. Therefore, appropriate and successful control approaches are also similar in principle to those that have already been established through applications in petroleum and chemical industries. The challenge with the pulp and paper industries is due to the stochastic nature of the raw material, highly interactive multivariable process behaviour, long time delays and grade change transients.

In this work, an understanding of the controller interactions in the stock preparation section has been developed by detailed dynamic modelling, including all of the existing control loops. Closer consideration of the plant behaviour revealed that some of the input flows were frequently constrained, and that the system had no automatic way of reacting under constrained conditions, causing excessive loss of specification at the paper machine. To deal with these constraints systematically, a static optimiser was created and implemented on the model. Both the model and its optimiser were represented in a simulator, designed with the graphical user interface (GUI) of Matlab. The simulator has then been used to explore control performance over the operating range, leading to the formulation of a set of tuning scenarios and some requirements relative to the limits of the simulator. It is intended that this will provide a basis for on-line implementation of the optimiser.

The modelling of the stock preparation area and the approach used were the biggest challenges, as the literature concerning such models is really limited. Whereas digesters, boilers, evaporators and headboxes have been widely modelled in the past, the stock preparation area remains non-modelled. This is mainly due to the complexity of all of the interacting loops involved. Simplifications relative to the flows and the process units have been done, but the resulting model still fully represents the time responses. All of the controllers are modelled and the interactions between them are present.

The question of the language was important. One common simulation language for these situations is Simulink (Matlab) but only a certain number of operations can be modelled and the structure of the model is pre-established. In order to be free from such constraints, it was decided to write the program in Matlab. This is more suited to the complexity of the system and its equations.

Different approaches have been tried. Two major ones have been explored in detail. The first approach led to a set of Ordinary Differential Equations (ODEs) to be solved using the discrete form. The main problem of this approach resided in the fact that all of the data were gathered into one set of second order differential equations. This resulted in a high degree of interaction between parts of the solution. Debugging was complicated by the lack of readability of the system, therefore the causes of instabilities were hard to identify. The second approach used a modular viewpoint with a basic element. The overall system is represented by interlinking standard elements. The solution of the system consisted of three well identified steps including advanced controller solution, slave controller solution and finally the response of the system to those changes. The readability was greatly improved, resulting in an easy understanding of interactions.

Once the above practical approach was found, plant data were gathered to get useful information for the model: unit dimensions, controller settings, P&I diagram arrangement, on-line measurements. Data were collected on the plant, dimensions being measured on the site. The P&I diagram was studied carefully so that none of the important flows were omitted. Plant data revealed that some flows around the blend chest area were constrained resulting in a loss of specification with no means of control.

Tuning the controllers was a significant challenge as lots of interactions were present. It was nonetheless simplified by the clarity of the model. The trial-and-error tuning technique proved to be efficient. Some careful tuning was necessary to get rid of loop cycling for certain parts of the system.

An automatic way of dealing with the constraints around the blend chest had to be implemented. The solution came in the form of an optimiser (static openloop feed-forward controller) with a weighted objective function.

A study of optimiser intrinsic parameters was carried out in order to determine the optimal settings. Weight factors were set according to the priority order established between the controlled demands.

The model and optimiser were gathered into a simulator created with Matlab's GUI, which provides an easy access to the model and improves the readability. Indeed, the user does not need to access the code to set the controllers or to plot the graphs. This can be done via the interface using slider settings. Consistencies, levels, flows and controller tunings are accessible by simply clicking on the appropriate button on the appropriate unit.

Finally, the simulator has been used to explore performance under constrained and unconstrained situations and proved to be efficient.

8.2 Recommendations

In the design of the model a lot of approximations were made to simplify the understanding of the system. Small flows were neglected. In the structure of the model, no attempt was made to track the different fibre concentrations through the system. When the consistency was predicted, it was the overall consistency. Additional parameters can be implemented to access those specific consistencies (e.g. long fibre, short fibre,..). Then their individual consistencies could be traced until the headbox. This would highlight the consequences of one constraint in the feed flows on the quality of the paper produced via the percentage of each type of fibre in the paper.

The design of the optimiser was really simple. It suits the model for this configuration but it might be inadequate if the complexity of the model is increased. At the moment, to get to the solution, the optimiser searches space in 8 dimensions. The crude testing of one direction at a time might be inadequate for more complex models, requiring more sophisticated searches.

Finally and most importantly, the model needs to be validated. To this point, data entered in the model were approximate, good enough to get an idea of the plant behaviour but not accurate enough to compare with the reality. Therefore accurate data need to be collected and included in the model, and the reaction of the model to one process variation must be observed and compared to the plant behaviour. For instance, it will be of interest to identify the response of the blend chest to a change in the feed pulp consistency. That could clarify the problem encountered on the blend chest when the paper grade is changed.

It is intended that this work will provide a basis for on-line implementation of the optimiser in the plant DCS. This will give an automatic way to the control system to deal with the blend chest feed constraints.

The optimiser is still at the experimental stage. It represents a good basis for further investigation and improvements in the control of the stock preparation area.

References

- Andreiev, N. (1981). A new dimension: A self-tuning controller that continually optimises PID constants. *Control Engineering*. **28** (8), p84.
- Astrom, K.J. (1983). Theory and application of adaptive control – a survey. *Automatica*. **19**(5), p471-486.
- Astrom, K.J. and Hagglund, T. (1995). PID Controllers: Theory, Design and Tuning. *Second edition. Research Triangle Park, NC: Instrument Society of America.*
- Astrom, K.J. and Wittenmark, B. (1973). On self-tuning regulators. *Automatica*. **9**, p185-199.
- Bengtsson, G. and Egardt, B. (1984). Experiences with self-tuning control in the process industry. *Proceedings of the 9th world IFAC congress, Budapest*. Paper CS-7.
- Bialkowski, W.L. (1991). Recent experiences with advanced control aimed at optimizing product uniformity. *In Process Control Conference. Tappi Press*, Atlanta, p 127-136.
- Bond, T. and G. Dumont (1988). On adaptive control of color for paper machines. *In American Control Conference,* p1965-1970.
- Borisson, V. (1979). Self-tuning regulators for a class of multivariable systems. *Automatica*, **15**, p209-215.
- Bristol, E. (1966). On a new measure of interaction for multivariable process control. *IEEE Trans. Autom. Control AC-11*, p133.
- Bussiere, S., A.A. Roche, and J. Paris (1992). Analysis and control of white water network perturbations in an integrated newsprint mill. *Pulp Pap. Can.*, **93**(4), p41-44.
- Camacho, E.F. and Bordons, C. (1995). Model predictive control in the process industry. *Spinger Verlag New-York.* ISBN 3-540-19924-1, 239 pages.
- Cegrell, T. and Hedqvist (1974). A new approach to continuous digester control. *IFAC Symp. On Digital Comp. Appl. To Process Control.*
- Cegrell, T. and Hedqvist (1975). Successful adaptive control of paper machines. *Automatica*, **11**, p53-60.
- Chang, T.S. and D.E. Seborg (1983). A linear programming approach for multivariable feedback control with inequality constraints. *Int. J. Control.* Vol. 37, no. 3, p583-597.
- Chen, S.C. (1988). Kalman filtering applied to sheet measurement. *In American Control Conference. American Automatic Control Council*, p643-647.

- Chen, S.-C. and L. Adler (1990). Cross machine profile control for hevy weight paper. *In Control Maintenance Environment: New Available Techniques and Current Trends,* p77-94.
- Chen, S.C. and R.G. Whilhem Jr. (1986). Optimal control of cross-machine direction web profile with constraints on the control effort. *In American Control Conference,* p1409-1415.
- Clarke, D.W. and Gawthrop, P.J. (1979). Self-tuning control. *IEEE Proceedings*, Part. D, vol 126, p633-640.
- Clarke, D.W., Mohtadi, C., Tuffs, P.S. (1987). Generalized predictive control – Part I. The basic algorithm. *Automatica.* **23**(2), p137-148.
- Clarke, D.W., Mohtadi, C., Tuffs, P.S. (1987). Generalized predictive control – Part II. Extension and interpretations. *Automatica.* **23**(2), p137-148.
- D'Hulster, F., DeKeyser, R. and Van Cauwenberghe (1980). Application of several parameter-adaptive controllers to a paper machine headbox. *4^{th} IFAC PRP conf.*
- Dahlin, E.B. (1968). Designing and tuning digital controllers. *Instr. And cont. system*, **41** (6), p77.
- Di Ruscio, D. and J.G. Balchen (1992). State space model for the TMP process. *In Control Systems 92.*
- Dumdie, D.P. (1988). Systems approach to consistency control and dry stock blend. *Tappi J.*, **71**(7), p135-139.
- Fjeld, M. and Wilhelm, R. (1981). Self-tuning regulators – the software way. *Control engineering* **28**.
- Fletcher, R. and Reeves, C.M. (1964). Function minimization by conjugate gradients. *Computer Journal*, **7**, p148-154.
- Garcia, C.E., Prett, D.M. and Morari, M. (1989). Model predictive control: theory and practice – a survey. *Automatica*, **25** (3), p335-348.
- Gullichsen, J., Paulapuro, H., Sundholm, J. (1999). Book 5 : Mechanical Pulping. *Papermaking Science and Technology*. Jyvaskyla, 427 pages.
- Haag and Wilson (2001). Large-scale dynamic paper machine models. *6^{th} World Congress of Chemical Engineering.*
- Halme, A. and Ahava, O. (1984). Automatic tuning of PID and other simple regulators in a digital process automation system. *1984 ACC.*
- Halme, A. and Selkainaho, J. (1982). Application of a non-linear filter to multivariable parameter adaptive control in a distributed microcomputer system. *6^{th} IFAC Symp. On ident. And Syst. Param. Estimation.*
- Hoopes, H.S., Hawk, W.M. and Lewis, R.C. (1983). A self-tuning controller. *The instrumentation, Systems and automation society transactions.* **22**, p49.
- Kalman, R.E. (1958). Design of a self-optimizing control system. *Transaction ASME.* 80, p468-478.

- Karlstrom and Koebe (1993). Modeling of wood-chip refining process. *Nord.Pulp Pap. Res. J.*, **8**(4), p384-388.
- Kaunonen, A., H. Kovio, P. Virtaten and P.Sumanen (1985). Modeling and simulation of paper machine's wet-end process. *In B. Walshtrom and K. Leiviska (Eds), IMACS World Congress on Systems Simulation and Scientific Computation*, v 3, p229-231.
- Kayihan, F. (1996). A review of modeling and control in the pulp and paper industries. In *Chemical process control – V*, p117-132.
- Kelly, W.H. (1982). Self-tuning control strategies: modern solutions to paper making control problems. *Can. Industr. Comp. Society Conf.*
- Kooi, S.B.L. and K. Khorasani (1992). Control of wood-chip refiner using neural networks. *Tappi J.*, **75**(6), p156-162.
- Krauss, T.W. and Myron, T.J. (1984). Self-tuning PID controller uses pattern recognition approach. *Control engineering.* **31**(6), p106-111.
- Leiviska Kauko. (1999). Process Control. *Papermaking Science and Technology.* ISBN 952-5216-14-4, 297 pages.
- Luyben, W.L. (1990). Process Modelling, Simulation and Control for Chemical Engineers: Second Edition. *McGraw and Hill International Editions-Chemical Engineering Series.* ISBN 0-07-100973-8, 726 pages.
- Mangold, M., Motz, S. and Gilles, E.D. (2002). A network theory for the structured modeling of chemical processes. *Chemical Engineering Science.* **57**, p4099-4116.
- Matlab official website: http://www.mathworks.com
- McAvoy, T.J. (1983). Interaction Analysis: Principles and Applications, *Instrument Society of America: Research Triangle Park*, NC
- Miles, K. B. and W. D. May (1993). Predicting the performance of a chip refiner, a constitutive approach. *J.Pulp Pap. Sci.*, **19**(6), p268-274.
- Morari (1983). *Chemical Engineering Science*, **38**, p.1881
- Niederlinski, A. (1971). A heuristic approach to the design of linear multivariable control systems. *Automatica 7*, p691
- Parks, P.C., Schaufelberger W., Schmid C. and Unbehauen H. (1980). Application of adaptive control systems. *Springer-Verlag, New-York.* In "*Methods and application in adaptive control, Lecture notes in control and information sciences, 24*", p161-198.
- Partanen, K. and H. Koivo (1984). On-line freeness control at a thermo-mechanical pulping plant. *In Advances in Instrumentation*, pt 2, vol. 39.
- Paulapuro Hannu. (2000). Papermaking Part 1, Stock Preparation and Wet End. *Papermaking Science and Technology.* ISBN 952-5216-08-X, 461 pages.
- Peterska, J. and Astrom, K.J. (1973). Control of multivariable systems with unknown but constant parameters. *Proceedings of the 3rd IFAC Symposium on identification and system parameter estimation*, p535-544.

- Pigeron, B., Mullot, H., Chaix A., Felix, L., Aubert Y. (1996). Boucles de régulation – Etude et mise au point. *Bhaly autoédition*. ISBN 2-950-9995-0-6.
- Piirto, M. and H. Koivo (1992). Advanced control of a paper machine wet end. *In Advanced Control of Chemical Processes: ADCHEM 91*, p47-52.
- Powell, M.J.D. (1965). An efficient method for finding the minimum of a function of several variables without calculating derivations. *Computer Journal*, vol. 7, p155-162.
- Press, W.H. (1992). Numerical Recipes in C: The art of Scientific Computing. *Cambridge University Press*. ISBN 0-521-43108-5.
- Qian, X. and P.J. Tessier (1993). Modeling and simulation of a chip refining process. *In Pacific Paper Expo Technical Conference*. Maclean Hunter, p116-122
- Qian, X., P.J. Tessier, and G.A. Dumont (1994). Fuzzy logic modeling and optimization of a wood chip refiner. *Tappi J.* 77(2), p181-186.
- Qin, S.J. and Badgwell, T.A. (1997). An overview of industrial model predictive control technology. *Fifth international conference on chemical process control*, p232-256.
- Richards, R.J. (1979). An introduction to dynamics and control. *Longman*. ISBN 0-582-44182-X
- Rosenbrock, H.H. (1974). Computer-aided control system design, *Academic Press*, London
- Sastry, V. (1978). Self-tuning control of Kamyr digester chip level. *Pulp and paper Canada*, **79**, T160-T163.
- Shinskey, F.G. (1979). Process Control Systems, *McGraw-Hill*, New York
- Shinskey, F.G. (1981). Controlling Multivariable Processes, *Instrument Society of America: Research Triangle Park*, NC
- Shinskey, F.G. (1996). Process control systems – 4th edition. *Mc Graw-Hill Editions, New York*. ISBN 0-07057-101-5, 439 pages.
- Sikora, R.F. (1983), Self-tuning control of moisture content. *3rd Int. Pulp and Paper Proc. Contr. Symp.*
- Sikora, R.F. and Bialkowski (1984). A self-tuning strategy for moisture control in paper making. *1984 ACC*.
- Spendley, W., Hext, G.R. and Himsworth, F.R. (1962). Sequential application of simplex design of optimization and evolutionary operations. *Technometrics*, **4**, p441-461.
- Strand, B.C. (1993). Analyzing the effects of raw material variation on mechanical pulp properties using integrated factor networks. *In Annual Meeting – Technical Section, Canadian Pulp and Paper Association, Preprints A, 79th*.
- Tessier, P. and X. Qian (1994). Modeling and simulation of a ctmp process. *In Annual Meeting – Technical Section, Canadian Pulp and Paper Association.*

- Vincent, J.P., T. Nguyen-Minh, W. Hagen, B. Lebeau, J.M. Dion and L. Dugard (1994). Advanced control strategies for the dyeing of white and colored papers. *Pulp Pap. Can.*, **95**(9), p34-38.
- Virtanen, P. (1988). Computer simulator for paper machine wet-end. *In Process Simulation Symposium.* CPPA, Montreal, p85-89.
- Wilhelm, R.G. (1982). Self-tuning control strategies: multifaceted solutions to paper machine control problems. *ISA Spring Symp.*
- Wilhelm, R.G. and Fjeld, M. (1983). Control algorithms for cross directional control: the state of the art. *5th IFAC PRP Conf.*
- Wong, K.P. and Seborg, D.E. (1986). A theoretical analysis of Smith and analytical predictors. *AIChE journal.* **32**, p1597-1605.
- Xia, Q., Rao, M. and Qian, J. (1993). Model algorithmic control for paper machines. *IEEE Conference on Control applications*, v.1. IEEE, Piscataway, p 203-208.
- Xia, Q., Rao, M., Shen, X. and Zhu, H. (1993). Adaptive control of basis weight and moisture content for a paperboard machine. *Journal of Process Control*, **3**(4), p203-209.
- Xia, Q., Rao, M., Shen, X. and Zhu, H. (1994). Adaptive control of a paperboard machine. *Pulp Pap. Can.*, **95**(5), p51-55.
- Ydstie, B.E., Kershenbaum L.S. and Sargent R.W. (1985). Theory and application of a robust tuning regulator. *AIChE journal.* **31**(11), p1771-1780.
- Ying, Y.Q., Rao, M. and Sun Y.X. (1991). A new design method for bilinear suboptimal systems. *American Control Conference.* American Automatic Control Council, Green Valley, p 1820-1821.

Appendix A: Different types of pulp

The different types of pulp derive their names from the process used to produce them. These are groundwood pulp (GWD), thermo-mechanical pulp (TMP), and chemical pulp. The first two types, made from long fibre wood, are sometimes referred to as mechanical pulps since the process for obtaining them is mechanical. A fourth process is the repulping of de-inked paper (newspapers and magazines), which recovers fibre (RCF).

- Groundwood: Debarked logs are conveyed to a stone trap, where they are sprayed with water to wash off the dirt and bits of bark, which would weaken and darken the paper. From there the logs are cut into 1,2 metre lengths, before being loaded into magazines and then pressed sideways against a large rotating grindstone which strips and separates the wood fibres. Water is sprayed onto the grindstone to wash the fibres off and to reduce the heat generated from the grinding process. The water and fibres form a thick pulp mixture. The groundwood stock is then screened to remove dirt, large fibre bundles and knots. After screening, the stock excess water is removed in order to reduce the space required for storage and increase the consistency (thickness) of the pulp. The stock is now ready for blending.

- Thermo-mechanical pulp: The second type of pulp is TMP (Thermo-mechanical pulp). The pulp produced from this process costs more than groundwood but the pulp has longer fibres, which will strengthen the paper. Logs are conveyed from the debarker to the chipper, which is a large flywheel with sharp blades mounted on its surface. The logs are fed through a chute onto the flywheel, which rotates at high speed and rotating blades chop across the log, cutting off chips. These chips are then blown by compressed air into chip silos for temporary storage. From the silos the chips are conveyed to screens where sawdust and oversized chips are removed. The remaining chips are washed to remove dirt and dust. The next stage in the TMP process is to break down the wood chips into fibres; they are fed into a digester, sometimes known as the pressure cooker. Here the chips are cooked under steam pressure to soften the lignin or glue, which holds the fibres together. From the digester the chips are fed into a series of refiners (two discs, rotating in opposite directions at high speed, covered with raised bars of varying size). As the chips pass between the discs they are torn apart and ground down to fibres. The refining process, by rubbing and cutting the fibres, reduces them to a suitable length and also bursts the cell walls, which assists the fibres to stick together. The stock leaving the refiners is then screened, cleaned and finally thickened before being pumped into storage tanks.

- Chemical pulp: Chemical pulp may be made from either long or short fibre wood. Chemical pulp is needed for blending with groundwood and Thermo-mechanical pulps to strengthen the paper. It is a very pure pulp, the fibres having been separated by dissolving the lignin which binds them, via a process of cooking the wood in a chemical 'soup'. Hence the fibres are not mechanically damaged. (Paper made from chemical pulp is usually referred to as woodfree.)

Dry chemical pulp arrives in bales, which, when required, are loaded onto a conveyer that feeds the dry pulp into the hydra-pulper to which water is then added. The hydra-pulper is a circular, cup shaped tank with rotating blades mounted in the base. It operates on the same principle as a kitchen blender: the rotating blades break the sheets of pulp into fibres suspended in water. From the hydra-pulper the pulp and water mixture is fed into a series of refiners where the fibres are treated further into a form suitable for papermaking.

De-inked pulp: This recovered fibre pulp (RFP) is produced by removing the printing ink from waste magazines and newspapers and reducing it to its original form. Waste paper is pulped in a mixture of water and chemicals (caustic soda, hydrogen peroxide and sodium silicate), which commences the process of loosening the printer's ink from the fibres. This grey stock is then screened and cleaned to remove first larger, then smaller contaminants (such as staples) and then moved to the flotation cells where a further chemical known as soap is added. This soap causes the mixture to foam and since printer's ink is hydrophobic (water hating), it clings to the air bubbles in the foam, which then forms a scum on the surface, which is easily skimmed off. Further cleaning, fine screening and thickening is followed by pressing, to remove most of the water and chemicals from the stock, which is then heated and refined to disperse any plastic particles not screened out earlier. Finally, sodium hydrosulphides are added to the stock to whiten it before it is stored, ready for use on the newsprint paper machines.

Appendix B: Plant data

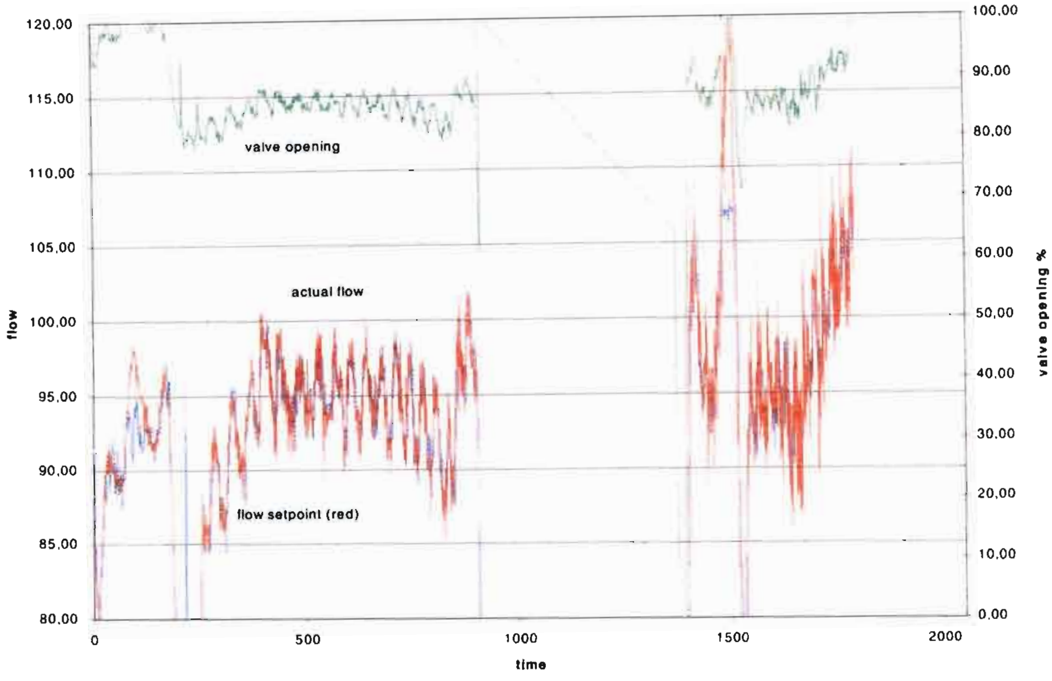

Figure B.1: Inside broke flow control

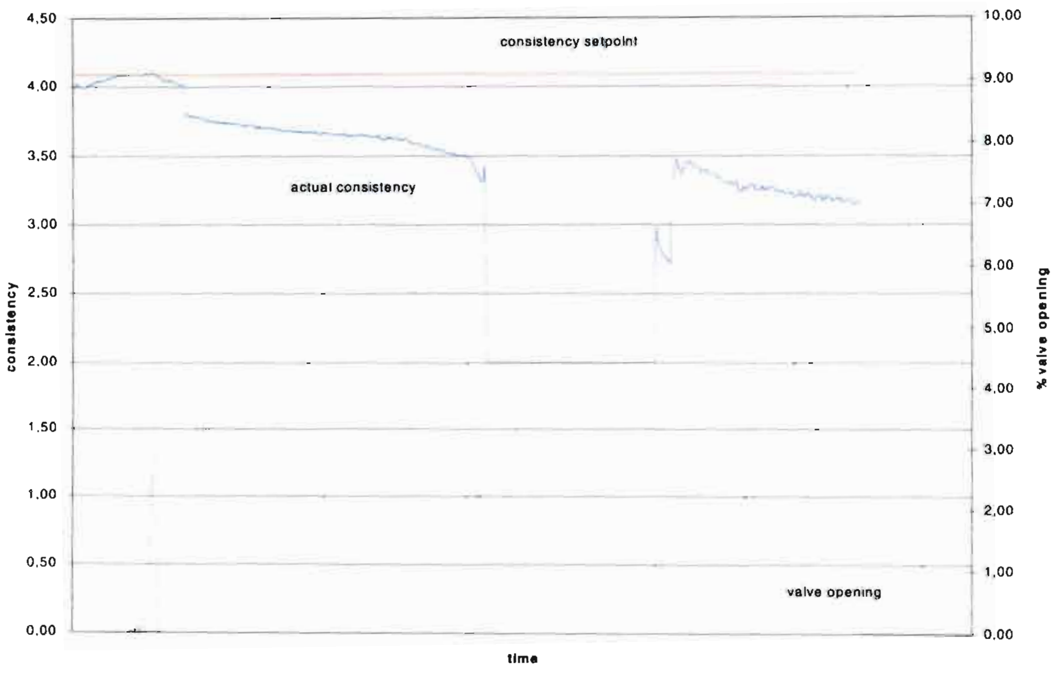

Figure B.2: Stock to blend chest consistency control

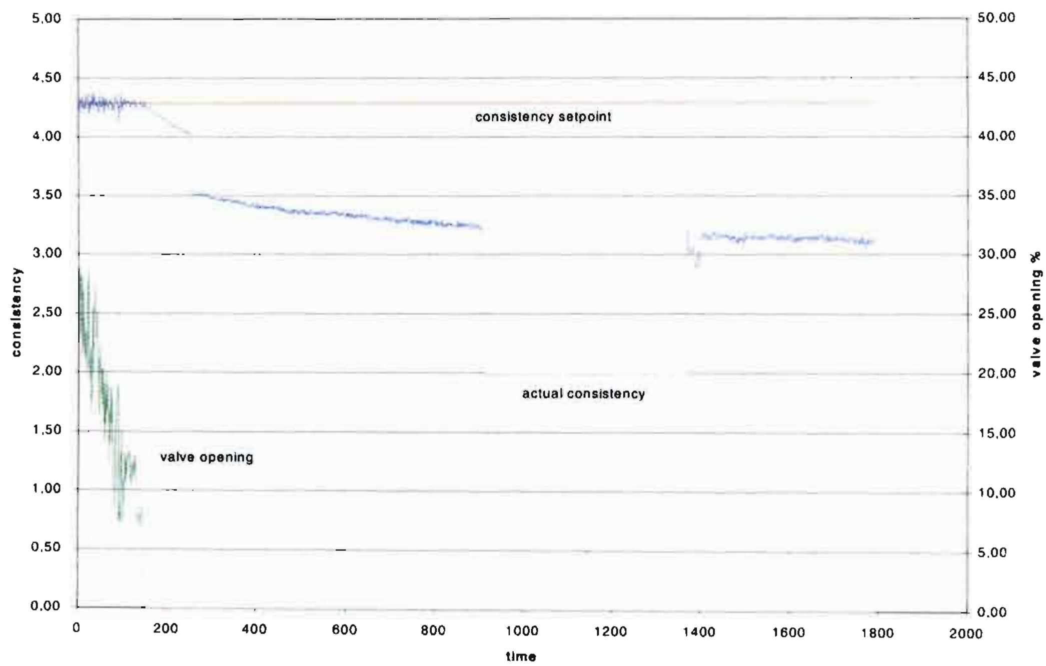

Figure B.3: Inside broke consistency control

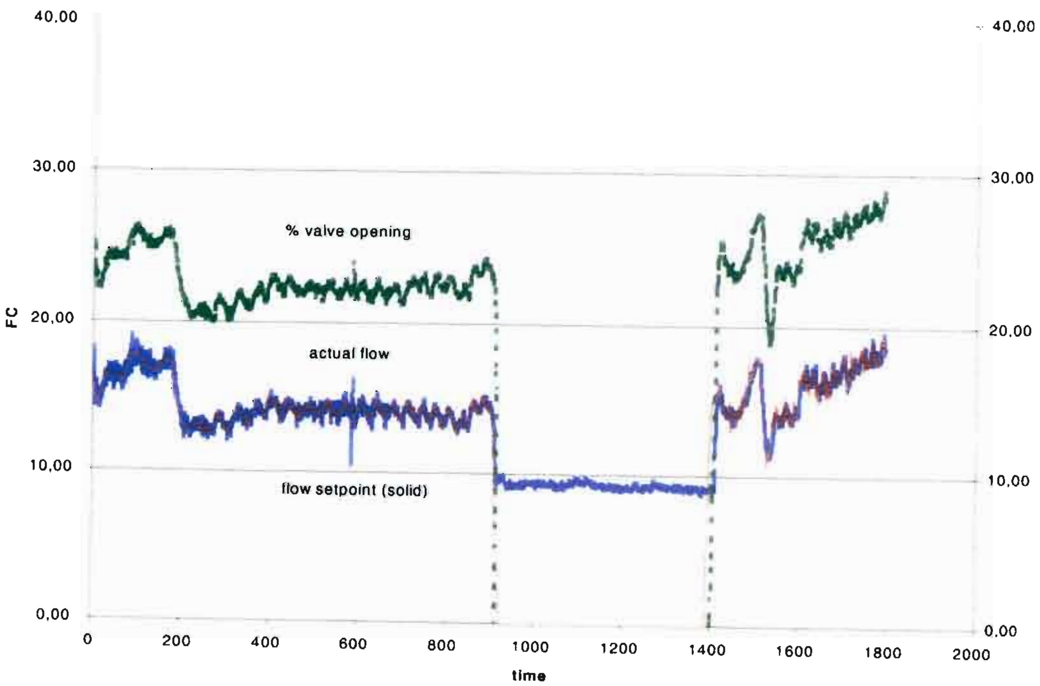

Figure B.4: Long fiber consistency control

Appendix C: Listing of the optimiser algorithm

Declaration of variables

```
TRUE=1;
FALSE=0;
SMALL=1e-5;   % divided by zero protection
MINVOL=100;   % minimum volume - divided by zero protection
npoints_max = 6000;   % maximum points stored for run at interval dtplot
maxconsistency = 7;   %   percent : maximum allowed inestimator
alpha_consistency_estimation = 0.02;   % smoothing factor for supply vessel consistency estimation
% Listing of variables
n=25;   % number of elements
q=n;    % number of exogenous flow
r=1;    % number of expanded variables
n_adv_cont_var=22;    % number of controlled variables
v=zeros(n,1);    % volumes
y=zeros(n,1);    % composition per vessel
ysupest = zeros(4,1);    % estimate of concentrations of supplies
firstcall = 1;    % firstcall of optimiser
secondcall = 1;    % secondcall of optimiser
f=zeros(4*n,1);    % flow vector f
fs=zeros(4*n,1);    % flow setpoint vector
fs_offset=zeros(4*n,1);    % fs offset
x=zeros(4*n,1);        % valve vector x
s=zeros(n,1);          % input nodes flow vector
eps=zeros(q,1);        % exogenous flow vector
phi=zeros(5*n,1);      % combined flow vector
D=zeros(n,4*n);        % flow distribution matrix
E=zeros(n,q);          % exogenous flow distribution matrix
R=zeros(n,4*n);        % outflow matrix
c=zeros(q,1);          % exogenous composition vector
Df=zeros(n,4*n);       % = D.* Mf ( Mf matrix with f in rows)
Eeps=zeros(n,q);       % = E.* Meps ( Meps matrix with ε in rows)
Rf=zeros(n,4*n);       % = R.* Mf ( Mf matrix with f in rows)
T=zeros(4*n,n);        % doubling matrix
A=zeros(6*n,6*n);      % L(n+1)=A*L(n)+B*N(n) A input matrix
B=zeros(6*n,2*q+4*n);      % L(n+1)=A*L(n)+B*N(n) B input matrix
L=zeros(6*n,1);        % L(n+1)=AL(n)+BN(n) L vector calculated
N=zeros(4*n+2*q,1);    % L(n+1)=AL(n)+BN(n) N input vector
Mde=zeros(n,n);    % transformation of the M matrix into a readable one
Mdf=zeros(n,n);    % transformation of the M matrix into a readable one
Mdg=zeros(n,n);    % transformation of the M matrix into a readable one
Mdh=zeros(n,n);    % transformation of the M matrix into a readable one
To=zeros(4*n,1);   % time constant
```

```
H=zeros(4*n,4*n);     % matrix with -1/To in its diagonal
CV=zeros(n_adv_cont_var,1);    % advanced controllers data
finter=zeros(n_adv_cont_var,1);    % data transfer from advanced controller to slave ones
fs_inter=zeros(4*n,1);    % flow setpoint
CVs=zeros(n_adv_cont_var,1);    % advanced controller setpoint
CVEM=zeros(2*n+4*n+r,2*n+4*n);    %Controlled variable expansion matrix
CSM=zeros(n_adv_cont_var,2*n+4*n+r);    %Control selection matrix
K=zeros(r,4*n);    % matrix created in case of a CC on the mix of two flows
ROM=zeros(4*n,n_adv_cont_var-1);    % reverse operation matrix
conversion=zeros(n_adv_cont_var,n_adv_cont_var);    % conversion matrix for controller
%  settings from plant units to model units
reconv=zeros(n,n);    % reconversion matrix from model to plant units
psi=zeros(10,1);    % weight parameter for the optimiser
VC=zeros(9,1);    % observed variables for the optimiser
VCSP=zeros(5,1);    % observed variables setpoint for the optimiser
CSP=zeros(4,1);    % observed intermediate consistency variables setpoint for the optimiser
fopt=zeros(4,1);    % input pulp flow variables for the optimiser
wopt=zeros(4,1);    % input white water flow variables for the optimiser
xint=zeros(14,1);    % intermediate consistency vector
topt=zeros(8,1);    % optimisation vector
topttemp=zeros(2*8,1);    % calculation vector for the optimisation
eta=zeros(4,1);    % ratios
toptmax=zeros(8,1);    % maximum values for topt
toptmin=zeros(8,1);    % minimum values for topt
fmax=zeros(4*n,1);    % maximum value for the flows
Par_cv=zeros(n_adv_cont_var,3);    % matrix gathering parameter tunings values
kc_x=zeros(4*n,1);    % flow controller proportional action
Ti_x=zeros(4*n,1);    % flow controller integral action
Td_x=zeros(4*n,1);    % flow controller derivative action
kc_cv=zeros(n_adv_cont_var,1);    % advanced controller proportional action
Ti_cv=zeros(n_adv_cont_var,1);    % advanced controller integral action
Td_cv=zeros(n_adv_cont_var,1);    % advanced controller derivative action
%  for optimisation variables
Jopt=1e+15;    % Objective function initialisation
FFMR=0.001;    % Fixed fraction of maximum range
itermax=1000;    % number maximum of iteration for the optimisation
```

```
function handles = DoPlot(hObject, eventdata, handles)

% Recover present values

npoints_max=handles.npoints_max;

savetime=handles.savetime;

savee=handles.savee;

savef=handles.savef;

saveg=handles.saveg;

saveh=handles.saveh;

savev=handles.savev;

savey=handles.savey;

savexe=handles.savexe;

savexf=handles.savexf;

savexg=handles.savexg;

savexh=handles.savexh;

savefs=handles.savefs;

saveCV=handles.saveCV;

saveCVs=handles.saveCVs;

saveVC=handles.saveVC;

saveVCSP=handles.saveVCSP;

saveCSP=handles.saveCSP;

saveoptsteps=handles.saveoptsteps;

saveJopt=handles.saveJopt;

saveit=handles.saveit;

saveJ=handles.saveJ;

optsteps=handles.optsteps;

Jopt=handles.Jopt;

savepsi=handles.savepsi;

TRUE=handles.TRUE;

FALSE=handles.FALSE;

SMALL=handles.SMALL;

maxconsistency = handles.maxconsistency;

alpha_consistency_estimation = handles.alpha_consistency_estimation;

MINVOL=handles.MINVOL;

n=handles.n;

q=handles.q;
```

```
r=handles.r;
n_adv_cont_var=handles.n_adv_cont_var;
v=handles.v;
y=handles.y;
ysupest=handles.ysupest;
firstcall=handles.firstcall;
secondcall=handles.secondcall;
f=handles.f;
fs=handles.fs;
fs_offset=handles.fs_offset;
x=handles.x;
s=handles.s;
eps=handles.eps;
phi=handles.phi;
D=handles.D;
E=handles.E;
R=handles.R;
c=handles.c;
Df=handles.Df;
Eeps=handles.Eeps;
Rf=handles.Rf;
%####vT=handles.vT;
A=handles.A;
B=handles.B;
L=handles.L;
N=handles.N;
Mde=handles.Mde;
Mdf=handles.Mdf;
Mdg=handles.Mdg;
Mdh=handles.Mdh;
To=handles.To;
H=handles.H;
CV=handles.CV;
finter=handles.finter;
fs_inter=handles.fs_inter;
```

```
dfinter=handles.dfinter;
CVs=handles.CVs;
CVEM=handles.CVEM;
CSM=handles.CSM;
K=handles.K;
ftot=handles.ftot;
ROM=handles.ROM;
conversion=handles.conversion;
reconv=handles.reconv;
psi=handles.psi;
VC=handles.VC;
VCSP=handles.VCSP;
CSP=handles.CSP;
fopt=handles.fopt;
wopt=handles.wopt;
xint=handles.xint;
%Matcomp=handles.Matcomp;
ttot=handles.ttot;
topt=handles.topt;
topttemp=handles.topttemp;
eta=handles.eta;
Mtopt=handles.Mtopt;
%fmax=handles.fmax;
toptmax=handles.toptmax;
toptmin=handles.toptmin;
Par_cv=handles.Par_cv;
kc_x=handles.kc_x;
Ti_x=handles.Ti_x;
Td_x=handles.Td_x;
kc_cv=handles.kc_cv;
Ti_cv=handles.Ti_cv;
Td_cv=handles.Td_cv;
t=handles.t;
dt=handles.dt;
dtplot=handles.dtplot;
```

```
tend=handles.tend;
tlastplot=handles.tlastplot;
firstloop=handles.firstloop;
secondloop=handles.secondloop;
ip=handles.ip;
M=handles.M;
vtotal=handles.vtotal;
fs_2=handles.fs_2;
fs_1=handles.fs_1;
fs=handles.fs;
f_2=handles.f_2;
f_1=handles.f_1;
x_2=handles.x_2;
x_1=handles.x_1;
finter_2=handles.finter_2;
finter_1=handles.finter_1;
CV_2=handles.CV_2;
CV_1=handles.CV_1;
CVs_2=handles.CVs_2;
CVs_1=handles.CVs_1;
CVs=handles.CVs;
ips=handles.ips;
ipe=handles.ipe;
item_viewing=handles.item_viewing;
tjump=handles.tjump;
twindow= handles.twindow;
tjumpend=handles.tjumpend;
tplotgap=handles.tplotgap;
fileupdate=handles.fileupdate;
itemswitch = handles.itemswitch;
% Individual controller details
citem = handles.citem;
cprop = handles.cprop;
cname = handles.cname;
```

```
% Slider data
Fb = handles.Fb;
Sp = handles.Sp;
Op = handles.Op;
Kc = handles.Kc;
Ti = handles.Ti;
Td = handles.Td;
%===========================================
% Time loop
min_ips=9999;
did_advance = FALSE;
while ((t<tend)&&(t<tjumpend))
   t=t+dt;
% Matrice interpretation
vtotal=M(:,17);
v=M(:,25);
y=M(:,18);
eps=M(:,19);
c=M(:,20);
for k=1:n
   Mk=M(k,9:12);
   Mkk=M(k,5:8);
   Mkkk=M(k,13:16);
   Mkkkk=M(k,21:24);
   f(4*k-3:4*k)=Mk';        % flow vector
   x(4*k-3:4*k)=Mkk';       % valve vector
   To(4*k-3:4*k)=Mkkk';     % To vector
   fmax(4*k-3:4*k)=Mkkkk';  % fmax vector
end
% flow distribution matrix
for i=1:n
  for j=1:n
     if M(j,1)==i
        Mde(i,j)=1;
    elseif M(j,2)==i
```

```
        Mdf(i,j)=1;
      elseif M(j,3)==i
        Mdg(i,j)=1;
      elseif M(j,4)==i
        Mdh(i,j)=1;
      end
    end
  end
  D(:,1)=Mde(:,1);
  D(:,2)=Mdf(:,1);
  D(:,3)=Mdg(:,1);
  for k=1:n-1
    D(:,4*k)=Mdh(:,k);
    D(:,4*k+1)=Mde(:,k+1);
    D(:,4*k+2)=Mdf(:,k+1);
    D(:,4*k+3)=Mdg(:,k+1);
  end
  D(:,4*n)=Mdh(:,n);
  %D=[0 0 0 0 0 0
  %  0 0 0 0 0 0
  %  0 1 1 0 0 0];
  E=eye(n);
  %E=[1 0 0       % exogenous matrix
  %  0 1 0      % can make it generic by E=eye(n)
  %  0 0 1];
  j=1;
  for i=1:n
    R(i,j)=1;          % outflow matrix
    R(i,j+1)=1;
    R(i,j+2)=1;
    R(i,j+3)=1;
    j=j+4;
  end
  T=R';        % doubling matrix
```

```
for i=1:4*n
  H(i,i)=-1/To(i);
  J(i,i)=fmax(i)/To(i);
end
% Creation of intermediate matrices and vectors
s=D*f;
phi(1:4*n,1)=f;
phi(4*n+1:5*n,1)=s;
for i=1:n
  Mf(i,1:4*n)=f';
  Meps(i,1:q)=eps';
end
Df=D.*Mf;
Rf=R.*Mf;
Eeps=E.*Meps;
% ----------------------------------------------------
% Initialisation
if ~firstloop
     fs_2=fs_1;
     fs_1=fs;
     f_2=f_1;
     f_1=f;
     x_2=x_1;
     x_1=x;
     finter_2=finter_1;
     finter_1=finter;
     if secondloop
       CV_2=CV;
       CV_1=CV;
       CVs_2=CV;
       CVs_1=CV;
    else
      CV_2=CV_1;
      CV_1=CV;
      CVs_2=CVs_1;
```

```
      CVs_1=CVs;
    end
    secondloop=FALSE;
else
    fs_2=f;
    fs_1=f;
    fs=f;
    f_2=f;
    f_1=f;
    x_2=x;
    x_1=x;
    finter_2=finter;
    finter_1=finter;
end
% ------------------------------------------------
% OPENLOOP equations
% Creation of vectors and matrices
for i=1:n
   My(i,i)=y(i);
   Mv(i,i)=1/(max(MINVOL,v(i)));
end
L(1:n)=v;
L(n+1:2*n)=y;
L(2*n+1:6*n)=f;
N(1:q)=eps;
N(q+1:2*q)=c;
N(2*q+1:2*q+4*n)=x;
A(1:n,2*n+1:6*n)=D-R;
A(n+1:2*n,n+1:2*n)=Mv*(Df*T-Rf*T);
A(n+1:2*n,2*n+1:6*n)=Mv*(My*R-My*D);
A(2*n+1:6*n,2*n+1:6*n)=H;
B(1:n,1:q)=E;
B(n+1:2*n,1:q)=-Mv*My*E;
B(n+1:2*n,q+1:2*q)=Mv*Eeps;
B(2*n+1:6*n,2*q+1:2*q+4*n)=J;
```

```
%--------------------------------------------------------------------

% Creation of the CV expansion matrix

% ( done for the purpose of the L7 CC : control a summation of flows )

CVEM(1:n,1:n)=eye(n);

CVEM(n+1:2*n,n+1:2*n)=eye(n);

CVEM(2*n+1:6*n,2*n+1:6*n)=eye(4*n);

% creation of the K matrix

if r~=0

   K=zeros(r,4*n);

   K(1,20)=1;

   K(1,21)=1;

  for i=1:r

     matf(i,:)=f';

  end

  Kk=K.*matf;

  ftot(1:r)=0;

  for i=1:r

     for j=1:4*n

        ftot(i)=ftot(i)+Kk(i,j);

     end

  end

  for i=1:r

     Kkk(i,:)=Kk(i,:)/max(ftot(i),SMALL);

  end

  CVEM(6*n+1:6*n+r,n+1:2*n)=Kkk*T;

end

% Creation of the control selection matrix

CSM(1,5)=1;

CSM(2,7)=1;

CSM(3,8)=1;

CSM(4,9)=1;

CSM(5,9)=1;

CSM(6,10)=1;

CSM(7,11)=1;

CSM(8,12)=1;
```

```
CSM(9,13)=1;
CSM(10,13)=1;
CSM(11,14)=1;
CSM(12,15)=1;
CSM(13,16)=1;
CSM(14,17)=1;
CSM(15,18)=1;
CSM(16,19)=1;
CSM(17,20)=1;
CSM(18,1)=1;
CSM(19,2)=1;
CSM(20,3)=1;
CSM(21,4)=1;
CSM(22,151)=1;
% controlled variables vector CV
 CV=CSM*CVEM*L;
CVs(22)= min(CVs(22), VCSP(5));  % cannot be higher - works on dilution
% advanced control equation
alpha1=0.1;  % for imperfect derivative
Mdt1=dt*ones(n_adv_cont_var,1);
beta1=Mdt1./Ti_cv;
gamma1=Mdt1./Td_cv;
for i=1:n_adv_cont_var
   c0(i,i)=alpha1+gamma1(i);
   c1(i,i)=-(2*alpha1+gamma1(i));
   c2(i,i)=alpha1;
   d0(i,i)=-kc_cv(i)*(alpha1+gamma1(i)+(alpha1*beta1(i))+(beta1(i)*gamma1(i))+1);
   d1(i,i)=kc_cv(i)*(2*alpha1+gamma1(i)+(alpha1*beta1(i))+2);
   d2(i,i)=-kc_cv(i)*(alpha1+1);
end
if ~firstloop
  for i=1:n_adv_cont_var
     g(i,1)=CV(i)-CVs(i);
     g_1(i,1)=CV_1(i)-CVs_1(i);
     g_2(i,1)=CV_2(i)-CVs_2(i);
```

```
  end
finter=inv(c0)*(-c1*finter_1-c2*finter_2+d0*g+d1*g_1+d2*g_2);
end
% clipping
%finter_tot=max(min(finter_max,finter_tot),zeros(n_cv,1));
finter=max(finter,zeros(n_adv_cont_var,1));
finter(1)=min(finter(1),1170); % finter (1) is taken out as it is used in the optimisation process
finter_real=finter(2:n_adv_cont_var);
% back to the flow setpoints
% Creation of the reverse operation matrix
ROM(20,1)=1;
ROM(25,2)=-1;
ROM(26,2)=1;
ROM(31,3)=1;
ROM(33,4)=1;
ROM(38,5)=1;
ROM(44,6)=1;
ROM(47,7)=1;
ROM(39,8)=1;
ROM(52,9)=1;
ROM(53,10)=0.7;
ROM(54,10)=0.3;
ROM(48,11)=1;
ROM(49,12)=1;
ROM(45,13)=1;
ROM(69,14)=1;
ROM(64,15)=1;
ROM(79,16)=1;
ROM(23,16)=0.05;
ROM(24,16)=0.95;
ROM(4,17)=1;
ROM(8,18)=1;
ROM(12,19)=1;
ROM(16,20)=1;
%#### ROM(82,17)=1;
```

```
%#### ROM(86,18)=1;
%#### ROM(90,19)=1;
%#### ROM(94,20)=1;
ROM(21,21)=1;
temp=SMALL*ones(size(finter_real,1),size(finter_real,2));
fs_offset(25)=fmax(25);                    % to be add by user
if ~firstloop
   fs_inter=ROM*(finter_real+temp)+fs_offset;           % no negative values for finter
end
for i=1:4*n
  ab=fs_inter(i);
  if ab~=0
    fs(i)=ab;
  end
end
```

% clipping

```
fs=max(min(fmax',fs),zeros(4*n,1));          % refer this to the table
%--------------------------------------------------------------------
```

% OPTIMISER to handle BLENDING CHEST CONSTRAINTS

% fopt and wopt initialisation

```
topt(1)=fs(82);
topt(2)=fs(86);
topt(3)=fs(90);
topt(4)=fs(94);
topt(5)=fs(97);  % #### force these to start with no flow .. seems to avoid local minima this way
topt(6)=fs(98);
topt(7)=fs(99);
topt(8)=fs(100);
```

% x intermediate calculation

```
xint(1)=y(21);
xint(2)=y(22);
xint(3)=y(23);
xint(4)=y(24);
xint(5)=y(25);
xint(6)=y(25);
```

```
xint(7)=y(25);

xint(8)=y(25);

xint(9)=y(1);

xint(10)=y(2);

xint(11)=y(3);

xint(12)=y(4);

xint(13)=y(7);

xint(14)=y(5);

VCSP(4)=finter(1);

% variables optimised

toptmax(1)=fmax(82);

toptmax(2)=fmax(86);

toptmax(3)=fmax(90);

toptmax(4)=fmax(94);

toptmax(5)=fmax(97);

toptmax(6)=fmax(98);

toptmax(7)=fmax(99);

toptmax(8)=fmax(100);

toptmin(:)=0;

% remember that we do not know the supply consistencies, so must estimate

% them backwards from the value after the CC dilution on each stream

if firstcall

   ysupest = 2.0*ones(4,1);  %Set first estimate of unknown supply consistencies all to 2%

   firstcall = 0;

else

   ysup_intantaneous=ysupest;

   ysup_intantaneous(1)= ( ( max(SMALL, f(82))+f( 97) )*xint( 9) - f( 97)*xint( 5) )/ max(SMALL, f(82));  % mass balance for
1st supply & dilution

   ysup_intantaneous(2)= ( ( max(SMALL, f(86))+f( 98) )*xint(10) - f( 98)*xint( 6) )/ max(SMALL, f(86));  % mass balance for
2nd supply & dilution

   ysup_intantaneous(3)= ( ( max(SMALL, f(90))+f( 99) )*xint(11) - f( 99)*xint( 7) )/ max(SMALL, f(90));  % mass balance for
3rd supply & dilution

   ysup_intantaneous(4)= ( ( max(SMALL, f(94))+f(100) )*xint(12) - f(100)*xint( 8) )/ max(SMALL, f(94));  % mass balance
for 4th supply & dilution

   % constrain it

   ysup_intantaneous= max(0,min(maxconsistency,ysup_intantaneous));
```

```
  if secondcall
    ysupest = ysup_intantaneous;
    secondcall = 0;
  else
    %Normal route
    ysupest= alpha_consistency_estimation*ysup_intantaneous + (1-alpha_consistency_estimation)*ysupest;
  end
end
% set up weights with special treatment for the total flow control
weight(1:5) = psi(1:5);
weight(4) = psi(4)*(1+psi(6)*abs(CV(1)-CVs(1))); % In the case of the FL requirement, increase if LC is having trouble
weight(6:9) = psi(7:10);
delta=0.1;
Jopt=1e+15;
% Fix allowed explorating steps
FAES=0.001;
% recalculated parameters---------------
% number of optmisation steps
itermax=1000;
for optsteps=1:itermax
  Joptlast=Jopt;
    for i=1:8
    for sense=-1:2:1
      topttemp=topt;
      topttemp(i)=topt(i)+sense*FAES*toptmax(i);
      topttemp = max(toptmin,min(toptmax,topttemp)); % do the whole vector every time
      VC = param_eval_static(topttemp,xint,ysupest);
      % objective function
      J=0;
      for h=1:5
        Jweight(h)=weight(h)*([VC(h)-VCSP(h)]^2); % ####MM031113 need to increase FC weight if LC is far from
setpoint
        J=J+Jweight(h);
      end
      for h=6:9
        Jweight(h)=weight(h)*([VC(h)-CSP(h-5)]^2); %
```

```
          J=J+Jweight(h);
        end
        if J<Jopt
          Jopt=J;
          iopt=i;
          senseopt=sense;
        end
      end
    end
    if Jopt<Joptlast                % assume no local minima
      topt(iopt)=topt(iopt)+senseopt*FAES*toptmax(iopt);
    else
      break;
    end
  end
  if optsteps>=itermax
    disp('########### ERROR : unconverged! ##############');
    halt;
  end
  % NOTE: The VC above is merely what would happen at steady-state - NOTE
  % actual VC calculation below
  % Back to the closed loop - optimised flows
  fs(82)=topt(1);
  fs(86)=topt(2);
  fs(90)=topt(3);
  fs(94)=topt(4);
  fs(97)=topt(5);
  fs(98)=topt(6);
  fs(99)=topt(7);
  fs(100)=topt(8);
  %----------------------------------------------------------------
  % clipping
  fs=max(min(fmax',fs),zeros(4*n,1));        % refer this to the table
  % Basic flow control
```

```
alpha=0.1;  % for imperfect derivative

Mdt=dt*ones(4*n,1);

beta=Mdt./Ti_x;

gamma=Mdt./Td_x;

for i=1:4*n

   a0(i,i)=alpha+gamma(i);

   a1(i,i)=-(2*alpha+gamma(i));

   a2(i,i)=alpha;

   b0(i,i)=-kc_x(i)*(alpha+gamma(i)+(alpha*beta(i))+(beta(i)*gamma(i))+1);

   b1(i,i)=kc_x(i)*(2*alpha+gamma(i)+(alpha*beta(i))+2);

   b2(i,i)=-kc_x(i)*(alpha+1);

end

% basic control equation

e=f-fs;

e_1=f_1-fs_1;

e_2=f_2-fs_2;

x=inv(a0)*(-a1*x_1-a2*x_2+b0*e+b1*e_1+b2*e_2);

% clipping

x=max(min(ones(4*n,1),x),zeros(4*n,1));          % refer this to the table

%-------------------------------------------------------------------

% link between the controllers and the open loop function

N(2*q+1:2*q+4*n)=x;    % add on the 28th of may

%-------------------------------------------------------------------

  % INTEGRATION

  % Equation to be solved

    % alphastar=expm(alpha*dt);

    % betastar=(alphastar-eye(itotal1))*inv(alpha)*beta;

    iexpm_max=1000;

     tol=0.0000001;

  % for singular A use series to find "expmAdt_IdivA" = [expm(A*dt)-I]*A^-1

  change=99;

  expmAdt_IdivA=dt*eye(6*n,6*n);

  changemat=dt*eye(6*n,6*n);

  Adt=A*dt;

  i=1;
```

```
  while change>tol
    i=i+1;
    changemat=(changemat*Adt)/i;
    change=sum(sum(abs(changemat))); % makes a 1-by-n vector with the sum of the columns as its entries
    expmAdt_IdivA=expmAdt_IdivA+changemat;
    if i>iexpm_max
      stop;
    end
  end
  % Now integrate using matrix exponential
  At=expmAdt_IdivA*A+eye(6*n,6*n);
  Bt=expmAdt_IdivA;
  Astar=At;
  Bstar=Bt*B;
  %Astar=expm(A*dt);
  %Bstar=(Astar-eye(4*n))*inv(A)*B;
% --------------------------
L=Astar*L+Bstar*N;
v=max(min(M(:,17),L(1:n)),zeros(n,1));
y=max(zeros(n,1),L(n+1:2*n));
f=max(min(L(2*n+1:6*n),fmax'),zeros(4*n,1));
%f=L(2*n+1:4*n);
% back to the M matrix
M(:,25)=v;
M(:,18)=y;
for k=1:n
  M(k,5) =x(4*k-3);
  M(k,6) =x(4*k-2);
  M(k,7) =x(4*k-1);
  M(k,8) =x(4*k-0);
  M(k,9) =f(4*k-3);
  M(k,10)=f(4*k-2);
  M(k,11)=f(4*k-1);
  M(k,12)=f(4*k-0);
end
```

```
% Effective conditions at machine chest input now
f_eff=zeros(4,1);
c_eff=zeros(4,1);
f_eff(1)=f(4);
f_eff(2)=f(8);
f_eff(3)=f(12);
f_eff(4)=f(16);
c_eff(1)=y(1);
c_eff(2)=y(2);
c_eff(3)=y(3);
c_eff(4)=y(4);
VC_eff = effective_value(f_eff,c_eff);
VC(1:5)=VC_eff;
VC(6:9)=c_eff(1:4);
%---------------------------------------------------
% plotting loop
if t-tlastplot >= dtplot
  if ip<min_ips
     min_ips=max(ip,1);
  end;
  ip=ip+1;
  ips= min(max(floor(ip-twindow/dtplot),1),min_ips);
  ipe=ip;
  tlastplot=tlastplot+dtplot;
  savetime(ip)=t;
  for i=1:n
     savee(ip,i)=M(i,9);
     savef(ip,i)=M(i,10);
     saveg(ip,i)=M(i,11);
     saveh(ip,i)=M(i,12);
     savev(ip,i)=M(i,25);
     savey(ip,i)=M(i,18);
     savexe(ip,i)=M(i,5);
     savexf(ip,i)=M(i,6);
     savexg(ip,i)=M(i,7);
```

```
    savexh(ip,i)=M(i,8);
  end
  for i=1:4*n
    savefs(ip,i)=fs(i);
  end
  for i=1:n_adv_cont_var
    saveCV(ip,i)=CV(i);
    saveCVs(ip,i)=CVs(i);
  end
  % convert it immediately
  saveCV(ip,:)=saveCV(ip,:)*reconv;
  saveCVs(ip,:)=saveCVs(ip,:)*reconv;
   for i=1:9
    saveVC(ip,i)=VC(i);
  end
  for i=1:5
    saveVCSP(ip,i)=VCSP(i);
  end
  for i=1:4
    saveCSP(ip,i)=CSP(i);
  end
  for i=1:10
    savepsi(ip,i)=psi(i);
  end
  saveoptsteps(ip,1)=optsteps;
  saveJopt(ip,1)=Jopt;
end
firstloop=FALSE;
did_advance=TRUE;
end % of while
```

Appendix D: Matlab software

Matlab, which stands for MATrix LABoratory, is an interactive software package for numerical computations and graphics. It has been edited by The Mathworks since 1984 and, since it was first released more than 15 years ago, it has been adopted as a standard for technical computing by major companies and research laboratories. An estimated 400000 engineers use this software regularly worldwide. The latest version that has been released, and is used in this work is Matlab 6.5 Release 13.

Matlab is a great programming tool with a friendly environment. As explained on the official website (http://www.mathworks.com), "Matlab is one of the fastest and most enjoyable ways to solve problems numerically. As the name suggests, Matlab is especially designed for matrix computations. It can be used in place of other languages like C and C++, with equal performance but less programming. For instance, the Matlab language enables the easy manipulation of scalars, vectors and matrices. In addition, the software has a variety of powerful graphical capabilities."

Matlab offers several non-negligible advantages: "Matlab can be accessed either by entering commands directly after the prompt (in that case it is used interactively) or by defining scripts (in which case, the information is collected in separate programs known as "M-files", nothing more than a text file with the extension ".m"). Those scripts can be programs or functions with special parameters. The functions are very convenient since every user can extend the capabilities of Matlab to his own domain of application. Because the syntax for using Matlab interactively is the same for writing programs, a code can be quickly converted into a reusable, automated analysis routine. Unlike most traditional languages, Matlab gives the freedom to focus on technical concepts rather than on programming details like memory management and variable declarations. Furthermore, M-files require no compiling or linking, which allows editing and step-by-step debugging of a program without having to leave Matlab."

Appendix E: PID controllers

The job of the controller is to compare the process signal from the transmitter with the setpoint signal and to send out an appropriate signal to the control valve. Analogue controllers use continuous electronic or pneumatic signals. The controllers see transmitter signals continuously, and control valves are changed continuously. Digital computer controllers are discontinuous in operation, looking at a number of loops sequentially. Each individual loop is only looked at once every sampling period.

There are three basic types of controllers that are commonly used for continuous feedback control.

- **Proportional action**

A proportional-only feedback controller changes its output signal (*CO*) in direct proportion to the error signal (*e*), which is the difference between the setpoint (*SP*) and the process measurement signal (*PM*) coming from the transmitter (eq E-1).

$$CO = bias \pm K_c(SP - PM) \qquad \textbf{(E-1)}$$

The bias signal is a constant and is the value of the controller output when there is no error. The K_c is called the controller gain. The larger the gain, the more the controller output will change for a given error. The gain on the controller can be made either positive or negative by setting a switch in an analog controller or specifying the desired sign in a digital controller. A positive gain results in the controller output decreasing when the process measurement increases. This increase-decrease action is called a *reverse-acting* controller. For a negative gain the controller output increases when the process measurement increases, and this is called a *direct-acting* controller.

- **Integral action (reset)**

The proportional action moves the control valve in direct proportion to the magnitude of the error. The integral action moves the control valve based on the time integral of the error (eq E-2).

$$CO = bias + \frac{1}{\tau_i}\int_0^t e(t)dt \qquad \textbf{(E-2)}$$

τ_i is the integral time or reset time.

Two scenarios can therefore occur. If there is no error ($e = 0$), the controller output does not move. As the error goes positive ($e > 0$) or negative ($e < 0$), the integral of the error drives the controller output either up or down, depending on the action (reverse or direct) of the controller. The basic purpose of the integral action is to drive the process back to its setpoint when it has been disturbed. A proportional controller will not usually return the controlled variable to the setpoint when a permanent load or setpoint disturbance occurs. This permanent error (*SP - PM*) is called the *steady state error* or *offset*. Integral action reduces the offset to zero. The integral action generally degrades the dynamic response of a control loop. It makes the control loop more oscillatory and moves it toward instability. But the integral action is usually needed if it is desirable to have zero offset. (Luyben, 1990)

- **Derivative action**

The purpose of the derivative action is to anticipate where the process is heading by looking at the time rate of change of the controlled variable (its derivative) (eq E-3).

$$CO = bias + \tau_d \frac{de}{dt} \qquad \textbf{(E-3)}$$

τ_d is the derivative time.

Generally the derivative action improves the dynamic response however it can make the output signal noisy, especially if the feedback signal is noisy, which can be undesirable.

The three actions described above are used individually or combined in commercial controllers. 60% of all the controllers are PI (proportional integral), 20% are PID (proportional integral derivative) and 20% are P only (proportional).

- **Practical aspects**

The system obtained when a PID regulator is connected to a linear system is well understood and can be analysed with great precision. When implementing a PID regulator it is necessary, however, to consider many issues like manual/automatic transfer, bumpless parameter changes, reset windup and non-linear output. Several PID regulators may also be connected via logical selectors. The systems obtained are then non-linear and the linear analysis of the ideal case is of limited value. Since the non-linear modifications give substantial improvements in performance they are widely used although they are poorly understood theoretically and not widely publicised (Astrom, 1983).