

**DYNAMIC REOPTIMIZATION AND CONTROL UNDER
SHUTDOWN CONDITIONS**

**DYNAMIC REOPTIMIZATION AND CONTROL
UNDER SHUTDOWN CONDITIONS**

by

ZHIWEN CHONG, B.Eng.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Applied Science

McMaster University

© Copyright by Zhiwen Chong, August 2006

MASTER OF APPLIED SCIENCE (2006)
(Chemical Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE: Dynamic Reoptimization and Control
 Under Shutdown Conditions
AUTHOR: Zhiwen Chong, B.Eng.
 (McGill University, Montreal, Quebec, Canada)
SUPERVISOR: Dr. Christopher L.E. Swartz
NUMBER OF PAGES: xv, 122

Abstract

A systematic control strategy is proposed for optimal operation of plants containing integrated process units in the event of unit shutdowns or failures. This entails manipulating the degrees-of-freedom available during and after a shutdown in such a way that production is restored in a cost-optimal fashion while meeting all safety and operational constraints. In this work, we investigate the problem of coordinating various buffer tanks and recycle streams during the period of transition to minimize production losses. The problem is cast in a dynamic optimization framework.

The case studies in our work are based on a simulation of a Kraft pulp mill where a process unit is shut down and taken off-line for a period of time, and is subsequently restored. Based on an estimate of the downtime, our proposed control system then computes and implements a set of optimal control trajectories that accommodates the shutdown.

This work extends prior studies ([8], [24]) by considering in addition two key issues – inclusion of feedback mechanisms to counter uncertainty, and the development of a software-based modeling tool. The downtime estimate is a crucial parameter for performing the control calculations. This estimate will usually be based on past operational experience or on direct information about the prognosis of the shutdown. In practice, this estimate will not correspond exactly to the actual downtime; thus we consider re-optimization based on revised downtime estimates. The remainder of the trajectory is re-optimized from the current state of the system, and the controller performs what is essentially a mid-course correction. This feedback approach has considerable advantages over a multi-scenario optimization approach for dealing with uncertainty in the estimated downtime, in that the resulting control trajectories are less conservative. The performance of this re-optimization scheme is studied in this work under various failure scenarios.

Uncertainty also exists due to model imperfections and unmeasured disturbances. We therefore account for this uncertainty by considering the trajectory optimization problem within an integrated nonlinear predictive control framework. The type of operation under consideration (response to partial shutdown conditions) is inherently unsteady in nature, and the control horizon as measured from the onset of the failure is fixed. Among the distinctive features of the controller are: a shrinking prediction

horizon, an economics-driven objective function and the use of a nonlinear differential-algebraic equation-based model. The controller is also “event-cognizant” in the sense that explicitly known future events such as shutdowns and startups can be specified and accommodated within the prediction algorithm. Case studies demonstrating the performance of the overall feedback strategy are presented.

In the course of this work, we developed a specialized software-based modeling tool that simplifies the tasks of representing, discretizing, and solving dynamic optimization problems. The main component of this tool is a domain-specific language named MLDO (Modeling Language for Dynamic Optimization). This tool is tailored to the representation of constructs specific to the dynamic optimization problem domain. Models written in MLDO are used as precursors for generating intermediate AMPL-based models (discretized using an implicit Runge-Kutta method), which are subsequently solved using a large-scale nonlinear optimizer, IPOPT.

Acknowledgements

First and foremost, the author of this work wishes to express profound appreciation to his thesis advisor, Dr. Christopher Swartz for his support, advice and guidance throughout the course of the research work that has come to culmination in this thesis. This thesis would not have been possible without his direction and vision.

The author wishes to acknowledge the McMaster Advanced Control Consortium and the Department of Chemical Engineering at McMaster University for financial support. He also wishes to thank his colleagues in the control group for their friendship and moral support, particularly during the difficult periods of his life.

The author owes a debt of gratitude to his parents, Chong Ah Chye and Chan Wee Siew, for making tremendous personal sacrifices in order to provide him with an education. This debt cannot ever be repaid.

The greatest thanks are due to God, who has sustained the author through all in spite of his failings, and it is to Him the author owes his existence. In the words of the English journalist G.K. Chesterton, "If my children wake up on Christmas morning and have someone to thank for putting candy in their stocking, have I no one to thank for putting two feet in mine?"

All errors and shortcomings are the author's alone.

Acknowledgements

First and foremost, the author of this work wishes to express profound appreciation to his thesis advisor, Dr. Christopher Swartz for his support, advice and guidance throughout the course of the research work that has come to culmination in this thesis. This thesis would not have been possible without his direction and vision.

The author wishes to acknowledge the McMaster Advanced Control Consortium and the Department of Chemical Engineering at McMaster University for financial support. He also wishes to thank his colleagues in the control group for their friendship and moral support, particularly during the difficult periods of his life.

The author owes a debt of gratitude to his parents, Chong Ah Chye and Chan Wee Siew, for making tremendous personal sacrifices in order to provide him with an education. This debt cannot ever be repaid.

The greatest thanks are due to God, who has sustained the author through all in spite of his failings, and it is to Him the author owes his existence. In the words of the English journalist G.K. Chesterton, "If my children wake up on Christmas morning and have someone to thank for putting candy in their stocking, have I no one to thank for putting two feet in mine?"

All errors and shortcomings are the author's alone.

Table of Contents

1	Introduction	1
1.1	Objective	1
1.2	Main Contributions	2
1.3	Thesis Overview	2
2	Literature Review	4
2.1	Process Description - Kraft Mill Fiber Line	4
2.2	Use of Buffer Capacities for Handling Unit Shutdowns	6
2.3	Dynamic Optimization	8
2.3.1	Sequential Method	9
2.3.2	Simultaneous Method	10
2.4	Orthogonal Collocation on Finite Elements (OCFE)	11
2.5	Feedback using Predictive Control Algorithms	15
2.5.1	Model Predictive Control	16
2.5.2	Control of Transient Processes using MPC	17

2.5.3	Incorporating Economics into the Control Algorithm	18
3	Development of the Model	20
3.1	A Modeling Language for Dynamic Optimization (MLDO)	20
3.1.1	Features of MLDO	22
3.1.2	Design of MLDO	25
3.1.3	The Code Generation Process	32
3.2	Modeling the Kraft Mill Fiber Line	35
3.2.1	Conventions and Nomenclature	36
3.2.2	Kraft Digester	36
3.2.3	Buffer Tank Units	39
3.2.4	Knotting Department	41
3.2.5	Washing Department	43
3.2.6	Seal Tank and Splitter	45
3.2.7	Screening Department	46
3.2.8	Delignification Department	48
3.3	Formulation Issues and Modeling Techniques	52
3.3.1	Objective Functionals with Integral Terms	52
3.3.2	Numerical Issues	53
3.4	Chapter Summary	54
4	Dynamic Optimization under Shutdown Conditions	56

4.1	Introduction	56
4.2	Dynamic Optimization Problem Formulation	58
4.2.1	Assumptions	61
4.3	Case Study 1: Individual Shutdowns in Different Units	62
4.4	Uncertainty in Downtime Estimate, d_{est}	66
4.4.1	Implementing the initial trajectory with naïve adjustments	70
4.4.2	Optimization Under Uncertainty using a multi-scenario approach	72
4.4.3	Trajectory Re-optimization using Feedback of Downtime Estimate	74
4.5	Computational Issue: Avoiding Control Policies that Induce Shutdowns	84
4.5.1	Barrier method	85
4.5.2	Case Study: Penalizing Induced Shutdowns	87
4.6	Chapter Summary	91
5	Integrated Predictive Control Framework for Shutdowns	92
5.1	Applying Predictive Control to the Shutdown Problem	92
5.2	Features of the Integrated Framework	94
5.2.1	Nonlinear Control of Transient Processes	94
5.2.2	Economics-based objective function	95
5.2.3	Events	95
5.3	Algorithm and Software Implementation	96
5.4	Interaction with Local Control Loops	98
5.5	State Estimation of a DAE System	99

5.6	Case Studies	102
5.6.1	Case 1: Implementing the Optimal Control Policy from Dynamic Optimization (Nominal Case) with the Predictive Controller	102
5.6.2	Case 2: Step Disturbance in Chip Feed	104
5.6.3	Case 3: Parametric Model Mismatch	106
5.6.4	Case 4: Reoptimization upon feedback of downtime estimate	108
5.7	Chapter Summary	110
6	Conclusion	111
6.1	Recommendations for Further Work	113
	References	115
A	Model of Kraft Paper Mill – Fiber Line	123
A.1	Model - digester	123
A.2	Model - blowtank	126
A.3	Model - hiq	129
A.4	Model - jonsson	131
A.5	Model - header	133
A.6	Model - drumwasher	135
A.7	Model - sealtank	137
A.8	Model - fullscreen	139
A.9	Model - o2feedpress	141

A.10 Model - generictank	142
A.11 Model - deligmixer	143
A.12 Model - reactor	146
A.13 Model - posto2washer	148
B Characterizing the fixed cost of shutdowns in an NLP	155
B.1 Counting the number of shutdowns	155

List of Figures

2.1	Example of OCFE on $\dot{x} = f(x, t)$, nFE = 5, nCOL = 2.	12
3.1	Model concatenation with MLDO.	22
3.2	Use of Sparklines for displaying variable trajectories.	26
3.3	MLDO-generated documentation (using L ^A T _E X).	31
3.4	Simplified diagram of the code generation process.	34
3.5	Simplified schematic of plant model.	35
3.6	Digester model.	37
3.7	(a) Schematic of integrated tank system; (b) Expanded diagram of the dynamic section of the tank.	39
3.8	Hi-Q knotter.	42
3.9	Washing department.	43
3.10	Vacuum drum unit washer.	44
3.11	Seal tank and splitter.	46
3.12	Screening department.	47
3.13	Delignification department.	48

3.14	Delignification mixer.	49
3.15	Delignification reactor.	51
4.1	Phases of the shutdown process – steady-state, shutdown ($t_{start}-t_{end}$), restoration ($t_{end}-t_{res}$), steady-state.	60
4.2	Simplified schematic of plant model.	62
4.3	Failure in digester ($digester(1).F_{in1} = 0$ for $t \in [2, 8]$).	67
4.4	Failure in Hi-Q knotter ($hiq(1).F_{in1} = 0$ for $t \in [2, 8]$).	68
4.5	Failure in O ₂ -delignification reactor ($reactor(1).F_{in1} = 0$ for $t \in [2, 8]$).	69
4.6	Case 1 ($d_{est} < d_{act}$).	71
4.7	Case 2 ($d_{est} > d_{act}$), Subcase 1.	71
4.8	Case 2 ($d_{est} > d_{act}$), Subcase 2.	71
4.9	Comparison between ideal trajectories (dotted lines) and trajectories obtained from multiscenario optimization (solid lines).	75
4.10	Case 1: Reoptimization at various stages, given that $d_{act} > d_{est}$	76
4.11	Case 2: Reoptimization at various stages, given that $d_{est} > d_{act}$	77
4.12	Trajectories for reoptimization 25% into estimated shutdown, where $d_{act} > d_{est}$	78
4.13	Trajectories for reoptimization 100% into estimated shutdown, where $d_{act} > d_{est}$	79
4.14	Trajectories for reoptimization 25% into estimated shutdown, where $d_{est} > d_{act}$	80
4.15	Trajectories for reoptimization 75% into estimated shutdown, where $d_{est} > d_{act}$	81

4.16 Exponential barrier function, with $A = 1$, $k = 10$ and $F_{shut} = 0$	86
4.17 No penalty method applied to avoid induced shutdowns.	89
4.18 Exponential penalty applied to avoid induced shutdowns.	90
5.1 Hierarchy of control, integrated dynamic optimization and control. . .	93
5.2 Software implementation of predictive control algorithm.	97
5.3 Simplified schematic of plant model.	102
5.4 Case Study 1: Dynamic Optimization Solution, closed loop.	103
5.5 Case Study 2: Disturbance in the feed composition.	105
5.6 Case Study 3: Plant-model mismatch in value of consistency in vacuum washer outlet.	107
5.7 Case Study 4: Reoptimization upon update of downtime estimate. . .	109
A.1 Digester Model	123
A.2 (a) Schematic of integrated blowtank system; (b) Expanded diagram of the dynamic section of the blowtank.	126
A.3 Hi-Q knotter	129
A.4 Jonsson knotter	131
A.5 Header box	133
A.6 Vacuum Drum Unit washer	135
A.7 Seal Tank	137
A.8 Screening Department	139
A.9 O ₂ Feed Press	141

A.10 Generic Tank	142
A.11 Delignification mixer	143
A.12 Delignification Reactor	146
A.13 Post O ₂ Washer	148
A.14 Overall system up to washers	152
A.15 Screening Department	153
A.16 Delignification	154
B.1 Modified Hyperbolic Switching Function with arctangent function, given $\gamma = 50$ and $F_{shut} = 2$	158
B.2 Procedure for deriving constraints to count shutdowns	160

List of Tables

4.1	Prices of Materials, based on ([8], [27], [35])	59
4.2	Manipulated Variables	63
4.3	Tank Capacities, Nominal Levels And Upper/Lower Bounds	63
4.4	Plant Profits and Throughputs under Unit Failures	66
4.5	Simulation Results for Trajectory Adjustment Method	70
4.6	Comparison of Profits from Multiscenario Optimization and Others	74
4.7	One-time Reoptimization at Various Stages into d_{est}	82
4.8	Comparison of Reoptimization and Multiscenario Optimization	84
4.9	Penalty Method for Induced Shutdowns	87
5.1	Composition of Chip Feed Stream	104
5.2	Plant-Model Mismatch	108

Chapter 1

Introduction

1.1 Objective

Unit shutdowns in a plant frequently have an adverse effect on the operating economics of a plant, chiefly due to loss of production. One way of circumscribing the impact of a unit shutdown to a localized subset of a plant is to employ buffer capacities. Buffer capacities function to decouple one part of the plant from another, and by properly managing them one can minimize the losses incurred.

Various methods have been proposed to optimally coordinate buffers in a plant. This work extends prior work by Balthazaar [8] and Dubé [24], both of which take an open-loop dynamic optimization approach to tackling the problem of computing optimal trajectories for control. In our work, the effect of uncertainty in the downtime estimate used in optimization is studied with the view of finding a means of accommodating it within the problem formulation.

We also investigate the integration of dynamic optimization with a predictive control-type algorithm for handling shutdowns. The predictive controller provides feedback to counter the effects of plant-model mismatch and process disturbances that occur during the transient shutdown and restoration periods.

A subset of the Kraft Paper mill model was chosen as a basis for demonstrating

the above ideas in the presence of nonlinear dynamics and recycles. The model is composed of several departments, with main ones separated by buffer tanks.

1.2 Main Contributions

The major contributions of this thesis can be summarized as follows:

1. **Modeling Language for Dynamic Optimization (MLDO).** The models in this work were developed using a domain-specific language called MLDO, a software tool created to ease the task of representing and solving dynamic optimization problems. MLDO was used to produce discretized code using Orthogonal Collocation on Finite Elements (OCFE), among other things.
2. **Feedback of Downtime Estimate.** The effects of uncertainty in downtime estimate and the use of re-optimization was studied. A method for representing fixed costs triggered by events (shutdowns, in particular) was proposed.
3. **Predictive Control.** A study was done integrating dynamic optimization with predictive control, and applying it to the problem of unit shutdowns. The control algorithm includes various features such as a shrinking-horizon prediction horizon, economic objective function, and the ability to embed explicitly known events into the prediction model.

1.3 Thesis Overview

- **Chapter 2—Literature Review.**

This chapter begins with a process description, and is followed by a review of existing methodologies for handling shutdowns using buffer capacities, dynamic optimization and predictive control.

- **Chapter 3—Development of the Model.**

This chapter describes the modeling involved in this work, and the details of the modeling language used to develop the models. Some modeling and formulation issues are considered.

- **Chapter 4–Reoptimization.**

This chapter describes several ways of dealing with the uncertainty in the downtime estimate, and demonstrates that re-optimization is the most suitable method for addressing the problem. Case studies are presented.

- **Chapter 5–Integrated Predictive Control for Shutdowns.**

This chapter describes an integrated predictive control system applied to the problems of shutdowns. Issues of plant-model mismatch and process disturbances were considered.

- **Chapter 6–Conclusions.**

This chapter summarizes the findings on this work and provides recommendations for future work.

Chapter 2

Literature Review

If I have seen farther it is by standing on the shoulders of giants.

- Sir Isaac Newton

If I have not seen as far as others, it is because giants were standing on my shoulders.

- Hal Abelson, MIT computer science professor

This chapter aims to familiarize the reader with the body of work at large that is pertinent to this project. It begins with a précis of the fiber line of a Kraft mill, followed by a survey of the literature dealing with buffer-coordination techniques that have particular emphases on shutdown handling. The discipline of dynamic optimization is reviewed, and a detailed explanation of the method of orthogonal collocation on finite elements (OCFE) is presented. An outline of relevant topics in constrained predictive control follows, and the chapter concludes by examining the issues that surround the integration of predictive control and dynamic optimization.

2.1 Process Description - Kraft Mill Fiber Line

The Kraft pulping process is composed of various production departments that are separated by buffer capacities. The major departments considered in this work are digestion, knotting, washing, screening and delignification.

The Kraft process begins with digestion. The digester vessel is filled with wood chips and white liquor (a concentrated solution of Na_2S and NaOH) and heated according to

a predetermined schedule in a process known as "cooking". The wood chips eventually disintegrate into fibers, forming pulp. Lignin, the organic component that holds cellulose fibers in wood together, reacts with the white liquor and is solubilized. The contents in the vessel are maintained at typical conditions of 170°C for 2 hours [24], and then discharged into an adjacent tank, called a blowtank. During this reaction, the white liquor turns into black liquor. The black liquor produced is channeled to a chemical recovery system which regenerates a fraction of it into white liquor [70] and uses the rest as fuel for producing steam. The off-vapors are sent to a heat exchanger where it used to heat water for pulp washing. The cooked pulp is then subjected to various physical and chemical separation processes designed to remove unprocessable wood, residual black liquor and lignin.

The knotting department consists of knotter machines that function to remove undigested chips and ill-sized wood pieces known as "knots" which hinder downstream processing.

Following that, the pulp stream is directed to the washing department and undergoes a process known as brownstock washing (or simply, washing), where the residual black liquor is separated from the pulp in a carefully controlled process. This involves feeding the pulp into a series of counter-current vacuum drum washers, in which black liquor is displaced.

The washed pulp is then conveyed to the screening department for removal of "shives", that is, wood pieces whose sizes lie between that of processable pulp and knots. Typically, vibrating pressure screens are employed for this task.

At this stage, the pulp stream would have been adequately prepared for delignification—the process by which the remaining lignin in the pulp is removed. The pulp stream is first mixed with chemical streams containing caustic soda and magnesium sulfate. This mixture is then fed in a counter-current direction with respect to an oxygen stream running within an O₂-delignification reactor. A reaction occurs in which the lignin separates from the cellulose fibers in the pulp stream. The delignified pulp stream is then ready for bleaching, where the pulp is whitened in a chlorination reactor.

In the Kraft fiber line, there are a number of recycles. The rejects from the last knotter are usually recycled to the digester, and the liquor in the brownstock washing stage is normally recycled to the agitated bottoms section of the blowtank, and also to

the header box (a mixing unit) in the washing department. Detailed model equations describing this plant are presented in the next chapter.

We end this section by noting that there is a significant body of research related to the optimal operation, design and control of pulp and paper mills. Kayihan [47] discusses the benefits of taking an optimization-based approach with respect to describing, analyzing and improving the process systems and management structures in the pulp and paper industry. Castro and Doyle [17] describe a plantwide control design framework for pulp mills based on decentralized control and unit-based predictive control.

2.2 Use of Buffer Capacities for Handling Unit Shutdowns

One of the benefits of having judiciously placed buffer capacities in a plant is the mitigation of process variation propagation along a production line. These intermediate storage units are not only able to dampen the effects of short term fluctuations, they are also able to deal with larger processing disturbances such as unit shutdowns if they can be coordinated correctly.

In 1969, Pettersson [61] developed a scheme for coordinating production in a pulp and paper mill, which included a strategy for managing the buffer tanks. A system comprising nine processing units and ten buffer tanks was considered for optimization, resulting in a production scheme for coordinating the plant in such a way that capacity restrictions were not violated. An example with a shutdown in the evaporator system was shown. In a later publication, Pettersson [62] considered the problem of producing an optimal plantwide production scheme that could account for maintenance shutdowns, limited buffer capacity and steam restrictions. The optimal control problem was solved using a scheme based on Pontryagin's Maximum Principle [63].

Lee and Reklaitis [51] proposed a method for systematically utilizing buffer capacities to decouple upstream equipment failures from downstream processes and vice versa. Using Fourier series constructions, they derived a set of analytical expressions for determining the minimum volume of intermediate storage required as a function of frequency of failure and of failure durations. In a related paper, Lee and Reklaitis [50] noted that intermediate storage has the effect of introducing delays in product change-overs, thus with respect to specifying storage capacities there is considerable

economic benefit to specifying only as much as is absolutely required.

Dubé [24] investigated a buffer storage operation strategy that minimizes time away from normal operation and prevents departmental shutdowns. This strategy was applied to a highly integrated Kraft pulp mill with the view of maximizing production. The author addressed the issue of determining the longest feasible shutdown time (or “independence” time). The effect of preparation time on the production was also studied and the coordination of buffer capacities for handling planned and unplanned shutdowns was illustrated. A numerical optimization procedure was used in the solution process.

Huang et. al. [41] advocated the idea of using a dynamic optimization approach for general fault accommodation and control redesign, as opposed to the manual table-lookup approach typically adopted by operators. The authors categorized faults in three categories:

1. **Gross Parameter Changes in Model.** Changes to the parameters in the model that do not alter the model structurally, such as sudden drop in heat transfer coefficients.
2. **Structural Changes in Model.** Changes that alter a model’s structure, such as equipment failure, stuck valves, controller failures, leaky pipes and so on.
3. **Malfunctioning Sensors and Actuators.** Faults that lead to a constant bias, out-of-range failures or a fixed failure.

Case studies for the first and third types of faults were presented. Unit shutdowns were deemed fall into the second category. This idea was augmented and generalized in a follow-up publication [42], where the issue of unit shutdowns was briefly addressed. With regard to optimizing dynamic models with unit failure representations, the authors suggested first removing all equations related to the unit, followed by the activation of a set of discrete transition equations triggered using integer variables. The authors emphasized that the integer variables only act as transition conditions, and are directly prescribed by a fault detection module, hence the final model to be solved does not contain integer variables.

Allison [4] turned to an analytical approach for determining a policy for averaging the loads on a set of surge tanks in series during a transient event (which can be taken

to include unit shutdowns). In this scheme, the impact of a surge on a single tank is distributed across the plant to avoid upper and lower level constraints on buffer capacities. Optimal control theory was employed in the solution process.

Continuing along the line of Dubé's work, Balthazaar [8] considered both a pre-emptive and a reactive response to shutdowns in a Kraft paper mill. In the pre-emptive case, the control problem is solved assuming knowledge of the shutdown ahead of time (as is the case in a scheduled maintenance scenario), thus allowing the plant to take preparatory action in anticipation of the shutdown. In the reactive case, the plant shutdown occurs without warning, and the control system is expected to respond immediately. Balthazaar also examined the problem of determining default steady-state buffer tank levels that are optimal for handling shutdowns, using information on the likelihood of process unit failure. An economics-based objective function was employed in all the case studies.

2.3 Dynamic Optimization

A dynamic optimization problem can be qualitatively described in the following terms: given a dynamic model of a process, find a set of control actions that will extremize some specified performance criterion [65]. Dynamic systems (which includes chemical processes) are ordinarily modeled with a differential-algebraic equation (DAE) system. The methods and means for solving such systems are described below. A DAE optimization problem can be stated as follows:

$$\begin{aligned}
 & \min_{\mathbf{u}(t)} \Phi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \boldsymbol{\theta}, t_f) \\
 \text{s.t.} \quad & \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \boldsymbol{\theta}, t) = \mathbf{0} \\
 & \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \boldsymbol{\theta}, t) = \mathbf{0} \\
 & \mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \boldsymbol{\theta}, t) \leq \mathbf{0} \\
 & \mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U \\
 & \mathbf{z}_L \leq \mathbf{z}(t) \leq \mathbf{z}_U \\
 & \mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \\
 & \mathbf{x}(0) = \mathbf{x}_0
 \end{aligned}$$

where Φ = scalar objective function, \mathbf{x} = differential state vector, \mathbf{z} = algebraic state vector, \mathbf{u} = control vector, $\boldsymbol{\theta}$ = time-invariant parameter vector. L-subscripts represent lower bounds and U-subscripts upper bounds.

The dominant means of solving dynamic optimization problems in the 1960s-70s were indirect methods based on calculus of variations and optimal control theory. Pontryagin's Maximum Principle [63] uses first-order necessary conditions to locate the extremum of an objective functional. In unconstrained problems, this leads to a two-point boundary value problem that is solvable via a number of standard methods. For constrained problems however, additional complementarity conditions have to be satisfied and multipliers determined. This results in a combinatorial problem that is computationally challenging even for relatively small problems.

Direct methods on the other hand, seek to transform infinite dimension problems into finite dimensional nonlinear programs by parameterizing the continuous time profiles into discrete-time approximations. Direct methods generally fall under two categories, *sequential* and *simultaneous* methods. Sequential methods are those that discretize only the control profile, while simultaneous methods are those in which both the state and control profiles are discretized. These methods will be considered in turn below.

2.3.1 Sequential Method

The sequential method (also known as control vector parameterization) is a two-layer solution strategy. The nonlinear optimization and DAE integration steps are performed separately. The control profile is first parameterized using typically piecewise polynomial or constant functions. Given a set of initial values for the DAE and a starting control profile, the DAE model is integrated (using specialized DAE solvers such as DASSL or DASPK [15]) and the objective function value evaluated. Based on the objective value and sensitivity information obtained from the system, the optimizer tries to find a set of control actions that will improve on the current solution. This new set of control actions are then inserted into the DAE problem, and the integration is repeated. This iterative procedure continues until a stipulated termination condition is met.

One attractive advantage of the sequential method is that for problems in which the number of state variables far exceed the number of control variables (i.e. few degrees of freedom available), the nonlinear programming problem arising is usually manageably

small in size. Also, the sequential method does not succumb to discretization errors with respect to profiles of state variables, as the DAE solver usually uses variable step sizes when performing the integration.

However, in practice, sequential methods often incur significant computational expense from having to integrate a DAE system on every iteration [1]. Bloss et. al. [13] point out that the calculation of constraint gradients by finite differencing or sensitivity/adjoint equations also contributes greatly to computational load. Inefficiencies related to integration are particularly pronounced in stiff systems [73]. Biegler [12] states that the sequential method is only robust for systems with exclusively stable modes (defined as modes in which profiles remain bounded as the time goes to infinity); otherwise, the problem may fail prematurely due to integration failure at unstable intermediate points, even if a stable final solution exists.

Because the optimization iterations are separated from the integration, bounds on the state variables cannot be directly enforced in the optimization step. Various methods for getting around this problem have been proposed, ranging from forcing a constraint violation penalty to zero through an endpoint constraint [78] to using initial point solvers that respect path constraints [30]. Huang et. al. [43] presented a decomposition-based method where the state variables are partitioned into two subsystems using an efficient maximum traversal algorithm [25]. The goal is to isolate control variables and state variables affected by path constraints into a separate subsystem. This subsystem is discretized and solved in the NLP phase using the simultaneous method, and thus path constraints on the state profiles are enforced within the optimization framework. The rest of the system is solved using the sequential method. Two obvious advantages of this method are: 1) The size of the resulting NLP is moderately small; and 2) Path constraints on the state variables are handled naturally.

2.3.2 Simultaneous Method

In the simultaneous method, both state and control profiles are discretized, resulting in a large scale sparse NLP that may benefit from special solution strategies [11]. Of note is the fact that no explicit integration step is performed because both the integration and optimization problems are converged simultaneously at the solution point.

A widely-used method for performing full discretization is orthogonal collocation on finite elements (OCFE). OCFE converts the DAE system into an algebraic equation system that can be solved with a conventional NLP solver. We have elected to use this method in our work, and it is described in some detail in the next section.

Simultaneous methods enjoy several advantages over sequential methods. First, path constraints are handled naturally within the optimization formulation. Further, because the DAE system is only solved once, unstable or nonexistent intermediate solutions are bypassed [11].

There are however some disadvantages associated with simultaneous methods. As stated before, large scale NLPs arise from this method and they can be challenging to solve. Reduced space methods have been proposed to alleviate some of the difficulties related to large-scale problems. It is worth noting that large scale solvers that exploit the sparsity of the problem have been developed [81] and have proven to be successful in finding the solutions to these types of problems. Model reduction techniques that employ a combination of projections and system identification have also been proposed [72].

The other issue with simultaneous methods relates to stability problems that arise from high-index constraints and singular arcs. Several practical methods to address these issues have been proposed [12].

2.4 Orthogonal Collocation on Finite Elements (OCFE)

Orthogonal Collocation on Finite Elements (OCFE) is a weighted-residual method for transforming a DAE system into a set of algebraic equations that are admissible as constraints in an optimization problem. It has been demonstrated that OCFE essentially corresponds to a well-conditioned implicit Runge-Kutta method for integrating DAEs [20]. Implicit Runge-Kutta methods have been shown to have good numerical properties such as having matrices that possess smaller condition numbers and that are less susceptible to rounding errors [6].

The basic idea behind OCFE is the segmentation of the continuous-time solution profile of the DAE system into components called finite elements. Within each finite element, a residual function containing a polynomial approximation of the solution is

forced to be exactly zero at collocation nodes (Figure 2.1). These collocation nodes are chosen as roots of an orthogonal polynomial family, such as the Jacobi polynomials. Function (and in some cases, derivative) continuity is imposed at the boundaries of finite elements (continuity nodes, Figure 2.1). Details of this method can be found in Cuthrell and Biegler [20]. On a practical note, according to Vasantharajan and

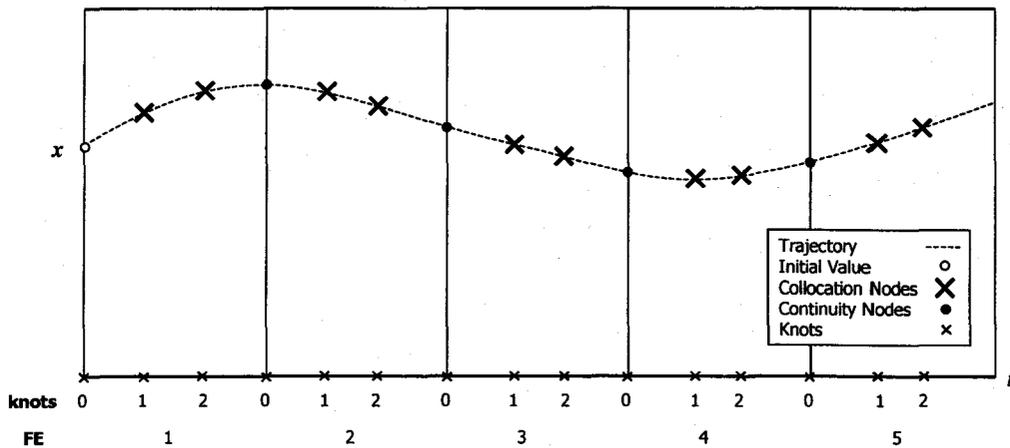


Figure 2.1: Example of OCFE on $\dot{x} = f(x, t)$, $nFE = 5$, $nCOL = 2$.

Biegler [77], two or three collocation points are generally sufficient for an accurate approximation, given a adequate number of finite elements. Approximation error can be further controlled using adaptive knot placement techniques [20].

Consider a differential algebraic equation system of the following form:

$$\frac{dx}{dt} = f(x(t), z(t), u(t)) \quad \text{with } x(t) = x_0 \quad (2.1)$$

$$h(x(t), z(t), u(t)) = 0 \quad (2.2)$$

where

x = vector of differential variables

z = vector of algebraic variables

u = vector of control input variables (decision variables in an optimization problem)

The collocation procedure is described below.

Time scaling equation

For convenience, the time ordinate is rescaled to the $[0,1]$ interval in each finite element

using the following equation:

$$\tau = \frac{t - t_{q-1}}{\delta} \quad \text{for } t_{q-1} \leq t \leq t_q, q = \{1, \dots, \text{nFE}\} \quad (2.3)$$

where q = finite element counter, t_{q-1} = left-hand side boundary of the current finite element q , δ = finite element length (assumed uniform), nFE = total number of finite elements. The scaled collocation points will be denoted as τ_i for $i = 1, \dots, \text{nCOL}$, where nCOL = the total number of collocation points per finite element. τ_0 is the time-scaled left-hand boundary point (i.e. $\tau_0 = 0$). All references to temporal variables henceforth will be made in terms of the variable τ .

Method of Ordinates: Lagrange Interpolation Polynomials

Each differential and algebraic variable is approximated as a polynomial. In lieu of the standard coefficients-based representation for polynomials, $P(x) = a_0 + a_1x + \dots + a_nx^n$ (method of coefficients), Lagrange interpolation polynomials are employed in what is known as the method of ordinates. The variables are represented thus:

$$\frac{d\mathbf{x}^{(q)}(\tau_i)}{d\tau} = \sum_{j=0}^{\text{nCOL}} \mathbf{x}_j^{(q)} \frac{d\phi_j(\tau_i)}{d\tau} \quad (2.4)$$

$$\mathbf{x}^{(q)}(\tau_i) = \sum_{j=0}^{\text{nCOL}} \mathbf{x}_j^{(q)} \phi_j(\tau_i) \quad (2.5)$$

$$\mathbf{z}^{(q)}(\tau_i) = \sum_{j=0}^{\text{nCOL}} \mathbf{z}_j^{(q)} \phi_j(\tau_i) \quad (2.6)$$

and

$$\phi_j(\tau_i) = D_j \prod_{\substack{l=0 \\ l \neq j}}^{\text{nCOL}} (\tau_i - \tau_l) \quad (2.7)$$

$$\frac{d\phi_j(\tau_i)}{d\tau} = D_j \sum_{\substack{k=0 \\ k \neq j}}^{\text{nCOL}} \prod_{\substack{l=0 \\ l \neq j, l \neq k}}^{\text{nCOL}} (\tau_i - \tau_l) \quad (2.8)$$

$$D_j = \prod_{\substack{l=0 \\ l \neq j}}^{\text{nCOL}} \frac{1}{(\tau_j - \tau_l)} \quad (2.9)$$

where ϕ = Lagrange polynomial basis function, D = Lagrange polynomial denominator.

Observe that equation (2.7) can be simplified into the following form:

$$\phi_j(\tau_i) = \begin{cases} 1, & \text{when } l = j \\ 0, & \text{when } l \neq j \end{cases} \quad (2.10)$$

$$(2.11)$$

It follows that the polynomial coefficients of $\mathbf{x}^{(q)}(\tau_i)$ and $\mathbf{z}^{(q)}(\tau_i)$ correspond simply to the values of the states at the collocation points, thus equations (2.5–2.7) can be dropped from the problem.

Residual equations

To obtain the residual equations, we replace the continuous differential and algebraic variables in (2.1–2.2) with their discretized forms using the following discretization mapping (expressions 2.12–2.15).

$$\frac{d\mathbf{x}}{dt} \rightarrow \left[\frac{d\mathbf{x}^{(q)}(\tau_i)}{d\tau} \right] \frac{d\tau}{dt} = \left[\sum_{j=0}^{\text{nCOL}} \mathbf{x}_j^{(q)} \frac{d\phi_j(\tau_i)}{d\tau} \right] \frac{1}{\delta} \quad (2.12)$$

$$\mathbf{x}(t) \rightarrow \mathbf{x}^{(q)}(\tau_i) = \mathbf{x}_i^{(q)} \quad (2.13)$$

$$\mathbf{z}(t) \rightarrow \mathbf{z}^{(q)}(\tau_i) = \mathbf{z}_i^{(q)} \quad (2.14)$$

$$\mathbf{u}(t) \rightarrow \mathbf{U}_i^{(q)} \quad (2.15)$$

The following residual equations are obtained:

$$\frac{1}{\delta} \sum_{j=0}^{\text{nCOL}} \mathbf{x}_j^{(q)} \frac{d\phi_j(\tau_i)}{d\tau} - \mathbf{f}[\mathbf{x}_i^{(q)}, \mathbf{z}_i^{(q)}, \mathbf{U}_i^{(q)}] = \mathbf{0} \quad \text{with } q = 1, \dots, \text{nFE}, i = 1, \dots, \text{nCOL} \quad (2.16)$$

$$\mathbf{h}[\mathbf{x}_i^{(q)}, \mathbf{z}_i^{(q)}, \mathbf{U}_i^{(q)}] = \mathbf{0} \quad \text{with } q = 1, \dots, \text{nFE}, i = 1, \dots, \text{nCOL} \quad (2.17)$$

$$\mathbf{x}_0^{(1)} = \mathbf{x}_0 \quad (2.18)$$

Control Input Vector

The control input vector is represented by the following equation:

$$\mathbf{U}_i^{(q)} = \mathbf{u}_p \quad \text{where } p = \text{sampling instance} = \{0, 1, \dots, \text{nSamples} - 1\} \quad (2.19)$$

$$p = \left\lfloor \frac{q-1}{\eta} \right\rfloor \quad \text{where } \eta = \text{finite elements per sample}, \lfloor \cdot \rfloor = \text{floor function} \quad (2.20)$$

The temporal ordinate of the control inputs is measured in terms of *sample units* (p). In a standard OCFE implementation, the length of one sample time corresponds exactly to the length of one finite element. In this implementation, multiple finite elements per sample may be specified (through the η parameter) for the purpose of improving accuracy [7].

Continuity equation

Continuity between finite elements is enforced in the differential variables by using

the polynomial approximation derived for the previous finite element to calculate the value of the differential variable at the left-hand side boundary of the current finite element (at the continuity node, refer to Figure 2.1) .

$$\mathbf{x}_0^{(q)} = \sum_{j=0}^{\text{nCOL}} \mathbf{x}_j^{(q-1)} \phi_j(1) \quad \text{for } q = 2, \dots, \text{nFE} \quad (2.21)$$

Jacobi Collocation Points

Collocation points are *optimal* knot placement points, which can be shown to correspond to the roots of an orthogonal polynomial family [20]. The Jacobi family of orthogonal polynomials is commonly used. The following recursion formula, called Rodrigues' formula, can be used to generate Jacobi polynomials:

$$P_n^{(\alpha, \beta)}(x) = \sum_{i=0}^n (-1)^{n-i} \gamma_i x^i \quad (2.22)$$

$$\gamma_i = \frac{(n-i+1)(n+i+\alpha+\beta)}{i(i+\beta)} \gamma_{i-1} \quad \text{with } \gamma_0 = 1 \quad (2.23)$$

Efficient FORTRAN routines for generating Jacobi roots can be found in Villadsen and Michelsen [79].

2.5 Feedback using Predictive Control Algorithms

In order for a control strategy to be applied successfully, the controller needs to possess an accurate and up-to-date view of the plant. Uncertainties in the process can distort the picture however. These uncertainties usually enter the plant in the following forms [53]:

1. Plant-model mismatch

- (a) Structural mismatches - arising from incomplete or erroneous models
- (b) Parametric mismatches - arising from incorrectly identified parameters

2. Disturbances

- (a) Fast - high frequency variations in the feed, conditions etc.
- (b) Slow - process drifts

3. Unknown initial conditions

4. Measurement Noise

There are a wide variety of methods and means to accommodate and counter these uncertainties, but for the purpose of this work we will concentrate on one particular methodology, that is, model-based feedback control using a nonlinear predictive control algorithm.

2.5.1 Model Predictive Control

Model Predictive Control (MPC) is an umbrella term for a family of algorithms that exploit a model to predict the future response of a plant given a set of manipulated variable adjustments [64]. The control scheme incorporates an optimization routine which computes a sequence of control actions required to drive the plant to some operating point in an optimal fashion. The first of this sequence of computed control actions is implemented in the plant and process measurements are taken. The feedback is introduced through a bias update on the disturbance estimate. This procedure is repeated at each control interval.

MPC has many favorable properties that have contributed to it becoming one of the most widely-used advanced control schemes today [64]. It is an inherently multivariable controller, therefore interaction effects are handled naturally through the model. In most variants of MPC, operating constraints are handled easily through hard constraints in the optimization problem or soft constraints via penalty terms in the objective function.

One of the earliest algorithms in the MPC family is Dynamic Matrix Control (DMC), presented by Cutler and Ramaker [21] in 1979, an algorithm which had already been successfully implemented in a real plant several years prior. DMC incorporated step-response models and was capable of handling linear dynamics. Uncertainties are handled through feedback. However, because the optimization problem in DMC was posed as an unconstrained one, constraint-handling was somewhat ad hoc, which led Garcia to extend DMC by reformulating the optimization problem as a quadratic program (QP) in which linear constraints are admissible [34]. This gave rise to an algorithm known as quadratic dynamic matrix control (QDMC).

Despite the fact that many chemical processes exhibit inherently nonlinear behavior, linear MPC continues to be pervasive, especially in continuous processes where the predominant objective is to maintain the plant at certain operating points (known as a *regulation problem*) [64]. In batch and transient processes however, the ability to move rapidly from one operating point to another (*servo problem*) is more important and hence nonlinear models are better suited for these applications. An early application of MPC to a nonlinear (batch) process can be found in Garcia [33], where QDMC was applied to a nonlinear time-varying process modeled using differential equations. The nonlinear model was integrated to calculate the future projection and a local linearization was simultaneously performed to obtain the step coefficients for the dynamic matrix, required for optimization.

The local linearization approach however is only suited to mildly nonlinear processes. Patwardhan et. al [60] presented a formulation of nonlinear MPC where control actions are computed by optimizing a full nonlinear model at every control interval. The model is discretized using orthogonal collocation and solved as a constrained nonlinear program.

2.5.2 Control of Transient Processes using MPC

Much of the work in the area of applications of MPC to transient processes have been on batch processes. Krothapally and Palanki [49] showed an online application of MPC to a batch polymerization process. Nagy and Braatz [56] developed a shrinking-horizon MPC algorithm which uses an economic objective function, and showed an example of an application of their algorithm to a batch crystallization simulation.

Hillestad and Anderson [39] applied nonlinear predictive control to the problem of grade transitions in polymerization reactors. The work of Wang et. al. [82] involved the use of nonlinear model predictive control to manage grade transitions on a polyethylene reactor. Feather et. al. [29] described a hybrid predictive control approach for the polymer grade transition problem. The nonlinearity in the process was modeled using approximate linear models, and an algorithm based on integer variables was used to switch between them, based on what the operating conditions were. A case study based on a propylene loop reactor simulation was presented.

2.5.3 Incorporating Economics into the Control Algorithm

In many large chemical plants today, economic optimization is typically performed by a Real-Time Optimization (RTO) module that computes the optimal operating point of a plant and sends the relevant setpoints to a lower-level MPC controller, whose duty is to track them. RTO systems are typically based on steady-state models. These work sufficiently well for steady-state operations, but there is some indication that they are inadequate for nonlinear dynamic processes such as grade transitions and batch systems [45]. In view of that, there has been a move towards exploring the use of dynamic models in RTO systems and making them computationally tractable.

Several strategies exist for performing economic optimization using dynamic models. Tosukhowong et. al. [74] suggested that a dynamic RTO model be used, but with the provision that the dynamic optimization be performed at a frequency that is significantly lower than that of the predictive controller in order to keep the computational requirements low enough for real-time applications. Reduced-order models for a chosen optimization frequency were also discussed.

In a similar vein, Kadam et. al. [45] proposed the hierarchical decomposition of the optimization problem into a higher-level economic optimization problem (with dynamic models) and lower level control problem. Instead of performing the economic optimization at a fixed frequency, Kadam suggested that reoptimization be performed contingent upon the detection of disturbances whose magnitudes exceed a certain pre-determined threshold. This disturbance detection is done through a disturbance sensitivity analysis of the optimal reference trajectories.

On smaller plants, there has been some interest in integrating the economic optimization problem directly with the predictive controller itself. Zanin et al. [86] proposed a 1-layer approach where the economic optimization problem is solved together with the MPC optimization problem. This is accomplished by adding an economic function as a penalty term in the objective function of the MPC controller. Their motivation for taking a 1-layer approach as opposed to a traditional 2-layer approach (where economics and dynamics are separated) is to address what they consider a major deficiency associated with traditional layered optimization strategies, that is, because the controller and the optimizer are not dealing with exactly the same pieces of information, conflicts may arise and the predicted optimal operational point may turn out to be suboptimal. Zanin reported success in applying this algorithm to an in-

dustrial process. The 1-layer approach suffers from several drawbacks however: 1) difficulty in apportioning of weights (in the optimization problem) for achieving regulation and economic objectives; 2) integral action is generally not achieved if inputs are penalized.

Another study of the single-level strategy can be found in Becerra et. al. [10]. The authors carried out simulations of Tennessee-Eastman process where an economics-based MPC controller was applied to the process.

Chapter 3

Development of the Model

The ultimate purpose of computing is insight, not numbers.

- Richard Hamming, applied mathematician

In this chapter, a detailed description of an in-house developed modeling language for representing dynamic optimization problems is presented. This is followed by the development of the Kraft paper mill model. This chapter concludes with an brief account of the various formulation and modeling issues encountered.

3.1 A Modeling Language for Dynamic Optimization (MLDO)

One of the primary contributions of this work to the corpus of modeling and optimization techniques at large is a domain-specific language for describing dynamic optimization problems, MLDO (Modeling Language for Dynamic Optimization). Model code written in MLDO is used to generate dynamic optimization code in a commercial mathematical programming language called AMPL [31]. This code is subsequently passed to an NLP solver for optimization.

The advantages of using the AMPL language in an intermediate step are many. AMPL provides facilities such as efficient automatic differentiation (exact first and second derivatives), bounds checking (via its presolve phase), interoperability with different solvers, problem reduction, and other features that aid in the efficient solution of nonlinear optimization problems.

MLDO is designed for dynamic optimization problems of the following form:

$$\min_{\mathbf{u}(t)} \Phi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) \quad (3.1)$$

subject to

$$\mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \quad (3.2)$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \quad (3.3)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) \leq 0 \quad (3.4)$$

$$\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U \quad (3.5)$$

$$\mathbf{z}_L \leq \mathbf{z}(t) \leq \mathbf{z}_U \quad (3.6)$$

$$\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \quad (3.7)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (3.8)$$

MLDO takes a representation of the above problem and generates the requisite code for optimization. Only first-order differential terms are supported in the problem formulation, but it is trivial to reduce the order of higher-order differential terms by introducing dummy variables.

Because it uses code to generate code, MLDO functions well as a rapid prototyping tool that allows for experimentation with various configurations and ideas. In contrast to software like DynoPC [85] and gPROMS [9] which are aimed at the practitioner, MLDO exposes lower-level code and gives the modeler the flexibility to extend dynamic optimization models in novel ways while saving them from writing repetitious “boilerplate” code.

To solve the NLP problem, we opted to use IPOPT 3.1.0, a primal-dual interior point solver originally developed at Carnegie-Mellon University. IPOPT [81] is a large-scale sparse nonlinear optimizer for continuous problems that implements a primal-dual interior point method. It performs line-searches based on “filter” methods originally proposed by Fletcher and Leyffer [80]. These methods have garnered attention in the research community for their simplicity and have managed to generate a flurry of research [36] in a short time. Filter techniques aim to promote global convergence without the need for a penalty function. The concept of a filter is based on the optimizer accepting a step if it reduces either the objective function or a constraint violation function. IPOPT also uses efficient sparse matrix routines for factorizing

large-scale sparse linear systems, and is capable of solving fairly large scale nonlinear problems. IPOPT is released under the Commons Public License, an open-source type license.

In the course of our work, several other solvers were tested, including MINOS, SNOPT, KNITRO and LOQO. The NEOS Server [22] was instrumental in this effort.

The solution data were channeled into MATLAB for visualization via a custom-developed AMPL-MATLAB interface.

The primary machine used for computation was an Intel Pentium 4 (3 GHz) computer with 1 GB RAM, running the Debian Linux (kernel version 2.4.27-2) operating system.

3.1.1 Features of MLDO

Discretization using Orthogonal Collocation on Finite Elements (Simultaneous Method)

In order to solve a dynamic optimization problem within an optimization-type framework, it is necessary to discretize the differential-algebraic equation (DAE) system. MLDO generates code for solution using a simultaneous method. Discretization is performed using Orthogonal Collocation on Finite Elements (OCFE) [20].

One of the features of the OCFE implementation in MLDO is that the number of Finite Elements (FE) that correspond to one sample period is unrestricted [7]. The major advantage of this feature is that it allows one to capture rapidly changing dynamic behavior without affecting the sample period.

Concatenation of Multiple MLDO Models

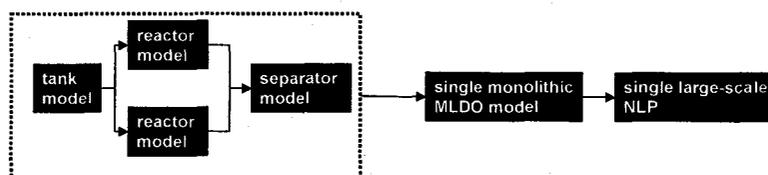


Figure 3.1: Model concatenation with MLDO.

A logical way of developing a plantwide process model is build the models of individual process units separately (with independent inputs and outputs) before combining them into one monolithic process model. This leads to a *decoupled design*. MLDO is capable not only of merging multiple process unit models into one model, it is also able to generate “arrays” of unit models. This feature saves the modeler the effort of writing duplicate model code for multiple instances of the same process unit. For example, in Figure 3.1, a model of a reactor can be treated as a template, and the modeler is allowed to derive multiple instances of it. At the final stage, models are concatenated to form a monolithic model from which a large-scale NLP problem is generated.

Models are linked to one another using “connection equations”. Connection equations relate the outputs of one model to the inputs of another. For instance, the modeler can specify a connection between a tank and reactor by equating the material stream exiting the tank to the inlet of the reactor:

$$F_{out}^{tank} = F_{in}^{reactor}$$

In collaborative optimization, where models are developed by different design teams, one may also choose to first perform a feasibility optimization to reconcile the models [3]. In simplistic terms, this entails performing a relaxation by specifying a penalty for the discrepancies arising and subsequently minimizing the penalty in the optimization problem. For example, the above connection may be written as:

$$F_{out}^{tank} = F_{in}^{reactor} + \epsilon_1$$

where ϵ_1 = discrepancy penalty to minimize.

NLP Initialization

One of the prerequisites for successfully solving large-scale NLPs (especially those arising from DAE systems) is starting a suitable initialization point. In the case of process models, such a point can be obtained through a process simulation. MLDO has the facility to translate model code written in MLDO syntax into the gPROMS [9] modeling language. This allows the modeler to run a parallel model in the gPROMS environment and exploit its DAE integration and nonlinear equation solver capabilities to obtain a nominal (unoptimized) profile. The results of the gPROMS simulation can then be used as an initial point for the dynamic optimization problem in AMPL.

MLDO provides the facility to read the output of a gPROMS simulation (in the form of a gPLOT file) and automatically generate code to initialize the dynamic optimization problem.

Automatic Generation of Documentation

Because MLDO model code closely resembles mathematical notation, it can easily be converted to a mathematically-typeset document via the route of a typesetting language, \LaTeX . This feature (also known as “pretty-printing” in the computing world) essentially makes MLDO code self-documenting, in that human-readable documentation can be instantly generated from model code. This also facilitates the exchange of model descriptions with parties who have no knowledge of MLDO syntax.

Miscellaneous features

The following methods are implemented in MLDO and they warrant a brief mention.

1. **Multi-period method for Optimization Under Certainty.** In this method [67], uncertain model parameters are treated as discrete variables. A set of parallel models, each corresponding a certain value of each parameter, are solved simultaneously in the same NLP problem to find a set of control inputs that are feasible for all the cases. These parallel model descriptions typically result in a high-dimensional optimization problem. The ability to automatically generate these parallel model descriptions is built into MLDO.
2. **Homotopy Methods.** Balthazaar [8] encountered computational difficulties when introducing drastic disturbances such as shutdowns. Homotopy was proposed as a numerical method for gently initializing trajectory profiles. The basic idea behind homotopy (first articulated by Henri Poincaré in 1900) is the continuous transformation of one function to another. In an optimization context, this idea is used to initialize NLPs by warm-starting the problem from a numerically benign point (such as a steady-state solution) and then slowly moving that point closer and closer to the actual solution (i.e. one that contains drastic changes in the trajectory, in our case) through the use of a forcing function. The solution of the previous solve stage is used as the initialization for the current solve stage.

3. **Two-tiered Solution Method.** Balthazaar [8] discovered that it was possible to obtain different trajectories that returned the same objective function value upon solving the nonlinear problem, suggesting that there are multiple solutions to the same problem or that the solution was non-unique. This is a common occurrence in problems where there are a large number of degrees of freedom. They may also occur when:

- 2 or more variables in the objective have equal weights
- 2 or more variables have no weights in the objective function
- 2 or more variables have unequal weights in the objective function, but the allowable trade-off between them does not cause any change in the objective value.

To counter this type of ill-conditioning, Balthazaar proposed a two-tiered approach where the optimization is performed twice. In the first tier, the optimization is performed using an economics-based objective function. In the second tier, the economics-based objective function is converted into a constraint, where it is equated to the objective value obtained in the first tier optimization (thus locking down the economics of the system), and a new objective function that minimizes control moves is solved. The trajectories resulting from this second tier optimization is deemed to have maximized the economics and minimized the control actions required to achieve that level of economics, and is regarded as a unique solution.

4. **Visualization with Sparklines.** The smallest meaningful result unit in a dynamic optimization solution is typically a trajectory. In an ordinary-sized problem the number of trajectories in the solution generated may range in the hundreds. Therefore it is necessary to devise some means of visualizing a large number of trajectories on screen and on the printed page. One such means is the use of a compact word-sized graphic format (proposed by the statistician Edward Tufte [75]) for displaying trend information, known as “Sparklines”. MLDO has the ability to generate pages of Sparkline figures (Figure 3.2) for the trajectories obtained from the solution of a dynamic optimization problem.

3.1.2 Design of MLDO

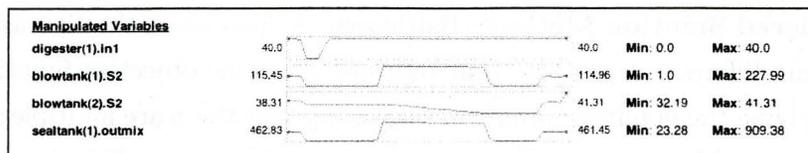


Figure 3.2: Use of Sparklines for displaying variable trajectories.

MLDO was developed using Python [55] (version 2.4.1), an open-source object-oriented programming language originally developed at the *Centrum voor Wiskunde en Informatica* (CWI) in the Netherlands. A number of software tools in process engineering have been written in Python, including a model integration tool (originating from RWTH-Aachen [69]) and a high-fidelity multidisciplinary design optimization (MDO) package [5].

MLDO is a domain-specific language built on the software engineering concept of “metaprogramming”, a technique by which code in one programming language is used to generate code in another. There has been a great deal of interest in recent years in the use of metaprogramming techniques to improve the quality of software engineering. This trend has not escaped the attention of the process control community, as witnessed by the existence of control-applications-specific languages such as ControlH [28] (a control specifications language developed at Honeywell that is capable of emitting C or Ada code).

Metaprogramming is particularly suited to situations where large amounts of complicated and highly customized computer code is required to solve a problem, and where such code is susceptible to human errors if written manually. Metaprogramming proposes to transfer the task of writing the code to the computer. The user is required to only specify the particular problem in a specially defined language (called a *domain-specific language*) that is processable by a program known as a *code generator*. The code generator analyzes the structure of the specified problem and generates the requisite lower-level programming code for solving the problem. In contrast to conventional programming languages, the input to the code generator is usually *declarative* rather than *imperative*; that is, it contains a *problem definition* rather than the explicit computer instructions required to solve the problem.

There have been a number of precedents in chemical engineering where metaprogramming techniques were used for rapidly producing computer codes that are main-

tainable and reliable. Pantelides [59], in his description of the SpeedupTM process simulation tool, mentions an example in which the Pascal programming language was leveraged to write routines which generated lower-level FORTRAN code for calculating derivatives.

In a similar vein, the chief purpose of MLDO is to allow the modeler to write dynamic optimization models/problems a compact canonical form that can be subsequently be transformed into another language (in our case, AMPL) for solution. This transformation process involves operations such as generation of code for performing orthogonal collocation, running two-tiered methods, imposing failure constraints and so on.

MLDO was designed to have a syntax that closely resembles the mathematical notation of dynamic optimization. Its syntax is illustrated in the following example. Consider the following mathematical description of a simple dynamic optimization problem:

$$\begin{aligned} \min_{U(t)} & -[1 - z_a(t_f) - z_b(t_f)] \\ \text{s.t.} \quad \frac{dz_a(t)}{dt} & = U(t) \cdot [10 \cdot z_b(t) - z_a(t)] \\ \frac{dz_b(t)}{dt} & = U(t) \cdot [z_a(t) - 10 \cdot z_b(t)] - [1 - U(t)] \cdot z_b(t) \\ z_a(0) & = 1 \\ z_b(0) & = 0 \\ 0 & \leq U(t) \leq 1 \end{aligned}$$

where $U(t)$ = control input variable, z_a, z_b = differential variables, t_f = terminal time.

The transcription of the above problem in MLDO syntax is as follows:

```

minimize:
  -(1 - za(tf_) - zb(tf_));
statevars:
  za; zb;
controlvars:
  U;
dae:
  $za = U*(10*zb - za);
  $zb = U*(za - 10*zb) - (1 - U)*zb;
init:
  za(0) = 1;
  zb(0) = 0;
constraints:

```

```

0 <= U <= 1;
ocfe:
    tf_ = 0.4; nFE_ = 16; nCOL_ = 2;

```

The syntax is largely self-explanatory. The objective function resides in the `minimize` section. Differential variables are declared in the `statevars` section, while control input variables are declared in the `controlvars` section. The `dae` section contains the differential algebraic equation (DAE) system, and `init` section, their initial values. The `$` sign denotes the differential operator. Operating constraints are written in the `constraints` section. The `ocfe` section allows the modeler to specify parameters for orthogonal collocation on finite elements, such as the terminal time (`tf_`), number of finite elements to use (`nFE_`) and the number of collocation points (`nCOL_`) in each finite element. The complete syntax is documented in an internal technical report [18].

AMPL Code

The AMPL code that is generated from the above problem description (with the table of Jacobi roots omitted) appears below:

```

# ===== OCFE Parameters =====
param tf_;
param nCOL_;
param nFEperSample_;
param sampleSize_;

param nSamples_ := tf_/sampleSize_; # number of samples
param nFE_ := nFEperSample_ * nSamples_; # number of finite elements
param delta_ := tf_/nFE_; # size of finite element
param JacobiRoots_{COL_ in 0..9, NCOL_ in 1..8}; # table of roots of the Jacobi polynomial
param tau_{COL_ in 0..nCOL_} := JacobiRoots_[COL_,nCOL_]; # collocation points
param FELowerBounds_{FE_ in 1..nFE_} := (FE_ - 1) * delta_; # lower bounds of finite elements
param t_{FE_ in 1..nFE_, COL_ in 0..nCOL_} := FELowerBounds_[FE_] + JacobiRoots_[COL_,nCOL_]*delta_;
param fe2samp_{FE_ in 1..nFE_} = floor((FE_ - 1)/nFEperSample_);

# ==== Lagrange Interpolation Polynomials ====
param LagrangeDenom_{j_ in 0..nCOL_} := # Lagrange polynomial denominator
    prod{l_ in 0..nCOL_: l_ != j_} (tau_[j_] - tau_[l_]);

param LagrangeDiff_{j_ in 0..nCOL_, COL_ in 0..nCOL_} := # Lagrange polynomial derivative
    1/LagrangeDenom_[j_] *
    sum{k_ in 0..nCOL_: k_ != j_} (prod{l_ in 0..nCOL_: l_ != j_ and l_ != k_} (tau_[COL_] - tau_[l_]));

param LagrangeContinuity_{j_ in 0..nCOL_} := # Expression for continuity equations
    1/LagrangeDenom_[j_] * prod{l_ in 0..nCOL_: l_ != j_} (1 - tau_[l_]);

```

```

# ===== Trial Function / Variable Declarations =====
var za{FE_ in 1..nFE_, COL_ in 0..nCOL_};
var Derivative_za_{FE_ in 1..nFE_, COL_ in 0..nCOL_};
var Continuity_za_{FE_ in 1..nFE_-1};

var zb{FE_ in 1..nFE_, COL_ in 0..nCOL_};
var Derivative_zb_{FE_ in 1..nFE_, COL_ in 0..nCOL_};
var Continuity_zb_{FE_ in 1..nFE_-1};

var U{Sample_ in 0..nSamples_-1};

# ==== Objective Function ====
minimize obj: -(1 - za[nFE_,nCOL_] - zb[nFE_,nCOL_]);

subject to
# ==== Trial Function Definitions ====
constraint1 {FE_ in 1..nFE_, COL_ in 0..nCOL_}:
    Derivative_za_[FE_,COL_] = sum{j_ in 0..nCOL_}(za[FE_,j_] * LagrangeDiff_[j_,COL_]);
constraint2 {FE_ in 1..nFE_, COL_ in 0..nCOL_}:
    Derivative_zb_[FE_,COL_] = sum{j_ in 0..nCOL_}(zb[FE_,j_] * LagrangeDiff_[j_,COL_]);

# ===== Differential Algebraic System - Residuals =====
constraint3 {FE_ in 1..nFE_, COL_ in 1..nCOL_}:
    Derivative_za_[FE_,COL_]/delta_ = U[fe2samp_[FE_]]*(10*zb[FE_,COL_] - za[FE_,COL_]);
constraint4 {FE_ in 1..nFE_, COL_ in 1..nCOL_}:
    Derivative_zb_[FE_,COL_]/delta_ = U[fe2samp_[FE_]]*(za[FE_,COL_] - 10*zb[FE_,COL_]
        - (1 - U[fe2samp_[FE_]])*zb[FE_,COL_]);

# ===== Initial Conditions =====
constraint5: za[1,0] = 1;
constraint6: zb[1,0] = 0;

# ===== Operating Constraints =====
constraint7 {FE_ in 1..nFE_, COL_ in 0..nCOL_}: 0 <= U[fe2samp_[FE_]] <= 1;

# ===== Continuity Constraints =====
constraint8 {FE_ in 1..nFE_-1}:
    Continuity_za_[FE_] = sum{j_ in 0..nCOL_}(za[FE_,j_] * LagrangeContinuity_[j_]);
constraint9 {FE_ in 1..nFE_-1}: Continuity_za_[FE_] = za[FE_+1,0];
constraint10 {FE_ in 1..nFE_-1}:
    Continuity_zb_[FE_] = sum{j_ in 0..nCOL_}(zb[FE_,j_] * LagrangeContinuity_[j_]);
constraint11 {FE_ in 1..nFE_-1}: Continuity_zb_[FE_] = zb[FE_+1,0];

data;
# ===== OCFE parameters =====
param tf_ := 0.4;
param nCOL_ := 4;
param nFEperSample_ := 1;
param sampleSize_ := 0.05;

```

⋮

[Table containing Jacobi Roots]

⋮

```
# ===== Solver Parameters =====
solve;
```

gPROMS Code

To demonstrate the ability of MLDO to generate gPROMS model code, we present the gPROMS MODEL and PROCESS code generated from the above problem description. (The objective function and constraints are specified in the graphical environment of gPROMS and are omitted in the generated code.)

```
# gPROMS MODEL: mdl
VARIABLE
    za AS notype
    zb AS notype
    U AS notype
EQUATION
    $za = U*(10*zb - za);
    $zb = U*(za - 10*zb) - (1 - U)*zb;

# gPROMS PROCESS: proc
UNIT
    mdl as mdl
INITIAL
    mdl.za = 1;
    mdl.zb = 0;
SOLUTIONPARAMETERS
    gPLOT := ON;
SCHEDULE
    CONTINUE FOR 0.4;
```

L^AT_EX Code

Finally, to demonstrate MLDO's self-documentation capability, we present the L^AT_EX typesetting code generated from the original problem description. This code can then be compiled into a mathematically-typeset document (PDF file).

```

% ===== Standard preamble =====
\documentclass[10pt]{article}
\usepackage{geometry}
\geometry{letterpaper}
\usepackage{amssymb,amsmath,amsfonts}
% ===== Document =====
\begin{document}
\section{Model - mdl}
\textbf{Differential Variables} \begin{quote}$z_a$, $z_b$\end{quote}
\textbf{Control Input Variables} \begin{quote}$U$\end{quote}
\textbf{Differential and Algebraic Equation System}\\
\begin{eqnarray}
\frac{d}{dt} z_a &= U \cdot (10 \cdot z_b - z_a) \\
\frac{d}{dt} z_b &= U \cdot (z_a - 10 \cdot z_b) - (1 - U) \cdot z_b
\end{eqnarray}
\textbf{Initial values}\\
\begin{eqnarray}
z_a(0) &= 1 \\
z_b(0) &= 0
\end{eqnarray}
\textbf{Constraints}\\
\begin{eqnarray}
0 &\leq U \leq 1
\end{eqnarray}
\end{document}

```

The resulting document is shown in Figure 3.3.

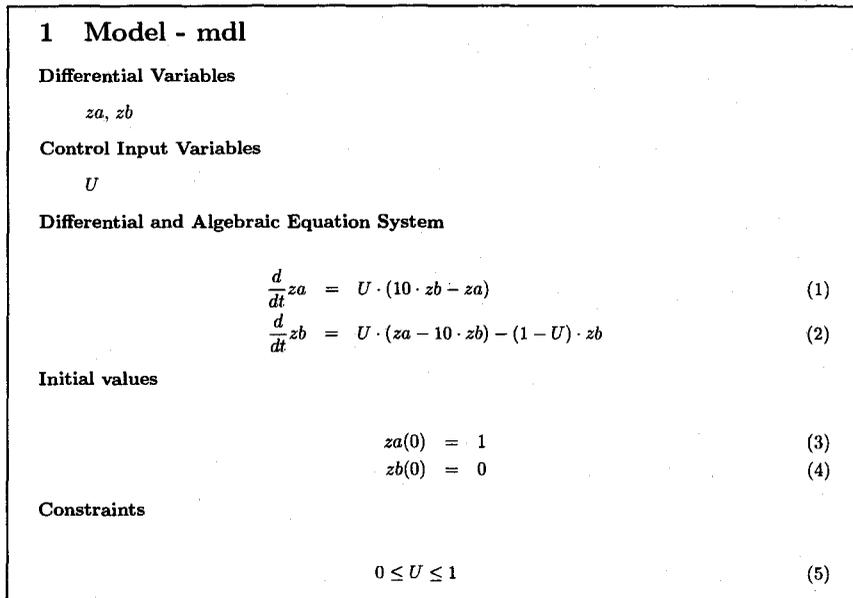


Figure 3.3: MLDO-generated documentation (using L^AT_EX).

3.1.3 The Code Generation Process

In order to create this domain-specific language, two techniques from computer science were employed:

1. Context-free Grammar

In human languages, a grammar defines the syntax of a language. Similarly, for a computer, a *context-free grammar* [2] is the formal way of defining a programming language. A context-free grammar is a set of rules applied to a language to break it down into smaller parts called tokens, through a process called *tokenization*. These tokens can then be manipulated using text transformation rules.

Context-free grammars are usually specified using a strict notation known as Backus-Naur form (BNF) [83]. The grammar of MLDO was expressed in an alternative form of the BNF. For instance, the following string:

```
unitDefn = Word(alphanums+"$", alphanums+"_")
```

defines the syntax of a unit token. The above declaration can be read as follows:

- (a) the token must take the form of a "word". A word is defined as a sequence of characters;
- (b) the word may start with any alphanumeric character or the dollar sign;
- (c) the sequence of characters past the 1st character must be either an alphanumeric character or an underscore.

Consider the following equation in MLDO:

$$\$za = U*(10*zb - za);$$

Tokenization of this equation (with the above definition of a unit token, plus a few other declarations) will produce the following array of tokens:

```
['$za', 'U', '10', 'zb', 'za']
```

In this example, the tokens represent the basic elements in an equation. To illustrate how these tokens are used to generate code, we present this illustration from OCFE. In OCFE, collocation is usually only applied to differential and algebraic variables, whereas control action variables are represented by

piecewise constant functions. In order to uniquely define the position of a differential/algebraic variable, 2 pieces of information are required (current Finite Element [FE], and current Collocation Point [COL] within that element). On the other hand, control input variables only require 1 piece of information to pinpoint (current Sample Time). In the AMPL code, these two categories of variables require different handling.

In MLDO, the user is required to explicitly declare each variable as either a *differential*, *algebraic*, or *control input* variable. This is known as “type-declaration”. When an equation is broken up into tokens, it is easy to map each token back to their type by doing a simple look-up on the type-declaration table. In the above case, “\$za”, “za” and “zb” are differential variables, whereas “U” is a control input variable. A routine is used to map the tokens to their types, and with the types correctly assigned, appropriate transformations can be performed (either with a simple string replacement, or with regular expressions, described below).

`$za → Derivative_za_[FE_,COL_]`

`za → za[FE_,COL_]`

`zb → zb[FE_,COL_]`

`U → U[fe2samp_[FE_]`

Note: `fe2samp_` is an internal function that returns the sample time that corresponds to the current finite element.

Mapping tokens of an equation to their correct types ensures that the different elements in an equation receive the appropriate treatment. Once the transformation is performed, the tokens are reassembled into a complete AMPL constraint equation.

2. Regular expressions

Regular expressions [40] originated from two branches of theoretical computer science, automata theory and formal language theory. Regular expressions are used to define a pattern for matching specific patterns in a set of strings. From a practical standpoint, they are used to transform strings, e.g. in our case, from MLDO notation to the keywords in the target language.

For example, the following regular expression:

$$\backslash\$([A-Za-z0-9_]+\backslashb\backslash\frac{d}{dt}\backslash1/$$

is used in the L^AT_EX generation procedure to transform all MLDO derivative

variables to its mathematical counterpart, expressed in \LaTeX code. The above regular expression converts an MLDO token, say $\$za$, to a \LaTeX string in a precise and unambiguous manner:

$$\$za \rightarrow \text{\frac{d}{dt}} za$$

The regular expression can be read as follows:

- (a) `/` is a delimiter that marks the boundaries of regular expression patterns. Regular expressions are typically specified in the form `/p/r/`, where `p` denotes the pattern to search for, and `r` denotes the replacement string;
- (b) `\$` matches a literal dollar sign, `$`;
- (c) `()` captures the expression enclosed in parentheses and assigns the results to a variable called a “backtracking” variable that can be referred to later;
- (d) `[A-Za-z0-9_]` matches the class of characters contained within the brackets. In this case, matches the following character classes: letters A to Z, a to z, numbers 0 to 9 and the underscore character;
- (e) `+` matches the preceding expression 1 or more times;
- (f) `\b` matches the start or end of a word;
- (g) `\1` is a backtracking variable, returns the value of the expression matched within parentheses;

If a regular expression is defined rigorously enough, very precise transformations can be performed. This technique is extremely powerful and can be used to do symbolic manipulation, syntax rearrangement and so on.

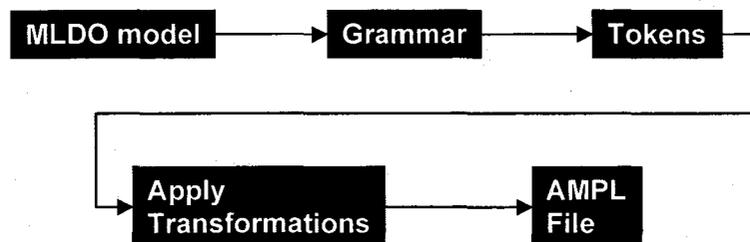


Figure 3.4: Simplified diagram of the code generation process.

The code generation essentially proceeds as follows (Figure 3.4): Given an MLDO model, the parser (that is, the part of the program that processes the model) reads the

code and passes it through the context-free grammar, a process which produces tokens. These tokens are then transformed using text-transformation techniques which include regular expression transformations. After the transformations are applied, an AMPL (or gPROMS or L^AT_EX) file is produced.

3.2 Modeling the Kraft Mill Fiber Line

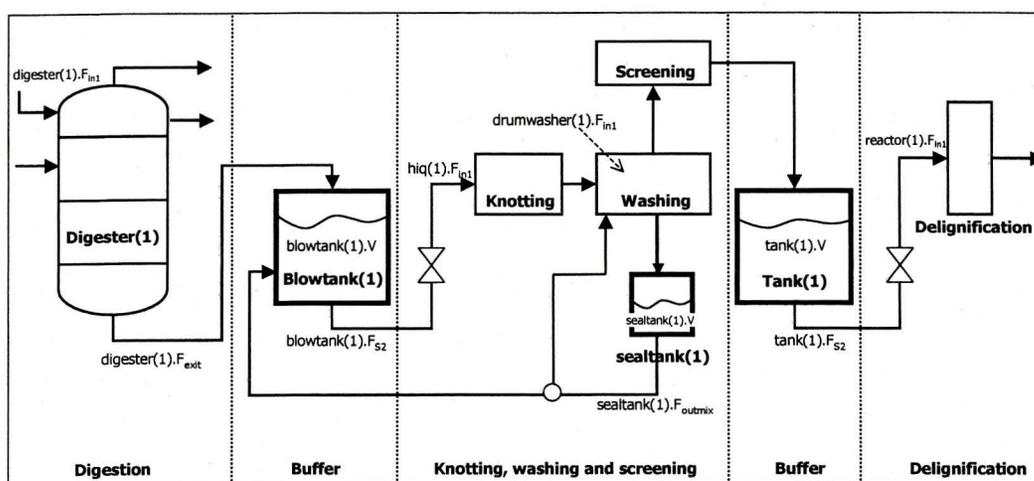


Figure 3.5: Simplified schematic of plant model.

The purpose of this section is to acquaint the reader with the Kraft Mill model used in the case studies. This model is based on prior work done by Balthazaar [8] and Dubé [24], with extensions based on various literature sources. The modeling effort was focused on a suitably chosen subset of the Kraft Paper Mill (i.e. the fiber line) which provided a reasonable representation of the process dynamics under shutdown conditions. The majority of the units are modeled with steady-state equations, with the major dynamics represented in the tanks.

The diagram in Figure 3.5 shows a simplified schematic of the Kraft Mill fiber line topology. (For a full schematic, the user is referred to Appendix A) The main departments in this section of the plant are the digestion, knotting & washing, and delignification departments. These departments are separated by buffer tanks, whose levels are manipulated in order to keep the plant operational when any unit in any given department is shut down.

In the following sections, we will give a brief explanation of the model equations that describe the major units in these departments. The complete set of equations and parameters are available in Appendix A.

In this model, we assume that the initial conditions of the system are only related to the dynamic equations (due to the pseudo-steady state assumption). In practice, each individual unit has a set of initial conditions from which it operates from.

3.2.1 Conventions and Nomenclature

The main variable names used in this model that are related to flow are constructed as follows:

$$F_s^i, \quad x_s^i$$

where

F = flowrate (t/h)

x = mass fraction

s = stream name

i = component in stream = { P : Pulp, DS : Dissolved solids, W : water}

Only three types of component streams are considered in the model: **pulp** (P), **dissolved solids** (DS) and **water** (W). The pulp component consists of cellulose, bound lignin, knots and shives. The dissolved solids component contains organic and inorganic subcomponents.

In pulp and paper parlance, the term *consistency* refers to the mass fraction of pulp fibers (dry) in a process stream. *White liquor* refers to a concentrated solution of chemicals, usually Na_2S and NaOH , used for cooking. *Black liquor* refers to the spent white liquor and dissolved organic material that remain after the cooking process.

3.2.2 Kraft Digester

The primary function of a digester is to convert solid wood chips (obtained by debarking and chipping softwood logs) into a pulp stream. It accomplishes this by cooking the chips and reacting them with white liquor. Cooking involves heating the mix and maintaining it at an elevated temperature. In the present model, the mix is heated

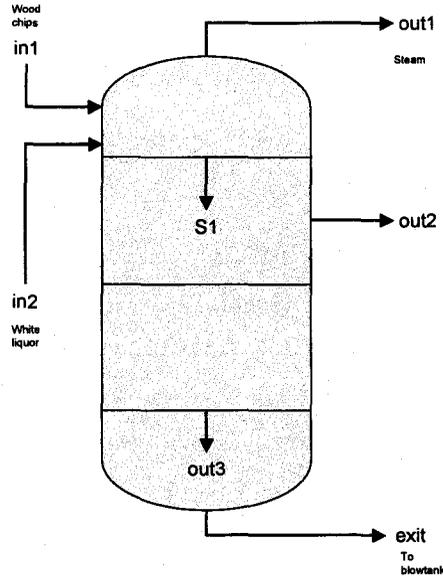


Figure 3.6: Digester model.

from 70°C to 170°C. During cooking, the organic component, lignin (the glue that bonds cellulose fibers together) is partially removed.

The steady-state section of the digester model is derived from the model developed by Wisniewski and Doyle [84] and Balthazaar [8]. The nominal wood chip charge rate is 80 tons/hr at steady state. As lignin is removed from the cellulose fibers in the cooking process, there is a loss in pulp mass that needs to be accounted for. This loss is captured using a parameter defined as the “pulp shrinkage factor”. The pulp shrinkage factor is related to another quantity ζ , the production factor, defined as the ratio of the prevailing production rate to the nominal production rate. The nonlinear relationship between these two variables was developed in Balthazaar’s [8] work. The two pulp shrinkage regression equations are given below:

$$a_{shrink1} = a_0 + a_1 \cdot \zeta + a_2 \cdot \zeta^2 + a_3 \cdot \zeta^3 \quad (3.9)$$

$$a_{shrink2} = b_0 + b_1 \cdot \zeta + b_2 \cdot \zeta^2 + b_3 \cdot \zeta^3 \quad (3.10)$$

where a_0, a_1, a_2, a_3 and b_0, b_1, b_2, b_3 are empirically-determined coefficients [8]. $a_{shrink1}$ represents the pulp shrinkage occurring in the top portion of the digester, while $a_{shrink2}$ is a similar quantity for the bottom part of the digester. A simple mass

balance (with shrinkages) governs the material flow in the top portion of the digester.

$$F_{S1}^P = (1 - a_{shrink1}/100) \cdot F_{in1} \cdot x_1^P \quad (3.11)$$

$$F_{S1}^{DS} = F_{in1} \cdot x_1^{DS} + F_{in2}^{DS} + (a_{shrink1}/100) \cdot F_{in1} \cdot x_1^P \quad (3.12)$$

$$F_{S1}^W = F_{in1} \cdot x_1^W + F_{in2}^W - F_{out1}^{ST} \quad (3.13)$$

The typical composition of softwood (in terms of mass fractions) is $x_1^P = 0.43$, $x_1^{DS} = 0.04$ and $x_1^W = 0.53$ [37]. Because the complexities within a digester are not fully understood, it is a common practice in industry to maintain a constant liquor-to-wood flowrate ratio [37]. The *liquorwoodratio* parameter, which regulates the proportion of liquor used relative to the amount of wood chips fed, is set at 3.6 by mass ([24], [37]).

$$F_{in2} = liquorwoodratio \cdot F_{in1} \quad (3.14)$$

A pressure relief valve at the top of the digester releases steam during the cook process. The amount of steam vented from the digester, F_{out1}^{ST} is related to the inlet water flow [24] by a proportionality factor, $a_{ST} = 0.04$.

$$F_{out1}^{ST} = a_{ST} \cdot (F_{in1} \cdot x_1^W + F_{in2}^W) \quad (3.15)$$

The mass fraction of dissolved solids in the liquor stream entering the digester was obtained by a stoichiometric unit conversion on the concentration of its constituents, 1.0 M NaOH and 0.8M Na₂S [37]. The dissolved solids fraction works out to 0.212, therefore it follows that the water fraction in the stream is $(1 - 0.212) = 0.788$.

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (3.16)$$

$$F_{in2}^W = 0.788 \cdot (F_{in2}^W + F_{in2}^{DS}) \quad (3.17)$$

The material transfer in the bottom part of the digester is modeled by mass balances (accounting for shrinkage).

$$F_{out3}^P = (1 - a_{shrink2}/100) \cdot F_{S1}^P \quad (3.18)$$

$$F_{out3}^{DS} = F_{S1}^{DS} + (a_{shrink2}/100) \cdot F_{S1}^P - F_{out2}^{DS} \quad (3.19)$$

$$F_{out3}^W = F_{S1}^W - F_{out2}^W \quad (3.20)$$

The *blowlinewater fraction* quantity has a value of 0.62, based on industrial data [8]. The relative composition of the liquor exiting the digester in stream 2 is taken to be identical to that in stream 3, and this is modeled using a bilinear equation.

$$F_{out3}^W = blowlinewater\ fraction \cdot F_{S1}^W \quad (3.21)$$

$$F_{out2}^{DS} \cdot F_{out3}^W = F_{out2}^W \cdot F_{out3}^{DS} \quad (3.22)$$

The F_{out3} stream is connected to the exit stream:

$$F_{exit} = F_{exit}^P + F_{exit}^W + F_{exit}^{DS} \quad (3.23)$$

$$F_{exit}^P = F_{out3}^P \quad (3.24)$$

$$F_{exit}^W = F_{out3}^W \quad (3.25)$$

$$F_{exit}^{DS} = F_{out3}^{DS} \quad (3.26)$$

The output of the digester is channelled to an adjacent tank called a blowtank. One of the assumptions made for the purposes of this thesis is that it is possible to set the chip feed to zero to simulate a shutdown. This is only done as a mathematical approximation, and in industrial practice, it is necessary to verify this assumption in order for the results obtained to be practicable.

3.2.3 Buffer Tank Units

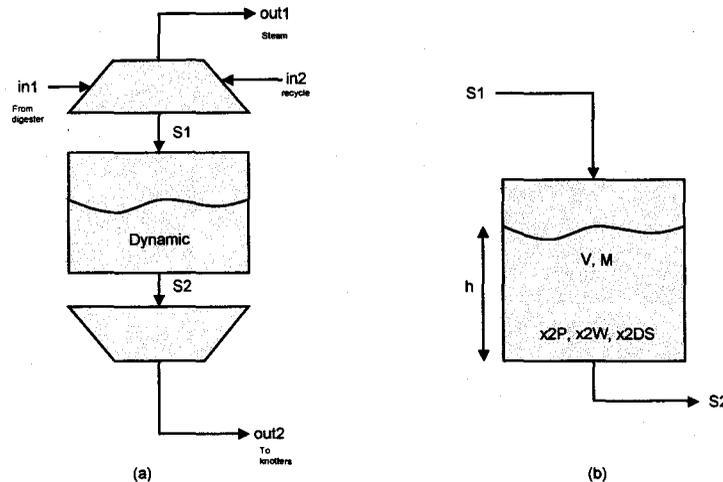


Figure 3.7: (a) Schematic of integrated tank system; (b) Expanded diagram of the dynamic section of the tank.

In this section, the equations that define the operation of a buffer tank are laid down. In order to ensure that the model is generalizable, it is modeled after an integrated blowtank unit, which comprises three sections, two steady-state sections and one dynamic section. The top of the tank, where steam is vented (in a blowtank-type configuration), is assumed to be steady-state. The bottom portion of the tank is agitated and is also assumed to operate at steady state.

The amount of steam vented from the tank (for tank pressure relief) is proportional to the mass fraction of water in the stream entering the tank, and the proportionality factor *steam fraction* is 0.02 [24]. The material balance equations for the top part of the tank is as follows:

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (3.27)$$

$$F_{in1}^P = F_{S1}^P \quad (3.28)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{S1}^W \quad (3.29)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{S1}^{DS} \quad (3.30)$$

$$F_{out1}^W = \text{steam fraction} \cdot F_{in1}^W \quad (3.31)$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (3.32)$$

$$F_{in2}^W = x_{in2}^W \cdot F_{in2} \quad (3.33)$$

$$F_{in2}^{DS} = x_{in2}^{DS} \cdot F_{in2} \quad (3.34)$$

$$F_{S1} = F_{S1}^P + F_{S1}^{DS} + F_{S1}^W \quad (3.35)$$

$$F_{S1}^P = x_1^P \cdot F_{S1} \quad (3.36)$$

$$F_{S1}^{DS} = x_1^{DS} \cdot F_{S1} \quad (3.37)$$

$$F_{S1}^W = x_1^W \cdot F_{S1} \quad (3.38)$$

The dynamic portion of the tank is governed by the equations below. This part of the tank exhibits the dynamic behavior of the buffer capacities.

$$V \cdot \rho_{avg} = M \quad (3.39)$$

$$V = A \cdot h \quad (3.40)$$

$$\frac{d}{dt}M = F_{S1} - F_{S2} \quad (3.41)$$

$$M \cdot \frac{d}{dt}x_2^P = F_{S1} \cdot (x_1^P - x_2^P) \quad (3.42)$$

$$M \cdot \frac{d}{dt}x_2^{DS} = F_{S1} \cdot (x_1^{DS} - x_2^{DS}) \quad (3.43)$$

$$x_2^P + x_2^W + x_2^{DS} = 1 \quad (3.44)$$

where V = volume of material in tank, h = tank level, M = mass holdup, A = area of the base of tank (the tank is assumed to be perfectly cylindrical). ρ_{avg} is the average density of the material in the tank (about 0.900 tons/m³). Strictly speaking, the

overall density of materials in the tank depends on instantaneous dynamic compositions, but the simplifying assumption that the materials are of constant density was made for numerical reasons.

Concentration quantities like x_2^P (mass fraction of pulp in the tank) and x_2^{DS} (mass fraction of dissolved solids), which are state variables in our system, are measurable. The former can be measured on-line using a myriad of methods [19], some of which are:

- pulp slurry conductivity
- pressure drop cause by flow through a fixed length of pipe
- intensity of transmitted microwaves/ultrasonic waves/reflected light
- load on a motor operating an agitator mixer
- head needed to maintain consistent flow through viscosity tube

The mass fraction of dissolved solids is typically measured using conductivity measurements or measurements taken with auto-titrators [57]. With x_2^P and x_2^{DS} known, x_2^W is trivially obtained.

The differential equations are initialized with the following values:

$$V(0) = V_0 \quad (3.45)$$

$$x_2^P(0) = x_1^P(0) \quad (3.46)$$

$$x_2^{DS}(0) = x_1^{DS}(0) \quad (3.47)$$

where $V_0 = 2050 \text{ m}^3$. The initial mass fractions of the pulp and dissolved solids streams are assumed to be the same as their respective compositions in the inlet stream (which essentially means that the system is starting from a steady-state point). This tank model is used as a template for both blowtanks (the tank adjacent to the digester) and standard tanks. Instances of this tank models are derived from this template, and configured to represent specific types of tanks.

3.2.4 Knotting Department

Knotters are designed to screen and remove large undigested chips and wood knots. The rejected knots are collected and recycled to the digester. In our model, we

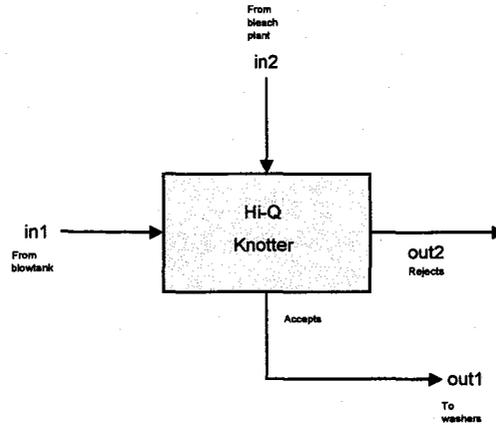


Figure 3.8: Hi-Q knotter.

consider two types of knotters operating in series, a Hi-Q knotter (manufactured by companies such as Ingersoll-Rand and GL&V) and a Jonsson knotter (manufactured by companies like Bird and Lamort). The pulp entering the Hi-Q knotter is mixed with wash liquor and the coarse, undigested chips are screened from the pulp stream. Accepted pulp is channeled to the brownstock washing department, while rejects are sent to the second (Jonsson) knotter for further screening.

Because the Jonsson knotter is functionally the same as the Hi-Q knotter, we will restrict our description to the Hi-Q knotter equations. The reader is referred to Appendix A for the equations describing the Jonsson knotter. The Hi-Q knotter is governed by the following overall and component mass balances:

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (3.48)$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (3.49)$$

$$F_{out1} = F_{out1}^P + F_{out1}^W + F_{out1}^{DS} \quad (3.50)$$

$$F_{out2} = F_{out2}^P + F_{out2}^W + F_{out2}^{DS} \quad (3.51)$$

$$F_{in1}^P = F_{out1}^P + F_{out2}^P \quad (3.52)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (3.53)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (3.54)$$

The ratio of the F_{in2} stream to the F_{in1} stream is measured by the dilution factor, $adil1 = 0.05$ [24]. The pulp loss is represented using a knot rejection factor,

$aknotrej1 = 0.10$ [37]. The pulp/water and water/dissolved solids ratios are assumed to be equal in streams F_{out1} and F_{out2} , and this is enforced using two bilinear equations.

$$F_{in2} = adil1 \cdot F_{in1} \quad (3.55)$$

$$F_{out2}^P = aknotrej1 \cdot F_{in1}^P \quad (3.56)$$

$$F_{out1}^W \cdot F_{out2}^P = F_{out1}^P \cdot F_{out2}^W \quad (3.57)$$

$$F_{out2}^W \cdot F_{out1}^{DS} = F_{out2}^{DS} \cdot F_{out1}^W \quad (3.58)$$

$$F_{in2}^W = water\ fraction \cdot F_{in2} \quad (3.59)$$

Stream F_{in2} is assumed to be made up of 95% water, therefore $water\ fraction = 0.95$.

3.2.5 Washing Department

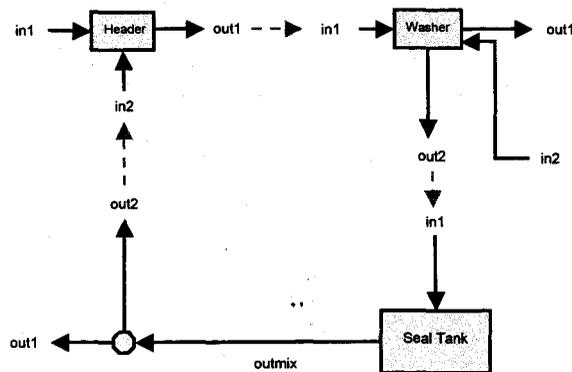


Figure 3.9: Washing department.

Washing (also known as brownstock washing) is a process for removing residual lignin in the pulp stream. The washing department is composed of three major process units: a header box, a vacuum drum washer and a seal tank. The header box mixes the contents of the incoming pulp and liquor streams and channels it to the vacuum drum washer for washing. The washer sprays the pulp stream with wash liquor and rejects are pushed to the seal tank. The contents of the seal tank are recycled to the header box and in some cases, conveyed to the agitated bottoms section of the blowtank as well. Washing stages are usually set up in a counter-current configuration. In a normal pulp mill, there may exist one or more washing stages. For simplicity, we will confine ourselves to a description of a single stage washing department. A detailed dynamic model of the washing system can be found in Kempe [48].

The equations for the header unit [8] are found the Appendix A, and are omitted here. The following subsections contain a description of the vacuum drum washer and the seal tank unit cum splitter, where streams are split and recycled.

Vacuum Drum Washer

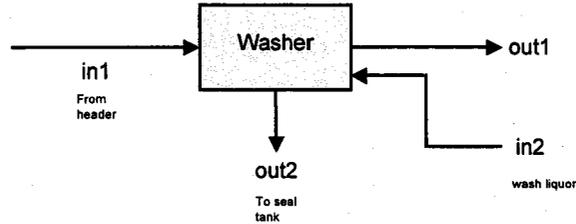


Figure 3.10: Vacuum drum unit washer.

The material flows in the vacuum drum washer is described by the following equations:

$$F_{in1}^P = F_{out1}^P \quad (3.60)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (3.61)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (3.62)$$

We assume that the consistency of the pulp exiting the washer (stream F_{out1}^P) is perfectly controlled and has a value of $outletconsistency = 0.12$ [76] and that the amount of dissolved solids in the shower water stream is $showerfraction = 0.02$.

$$F_{out1}^P = outletconsistency \cdot (F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) \quad (3.63)$$

$$F_{in2}^{DS} = showerfraction \cdot (F_{in2}^W + F_{in2}^{DS}) \quad (3.64)$$

Stream F_{in2} is known as the shower stream, and it conveys liquor to a set of showers to spray washing liquor over the pulp. We assume the mass fraction of water in the F_{out1} stream to be equal that of the F_{out2} stream, and this is expressed in a bilinear equation:

$$F_{out1}^W \cdot (F_{out2}^W + F_{out2}^{DS}) = F_{out2}^W \cdot (F_{out1}^W + F_{out1}^{DS}) \quad (3.65)$$

The displacement ratio (D_R) measures the washing efficiency of the solid in terms of percent dissolved solids removal [57]. This ratio can easily be obtained by collecting

samples on the washers. Unlike the Norden efficiency (below), this measure varies significantly with operating conditions (liquor concentration in shower, dilution factor, etc.)

$$D_R = \frac{(x_b - x_d)}{(x_b - y_c)} \quad (3.66)$$

$$x_d = \frac{F_{out1}^{DS}}{(F_{out1}^{DS} + F_{out1}^W)} \quad (3.67)$$

$$x_b = \frac{F_{in1}^{DS}}{(F_{in1}^{DS} + F_{in1}^W)} \quad (3.68)$$

$$y_c = \text{shower fraction} \quad (3.69)$$

The Norden efficiency [58] represents the number of countercurrent ideal mixing stages required to achieve the same washing performance as the one in the process. One property of the Norden efficiency that makes it suitable as a modeling specification is its lack of sensitivity to changes in the washing process. This allows it to be approximated as a constant over a certain range of operation. In order to calculate the Norden efficiency (N_{eff}), two quantities, R and W need to be defined.

R is the wash liquor ratio (ratio of liquor entering to liquor leaving the pulp):

$$R = \frac{(F_{in2}^W + F_{in2}^{DS})}{(F_{out1}^W + F_{out1}^{DS})} \quad (3.70)$$

and W is the liquor weight ratio (ratio of filtrate to liquor in the entering stream):

$$W = \frac{(F_{out2}^W + F_{out2}^{DS})}{(F_{in1}^W + F_{in1}^{DS})} \quad (3.71)$$

The Norden efficiency (N_{eff}) is defined implicitly in this equation:

$$D_R \cdot (W \cdot R^{N_{eff}} - 1) = (W \cdot R^{N_{eff}} - R) \quad (3.72)$$

The value of the Norden efficiency for the washer is assumed to be 3.00 (an average value based on Turner et. al. [76].)

One of the assumptions made is that it is possible to set turn off all flows to the washer in order to simulate a shutdown. This assumption needs to be examined further when applying the results to a real plant.

3.2.6 Seal Tank and Splitter

The seal tank serves to create sufficient vacuum through the drop leg for proper washer operation. It is also used to store filtrate necessary for startup and functions

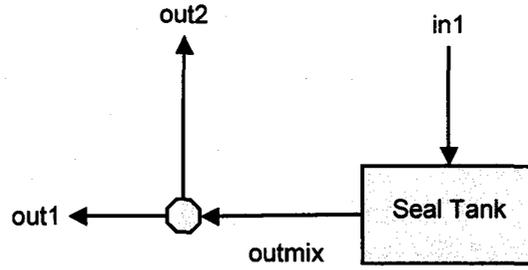


Figure 3.11: Seal tank and splitter.

as a buffer as well. The seal tank can be modeled simply as a continuous stirred tank. The basic component mass balances are as follows:

$$F_{in1} = F_{in1}^W + F_{in1}^{DS} \quad (3.73)$$

$$F_{outmix} = F_{outmix}^W + F_{outmix}^{DS} \quad (3.74)$$

$$x_{in1}^W \cdot F_{in1} = F_{in1}^W \quad (3.75)$$

$$F_{outmix}^W = x_{outmix}^W \cdot F_{outmix} \quad (3.76)$$

The dynamic portion of the seal tank is as follows:

$$\frac{d}{dt}M = F_{in1} - F_{outmix} \quad (3.77)$$

$$M \cdot \frac{d}{dt}x_{outmix}^W = F_{in1} \cdot (x_{in1}^W - x_{outmix}^W) \quad (3.78)$$

$$(x_{outmix}^W + x_{outmix}^{DS}) = 1 \quad (3.79)$$

The splitter connected to the seal tank is modeled as follows:

$$F_{outmix}^W = F_{out1}^W + F_{out2}^W \quad (3.80)$$

$$F_{outmix}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (3.81)$$

$$F_{out2}^W \cdot (F_{out1}^W + F_{out1}^{DS}) = F_{out1}^W \cdot (F_{out2}^W + F_{out2}^{DS}) \quad (3.82)$$

The F_{out2} stream is returned to the header box of the washing stage while F_{out1} stream is recycled (not modeled).

3.2.7 Screening Department

Shives refer to wood whose size is between that of processable pulp and knots. They hinder the bleaching process and cause deformities in formed sheets, and therefore

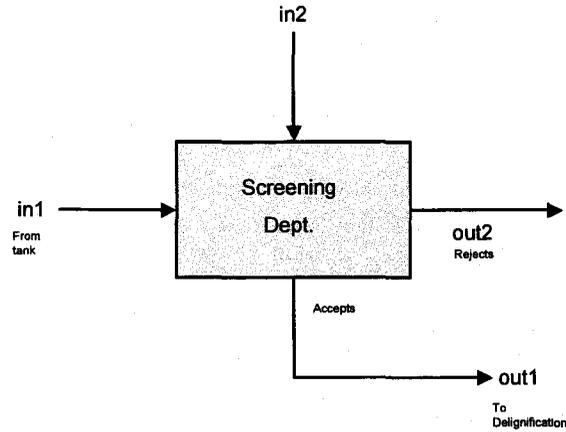


Figure 3.12: Screening department.

need to be removed via the screening process. A simplified pressure screen model is described here. The function of pressure screens is to filter shives from the process stream. Balthazaar [8] offers a three-step screen model, which we have simplified into a one stage model while retaining the macro-effects and properties of the process.

The screening model is composed of a component mass balance:

$$F_{in1}^P = F_{out1}^P + F_{out2}^P \quad (3.83)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (3.84)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (3.85)$$

The pulp/water and pulp/dissolved solids compositions between streams F_{out1} and F_{out2} are taken to be equal:

$$F_{out1}^P \cdot F_{out2}^W - F_{out1}^W \cdot F_{out2}^P = 0 \quad (3.86)$$

$$F_{out1}^P \cdot F_{out2}^{DS} - F_{out1}^{DS} \cdot F_{out2}^P = 0 \quad (3.87)$$

5% of pulp is lost in a screening unit due to the extraction of shives, therefore the pulp loss coefficient, $pulplosscoeff = 0.05$. The mandated acceptable output consistency for downstream processing in O_2 -delignification unit is 4.5%, which leads to $outletconsistency = 0.045$. The dilution stream is taken to have a water content of

95%, which means $waterfraction = 0.95$.

$$F_{out1}^P = pulplosscoeff \cdot F_{in1}^P \quad (3.88)$$

$$F_{out1}^P = outletconsistency \cdot (F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) \quad (3.89)$$

$$F_{in2}^W = waterfraction \cdot (F_{in2}^W + F_{in2}^{DS}) \quad (3.90)$$

The rejects from the screening department are discarded.

3.2.8 Delignification Department

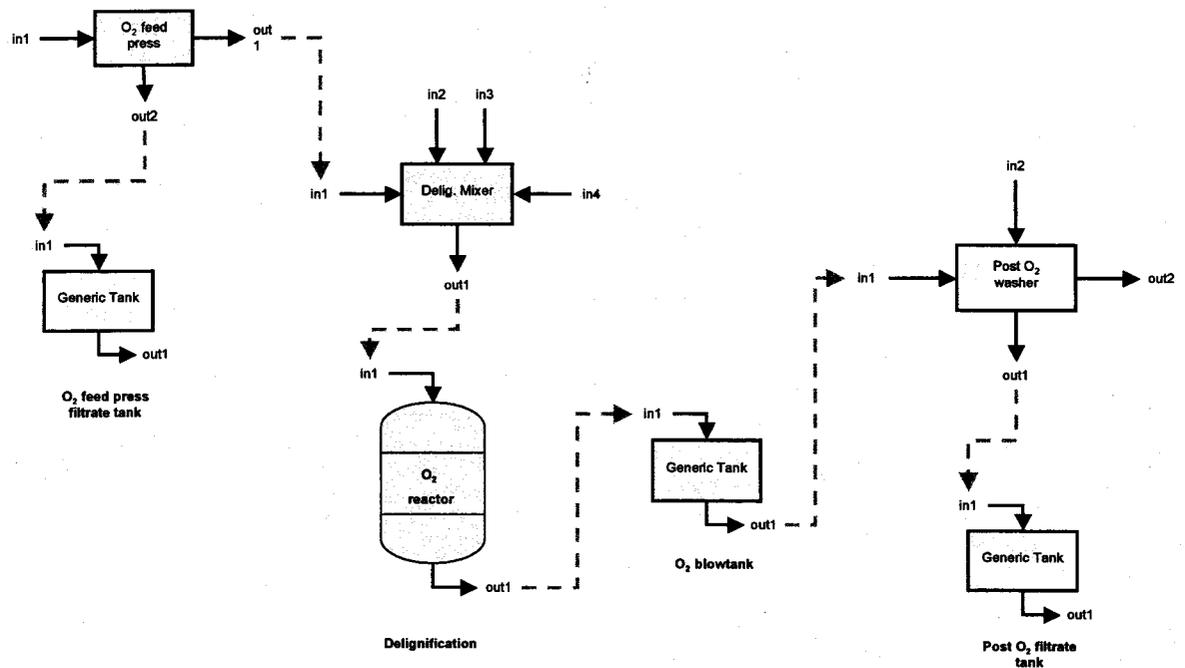


Figure 3.13: Delignification department.

At this stage of the process, the pulp is typically still dark in color (due to the bound lignin) and will need to undergo bleaching. Unfortunately bleaching entails the use of chlorine, the emissions of which have a deleterious effect on the environment. In order to reduce the utilization of chlorine in bleaching, oxygen delignification is performed to remove as much as of the residual lignin as possible. This has the positive effect of reducing the amount of chlorine required by the chlorination reactor in the bleaching department downstream.

In delignification, the pulp stream is heated with steam and mixed with caustic. It is then run down the O₂ reactor in a counter-current fashion. In this process, the impregnated pulp stream reacts with oxygen and a separation of lignin from the cellulose fibers occurs. It should be mentioned if this is carried out too far, carbohydrate degradation also occurs, which results in an undesirable drop in viscosity. Process conditions must therefore be tightly controlled to preserve the integrity of the pulp.

The O₂ delignification department is made up of several units. The major ones are the O₂ feed presses, mixer, reactor and the post-O₂ washer. The O₂ feed press functions to raise the consistency of the pulp from 4.5% to 30% and is modeled by a simple mass balance. The mixer mixes the pulp with steam and caustic, while the reactor is the main vehicle for performing the delignification. Generic tanks have fast dynamics and are modeled without accumulation. The post-O₂ washer removes remaining dissolved solids and is modeled by a straightforward mass balance.

In this section, we will focus on the models of the delignification mixer and reactor. The equations for the other models can be found in Appendix A.

Delignification Mixer

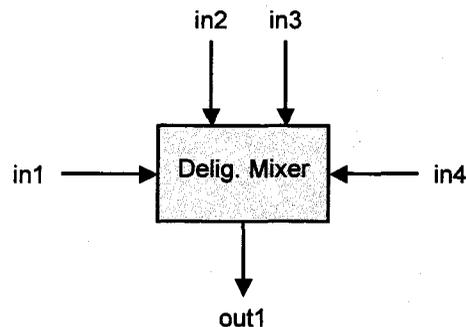


Figure 3.14: Delignification mixer.

The delignification mixer is modeled as a perfect steady-state mixer [8]. A mass

balance around the mixer yields the following equations:

$$F_{in1}^P = F_{out1}^P \quad (3.91)$$

$$F_{in1}^{DS} + F_{in2}^{DS} + F_{in3}^{DS} = F_{out1}^{DS} \quad (3.92)$$

$$F_{in1}^W + F_{in2}^W + F_{in3}^W + F_{in4}^{ST} = F_{out1}^W \quad (3.93)$$

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (3.94)$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (3.95)$$

$$F_{in3} = F_{in3}^W + F_{in3}^{DS} \quad (3.96)$$

$$F_{out1} = F_{out1}^P + F_{out1}^W + F_{out1}^{DS} \quad (3.97)$$

The pulp stream (F_{in1}) enters the mixer at 25°C, and likewise for the caustic (F_{in2}) and magnesium sulfate (F_{in3}) streams. Magnesium sulfate is added for pulp protection. F_{in4}^{ST} is the stream containing medium-pressure steam. The energy balance is used to calculate the steam consumption of the system.

$$L1 = 0 \quad (3.98)$$

$$L2 = 0 \quad (3.99)$$

$$L3 = 0 \quad (3.100)$$

$$L4 = F_{in4}^{ST} \cdot (H_{stm} - H_{ref}) \quad (3.101)$$

$$R1 = F_{out1} \cdot c_{ppulp1} \cdot (T_{set} - T_{ref}) \quad (3.102)$$

$$(L1 + L2 + L3 + L4) = R1 \quad (3.103)$$

where $L1$, $L2$, $L3$ and $L4$ are the enthalpies of streams F_{in1} , F_{in2} , F_{in3} and F_{in4} respectively. $R1$ is the enthalpy of the F_{out1} stream. Using a reference temperature of $T_{ref} = 25^\circ\text{C}$, the reference enthalpy of saturated steam is $H_{ref} = 2547.3\text{MJ/t}$. The enthalpy of medium-pressure steam, $H_{stm} = 3267.5\text{MJ/t}$ at 49 bar and 415°C. This steam is used to heat the mixture to the required temperature of a setpoint temperature of $T_{set} = 100^\circ\text{C}$. c_{ppulp1} is the average heat capacity of the exiting pulp, whose value is 3.972 MJ/t·°C.

The chemical dosage equations which determine the composition of the inlet streams

are as follows:

$$F_{in2}^{DS} = NaOHdosage \cdot F_{in1}^P \quad (3.104)$$

$$\frac{F_{in2}^{DS}}{F_{in2}^W + F_{in2}^{DS}} = 0.08 \quad (3.105)$$

$$F_{in3}^{DS} = MgSO_4dosage \cdot F_{in1}^P \quad (3.106)$$

$$\frac{F_{in3}^{DS}}{F_{in3}^W + F_{in3}^{DS}} = 0.045 \quad (3.107)$$

$$(3.108)$$

where $NaOHdosage = 0.02$ t NaOH/t pulp and $MgSO_4dosage = 0.002$ t $MgSO_4$ /t pulp [37]. The concentrations of the NaOH and $MgSO_4$ streams are 8% and 4.5% respectively.

O₂ delignification reactor

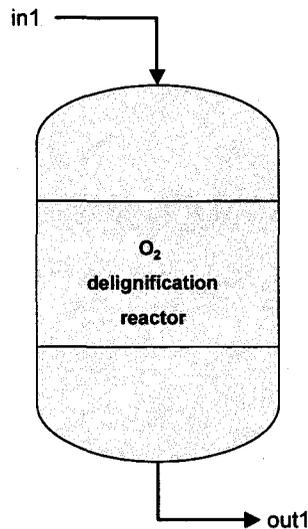


Figure 3.15: Delignification reactor.

The O₂ delignification reactor is a counter-current reactor where the lignin in the pulp stream reacted with oxygen. The following equations are derived from a mass

balance in the reactor.

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (3.109)$$

$$F_{out1}^P = (1 - 1 \times 10^{-2} \cdot a_{O2}) \cdot F_{in1}^P \quad (3.110)$$

$$F_{out1}^{DS} = F_{in1}^{DS} + 1 \times 10^{-2} \cdot a_{O2} \cdot F_{in1}^P \quad (3.111)$$

$$F_{out1}^W = F_{in1}^W \quad (3.112)$$

where a_{O2} is a shrinkage factor to account for the fact that pulp shrinkage occurs in the reactor. This factor is related to the production factor (ζ) through the following regression equation derived from industrial data [8]

$$a_{O2} = a_0 + a_1 \cdot \zeta + a_2 \cdot \zeta^2 + a_3 \cdot \zeta^3 \quad (3.113)$$

where the values of the coefficients are $a_1=-0.0022$, $a_0=2.301$, $a_3=-0.0113$, and $a_2=0.0116$.

3.3 Formulation Issues and Modeling Techniques

In this section, techniques related to the modeling of DAE systems are discussed. A brief summary of numerical pitfalls is provided.

3.3.1 Objective Functionals with Integral Terms

Objective functionals containing integral terms are easily accommodated in our dynamic optimization problem structure. The following objective functional:

$$\max \int_0^{t_f} \Phi(t) dt \quad (3.114)$$

can easily be rewritten into the standard DAE optimization form by introducing a dummy variable z , as follows:

$$\max z(t_f) \quad (3.115)$$

s.t.

$$\frac{dz(t)}{dt} = \Phi(t) \quad \text{with } z(0) = 0 \quad (3.116)$$

3.3.2 Numerical Issues

A significant portion of the modeling effort was devoted to detecting and overcoming numerical problems with the model. We present here a list of selected numerical pitfalls and their corresponding solutions.

1. **Avoiding subtly degenerate constraints.** Constraints that enforce the non-negativity of variables abound in chemical process modeling, e.g. $x \geq 0$. In the presence of such constraints, tacking on additional constraints $x = 0$ (to simulate failures, for instance) may give rise to degeneracy in the constraint set. A strategy to circumvent this is to rewrite the constraints as follows: $x \geq 0$ and $x \leq \epsilon$, where ϵ is a small number, e.g. 1×10^{-4} .
2. **Scaling.** When values of certain variables/expressions are orders of magnitude larger or smaller than the rest of the equation system, the linear systems (Jacobians, Hessians) required to solve the nonlinear problem become ill-conditioned. A standard way of treating this problem is to multiply both the left and right-hand-sides of equations by an appropriate scaling factor that will bring their magnitudes into a reasonable, uniform range, such as $[0.01, 100]$. Most solvers perform some degree of automatic scaling internally, but manual equation scaling tends to be preferable as it almost always leads to better results because the modeler deemed to have more insight into the model than the solver has. The reader is referred to McCarl [54] for more information on this topic.
3. **Avoiding complex-valued expressions.** Suppose we have an expression where the variable x raised raised to n -th power, x^n (where $x, n \geq 0$). Certain solvers like IPOPT perform relaxations on user-supplied bounds (i.e. $x \geq 0 \Rightarrow x \geq -\epsilon$, where $\epsilon = 1 \times 10^{-8}$) that render the evaluation of this expression impossible in some cases. We observe this behavior when the value of x is negative and n is non-integral, where the x^n expression veers into the complex realm, thus causing an evaluation error in the solver. A remedy is to declare bounds that are slightly positive, i.e. $x \geq 1 \times 10^{-8}$ instead of $x \geq 0$.
4. **Division by Zero errors.** The division of any variable by another variable (e.g. $x/y = 2$) is to be avoided in general. During solver iterations, it is possible for the denominator to take a value of zero, thus inadvertently triggering a division-by-zero operation. An effective strategy is to eliminate denominators

by multiplying both sides of the equation by denominator terms (e.g. $x = 2y$). A side benefit of this is that the derivatives of the pre-multiplied expression become much simpler.

5. **Avoiding $\mathbf{x} = \mathbf{0}$ as the initial guess.** It is commonly recognized that the zero vector ($\mathbf{x} = \mathbf{0}$) is poor initialization point for nonlinear systems. One of the reasons for this, apart from the fact that the $\mathbf{0}$ point may be very far away from the solution, is the effect it has on derivatives of product terms. Consider the bilinear expression, $f(x, y) = x \cdot y$, whose gradient is the following:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}$$

When a vector of zeros is used to initialize this system, the initial derivative vector becomes $\mathbf{0}$. This gives the solver the wrong impression that the function does not depend on the variable [23]. The solution is to initialize the problem with a feasible point. In our case, we supplied the output of a process simulator as an initial guess.

6. **Simpler expressions are better than complex ones.** It is better for compound expression such as $f = \exp(F(G(x)))$ to be broken up and rewritten simply as follows:

$$\begin{aligned} f_1 &= \exp(f_2) \\ f_2 &= F(f_3) \\ f_3 &= G(x) \end{aligned}$$

This latter form results in simpler derivatives and sparser Hessians.

7. **Relaxing equalities.** When one encounters infeasibilities caused by equality constraints that are too tight (e.g. $x = 40$), the following relaxation is often helpful: $40 - \epsilon \leq x \leq 40 + \epsilon$. This relaxation was instrumental in aiding problem convergence in the presence of level-restoration constraints (described in the next chapter).

3.4 Chapter Summary

In this chapter, we described the details of the Kraft paper mill model and the assumptions that are made in the course of its development. We described our computational

environment and gave a brief sketch of the features in our in-house developed modeling language for dynamic optimization, MLDO. We concluded the chapter by discussing numerical issues related to model formulation and supplied a list of common pitfalls associated with modeling.

Chapter 4

Dynamic Optimization under Shutdown Conditions

In theory, there is no difference between theory and practice.

But, in practice, there is.

- Jan L.A. van de Snepscheut, Dutch computer scientist

This chapter describes a control scheme for optimal handling of unit shutdowns in a plant. Dynamic optimization under shutdown conditions is briefly introduced, followed by various approaches used to handle uncertainty in the downtime estimate, with the primary method being re-optimization. Several case studies are presented. The issue of avoiding policies that induce multiple shutdowns is considered and several methods for penalizing shutdowns are described.

4.1 Introduction

In a typical chemical plant, process units are shut down from time to time either for maintenance or due to equipment failure. From an operations perspective, unit shutdowns can be classified as either critical or non-critical, with the former being those that lead to the shutdown of the entire plant and the latter, those that do not.

In the case of critical shutdowns, the entire system is invariably forced to shut down, and under such circumstances the usefulness of an optimal control policy is limited.

Under non-critical shutdown scenarios however, it is frequently possible for an operator to pursue certain courses of action that will permit the unaffected units to continue operating to some degree. Possible courses of action include reconfiguring the process pathways, re-routing material streams, slowing down production, making use of buffer capacities and so on.

One way to determine if the type of shutdown in a plant is of the critical or non-critical variety (which has direct consequences as to whether our proposed control scheme is viable under that particular situation) is to verify if the model of the plant is feasible under optimization. For a correctly modeled plant with shutdown constraints enforced, an infeasible solution is often an indication that the shutdown is critical, possibly signifying that critical dependencies are not available or that the plant is not dynamically operable in some way, and thus a system shutdown is imminent.

The case studies in our work are based on a model of a subset of a Kraft paper mill (described in the modeling chapter) where a process unit is shut down and taken off-line for a period of time, and is subsequently restored. Based on an estimate of the downtime (specified by the operator), our proposed control scheme is used to compute and implement a set of optimal control trajectories that accommodates the shutdown.

In this study, we will be examining the optimal use of buffer capacities and the manipulation of production rates and recycles during the shutdown period. The essential goal of our control scheme is to arrive at a set of optimal control inputs for controlling the process transitions during the shutdown and restoration periods. This work represents our preliminary progress towards this goal.

A critical parameter in the control scheme is the downtime estimate. A few approaches for dealing with the uncertainty in the downtime estimate will be considered in detail, with an emphasis on re-optimization.

There are two types of policies for handling process unit shutdowns:

1. **Pre-emptive Policy**, where the shutdown is known in advance, so the control system is able to anticipate it and make advance preparations. Example: scheduled maintenance of a reactor.
2. **Reactive Policy**, where the shutdown is unscheduled. Example: equipment failure.

In this study, we will only be considering reactive shutdown policies in order to concentrate on issues surrounding uncertain downtime estimates. However, it should be noted that accommodation of pre-emptive type policies is straightforward within the proposed framework below.

4.2 Dynamic Optimization Problem Formulation

The problem of determining optimal control inputs for operating the plant under shutdown conditions is cast into a dynamic optimization framework. The formulation is as follows:

Objective

$$\max_{\mathbf{u}(t)} [\Phi_{\text{economics}} - \varphi] \quad (4.1)$$

subject to

Economics-based Objective Function

$$\Phi_{\text{economics}} = \sum_m \left[C_m \int_0^{t_f} F_m dt \right] \quad (4.2)$$

Model Equations and Constraints

$$\mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \quad (4.3)$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \quad (4.4)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) \leq 0 \quad (4.5)$$

Variable bounds

$$\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U \quad (4.6)$$

$$\mathbf{z}_L \leq \mathbf{z}(t) \leq \mathbf{z}_U \quad (4.7)$$

$$\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \quad (4.8)$$

for $t \in [0, t_f]$

Initial conditions

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (4.9)$$

Restoration constraints

$$\mathbf{x}_0 - \epsilon \leq \mathbf{x}(t) \leq \mathbf{x}_0 + \epsilon \quad \text{for } t_{res} < t \leq t_f \quad (4.10)$$

$$\mathbf{z}_0 - \epsilon \leq \mathbf{z}(t) \leq \mathbf{z}_0 + \epsilon \quad \text{for } t_{res} < t \leq t_f \quad (4.11)$$

$$\mathbf{u}_0 - \epsilon \leq \mathbf{u}(t) \leq \mathbf{u}_0 + \epsilon \quad \text{for } t_{res} < t \leq t_f \quad (4.12)$$

Shutdown constraints

$$F_{in,unit}(t) = 0 \quad \text{for } t_{start} \leq t \leq t_{end} \quad (4.13)$$

where

t_f = final time

t_{res} = time at the end of restoration period

t_{start} = time at which shutdown commences

t_{end} = time at which shutdown ends

φ = general penalty function. Penalty function forms for preventing induced shutdowns are described in section 4.5

$\mathbf{x}(t)$ = differential state vector

$\mathbf{z}(t)$ = algebraic state vector

$\mathbf{u}(t)$ = control input vector

ϵ = relaxation parameter

$F_{in,unit}(t)$ = mass flow into a process unit, where *unit* denotes a specific process unit that is shut down

$\Phi_{economics}$ = profit function

m = materials produced or consumed = {pulp, chips, chemicals, steam, liquor}

C_m = price of material m

F_m = flowrate of material m

Material	Model Variable, F_m	Prices, C_m (\$/ton)
Pulp	post2washer(1).out2P	725
Chips	digester(1).in1	-25
Chemicals	deligmixer(1).in2	-100
Steam	deligmixer(1).in4ST	-7.31
Energy from Black Liquor	digester(1).out2DS	0.348

Table 4.1: Prices of Materials, based on ([8], [27], [35])

The model used in this work was a differential-algebraic equation (DAE) system, but the present formulation readily admits other model types, such as state-space

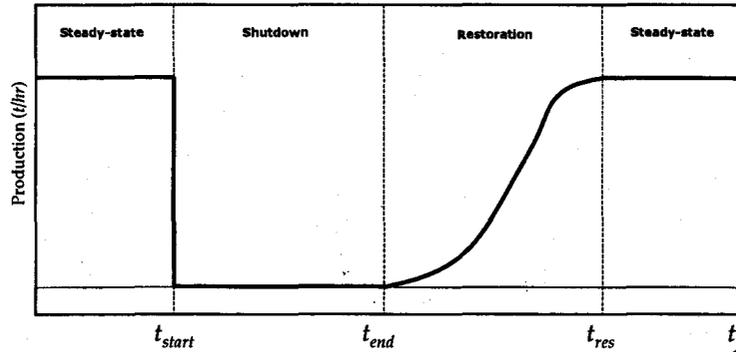


Figure 4.1: Phases of the shutdown process – steady-state, shutdown (t_{start} – t_{end}), restoration (t_{end} – t_{res}), steady-state.

models (obtained for example via subspace identification), Laplace transform transfer function models (for which a state-space realization is possible), and others.

Figure 4.1 shows the different phases of the shutdown process. Initially, a plant will be running at a certain steady-state operating point. The shutdown phase begins at t_{start} , when all input and output flows to a failing/shut-down process unit are forced to be 0. The shutdown phase continues until t_{end} . In between t_{start} and t_{end} , we assume measures are taken to restore or repair the unit. At time t_{end} , the unit is deemed to be ready for operation, and the restoration phase begins. In the restoration phase, control actions are prescribed to the plant to return it to its original steady-state operating point. The restoration phase terminates at time t_{res} , a juncture at which the plant has been successfully restored to normal steady-state operation.

The sample time used in our simulations was 0.5 hour. State and input restoration constraints (equations 4.10–4.12) are imposed for the purpose of bringing the system to its pre-shutdown steady-state at the end of the restoration period. This is important as there are reliability and economic implications to not restoring the process to its original state, discussed below.

From the viewpoint of reliability, it is imperative that the buffer tank levels be restored to their original levels after the shutdown/restoration transition period so that the system is at a state where it is prepared to accommodate subsequent shutdowns should they occur. Balthazaar [8] demonstrated that the effect of these constraints corresponds to enforcing integral action. The restoration of constraints also has a

bearing on the optimality of the problem. Without these restoration constraints, the optimizer will prescribe control actions to empty out the tanks to obtain as much product as possible out of the last unit (due to the pulp production throughput's influence on the objective function). Allison [4] demonstrated that the optimal solution for such problems over a fixed horizon is to drive the storage vessels empty. This is clearly undesirable as it deprives the plant of the ability to anticipate any further shutdowns. Also, in such scenarios, inventory costs need to be taken into account.

From an economic standpoint, compositions in the tanks must also be restored to their pre-shutdown levels. In the simulations, it was discovered that if the pulp concentrations in the buffer tanks were not required to return to their original values after the restoration period, the optimizer may prescribe a set of input actions that maximizes the pulp throughput on the outlet of the system while also diluting the pulp concentrations in the tanks, which is undesirable.

4.2.1 Assumptions

A simplifying assumption is made where either the startup procedure of a unit (i.e. a sequence of discrete actions such as switching or priming pumps, redirecting flows, etc.) is executed off-line or that start-up time is small relative to the overall process dynamics. The implication of both these assumptions is that a unit is deemed to be immediately ready for operation at the end of a shutdown phase (i.e. at t_{end} , refer to Figure 4.1). In our view, this appears to be a reasonable assumption in cases where unit startup dynamics are fast. In other cases, it may be necessary to model the dynamics of startup.

Another assumption made is that the shutdown of a unit is perfectly modeled by turning off the inlet/outlet flows to that unit, and that the relevant shutdown procedure for a unit is followed. This shutdown procedure is not modeled; we assume that a manual or an automated procedure for start-ups and shutdowns is in place [42].

4.3 Case Study 1: Individual Shutdowns in Different Units

In this case study, the effect of shutdowns triggered at various locations relative to the buffers is investigated. In all the cases below, the shutdown duration is 6 hrs. The simulations are performed for a 24 hour period. The three individual failures are in three different units situated before (digester), between (Hi-Q knotter) and after (O_2 -delignification reactor) the buffer capacities (refer to the simplified schematic in Figure (4.2)). There is also a buffering tank (the sealtank) in the washing department which determines the internal recycle flowrates. When a process unit shuts down, an entire department is typically usually forced to shut down unless there is internal buffering capacity in the departments. The manipulated variables and their bounds are given in Table 4.2. The capacities of the tanks are listed in Table 4.3.

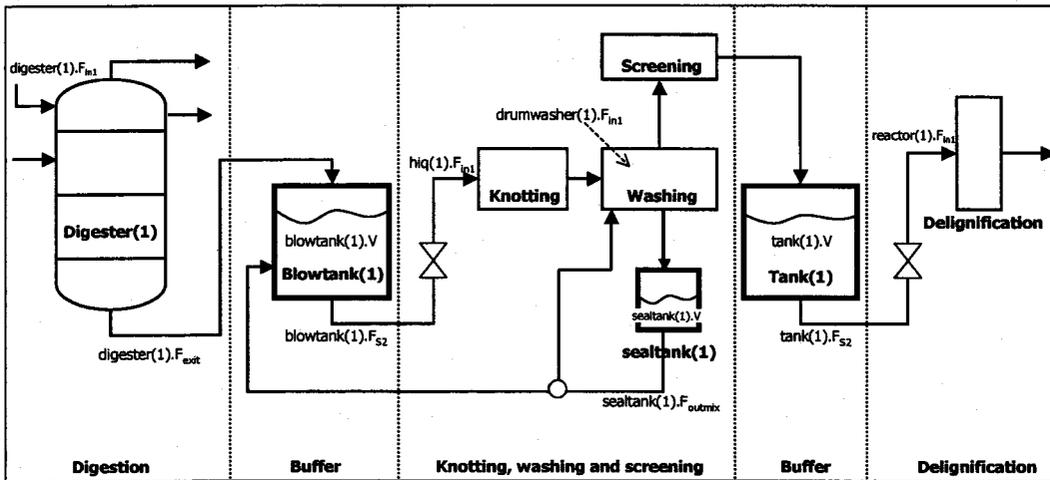


Figure 4.2: Simplified schematic of plant model.

The plant is assumed to be operating at steady-state initially, when the shutdown suddenly occurs. The economics-driven objective function was used for this problem. A slight move suppression penalty (equation 4.14) was found to significantly aid convergence and improve the robustness of the solution, therefore it is applied throughout for consistency. Balthazaar's two-tiered approach was not applied in the examples due to numerical difficulties encountered.

$$\varphi = \rho \sum_{i=1}^{N_{inp}} \sum_{k=1}^{N_{samp}} \Delta u_{i,k}^2 \quad (4.14)$$

Manipulated Variables	Min (t/h)	Max (t/h)
digester(1). F_{in1}	0	80
blowtank(1). F_{S2}	0	456
sealtank(1). F_{outmix}	0	600
tank(1). F_{S2}	0	456

Table 4.2: Manipulated Variables

Tank Name	Capacity (m ³)	Min (m ³)	Nominal (m ³)	Max (m ³)
blowtank(1)	2050	205	1025	1845
sealtank(1)	310	20	130	280
tank(1)	2050	205	1025	1845

Table 4.3: Tank Capacities, Nominal Levels And Upper/Lower Bounds

where $\rho = 0.1$, $u =$ control input variable, N_{inp} = total number of input variables, N_{samp} = total number of sample periods.

The cases are:

1. Case 1: Failure in digester only

This failure occurs upstream of all buffer tanks. The chip feed stream is shut off by setting the value of the flow variable digester(1). F_{in1} to zero. As a consequence, the white liquor inlet stream is automatically shut off because its value is held in proportion to the chip feed stream. In a digester cleanout, all residual material is rerouted and disposed, therefore no new material enters the blowtank during the shutdown.

Observations

A shutdown in the digester has a significant impact on the process, as no pulp can be sent to downstream units for processing. Referring to trajectories in Figure 4.3, the output from the digester (digester(1). F_{exit}) is zero during the failure, therefore no material is blown into the blowtank during this time. In order to keep downstream units running, the blowtank needs to continue discharging its contents, albeit at a lower rate to avoid from emptying itself out. This decrease in flowrate can be seen in the trajectory for blowtank(1). F_{S2} where the outlet flowrate is gradually decreased and the blowtank's accumulation level (represented by the volume of material, blowtank(1). V) steadily descends. After the

failure period is over, the contents of the blowtank are built up again by keeping the blowtank discharge rate low ($\text{blowtank}(1).F_{S2}$) and the digester production high ($\text{digester}(1).F_{exit}$) for a period. This continues until the level of blowtank reaches its original steady state.

During the failure phase, the flowrates into downstream units such as the Hi-Q knotter ($\text{hiq}(1).F_{in1}$) and vacuum drum washer ($\text{drumwasher}(1).F_{in1}$) decrease, which reflects a lowered production caused by the upstream failure. In the restoration phase, production is gradually restored and the flowrates of these units are returned to their original levels.

It is observed that the volume of the material in the sealtank ($\text{sealtank}(1).V$) increased during the failure and restoration phases. This is a result of control actions that constrict the amount of material discharged from the sealtank, which essentially corresponds to limiting the recycle flowrate to the header box of the washing department and to blowtank(1). This is done in order to satisfy restoration constraints. During the digester failure and restoration, the feed to the O₂ delignification experienced a transient drop, but does not shut down. Thus, we can see that throughout the shutdown and restoration of the digester, the downstream units manage to remain in operation.

2. Case 2: Failure in Hi-Q knotter only

This failure occurs in between buffer tanks. The flowrate into the Hi-Q knotter, $\text{hiq}(1).F_{in1}$ is shut off for this failure, effecting a shutdown in the knotter.

Observations

The reader is referred to trajectories in Figure 4.4. During the failure, the digester continues to produce pulp at a high rate and production is not severely impacted. The production rate of the digester is scaled down briefly (by decreasing the chip feed, refer to $\text{digester}(1).F_{in1}$ trajectory) in order to enable the blowtank to operate within its volume constraints. The blowtank discharge ($\text{blowtank}(1).F_{S2}$) which is connected to the inlet of the Hi-Q knotter ($\text{hiq}(1).F_{in1}$) is shut off to accommodate the failure. This leads to a build-up of material in the blowtank, as witnessed by the increase in the volume of material accumulated ($\text{blowtank}(1).V$), which is subsequently discharged as the Hi-Q knotter is restored.

This effect of this failure cascades to the washing and screening departments, as there are no buffer capacities separating the knotter from these downstream units. One of the consequences of this is that the recycle loop from the sealtank

to washing department and to the blowtank has to be shut off during this failure (illustrated in the $\text{sealtank}(1).F_{outmix}$ trajectory). The shut off of the recycle causes the material volume in the seal tank to plateau ($\text{sealtank}(1).V$). As the knotter is being restored, the contents of the sealtank are depleted further in order to start the recycle again. Once the restoration is complete, the sealtank level is restored to its previous state.

Throughout the failure and restoration, the volume of material in the tank(1) buffer capacity drops temporarily but inventory is built up again once the knotter is restored. The delignification reactor experiences a decrease in its feed ($\text{reactor}(1).F_{in1}$) but is prevented from shutting down.

3. Case 3: Failure in O₂-delignification reactor only

This failure occurs downstream of the major buffer tanks. The flow through the O₂ reactor, $\text{reactor}(1).F_{in1}$ is shut off for the failure period.

Observations

Refer to trajectories in Figure 4.5. The digester is continues operating at steady-state and is not affected by the shutdown. The inventory in tank(1) (see $\text{tank}(1).V$) downstream is built up in the anticipation of the O₂ delignification reactor resuming its processing as soon as it comes up.

As soon as the shutdown phase is over and the O₂ delignification reactor is up and running, the restoration phase kicks in and the blowtank material levels ($\text{blowtank}(1).V$) are made to rise to their original levels through a decrease in the blowtank discharge flowrate ($\text{blowtank}(1).F_{S2}$). Meanwhile, tank(1) discharges material from its storage to the O₂ reactor. The reactor experiences a surge in its feed, which gradually decreases as the system reaches the end of the restoration phase. This type of system behavior tracks with intuition, and is a clear demonstration of how the optimal control policy is able coordinate buffer capacities to minimize overall production losses in the event of shutdowns.

General Results

One general observation that can be made from these results is that the further downstream a failing unit is, the less of a loss of production is incurred (that is, the higher the overall profit achievable by the optimization), as shown by the profits and pulp produced in the 3 cases (Table 4.3).

Failure in	Figure	Profit (\$)	Pulp Produced (t)
Digester	4.3	81,164	150
Hi-Q knotter	4.4	102,106	189
O ₂ -delignification	4.5	111,294	206

Table 4.4: Plant Profits and Throughputs under Unit Failures

When a failure occurs upstream, it will invariably result in loss of material for the downstream processing. If adequate stored material is available, then the downstream train can continue processing. In order to meet restoration constraints, the processing rate may need to be reduced or throttled up in some cases.

4.4 Uncertainty in Downtime Estimate, d_{est}

The estimated duration of the shutdown (d_{est}) is an important parameter in calculation of the optimal trajectories because it dictates the shape of the resulting trajectories and has a negative correlation with the overall pulp throughput achievable.

The downtime estimate would typically be provided by the operator to the control system, based on past operational experience or direct information about the prognosis of the shutdown. In practice, this estimate will not correspond exactly to the actual downtime (d_{act}) for various reasons, not least the fact that it may be very difficult in some cases to make an accurate prediction of a unit's downtime. Therefore it is necessary to look into ways to either account for the uncertainty in the downtime or including some feedback mechanism so that the initial estimate may be revised and the original trajectories corrected.

In this section, we will study a few approaches for dealing with the uncertainty in the downtime parameter:

1. Implementing the initial trajectory with naïve adjustments (4.4.1)
2. Optimization Under Uncertainty using a multi-scenario approach (4.4.2)
3. Trajectory Re-optimization using Feedback of Downtime Estimate (4.4.3)

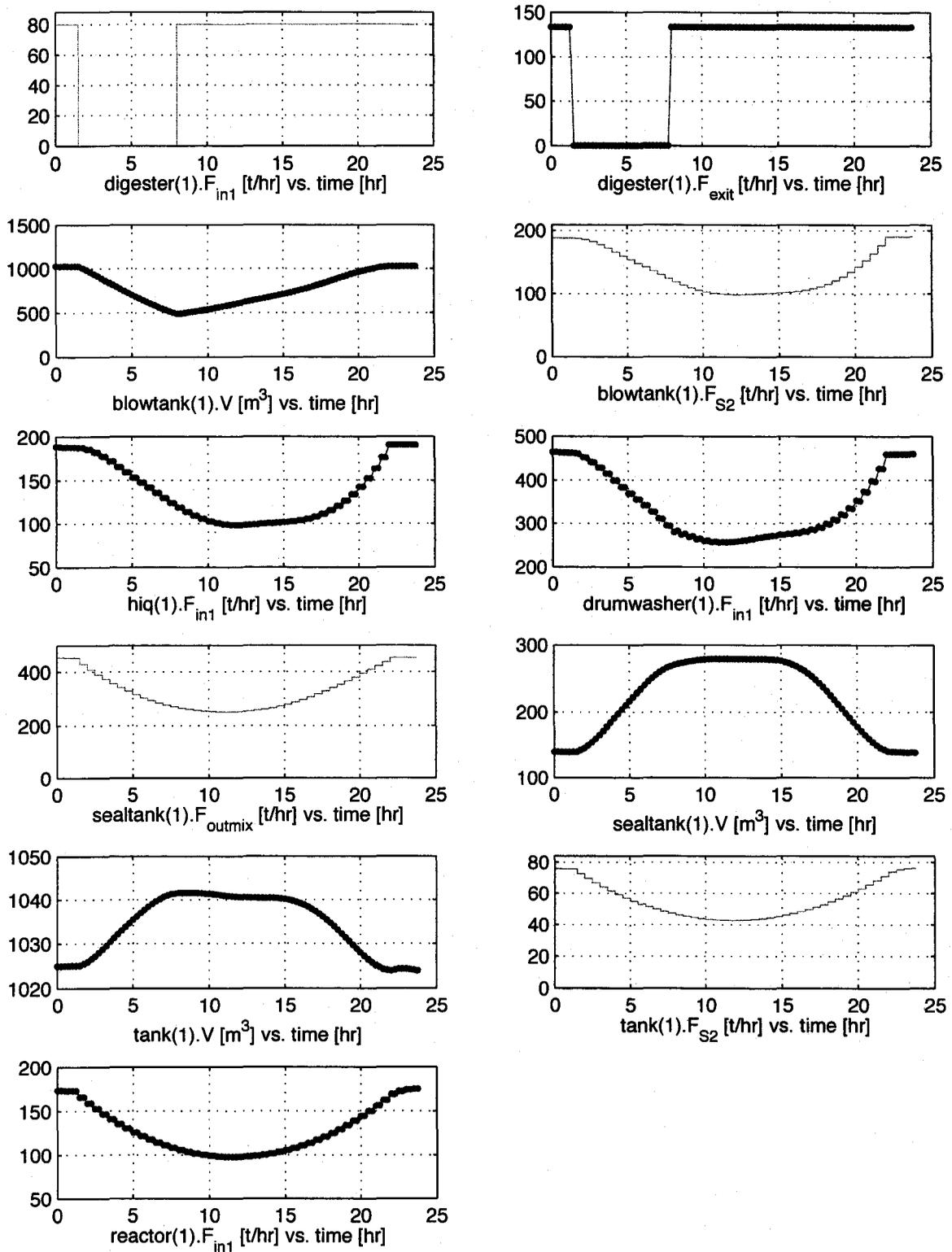


Figure 4.3: Failure in digester (digester(1).F_{in1} = 0 for t ∈ [2, 8]).

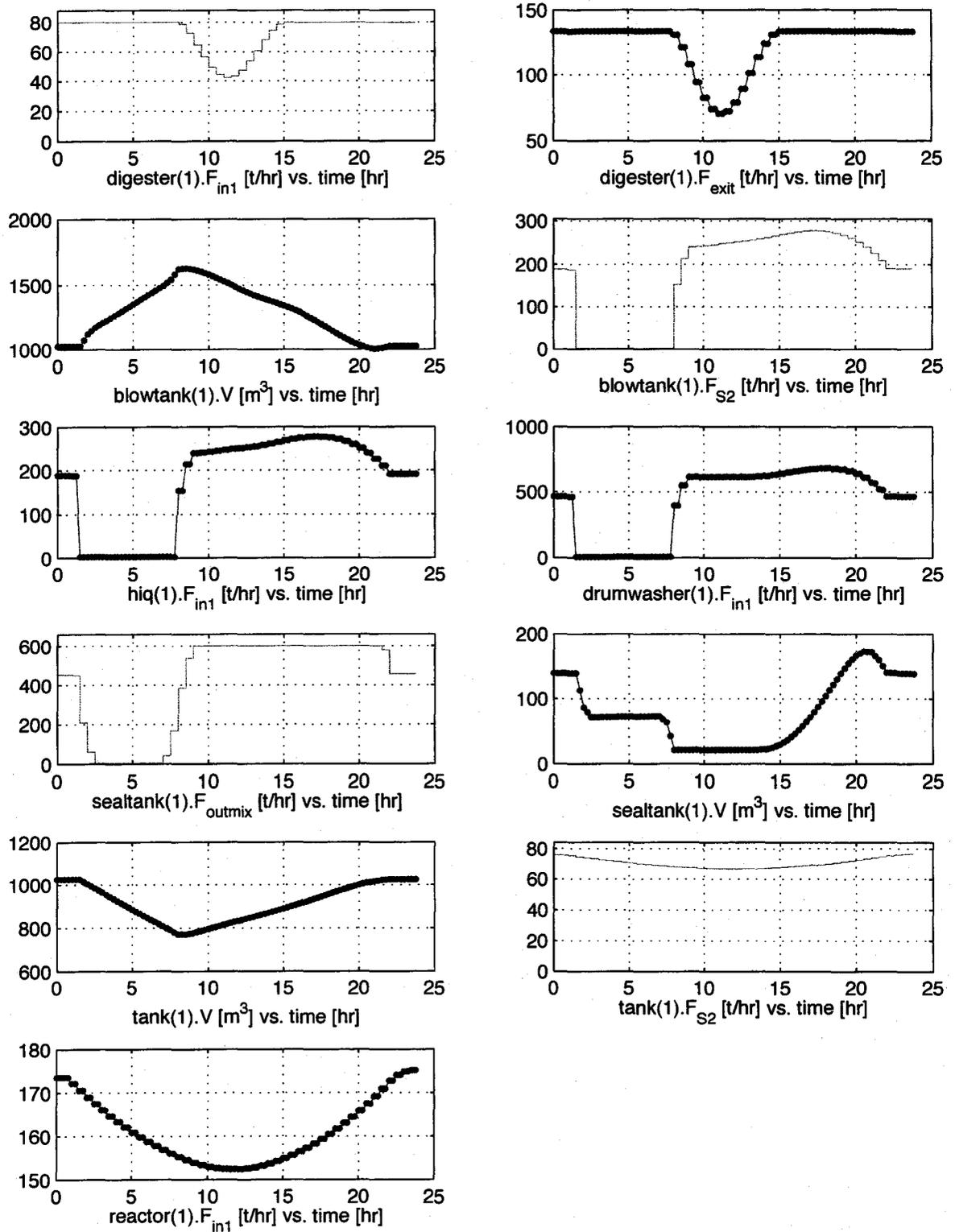


Figure 4.4: Failure in Hi-Q knotter ($hiq(1).F_{in1} = 0$ for $t \in [2, 8]$).

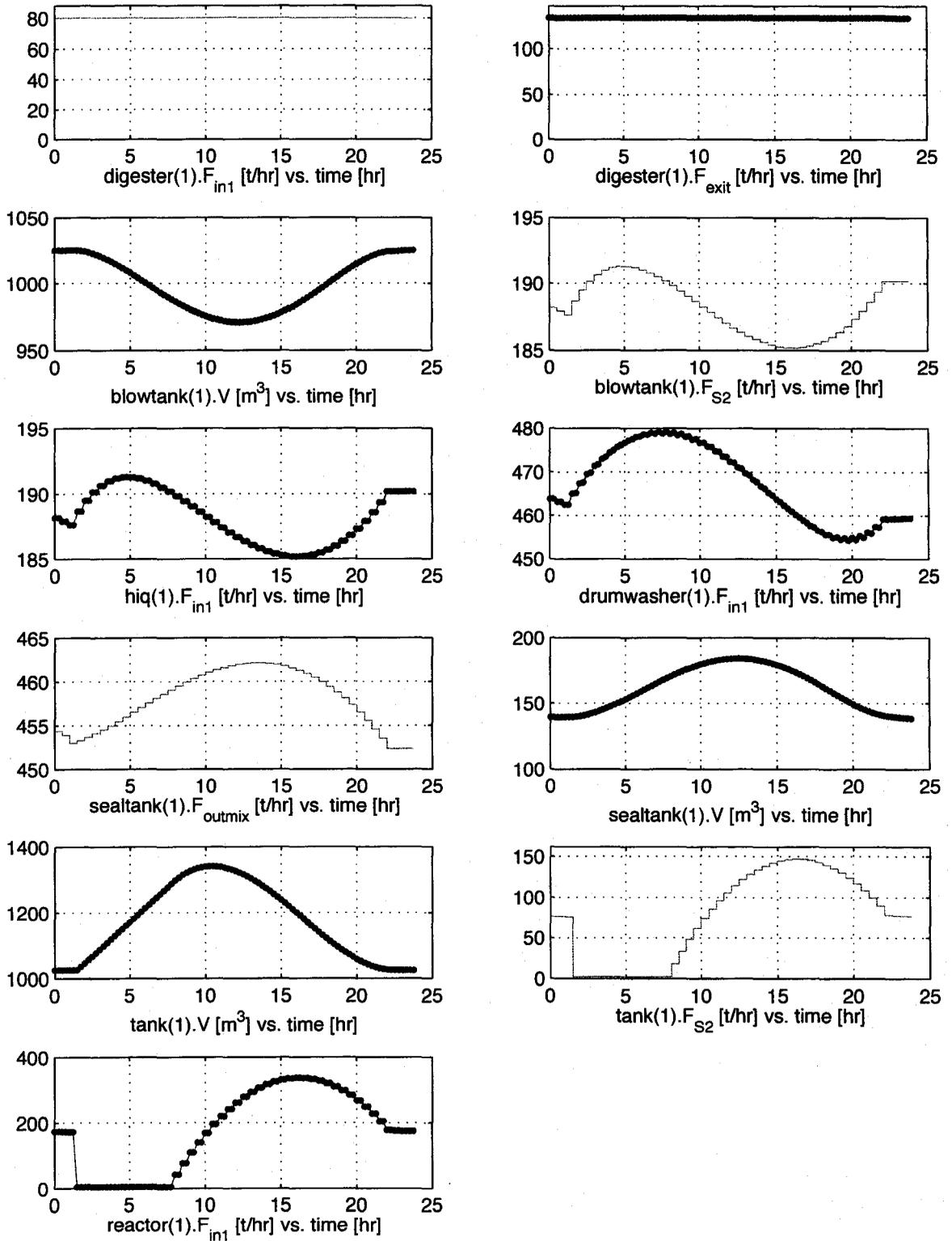


Figure 4.5: Failure in O₂-delignification reactor (reactor(1).F_{in1} = 0 for t ∈ [2, 8]).

4.4.1 Implementing the initial trajectory with naïve adjustments

One of the most obvious things that can be done to correct for an incorrect downtime estimate is to either shift the initial trajectory (the trajectory calculated for the initial d_{est}) backwards or forward. For instance, if at some point during the the shutdown period it is discovered that the actual downtime (d_{act}) is either longer or shorter than the initial downtime estimate d_{est} , the control system can take the following actions based on the situation:

1. **Case 1 ($d_{est} < d_{act}$) Shift trajectory backwards in time:** This is a case in which the estimated downtime is shorter than the actual (refer to Figure 4.6). The optimizer is instructed to hold the last input action (at t_{est}) until the system is ready to come online. This is equivalent to delaying and shifting the original trajectories backwards.
2. **Case 2 ($d_{est} > d_{act}$):** In this case, the estimated downtime is longer than the actual downtime.
 - (a) **Subcase 1: Implement original trajectory** (refer to Figure 4.7). The optimizer is instructed to continue implementing the original trajectories.
 - (b) **Subcase 2: Shift trajectory forward in time** (refer to Figure 4.8). The optimizer is instructed to shift the original trajectories forward.

This approach, while seemingly simple, is problematic. Simulations to test these cases were performed, where the Hi-Q knotter is shut down. The overall horizon used was 24 hours. The results are summarized in table below (Table 4.5):

Case	d_{est} (hrs)	d_{act} (hrs)	Results
1 (Shift Trajectory Backwards)	6	7.5	infeasible
2, subcase 1 (Implement original trajectory)	6	4.5	feasible, suboptimal
2, subcase 2 (Shift trajectory forward)	6	4.5	infeasible

Table 4.5: Simulation Results for Trajectory Adjustment Method

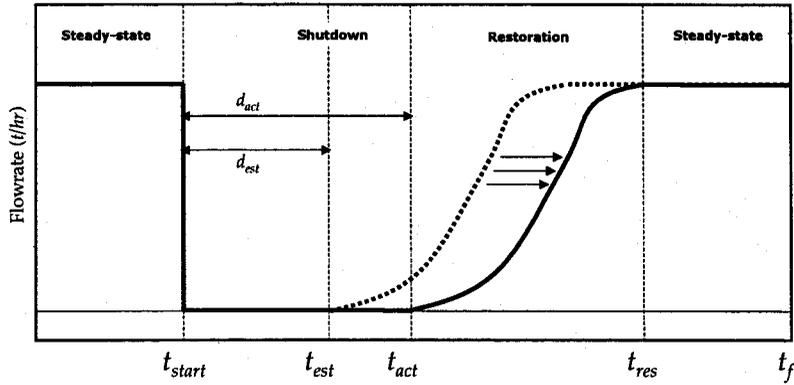


Figure 4.6: Case 1 ($d_{est} < d_{act}$).

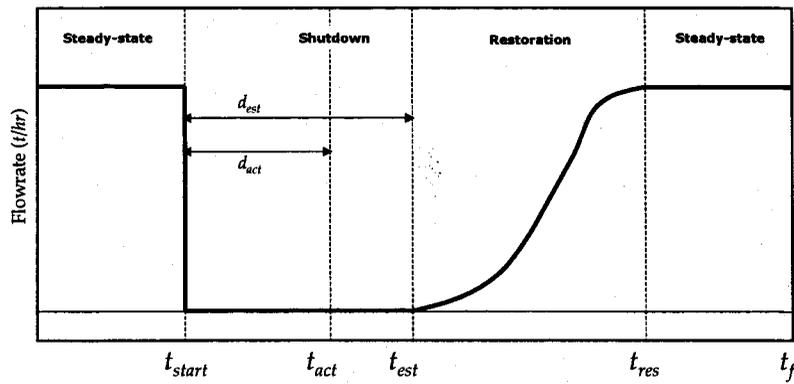


Figure 4.7: Case 2 ($d_{est} > d_{act}$), Subcase 1.

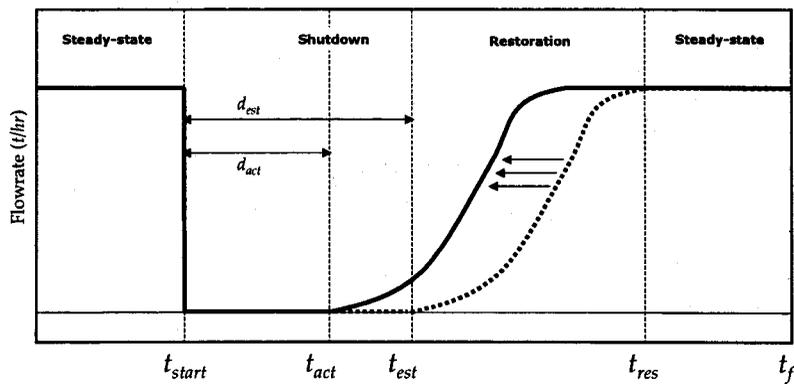


Figure 4.8: Case 2 ($d_{est} > d_{act}$), Subcase 2.

(Legend - Dotted line: originally calculated trajectory, Solid line: implemented trajectory)

In Case 1, the optimizer reported infeasibilities because despite the hold, there were inadequate degrees of freedom to satisfy other constraints (particularly the restoration constraints). This is largely due to the fact that when the trajectories were shifted, the system states after the shift were no longer the same as the original, therefore the control actions had become invalid.

In Case 2 (subcase 1), the optimizer returns a suboptimal result because the system is being shut-down longer than it has to be.

In Case 2 (subcase 2), the optimizer reported infeasibilities because the states at t_{act} are not the states at t_{est} (with which the initial trajectory was calculated), which led to constraints being violated.

It is clear that the solutions obtained with this method are impractical (due to their suboptimality and infeasibility), therefore we abandon this approach.

4.4.2 Optimization Under Uncertainty using a multi-scenario approach

In this approach, the uncertainty in the downtime estimate is accounted for from the onset. This requires two pieces of information: 1) the initial downtime estimate, d_{est} ; 2) uncertainty bounds on d_{est} , i.e. $d_{est}^{lb} \leq d_{est} \leq d_{est}^{ub}$.

The upper-bound and lower-bound information on d_{est} is used in a multi-scenario optimization formulation to come up with a control trajectory that is *optimal* for the nominal case, while simultaneously remaining *feasible* for any downtime duration between the given bounds. Thus, the optimizer will try to find a solution that will satisfy the worst case scenario. As a result, the resulting control trajectories are conservative in nature.

The problem is formulated by solving a set of parallel models, with the downtime estimate as the uncertain parameter. This may result in a very high dimensional DAE optimization problem, depending on the number of scenarios considered.

Objective Function (Nominal case)

$$\max_{\mathbf{u}(t)} \Phi(\mathbf{x}_{(0)}, \mathbf{z}_{(0)}(t), \mathbf{u}(t), t)$$

s.t.

Model Equations - Nominal Case

$$\mathbf{f}(\dot{\mathbf{x}}_{(0)}(t), \mathbf{x}_{(0)}(t), \mathbf{z}_{(0)}(t), \mathbf{u}(t), t) = 0 \quad (4.15)$$

$$\mathbf{g}(\mathbf{x}_{(0)}(t), \mathbf{z}_{(0)}(t), \mathbf{u}(t), t) \leq 0 \quad (4.16)$$

$$\mathbf{h}(\mathbf{x}_{(0)}(t), \mathbf{z}_{(0)}(t), \mathbf{u}(t), t) = 0 \quad (4.17)$$

Model Equations - Feasible Cases

$$\mathbf{f}(\dot{\mathbf{x}}_{(s)}(t), \mathbf{x}_{(s)}(t), \mathbf{z}_{(s)}(t), \mathbf{u}(t), t) = 0 \quad (4.18)$$

$$\mathbf{g}(\mathbf{x}_{(s)}(t), \mathbf{z}_{(s)}(t), \mathbf{u}(t), t) \leq 0 \quad (4.19)$$

$$\mathbf{h}(\mathbf{x}_{(s)}(t), \mathbf{z}_{(s)}(t), \mathbf{u}(t), t) = 0 \quad \forall s \in \{1, 2\} \quad (4.20)$$

Shutdown constraints

$$\mathbf{F}_{(0),in,unit}(t) = 0 \quad \text{for } t_{down} \leq t \leq t_{start} + d_{est} \quad (4.21)$$

$$\mathbf{F}_{(1),in,unit}(t) = 0 \quad \text{for } t_{down} \leq t \leq t_{start} + d_{est}^{lb} \quad (4.22)$$

$$\mathbf{F}_{(2),in,unit}(t) = 0 \quad \text{for } t_{down} \leq t \leq t_{start} + d_{est}^{ub} \quad (4.23)$$

where

s = scenarios

subscripted (0) = the nominal scenario, i.e. where the downtime estimate is d_{est}

It is found that when there is only one uncertain parameter (*i.e.* d_{est} in our case) and the trajectories are directly dependent on this parameter, the solution will correspond to that obtained when $d_{est} = d_{est}^{ub}$ (*i.e.* the worst case, which encompasses all cases). This is confirmed by the results of our simulations below where the trajectories obtained using the multi-scenario are identical to those obtained by assigning $d_{est} = d_{est}^{ub}$ (worst case). It follows that the more conservative the upper bound is, the more suboptimal the result will be, because the optimizer has to maintain feasibility for a larger region.

It has to be mentioned that in general, the solution of a multiscenario optimization problem does not necessarily correspond to picking the worst case in each uncertain variable. In a 1983 paper, Grossman and Morari [38] demonstrated in their study of a network of heat exchangers that oversizing every heat exchanger (the most conservative case) does not lead to optimal flexibility.

Scenario

The Hi-Q knotter is shut down. The actual duration of the shutdown is $d_{act} = 6$

hours. The estimated duration is in the range $d_{est}^{lb} \leq d_{est} \leq d_{est}^{ub}$, where $d_{est}^{lb} = 4.5$ hours and $d_{est}^{ub} = 7.5$ hours. The horizon is 24 hours. The goal is to solve the problem to obtain a set of trajectories that is feasible for any downtime estimate that lies within the range $4.5 \leq d_{est} \leq 7.5$.

Results

The resulting trajectories can be seen in Figure 4.9. From Table 4.6, it is observed

Case	Profit (\$)
Ideal (d_{act} is known exactly)	102,106
Multiscenario optimization	93,021
Optimization assuming worst case ($d_{est} = d_{est}^{ub}$)	93,021

Table 4.6: Comparison of Profits from Multiscenario Optimization and Others

that the solution from the multiscenario optimization produces a 24-hour profit that is lower than if the operator had perfect knowledge of the actual duration of the shutdown. The lower profit seen in the multiscenario optimization (Table 4.6) is due to fact that in order to be feasible for a longer downtime, the Hi-Q unit is forced to shut down for a longer period ($hiq(1).F_{in1}$), which lowers overall production. This also induces a shutdown in the digester (as seen in $digester.F_{in1}$), which further lowers the throughput of the system in order to meet constraints. The lower throughput is most clearly reflected in the lowered feed rate to the O₂ delignification reactor ($reactor(1).F_{in1}$).

Conclusion

This approach yields conservative trajectories that are suboptimal but feasible within the given bounds. The advantage of this method is that only a one-time calculation is required.

4.4.3 Trajectory Re-optimization using Feedback of Downtime Estimate

This approach allows the operator to revise any downtime estimates dynamically. At any time during the shutdown, the operator can enter corrected downtime estimates as information arrives and the remainder of the trajectory is re-optimized from the current state of the system, and the controller performs what is essentially a mid-

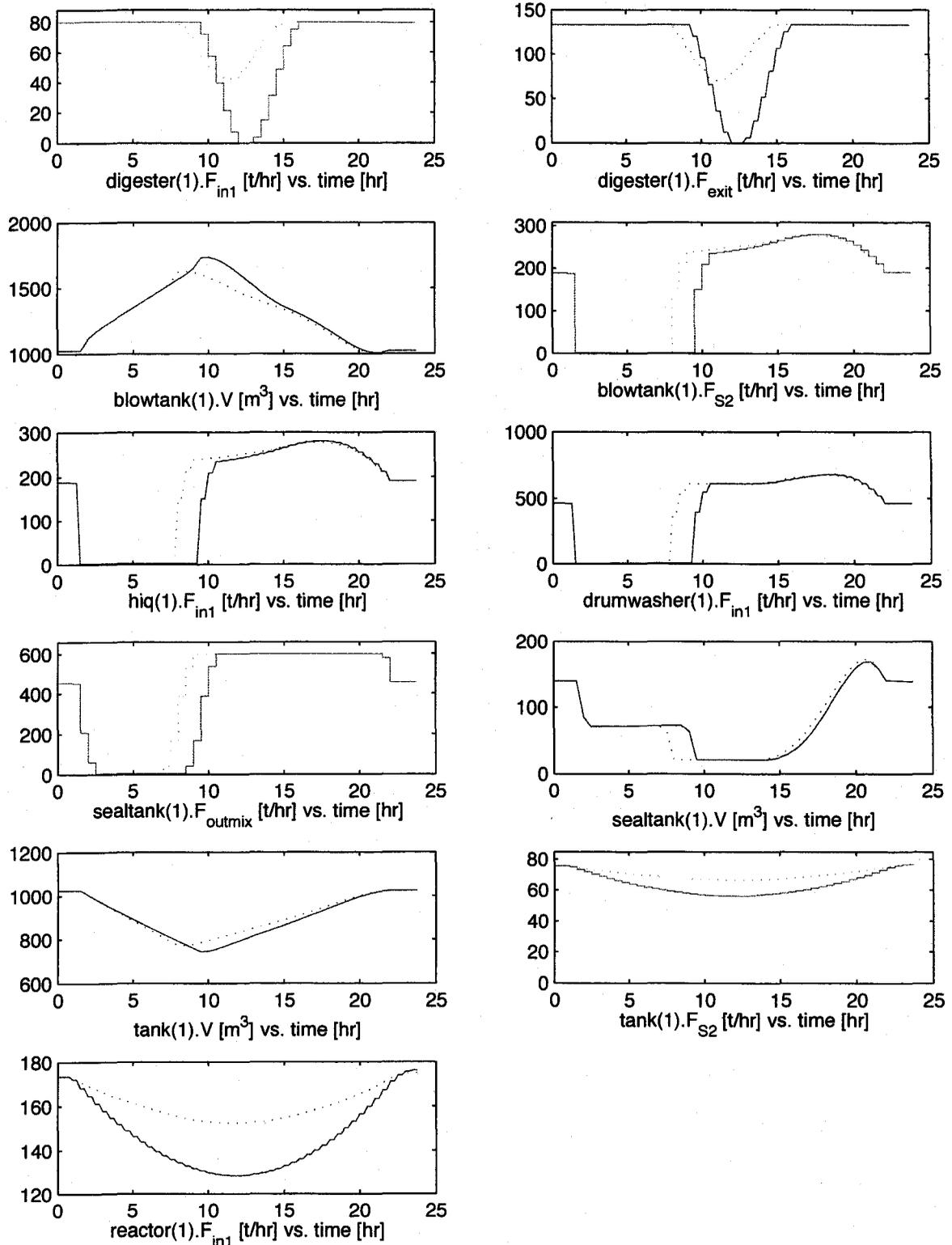


Figure 4.9: Comparison between ideal trajectories (dotted lines) and trajectories obtained from multiscenario optimization (solid lines).

course correction.

The effectiveness of active reoptimization depends on:

1. the degrees-of-freedom available to the system (amount of buffering capacity that is available, for instance)
2. how early the reoptimization is done
3. where the failing unit is situated with respect to buffers.

This feedback approach has considerable advantages over a multi-scenario optimization approach for dealing with uncertainty in the estimated downtime, in that the resulting control trajectories are less conservative. The performance of this re-optimization scheme is studied in this work under various failure scenarios.

Case studies: Trajectory Re-optimization - Effect of a one-time reoptimization at various stages

This case study considers the effect of early reoptimization on the optimal throughput. The failure occurs in the Hi-Q unit, between two buffer tanks. A 24 hour horizon was used for the simulations. In each case, the reoptimization is performed *once and only once*.

Case 1: Actual downtime is longer than estimated downtime, Figure 4.10

($d_{act} > d_{est}$, where $d_{act} = 10$ hrs and $d_{est} = 8$ hrs)

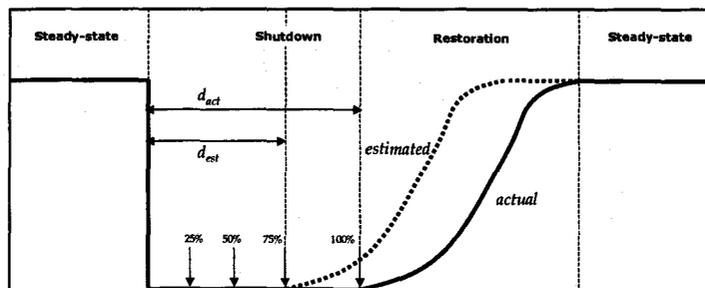


Figure 4.10: Case 1: Reoptimization at various stages, given that $d_{act} > d_{est}$.

1. Case 1.1: Re-optimize 25% into the estimated shutdown duration (Figure 4.12)
2. Case 1.2: Re-optimize 50% into the estimated original shutdown duration
3. Case 1.3: Re-optimize 75% into the estimated shutdown duration
4. Case 1.4: Re-optimize 100% into the estimated shutdown duration (Figure 4.13)
5. Case 1.5: Ideal trajectory (assuming d_{act} was known from the onset)

Case 2: Estimated downtime is longer than actual downtime, Figure 4.11

($d_{est} > d_{act}$, where $d_{act} = 6$ hrs and $d_{est} = 8$ hrs)

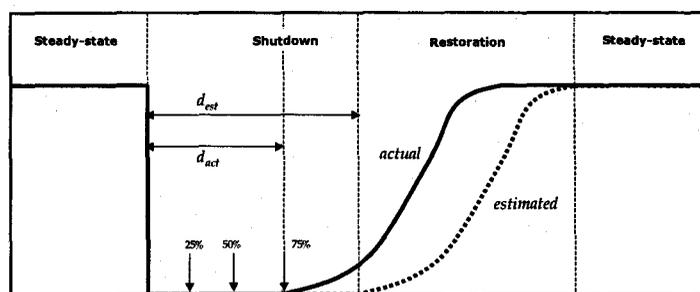


Figure 4.11: Case 2: Reoptimization at various stages, given that $d_{est} > d_{act}$.

1. Case 2.1: Re-optimize 25% into the estimated shutdown duration (Figure 4.14)
2. Case 2.2: Re-optimize 50% into the estimated original shutdown duration
3. Case 2.3: Re-optimize 75% into the estimated shutdown duration (Figure 4.15)
4. Case 2.4: Ideal trajectory (assuming d_{act} was known from the onset)

Table 4.7 below summarizes the results:

Trajectory Re-optimization: Findings

The general trend that can be gleaned from the data in Table 4.7 is that the later a reoptimization is performed, the poorer the result will be. The later a reoptimization is performed, the fewer the degrees of freedom that is available to the optimizer.

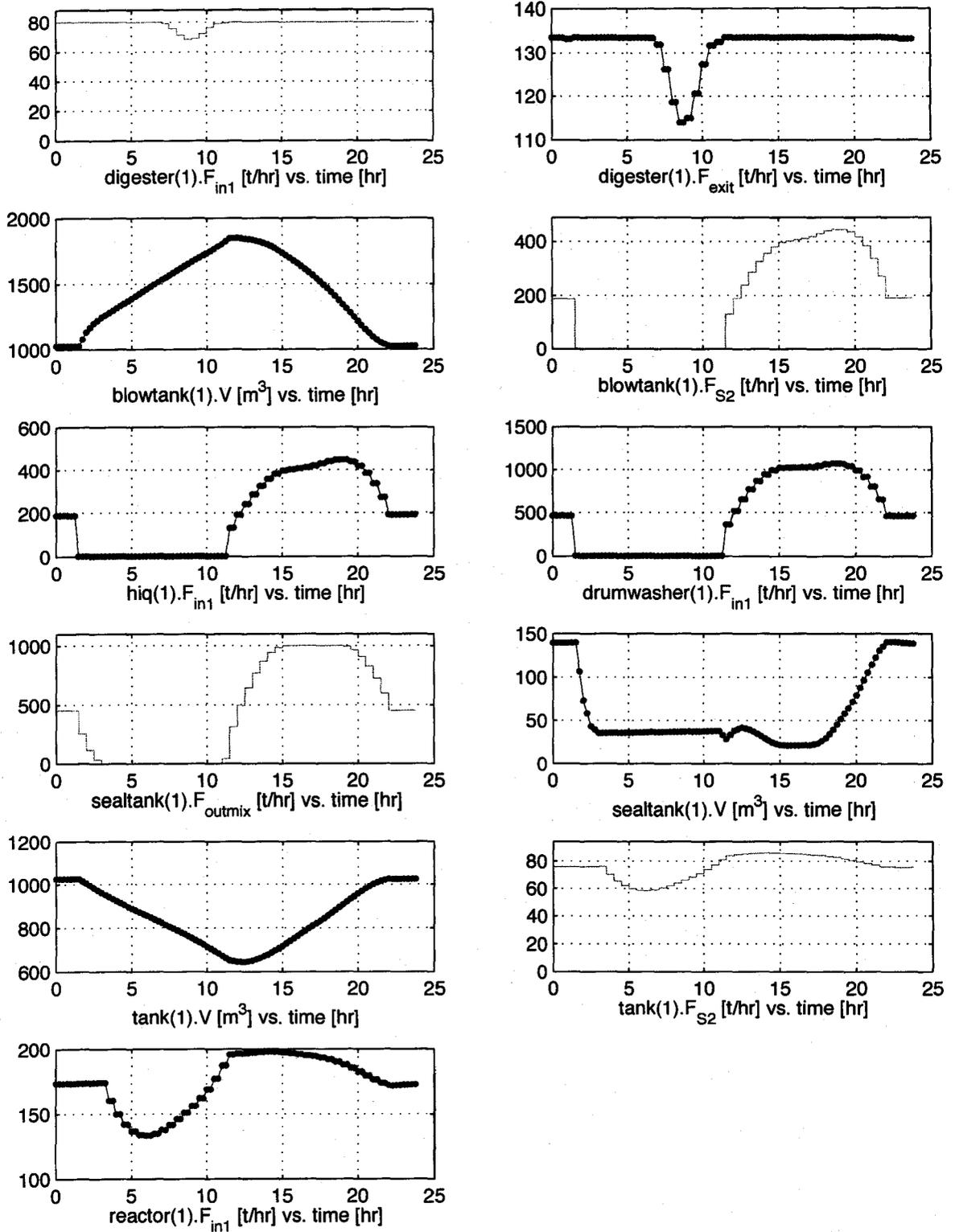


Figure 4.12: Trajectories for reoptimization 25% into estimated shutdown, where $d_{act} > d_{est}$.

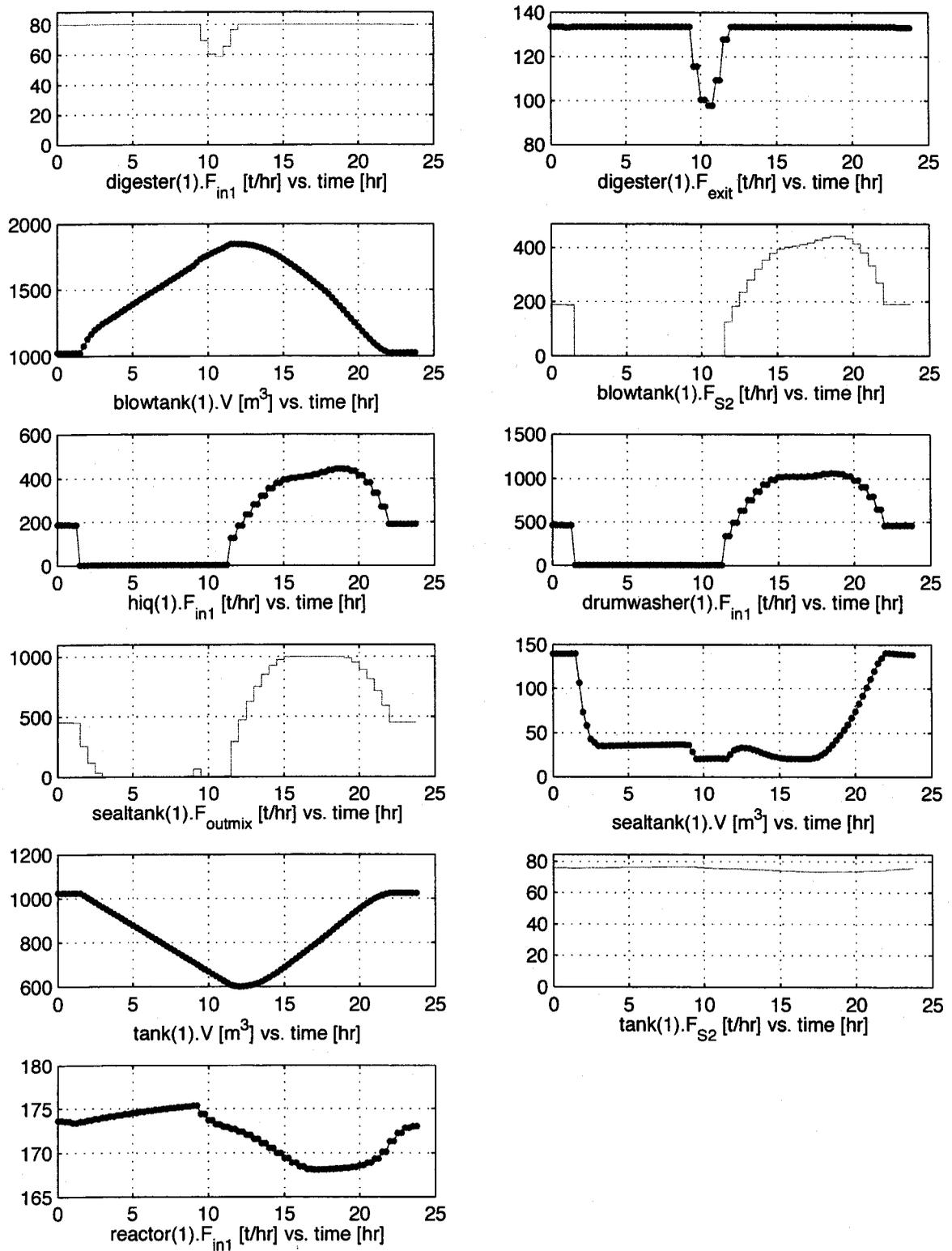


Figure 4.13: Trajectories for reoptimization 100% into estimated shutdown, where $d_{act} > d_{est}$.

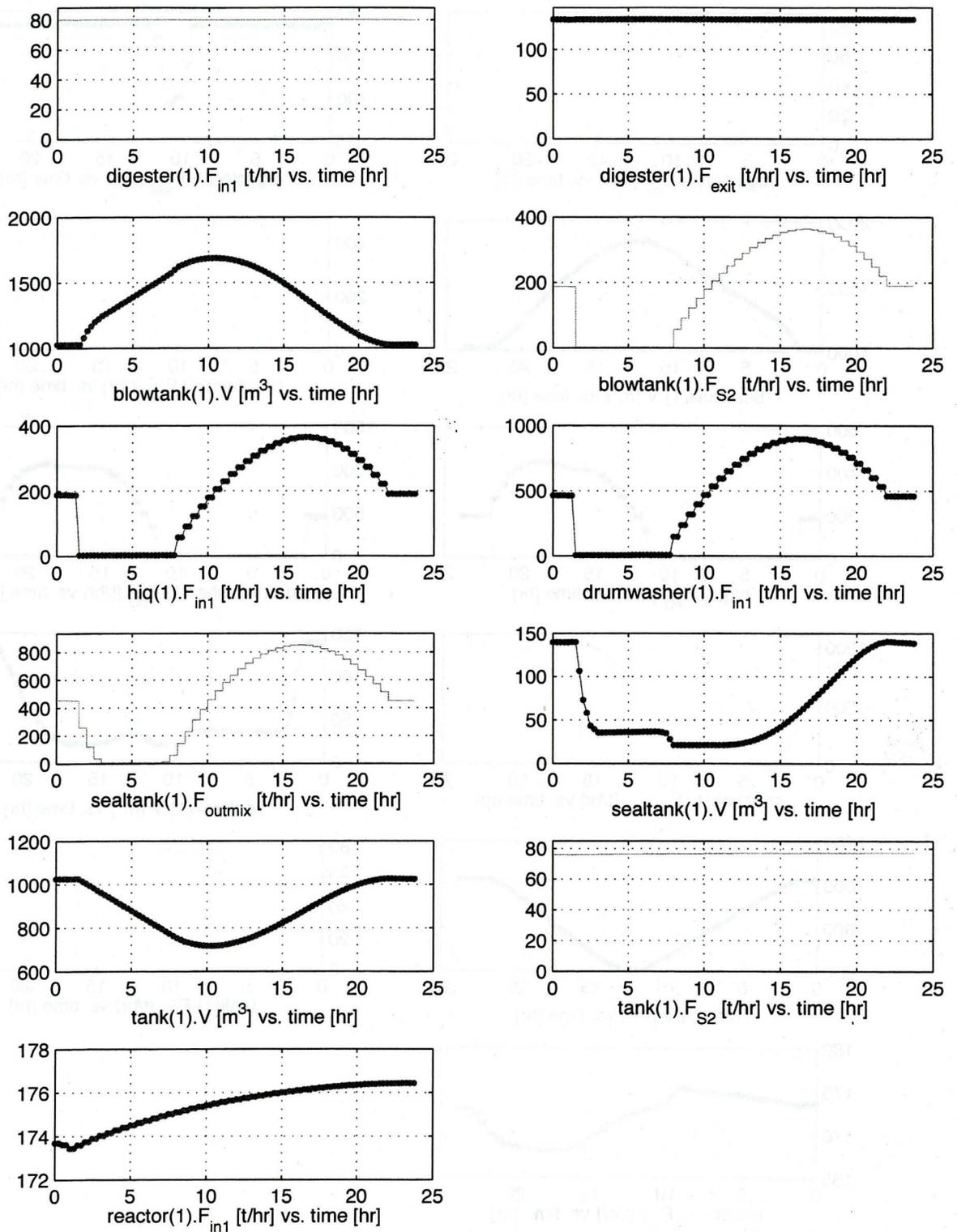


Figure 4.14: Trajectories for reoptimization 25% into estimated shutdown, where $d_{est} > d_{act}$.

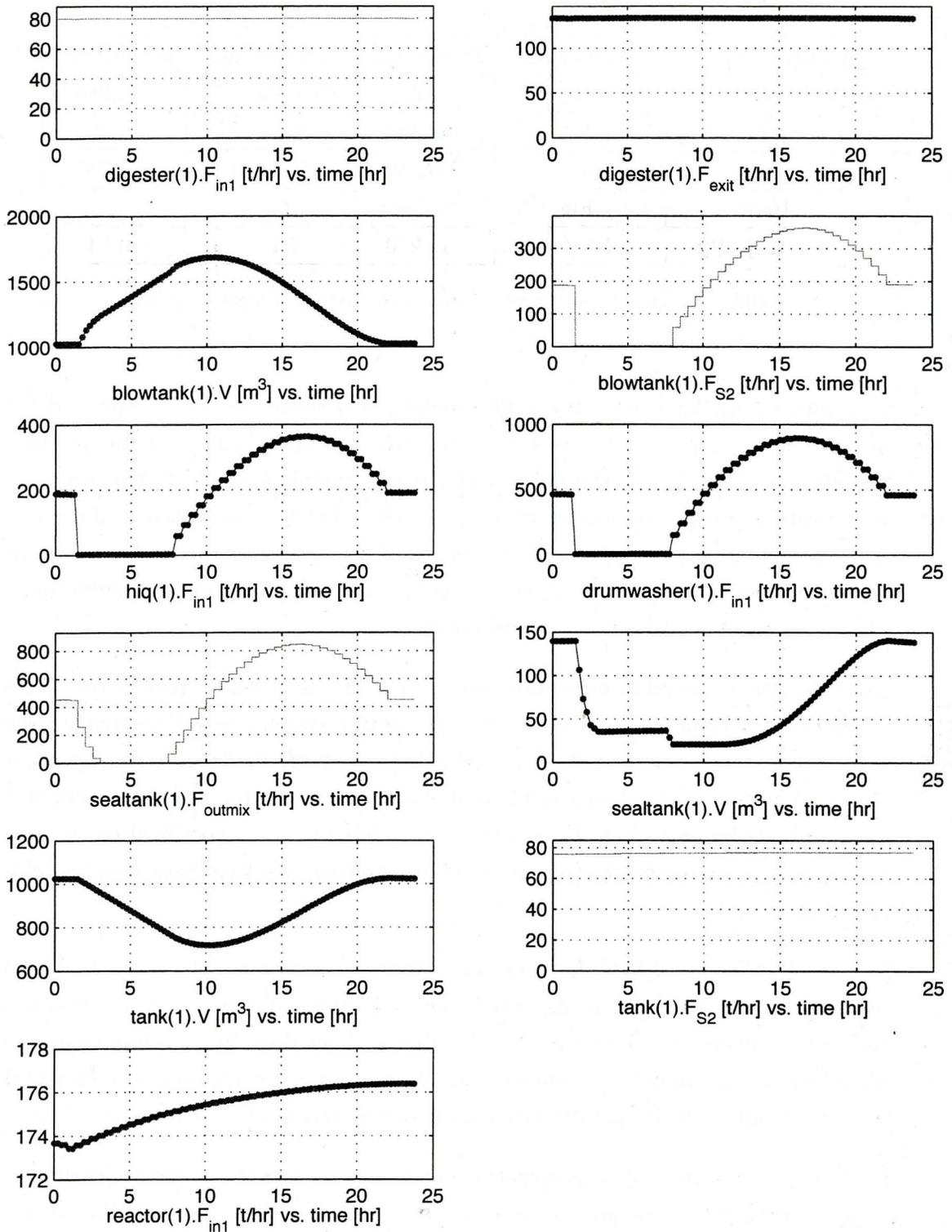


Figure 4.15: Trajectories for reoptimization 75% into estimated shutdown, where $d_{est} > d_{act}$.

Reoptimize at	$d_{act} > d_{est}$		$d_{est} > d_{act}$	
	Case/Fig.	Profit (\$)	Case/Fig.	Profit (\$)
25%	1.1 / Fig 4.12	110,358	2.1 / Fig 4.14	111,288
50%	1.2 / -	109,760	2.2 / -	111,287
75%	1.3 / -	109,752	2.3 / Fig 4.15	111,287
100%	1.4 / Fig 4.13	92,164	- / -	-
Ideal trajectory	1.5 / -	111,250	2.4 / -	111,312

Table 4.7: One-time Reoptimization at Various Stages into d_{est}

Generally, except for Case 1.4, the degradation of performance with respect to the lateness of the reoptimization was not large, which is an indication of the system's flexibility in responding to shutdowns. This is attributable to the fact that inventory movements are largely reversible actions (through either the manipulation of recycles or by manipulating production rates). The optimizer recognizes this and exploits the degrees of freedom available to drive the system to optimality, even after suboptimal control actions steps have been implemented.

As a side note, it should be noted that reoptimization cannot always reverse the effects of erroneous control. This is often true for reacting systems with irreversible reactions. For instance, if a certain set of control actions (performed before the downtime feedback) gives rise to a change in the feed of a reactor and if the reaction proceeds in an undesired direction, then the product is considered to be off-specification. In such instances, it is impossible for the control system to steer a system back to a desirable operating point.

In Case 1.1 (Figure 4.12), the optimizer receives information (at $t=3.5$ hrs) that the shutdown is longer than expected, and thus the build-up in the blowtank is higher than expected. Therefore it prescribes a slight drop in the digester output (digester(1). F_{exit}) in order to prevent the blowtank from overflowing (the blowtank level, blowtank(1). V , is operating close to its upper bound of 1845 m³ at $t=12$ hrs).

In Case 1.4, the effect of reoptimization (performed at $t=8$ hrs) is most clearly seen in the material flowrates into the delignification reactor. If the reactor(1). F_{in1} trajectories in Figures 4.12 and 4.13 were to be compared, one notices that in the former case, the material flowrate into the reactor drops significantly for a period but makes a rapid ascent before returning to the original steady-state. In the latter case (when

reoptimization is performed later, at $t=8\text{hrs}$), the input into the reactor stays within a small band ($\pm 10\text{ t/hr}$), and the $\text{tank}(1).V$ level is allowed to drop a little lower than in Fig 4.12. Another indication of the reoptimization taking place can be seen in the $\text{sealtank}(1).F_{outmix}$ trajectory. At $t=8\text{hrs}$, sealtank begins to discharge material ($\text{sealtank}(1).F_{outmix}$) again after shutting down, but when given the information that the downtime is longer than originally predicted, the control system shuts off the flow once more and keeps the sealtank discharge at zero until the end of the actual downtime. The profit obtained, \$92,164, (Table 4.7) is significantly below the ideal profit. We deduce that in this case, the system has moved past a certain threshold and is unable to successfully reverse most of the effects of past control, therefore the profit suffers.

In Case 2.1 (Figure 4.14) and Case 2.3 (Figure 4.15), the trajectories are almost identical. In these cases, the estimated downtime is longer than the actual (i.e. the reality is less severe than projected). This means that the original policy was a conservative one, and when reoptimization is performed, the extra degrees of freedom gained are used to effect a less conservative solution. As such, reoptimization at various stages does not have as big an effect as in the case where the estimated downtime is shorter than the actual. The profits in Table 4.7 show very little profit degradation at various stages of reoptimization. However, the degree to which reoptimization may benefit the operation would be difficult to ascertain *a priori* without rigorous optimization.

Simple Comparison of Reoptimization and Multiscenario Optimization

Consider a scenario in which the digester shuts down. The estimated failure time is $d_{est} = 8$ hours, with uncertainty being ± 2 hours. The actual downtime is $d_{act} = 9$ hours. The operator is given the actual downtime information 1 hour into the shutdown, and the system reoptimizes based on the new information.

Results

From Table 4.8, reoptimization emerges as the best-performing method for handling the uncertainty in downtime estimates by a considerable margin. The control policy from multiscenario optimization is necessarily conservative for all $d_{act} < 10 = d_{est}^{ub}$ (and optimal for $d_{act} = 10$). Reoptimization, on the other hand, attempts to arrive at the optimal trajectory solution using the best available information at a given time

Method	Profit (\$)
Reoptimization	66,485
Multiscenario Optimization	61,783
Ideal Scenario	66,506

Table 4.8: Comparison of Reoptimization and Multiscenario Optimization

and thus usually delivers good performance.

4.5 Computational Issue: Avoiding Control Policies that Induce Shutdowns

The optimizer will occasionally yield an optimal policy that prescribes shutting off material flows to certain units (essentially shutting them down) to accommodate the original shutdown. Henceforth we shall denote this type of shutdown a “induced shutdown”.

Induced shutdowns usually occur in response to long shutdown durations. A single buffer tank has the capacity to buffer shutdowns for a limited time before it overflows or empties. Therefore, when a long shutdown occurs, it is sometimes necessary to throttle down the production upstream in order to keep from violating the level constraints in the buffer capacities.

However, instead of reducing production upstream over a long horizon, the optimizer may find it more profitable to completely shut down upstream production for a short period of time.

Induced shutdowns are undesirable because each shutdown incurs both a fixed cost (in terms of the manpower, time and resources required to start a unit up again) and variable cost (in the form of lost production over time). In order to force the optimizer to avoid this scenario, it is necessary to penalize induced shutdowns in the objective function. In this section, we investigate the use of a barrier function to penalize shutdowns.

4.5.1 Barrier method

In this approach, we formulate a penalty that activates as soon as unit outlet flowrates, F_i reach their shutdown flowrate, $F_{i,shut}$ (a flowrate at which a unit is said to have shut down).

$$\max_{\mathbf{F}(t)} [\Phi_{economics} - \varphi]$$

s.t.

$$\varphi = \sum_{i=1}^{N_{units}} \left[\int_0^{t_f} \omega_i(t) dt \right] \quad (4.24)$$

$$\omega_i(t) = \begin{cases} A_i, & \text{for } F_i(t) = F_{i,shut} \\ 0, & \text{for } F_i(t) > F_{i,shut} \end{cases} \quad (4.25)$$

$$F_i(t) \geq F_{i,shut} \quad (4.27)$$

for $i = 1..N_{units}$

where

i = index denoting the i -th process unit in the plant

t_f = final time, end of prediction horizon

A_i = barrier weight for flowrate in process unit i , an indicator of the relative severity of the shutdown in unit i with respect to other units

$F_i(t)$ = flowrate i , through a process unit

$F_{i,shut}$ = shutdown threshold flowrate i , a flowrate value at which a shutdown is forced. $\omega_i(t)$ = barrier function

N_{units} = number of process units with manipulated variables

In this formulation, whenever $F_i(t) = F_{i,shut}$, $\omega_i(t)$ activates and takes on a value of A_i . The integral of $\omega_i(t)$ over the time horizon will yield a penalty representing the shutdown length multiplied by the barrier weight. However, the function $\omega_i(t)$ is discontinuous, therefore in order to solve it as a continuous optimization problem, we propose the following barrier function approximation.

Use of an Exponential Barrier Function to Approximate Switching

Barrier functions are often used in interior-point algorithms to steer the algorithm away from constraints. They do this by causing the objective to degrade dramatically

if the current iterate moves towards a constraint boundary. Here, we will attempt to use this concept to formulate a penalty which will activate if the optimizer induces a shutdown. The following is a function that tends to A_i when F_i is close to $F_{i,shut}$, and closely approximates the behavior of equations (4.25 - 4.26).

$$\omega_i(t) = A_i \exp[-k(F_i(t) - F_{i,shut})] \quad \forall i \in \{1..N_{units}\} \quad (4.28)$$

where

k = tuning parameter for sharpness of switching interval (positive number)

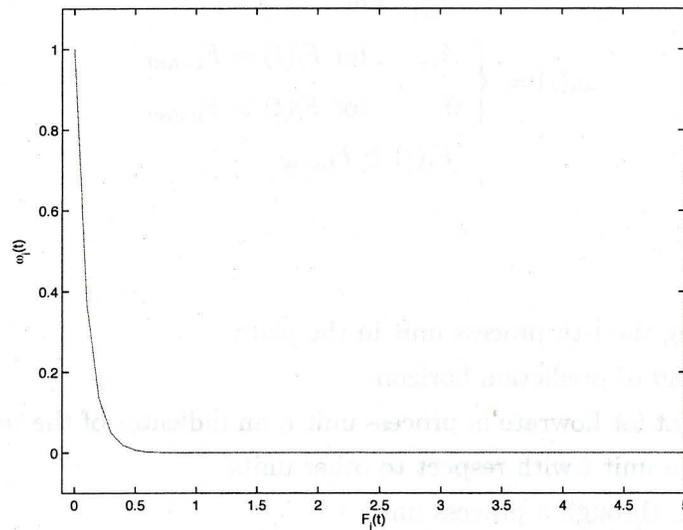


Figure 4.16: Exponential barrier function, with $A = 1$, $k = 10$ and $F_{shut} = 0$.

The main advantages of this function are:

1. Continuous, differentiable everywhere
2. Provides good approximation of switching.
3. All terms are bounded within the domain of interest

The disadvantage of this function is that it:

1. Requires trial-and-error to fix the tuning parameter

2. Is a purely numerical construct, and does not accurately represent the economic cost of failure. The penalty term amounts to the product of the barrier weight and the total shutdown length over all process units, which is not a true representation of the actual cost of shutdown.

4.5.2 Case Study: Penalizing Induced Shutdowns

Description

In this case study, we investigate the effect of penalizing induced shutdowns. The drum washer is shut down for a duration of 8 hours. The fixed cost associated with this shutdown is \$1000. The fixed cost associated with a digester cleanout is \$3000. A nominal trajectory is obtained by solving the problem. The shutdown of the digester upstream is triggered. The problem is then solved again with the Barrier method, with weights $A_{\text{digester}}=3000$ and $A_i=1000$ where i represents units other than the digester. It is important to recognize that the barrier weights merely reflect the relative severity of a shutdown, and is not representative of the true economic cost. The flowrate corresponding to a shutdown is $F_{i,\text{shut}} = 0$ in all cases.

Results

The results are summarized in Table 4.9, with references to the figures containing corresponding trajectories. “Apparent Profit” refers to the economic objective function in the optimization problem. “Adjusted Profit” refers to the true profit after accounting for the fixed cost of shutdowns.

	Fig.	Apparent Profit (\$)	Shutdown Costs (\$)	Adjusted Profit (\$)
No penalty	4.17	89,990	4000	85,990
Barrier	4.18	89,938	1000	88,938

Table 4.9: Penalty Method for Induced Shutdowns

The total fixed cost of shutdowns in the first case is \$4,000 (\$3,000 due to the induced shutdown in the digester [$\text{digester}(1).F_{\text{exit}}$], \$1,000 due to the shutdown in the vacuum drum washer [$\text{drumwasher}(1).F_{\text{in1}}$], refer to Figure 4.17). The digester production ($\text{digester}(1).F_{\text{exit}}$) was shut down temporarily in order to prevent the blowtank (which was operating close to its upper bound) from overflowing.

The total fixed cost of shutdowns in the second case (Barrier) is \$1,000 (due to the shutdown in the vacuum drum washer [$\text{drumwasher}(1).F_{in1}$], Figure 4.18). The digester production ($\text{digester}(1).F_{exit}$) was reduced but stopped short of shutting down.

The reader's attention is drawn to the fact that the apparent profit (which is being maximized in the NLP) when the induced shutdown is allowed to occur is higher than if the digester were kept running. This suggests that the induced shutdown leads to a higher profit. However, the apparent profit disregards the fact that there is a cost (\$3000) incurred every time the digester is shut down. When one factors in the cost of shutting down the digester (adjusted profit), the solution with the barrier function applied is the actual optimal solution. In general, induced shutdowns tend to be highly undesirable and should be penalized in almost all instances.

The question that will inevitably arise is why the true shutdown costs were not taken into account in the objective function in the first place. While the barrier method provides a means of guiding the optimizer away from an induced shutdown, it is merely a numerical construct and the barrier term is not a representation of the true cost of a shutdown. The true cost is represented by two components: fixed and variable. The variable cost associated with a shutdown is typically in the form of lost production—this cost is captured implicitly in the NLP. The fixed cost of a shutdown, however, comprises costs that are incurred *per shutdown*, regardless of length. Examples include manpower costs, use of cleaning chemicals, and so on. In order to account for this, we require a way to count the number of shutdowns that occur ($N_{shutdown}$) and multiply it with its fixed costs. However, since induced shutdowns are the result of the solution of an NLP and we have no way of knowing the total number of induced shutdowns a priori, it stands to reason that the counting of $N_{shutdown}$ must be done within the NLP itself.

It is possible to obtain $N_{shutdown}$ within an NLP formulation (refer to Appendix B for a derivation of a method for counting shutdowns). Unfortunately, the resulting formulation is highly nonlinear. While this formulation is suitable for moderate-sized problems, it produces difficulties for large-scale ones due to its use of hyperbolic continuous switching functions. It is possible to replace the continuous switching functions with integer variables, with the consequence that the resulting problem is a mixed-integer nonlinear program, which will likely be intractable for large-scale systems.

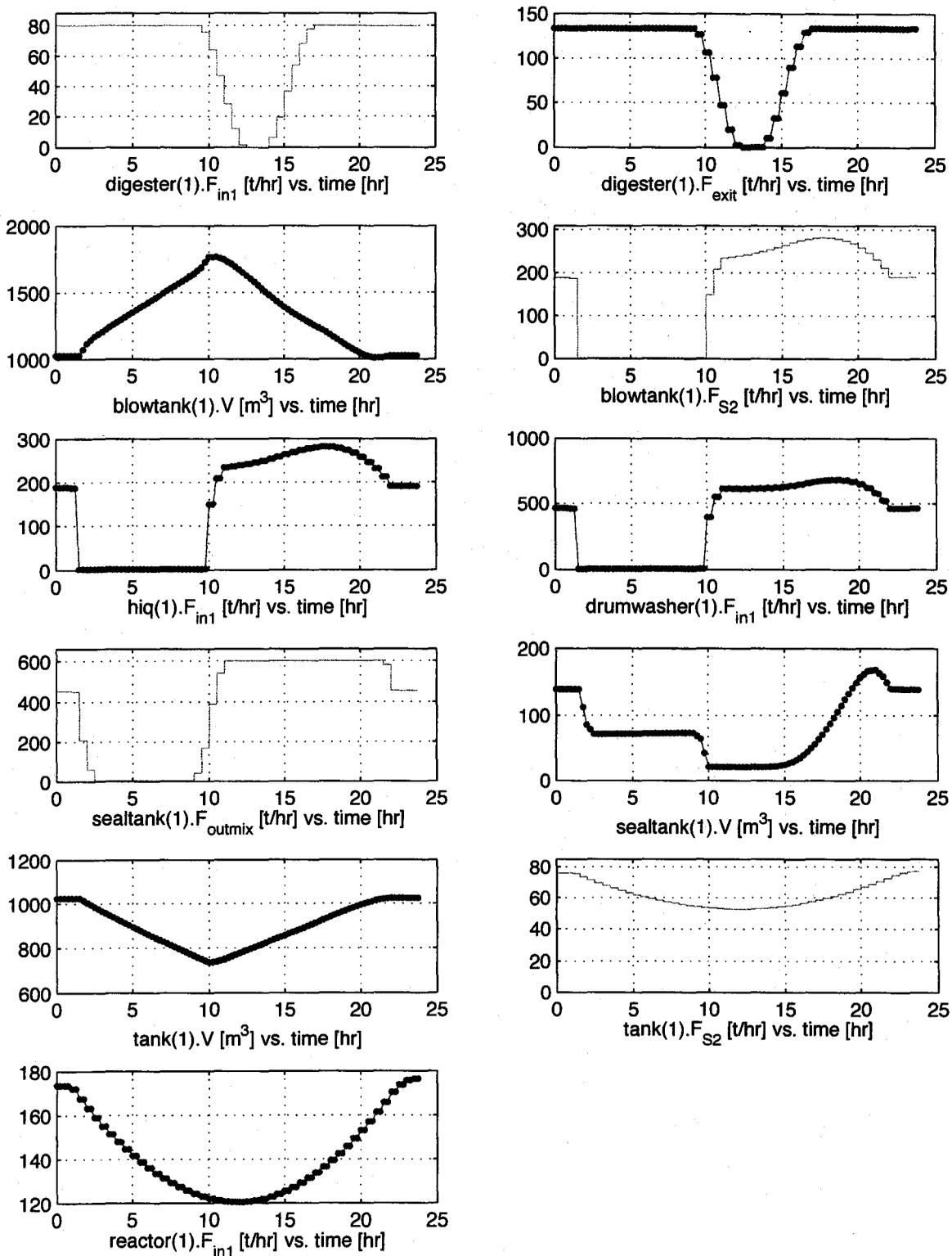


Figure 4.17: No penalty method applied to avoid induced shutdowns.

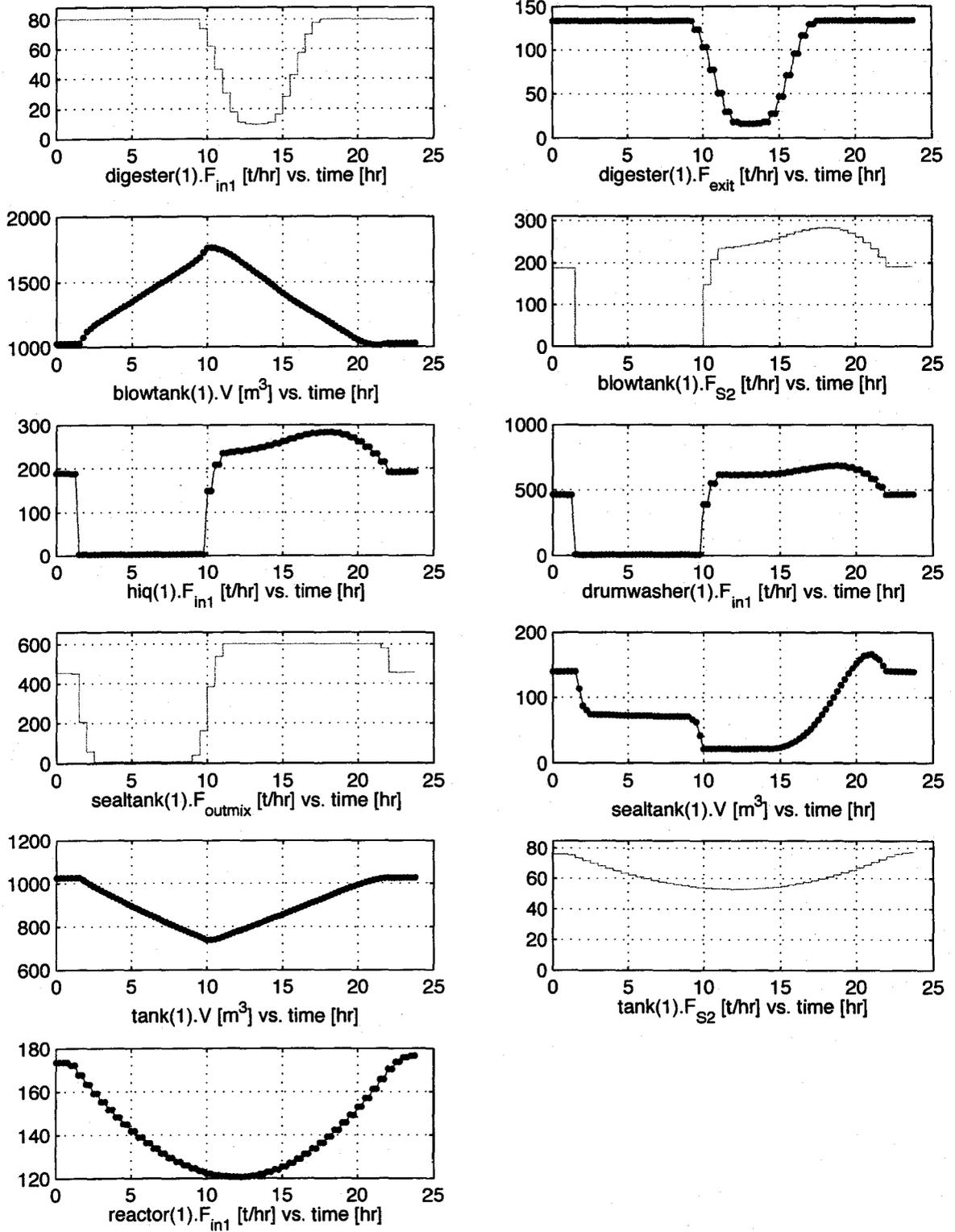


Figure 4.18: Exponential penalty applied to avoid induced shutdowns.

4.6 Chapter Summary

In this chapter, we looked into the use of dynamic optimization for generating policies for operating a plant under unit shutdown conditions. The location of the shut down unit relative to the buffers was studied. Various ways of handling the uncertain downtime estimate parameter were presented, and re-optimization based on operator feedback of downtime estimate was found to be the most effective. A method for addressing the problem of induced shutdowns was presented.

Chapter 5

Integrated Predictive Control Framework for Shutdowns

Life can only be understood backwards, but it must be lived forwards.

- Søren Kierkegaard, existentialist philosopher

In the previous chapter, the idea of using dynamic optimization for obtaining optimal operating policies for shutdown scenarios was put forward. In this chapter, we will be extending the idea by proposing an integrated predictive control framework where measurement-based feedback is used to deal with plant-model mismatch and disturbances. This framework consists of ideas drawn from the domains of model predictive control and nonlinear optimization. Case studies that demonstrate the performance of this scheme under parametric mismatch and process disturbance scenarios are presented.

5.1 Applying Predictive Control to the Shutdown Problem

Operating a plant undergoing a unit shutdown requires a suitable transient-control scheme. In the previous chapter, policies for operating the plant were generated using dynamic optimization. During the period of transience, unforeseen process disturbances and model mismatch can invalidate these policies, therefore some kind

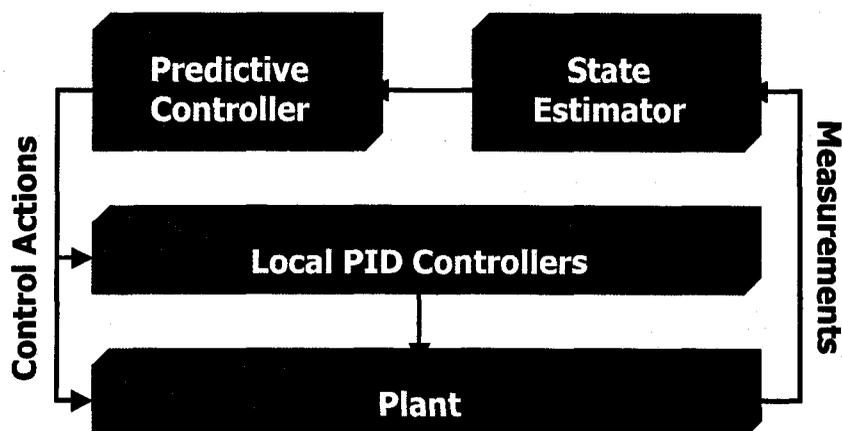


Figure 5.1: Hierarchy of control, integrated dynamic optimization and control.

of feedback mechanism is imperative in order for the control scheme to be practicable. Toward this end, we propose an integrated framework based on model predictive control and dynamic optimization as a means of implementing abnormal-situation control on the plant. The predictive control algorithm was selected for its centralized multivariate structure and constraint-handling capabilities.

We envisage a scenario where the operator switches the plant mode from “normal operation” to “abnormal operation” when a shutdown occurs, in which the proposed integrated control system below kicks in and takes over from the standard control system. The goal of our study is to demonstrate this framework in the role of a controller for the fiber line of the Kraft paper mill under a shutdown scenario.

Referring to Figure 5.1, the dynamic optimization component embedded within the predictive controller performs an open-loop model-based optimization based on the current states of the system. A set of control trajectories is obtained and the first step of the calculated trajectories is implemented. To carry out the control actions, the predictive controller can either send setpoints to lower-level PID controllers in local control loops or send control signals to the plant itself to drive it to the desired operating point. Plant measurements are taken and fed into the state estimator, which typically takes the form of an Extended Kalman Filter (EKF). The estimate of the states is then conveyed back to the predictive controller, where it is used as initial values for the next iteration. This process is repeated until the system is restored to its nominal steady state operation.

It is important to note that the predictive controller here is distinct from a conventional MPC controller in that its objective is not to track given setpoints, but instead to implement manipulated variable adjustments to optimize an economic objective.

5.2 Features of the Integrated Framework

5.2.1 Nonlinear Control of Transient Processes

Transient chemical processes have several characteristics which demand special attention. Firstly, they exhibit unsteady-state behavior which is generally nonlinear, and thus entails the use of some form of a nonlinear model in order to obtain a reasonably accurate characterization of the nonlinear dynamics in the system. In our scheme, the predictive controller is integrated with a dynamic optimizer (based on first-principles DAE models) where a series of general nonlinear programs (NLPs) are solved in succession.

General NLPs can be challenging to solve. Due to the nonlinear character of NLPs, a feasible starting point is crucial to obtaining a good solution. There are many ways of obtaining such a feasible initial point, including steady-state simulations, homotopy continuations or multiphase warm-starts. In addition, most general nonlinear problems in engineering have nonconvex formulations, which implies the existence of local optima [71], therefore global optimality cannot be guaranteed. Global optimization is one means to guarantee a global optimum but current global optimization software such as BARON [68] are currently only able to solve problems of a modest scale at great computational expense, effectively disqualifying them from medium to large-scale on-line applications.

In spite of these issues, solving repeated NLPs holds some appeal because the solution of the previous time step represents a good starting point for the current time instant, and by exploiting this property, successive NLPs can be solved efficiently [14]. In this work, single NLPs with 35,000 variables and 35,000 constraints were routinely solved in 0.5–2 minutes.

The transient processes we are considering are finite in duration, therefore the prediction/control horizon length in the controller is finite. Therefore, in lieu of the ordinary receding prediction/control horizon, a shrinking horizon (in which the prediction and

control horizons decrease as the the controller advances toward the end of the time horizon) seems to be the more natural choice and is adopted in this framework.

In the shutdown problem, the prediction horizon length for optimal control needs to correspond to the duration of the shutdown and restoration combined. If a shorter length is chosen, suboptimal control may result from the controller not having an adequate picture of the full transient process.

5.2.2 Economics-based objective function

Economics optimization is integrated directly into the predictive controller, which constitutes a 1-layer MPC with economics approach (in contrast to a multi-tier Dynamic RTO approach).

5.2.3 Events

A rudimentary mechanism for representing certain process events is built into the predictive controller. In this work, we consider two types of events:

1. Explicitly-known events

Due to integration with the dynamic optimizer, explicitly known discrete events (such as startups and shutdowns, or in our case, restoration) can be embedded directly into the prediction model. These events can be specified either through operator invention or automatically via monitoring subsystem, such as a process fault monitoring module. This form of process event anticipation is distinct from traditional feedforward control in which disturbances are detected solely through plant measurements.

The temporal entry and exit points of this type of transient event are usually specified relative to the left-hand boundary of the prediction window. Because the shrinking-horizon causes this reference boundary to move as it advances to the end of the time horizon, the entry/exit points of the transient event must be shifted accordingly as time progresses.

For a process that runs from $t \in [0, t_f]$, suppose a transient event $T(t)$ occurs between t_{start} and t_{end} , and the left-hand boundary of the prediction horizon is currently at time t_{curr} . In the predictive control optimization problem, these

variables take the following form:

$$t^{(opt)} \in [0, t_f - t_{curr}] \quad (5.1)$$

$$t_{start}^{(opt)} = t_{start} - t_{curr} \quad (5.2)$$

$$t_{end}^{(opt)} = t_{end} - t_{curr} \quad (5.3)$$

$$T(t^{(opt)}) \quad \text{for } t^{(opt)} \in [t_{start}^{(opt)}, t_{end}^{(opt)}] \quad (5.4)$$

where the superscript $^{(opt)}$ indicates variables within the optimization problem solved at time instant t_{curr} . As t_{curr} moves beyond t_{start} (i.e. $t_{curr} > t_{start}$), equation 5.2 is dropped and $t_{start}^{(opt)} = 0$. As t_{curr} moves beyond t_{end} (i.e. $t_{curr} > t_{end}$), equations 5.3 and 5.4 are dropped.

2. Optimization Induced events

Through the use of barrier functions discussed in the previous chapter (and other similar continuous switching functions), the predictive controller is able to deal with simple optimization-induced events, such an induced shutdown or a discontinuity in operation, without resorting to mixed-integer programming.

Process interlocks and stream redirections can be specified using continuous switching functions. For instance, if a shutdown is triggered and procedure dictates that under such circumstances, the flow of material in outlet F_1 (which normally connects F_2) is required to redirect to stream $F_{emergency}$, the following mathematical representation is possible:

$$F_1 = (1 - \omega)F_2 + \omega F_{emergency} \quad (5.5)$$

$$\omega = \begin{cases} 1, & \text{when failure occurs} \\ 0, & \text{otherwise} \end{cases} \quad (5.6)$$

where ω is approximated using some continuous switching function.

In either case, no integer variables are introduced and no hybrid structure is used, therefore the resulting optimization problem is solvable in principle using a conventional continuous NLP solver.

5.3 Algorithm and Software Implementation

The predictive control algorithm was implemented by interfacing a Python program and the dynamic optimization model (MLDO-generated AMPL model). A schematic

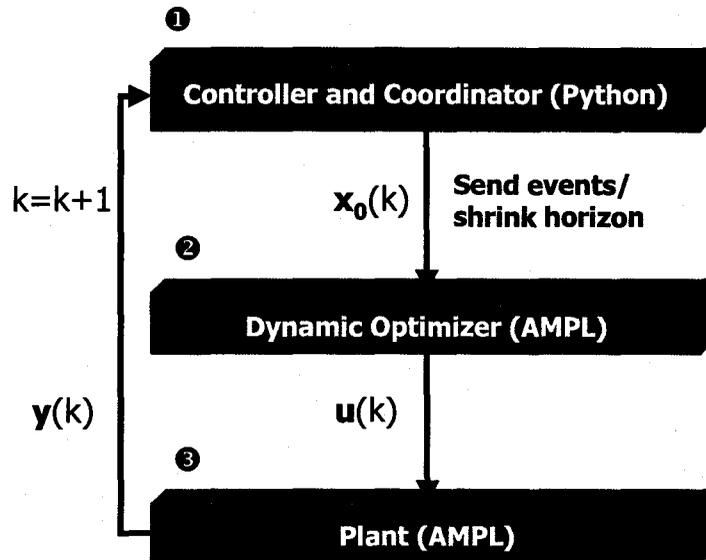


Figure 5.2: Software implementation of predictive control algorithm.

of the interactions between the different software is presented in Figure 5.2. The algorithm proceeds as follows:

- 1. Controller and Coordinator.** The coordinator program invokes MLDO and generates the dynamic optimization model. The initial state vector $\mathbf{x}(0)$ and events such as shutdowns are embedded in the model.
- 2. Dynamic Optimizer.** The dynamic optimizer has two parts: the AMPL modeling software and the IPOPT optimizer. Given a suitable model, the solution of the dynamic optimization problem is computed, and the control actions $\mathbf{u}(k)$ for the current time k are sent to the plant model. Note: only the first step is implemented.
- 3. Plant.** The plant model is integrated for 1 time step, and the measured variable vector $\mathbf{y}(k+1)$ is extracted and sent to the coordinator. The value of k is incremented by 1. The prediction/control horizons are reduced by 1 step.

Since the dynamic optimization problem is a nonlinear one, initializing the problem with a feasible starting guess is of paramount importance. The starting guess has the potential to dictate the region in which the optimizer moves and has a huge bearing on problem convergence. In the predictive control problem, the solution from the

previous optimization (at time $k - 1$) is used as an initial guess for the optimization problem at time k .

5.4 Interaction with Local Control Loops

In our Kraft model, the assumption is that the dynamics of the local controllers are fast in relation to the sampling rate, and may therefore be neglected. In our view, this is a reasonable assumption because all of the local loops considered are related to the control of flowrates, which have fast dynamics.

With respect to local level loops, they are required to deactivate during a shutdown because their role is subsumed by the dynamic optimizer, which controls the levels directly by manipulating the outlet flowrates of buffer units.

In general however, it may be desirable to take PID controller dynamics into account in the optimization problem [42]. PID control equations are easily accommodated in the dynamic optimization framework. Consider the following PID equations:

$$u = K_P \cdot e + K_I \int e dt + K_D \frac{de}{dt} \quad (5.7)$$

$$e = y_{set} - y \quad (5.8)$$

where

u = manipulated variable

K_P, K_I, K_D = proportional, integral and derivative gains, respectively.

e = error term

y = controlled variable

y_{set} = reference or setpoint value

These equations can be cast into a form amenable to solution with a DAE-based method as follows:

$$\frac{dz}{dt} = e, \quad \text{with } z(0) = 0 \quad (5.9)$$

$$K_D \frac{de}{dt} = u - K_P \cdot e - K_I z, \quad \text{with } e(0) = 0 \quad (5.10)$$

$$e = y_{set} - y \quad (5.11)$$

These equations can be enforced as constraints in the dynamic optimization problem.

5.5 State Estimation of a DAE System

The nonlinear predictive controller requires knowledge of the current state in order to compute the optimal trajectories for the remainder of the horizon. Frequently, only certain state variables are measurable¹ Some means of inferring the state information from the measurements must thus be employed. One algorithm that has been applied widely since the 1970s is the Kalman filter [46], which is an optimal linear estimation technique originally developed to estimate the states of stochastic processes. In view of the fact that our proposed framework uses nonlinear models, we will describe the nonlinear variant of this filter, the Extended Kalman Filter (EKF) [44].

The primary goal of an EKF is to be able to reconstruct state information from a set of (noisy) measurements. The EKF combines the variance of the measurements and the variance of the model states to estimate the the true process states and variance in a way that minimizes the variance of the true process estimate [16]. The basic idea is that given a vector of plant measurements at time k (denoted \mathbf{y}_k), a stochastically-optimal estimate of the state vector $\mathbf{x}_{k+1|k}$ (states at time $k+1$, given information at time k) can be derived. This estimate is then used as the initial values of the DAE system in the predictive control problem. The covariance matrix \mathbf{P}_k is updated at every time step.

In order to apply the EKF, we require the model of the system to be in a state-space type representation. Given a continuous DAE model of the following form:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \quad (5.12)$$

where \mathbf{x} = differential states, \mathbf{z} = algebraic states, \mathbf{u} = control inputs, the model can be rewritten in discretized form as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} \mathbf{F}(\mathbf{x}, \mathbf{z}, \mathbf{u}) dt = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) \quad (5.13)$$

where Δt = system sample time. The integration of the DAE model is performed implicitly using Orthogonal Collocation on Finite Elements. Combining this with plant measurements (\mathbf{y}_k), and adding stochastic noise terms, we obtain a state-space

¹In this work, which is based on the Kraft mill fiber line model, the use of the EKF state estimator is bypassed because the values of the differential algebraic states necessary to uniquely determine the current operating point of the system are directly measurable.

type representation below.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k \quad (5.14)$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, k) + \mathbf{v}_k \quad (5.15)$$

where $\mathbf{w}_k \sim N(0, \mathbf{Q})$ and $\mathbf{v}_k \sim N(0, \mathbf{R})$.

Initialization

At time 0, the state vector \mathbf{x} is assigned a vector of values (obtained by some reasonable means) whose function is that of a "first guess". The \mathbf{Q} matrix is used the initial covariance matrix.

$$\mathbf{P}_{0|0} = \mathbf{Q} \quad (5.16)$$

$$\mathbf{x}_{0|0} = \mathbf{x}_0 \quad (5.17)$$

Estimation

The state estimation involves two phases, an update phase and a predict phase (Henson, [26]). The nonlinear equation system is linearized at each instant through the use of Jacobians.

1. Update Phase

In the update phase, the Kalman gain (\mathbf{K}_k) is calculated using equation 5.18. This gain is primarily for updating the covariance matrix (through equation 5.19) and for determining how the feedback measurements are used to update the current states (equation 5.20).

$$\mathbf{K}_k = \hat{\mathbf{P}}_{k|k-1} \mathbf{G}_k^T (\mathbf{G}_k \hat{\mathbf{P}}_{k|k-1} \mathbf{G}_k^T + \mathbf{R})^{-1} \quad (5.18)$$

$$\hat{\mathbf{P}}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k) \hat{\mathbf{P}}_{k|k-1} \quad (5.19)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{x}}_{k|k-1}, k)) \quad (5.20)$$

2. Predict Phase

In the predict phase, the state covariance matrix is updated (equation 5.21) and the current state is propagated nonlinearly using equation 5.22. In a discretized DAE model, the propagation function is an implicit function in the form of equation 5.13.

$$\hat{\mathbf{P}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{P}}_{k|k} \mathbf{F}_k^T + \mathbf{Q} \quad (5.21)$$

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) \quad (5.22)$$

where

$$\mathbf{G}_k = \left. \frac{\partial \mathbf{g}(\mathbf{x}, k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}} \quad (5.23)$$

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}, \mathbf{u}=\mathbf{u}_k} \quad (5.24)$$

$\mathbf{x}_{a|b}$ = state vector at time a , given information at time b .

\mathbf{Q}, \mathbf{R} = covariance matrices of \mathbf{w}_k and \mathbf{v}_k , respectively.

\mathbf{F}, \mathbf{G} = Jacobians of \mathbf{f} and \mathbf{g} , respectively.

$\mathbf{P}_{a|b}$ = state covariance matrix at time a , given information at time b .

\mathbf{y}_k = plant measurement vector at time k

\mathbf{K}_k = Kalman gain at time k

\mathbf{I} = Identity matrix

Kozub and MacGregor demonstrate that the augmentation of the state-space equations with random-walk stochastic states provides the state estimator with integral action, which is essential for handling process nonstationarity [52]. Without integral action, the model is unable to account for say, a transition to a different steady state. From a practical point of view, Froisy et al. [32] stress that a minimum set of stochastic states should be used. Kozub and MacGregor also demonstrate a "reiterative"-EKF, where the first few measurements are used to estimate the initial states of the system (\mathbf{x}_0), which are frequently not known.

Froisy et. al. [32] make the comment that in large-scale applications where there are thousands of states, it is impractical to update all of them, therefore the selection of which states to update is dependent on the strength of the connection between the measurements and the set of states.

In order for state estimation to be effective, it is vital for the system to possess the quality of observability. While the observability of a linear system is a global property, the observability of nonlinear system can only be determined around the neighborhood of a given state or equilibrium point by means of a linearized test [26]. Ray [66] notes however that observability of nonlinear systems is often dictated by the structure of a system and does not depend on the state in a complex fashion, hence linearized observability tests are usually sufficient.

5.6 Case Studies

In this section, the performance of the predictive control algorithm is examined under different scenarios. In all of the cases below, the vacuum drum washer was shut down for 8 hours (between hours 2–10 in the figures) and controller is expected adjust the production rate and outlet flowrates of buffer tanks in order to achieve optimal operation. The length of each simulation is 24 hours. In this study, the effects of step disturbances and plant-model mismatch are considered. We also present a case where reoptimization is performed to account for an updated estimated downtime to demonstrate the use of the reoptimization scheme under predictive control. The simplified schematic of the plant is repeated here for the reader's convenience (Figure 5.3).

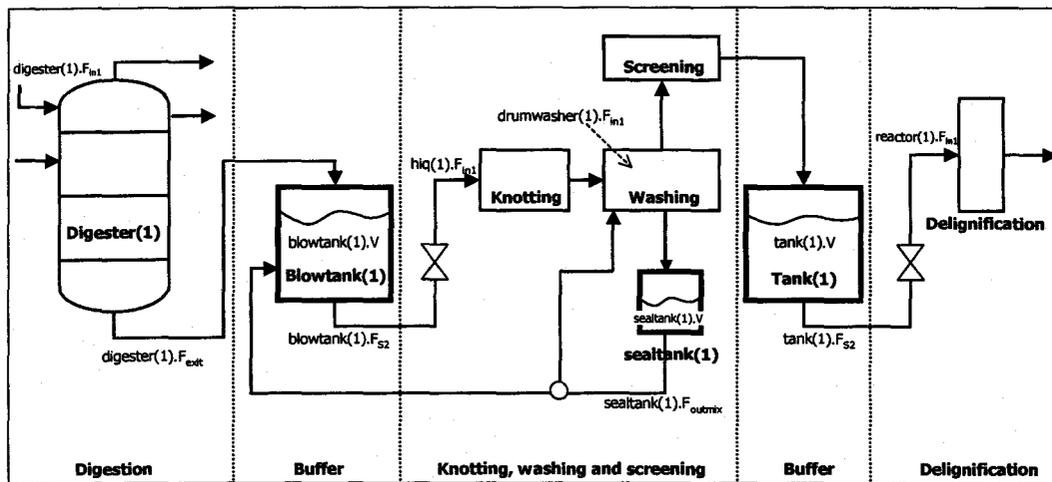


Figure 5.3: Simplified schematic of plant model.

5.6.1 Case 1: Implementing the Optimal Control Policy from Dynamic Optimization (Nominal Case) with the Predictive Controller

Description

In case 1, the trajectories obtained from the dynamic optimization problem were implemented on the plant, under perfect model conditions. This case is used as a benchmark for the other cases.

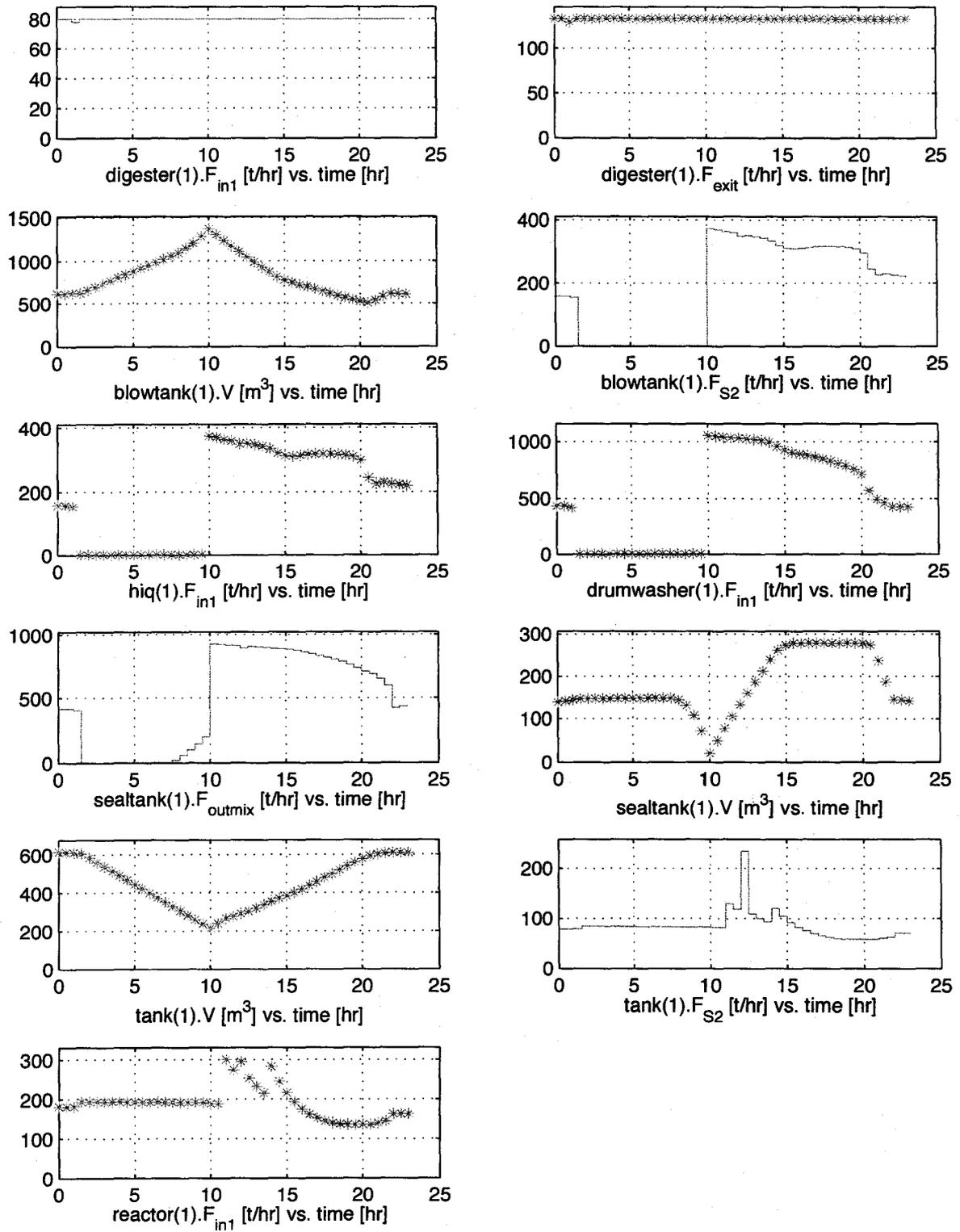


Figure 5.4: Case Study 1: Dynamic Optimization Solution, closed loop.

Results

In the nominal case (refer to Figure 5.4), the application of the predictive control algorithm produced results that were identical to those computed with the open-loop dynamic optimizer. During most of the shutdown period, the digester is seen to be operating normally. Because the vacuum drum washer is shut down, the blowtank discharge ($\text{blowtank}(1).F_{S2}$) is shut off. The material entering the blowtank from the digester causes the material level in the blowtank to rise during this period.

The recycle is shut off during most of the shutdown period, as can be seen in the $\text{sealtank}(1).F_{outmix}$ trajectory. The controller anticipates the end of the shut down time by slowly starting up the recycle by increasing the $\text{sealtank}(1).F_{outmix}$ flowrate, which in turn causes a sudden drop in the level of the sealtank ($\text{sealtank}(1).V$). This drop allows the sealtank to accommodate more material during the restoration phase, and we can see that as the system is restored, the output from the blowtank ($\text{blowtank}(1).F_{S2}$) surges and the sealtank level is pushed to its upper bound before being restored to its original level.

Throughout the shutdown and restoration, the delignification reactor experienced spikes and a temporary decline in production.

5.6.2 Case 2: Step Disturbance in Chip Feed

Description

The chip feed stream in the digester units comprises chipped softwood lumber, dissolved solids and water. A feed disturbance is simulated by effecting a transient step change in the wood concentration (see Table 5.1) from time 11–21 hrs, during the restoration phase. The composition levels are subsequently restored to their original values after the 21st hour. The purpose of this case study is demonstrate how the predictive controller handles step disturbances.

	Original Composition	Composition during Disturbance
Wood	0.43	0.20
Water	0.53	0.76
Dissolved solids	0.04	0.04

Table 5.1: Composition of Chip Feed Stream

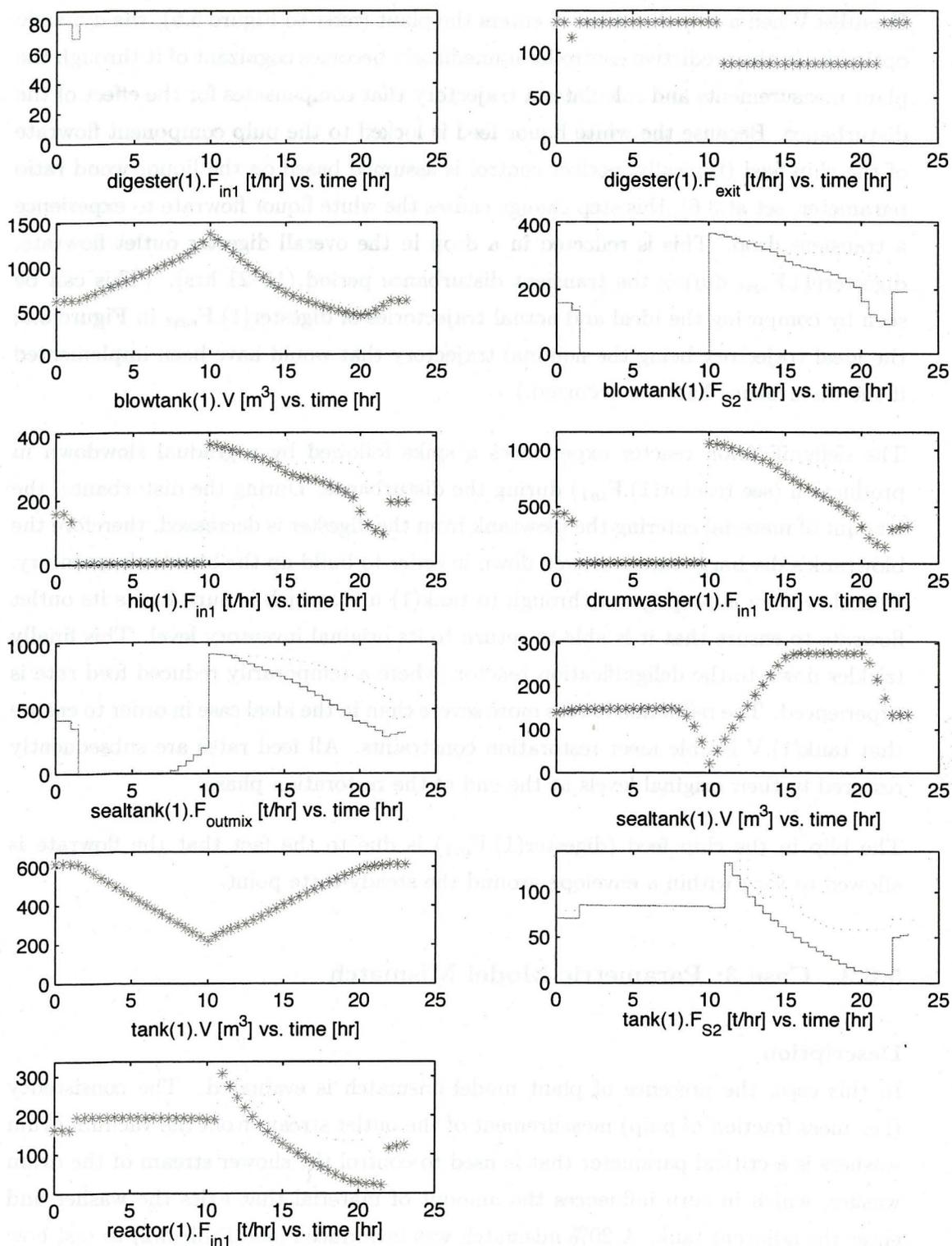


Figure 5.5: Case Study 2: Disturbance in the feed composition.

Legend: dotted lines – ideal trajectory. asterisks/solid line – actual trajectory.

Results When a step disturbance enters the plant (refer to Figure 5.5), the dynamic optimizer in the predictive controller immediately becomes cognizant of it through the plant measurements and calculates a trajectory that compensates for the effect of the disturbance. Because the white liquor feed is locked to the pulp component flowrate of the chip feed (typically, perfect control is assumed based on the liquor-wood ratio parameter, set at 3.6), this step change causes the white liquor flowrate to experience a transient drop. This is reflected in a drop in the overall digester outlet flowrate, $\text{digester}(1).F_{exit}$ during the transient disturbance period (11–21 hrs). (This can be seen by comparing the ideal and actual trajectories of $\text{digester}(1).F_{exit}$ in Figure 5.5; the ideal trajectory being the nominal trajectory that would have been implemented if the disturbance had not occurred.)

The delignification reactor experiences a spike followed by a gradual slowdown in production (see $\text{reactor}(1).F_{in1}$) during the disturbance. During the disturbance, the amount of material entering the blowtank from the digester is decreased, therefore the blowtank's discharge rate is slowed down in order to build up the blowtank inventory. This slowdown is propagated through to tank(1) unit, which in turn limits its outlet flowrate to ensure that it is able to return to its original inventory level. This finally trickles down to the delignification reactor, where a temporarily reduced feed rate is experienced. The reduction here is more severe than in the ideal case in order to ensure that $\text{tank}(1).V$ is able meet restoration constraints. All feed rates are subsequently restored to their original levels at the end of the restoration phase.

The blip in the chip feed ($\text{digester}(1).F_{in1}$) is due to the fact that the flowrate is allowed to vary within an envelope around the steady-state point.

5.6.3 Case 3: Parametric Model Mismatch

Description

In this case, the presence of plant model-mismatch is evaluated. The consistency (i.e. mass fraction of pulp) measurement of the outlet stream from the vacuum drum washers is a critical parameter that is used to control the shower stream of the drum washer, which in turn influences the amount of material that exits the washer and enter the adjacent tank. A 20% mismatch was introduced (see Table 5.2) to test how the control system would react.

Results

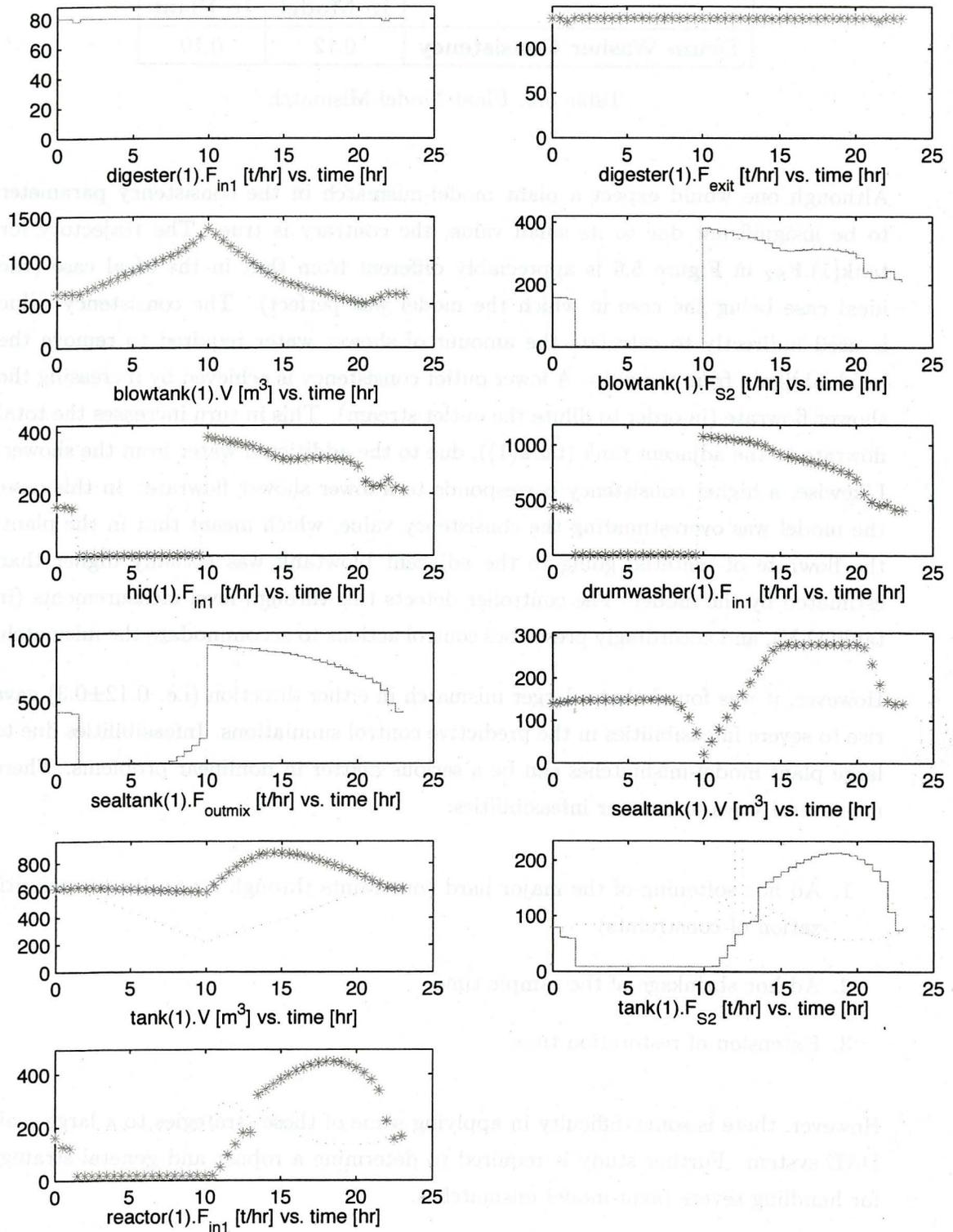


Figure 5.6: Case Study 3: Plant-model mismatch in value of consistency in vacuum washer outlet.

Legend: dotted lines – ideal trajectory. asterisks/solid line – actual trajectory.

	In Model	In Plant
Drum Washer Consistency	0.12	0.10

Table 5.2: Plant-Model Mismatch

Although one would expect a plant model-mismatch in the consistency parameter to be insignificant due to its small value, the contrary is true. The trajectory for $\text{tank}(1).F_{S2}$ in Figure 5.6 is appreciably different from that in the ideal case (the ideal case being the case in which the model was perfect). The consistency value is used indirectly to calculate the amount of shower water required to remove the residual lignin from the pulp. A lower outlet consistency is achieved by increasing the shower flowrate (in order to dilute the outlet stream). This in turn increases the total flowrate to the adjacent tank ($\text{tank}(1)$), due to the additional water from the shower. Likewise, a higher consistency corresponds to a lower shower flowrate. In this case, the model was overestimating the consistency value, which meant that in the plant, the flowrate of material going to the adjacent blowtank was actually higher than estimated by the model. The controller detects this through level measurements (in $\text{tank}(1).V$), and accordingly prescribes control actions to accommodate the mismatch.

However, it was found that a larger mismatch in either direction (i.e. 0.12 ± 0.3) gave rise to severe infeasibilities in the predictive control simulations. Infeasibilities due to large plant model-mismatches can be a serious matter in nonlinear problems. There are several ways to counter infeasibilities:

1. Ad hoc softening of the major hard constraints through a penalty (or prioritization of constraints)
2. Ad hoc shrinkage of the sample time
3. Extension of restoration time

However, there is some difficulty in applying some of these strategies to a large scale DAE system. Further study is required to determine a robust and general strategy for handling severe plant-model mismatches.

5.6.4 Case 4: Reoptimization upon feedback of downtime estimate

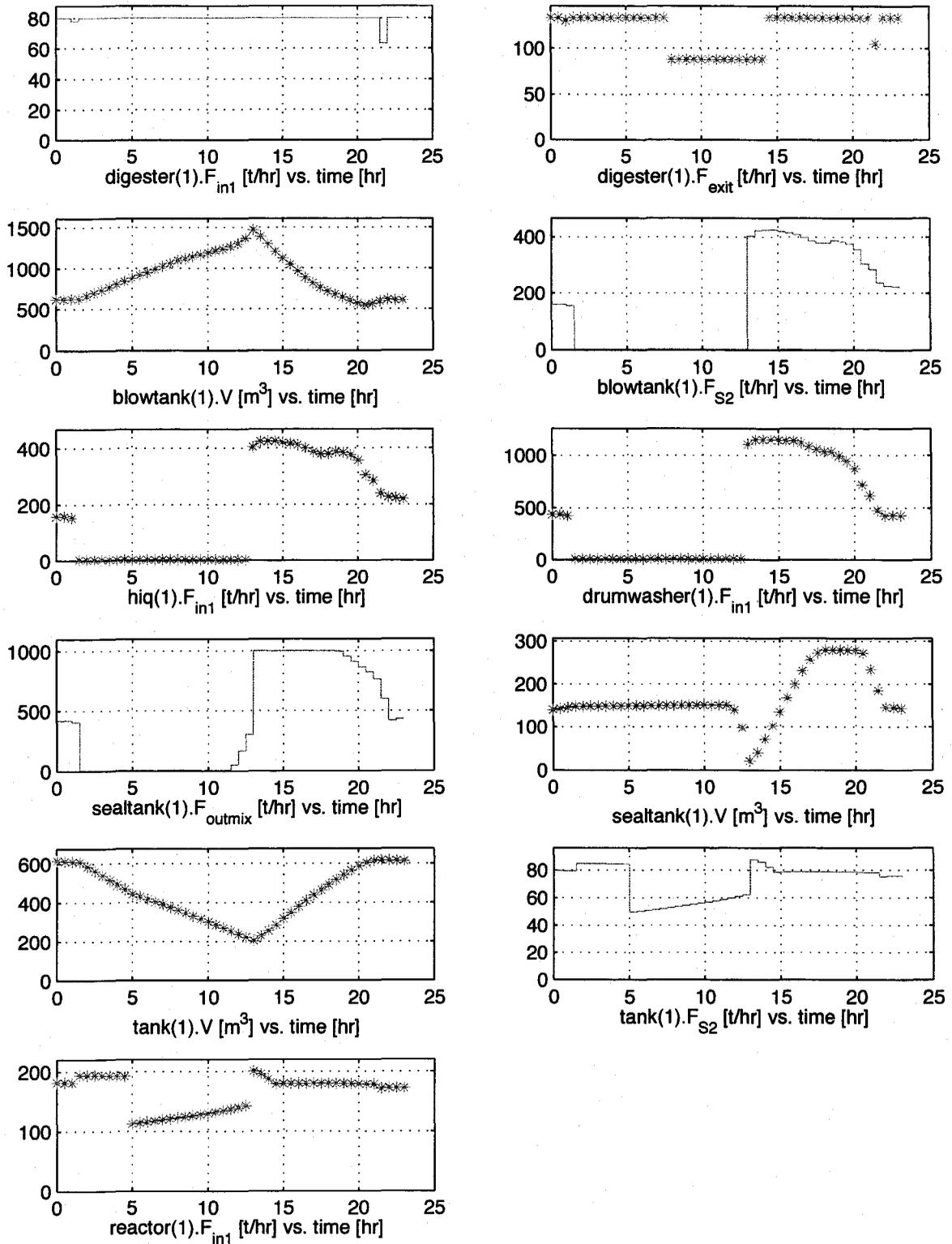


Figure 5.7: Case Study 4: Reoptimization upon update of downtime estimate.

Description

In this scenario, a shutdown occurs at $t=2$ hrs. The estimated downtime (d_{est}) is 8 hours, but the actual downtime (d_{act}) is 11 hours. The operator is apprised of the actual downtime at $t=5$ hrs. This information is fed to the predictive controller, and it reoptimizes the remainder of the trajectory to account for the extended downtime. A disturbance in the feed (of the type described in Case 2, above) enters the system between $t=8-14$ hrs. This case study demonstrates the application of the reoptimization scheme under a predictive control framework.

Results

The blips in the chip feed ($digester(1).F_{in1}$) are due to the flowrate being allowed to vary within an envelope around the steady-state point. The disturbance in the feed causes a step down of the digester output in the $digester(1).F_{exit}$ trajectory between $t=8-14$ hrs.

The trajectory of $tank(1).F_{S2}$ in Figure 5.7 shows visible indications of a mid-course correction at time $t=5$ hrs. This is propagated to the inlet of the delignification reactor, seen in the $reactor(1).F_{in1}$ trajectory. This was done to control the descent rate of the volume level in $tank(1)$ and to aid in the restoration of the system to the original steady-state.

This case demonstrates two kinds of feedback in action: downtime feedback to correct for an incorrect downtime estimate, and feedback to deal with disturbances.

5.7 Chapter Summary

In this chapter, a predictive control framework for implementing shutdown policies was proposed. This framework comprises a nonlinear predictive control system which is able to accommodate events and an objective function based on economics. Interaction with local PID loops and the use of a state estimator within this framework was discussed. The case studies showed successful applications of this control scheme to feed disturbance, plant-model mismatch and uncertain downtime scenarios.

Chapter 6

Conclusion

The primary focus of this work has been on the use of dynamic optimization for coordinating the buffer capacities in a plant in order to mitigate the effects of a unit shutdown. The model employed was a large-scale differential algebraic equation system which was solved using a simultaneous-type method. Uncertainty in the downtime estimate parameter, a parameter which has a significant effect on the optimality of the problem, was considered. Methods to handle this uncertainty were proposed and compared. A predictive control framework for implementing the above optimal control policies was tendered for the detection and correction of disturbances and/or plant-model mismatch. Several case studies illustrating its application were presented.

Key findings from this work include:

1. **Modeling Language for Dynamic Optimization.** A modeling language for describe dynamic optimization problems was presented in this thesis. Code written in this canonical form can be used to generate code in other languages. Examples include AMPL code for collocation of finite elements, gPROMS code for NLP initialization, \LaTeX code for automatic documentation, etc. The modeling environment, which is based on the principles of text transformations, enables the process control researcher to do rapid prototyping and feature experimentation without writing an excessive amount of boilerplate code.
2. **Reoptimization - Feedback of Downtime Estimate.** Because the downtime estimate is a parameter required from the onset, and because there is often

no means of obtaining this number *a priori* apart from the prognostications of operators and maintenance personnel (based on their experience), it therefore becomes necessary to develop some means of addressing the uncertainties of the initial estimate. One way is to assign uncertainty bounds to the estimate, and optimize to obtain a set of trajectories that are feasible for any estimate that lie within those bounds through multiscenario optimization. This results in fairly conservative trajectories. The more optimal way is to allow the operator to input an initial estimate, and then to update that estimate as new information arrives. The control system performs a reoptimization upon the update and prescribes a midcourse correction to the plant. In our studies, this method performs significantly better than multiscenario optimization in terms of economics.

3. **Restoration of states and inputs.** Essentially, it is important to recognize that a unit shutdown is a transient event which ought not to permanently shift the original operating point of the process. As such, it is mandatory that states and inputs are made to return to their original pre-shutdown values through restoration constraints. This ensures that the result obtained from the optimization is meaningful; if the states and inputs were not restored, the optimizer would pursue avenues for optimizing the objective by coercing the system to a new operating point, which violates the mandate of a control system that is designed to handle only the transient event. It also complicates the quantification of the economics of the system as inventory deviation costs will then have to be accounted for.
4. **Induced shutdowns.** Induced shutdowns are triggered by long shutdowns in a specific unit. When a unit shutdown is very long, it is liable to trigger the violation of level constraints in buffer capacities (that is, either the tanks overflow or empty). The optimizer responds to this by lowering or increasing the rate of production in upstream units. In the case where a reduction in upstream production is needed, instead of throttling down the production over the horizon, the optimizer may prescribe shutting off the upstream production completely for a limited time. This is typically undesirable as the costs and manpower requirements of a shutdown often exceeds the losses incurred through a reduced production rate. In this study, we showed that induced shutdowns can and need to be penalized appropriately in the objective function.
5. **Predictive Control.** Disturbances and model uncertainties during the shut-

down can cause the trajectories to deviate from the optimal control policy. A predictive control algorithm encompassing a dynamic optimizer was proposed for countering these effects by means of feedback. Because a dynamic optimizer is encapsulated within the framework, economic objective functions and anticipation of explicitly known future events are possible. Case studies were presented to illustrate the performance of this scheme.

6.1 Recommendations for Further Work

This section lists several possibilities for extensions to this work.

1. **Quasi-sequential Approach for DAE Optimization.** One of the biggest problems in terms of solving the problems with shutdowns with the simultaneous method is finding a suitable initialization for the NLP. Currently, the NLP is initialized with a steady-state profile obtained from a process simulation. However, due to the fact that shutdowns entail drastic changes in the shape of the trajectory, initialization with a steady-state profile can occasionally fail because it does not adequately capture the actual shape of the optimal state profiles. The simultaneous method generally works well with fairly well-behaved profiles but the profiles in this work are far from well-behaved. The sequential method, on the other hand, does not suffer from the problem of poor initialization due to its use of an integrator to obtain profiles. There have been attempts to hybridize the two approaches, resulting in a quasi-sequential type approach. This type of approach appears to be a promising candidate method for solving shutdown optimization problems.
2. **Multiple Shutdowns in Series.** In this thesis, apart from induced shutdowns, only individually occurring shutdowns were considered. However it is possible for two or more units to shutdown independently of each other. Depending on the level of integration, this may result in a very challenging buffer coordination problem.
3. **Dynamic Recovery from Infeasibilities.** In nonlinear predictive control applications, particularly in shutdown situations when drastic control actions are prescribed, severe plant model-mismatch can trigger infeasibilities in the

problem. The conventional way to ensure a feasible problem is through constraint softening, but in a large scale problem determining which constraints to soften (and assigning them appropriate penalty weights) may be problematic. Improperly chosen penalty weights can also affect the optimality of the problem. Finding a method that reconfigures the problem dynamically upon hitting an infeasibility is a subject for further study.

4. **Models with Heat and Mass Transfer.** In our model, only the problem of managing material inventories was considered. The framework however, is general and can be extended to processes that involve chemical reactions or physical separations. For instance, it may sometimes (albeit rarely) be possible to slow down a reaction by limiting catalyst addition to a reactor in response to the need to throttle down production rate in order to accommodate a shutdown in some unit.
5. **Minimization of Restoration Time.** In most cases, it is desirable to minimize restoration time (the time taken to return a shut down system to its initial operating point). One conceivable way of doing this is to successively solve the dynamic optimization problem, decrementing t_{res} by 1 unit each time, and terminating when the problem becomes infeasible. However, if there is a process disturbance or model mismatch, minimization of restoration time may give rise to infeasibilities (arising out of the failure to meet restoration constraints), therefore there needs to be a mechanism to extend restoration time when infeasibilities are detected.
6. **Relaxation of Pseudo-Steady State Assumption.** In this work, most of the process units (apart from the tanks) were assumed be in pseudo-steady state operation. Inclusion of actual dynamics in process units may lead to improved performance in actual industrial application.
7. **Inclusion of Startup and Shutdown Dynamics.** In an industrial setting, there is usually an explicit procedure in place for starting up or shutting down a process unit. This procedure may be in the form of a sequence of manual operator actions from a look-up table, or a set of control actions from a process automation system. In order to fully capture the effects resulting from the implementation of a startup or shutdown procedure, it is recommended that the dynamics of the startup and shutdown included in the model.

List of References

- [1] O. Abel and W. Marquardt. Scenario-integrated modeling and optimization of dynamic systems. *AIChE Journal*, 46(4):803 – 823, 2000.
- [2] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques and Tools*. Addison Wesley, 1986.
- [3] N. M. Alexandrov and R. M. Lewis. Analytical and computational aspects of collaborative optimization (<ftp://techreports.larc.nasa.gov/pub/techreports/larc/2000/tm/NASA-2000-tm210104.pdf>). Technical report, NASA Langley Research Center, Hampton VA, 2000.
- [4] B. Allison. Averaging level control for surge tanks in series. In *Control Systems Conference*. Technical Association of the Pulp and Paper Industry Press, 2004.
- [5] J. J. Alonso, P. LeGresley, E. Van Der Weide, J. R. R. A. Martins, and J. J. Reuther. pyMDO: A framework for high-fidelity multi-disciplinary optimization. *Collection of Technical Papers – 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 3:2001 – 2019, 2004.
- [6] U. M. Ascher and L. R. Petzold. Projected implicit Runge-Kutta methods for differential-algebraic equations. *SIAM Journal on Numerical Analysis*, 28(4):1097 – 1120, 1991.
- [7] R. Baker and C. L. E. Swartz. Rigorous handling of input saturation in the design of dynamically operable plants. *Industrial and Engineering Chemistry Research*, 43(18):5880 – 5887, 2004.
- [8] A. K. S. Balthazaar. Dynamic optimization of multi-unit systems under failure conditions. Master’s thesis, McMaster University, 2005.

- [9] P. I. Barton and C. C. Pantelides. gPROMS – a combined discrete/continuous modelling environment for chemical processing systems. *International Conference on Simulation in Engineering Education. Proceedings of the 1993 SCS Western Simulation Multiconference on Simulation in Engineering Education*, pages 25 – 34, 1993.
- [10] V. M. Becerra, Z. H. Abu-El-Zeet, and P. D. Roberts. Integrating predictive control and economic optimisation. *IEE Colloquium (Digest)*, (96):11 – 15, 1999.
- [11] L. T. Biegler, A. M. Cervantes, and A. Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57(4):575 – 593, 2002.
- [12] L. T. Biegler and I. E. Grossmann. Retrospective on optimization. *Computers and Chemical Engineering*, 28(8):1169 – 1192, 2004.
- [13] K. F. Bloss, L. T. Biegler, and W. E. Schiesser. Dynamic process optimization through adjoint formulations and constraint aggregation. *Industrial and Engineering Chemistry Research*, 38(2):421 – 432, 1999.
- [14] D. Bonvin. Optimal operation of batch reactors - a personal view. *Journal of Process Control*, 8(5-6):355 – 368, 1998.
- [15] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM J. Sci. Comput. (USA)*, 15(6):1467 – 88, 1994.
- [16] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley and Sons, 1992.
- [17] J. J. Castro and F. J. Doyle III. A pulp mill benchmark problem for control: application of plantwide control design. *J. Process Control (UK)*, 14(3):329 – 47, 2004.
- [18] Z. Chong. Technical Report on a Modeling Language for Dynamic Optimization (MLDO). Technical report, McMaster University, 2006.
- [19] Thermo Electron Corporation. Gamma based consistency measurement in the pulp and paper industry (http://www.thermo.com/eThermo/CMA/PDFs/Product/productPDF_14149.pdf). Technical report, Thermo Electron Corporation, 2006.

- [20] J.E. Cuthrell and L.T. Biegler. On the optimization of differential-algebraic process systems. *AIChE J. (USA)*, 33(8):1257 – 70, 1987.
- [21] C. R. Cutler and B. L. Ramaker. Dynamic matrix control – a computer control algorithm. *AIChE 86th National Meeting, Houston, TX, I*, 1979.
- [22] J. Czyzyk, M. P. Mesnier, and J. J. More. NEOS server. *IEEE Computational Science & Engineering*, 5(3):68 – 75, 1998.
- [23] A. Drud. *GAMS: The Solver Manuals (CONOPT)*. GAMS Development Corporation, 2004.
- [24] J. F. H. Dubé. Pulp mill scheduling: Optimal use of storage volumes to maximize production. Master's thesis, McMaster University, 2000.
- [25] I. S. Duff. On algorithms for obtaining a maximum traversal. *ACM Transactions on Mathematical Software*, 1981.
- [26] T. Edgar and K. Muske. *Nonlinear Process Control*. Prentice-Hall, 1996.
- [27] Energy Efficiency and Renewable Energy. How to calculate the true cost of steam. Technical report, U.S. Department of Energy, 2003.
- [28] M. Englehart. High quality automatic code generation for control applications. In *Proceedings., IEEE/IFAC Joint Symposium on 7-9 March 1994*, pages 363 – 367, Tucson, AZ, USA, 1994.
- [29] D. Feather, D. Harrell, R. Lieberman, and F. J. Doyle III. Hybrid approach to polymer grade transition control. *AIChE Journal*, 50(10):2502 – 2513, 2004.
- [30] W. F. Feehery and P. I. Barton. Dynamic optimization with state variable path constraints. *Comput. Chem. Eng. (UK)*, 22(9):1241 – 56, 1998.
- [31] R. Fourer, D. M. Gay, and B. W. Kernighan. A modeling language for mathematical programming. *Manage. Sci. (USA)*, 36(5):519 – 54, 1990.
- [32] B. Froisy, C. Hart, S. Lingard, and S. Papastratos. Industrial application of online first principle dynamic models using state estimation. In *AIChE Annual Meeting*, 1999.
- [33] C. E. Garcia. Quadratic/dynamic matrix control of nonlinear processes - an application to a batch reaction process. *AIChE 91st National Meeting, San Francisco, CA*, 1984.

- [34] C. E. Garcia and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.
- [35] R. Gebart, L. Westerlund, and I. Landalv. Black liquor gasification—the fast lane to the biorefinery. In *Energy Conference (Risø National Laboratory)*, 2005.
- [36] N. I. M. Gould and Ph. L. Toint. How mature is nonlinear optimization? (<ftp://ftp.numerical.rl.ac.uk/pub/reports/gtFUNDP0304.pdf>). Technical report, Rutherford Appleton Laboratory and University of Namur, 2004.
- [37] T. M. Grace, B. Leopold, and E. W. Malcolm. Pulp and paper manufacture - alkaline pulping. vol 5. In *The Joint Textbook Committee of the Paper Industry, Montreal, QC, Canada*, 1989.
- [38] I. E. Grossmann and M. Morari. Operability, resiliency and flexibility - process design objectives for a changing world. In *Proceedings of the 2nd International Conference on Foundations of Computer-Aided Process Design*, 1984.
- [39] M. Hillestad and K. S. Andersen. Model predictive control for grade transitions of a polypropylene reactor. *Fourth European Symposium on Computer Aided Process Engineering. ESCAPE 4*, page 41.6, 1994.
- [40] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation (2nd Edition)*. Addison Wesley, 2000.
- [41] Y. J. Huang, G. V. Reklaitis, and V. Venkatasubramanian. Dynamic optimization based fault accommodation. *Computers and Chemical Engineering*, 24(2):439 – 444, 2000.
- [42] Y. J. Huang, V. Venkatasubramanian, and G. V. Reklaitis. A model-based fault accommodation system. *Industrial and Engineering Chemistry Research*, 41(16):3806 – 3821, 2002.
- [43] Y. J. Huang, V. Venkatasubramanian, and G. V. Reklaitis. Model decomposition based method for solving general dynamic optimization problems. *Computers and Chemical Engineering*, 26(6):863 – 873, 2002.
- [44] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

- [45] J. V. Kadam, M. Schlegel, W. Marquardt, R. L. Toussain, D. H. van Hessem, J. van den Berg, and O. H. Bosgra. A two-level strategy of integrated dynamic optimization and control of industrial processes - a case study. Technical report, Lehrstuhl für Prozesstechnik, RWTH Aachen, 1999.
- [46] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, pages 33–45, 1960.
- [47] F. Kayihan. Process systems management in the pulp and paper industries: an optimization approach. *Pulp and Paper Canada*, 98(8):37–40, 1997.
- [48] M.J. Kempe. Dynamic modelling of a vacuum drum washing system. Master's thesis, McMaster University, 1995.
- [49] M. Krothapally and S. Palanki. On-line optimization of batch polymerization processes. *Proceedings of the American Control Conference*, 2:1187 – 1190, 1997.
- [50] E. S. Lee and G. V. Reklaitis. Intermediate storage and operation of batch processes under batch failure. *Computers and Chemical Engineering*, 13(4-5):491 – 498, 1989.
- [51] E. S. Lee and G. V. Reklaitis. Intermediate storage and the operation of periodic processes under equipment failure. *Comput. Chem. Eng. (UK)*, 13(11-12):1235 – 43, 1989.
- [52] J. F. MacGregor, D. J. Kozub, A. Penlidis, and A. E. Hamielec. State estimation for polymerization reactors. *Dynamics and Control of Chemical Reactors and Distillation Columns. Selected Papers from the IFAC Symposium*, pages 147 – 52, 1988.
- [53] R. D. M. MacRosty. *Modeling, Optimization and Control of an Electric Arc Furnace*. PhD thesis, McMaster University, 2005.
- [54] B. A. McCarl. GAMS user guide <http://www.gams.com/dd/docs/bigdocs/gams2002/mccarlgamsuserguide.pdf>. Technical report, Texas A and M University, 2002.
- [55] A. Mione. A look at Python. *Digit. Syst. Rep. (USA)*, 20(2):1 – 7, Summer 1998.
- [56] Z. K. Nagy and R. D. Braatz. Robust nonlinear model predictive control of batch processes. *AIChE Journal*, 49(7):1776 – 1786, 2003.

- [57] A. Noël, M. Savoie, H. Budman, and L. Lafontaine. Advanced brownstock washer control: Successful industrial implementation at James McLaren. In *2nd IEEE Conference on Control Applications, Vancouver B.C.*, Sep. 13-16, 1993.
- [58] H. V. Norden and M. Jarvekainen. Analysis of a pulp washing system consisting of unequal stages in series. *Kemian Toellisuus*, 23(4):344-351, 1966.
- [59] C. C. Pantelides. SpeedUp – recent advances in process simulation. *Comput. Chem. Eng. (UK)*, 12(7):745 – 55, 1988.
- [60] A. A. Patwardhan, J. Rawlings, and T. Edgar. Nonlinear model predictive control. *Chemical Engineering Communications*, 87:123-141, 1990.
- [61] B. Pettersson. Production control of a complex integrated pulp and paper mill. *TAPPI Journal*, 52(11):2155-2159, 1969.
- [62] B. Pettersson. Optimal production schemes coordinating subprocesses in a complex integrated pulp and paper mill. *Pulp and Paper Canada*, 71(5):59-63, 1970.
- [63] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The mathematical theory of optimal processes*. Interscience, NY, 1962.
- [64] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.
- [65] W. F. Ramirez. *Process Control and Identification*. Academic Press, 1993.
- [66] W. H. Ray. *Advanced Process Control*. McGraw-Hill, New York, 1981.
- [67] D. Ruppen, C. Benthack, and D. Bonvin. Optimization of batch reactor operation under parametric uncertainty - computational aspects. *Journal of Process Control*, 5(4):235 – 240, 1995.
- [68] N. V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201 – 205, 1996.
- [69] G. Schopfer, A. Yang, L. von Wedel, and W. Marquardt. Cheops: a tool-integration platform for chemical process modelling and simulation. *Int. J. Softw. Tools Technol. Transf. (Germany)*, 6(3):186 – 202, 2004.
- [70] G. A. Smook. *Handbook for Pulp and Paper Technologists*. Angus Wilde Publications, 2nd edition, 1999.

- [71] B. Srinivasan, E. Visser, and D. Bonvin. Optimization-based control with imposed feedback structures. In *ADCHEM Proceedings*, 1997.
- [72] C. Sun and J. Hahn. Reduction of stable differential-algebraic equation systems via projections and system identification. *Journal of Process Control*, 15(6):639 – 650, 2005.
- [73] I. B. Tjoa and L. T. Biegler. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial and Engineering Chemistry Research*, 30(2):376 – 385, 1991.
- [74] T. Tosukhowong, J. M. Lee, J. H. Lee, and J. Lu. An introduction to a dynamic plant-wide optimization strategy for an integrated plant. *Computers and Chemical Engineering*, 29(1):199 – 208, 2004.
- [75] E. Tufte. Sparklines: theory and practice (http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00010R&topic_id=1). Technical report, 2006.
- [76] P. A. Turner, A. A. Roche, J. D. McDonald, and A. R. P. van Heiningen. Dynamic behavior of a brown stock washing plant. *Pulp and Paper Canada*, 94(9):37–42, 1993.
- [77] S. Vasantharajan and L. T. Biegler. Simultaneous strategies for optimization of differential-algebraic systems with enforcement of error criteria. *Computers and Chemical Engineering*, 14(10):1083 – 1100, 1990.
- [78] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. *Industrial and Engineering Chemistry Research*, 33(9):2123 – 2133, 1994.
- [79] J. Villadsen and M. Michelsen. *Solution of differential equation models by polynomial approximation*. Prentice-Hall, 1978.
- [80] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 16(1):32 – 48, 2006.
- [81] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25 – 57, 2006.

- [82] Y. Wang, H. Seki, S. Ooyama, K. Akamatsu, M. Ogawa, and M. Ohshima. Non-linear predictive control for optimal grade transition of polymerization reactors. *Advanced Control of Chemical Processes 2000 (ADCHEM 2000). Proceedings volume from the IFAC Symposium*, vol.3:701 – 6, 2001.
- [83] N. Wirth. What can we do about the unnecessary diversity of notation for syntactic definitions? Technical report, Communications of the ACM, 1977.
- [84] P. A. Wisnewski, F. J. Doyle III, and F. Kayihan. Fundamental continuous-pulp-digester model for simulation and control. *AIChE Journal*, 43(12):3175 – 3192, 1997.
- [85] L. T. Biegler Y. D. Lang, A. Cervantes. DynoPC: A dynamic optimization tool for process engineering. In *INFORMS National Meeting, Seattle, WA*, October, 1998.
- [86] A. C. Zanin, M. T. de Gouvêa, and D. Odloak. Integrating real-time optimization into the model predictive controller of the FCC system. *Control Engineering Practice*, 10:819–831, 2002.
-

Appendix A

Model of Kraft Paper Mill – Fiber Line

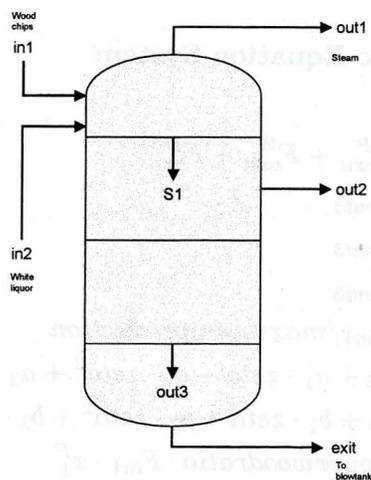


Figure A.1: Digester Model

Description: Digester Model, Digestion Department

A.1 Model - digester

Differential Variables

pulpout, chipusage, blackliquorproduction

Control Input Variables

$$F_{in1}$$

Algebraic Variables

$$a_{shrink1}, a_{shrink2}, zeta, F_{in2}, F_{in2}, F_{in2}, F_{out1}, F_{S1}, F_{S1}, F_{S1}, F_{out2}, F_{out2}, F_{out3}, F_{out3}, F_{out3}, F_{out3}, F_{exit}, F_{exit}^P, F_{exit}^W, F_{exit}^{DS}$$

Parameters

$$a_0=14.2390, a_{ST}=0.04, maximumproduction=80, liquorwoodratio=3.6, a_1=-3.9384, blowlinewaterfraction=0.62, a_3=0, a_2=0.3512, b_0=12.444, b_1=-5.2384, b_2=2.5357, b_3=-0.5588, x_1^P=0.43, x_1^{DS}=0.04, x_1^W=0.53$$

Differential and Algebraic Equation System

$$F_{exit} = F_{exit}^P + F_{exit}^W + F_{exit}^{DS} \quad (A.1)$$

$$F_{exit}^P = F_{out3} \quad (A.2)$$

$$F_{exit}^W = F_{out3} \quad (A.3)$$

$$F_{exit}^{DS} = F_{out3} \quad (A.4)$$

$$zeta = F_{in1}/maximumproduction \quad (A.5)$$

$$a_{shrink1} = a_0 + a_1 \cdot zeta + a_2 \cdot zeta^2 + a_3 \cdot zeta^3 \quad (A.6)$$

$$a_{shrink2} = b_0 + b_1 \cdot zeta + b_2 \cdot zeta^2 + b_3 \cdot zeta^3 \quad (A.7)$$

$$F_{in2} = liquorwoodratio \cdot F_{in1} \cdot x_1^P \quad (A.8)$$

$$F_{S1} = (1 - a_{shrink1}/100) \cdot F_{in1} \cdot x_1^P \quad (A.9)$$

$$F_{S1} = F_{in1} \cdot x_1^{DS} + F_{in2} + (a_{shrink1}/100) \cdot F_{in1} \cdot x_1^P \quad (A.10)$$

$$F_{S1} = F_{in1} \cdot x_1^W + F_{in2} - F_{out1} \quad (A.11)$$

$$F_{out1} = a_{ST} \cdot (F_{in1} \cdot x_1^W + F_{in2}) \quad (A.12)$$

$$F_{in2} = 0.788 \cdot (F_{in2} + F_{in2}) \quad (A.13)$$

$$F_{in2} = F_{in2} + F_{in2} \quad (A.14)$$

$$F_{out3} = (1 - a_{shrink2}/100) \cdot F_{S1} \quad (A.15)$$

$$F_{out3} = F_{S1} + (a_{shrink2}/100) \cdot F_{S1} - F_{out2} \quad (A.16)$$

$$F_{out3} = F_{S1} - F_{out2} \quad (A.17)$$

$$F_{out3} = blowlinewaterfraction \cdot F_{S1} \quad (A.18)$$

$$F_{out2} \cdot F_{out3} = F_{out2} \cdot F_{out3} \quad (A.19)$$

$$F_{out3} = F_{out3} + F_{out3} + F_{out3} \quad (A.20)$$

$$\frac{d}{dt}pulpout = F_{exit}^P \quad (A.21)$$

$$\frac{d}{dt}chipusage = F_{in1} \cdot x_1^P \quad (A.22)$$

$$\frac{d}{dt}blackliquorproduction = F_{out2} \quad (A.23)$$

Initial values

$$pulpout(0) = 0 \quad (A.24)$$

$$chipusage(0) = 0 \quad (A.25)$$

$$blackliquorproduction(0) = 0 \quad (A.26)$$

Constraints

$$F_{exit}^P \geq 0 \quad (A.27)$$

$$F_{exit}^W \geq 0 \quad (A.28)$$

$$F_{exit}^{DS} \geq 0 \quad (A.29)$$

$$0 \leq zeta \leq 1 \quad (A.30)$$

$$0 \leq F_{in1} \leq maximumproduction \quad (A.31)$$

$$F_{in2} \geq 0 \quad (A.32)$$

$$F_{in2} \geq 0 \quad (A.33)$$

$$F_{in2} \geq 0 \quad (A.34)$$

$$F_{out1} \geq 0 \quad (A.35)$$

$$F_{S1} \geq 0 \quad (A.36)$$

$$F_{S1} \geq 0 \quad (A.37)$$

$$F_{S1} \geq 0 \quad (A.38)$$

$$F_{out2} \geq 0 \quad (A.39)$$

$$F_{out2} \geq 0 \quad (A.40)$$

$$F_{out3} \geq 0 \quad (A.41)$$

$$F_{out3} \geq 0 \quad (A.42)$$

$$F_{out3} \geq 0 \quad (A.43)$$

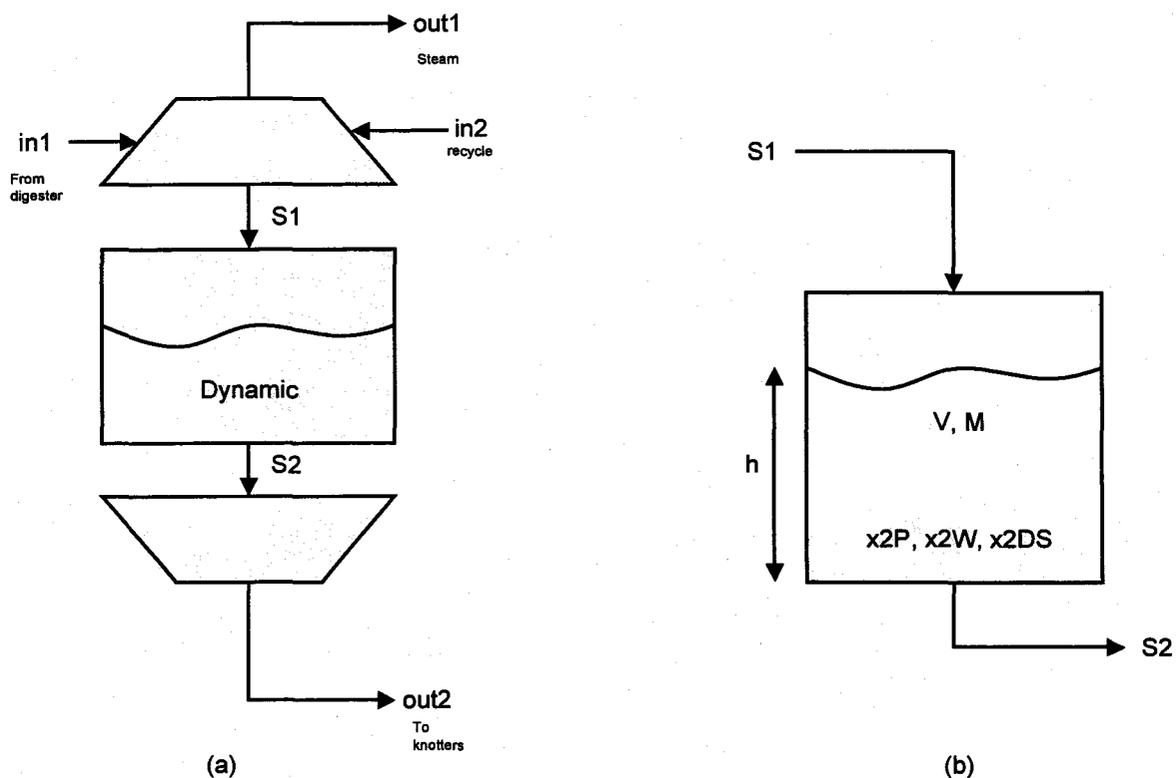


Figure A.2: (a) Schematic of integrated blowtank system; (b) Expanded diagram of the dynamic section of the blowtank.

Description: Tank Model (Blowtank and Tank)

A.2 Model - blowtank

Differential Variables

$$M, x_2^P, x_2^{DS}$$

Control Input Variables

$$F_{S2}$$

Algebraic Variables

$$F_{in1}, F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}, F_{in2}^W, F_{in2}^{DS}, F_{out1}, F_{out2}, F_{out2}^P, F_{out2}^W, F_{out2}^{DS}, F_{S1}, F_{S1}^P, F_{S1}^W, F_{S1}^{DS}, x_1^P, x_1^{DS}, x_1^W, F_{S2}, F_{S2}^P, F_{S2}^W, F_{S2}^{DS}, x_2^W, V, h$$

Parameters

$A=100$, $\rho_P=0.600$, $\rho_{DS}=1.100$, $\text{steam fraction}=0.02$, $\rho_W=0.998$,
 $V_0=0.3*2050$, $\text{blow tank vol}=2050$, $\rho_{\text{Average}}=0.900$

Differential and Algebraic Equation System

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (\text{A.44})$$

$$100 \cdot x_1^P + 100 \cdot x_1^W + 100 \cdot x_1^{DS} = 100 \quad (\text{A.45})$$

$$F_{S1}^P = x_1^P \cdot F_{S1} \quad (\text{A.46})$$

$$F_{S1}^{DS} = x_1^{DS} \cdot F_{S1} \quad (\text{A.47})$$

$$F_{S1}^W = x_1^W \cdot F_{S1} \quad (\text{A.48})$$

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (\text{A.49})$$

$$F_{in1}^P = F_{S1}^P \quad (\text{A.50})$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{S1}^W \quad (\text{A.51})$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{S1}^{DS} \quad (\text{A.52})$$

$$F_{out1}^W = \text{steam fraction} \cdot F_{in1}^W \quad (\text{A.53})$$

$$V \cdot \rho_{\text{Average}} = M \quad (\text{A.54})$$

$$V = A \cdot h \quad (\text{A.55})$$

$$\frac{d}{dt}M = F_{S1} - F_{S2} \quad (\text{A.56})$$

$$M \cdot \frac{d}{dt}x_2^P = F_{S1} \cdot (x_1^P - x_2^P) \quad (\text{A.57})$$

$$M \cdot \frac{d}{dt}x_2^{DS} = F_{S1} \cdot (x_1^{DS} - x_2^{DS}) \quad (\text{A.58})$$

$$100 \cdot x_2^P + 100 \cdot x_2^W + 100 \cdot x_2^{DS} = 100 \quad (\text{A.59})$$

$$F_{S2}^P = x_2^P \cdot F_{S2} \quad (\text{A.60})$$

$$F_{S2}^{DS} = x_2^{DS} \cdot F_{S2} \quad (\text{A.61})$$

$$F_{S2}^W = x_2^W \cdot F_{S2} \quad (\text{A.62})$$

$$F_{out2} = F_{out2}^P + F_{out2}^{DS} + F_{out2}^W \quad (\text{A.63})$$

$$F_{out2}^P = F_{S2}^P \quad (\text{A.64})$$

$$F_{out2}^W = F_{S2}^W \quad (\text{A.65})$$

$$F_{out2}^{DS} = F_{S2}^{DS} \quad (\text{A.66})$$

Initial values

$$V(0) = V_0 \quad (\text{A.67})$$

$$x_2^P(0) = x_1^P(0) \quad (\text{A.68})$$

$$x_2^{DS}(0) = x_1^{DS}(0) \quad (\text{A.69})$$

Constraints

$$F_{S2} \leq 456 \quad (\text{A.70})$$

$$0.1 \cdot \text{blowtankvol} \leq V \leq 0.9 \cdot \text{blowtankvol} \quad (\text{A.71})$$

$$M \geq 0 \quad (\text{A.72})$$

$$h \geq 0 \quad (\text{A.73})$$

$$0 \leq x_2^P \leq 1 \quad (\text{A.74})$$

$$0 \leq x_2^{DS} \leq 1 \quad (\text{A.75})$$

$$0 \leq x_2^W \leq 1 \quad (\text{A.76})$$

$$F_{out1}^W \geq 0 \quad (\text{A.77})$$

$$F_{out2} \geq 0 \quad (\text{A.78})$$

$$F_{out2}^P \geq 0 \quad (\text{A.79})$$

$$F_{out2}^W \geq 0 \quad (\text{A.80})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.81})$$

$$F_{S1} \geq 0 \quad (\text{A.82})$$

$$F_{S1}^P \geq 0 \quad (\text{A.83})$$

$$F_{S1}^W \geq 0 \quad (\text{A.84})$$

$$F_{S1}^{DS} \geq 0 \quad (\text{A.85})$$

$$0 \leq x_1^P \leq 1 \quad (\text{A.86})$$

$$0 \leq x_1^{DS} \leq 1 \quad (\text{A.87})$$

$$0 \leq x_1^W \leq 1 \quad (\text{A.88})$$

$$F_{S2} \geq 1 \times 10^{-8} \quad (\text{A.89})$$

$$F_{S2}^P \geq 0 \quad (\text{A.90})$$

$$F_{S2}^W \geq 0 \quad (\text{A.91})$$

$$F_{S2}^{DS} \geq 0 \quad (\text{A.92})$$

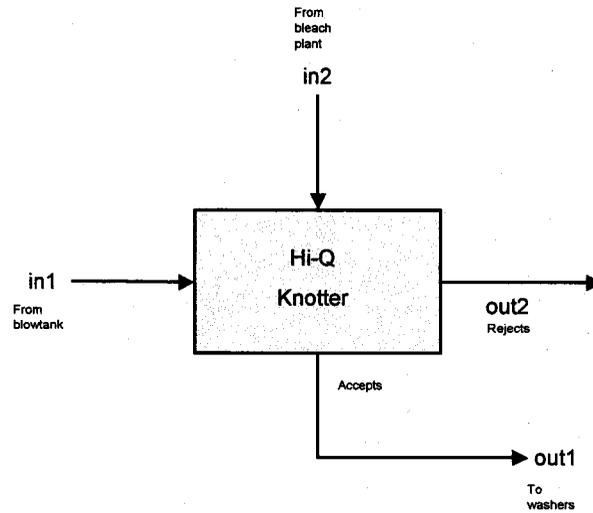


Figure A.3: Hi-Q knotter

Description: Hi-Q Knotter Model, Knotting Department

A.3 Model - hiq

Algebraic Variables

$$F_{in1}, F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}, F_{in2}^W, F_{in2}^{DS}, F_{out2}, F_{out2}^P, F_{out2}^W, F_{out2}^{DS}, F_{out1}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}$$

Parameters

$$amoist=0.101, aknotrej1=0.668, adil1=0.05, water\ fraction=0.95, adil2=0.10$$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (\text{A.93})$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (\text{A.94})$$

$$F_{out2} = F_{out2}^P + F_{out2}^W + F_{out2}^{DS} \quad (\text{A.95})$$

$$F_{out1} = F_{out1}^P + F_{out1}^W + F_{out1}^{DS} \quad (\text{A.96})$$

$$F_{in2} = \text{adil1} \cdot F_{in1} \quad (\text{A.97})$$

$$F_{out2}^P = \text{aknotrej1} \cdot F_{in1}^P \quad (\text{A.98})$$

$$F_{in1}^P = F_{out1}^P + F_{out2}^P \quad (\text{A.99})$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (\text{A.100})$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (\text{A.101})$$

$$100 \cdot F_{out1}^W \cdot F_{out2}^P = 100 \cdot F_{out1}^P \cdot F_{out2}^W \quad (\text{A.102})$$

$$100 \cdot F_{out2}^W \cdot F_{out1}^{DS} = 100 \cdot F_{out2}^{DS} \cdot F_{out1}^W \quad (\text{A.103})$$

$$F_{in2}^W = \text{water fraction} \cdot F_{in2} \quad (\text{A.104})$$

Constraints

$$F_{in1} \geq 0 \quad (\text{A.105})$$

$$F_{in1}^P \geq 0 \quad (\text{A.106})$$

$$F_{in1}^W \geq 0 \quad (\text{A.107})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.108})$$

$$F_{in2} \geq 0 \quad (\text{A.109})$$

$$F_{in2}^W \geq 0 \quad (\text{A.110})$$

$$F_{in2}^{DS} \geq 0 \quad (\text{A.111})$$

$$F_{out2} \geq 0 \quad (\text{A.112})$$

$$F_{out2}^P \geq 0 \quad (\text{A.113})$$

$$F_{out2}^W \geq 0 \quad (\text{A.114})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.115})$$

$$F_{out1}^P \geq 0 \quad (\text{A.116})$$

$$F_{out1}^W \geq 0 \quad (\text{A.117})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.118})$$

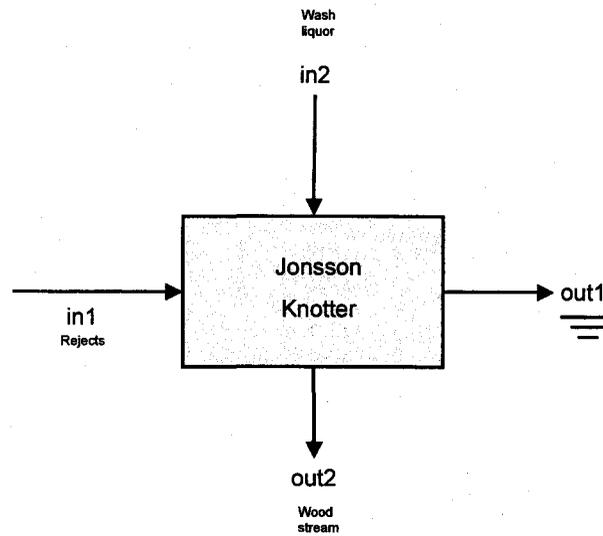


Figure A.4: Jonsson knotter

Description: Jonsson Knotter Model, Knotting Department

A.4 Model - jonsson

Algebraic Variables

$$F_{in1}, F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}, F_{in2}^W, F_{in2}^{DS}, F_{out1}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, F_{out2}, F_{out2}^P, F_{out2}^W, F_{out2}^{DS}$$

Parameters

$$aknotrej2=0.05, amoist=0.101, water\ fraction=0.95, adil2=0.10$$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (\text{A.119})$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (\text{A.120})$$

$$F_{out1} = F_{out1}^P + F_{out1}^W + F_{out1}^{DS} \quad (\text{A.121})$$

$$F_{out2} = F_{out2}^P + F_{out2}^W + F_{out2}^{DS} \quad (\text{A.122})$$

$$F_{in2} = \text{adil2} \cdot F_{in1} \quad (\text{A.123})$$

$$F_{out1}^P = \text{aknotrej2} \cdot F_{in1}^P \quad (\text{A.124})$$

$$F_{in1}^P = F_{out1}^P + F_{out2}^P \quad (\text{A.125})$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (\text{A.126})$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (\text{A.127})$$

$$F_{out1}^W + F_{out1}^{DS} = \text{amoist} \cdot (F_{out1}^P + F_{out1}^{DS} + F_{out1}^W) \quad (\text{A.128})$$

$$100 \cdot F_{out1}^W \cdot F_{out2}^{DS} = 100 \cdot F_{out1}^{DS} \cdot F_{out2}^W \quad (\text{A.129})$$

$$F_{in2}^W = \text{water fraction} \cdot F_{in2} \quad (\text{A.130})$$

Constraints

$$F_{out2} \geq 0 \quad (\text{A.131})$$

$$F_{out2}^P \geq 0 \quad (\text{A.132})$$

$$F_{out2}^W \geq 0 \quad (\text{A.133})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.134})$$

$$F_{in1} \geq 0 \quad (\text{A.135})$$

$$F_{in1}^P \geq 0 \quad (\text{A.136})$$

$$F_{in1}^W \geq 0 \quad (\text{A.137})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.138})$$

$$F_{out1} \geq 0 \quad (\text{A.139})$$

$$F_{out1}^P \geq 0 \quad (\text{A.140})$$

$$F_{out1}^W \geq 0 \quad (\text{A.141})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.142})$$

$$F_{in2} \geq 0 \quad (\text{A.143})$$

$$F_{in2}^W \geq 0 \quad (\text{A.144})$$

$$F_{in2}^{DS} \geq 0 \quad (\text{A.145})$$

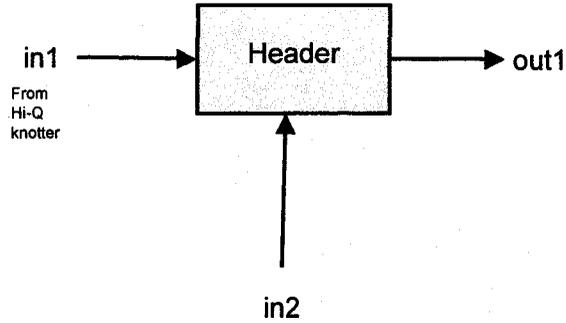


Figure A.5: Header box

Description: Header Box Model, Washing department

A.5 Model - header

Algebraic Variables

$$F_{in1}, F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}^W, F_{in2}^{DS}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, outletconsistency$$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (A.146)$$

$$F_{in1}^P = F_{out1}^P \quad (A.147)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W \quad (A.148)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} \quad (A.149)$$

$$F_{out1}^P = outletconsistency \cdot (F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) \quad (A.150)$$

Constraints

$$F_{in1}^P \geq 0 \quad (\text{A.151})$$

$$F_{in1}^P \geq 0 \quad (\text{A.152})$$

$$F_{in1}^W \geq 0 \quad (\text{A.153})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.154})$$

$$F_{out1}^P \geq 0 \quad (\text{A.155})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.156})$$

$$F_{out1}^W \geq 0 \quad (\text{A.157})$$

$$F_{in2}^W \geq 0 \quad (\text{A.158})$$

$$F_{in2}^{DS} \geq 0 \quad (\text{A.159})$$

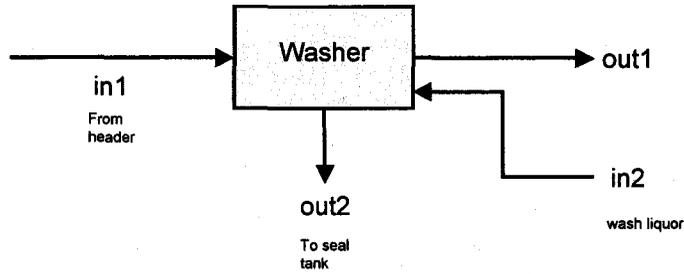


Figure A.6: Vacuum Drum Unit washer

Description: Vacuum Drum Washer Model, Washing Department

A.6 Model - drumwasher

Algebraic Variables

$F_{in1}, F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}, F_{in2}^P, F_{in2}^W, F_{in2}^{DS}, F_{out1}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, F_{out2}, F_{out2}^P, F_{out2}^W, F_{out2}^{DS}, outletconsistency,$
shower fraction, DR, R, W, x_b, x_d, y_c

Parameters

$nominalconsistency=0.12, N_{eff}=3$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (A.160)$$

$$F_{in1}^P = F_{out1}^P \quad (A.161)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (A.162)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (A.163)$$

$$(F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) = F_{out1}^P / outletconsistency \quad (A.164)$$

$$F_{in2}^{DS} = shower\ fraction \cdot (F_{in2}^W + F_{in2}^{DS}) \quad (A.165)$$

$$F_{out1}^W \cdot (F_{out2}^W + F_{out2}^{DS}) = F_{out2}^W \cdot (F_{out1}^W + F_{out1}^{DS}) \quad (A.166)$$

$$R \cdot (F_{out1}^W + F_{out1}^{DS}) = F_{in2}^W + F_{in2}^{DS} \quad (A.167)$$

$$W \cdot (F_{in1}^W + F_{in1}^{DS}) = F_{out2}^W + F_{out2}^{DS} \quad (A.168)$$

$$100 \cdot DR \cdot (W \cdot R^{Neff} - 1) = 100 \cdot (W \cdot R^{Neff} - R) \quad (A.169)$$

$$100 \cdot DR \cdot (x_b - y_c) = 100 \cdot (x_b - x_d) \quad (A.170)$$

$$x_d \cdot (F_{out1}^{DS} + F_{out1}^W) = F_{out1}^{DS} \quad (A.171)$$

$$x_b \cdot (F_{in1}^{DS} + F_{in1}^W) = F_{in1}^{DS} \quad (A.172)$$

$$y_c = \text{shower fraction} \quad (A.173)$$

Constraints

$$F_{in1}^P \geq 0 \quad (A.174)$$

$$F_{in1}^{DS} \geq 0 \quad (A.175)$$

$$F_{in1}^W \geq 0 \quad (A.176)$$

$$F_{out1}^P \geq 0 \quad (A.177)$$

$$F_{out1}^W \geq 0 \quad (A.178)$$

$$F_{out1}^{DS} \geq 0 \quad (A.179)$$

$$F_{out2}^W \geq 0 \quad (A.180)$$

$$F_{out2}^{DS} \geq 0 \quad (A.181)$$

$$F_{in2}^W \geq 0 \quad (A.182)$$

$$F_{in2}^{DS} \geq 0 \quad (A.183)$$

$$F_{in2}^W \leq 6000 \quad (A.184)$$

$$DR \geq 0 \quad (A.185)$$

$$0 \leq R \leq 40 \quad (A.186)$$

$$0 \leq W \leq 40 \quad (A.187)$$

$$0 \leq x_b \leq 1 \quad (A.188)$$

$$0 \leq x_d \leq 1 \quad (A.189)$$

$$0 \leq y_c \leq 1 \quad (A.190)$$

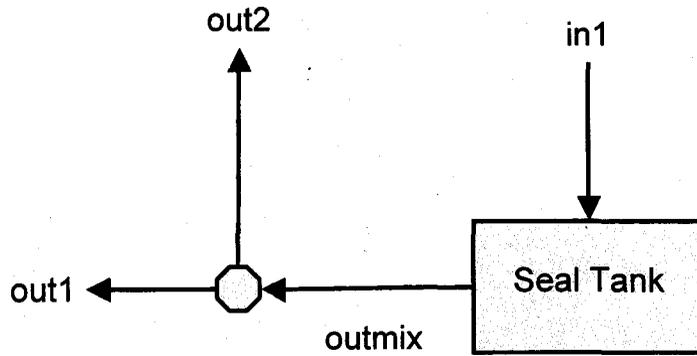


Figure A.7: Seal Tank

Description: Seal Tank Model, Washing Department

A.7 Model - sealtank

Differential Variables

$$M, x_W$$

Control Input Variables

$$F_{outmix}$$

Algebraic Variables

$$V, x_{inW}, x_{DS}, F_{in1}, F_{in1}^W, F_{in1}^{DS}, F_{outmix}^W, F_{outmix}^{DS}, F_{out1}, F_{out1}^W, F_{out1}^{DS}, F_{out2}, F_{out2}^W, F_{out2}^{DS}$$

Parameters

$$M_0=133, \rho=1.049$$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^W + F_{in1}^{DS} \quad (\text{A.191})$$

$$F_{outmix} = F_{outmix}^W + F_{outmix}^{DS} \quad (\text{A.192})$$

$$x_{inW} \cdot F_{in1} = F_{in1}^W \quad (\text{A.193})$$

$$F_{outmix}^W = x_W \cdot F_{outmix} \quad (\text{A.194})$$

$$V = \rho \cdot M \quad (\text{A.195})$$

$$\frac{d}{dt} M = F_{in1} - F_{outmix} \quad (\text{A.196})$$

$$M \cdot \frac{d}{dt} x_W = F_{in1} \cdot (x_{inW} - x_W) \quad (\text{A.197})$$

$$100 \cdot (x_W + x_{DS}) = 100 \quad (\text{A.198})$$

$$F_{outmix}^W = F_{out1}^W + F_{out2}^W \quad (\text{A.199})$$

$$F_{outmix}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (\text{A.200})$$

$$F_{out2}^W \cdot (F_{out1}^W + F_{out1}^{DS}) = F_{out1}^W \cdot (F_{out2}^W + F_{out2}^{DS}) \quad (\text{A.201})$$

Initial values

$$M(0) = M_0 \quad (\text{A.202})$$

$$x_W(0) = x_{inW}(0) \quad (\text{A.203})$$

Constraints

$$20 \leq M \leq 266 \quad (\text{A.204})$$

$$F_{outmix} \leq 500 \quad (\text{A.205})$$

$$F_{outmix}^W \geq 0 \quad (\text{A.206})$$

$$F_{outmix}^{DS} \geq 0 \quad (\text{A.207})$$

$$x_{inW} \geq 0 \quad (\text{A.208})$$

$$x_{DS} \geq 0 \quad (\text{A.209})$$

$$F_{in1}^W \geq 0 \quad (\text{A.210})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.211})$$

$$F_{out1}^W \geq 0 \quad (\text{A.212})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.213})$$

$$F_{out2}^W \geq 0 \quad (\text{A.214})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.215})$$

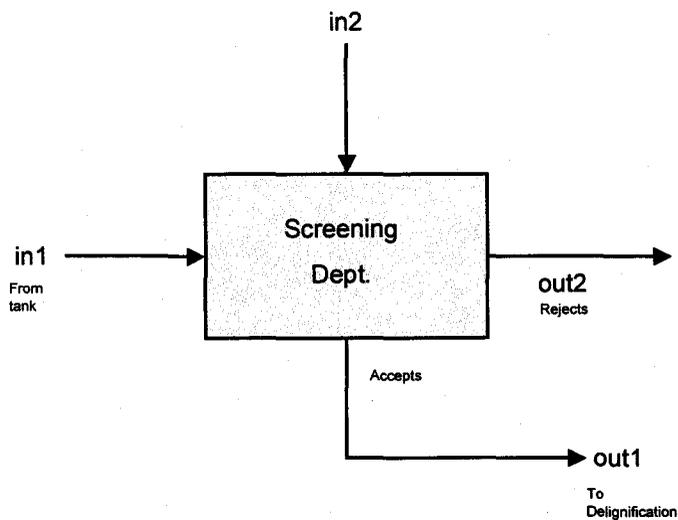


Figure A.8: Screening Department

Description: Screening Unit Model, Screening Department

A.8 Model - fullscreen

Algebraic Variables

$$F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}^W, F_{in2}^{DS}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, F_{out2}^P, F_{out2}^W, F_{out2}^{DS}$$

Parameters

$$outletconsistency=0.045, water\ fraction=0.8, pulplosscoeff=0.95$$

Differential and Algebraic Equation System

$$F_{in1}^P = F_{out1}^P + F_{out2}^P \quad (\text{A.216})$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (\text{A.217})$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (\text{A.218})$$

$$F_{out1}^P \cdot F_{out2}^W - F_{out1}^W \cdot F_{out2}^P = 0 \quad (\text{A.219})$$

$$F_{out1}^P \cdot F_{out2}^{DS} - F_{out1}^{DS} \cdot F_{out2}^P = 0 \quad (\text{A.220})$$

$$F_{out1}^P = \text{pulplosscoeff} \cdot F_{in1}^P \quad (\text{A.221})$$

$$\text{outletconsistency} \cdot (F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) = F_{out1}^P \quad (\text{A.222})$$

$$\text{water fraction} \cdot (F_{in2}^W + F_{in2}^{DS}) = F_{in2}^W \quad (\text{A.223})$$

Constraints

$$F_{in1}^P \geq 0 \quad (\text{A.224})$$

$$F_{in1}^W \geq 0 \quad (\text{A.225})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.226})$$

$$F_{in2}^W \geq 0 \quad (\text{A.227})$$

$$F_{in2}^{DS} \geq 0 \quad (\text{A.228})$$

$$F_{out1}^P \geq 0 \quad (\text{A.229})$$

$$F_{out1}^W \geq 0 \quad (\text{A.230})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.231})$$

$$F_{out2}^P \geq 0 \quad (\text{A.232})$$

$$F_{out2}^W \geq 0 \quad (\text{A.233})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.234})$$

$$\text{outletconsistency} \geq 0 \quad (\text{A.235})$$

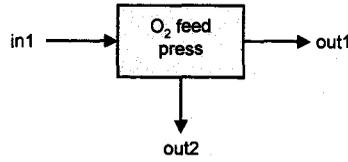


Figure A.9: O₂ Feed Press

Description: Oxygen Feedpress Model, Delignification Department

A.9 Model - o2feedpress

Algebraic Variables

$$F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, F_{out2}^W, F_{out2}^{DS}$$

Differential and Algebraic Equation System

$$F_{in1}^P = F_{out1}^P \quad (\text{A.236})$$

$$F_{in1}^W = F_{out1}^W + F_{out2}^W \quad (\text{A.237})$$

$$F_{in1}^{DS} = F_{out1}^{DS} + F_{out2}^{DS} \quad (\text{A.238})$$

$$F_{out1}^P = 0.3 \cdot (F_{out1}^P + F_{out1}^W + F_{out1}^{DS}) \quad (\text{A.239})$$

$$F_{out1}^W \cdot F_{out2}^{DS} = F_{out2}^W \cdot F_{out1}^{DS} \quad (\text{A.240})$$

Constraints

$$F_{in1}^P \geq 0 \quad (\text{A.241})$$

$$F_{in1}^W \geq 0 \quad (\text{A.242})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.243})$$

$$F_{out1}^P \geq 0 \quad (\text{A.244})$$

$$F_{out1}^W \geq 0 \quad (\text{A.245})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.246})$$

$$F_{out2}^W \geq 0 \quad (\text{A.247})$$

$$F_{out2}^{DS} \geq 0 \quad (\text{A.248})$$

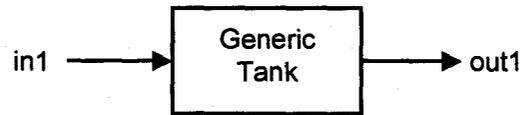


Figure A.10: Generic Tank

Description: Generic Tank Model, Delignification Department

A.10 Model - generictank

Algebraic Variables

$$F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}$$

Differential and Algebraic Equation System

$$F_{in1}^P = F_{out1}^P \quad (\text{A.249})$$

$$F_{in1}^W = F_{out1}^W \quad (\text{A.250})$$

$$F_{in1}^{DS} = F_{out1}^{DS} \quad (\text{A.251})$$

Constraints

$$F_{in1}^W \geq 0 \quad (\text{A.252})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.253})$$

$$F_{out1}^W \geq 0 \quad (\text{A.254})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.255})$$

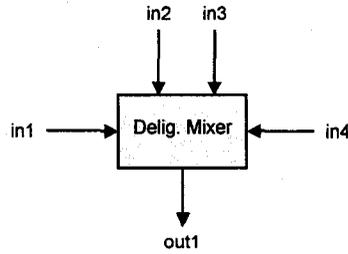


Figure A.11: Delignification mixer

Description: Mixer Model, Delignification Department

A.11 Model - deligmixer

Differential Variables

chemicalusage, steamusage

Algebraic Variables

$F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}^W, F_{in2}^{DS}, F_{in3}^W, F_{in3}^{DS}, F_{in4}^{ST}, F_{out1}^P, F_{out1}^W, F_{out1}^{DS}, F_{X0}, c_{pulp0}, F_{X1}, c_{pulp1}, L1, L2, L3, L4, R1$

Parameters

$T_{set}=100, NaOH_{dosage}=0.02, T_{inchem}=25, MgSO4_{dosage}=0.002, T_{ref}=25, H_{ref}=2547.3, H_{stm}=3267.5, c_{pwater}=4.18, c_{paper}=1.34, T_{inpulp}=25$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (A.256)$$

$$F_{in2} = F_{in2}^W + F_{in2}^{DS} \quad (A.257)$$

$$F_{in3} = F_{in3}^W + F_{in3}^{DS} \quad (A.258)$$

$$F_{out1} = F_{out1}^P + F_{out1}^W + F_{out1}^{DS} \quad (A.259)$$

$$F_{in1}^P = F_{out1}^P \quad (A.260)$$

$$F_{in1}^{DS} + F_{in2}^{DS} + F_{in3}^{DS} = F_{out1}^{DS} \quad (A.261)$$

$$F_{in1}^W + F_{in2}^W + F_{in3}^W + F_{in4}^{ST} = F_{out1}^W \quad (A.262)$$

$$L1 = 0 \quad (\text{A.263})$$

$$L2 = 0 \quad (\text{A.264})$$

$$L3 = 0 \quad (\text{A.265})$$

$$L4 = F_{in4}^{ST} \cdot (H_{stm} - H_{ref}) \quad (\text{A.266})$$

$$R1 = F_{out1} \cdot c_{ppulp1} \cdot (T_{set} - T_{ref}) \quad (\text{A.267})$$

$$(L1 + L2 + L3 + L4) \cdot 1 \times 10^{-4} = R1 \cdot 1 \times 10^{-4} \quad (\text{A.268})$$

$$100 \cdot F_{in2}^{DS} = 100 \cdot NaOH_{dosage} \cdot F_{in1}^P \quad (\text{A.269})$$

$$F_{in2}^W = 11.5 \cdot F_{in2}^{DS} \quad (\text{A.270})$$

$$100 \cdot F_{in3}^{DS} = 100 \cdot MgSO4_{dosage} \cdot F_{in1}^P \quad (\text{A.271})$$

$$F_{in3}^W = 21.222 \cdot F_{in1}^{DS} \quad (\text{A.272})$$

$$F_{X0} \cdot F_{in1} = 100 \cdot F_{in1}^P \quad (\text{A.273})$$

$$F_{X1} \cdot F_{out1} = 100 \cdot F_{out1}^P \quad (\text{A.274})$$

$$\frac{d}{dt} \text{chemicalusage} = F_{in2} \quad (\text{A.275})$$

$$\frac{d}{dt} \text{steamusage} = F_{in4}^{ST} \quad (\text{A.276})$$

Initial values

$$\text{chemicalusage}(0) = 0 \quad (\text{A.277})$$

$$\text{steamusage}(0) = 0 \quad (\text{A.278})$$

Constraints

$$F_{in1} \geq 0 \quad (\text{A.279})$$

$$F_{in1}^P \geq 0 \quad (\text{A.280})$$

$$F_{in1}^W \geq 0 \quad (\text{A.281})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.282})$$

$$F_{in2} \geq 0 \quad (\text{A.283})$$

$$F_{in2}^{DS} \geq 0 \quad (\text{A.284})$$

$$F_{in2}^W \geq 0 \quad (\text{A.285})$$

$$F_{in3} \geq 0 \quad (\text{A.286})$$

$$F_{in3}^{DS} \geq 0 \quad (\text{A.287})$$

$$F_{in3}^W \geq 0 \quad (\text{A.288})$$

$$F_{in4}^{ST} \geq 0 \quad (\text{A.289})$$

$$F_{out1} \geq 0 \quad (\text{A.290})$$

$$F_{out1}^P \geq 0 \quad (\text{A.291})$$

$$F_{out1}^W \geq 0 \quad (\text{A.292})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.293})$$

$$F_{X0} \geq 0 \quad (\text{A.294})$$

$$cppulp0 \geq 0 \quad (\text{A.295})$$

$$F_{X1} \geq 0 \quad (\text{A.296})$$

$$cppulp1 \geq 0 \quad (\text{A.297})$$

$$T_{set} \geq 0 \quad (\text{A.298})$$

$$T_{inchem} \geq 0 \quad (\text{A.299})$$

$$T_{inpulp} \geq 0 \quad (\text{A.300})$$

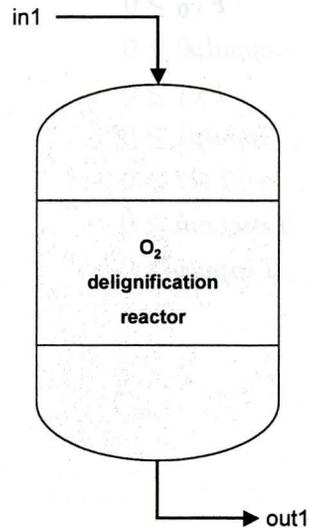


Figure A.12: Delignification Reactor

Description: Oxygen Delignification Reactor Model, Delignification Department

A.12 Model - reactor

Algebraic Variables

$$zeta, aO2, F_{in1}, F_{in1}^P, F_{in1}^{DS}, F_{in1}^W, F_{out1}^P, F_{out1}^{DS}, F_{out1}^W$$

Parameters

$$a1=-0.0022, a0=2.301, a3=-0.0113, a2=0.0116$$

Differential and Algebraic Equation System

$$F_{in1} = F_{in1}^P + F_{in1}^W + F_{in1}^{DS} \quad (A.301)$$

$$F_{out1}^P = (1 - 1 \times 10^{-2} \cdot aO2) \cdot F_{in1}^P \quad (A.302)$$

$$F_{out1}^{DS} = F_{in1}^{DS} + 1 \times 10^{-2} \cdot aO2 \cdot F_{in1}^P \quad (A.303)$$

$$F_{out1}^W = F_{in1}^W \quad (A.304)$$

$$aO2 = a0 + a1 \cdot zeta + a2 \cdot zeta^2 + a3 \cdot zeta^3 \quad (A.305)$$

Constraints

$$zeta \geq 0 \quad (\text{A.306})$$

$$aO2 \geq 0 \quad (\text{A.307})$$

$$F_{in1}^P \geq 0 \quad (\text{A.308})$$

$$F_{in1}^{DS} \geq 0 \quad (\text{A.309})$$

$$F_{in1}^W \geq 0 \quad (\text{A.310})$$

$$F_{out1}^P \geq 0 \quad (\text{A.311})$$

$$F_{out1}^{DS} \geq 0 \quad (\text{A.312})$$

$$F_{out1}^W \geq 0 \quad (\text{A.313})$$

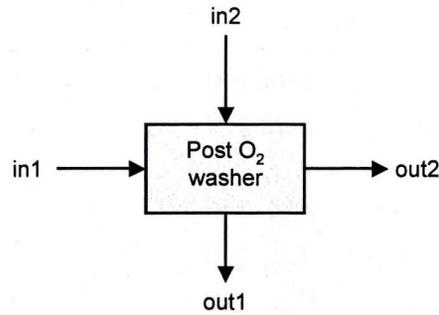


Figure A.13: Post O₂ Washer

Description: Post Oxygen Washer Model, Delignification Department

A.13 Model - postto2washer

Differential Variables

$totalpulpout$

Algebraic Variables

$F_{in1}^P, F_{in1}^W, F_{in1}^{DS}, F_{in2}^W, F_{in2}^{DS}, F_{out1}^W, F_{out1}^{DS}, F_{out2}^P, F_{out2}^W$

Parameters

$outletconsistency=0.10, adil=0.05, waterfraction=0.98$

Differential and Algebraic Equation System

$$F_{in1}^P = F_{out2}^P \quad (A.314)$$

$$F_{in1}^W + F_{in2}^W = F_{out1}^W + F_{out2}^W \quad (A.315)$$

$$F_{in1}^{DS} + F_{in2}^{DS} = F_{out1}^{DS} \quad (A.316)$$

$$F_{out2}^P = outletconsistency \cdot (F_{out2}^P + F_{out2}^W) \quad (A.317)$$

$$F_{in2}^W + F_{in2}^{DS} = adil \cdot (F_{in1}^P + F_{in1}^W + F_{in1}^{DS}) \quad (A.318)$$

$$F_{in2}^W = waterfraction \cdot (F_{in2}^W + F_{in2}^{DS}) \quad (A.319)$$

$$\frac{d}{dt}totalpulpout = F_{out2}^P \quad (A.320)$$

Initial values

$$totalpulpout(0) = 0 \quad (A.321)$$

Constraints

$$F_{in1}^P \geq 0 \quad (A.322)$$

$$F_{in1}^W \geq 0 \quad (A.323)$$

$$F_{in1}^{DS} \geq 0 \quad (A.324)$$

$$F_{in2}^W \geq 0 \quad (A.325)$$

$$F_{in2}^{DS} \geq 0 \quad (A.326)$$

$$F_{out1}^W \geq 0 \quad (A.327)$$

$$F_{out1}^{DS} \geq 0 \quad (A.328)$$

$$F_{out2}^P \geq 0 \quad (A.329)$$

$$F_{out2}^W \geq 0 \quad (A.330)$$

Connections

Description: Material stream connections

$digester(1).F_{exit}^P$	=	$blowtank(1).F_{in1}^P$	(A.331)
$digester(1).F_{exit}^{DS}$	=	$blowtank(1).F_{in1}^{DS}$	(A.332)
$digester(1).F_{exit}^W$	=	$blowtank(1).F_{in1}^W$	(A.333)
$blowtank(1).F_{out2}^P$	=	$hiq(1).F_{in1}^P$	(A.334)
$blowtank(1).F_{out2}^W$	=	$hiq(1).F_{in1}^W$	(A.335)
$blowtank(1).F_{out2}^{DS}$	=	$hiq(1).F_{in1}^{DS}$	(A.336)
$hiq(1).F_{out2}^P$	=	$jonsson(1).F_{in1}^P$	(A.337)
$hiq(1).F_{out2}^W$	=	$jonsson(1).F_{in1}^W$	(A.338)
$hiq(1).F_{out2}^{DS}$	=	$jonsson(1).F_{in1}^{DS}$	(A.339)
$hiq(1).F_{out1}^P$	=	$header(1).F_{in1}^P$	(A.340)
$hiq(1).F_{out1}^W$	=	$header(1).F_{in1}^W$	(A.341)
$hiq(1).F_{out1}^{DS}$	=	$header(1).F_{in1}^{DS}$	(A.342)
$header(1).outletconsistency$	=	0.02	(A.343)
$sealtank(1).F_{out2}^W$	=	$header(1).F_{in2}^W$	(A.344)
$sealtank(1).F_{out2}^{DS}$	=	$header(1).F_{in2}^{DS}$	(A.345)
$sealtank(1).F_{out1}^W$	=	$blowtank(1).F_{in2}^W$	(A.346)
$sealtank(1).F_{out1}^{DS}$	=	$blowtank(1).F_{in2}^{DS}$	(A.347)
$header(1).F_{out1}^P$	=	$drumwasher(1).F_{in1}^P$	(A.348)
$header(1).F_{out1}^W$	=	$drumwasher(1).F_{in1}^W$	(A.349)
$header(1).F_{out1}^{DS}$	=	$drumwasher(1).F_{in1}^{DS}$	(A.350)
$drumwasher(1).outletconsistency$	=	$drumwasher(1).nominalconsistency$	(A.351)
$drumwasher(1).showerfraction$	=	0.02	(A.352)
$drumwasher(1).F_{out2}^W$	=	$sealtank(1).F_{in1}^W$	(A.353)
$drumwasher(1).F_{out2}^{DS}$	=	$sealtank(1).F_{in1}^{DS}$	(A.354)
$drumwasher(1).F_{out1}^P$	=	$blowtank(2).F_{in1}^P$	(A.355)
$drumwasher(1).F_{out1}^{DS}$	=	$blowtank(2).F_{in1}^{DS}$	(A.356)
$drumwasher(1).F_{out1}^W$	=	$blowtank(2).F_{in1}^W$	(A.357)
$blowtank(2).F_{in2}^W$	=	0	(A.358)
$blowtank(2).F_{in2}^{DS}$	=	0	(A.359)
$blowtank(2).F_{in3}^P$	=	0	(A.360)
$blowtank(2).F_{in3}^W$	=	0	(A.361)
$blowtank(2).F_{in3}^{DS}$	=	0	(A.362)

$$\begin{aligned}
\text{blowtank}(2).F_{out2}^P &= \text{fullscreen}(1).F_{in1}^P & (\text{A.363}) \\
\text{blowtank}(2).F_{out2}^W &= \text{fullscreen}(1).F_{in1}^W & (\text{A.364}) \\
\text{blowtank}(2).F_{out2}^{DS} &= \text{fullscreen}(1).F_{in1}^{DS} & (\text{A.365}) \\
\text{fullscreen}(1).F_{out1}^P &= \text{o2feedpress}(1).F_{in1}^P & (\text{A.366}) \\
\text{fullscreen}(1).F_{out1}^W &= \text{o2feedpress}(1).F_{in1}^W & (\text{A.367}) \\
\text{fullscreen}(1).F_{out1}^{DS} &= \text{o2feedpress}(1).F_{in1}^{DS} & (\text{A.368}) \\
\text{generictank}(1).F_{in1}^P &= 0 & (\text{A.369}) \\
\text{o2feedpress}(1).F_{out2}^W &= \text{generictank}(1).F_{in1}^W & (\text{A.370}) \\
\text{o2feedpress}(1).F_{out2}^{DS} &= \text{generictank}(1).F_{in1}^{DS} & (\text{A.371}) \\
\text{o2feedpress}(1).F_{out1}^P &= \text{deligmixer}(1).F_{in1}^P & (\text{A.372}) \\
\text{o2feedpress}(1).F_{out1}^W &= \text{deligmixer}(1).F_{in1}^W & (\text{A.373}) \\
\text{o2feedpress}(1).F_{out1}^{DS} &= \text{deligmixer}(1).F_{in1}^{DS} & (\text{A.374}) \\
\text{reactor}(1).\text{zeta} &= \text{digester}(1).\text{zeta} & (\text{A.375}) \\
\text{deligmixer}(1).F_{out1}^P &= \text{reactor}(1).F_{in1}^P & (\text{A.376}) \\
\text{deligmixer}(1).F_{out1}^W &= \text{reactor}(1).F_{in1}^W & (\text{A.377}) \\
\text{deligmixer}(1).F_{out1}^{DS} &= \text{reactor}(1).F_{in1}^{DS} & (\text{A.378}) \\
\text{reactor}(1).F_{out1}^P &= \text{generictank}(2).F_{in1}^P & (\text{A.379}) \\
\text{reactor}(1).F_{out1}^W &= \text{generictank}(2).F_{in1}^W & (\text{A.380}) \\
\text{reactor}(1).F_{out1}^{DS} &= \text{generictank}(2).F_{in1}^{DS} & (\text{A.381}) \\
\text{generictank}(2).F_{out1}^P &= \text{postto2washer}(1).F_{in1}^P & (\text{A.382}) \\
\text{generictank}(2).F_{out1}^W &= \text{postto2washer}(1).F_{in1}^W & (\text{A.383}) \\
\text{generictank}(2).F_{out1}^{DS} &= \text{postto2washer}(1).F_{in1}^{DS} & (\text{A.384}) \\
\text{generictank}(3).F_{in1}^P &= 0 & (\text{A.385}) \\
\text{postto2washer}(1).F_{out1}^W &= \text{generictank}(3).F_{in1}^W & (\text{A.386}) \\
\text{postto2washer}(1).F_{out1}^{DS} &= \text{generictank}(3).F_{in1}^{DS} & (\text{A.387})
\end{aligned}$$

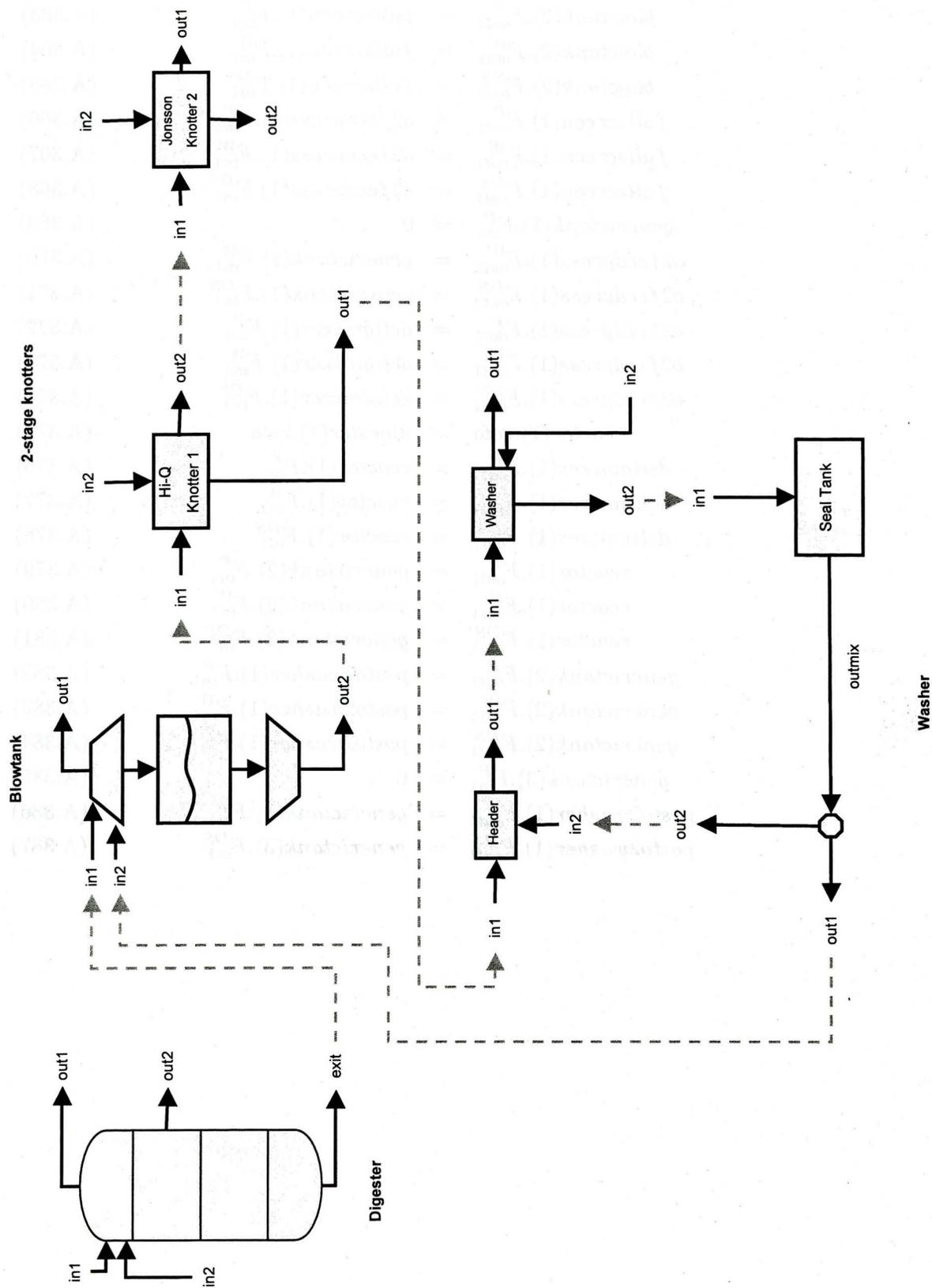


Figure A.14: Overall system up to washers

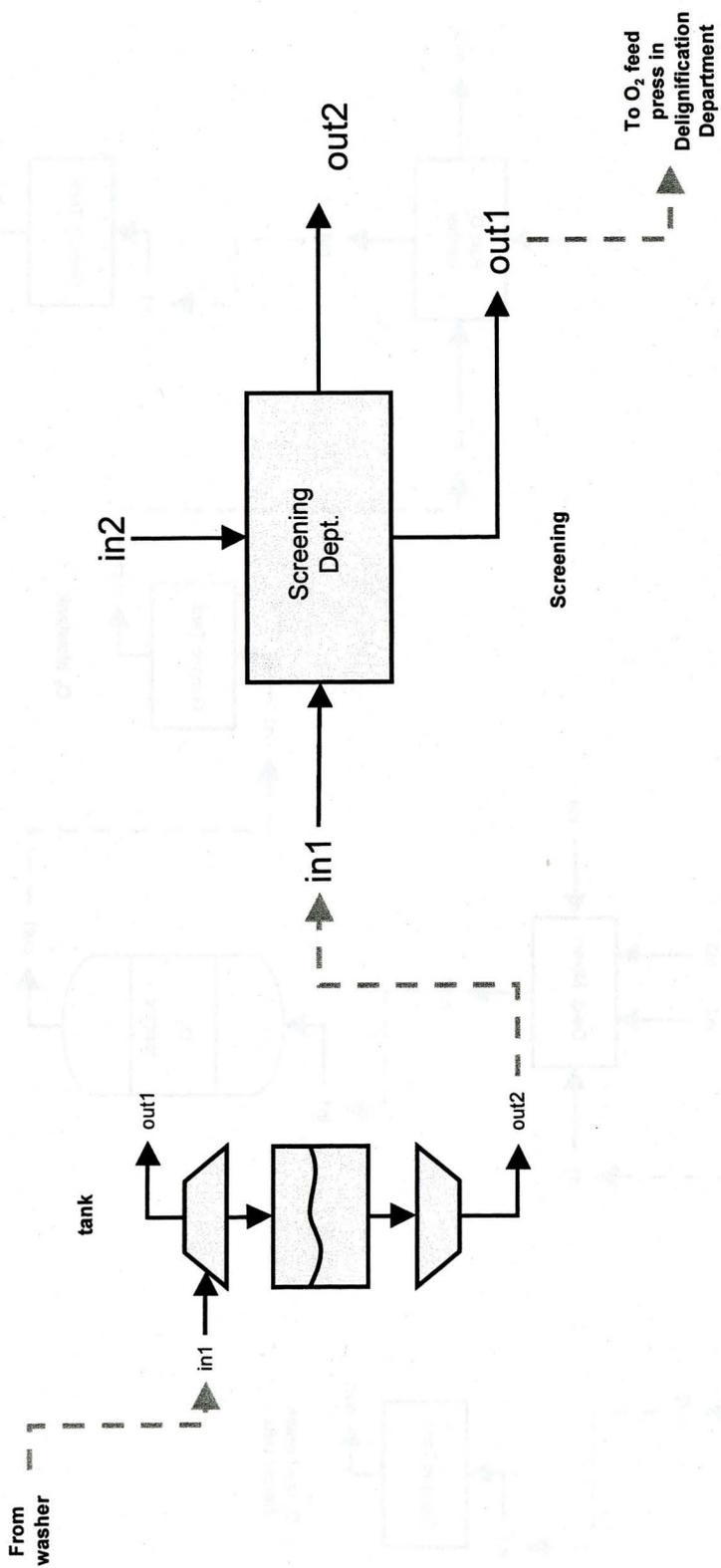


Figure A.15: Screening Department

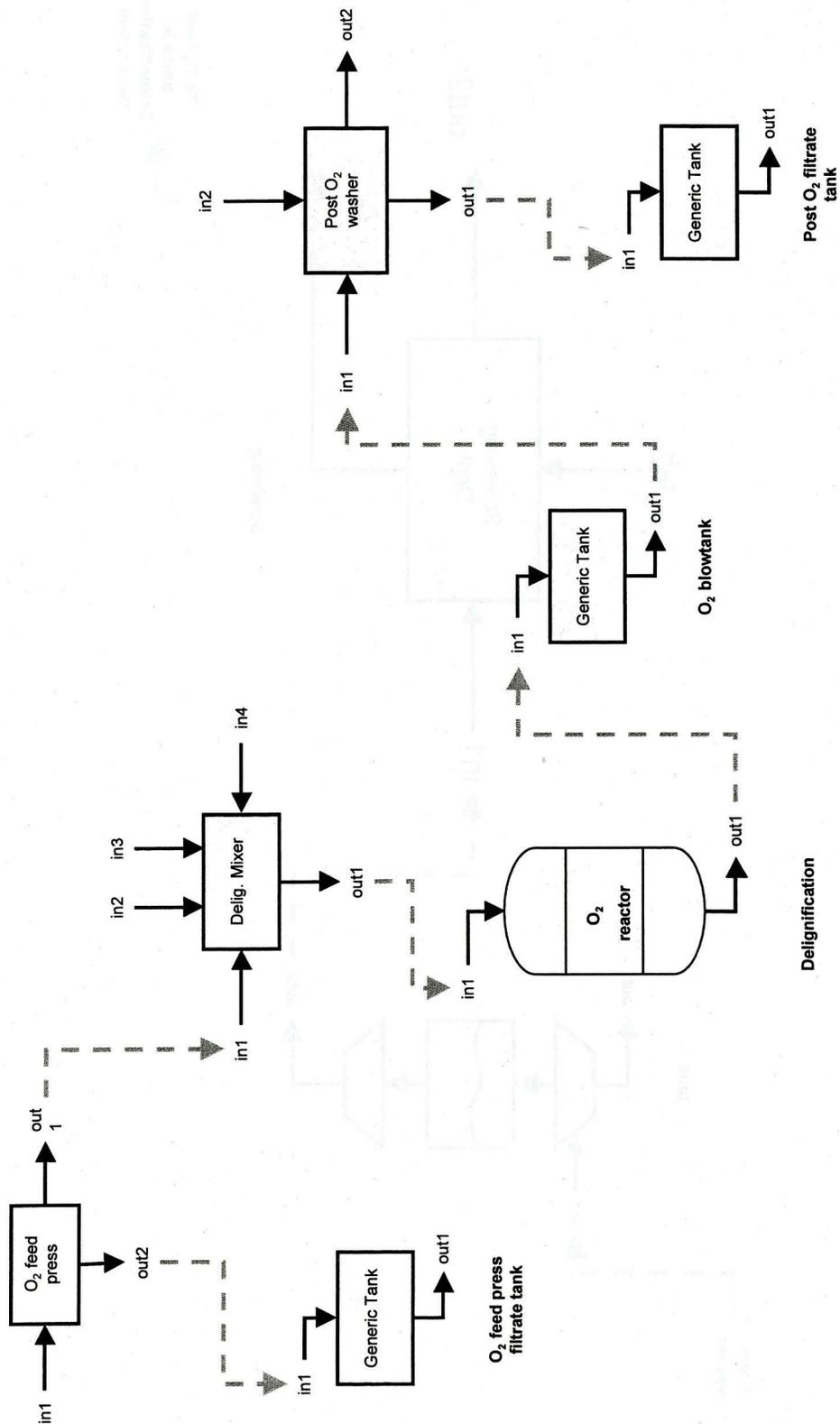


Figure A.16: Delignification

Appendix B

Characterizing the fixed cost of shutdowns in an NLP

B.1 Counting the number of shutdowns

A shutdown is characterized by the flows to a unit going below the minimum production flow, *i.e.* $F \leq F_{shut}$. Apart from a time-dependent variable cost (a quantity which is usually implicitly captured in the problem as lost production), every shutdown has an associated fixed cost that is independent of the duration of the shutdown. Each fixed cost is directly related to the exact number of shutdown instances. For instance, for every shutdown that occurs, there is a manpower/resource cost of starting the system up again, which is a fixed cost that is independent of the shutdown period.

In order to accurately represent these fixed costs in the optimization problem, it is necessary to come up with a way to count the number of shutdowns that occur. We propose the following method for counting the number of shutdowns that are actually occurring or are induced, and penalizing them appropriately. Note: this method only works in discrete time.

$$\max_{\mathbf{F}(k)} \left[\Phi_{economics} - \sum_i^{N_{units}} N_{i,shutdown} \cdot C_{i,shutdown} \right]$$

s.t.

$$\omega_i(k) = \begin{cases} 1, & \text{for } 0 \leq F_i(k) < F_{i,shut} \\ 0, & \text{for } F_i(k) \geq F_{i,shut} \end{cases} \quad (\text{B.1})$$

$$\omega_i(k) = \begin{cases} 1, & \text{for } 0 \leq F_i(k) < F_{i,shut} \\ 0, & \text{for } F_i(k) \geq F_{i,shut} \end{cases} \quad (\text{B.2})$$

$$\Delta\omega_i(k) = \omega_i(k+1) - \omega_i(k) \quad \forall k \in \{0..n_{samples} - 1\} \quad (\text{B.3})$$

$$N_{i,shutdown} = \frac{1}{2} \left[\omega_i(0) + \sum_{k=0}^{n_{samples}-1} [\Delta\omega_i(k)]^2 \right] \quad (\text{B.4})$$

for $i = 1..N_{units}$

where

k = discrete time index, where $t = k\Delta t$

$\mathbf{F}(k)$ = flowrate variable vector

$\Phi_{economic}$ = economics-based objective function

$N_{shutdown}$ = number of shutdowns in unit i

$C_{shutdown}$ = cost (\$) per shutdown in unit i

$n_{samples}$ = number of samples

N_{units} = number of process units

The advantage of this method is that it avoids the arbitrariness of assigning numerical weights, because the penalty term is a purely economic term.

In this work, we propose using the modified hyperbolic equations (B.7, B.8) below as a continuous approximation for ω_i .

A Modified Hyperbolic Tangent Switching Function

We investigated the use of a variant of the widely-known hyperbolic tangent switching function (also known as a parameterized logistic function), commonly used as a continuous approximation for a step function. This function behaves as follows: if a certain flowrate is at or below a value F_{shut} , the function activates and returns a value of 1, otherwise it returns a value of 0.

$$\omega_i = \left[-\frac{1}{2} \tanh[\gamma(F_i(k) - F_{i,shut})] + \frac{1}{2} \right] \quad \forall i \in \{1..N_{units}\} \quad (\text{B.5})$$

where

γ = tuning parameter, correlates with the sharpness of the switching interval (positive number)

The advantages of this type of function are:

1. Continuous, differentiable everywhere
2. Provides good approximation of switching.

Unfortunately, this function (and its gradient) is likely to cause numerical overflows in the optimizer. To show this, we rewrite equation function (B.5) as follows:

$$\omega_i = \left[\frac{1}{1 + \exp(2\gamma(F_i(k) - F_{i,shut}))} \right] \quad \forall i \in \{1..N_{units}\} \quad (\text{B.6})$$

In order to obtain a good approximation, it is necessary to choose a value of γ that is sufficiently high. A value of $\gamma = 30$ was found to be adequate for inducing a sharp switch. With nominal values for the expression $(F_i(k) - F_{i,shut})$ being in the range of 400 during steady-state, the exponential function value becomes $\exp(2 \cdot 30 \cdot 400)$, which is a number that is sufficiently big to cause a numerical overflow in most optimizers¹.

We note that the type of hyperbolic switching functions that are reported in literature as having been successfully applied are those of the positive variant (*i.e.* having a positive sign associated with the first $\frac{1}{2}$ in equation B.5). The positive sign leads to bounded exponential terms by virtue of the fact that the a negative number is exponentiated. In the above case however (equation B.6), the exponential term tends to infinity which leads to a numerical overflow when evaluated.

Therefore, in order to condition the function, we propose modifying the above switching function by replacing the expression $(F_i(k) - F_{i,shut})$ with another function, r_i whose range is a moderate $(-\pi/2, \pi/2)$ and intercepts the x-axis at $F_i(t) = F_{i,shut}$:

$$\omega_i = \left[-\frac{1}{2} \tanh(\gamma \cdot r_i) + \frac{1}{2} \right] \quad \forall i \in \{1..N_{units}\} \quad (\text{B.7})$$

$$r_i = \tan^{-1}(F_i(k) - F_{i,shut}) \quad (\text{B.8})$$

This new function has virtually the same behavior as the previous except that the evaluation of large exponential terms is bypassed in this case. It has to be noted that

¹The problem stems from the fact that numerical software are forced to adhere to the PEM-DAS (Parentheses Exponents Multiplication Division Addition Substraction) rule when evaluating expressions, thus the expression within the *exp* term is invariably evaluated first.

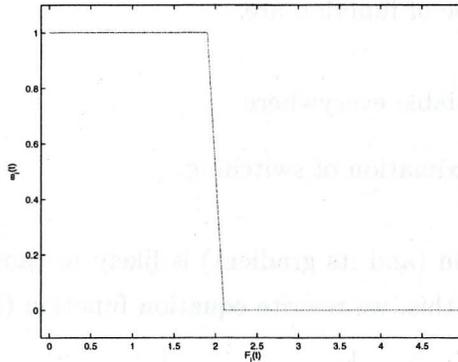


Figure B.1: Modified Hyperbolic Switching Function with arctangent function, given $\gamma = 50$ and $F_{shut} = 2$

there is one major disadvantage arising from the use of the arctangent function, that is, it introduces severe nonlinearities into the problem.

Derivation: Counting the number of shutdowns

Since fixed-costs only apply once per shutdown, we propose an algebraic formulation for counting the number of shutdowns within an NLP formulation. This formulation does not require integer variables, therefore it can be executed in a standard continuous NLP solver package. The logic is as follows:

1. First, we require a switching function that will yield a value of 1 when $F_i(k) \leq F_{i,shut}$ and 0 elsewhere. (Equations B.1 - B.2). In order to avoid integer variables, this discrete function is approximated by an appropriate continuous switching function. In our case, we chose to go with the modified hyperbolic switching function (Equations B.7 - B.8).
2. We then require another function that represents the number of times $F(k)$ changes its value from 0 to 1 and vice versa. (Equation B.3).
3. To count the number of shutdowns that occur, equation B.3 is squared (to discard all negative values) and the sum of the squared values over the time horizon is taken. Since a shutdown and subsequent startup requires two changes in value, this sum is divided by 2 to obtain the number of shutdowns.

Note: we also add $\omega(0)$ to account for the fact that a system may start at a shutdown state. If it the system starts from a state of shutdown, then $\omega(0) = 1$, else $\omega(0) = 0$. Thus, we arrive at the shutdown count equation, eqn (B.4)

Assumption: we assume that the final state of the system is a non-shutdown state, therefore the expression below evaluates to an even number, and is therefore divisible by 2.

$$\left(\omega_i(0) + \sum_{k=0}^{n_{samples}-1} [\Delta\omega_i(k)]^2 \right)$$

Example

To illustrate this, consider a trajectory $F(t)$ that touches F_{shut} exactly 4 times (Figure B.2). Using equations (B.1 - B.4), we obtain a value for $N_{shutdown}$ that represents the number of shutdowns that occur (see the $N_{shutdown}$ graph in Figure B.2, where $N_{shutdown} = 4$).

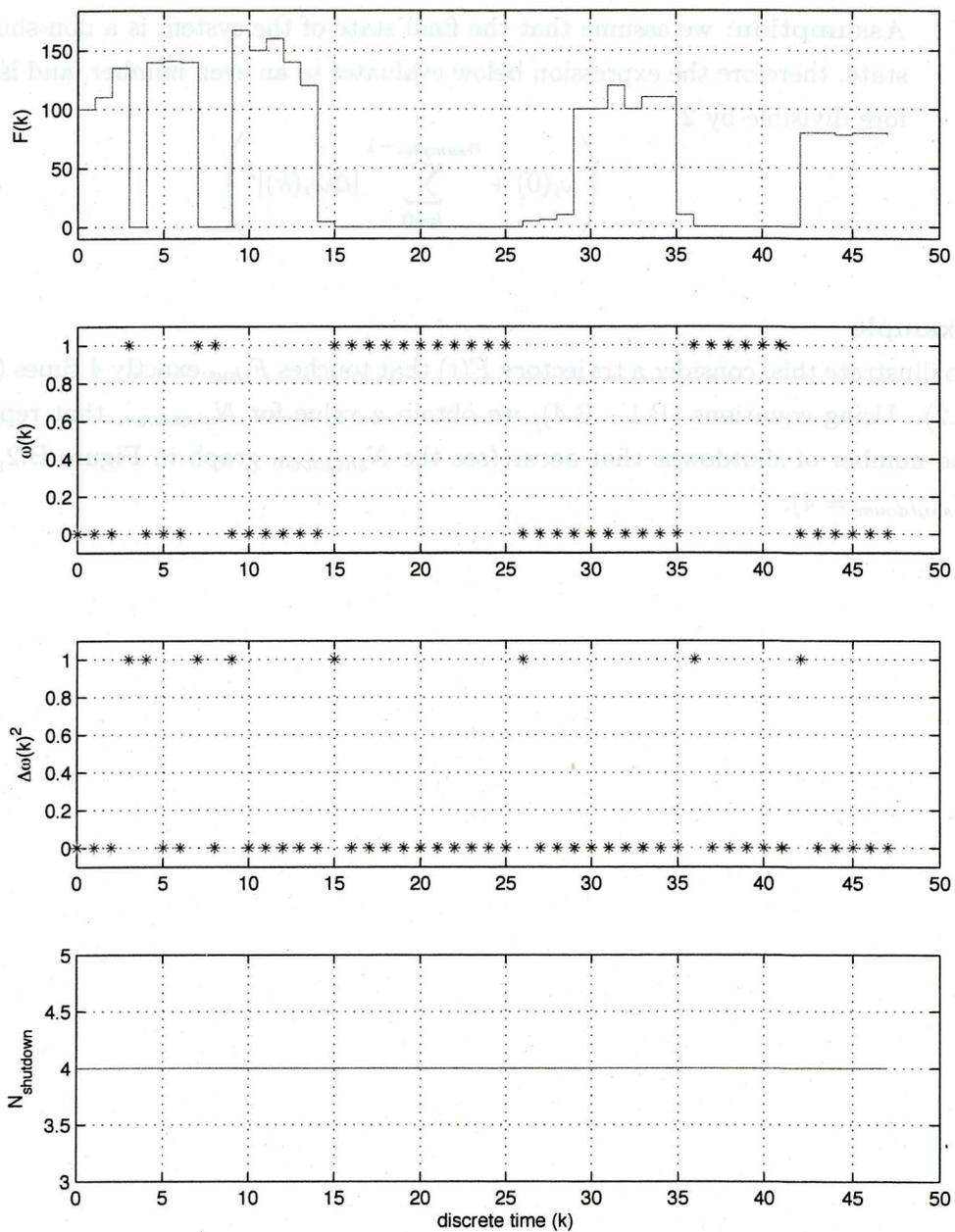


Figure B.2: Procedure for deriving constraints to count shutdowns